

Score-based Bayesian network structure learning algorithms for modeling radioisotope levels in nuclear power plant reactors

Pablo Ramirez-Hereza^{a,*}, Daniel Ramos^a, Doroteo T. Toledano^a, Joaquin Gonzalez-Rodriguez^a, Alicia Ariza-Velazquez^b, Nuria Doncel^b

^a AUDIAS Laboratory - Audio, Data Intelligence and Speech, Escuela Politecnica Superior, Universidad Autónoma de Madrid, Calle Francisco Tomás y Valiente 11, 28049 Madrid, Spain

^b ENUSA Industrias Avanzadas S. A., Calle Santiago Rusiñol, 12, 28040 Madrid, Spain

ARTICLE INFO

Keywords:

Nuclear chemistry
Primary loop chemistry
Bayesian networks
Probabilistic machine learning
Structure learning

ABSTRACT

Radioactive corrosion products released into the primary coolant loop dominate the final shutdown radiation fields of pressurized water reactors. Thus, reducing the concentration of these corrosion products is a paramount duty in the optimization process of the reactor performance. However, the complexity and uncertainty present in this process make it difficult to predict their evolution in a theoretical way. We propose the application of structural learning of Bayesian networks to discover the complex relations between the corrosion products and the most relevant variables in the primary loop, giving rise to probabilistic models that obtain accurate and reliable predictions of the corrosion products. Our analysis of 5 power plants demonstrates that our approach results in simpler and more reliable models. Additionally, we conclude that the learned structures may represent an interpretable tool for power plant technicians since they reveal useful information that can be directly employed to improve the reactor operation.

1. Introduction

The operation of a pressurized water reactor (PWR) [1] is based on two separate water circuits, which are generally called primary and secondary coolant loops. The electricity generation process carried out by these reactors is summarized in Fig. 1. As it is shown, in the primary loop, heat is extracted from the reactor core by streams of pressurized water. Inside the steam generator, this heat vaporizes the water in the secondary loop, generating steam. Then, the secondary loop steam produces electricity by driving a turbo generator. Finally, the unused steam is exhausted to the condenser, where it is converted into water and pumped back to the steam generator. The condenser is itself cooled by a so-called third water loop.

It is worth noting that heat transfer between primary and secondary loops is performed without mixing the water from both circuits. Thus, while the water of the secondary loop remains pure throughout the process, the primary loop water is recirculated with corrosion products and exposed to radiation as it passes through the nuclear reactor core. These corrosion products come from the surfaces of different components of the primary loop (such as the steam generator, the pipes, or the fuel itself), and they are constantly released into the reactor coolant system, either in a soluble or insoluble form. The radioactive corrosion

products dominating the shutdown radiation fields are generally the radiocobalts ^{60}Co and ^{58}Co , with lesser contributions from ^{51}Cr , ^{59}Fe , ^{54}Mn , and others.

Although the chemical and volume control system (CVCS) helps to minimize the amount of radioactive material in the coolant, the corrosion products deposited on the different surfaces of the primary coolant loop represent the major part of the radiation dose rates after an operation cycle [2]. Thus, in order to limit the corrosion products concentration, the chemistry staff may introduce chemical additives to, for example, adjust the pH in the primary loop, control the hydrogen, or implement Zn injection, among others. For this reason, these chemical variables are considered crucial for reducing the final radioactive waste, hence optimizing the reactor operation. On the one hand, pH is measured to control the corrosive and aggressive environment. Maintaining a high pH level by the introduction of chemical additives (lithium hydroxide and boron) decreases general corrosion. On the other hand, the addition of hydrogen suppresses water radiolysis and prevents the increase of corrosion rate. Nevertheless, an excess of hydrogen may cause material embrittlement. Finally, zinc injection has been implemented in some nuclear power plants because it has been found to function as a good radiation control measure, which reduces

* Corresponding author.

E-mail address: pablo.ramirez@estudiante.uam.es (P. Ramirez-Hereza).

URL: <http://audias.ii.uam.es> (P. Ramirez-Hereza).

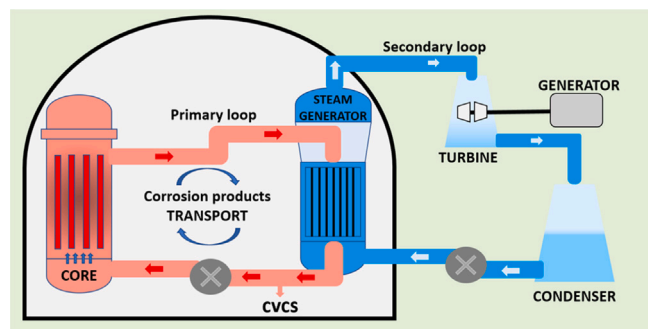


Fig. 1. Primary and secondary loops in a pressurized-water nuclear power plant reactor. Redrawn by ENUSA.

the concentration of radiocobalts in the primary water loop. Apart from these chemical variables, experts highlight two more parameters that may have a critical impact on the radioisotope levels: Thermal power in the reactor and the letdown flow. The letdown flow is defined as the percentage of the total coolant which is derived towards the aforementioned CVCS, to clean primary loop water.

Thus, the evolution of the concentration of the corrosion products, and more specifically the radiocobalts ^{58}Co and ^{60}Co , is the result of all these chemical processes, which are included in the so-called primary loop chemistry. Unfortunately, the high complexity and the uncertainty present in the process make it difficult to determine the radiocobalts evolution by theoretical models. It is therefore desirable to gather knowledge about the chemistry of the primary loop in two ways: first, modeling the relationships between the variables will increase our awareness of the internal interactions in the reactor during operation. Second, a more informed prediction model will be able to be more accurate in determining the amount of radioisotopes in the loop by the observation of the control variables. The resulting approaches are expected to have an impact in the field since they will allow the reduction of uncertainty about the behavior of the main corrosion products. In this way, the increase of knowledge implied by these models may allow the validation of new operational strategies to optimize operation, ensure material and personnel integrity, and reduce radiation dose at reactor shutdown.

However, the proposed approach presents several important challenges characterized by: the scarcity of data for training and evaluating the models, the remarkable uncertainty in the primary loop chemistry, as explained above, and the existence of a high-risk scenario, in which a missed prediction can lead to the degradation of the reactor facilities or personnel exposure to higher radiation doses. In this context, probabilistic machine learning models, specifically Bayesian networks, have demonstrated well behavior for the available data in this particular application [3]. These models present great advantages over other regression models. First, Bayesian networks are probabilistic models, they quantify the uncertainty present in the problem using probabilistic theory, in contrast with kernel-based approaches, like Support Vector Regressors (SVR) [4]. Second, these models are generative models, which means that they model the joint probability distribution of all the variables in a problem. This makes the Bayesian Networks flexible models as they allow to predict any variable of the model given an evidence. Third, they allow to interpret the relationships between the variables, in contrast with other models such as Gaussian Processes (GPs) [5]. Finally, they are less data-hungry than more complex models such as deep neural networks (as it has been demonstrated in [3] for this very same problem). Bayesian networks are already being used in other nuclear-related fields such as nuclear data evaluation [6], source term prediction in nuclear accidents [7], power plant safety analysis [8], or failure diagnosis [9].

In the primary loop chemistry scenario described above, [3] demonstrates the capabilities of Bayesian networks and Dynamic Bayesian

networks to accurately predict the concentration of the radioisotopes ^{58}Co , ^{60}Co and ^{51}Cr , from observed values of certain *Control* variables. Within the control variables considered, we find the *pH* in the primary loop, the thermal power of the reactor *P*, the letdown flow *L*, and the concentration of *H2* and *Zn*. Specifically, the model proposed is a Gaussian Bayesian network (GBN) whose structure is previously defined by the expert chemical knowledge of the authors. This model is considered the *Baseline* of this work, and it is described in Section 5. In fact, to perform the predictions in [3], the first step is fixing the model structure by expert knowledge, allowing the definition of a Bayesian network graph with a chemical interpretation. However, it also involves the definition of important assumptions considering conditional independencies between the variables of the problem. These assumptions may not be reflected in data due to the complexity of the problem and the presence of many factors that can be hidden or latent. This would lead to the generation of models that do not fit the empirical joint distribution of the variables.

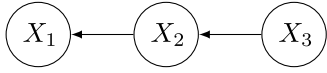
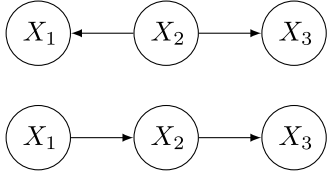
In addition to the generation of poorly calibrated models that lead to unreliable predictions, the definition of a fixed structure can cause the interpretability of Bayesian networks to become a double-edged sword, since it leads technicians to draw erroneous conclusions about the variables. For this reason, in order to infer the structure of variables in a data-driven way, this paper proposes the discovery of relationships using structure learning algorithms. In order to analyze the relationships between variables and measure the reliability of the generated structures, the obtained models are compared with the Static Bayesian Network defined in [3] in terms of generalization power and likelihood. Note that this manuscript aims at discovering relationships among variables in a given moment in time, and therefore, a static Bayesian network is more adequate than a dynamic Bayesian network, as the explored in [3].

To do this, we consider three classes of algorithms to learn the structure of Bayesian networks from data [10,11]. First, *constraint-based algorithms* (e.g., PC [12], GS [13]), which perform conditional independence tests in data and, then, they find the network that best explains these independencies. Second, *score-based algorithms* (e.g., *tabu search* [14] K2 [15], GES [16]), which use goodness-of-fit scores as objective functions to maximize. Finally, *hybrid algorithms* (e.g., H^2PC [17], MMHC [18]) that combine both approaches. According to the literature, score-based algorithms pose a search problem that may not have an elegant and efficient solution, however, they are generally more accurate and faster than constraint-based and hybrid algorithms.

For this reason, in this work we compare the performance of three widely known score-based algorithms in the primary loop chemistry scenario. First, the *Greedy Hill-Climbing* [13], which evaluates the score function over the space of possible candidates by the Hill-Climbing search algorithm. Second, the *K2 algorithm* [15], which assumes the topological ordering of the structure to be known a priori, reducing in this way the candidate structure space and the computational cost of the process. Finally, the *greedy equivalence search algorithm* [16], that defines an equivalence class space as the search space. Graphs resulting from these structure learning algorithms are expected to better fit the data than the baseline model, based solely on expert knowledge. In other words, the process of learning the set of conditional independencies present in data is expected to generate more calibrated and reliable Gaussian Bayesian networks than the Gaussian Bayesian network with a structure fixed by expert knowledge, as in [3].

