International Journal of Machine Learning and Cybernetics (2023) 14:229–251
https://doi.org/10.1007/s13042-022-01577-9

ORIGINAL ARTICLE

Merits of Bayesian networks in overcoming small data challenges: a meta-model for handling missing data

Hanen Ameur[1,2] · **Hasna Njah**[1,3] · **Salma Jamoussi**[1,2]

Received: 6 July 2021 / Accepted: 8 May 2022 / Published online: 27 June 2022

Abstract
The abundant availability of data in Big Data era has helped achieving significant advances in the machine learning field. However, many datasets appear with incompleteness from different perspectives such as values, labels, annotations and records. By discarding the records yielding ambiguousness, the exploitable data settles down to a small, sometimes ineffective, portion. Making the most of this small portion is burdensome because it usually yields overfitted models. In this paper we propose a new taxonomy for data missingness, in the machine learning context, along with a new metamodel to address the missing data problem within real and open data. Our proposed methodology relies on a H2S Kernel whose ultimate goal is the effective learning of a generalized Bayesian network from small input datasets. Our contributions are motivated by the strong probabilistic foundation of the Bayesian network, on the one hand, and on the ensemble learning effectiveness, on the other hand. The highlights of our kernel are the new strategy for multiple Bayesian network structure learning and the novel technique for the weighted fusion of Bayesian network structures. To harness on the richness of the merged network in terms of knowledge, we propose four H2S-derived systems to address the missing values/records impacts involving the annotation, the balancing, missing values imputation and data over-sampling. We combine these systems into a meta-model, and we perform a step-by-step experimental study. The obtained results showcase the efficiency of our contributions to deal with multi-class problems and with extremely small datasets.

Keywords Bayesian network · Missing data · Ensemble learning · Structure fusion · Small data · Imbalance data · Data generation

1 Introduction

The Fourth Industrial Revolution had significantly boosted various techniques of capturing enormous flow of multiple-typed data issued from a panoply of sources. Obviously, these large datasets, commonly known as Big Data, present a richness in terms of volume and variety. This richness incites the data analysts to develop scalable and robust algorithms for extracting the underlying knowledge of the data. Nevertheless, the concept of large datasets remains a double-edged sword because of the questionable quality and relevance of such datasets. Among others, missingness is a major issue that hampers the efficient exploitation of the data and impacts substantially the extracted knowledge quality.

The effect of missing data is dramatic on the quality of the learnt models. Actually, most of the Machine learning algorithms require complete input data in order to achieve representative models [39]. Missing features' values and/or missing labels lead to misleading models and even to the inability to learn a model. The common practice aims at performing the considered analysis on the most representative complete subset of the input data. Different methods harness the information richness of the obtained small data to impute the missing values and/or to generate missing labels. Regrettably, this practice yields two main challenges. On the one hand, the discarded records and/or features, which contain a considerable rate of missing data, may be relevant and may carry important knowledge. On the other hand, most of the machine learning methods struggle with learning

✉ Hanen Ameur
ameurhanen@gmail.com

[1] Multimedia, InfoRmation Systems and Advanced Computing Laboratory, Sfax, Tunisia

[2] Present Address: Higher Institute of Computer Sciences and Multimedia, University of Sfax, Sfax, Tunisia

[3] Higher Institute of Computer Sciences and Multimedia, University of Gabes, Gabes, Tunisia

accurate models from small datasets [31, 50], in most of cases. The overfitting trap hinders the quality of the learnt model; hence, the lack of generalization and reliability.

Unlike the common Machine Learning techniques, the probabilistic graphical methods are based on the statistical characteristics of the small data. They provide a complete scheme of (conditional) probability distributions associated to the data features. The graphical probabilistic methods stand out as a reliable approach for knowledge extraction and representation under various constraints including the uncertainty, missingness and imbalance. Their learnt model, from input small data, is potentially generalized over a whole population. It maintains the original distribution of the features, which are initially affected by the missingness issues. Additionally, the graphical probabilistic methods offer the possibility to step into a more explainable model's representation through the user-friendly graphical abstraction (e.g. a directed graph).

Motivated by the robustness of the probabilistic graphical methods for learning efficient models from small data, we capitalize on the performance of Bayesian networks [47] in tackling the challenges of data missingness. These networks are defined as a composite entity of a directed acyclic graph and a set of conditional probability distributions [19]. The Bayesian networks are studied and exploited in different applications by dint of their strong statistical foundation and their representative associated graphical representation [29, 49, 62]. The polyvalence of the Bayesian networks achieved a noteworthy success by the capacity of the inference. Each feature's value is predictable by the inference: it means propagating the probability functions through the feature's parents, in the corresponding graph, by the application of the Bayes theorem.

In the present paper, we propose to reckon on the strong mathematical foundation of the Bayesian network in order to tackle the underlying challenges of data incompleteness. Our paper holds a systematic contribution. We, firstly, start by introducing a new *missing data taxonomy* in order to differentiate the different cases of missingness including missing values and missing records. We introduce a new *meta-model* that holds a full pack of systems, as well as their smooth exploitation framework. Then, we present a new kernel, referred as H2S, for learning generalized Bayesian network to handle missingness in large input data. It raises on the ensemble-based Bayesian network learning, the structure-driven weighting and the strategic optimized fusion to mitigate the impact of over-fitting. We consider this procedure as the kernel of our contribution. The learnt model is potentially a generalized synthetic Bayesian network that accurately describes the small complete portion of an input data. Subsequently, we bring forward the extensibility and the reproducibility of our kernel in order to handle different cases of data missingness . Afterwards, we propose new H2S-based systems as remedies to the different types of missing data (variables and/or records) challenges. Depending on each system's mission, we alter between various ensemble-based strategies including, boosting and weighted voting. Finally, to showcase the merits of the Bayesian network in handling the challenges derived from learning models based on small data, we normalize the ensemble of our H2S-based systems according to our proposed meta-model.

The upcoming sections of our paper inaugurate by reviewing the most relevant related research studies. We propose a new taxonomy and a new meta-model for handling data missingness. Then, provide a theoretical background on Bayesian network including the scoring functions, learning algorithms and Markov blanket concept. Accordingly, we elaborate on our main contributions starting from detailing H2S kernel's steps, arriving at the proposal of a new Bayesian network meta model for handling missing data. Meanwhile, we emphasize on the extensibility of the kernel's brick stones into four systems. Later, we carry out an exhaustive experimental study including the datasets' description, the experimental protocol and the results' analysis. We wrap up the paper by summarizing its main contributions and providing some future work.

2 Literature review

At a first glance, the challenge of data missingness in large datasets seems to be contradictory. Commonly, large data means the abundance of records and features in a way to integrally model a real-life situation. Yet, with the increasing cost of data acquisition, large datasets present different types of missing data ranging from distinct values to entire records. The impact of data missingness on learning and exploiting the underlying model raises the researchers' attention, in a variety of studies. To deal with incomplete features' values, the practical approaches mainly consider the types of missingness (e.g. MAR and NMAR) [33, 37], on different data types (e.g. nominal and numerical) and the rate of missing values. Depending on the goal of study, data imputation can rely on the statistical characteristics of the features (e.g. replace with the average, for numeric features, and with the most frequent value, for nominal features) and/or on the machine learning methods such as generative models [28, 78]. The severity of this type of incompleteness becomes portentous when it occurs in the class variable; hence, the obtainment of data with missing labels.

Although the challenge of missing labels seems unusual, many large datasets describing massive-scaled applications, such as social network mining and medical assisting, consist of a considerable rate of missing labels among the records. Generally, this phenomenon occurs when data labeling is expensive in terms of time, effort and cost.