2. Bayesian networks fundamentals

Probabilistic graphical models [10] use graph-based representations for compactly encoding a complex joint probability distribution of a set of random variables $\mathbf{X} = \{X_1, \dots, X_N\}$. In this graphical representation, also called *structure* or *graph*, nodes correspond to the variables in the problem while the edges correspond to direct probabilistic interactions between them, and encode conditional independence relationships.

Fig. 2. Example of directed acyclic graph G_1 with 3 variables X_1 , X_2 and X_3 .Fig. 3. I-equivalent structures of G_1 . Upside G_2 , downside G_3 .

Bayesian networks (BNs) are probabilistic graphical models whose structure is a *directed acyclic graph* (DAG) \mathcal{G} . This means that each edge has a source (parent) and a target (child), and there is not any circular path in the graph. Fig. 2 shows a simple example of a Bayesian network structure where X_3 is the parent node of X_2 , X_2 is the parent node of X_1 and therefore, X_1 is the child node of X_2 , and X_2 is the child node of X_3 .

The Bayesian network structure can be considered a compact representation of a set of conditional independence assumptions about the joint distribution of a problem and, equivalently, as a representation of the joint distribution in a factorized way. This factorization is defined by the so-called *chain rule* of a Bayesian network, which states that the joint probability distribution of the model variables can be obtained by the product of the conditional probability distribution (CPD) of each variable X_i given its parents pa_{X_i} .

$$p(\mathbf{X}|\mathcal{G}, \Theta) = \prod_{i=1}^K p(X_i|\text{pa}_{X_i}, \Theta_{X_i}) \quad (1)$$

In other words, \mathcal{G} induces the decomposition of the joint distribution of \mathbf{X} (with parameters Θ) in one local distribution for each X_i (with parameters Θ_{X_i}) [11]. So, given the structure in Fig. 2 and following the chain rule (Eq. (1)), the joint probability distribution of our example can be computed as:

$$\begin{aligned} P(X_1, X_2, X_3|\mathcal{G}_1) &= P(X_1|X_2, X_3)P(X_2|X_3)P(X_3) \\ &= P(X_1|X_2)P(X_2|X_3)P(X_3) \end{aligned} \quad (2)$$

It is worth noting that two different DAGs can represent the same set of conditional independencies. In such a case, it is said that both graphs fall into the same *equivalence class*, or that they are *I-equivalent*. As an example, we can develop the following equivalence from Eq. (2), using the product rule of probability.

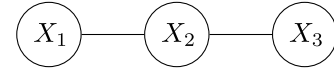
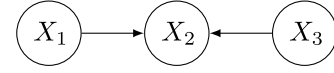
$$\begin{aligned} P(X_1, X_2, X_3|\mathcal{G}_1) &= P(X_1|X_2)P(X_2, X_3) \\ &= P(X_1|X_2)P(X_3|X_2)P(X_2) = P(X_1, X_2, X_3|\mathcal{G}_2) \end{aligned} \quad (3)$$

$$\begin{aligned} P(X_1, X_2, X_3|\mathcal{G}_1) &= P(X_1, X_2)P(X_3|X_2) \\ &= P(X_2|X_1)P(X_1)P(X_3|X_2) = P(X_1, X_2, X_3|\mathcal{G}_3) \end{aligned} \quad (4)$$

Resulting in the structures \mathcal{G}_2 and \mathcal{G}_3 , shown in Fig. 3. All these structures imply $X_1 \perp\!\!\!\perp X_3|X_2$, hence they are I-equivalent.

Each equivalence class is represented by a *completed partially-directed acyclic graph* or CPDAG. In this representation, those edges which define an equivalence class independently of their direction are represented as undirected. In the opposite case, those edges whose orientation flip results in different equivalence classes remain directed. As an example, Fig. 4 shows the CPDAG which schematized the equivalence class of \mathcal{G}_1 , \mathcal{G}_2 and \mathcal{G}_3 .

It is important to remark that, DAGs are I-equivalent if, and only if, they have the same skeleton (i.e. the same edge support) and the same set of V-structures (families of the form $\bigcirc \rightarrow \bigcirc \leftarrow \bigcirc$) [19]. Therefore, the CPDAG above does not include the DAG \mathcal{G}_4 , represented in Fig. 5.

Fig. 4. CPDAG representing the equivalence class of G_1 , G_2 and G_3 .Fig. 5. DAG and CPDAG representing G_4 .

Bayesian networks framework presents many advantages that motivates its use, specifically in high-risk applications where managing uncertainty is needed:

1. They are highly-interpretable models, whose independence properties can be easily obtained by inspection of the graph.
2. They are flexible models, as any set of variables can be inferred by introducing evidence of any set of observed variables.
3. They are probabilistic models, so they quantify the uncertainty of the problem using the probability theory.

Additionally, it is important to keep in mind that the Bayesian network structure represents the independence assumptions of the model regardless of the form of the probability distributions of the variables themselves. In this work, all the models described are going to be considered Gaussian Bayesian networks, whose variables present a continuous univariate Gaussian distribution.

Although the next sections address the concepts needed for the correct understanding of this work, they are not intended to be an in-depth tutorial on Bayesian networks. Therefore, we encourage the reader to explore other references in order to better understand the ins and outs of these models [10,20–22].

2.1. Introduction to Gaussian Bayesian networks

Gaussian Bayesian networks GBNs [23] assume that each variable X_i has a univariate Gaussian distribution and that all the conditional probability distributions are linear Gaussians. Let the variable X_i have a total of K parent variables in \mathcal{G} , $\mathbf{U} = \{U_1, \dots, U_K\}$, the linear Gaussian CPD of X_i takes the form below:

$$P(X_i|\mathbf{U}) = \mathcal{N}\left(X_i \left| \beta_{0i} + \sum_{k \in K} \beta_{ki} u_k; \sigma_i^2 \right.\right) \quad (5)$$

where $\beta_i = \{\beta_{0i}, \dots, \beta_{Ki}\}$ are the linear parameters governing the mean, u_k are the values of the parent variables of X_i , and σ_i^2 is the variance of the conditional distribution of X_i . Thus, the mean of the CPD of X_i is taken to be a linear combination of the values of its parent nodes. Then, a GBN of N variables is described by parameters $\Theta = \{\beta, \sigma^2\}$, where $\beta = \{\beta_0, \dots, \beta_N\}$ and $\sigma^2 = \{\sigma_0^2, \dots, \sigma_N^2\}$. An important result is that a GBN is an alternative representation of a multivariate Gaussian distribution of the variables [22]. Thus, assuming that the input variables follow a Multivariate Gaussian distribution implies that the linearity assumption in Eq. (5) is fulfilled [10]. So, given the parameters Θ of the GBN, the parameters of the corresponding multivariate Gaussian joint distribution can be computed as described in Eqs. (6) and (7).

The mean $\mathbb{E}[\mathbf{X}]$ can be computed recursively as follows:

$$\mathbb{E}[X_i] = \beta_{0i} + \sum_{k \in \text{pa}_{X_i}} \beta_{ki} \mathbb{E}[U_k] \quad (6)$$

The covariance matrix can also be computed recursively, getting each component j, i by the next expression.

$$\text{Cov}[X_i, X_j] = I_{ij} \sigma_i^2 + \sum_{k \in \text{pa}_{X_j}} \beta_{ik} \text{cov}[X_i, U_k] \quad (7)$$

In those scenarios where managing continuous variables is needed, the Gaussian Bayesian networks represent a very interesting approach due to the statistical properties of the multivariate Gaussian distribution.

2.2. Parameter learning in Gaussian Bayesian networks

A Bayesian network is characterized by a dependence DAG \mathcal{G} and a set of conditional probability distributions given this DAG. Thus, the learning process of a Bayesian network can be divided in parameter learning or *model fitting*, and structure learning or *model selection* (see Section 3).

In the case of having a complete database \mathcal{D} (i.e. there are no missing variables), the learning problem of a Bayesian network by maximum likelihood estimation reduces to a set of local learning problems, one for each variable. In other words, we can maximize each local likelihood function independently of the rest of the network. As it can be shown in Eq. (5), the maximum likelihood estimation of the Gaussian Bayesian network parameters reduces to the estimation of the linear coefficients $\beta_i = \{\beta_{0i}, \dots, \beta_{ji}, \dots, \beta_{Ki}\}$, and the variance of each linear Gaussian CPD σ_i^2 , for each variable X_i of the model.

As a result of the maximum likelihood estimation, the parameters governing the mean of each variable can be obtained algebraically by solving the system of $K + 1$ linear equations which are obtained particularizing the expression (8) for each parent of X_i .

$$\begin{aligned} \mathbb{E}_{\mathcal{D}}[X_i] &= \beta_{0i} + \sum_{k \leq K} \beta_{ki} \mathbb{E}_{\mathcal{D}}[U_k] \\ \mathbb{E}_{\mathcal{D}}[X_i \cdot U_k] &= \beta_{0i} \mathbb{E}_{\mathcal{D}}[U_k] + \sum_{j \leq K} \beta_{ji} \mathbb{E}_{\mathcal{D}}[U_j \cdot U_k] \end{aligned} \quad (8)$$

Additionally, the covariance parameter of each variable can be computed following Eq. (9).

$$\sigma_i^2 = \text{Cov}_{\mathcal{D}}[X_i; X_i] - \sum_{j \leq K} \sum_{k \leq K} \beta_{ji} \beta_{ki} \text{Cov}_{\mathcal{D}}[U_j; U_k] \quad (9)$$

The mathematical process resulting in both expressions is more detailed in [10,22].

2.3. Inference in Gaussian Bayesian networks

The introduction of an evidence in some variables of a Bayesian network updates the knowledge of the rest of the unobserved variables of the model. Let \mathbf{X} be the total set of N variables of the model, \mathcal{O} the subset of variables observed with observations $\mathcal{O} \sim o$, and \mathcal{L} the subset of unobserved or latent variables, where $\mathbf{X} = \mathcal{O} \cup \mathcal{L}$. Additionally, let \mathcal{S} be the subset of variables to infer and \mathcal{T} the unobserved variables out of interest, where $\mathcal{L} = \mathcal{S} \cup \mathcal{T}$. The inference process, with continuous probability densities (replace the integrals by sums in the discrete case), is defined by Eq. (10).

$$\begin{aligned} P(\mathcal{S} | \mathcal{O} = o) &= \frac{P(\mathcal{S}, \mathcal{O} = o)}{P(\mathcal{O})} \\ &= \frac{\int_{\mathcal{T}} P(\mathcal{S}, \mathcal{O} = o, t) dt}{P(\mathcal{O} = o)} \\ &= \frac{\int_{\mathcal{T}} P(\mathcal{S}, \mathcal{O} = o, t) dt}{\int_{\mathcal{L}} P(\mathcal{O} = o, l) dl} \end{aligned} \quad (10)$$

Due to the need of the computation of several marginalizations, the inference process with continuous variables can be computationally intensive and generally intractable, involving the use of approximate inference techniques based on Monte-Carlo methods, Loopy Belief propagation and variational inference [24,25].