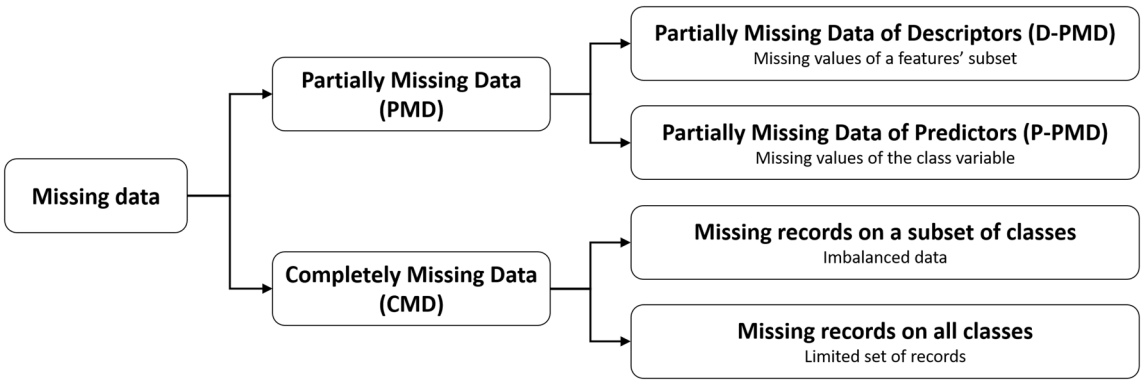


Fig. 1 Missing data taxonomy

Essentially, tackling this challenge is the fundamental aim of the semi-supervised learning. The literature introduces various methods categorized into the semi-supervised classification taxonomy [71]. Depending on the modality of data processing, some underlying approaches focus on the intrinsic characteristics of the data. They handle the missing labels through data perturbation [8, 42], enhancing maximum margin between the data points and decision boundary [2] and manifold regularization over generative models [83]. Other approaches make the most from the labeled portion of the dataset by adopting the wrapping principle over base classifiers. Hence, the used classifier(s) is (are) iteratively trained on the labeled and the pseudo-labeled portions of the input data according to various fashions. Among the oldest and the most widely known wrapper methods, we distinguish self-training [13, 57, 66, 76], co-training [74, 77, 80] and boosting [6, 38, 82]. These methods use a base learner for pseudo-labelling the records and rely on this learner's aptitude to correctly distinguish between the classes and to effectively scoring the labels. Obviously, the key characteristic of these methods is the faculty of choosing the most appropriate base learner. However, as the labeled portion of the input dataset tends to be relatively small, the usual base classifiers tend to fall into the overfitting trap. Although the boosting methods are theoretically the most suitable remedy to this issue, they tend to magnify the misclassification especially for multi-classification tasks.

Along with the optimizing the compromise between the obtainment of reliable classifiers and the overfitting, imbalanced data constitutes a significant challenge. It refers to the case of learning from data having the property of missing records over a single (or multiple) minority class(es). Basically, an unbalanced dataset will bias the predictive model towards the majority class. Properties, such as the severely skewed class distribution and the unequal misclassification costs, often deteriorate the classification performance. Basically, data pre-processing via under-sampling and over-sampling addresses the classification imbalance problem. In the former, the records' set of the majority class is approximately reduced to the cardinality of the minority class and, in the latter, the minority class is used to generate a random larger sample that grows up to the size of the majority class. Between information loss and over-fitting, both sampling approaches would not deal with impact of "the lack of records in large datasets". To ensure the compromise between these challenges, practitioners opt for hybrid methods (e.g., combining over-sampling and under-sampling), class-wise methods (e.g., Synthetic Minority Oversampling TEchnique (SMOTE) [17, 21]) and model-based methods (e.g., Bayesian network imputation [53]). To cope with severe imbalance, ensemble learning demonstrates a high potential reducing the impact of the skewed distribution [16, 75].

3 New taxonomy and meta-model for handling data missingness

3.1 Missing data taxonomy

Commonly, the data acquired from real-world applications is far from the perfect complete dataset. Actually, it is more likely that the large datasets contain a considerable amount of not assigned data. According to our exploratory analysis, we categorize data missingness into two main categories: Partial Missing Data (PMD) and Complete Missing Data (CMD). We present our proposal of missing data taxonomy in Fig. 1.

Partially Missing Data (PMD) corresponds to the case where at least one attribute of the dataset entails one (or multiple) record(s) having missing values. In this case, a record can present missing values of some descriptors (i.e. the descriptive features) and/or of a predictor (i.e. the class variable). In the former, we deal with Descriptor Partial Missing Data (D-PMD) where the data incompleteness is usually random (a.k.a. missing at random (MAR) [27]) due

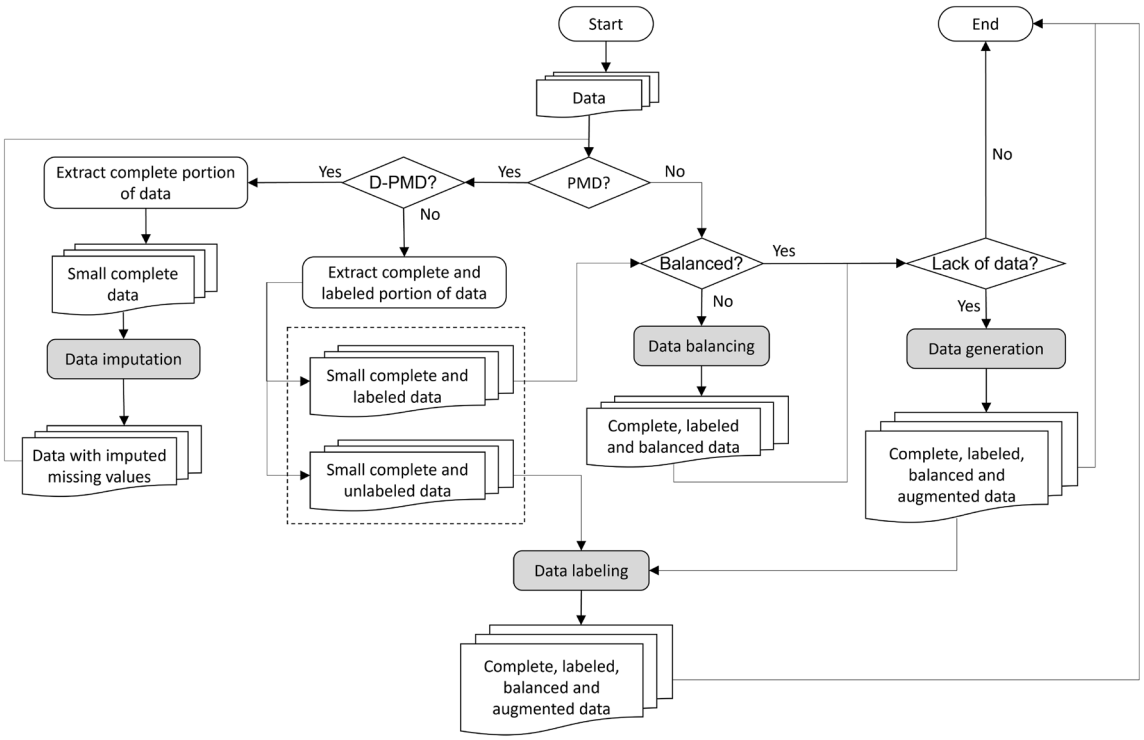


Fig. 2 Meta-model for handling data missingness

to data corruption, data acquisition anomaly, mechanical issues, etc. Furthermore, D-PMD can be missing not at random (MNAR) [27, 73] where the incompleteness underlies a "meaning" hence, the blank fields are left null for a reason. For example, in a crime-related questionnaire, some witnesses could potentially leave fields blank if they have a direct or indirect relationship to the crime.

In the latter, i.e. Predictor Partial Missing Data (P-PMD), the data is not fully labelled. The most accurate scenario of this situation occurs in diverse critical applications (e.g. medical applications). Consequently, the hand-crafted labelling requires the intervention of experts' skills, which are consuming in terms of time, effort and cost. Usually, the practitioners are based on the small labeled subset of the data for annotating the remaining records. They rely on the semi-supervised learning-based methods for data labeling.

Completely Missing Data (CMD) corresponds to the case where a set of whole records is missing. The CMD lack of data presents a classical challenge for machine learning algorithms. The instances factorization for is crucial for determining the combinations of the features' values in order to ensure the model generalization. Particularly, the methods that optimize the empirical risk (e.g. the Artificial Neural Networks and their variants) are extremely greedy for larger training dataset. Trying to mimic the Human brain, these methods require as many observations as possible in order to achieve a reliable performance. Along with the empirical approach, the structural methods (e.g. Support Vector Machines) struggle with adapting the parameters of the "optimal" structure learning using small data. At variance, the probabilistic methods are robust to the lack of data as they only require a smallest representative records' sample to escape the over-fitting trap.

Particularly, the lack of data in a given class yields the classical challenge of *data imbalance*. In the general case, the instances of the imbalanced datasets do not approximately have equal classes counts. Thus, the observations' labels are categorized into majority and minority classes [36]. Following this particular case of CMD, the supervised machine learning methods tend to produce class-wise over-fitted models; the learning process is biased towards the majority class [81]. This performance deterioration is explained by the fact that the small data subset that represents the minority class/es is not sufficient to learn the corresponding characteristics [12].

3.2 Meta-model for handling data missingness

Each type of data missingness yields specific challenges, which are tackled by distinct procedures. According to our

proposed taxonomy in Fig. 1, we synthesize these procedures into: Data imputation, Data labeling, Data balancing and Data generation. We rely on the *activity diagram* of Unified Modeling Language[1] to propose a new meta-model for missing data handling (see Fig. 2).