However, in the case of a GBN, hence a multivariate Gaussian distribution, the exact inference has close-form equations that are tractable in problems where data requirements are not intensive. Thus, using the

previous notation, the vector mean μ and covariance matrix Σ of the multivariate Gaussian distribution can be partitioned as follows.

$$\begin{aligned} \mathbf{X} &= \begin{bmatrix} \mathcal{L} \\ \mathcal{O} \end{bmatrix}; \text{size} \begin{bmatrix} l \times 1 \\ (N-l) \times 1 \end{bmatrix} \\ \mu &= \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}; \text{size} \begin{bmatrix} l \times 1 \\ (N-l) \times 1 \end{bmatrix} \\ \Sigma &= \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}; \text{size} \begin{bmatrix} l \times l & l \times (N-l) \\ (N-l) \times l & (N-l) \times (N-l) \end{bmatrix} \end{aligned} \quad (11)$$

Then, given a vector of observations $\mathcal{O} = o$, the posterior distribution $p(\mathcal{L} | \mathcal{O} = o)$ is also a Gaussian probability density function, defined by the following parameters:

$$\begin{aligned} \mu_{1|2} &= \mu_1 + \Sigma_{12} \Sigma_{22}^{-1} (o - \mu_2) \\ \Sigma_{1|2} &= \Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21} \end{aligned} \quad (12)$$

In this case, the computation of these matrices is not intensive and remains tractable. In more complex problems, the computational complexity of the inference process can be reduced using exact inference algorithms, such as *Variable elimination* or the *Junction tree algorithm* [10,20,21]. Nevertheless, given the size of our dataset, these algorithms are out of the scope of this work.

3. Score-based Bayesian network structure learning

In the problem at hand, the process of learning the dependence DAG \mathcal{G} of a Bayesian network has two main goals. First, it aims to discover the set of conditional independencies present in the data, in order to increase the knowledge of the primary loop chemistry problem. Second, it also aims to train a structure capable of generalizing. In other words, the learned model must be capable of predicting accurately radioisotope values from new observations of the control variables. Structure learning score-based algorithms face the structure learning process as a score-optimization problem. Accordingly, the operation of these algorithms is based on the definition of a *scoring function*, a *search space*, and a *search algorithm*.

The scoring function is the optimization metric, in such a way that the structure that maximizes this score is the selected one. The search space defines the set of structure candidates which are going to be evaluated and the different operators that connect each candidate. Finally, the search algorithm performs these operators to scroll through the search space.

The most intuitive approach is a simple greedy algorithm based on the evaluation of the scoring function on the total set of possible structures with N variables. Then, the final structure is the one that maximizes the scoring function. This procedure guarantees the selection of the global maximum of the scoring function, hence the selection of the structure that better fits to train data. However, the number of possible structures varies super-exponentially with the number of variables N . This makes the greedy search a computationally inefficient approach, resulting intractable for $N \geq 7$. For this reason, most score-based algorithms are based on the reduction of the search space and the definition of an efficient search algorithm. Before describing the different algorithms proposed for the problem at hand, the definition of the selected scoring function is needed.

3.1. Scoring function

Different scoring functions can be found in literature [10,13]. A natural choice is the likelihood score, which measures the probability of the data given the Bayesian network model. It is defined as:

$$\text{score}_{\mathcal{L}}(\mathcal{G} : \mathcal{D}) = \ell(\hat{\Theta}_{\mathcal{G}} : \mathcal{D}) \quad (13)$$

where $\ell(\hat{\Theta}_{\mathcal{G}} : \mathcal{D})$ is the logarithm of the likelihood function, and $\hat{\Theta}_{\mathcal{G}}$ is the maximum likelihood parameters given \mathcal{G} . As a reminder, in Gaussian Bayesian networks these parameters are defined in Section 2.2

as $\hat{\Theta}_G = \{\beta, \sigma^2\}$. Nevertheless, this score tends to overfit to the training database, generating models that fail to generalize well for new data.

In order to avoid this limitation, Bayesian scores [26] can be computed by adopting a Bayesian perspective of the structure learning process. However, the complexity of the Bayesian score usually induces the use of approximations as the *Bayesian information criterion* (BIC). The BIC score can be computed as follows:

$$\text{score}_{\text{BIC}}(G : D) = \ell(\hat{\Theta}_G : D) - \frac{\log N}{2} |G| \quad (14)$$

where $|G|$ is the graph complexity, defined as the number of parameters of a certain structure candidate G .

As it can be shown in Eq. (14), BIC penalizes more complex structures with the introduction of a penalty term to the likelihood score. In this way, the BIC score avoids the overfitting problem of the likelihood score and tends to learn structures with higher generalization capabilities. Additionally, the BIC score satisfies three essential characteristics for the optimization of the structure learning procedure (see [10] for proofs and demonstrations):

- Equivalence: if G_1 and G_2 are I-equivalent, then $\text{score}_{\text{BIC}}(G_1 : D) = \text{score}_{\text{BIC}}(G_2 : D)$
- Consistency: if the training data D tends to infinite, the global maximum of the BIC score corresponds to the structure that induces the underlying distribution.
- Decomposition: the BIC score of a graph G can be computed as the sum of local scores calculated for each variable.

$$\text{score}_{\text{BIC}}(G : D) = \sum_k \text{score}_{\text{BIC}}(X_k | \text{pa}_{X_k}^G : D) \quad (15)$$

These characteristics are the reason why the BIC score has been selected for all the structure learning experiments of this work. Nevertheless, a wide description of other alternative scoring functions can be found in [27].

3.2. Structure learning algorithms

After defining the scoring function, in this section, we briefly describe the different structure learning algorithms proposed in this work.

3.2.1. Greedy Hill-Climbing (GHC)

Unlike the most intuitive and naive greedy search approach, the Greedy Hill-Climbing algorithm *GHC* uses the local search algorithm Hill-climbing for scrolling the space of DAG candidates for a set of variables.

The Hill-climbing algorithm starts from either an empty, a random, or a previously defined DAG candidate. This first candidate is also called *seed*. After computing the score of the initial candidate, the algorithm's main loop consists in attempting every possible single-edge addition, removal, or reversal. As a result of the application of these operators, a set of structure neighbors of the actual candidate is generated. Then, the scoring function is evaluated over the neighbors. The structure with a higher BIC score is then selected as the actual candidate and the process is repeated until convergence.

Apart from its high computational cost, the main limitation of this algorithm is its tendency to converge in two undesired situations: local maxima, when there are no neighbors with a higher score than the actual candidate; and *plateaus*, when all the neighbors are I-equivalent to the actual candidate, hence they all have the same BIC value. Several alternatives are found to improve the performance of this local search algorithm. Tabu search [14] accepts worsening moves if no neighbor improves the score, and it also uses memory structures to avoid coming back to previously-visited candidates. Greedy equivalence search defines a search space of I-equivalence to avoid getting stuck in Plateaus. Finally, some algorithms as K2, limit the search space to the set of DAGs which satisfy a defined node ordering.

3.2.2. Greedy Equivalence Search (GES)

As it was mentioned before, *Greedy Equivalence Search (GES)* [16] defines an I-equivalence search space, where each candidate is a set of structures belonging to the same equivalence class. In this way, GES indeed proposes a CPDAG search space, in which neighbors of a certain CPDAG differ only by a single operation of edge-addition or edge-deletion.

The GES algorithm consists of a two-phase greedy search through the space of DAG equivalence classes. The first phase is called *forward equivalence search (FES)*. It starts with an empty CPDAG and it greedily applies edge-addition operators until it reaches a local maximum (i.e. no more edge-additions increments the score). Then, the second phase, called *backward equivalence search (BES)*, is performed. This phase starts with the resulting CPDAG of FES and greedily applies edge-deletion operations. The algorithm converges when no more edge-deletions increments the score. Finally, a DAG is extracted from the CPDAG obtained.

3.2.3. K2

K2 algorithm [15] is the basic structure learning method with a prior node ordering. This topological ordering defines the parent candidates of each variable. So, the potential parents set of a certain node can only include those nodes that precede it in the input order. Thus, if node X_i comes before node X_j in the given order, X_j cannot be a parent of node X_i . In this way, the K2 algorithm reduces the computational complexity obtained in greedy approaches by reducing the candidate structure space to the set of graphs that satisfy the established order.

This method starts by assuming that the current node lacks ancestors. Then, it proposes a greedy algorithm to determine the parent nodes of the rest of the variables in the problem. In every iteration of this algorithm, those parents whose introduction increases the scoring function are added incrementally. Finally, K2 stops adding parents to the current node when the addition of any candidate parent cannot increase the scoring function. This procedure is repeated for each node in the input order.

The main limitation of this algorithm is its sensitivity to the input order. Thus, finding a good topological sort is crucial for the algorithm success. However, the definition of an optimal ordering not only results in a faster algorithm than the basic greedy search but also allows the introduction of some expert knowledge to the structure learning process, by the establishment of the order following a given criteria (causality assumptions, and so on).

4. Data treatment

In this section, we summarize the most relevant aspects concerning the database and the data pre-processing procedure. However, we encourage the reader to look through [3] for a more detailed description.

4.1. Database

The database used for training and evaluating the Bayesian network models is provided by ENUSA Industrias Avanzadas S. A. It considers measurements of 8 chemical and physical variables of 5 PWR Spanish power plants (named throughout this work as *Plant1*, *Plant2*, and so on). These variables are divided into *control* and *radioisotope variables*. Control variables are *pH* (the pH in the primary loop, dimensionless), *L* (letdown flow measured in m³/hour), *P* (thermal power in percentage), *H2* (hydrogen measured in cc/kg), and *Zn* (zinc in ppm, only present in *Plant3*, *Plant4*, and *Plant5*). Radioisotope variables are ⁵⁸Co, ⁶⁰Co, and ⁵¹Cr, all measured in Beckerel/ml (Bq/ml). It is important to note that, due to the existing differences in the power plants behavior and the measurement acquisition process, certain variables may have different ranges depending on the nuclear power plant. This forces us to train and evaluate a Gaussian Bayesian network for each plant separately.

Table 1

Distribution of the complete vectors over the operation cycles of each power plant.

Plant name	Operation cycles				
	1	2	3	4	5
Plant 1	57	62	62	57	46
Plant 2	58	61	64	60	53
Plant 3	61	65	64		
Plant 4	101	64	66	54	
Plant 5	68	69	64	61	

4.2. Data pre-processing

Data pre-processing step performed in this work can be divided in the following steps: *data cleanup*, *control variables interpolation*, *data transformation*, and *data Gaussianization*.

1. Data cleanup: For every day of each operation cycle, measurements were arranged in 7 or 8-dimensional vectors, depending on the presence of *Zn* injection in the power plant. Then, to avoid high data imputation errors, all the vectors with any missing radioisotope variable were discarded.
2. Control variables interpolation: A linear interpolation of the missing control variables was applied to get a higher number of complete vectors.
3. Data transformation: That includes the summation of the soluble and insoluble form for each radioisotope variables and the differentiation of the *Letdown*.
4. Data Gaussianization: As we have seen in Section 2.1, Gaussian Bayesian Networks assume that the input variables are jointly Gaussian. To better fit data to this assumption, an histogram-based Gaussian transformation of each variable is applied in an independent way. This transformation is invertible, in such a way that transformation g transforms the *input space* into a *Gaussian* or *model space*, and function f transform the *Gaussian space* into the interpretable *input space*. Both functions are defined in Eq. (16).

$$\begin{aligned} g(X_i) &= \phi^{-1}(F(X_i)) \\ f(X_i) &= F^{-1}(\phi(X_i)) \end{aligned} \quad (16)$$

where X_i is a univariate variable, $\hat{F}(X_i)$ is its empirical Cumulative Density Function (CDF), and Σ is the CDF of a gaussian variable. It is worth noting that, even if a marginal Gaussianization of the variables do not ensure the input space to be jointly Gaussian, this transformation has proven [3] to enhance considerably the performance of the Gaussian Bayesian networks. Moreover, a joint Gaussianization of the input space with Real-NVP normalization flow [28] has been tested without obtaining any improvement, and therefore, following Occam's principle, we have selected the simplest approach to Gaussianize the data.

Table 1 shows the number of available vectors that make up the final database after this data pre-processing step. As it can be seen, a different number of operation cycles are available for each nuclear power plant. Furthermore, each operation cycle presents only a set of 61 complete vectors on average. Thus, the final database for every nuclear power plant is characterized by having a scarce number of complete vectors. In fact, in every plant we dispose of less than 300 complete vectors for both, training and evaluating the probabilistic models.

Under these conditions, it is necessary to define a reliable evaluation protocol, which allows an honest comparison of the Bayesian networks, trained by the different structure learning algorithms.

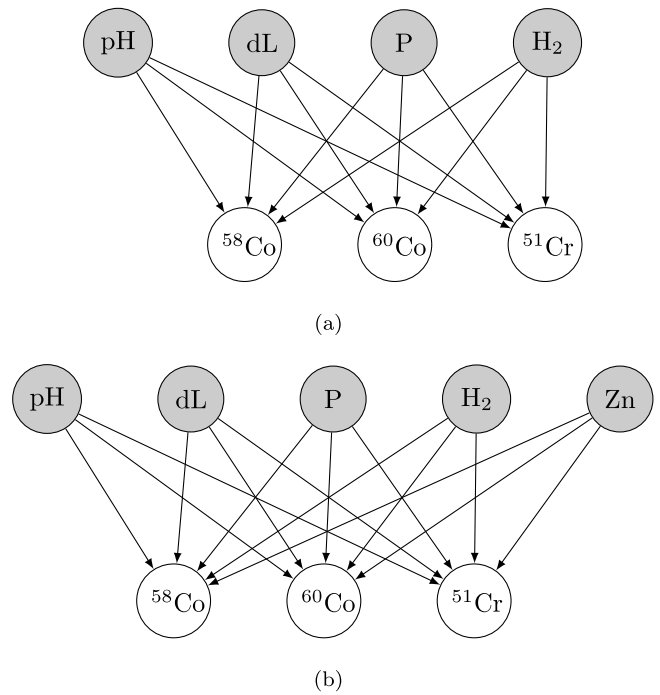


Fig. 6. (a) Baseline structure for *Plant1* and *Plant2*; (b) Baseline structure for *Plant3*, *Plant4* and *Plant5*.

5. Baseline

The baseline model [3] for each nuclear power plant consists on a Gaussian Bayesian network whose parameters are trained by maximum likelihood estimation (see Section 2.2), and whose structure is directly defined by the expert knowledge of ENUSA engineers. Thus, the set of conditional independencies defined by the baseline model structure may not correspond to the set of empirical independencies present in the data. They instead correspond to some theoretical assumptions based on the prior knowledge we have of the problem. Figs. 6(a) and 6(b) represent the predefined structures of the GBNs depending on the presence of *Zn* in the nuclear power plant. Despite the flexibility of the BNs, these models will be evaluated in the specific task of radioisotope prediction from observations of control variables. Thus, in both representations, gray nodes correspond to the observed variables, while white nodes correspond to the unknown variables which value we want to infer in the problem at hand.

The conditional independence assumptions of the models can be extracted directly by the inspection of both structures. First, each of the radioisotope variables is assumed to be directly related with all the control variables in the model. Second, the control variables are to be assumed independent if radioisotope variables are unknown. Finally, radioisotope variables are assumed to be independent if all the control variables are observed.

6. Experiments

6.1. Protocol

The primary loop chemistry problem is a challenging task due to the limited database available for both, training and evaluating the Bayesian network models. For this reason, the evaluation of the different structure learning algorithms performance under data sparsity conditions is considered critical. Thus, a preliminary experiment is performed to evaluate the capability of the algorithms to capture the real set of conditional independencies.

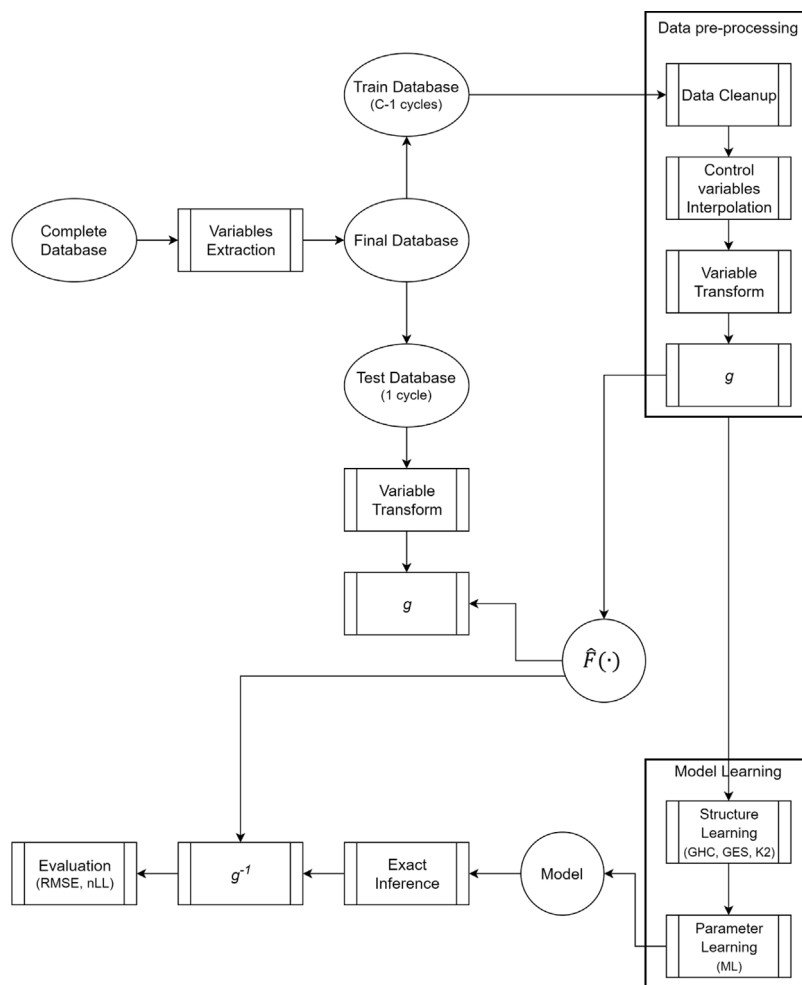


Fig. 7. Block diagram explaining the training and prediction process performed.

This experiment is based on the generation of synthetic training databases with variable sizes (N), and a testing dataset of 1000 samples. These databases are obtained by sampling the baseline model described in Section 5, whose structure is known. Then, a comparison of the structure learning algorithms is performed for each size of the training data. Results of this experiment are presented in Section 6.3.1.

After comparing algorithms reliability with scarce data, the generalization capabilities of the models are evaluated. For this, a leave-one-cycle-out cross-validation procedure is proposed to make this evaluation honestly. Following this process, for each nuclear power plant, the Gaussianization transformation and the different models are trained using all the operation cycles available except one, which is used for testing. This process is repeated until every cycle has been predicted.

Fig. 7 shows, in a schematic form, the different steps performed for each iteration of the leave-one-cycle-out cross-validation. Each of these iterations simulates a realistic scenario, in which all the cycles available in the past are used to predict radioisotopes in an unobserved new cycle. This procedure can be divided into the following steps:

1. First, from the complete set of measured variables, those which make up the proposed models are extracted.
2. The database is divided into a training set and a testing set. The training set is formed by the measurements of all the operation cycles available except one, which is going to be used for evaluating the predictions performed by the different models.
3. The training data is pre-processed as described in Section 4.2. In this pre-processing step, the empirical CDFs $\hat{F}(\cdot)$ are trained in order to apply the Gaussianization transformation. On the other

hand, the testing data is Gaussianized using the CDFs previously computed with the training set.

4. Training data in the model space is used to train the GBN models, characterized by $\lambda = (\mathcal{G}, \Theta)$. Note that the structure of the Bayesian networks is trained using the different algorithms explained in Section 3, while the parameter learning step is performed by Maximum Likelihood.
5. Then, the unobserved radioisotope variables are inferred from the Gaussianized control variables of the test cycle. Note that the values inferred by the GBNs remain in the model space.
6. Finally, in order to evaluate and interpret the predictions, the inferred values are transformed back to the chemical space.

The previous process is repeated for each nuclear plant until all the cycles are predicted and evaluated. Then, the predictions of all the cycles are considered for measuring the performance of the models trained by the different structure learning algorithms.

6.2. Performance measures

In this section, we describe the metrics used to evaluate the models trained by the different structure learning algorithms proposed. To evaluate the performance of the different structure learning algorithms under data scarcity conditions, two metrics are proposed:

- *Editing measure* or *em* [29]: Defines the minimum number of operations (addition, subtractions, and orientation changes of edges) which must be applied on a graph in order to obtain the structure of the generative model of data.

- *Bayesian Information Criterion* or *BIC* evaluated over the testing dataset of fixed-length.

Additionally, to evaluate the generalizing power of the structures learned, we are going to consider two different metrics in the leave-one-cycle-out process described before (Section 6.1).

- *Root mean square error* or *RMSE* of the radioisotope predictions with respect to its real value. Let p be a power plant with a total of C operation cycles available in the database. The RMSE for the prediction of the operation cycle $c \in C$ is computed as:

$$r^{p,c}(\hat{\mu}_t^{p,c}) = \sqrt{\frac{\sum_{t=1}^T (\hat{\mu}_t^{p,c} - \mu_t^{p,c})^2}{T}} \quad (17)$$

where $\mu_t^{p,c}$ is the real value of the variable we want to infer, and $\hat{\mu}_t^{p,c}$ is the prediction performed by the Gaussian Bayesian network. Then, in order to obtain a unique metric for each power plant, we average the resulting RMSE for the prediction of each cycle.

$$R(p) = \frac{1}{C} \sum_c r^{p,c} \quad (18)$$

This metric is called in [3] *RMSE for all days*.

- Finally, to measure the consistency of the different Gaussian Bayesian networks to the test data, the negative log-likelihood (nLL) is evaluated over the test set. The nLL can be computed as follows:

$$nLL^{p,c} = -\frac{1}{T} \sum_{t=1}^T \log p(\mu_t^{p,c} | \mathcal{G}, \hat{\theta}_{\mathcal{G}}, \mathbf{o}_t) \quad (19)$$

where \mathcal{G} is the structure learned from data, $\hat{\theta}_{\mathcal{G}}$ are the maximum likelihood parameters of the GBN model, and \mathbf{o}_t is the vector of the observed variables in time t . It is worth noting that, in the problem at hand, the observed variables are the previously defined control variables, while the unknown variables are the radioisotopes.

It is important to remark that the negative sign of the metric makes the nLL to be interpreted as an error in the consistency of the model with the data. Equivalently to the RMSE, this error metric is evaluated for the prediction of each operation cycle. Then, in order to have a unique metric for each power plant, we average it to obtain the so-called *negative log-likelihood for all days*.

$$L(p) = \frac{1}{C} \sum_{c=1}^C nLL^{p,c} \quad (20)$$

6.3. Results

In this section, we present the results of the experiments previously described in Section 6.1. Additionally, we represent and analyze the final models proposed alternatively to the baseline.

6.3.1. Data needs comparison

The first experiment aims to compare the structure learning algorithms performance with a train dataset of variable size. As it was mentioned in 6.1, databases of different sizes are generated by sampling the baseline model. Then, for each training database, GBNs are trained making use of different structure learning algorithms. Finally, the performance of these algorithms over a test dataset is compared in terms of BIC, *em*, and computational cost. This whole procedure has been repeated 20 times to report the uncertainty (or *Monte Carlo error*) associated to the finite nature of the simulations. Table 2 shows, for each size N of the training database, mean and standard deviation values of: the BIC evaluated over the fixed-size test dataset, the *em* between the learned structures and the structure of the generative model of data, and the time (in seconds) consumed by each algorithm.

Additionally, examples of the structures learned throughout the process is included.

Three algorithms are evaluated: the so-called Greedy Hill-Climbing (GHC), K2, and Greedy Equivalence Search (GES). For GHC, results for two different initial structures (or seed structures) are registered. Structures labeled as GHC_1 are learned using an empty seed, while structures labeled as GHC_2 are learned using a random seed. For the K2 algorithm, experiments have been performed using two different topological orderings, both defined by ENUSA experts from their prior knowledge of the problem.

$$pH \rightarrow P \rightarrow H2 \rightarrow dL \rightarrow {}^{51}\text{Cr} \rightarrow {}^{60}\text{Cr} \rightarrow {}^{58}\text{Cr} \quad (21)$$

$$P \rightarrow pH \rightarrow H2 \rightarrow dL \rightarrow {}^{51}\text{Cr} \rightarrow {}^{60}\text{Cr} \rightarrow {}^{58}\text{Cr} \quad (22)$$

The $K2_1$ entry of the table corresponds to the results of K2 using the ordering (21), while the entry $K2_2$ is related to the results using the ordering (22). Note that both orderings only differ in the position of the two first variables (pH and P).

From the analysis presented in Table 2, some conclusions can be extracted. First, it can be seen that all the algorithms reconstruct the structure of the data generating model when a dataset of 5000 samples is available. However, this case does not represent a realistic scenario, where the amount of complete vectors for training the models is scarce. In such cases, the Greedy Hill Climbing algorithm converges on structures less similar (*i.e.* with higher *em*) to the data generator than those obtained by the other proposed algorithms. This behavior remains independently of the seed structure defined. Additionally the GHC training process is the most time-consuming alternative, growing exponentially with the size of the available database. Contrariwise, K2 is the fastest algorithm, being up to 30 times faster than GHC and 25 times faster than GES. Moreover, it turns out to be a fairly insensitive algorithm to the size of the training set, reaching a low value of the *em* metric for a training set of 250 samples. However, the main limitation of this algorithm, as it is mentioned in Section 3, lies in its sensitivity to the topological ordering.

It is worth noting that, in this evaluation, the generative structure of data does not violate the topological ordering defined by ENUSA personnel. Otherwise, the data generating structure could not be learned with this algorithm. Moreover, regarding the Greedy equivalence search algorithm, it gets the best performance in terms of BIC and *em* for scarce databases ($N < 500$). Finally, it can be observed that the *BIC* and *em* measures are consistent for different repetitions of the simulation procedure, *i.e.* the standard deviations of both metrics are null or almost null. This indicates that the same structures, or I-equivalent structures, are obtained for an specific algorithm and database size.

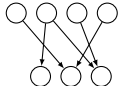
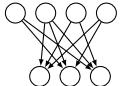
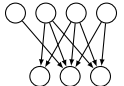
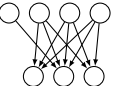

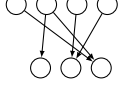
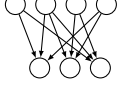

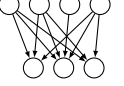
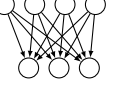










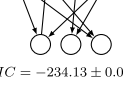
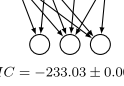
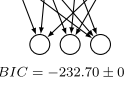
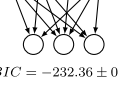
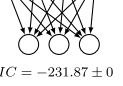
6.3.2. Generalization comparison

After performing the comparison of the different structure learning algorithms in terms of their learning capabilities with a limited training database, in this section, we compare the generalization power of the models trained by these techniques. For this, the leave-one-cycle-out cross-validation procedure, described in Section 6.1, is performed.

The results achieved in terms of *RMSE for all days* (R) are represented in Figs. 8 and 9, for the prediction of ${}^{58}\text{Co}$ and ${}^{60}\text{Co}$ respectively. It can be seen that the baseline model and the models trained by the structure learning algorithms get similar results in terms of RMSE *for all days*. In fact, models trained by K2 and GES generally get a slight improvement in the radioisotope predictions, especially in the prediction of ${}^{58}\text{Co}$. In contrast, in the case of *Plant 5*, structure learning entails a remarkably worse performance than the model whose structure is predefined by expert knowledge. Thus, given the results in terms of RMSE, we cannot conclude that learning the structures in a data-driven way implies the generation of models with higher generalization capabilities.

Table 2

For each structure learning algorithm and train database size, it represents an example of the Learned structures. In addition BIC/100, em and time consumed ($t(s)$), are included using the following format: $\text{mean}(m) \pm \text{std}(m)$, where m is the chosen metric.

Algorithm	$N_{train} = 100$	$N_{train} = 250$	$N_{train} = 500$	$N_{train} = 1000$	$N_{train} = 5000$
$K2_1$	 $BIC = -234.82 \pm 0$ $em = 7 \pm 0$ $t(s) = 0.433 \pm 0.116$	 $BIC = -233.73 \pm 0$ $em = 3 \pm 0$ $t(s) = 1.088 \pm 0.156$	 $BIC = -233.03 \pm 0.01$ $em = 3 \pm 0$ $t(s) = 2.103 \pm 0.179$	 $BIC = -232.36 \pm 0$ $em = 2 \pm 0$ $t(s) = 4.307 \pm 0.261$	 $BIC = -231.87 \pm 0$ $em = 0 \pm 0$ $t(s) = 21.873 \pm 2.54$
$K2_2$	 $BIC = -234.82 \pm 0.001$ $em = 7 \pm 0.23$ $t(s) = 0.383 \pm 0.073$	 $BIC = -233.73 \pm 0$ $em = 3 \pm 0.23$ $t(s) = 1.021 \pm 0.051$	 $BIC = -233.03 \pm 0.001$ $em = 3 \pm 0$ $t(s) = 1.971 \pm 0.055$	 $BIC = -232.36 \pm 0$ $em = 2 \pm 0$ $t(s) = 4.046 \pm 0.177$	 $BIC = -231.87 \pm 0$ $em = 0 \pm 0$ $t(s) = 20.476 \pm 1.365$
GHC_1	 $BIC = -234.55 \pm 0$ $em = 7 \pm 0$ $t(s) = 17.934 \pm 2.825$	 $BIC = -233.75 \pm 0.001$ $em = 5 \pm 0.23$ $t(s) = 54.723 \pm 2.136$	 $BIC = -233.03 \pm 0$ $em = 4 \pm 0$ $t(s) = 108.775 \pm 2.387$	 $BIC = -232.38 \pm 0$ $em = 3 \pm 0$ $t(s) = 223.25 \pm 19.776$	 $BIC = -231.87 \pm 0$ $em = 0 \pm 0$ $t(s) = 1125.527 \pm 21.262$
GHC_2	 $BIC = -234.54 \pm 0.001$ $em = 7 \pm 0.23$ $t(s) = 17.014 \pm 1.107$	 $BIC = -233.75 \pm 0.001$ $em = 5 \pm 0$ $t(s) = 38.414 \pm 0.85$	 $BIC = -233.03 \pm 0$ $em = 4 \pm 0$ $t(s) = 60.239 \pm 1.73$	 $BIC = -232.38 \pm 0$ $em = 3 \pm 0$ $t(s) = 127.371 \pm 7.95$	 $BIC = -231.87 \pm 0$ $em = 0 \pm 0$ $t(s) = 763.568 \pm 18.28$
GES	 $BIC = -234.13 \pm 0.001$ $em = 6 \pm 0$ $t(s) = 10.51 \pm 0.83$	 $BIC = -233.03 \pm 0.001$ $em = 3 \pm 0$ $t(s) = 32.71 \pm 0.76$	 $BIC = -232.70 \pm 0$ $em = 2 \pm 0$ $t(s) = 55.38 \pm 1.58$	 $BIC = -232.36 \pm 0$ $em = 2 \pm 0$ $t(s) = 101.183 \pm 7.64$	 $BIC = -231.87 \pm 0$ $em = 0 \pm 0$ $t(s) = 557.581 \pm 17.22$

However, the RMSE metric only considers the mean value of the Gaussian conditional distribution which results of the exact inference process of the GBN (see Section 2.3). Thus, in order to evaluate the degree in which the model probabilistically represents the test predictions, we compare the algorithms in terms of the negative log-likelihood error metric.

In a similar way to the previous representations, Fig. 10 plots the resulting nLL for all days in the prediction of both radioisotopes, ^{58}Co and ^{60}Co . In view of the results, all the proposed structure learning algorithms generate GBN models that reduce the nLL for all days in the prediction of the radioisotopes of interest. In other words, the models whose structure is learned in a data-driven way probabilistically represent better new (i.e. previously unseen by the model) data than the model with a structure fixed by expert knowledge.

This means that the structures learned from data are closer to encode the real set of conditional independences of the variables of the model. Thus, despite obtaining similar results in terms of RMSE, the resulting models are more calibrated, hence more reliable, than the baseline model. This is an essential advantage of the resulting models, especially in a high-risk scenario in which the quantification of the uncertainty present in the predictions is crucial.

Nevertheless, it is important to remark that these conclusions do not hold in *Plant 5*. In this case, both metrics, RMSE and nLL, are degraded when using any structure learning algorithm. In a fine-grained analysis, we conclude that this degradation may be caused by the high data imputation error committed in the interpolation of the variable $H2$, whose acquisition frequency is much lower in this plant than in the others. As a matter of illustration, Fig. 11 compares the acquisition frequency map of the third cycle available in the database for this nuclear power plant and the fourth cycle for *Plant 2*.

For the rest of the nuclear power plants, models learned by GES and $K2$ are the ones that better generalize to new data. Specifically,

$K2$ minimizes the nLL for *Plant 1* and *Plant 4*, while GES obtains a higher performance in *Plant 3* and *Plant 4*. Finally, it can be seen that GHC generally results in models with a lower fit to the test data than the other structure learning alternatives. For this reason, the GHC algorithm is discarded from posterior analysis in this work.

6.3.3. Final models proposed

The results described in the previous section prove the generalization capabilities of the GBN models learned with structure learning algorithms. Additionally, it shows the clear advantages related with calibration and reliability of the resulting models, in comparison with a GBN with a predefined structure.

However, for a certain power plant, the structures learned by these algorithms for the prediction of each cycle may be different. In fact, in scarce cases of the leave-one-cycle-out procedure, the structures learned for the prediction of two different cycles belong to different equivalence classes. As an illustration of this, Fig. 12(a) represents the structure learned for the prediction of the first cycle available for *Plant 1*. Furthermore, Fig. 12(b) represents the structure learned for the prediction of the second cycle available for the same power plant. Both structures have been learned by $K2$.

Following these results, it could be concluded that the conditional independences between the variables change depending on the operation cycle in which they are captured. However, ENUSA personnel discarded this hypothesis because it does not fit with the theoretical knowledge they have about the primary loop chemistry. Then, we conclude that the possible differences between the learned structures may be due to:

- The scarce data available to train the models. As it was seen in 6.3.1, structure learning algorithms needed a higher number of samples to learn the real conditional independence set of the variables.

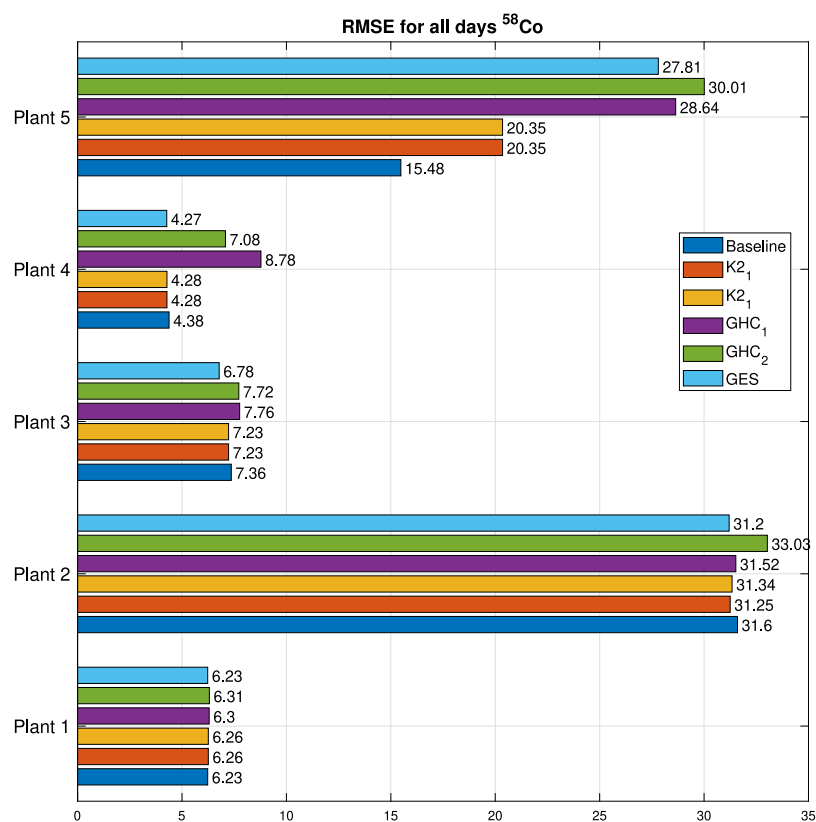


Fig. 8. RMSE for all days for the prediction of the radioisotope ^{58}Co .

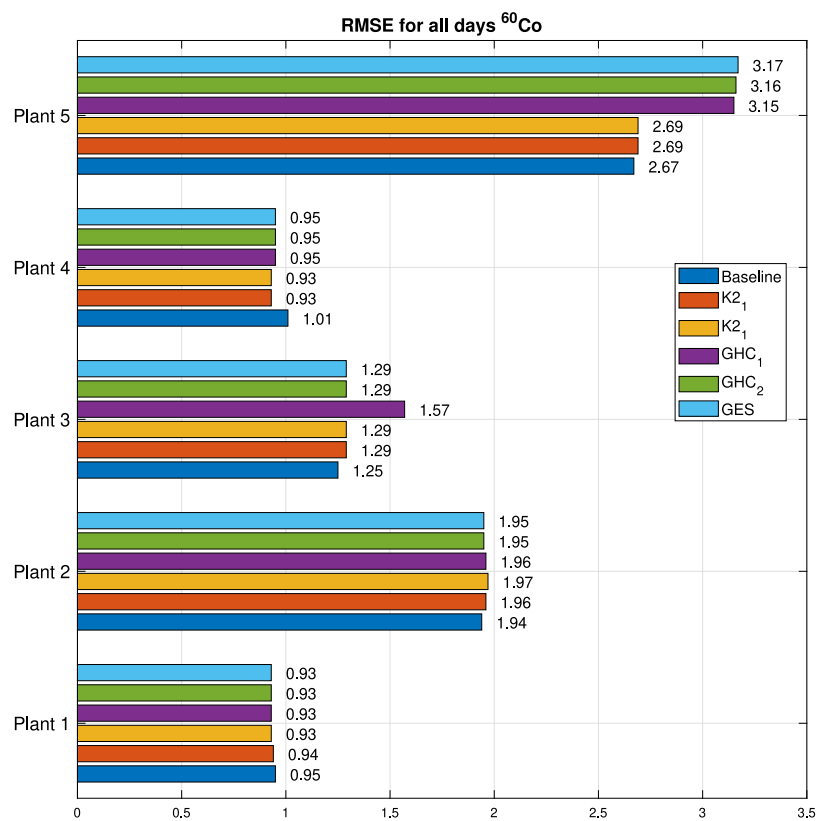


Fig. 9. RMSE for all days for the simultaneous prediction of the radioisotope ^{60}Co .

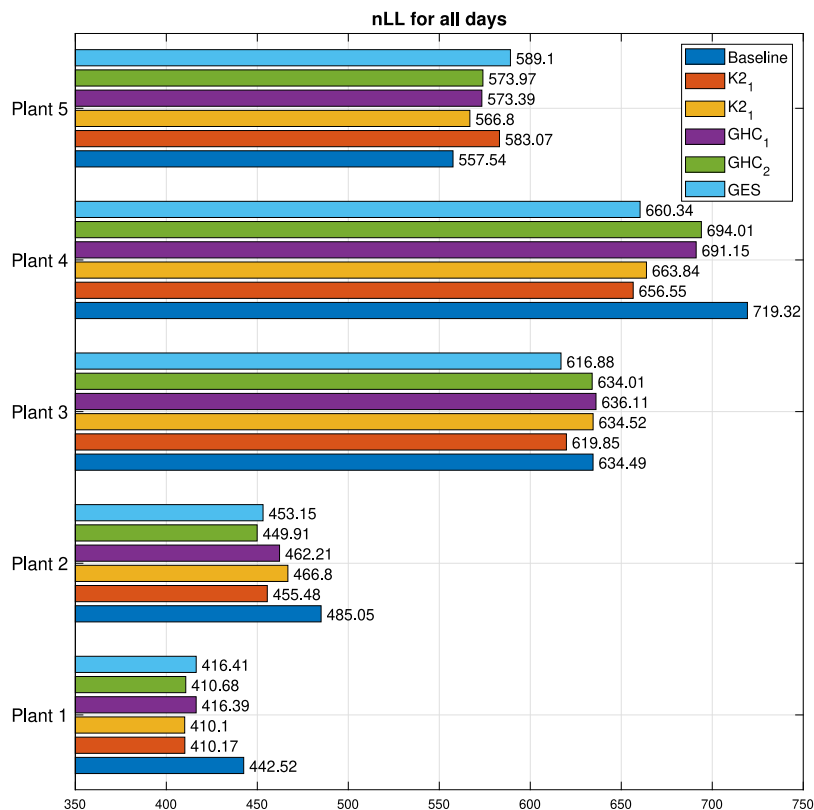


Fig. 10. nLL for all days for the simultaneous prediction of the radioisotopes ^{58}Co and ^{60}Co .

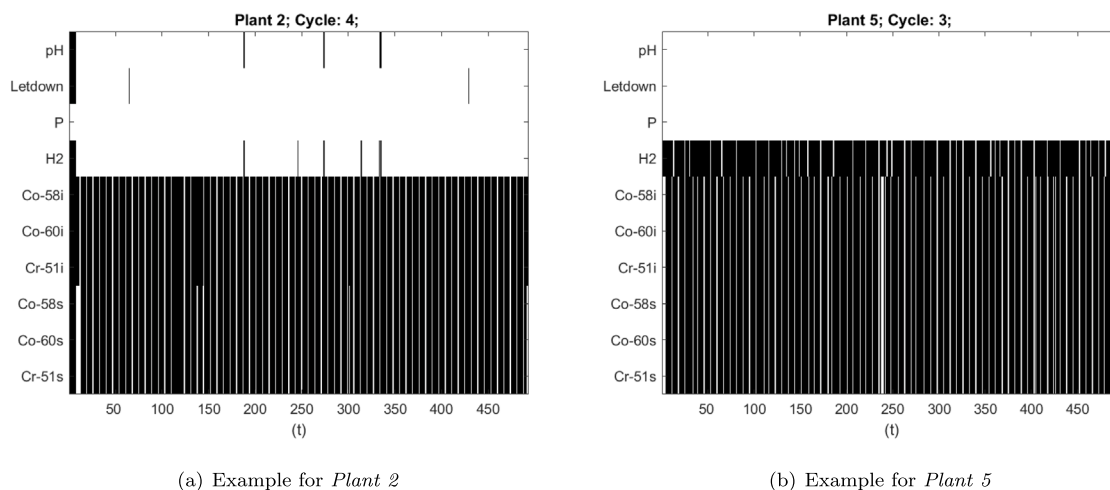


Fig. 11. Acquisition maps comparison. It represent: in black, those days in which no measurements have been recorded; in white, those days where at least one measure has been included in the database.

- The presence of statistic noise in the database.
- The existence of *outliers* in the database, probably caused by an error in the acquisition process of the measurement.

Thus, two structures per power plant are finally proposed as alternatives to the ones pre-defined (represented in Figs. 6(a) and 6(b)). Both structures are trained with all the operation cycles available for each nuclear power plant, and are represented in Figs. 13 to 17. Left-side structure are trained with K2 with the ordering labeled as K2₁ (note that the results obtained with both orderings pre-defined by ENUSA experts are similar). Right-side structure are trained with GES, which has proven to be the best alternative in the prediction of radioisotopes in certain power plants.

Some conclusions can be extracted directly from the analysis of these structures. First, it can be seen that the structures learned for each nuclear power plant are different. In fact, different power plants generally present non-I-equivalent structures. This means that the set of conditional independences of the variables is different depending on the nuclear power plant, its measurement and acquisition protocols, and their procedures regarding the primary loop chemistry problem. As an exception, the algorithm K2 generates exactly the same structures for power plants without zinc injection: *Plant 1* and *Plant 2*.

Moreover, it is important to remark that the variable dL remains unlinked in any case. Thus, the differentiated letdown flow does not introduce any information about the rest of the variables of the model. For this reason, following these results, this variable could perfectly

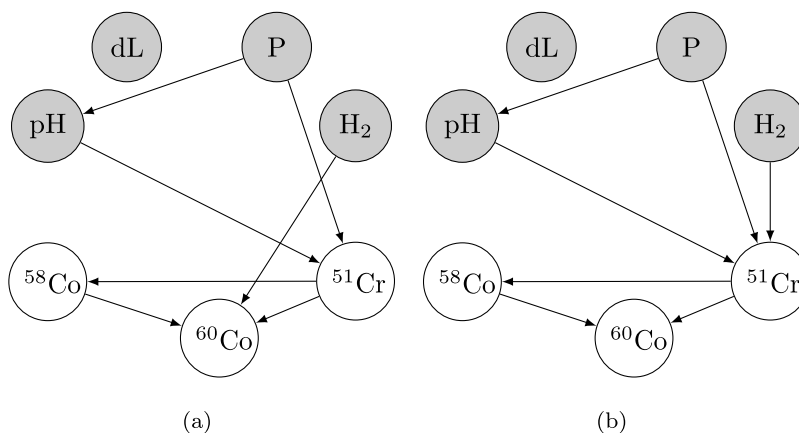


Fig. 12. (a) Structure learned for the prediction of the first available cycle of *Plant1*; (b) Structure learned for the prediction of the second available cycle of *Plant1*.

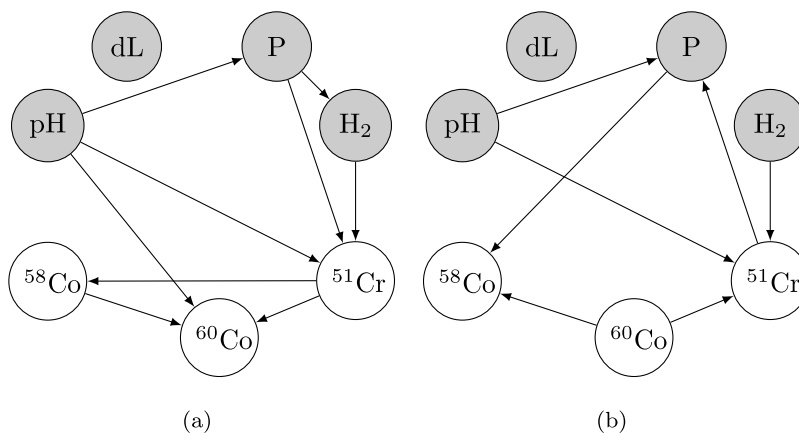


Fig. 13. Final alternative models for *Plant 1* (a) Generated by K2 (b) Generated by GES.

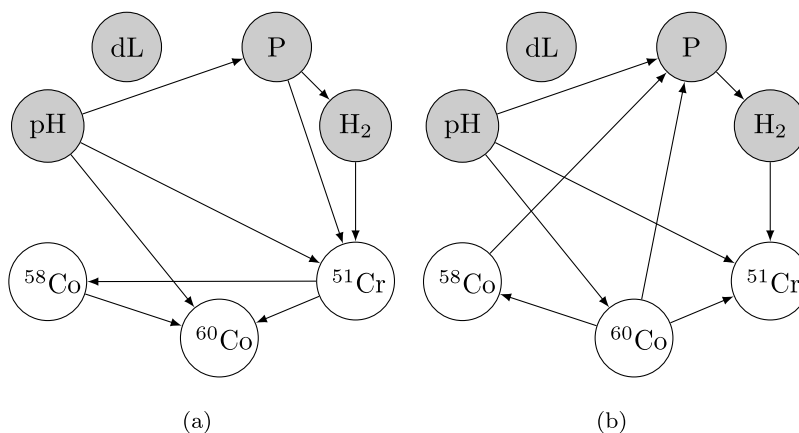


Fig. 14. Final alternative models for *Plant 2* (a) Generated by K2 (b) Generated by GES.

be removed from the model. This does not occur with the rest of the control variables, which are all generally linked by an active trail with the radioisotopes. Within the control variables, the importance of the pH, zinc, and H2 stands out, since both have a direct dependence on radioisotopes in every nuclear power plant. To see the effectiveness of both algorithms from an optimization perspective, Table 3 represents the BIC values obtained by each algorithm. Both algorithms result in structures with a BIC much higher than the one obtained with the baseline models. Moreover, although GES usually generates structures

with higher BIC values than K2, the difference on the BIC values is not very significant despite the restrictions defined by K2 ordering.

Additionally, we can extract some differences between the baseline structures and these alternative structures. In contrast with the baseline model, the structures learned by K2 and GES reveal that there are direct dependencies between the variables of the same group (control variables and radioisotope variables). Furthermore, the resulting models have less number of parameters to be learned, hence they are simpler, than the baseline model. This can be seen in Table 4, which compares the complexity of the different models.

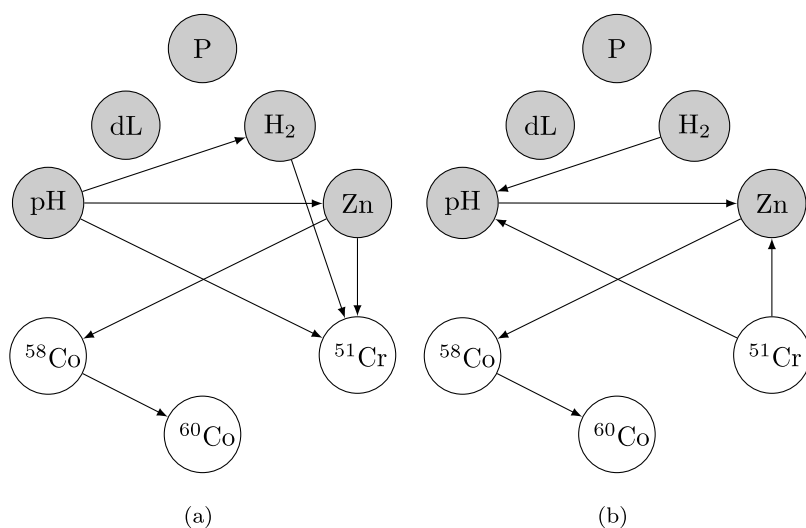


Fig. 15. Final alternative models for Plant 3 (a) Generated by K2 (b) Generated by GES.

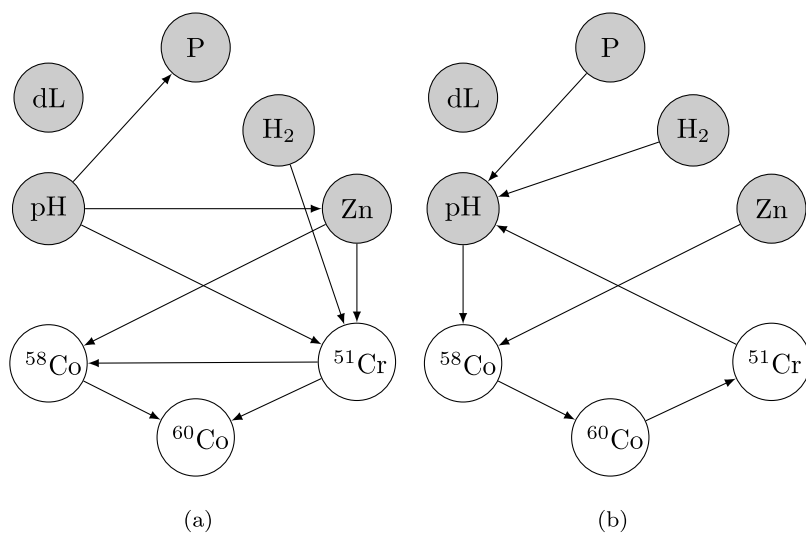


Fig. 16. Final alternative models for Plant 4 (a) Generated by K2 (b) Generated by GES.

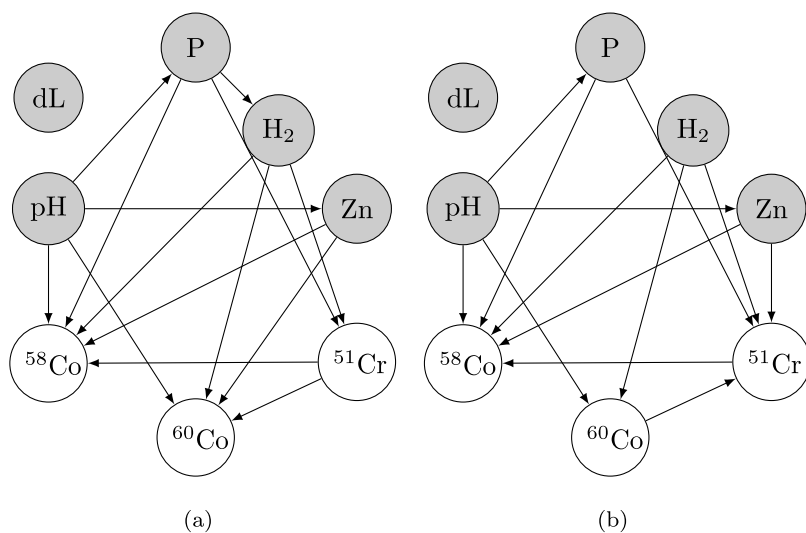


Fig. 17. Final alternative models for Plant 5 (a) Generated by K2 (b) Generated by GES.

Table 3

BIC score for each model.

	Plant 1	Plant 2	Plant 3	Plant 4	Plant 5
Baseline	−3146.389	−3977.062	−2438.119	−3608.490	−3399.339
K2	−2813.749	−3639.365	−2246.651	−3339.055	−3300.035
GES	−2806.529	−3638.939	−2244.031	−3334.705	−3300.597

Table 4Complexity in terms of number of trainable parameters (β, σ^2) for each model.

	Plant 1	Plant 2	Plant 3	Plant 4	Plant 5
Baseline	26	26	31	31	31
K2	23	23	23	25	30
GES	21	23	22	23	29

Finally, with a fine-grained analysis of the structures, certain dependencies are found to be difficult to chemically interpret. This behavior only occurs in the structures learned by Greedy equivalence search. In contrast, the definition of a topological ordering in K2 allows the introduction of some expert knowledge about the problem, which facilitates the interpretation of the resulting structures.

7. Conclusions

Bayesian networks framework has proven to be a powerful methodology for data interpretation and inference of some latent variables from observations of other variables. In certain scenarios, the probabilistic reasoning, the flexibility, and the easy interpretability of these models, make them stand out from other models such as neural networks (NNs) or Gaussian processes (GPs). The primary loop chemistry problem is a clear example of such scenarios. In this context, previous works have demonstrated the capabilities of Gaussian Bayesian networks (GBNs) to predict the concentration of the main corrosion products in a PWR reactor, from certain control variables. However, the structure of the GBN models proposed was pre-fixed by expert knowledge. This fact implies the definition of strong assumptions, regarding the relations of the different chemical variables involved in the problem.

In this work, we propose alternative GBN models, whose structures are learned in a data-driven way. To do that, three structure learning algorithms have been compared: Greedy-hill Climbing, K2, and GES. The resulting models have proven to be more consistent with unseen data than the models with a pre-fixed structure. Thus, the models proposed in this work represent a more calibrated and reliable alternative in the corrosion products prediction task. Furthermore, these models represent themselves as a very useful tool for chemical experts in the field, because they encode, in an interpretable way, the existing relations of the chemical variables in the primary loop chemistry. In this way, it is expected that the proposed models facilitate the definition of procedures that may allow, for example, the reduction of infrastructure degradation, the optimization of the reactor operation, and the reduction of the final radiation dose.

CRedit authorship contribution statement

Pablo Ramirez-Hereza: Conceptualization, Methodology, Software, Formal analysis, Writing – original draft, Visualization, Investigation. **Daniel Ramos:** Conceptualization, Methodology, Formal analysis, Writing – review & editing, Supervision. **Doroteo T. Toledano:** Writing – review & editing. **Joaquin Gonzalez-Rodriguez:** Writing – review & editing. **Alicia Ariza-Velazquez:** Data curation, Resources, Writing – review & editing. **Nuria Doncel:** Data curation, Resources, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

The authors from the UAM have been supported by the Spanish Ministerio de Ciencia e Innovación, Agencia y del Fondo Europeo de Desarrollo Regional (grant reference PID2021-125943OB-I00, MCIN/AEI/10.13039/501100011033/FEDER, UE). The work has been conducted in the context of a signed collaboration agreement between AUDIAS-UAM and ENUSA Industrias Avanzadas S.A.

References

- [1] S. Glasstone, A. Sesonske, *Nuclear Reactor Engineering. Reactor Systems Engineering*, Springer, 1984.
- [2] *Coolant Technology of Water Cooled Reactors, Volume 1: Chemistry of Primary Coolant in Water Cooled Reactors*, in: TECDOC Series (no. 667) International Atomic Energy Agency, Vienna, 1992.
- [3] D. Ramos, P. Ramirez-Hereza, D.T. Toledano, J. Gonzalez-Rodriguez, A. Ariza-Velazquez, D. Solis-Tovar, C. Muñoz-Reja, Dynamic Bayesian networks for temporal prediction of chemical radioisotope levels in nuclear power plant reactors, *Chemometr. Intell. Lab. Syst.* 214 (2021) 104327, <http://dx.doi.org/10.1016/j.chemolab.2021.104327>.
- [4] B. Scholkopf, The kernel trick for distances, in: *Advances in Neural Information Processing Systems*, Vol. 13, MIT Press, 2001, pp. 301–307.
- [5] C.E. Rasmussen, C.K.I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*, The MIT Press, 2005.
- [6] G. Schnabel, R. Capote, A. Koning, D. Brown, Nuclear data evaluation with Bayesian networks, 2021, <http://dx.doi.org/10.48550/arXiv.2110.10322>, arXiv preprint [arXiv:2110.10322](https://arxiv.org/abs/2110.10322).
- [7] Y. Zhao, J. Tong, L. Zhang, Rapid source term prediction in nuclear power plant accidents based on dynamic Bayesian networks and probabilistic risk assessment, *Ann. Nucl. Energy* 158 (2021) 108217, <http://dx.doi.org/10.1016/j.anucene.2021.108217>.
- [8] S. Chen, L. Zhang, T. Qing, X. Liu, Use of Bayesian networks and improved SPAR-H for quantitative analysis of human reliability during severe accidents mitigation process in nuclear power plant, *J. Nucl. Sci. Technol.* 58 (10) (2021) 1099–1112, <http://dx.doi.org/10.1080/00223131.2021.1915893>.
- [9] Y. Zhao, J. Tong, L. Zhang, G. Wu, Diagnosis of operational failures and on-demand failures in nuclear power plants: An approach based on dynamic Bayesian networks, *Ann. Nucl. Energy* 138 (2020) 107181, <http://dx.doi.org/10.1016/j.anucene.2019.107181>.
- [10] D. Koller, N. Friedman, *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*, The MIT Press, Cambridge, MA, USA, 2009.
- [11] M. Scutari, C.E. Graafland, J.M. Gutiérrez, Who learns better Bayesian network structures: Accuracy and speed of structure learning algorithms, *Internat. J. Approx. Reason.* 115 (2019) 235–253, <http://dx.doi.org/10.1016/j.ijar.2019.10.003>.
- [12] D. Colombo, M.H. Maathuis, et al., Order-independent constraint-based causal structure learning, *J. Mach. Learn. Res.* 15 (1) (2014) 3741–3782.
- [13] D. Margaritis, *Learning Bayesian Network Model Structure from Data*, Tech. Rep., Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science, 2003.
- [14] F. Glover, M. Laguna, Tabu search, in: *Handbook of Combinatorial Optimization*, Springer, 1998, pp. 2093–2229, <http://dx.doi.org/10.1287/ijoc.1.3.190>.
- [15] G.F. Cooper, E. Herskovits, A Bayesian method for the induction of probabilistic networks from data, *Mach. Learn.* 9 (4) (1992) 309–347, <http://dx.doi.org/10.1007/BF00994110>.
- [16] D.M. Chickering, Optimal structure identification with greedy search, *J. Mach. Learn. Res.* 3 (Nov) (2002) 507–554.
- [17] M. Gasse, A. Aussem, H. Elghazel, An experimental comparison of hybrid algorithms for Bayesian network structure learning, in: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2012, pp. 58–73, http://dx.doi.org/10.1007/978-3-642-33460-3_9.
- [18] I. Tsamardinos, L.E. Brown, C.F. Aliferis, The max-min hill-climbing Bayesian network structure learning algorithm, *Mach. Learn.* 65 (1) (2006) 31–78, <http://dx.doi.org/10.1007/s10994-006-6889-7>.

- [19] T. Verma, J. Pearl, in: P. Bonissone, M. Henrion, L. Kanal, J. Lemmer (Eds.), *Equivalence and Synthesis of Causal Models* In Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence, Elsevier Science, New York, NY, 1990, pp. 220–227, <http://dx.doi.org/10.48550/arXiv.1304.1108>.
- [20] J. Pearl, *Bayesian Networks: A Model of Self-Activated Memory for Evidential Reasoning*, Tech. Rep. CSD-850017, UCLA, 1985.
- [21] R.G. Cowell, A.P. Dawid, S.L. Lauritzen, D.J. Spiegelhalter, *Probabilistic Networks and Expert Systems*, Springer, 1999, <http://dx.doi.org/10.1007/b97670>.
- [22] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
- [23] D. Geiger, D. Heckerman, Learning gaussian networks, in: *Uncertainty Proceedings 1994*, Elsevier, 1994, pp. 235–243, <http://dx.doi.org/10.1016/B978-1-55860-332-5.50035-3>.
- [24] M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, L.K. Saul, An introduction to variational methods for graphical models, *Mach. Learn.* 37 (1999) 183–233, <http://dx.doi.org/10.1023/A:1007665907178>.
- [25] C. Li, S. Mahadevan, Efficient approximate inference in Bayesian networks with continuous variables, *Reliab. Eng. Syst. Saf.* 169 (2018) 269–280, <http://dx.doi.org/10.1016/j.ress.2017.08.017>.
- [26] N. Friedman, D. Koller, Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks, *Mach. Learn.* 50 (1) (2003) 95–125, <http://dx.doi.org/10.1023/A:1020249912095>.
- [27] Z. Liu, B. Malone, C. Yuan, Empirical evaluation of scoring functions for Bayesian network model selection, in: *BMC Bioinformatics*, Vol. 13, No. 15, Springer, 2012, pp. 1–16.
- [28] L. Dinh, J. Sohl-Dickstein, S. Bengio, Density estimation using real NVP, 2016, [arXiv:1605.08803](https://arxiv.org/abs/1605.08803).
- [29] P. Leray, O. Francois, *BNT Structure Learning Package: Documentation and Experiments*, 2004.