A user aiming at tackling a real-world application starts by checking if the input data (i.e., a data that presents any type of missingness) is PMD. Subsequently, a positive test of D-PMD yields the application of *"data imputation"* while a negative test yields the application of *"data labeling*. Otherwise, an imbalanced dataset is the input of multi-class *data balancing* module. Finally, a dataset that satisfies all the previously mentioned tests is checked for the lack of data challenge, independently from the class variable. In such case, *data generation* constitutes the solution to this issue.

Following our proposed meta-model, we reckon on the Bayesian network framework (see Sect. 4) for establishing a new kernel, called H2S, that learns generalized models for representing data with different types of missingness (see Sect. 5). Subsequently, we extend our kernel into four systems for performing data imputation, data labeling, data balancing and data generation (see Sect. 6).

4 Bayesian network: theoretical background

In the machine learning context, the dataset is the feedstock for learning the model that reflects the underlying patterns and regularities. Commonly, a dataset is formally described as set of features (i.e. descriptors or attributes) used for representing a set of records (i.e. instances). For the supervised studies, a particular feature, known as "class" or "target", discriminates the records into classes.

Definition 1 (*Data formulation*) A dataset is represented as a tuple $T = (X, N)$ where:

- X is a n-dimensional finite random vector that represents the features' set. Therefore, X is represented as $X = \{X_i : 1 \leq i \leq n\} \cup \{X_c\}$ such that n is the total number of features and X_c is the class where $X_c = \{X_{c_k} : k > 1\}$. For binary classification, we have $k = 2$.
- N denotes the total number of records.

4.1 Bayesian network

Bayesian Network is graphical and probabilistic model that is generally learnt from data to represent (conditional) dependencies of variables and their corresponding (conditional) probability distributions [19].

[1] https://www.uml.org/.

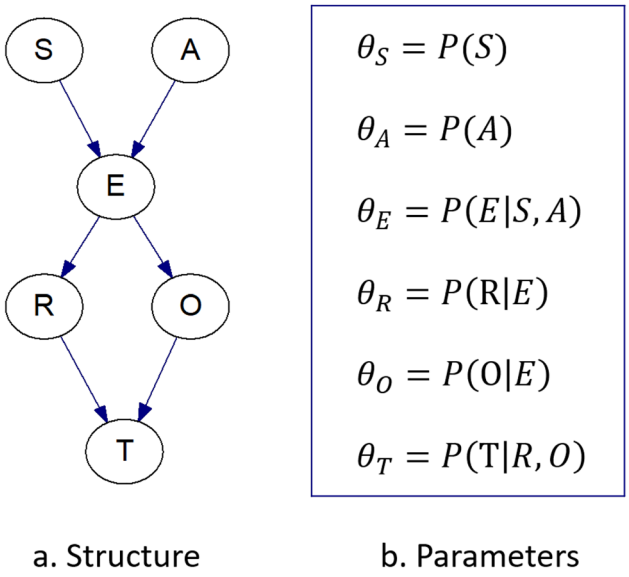


Fig. 3 Example of "Survey" Bayesian network

Definition 2 (*Bayesian network*) A Bayesian Network (BN) is a tuple $B = (G, \Theta)$ where:

- $G = (V, E)$ is a directed acyclic graph that represents the structure of B where
  - $V = \{V_i : 1 \leq i \leq n+1\}$ denotes the vertices (i.e. nodes) representing respectively the features X such that each feature X_i is associated to a vertice V_i and
  - $E = \{(V_i \rightarrow V_j) : V_i, V_j \in V\}$ constitutes the edges representing direct dependencies between the vertices.
- $\Theta = \{\theta_{ijk} : 1 \leq i \leq n+1, j \in D_{pa(V_i)}, k \in D_i\}$ represents the conditional probability tables, a.k.a. the parameters, of the Bayesian network B using the following formula:

$$\theta_{ijk} = P\big(X_i = x_{ik} pa(V_i) = w_{ij}\big) \tag{1}$$

  - $D_p a(V_i)$ is the joint domain of all the features in $pa(V_i)$ which is the set of parents of the node V_i in G
  - x_{ik} is the k *th* value of X_i and w_{ij} is the j *th* configuration of $pa(V_i)$.

Example 1 In Fig. 3, we present the example of Survey[2] benchmark BN. The network is composed of 6 nodes whose corresponding structure is visualized, in Fig. 3a. For each feature, the conditional probability distributions are enclosed

[2] Survey network is downloadable, from BNLEARN website, in different formats at: https://www.bnlearn.com/bnrepository/discrete-small.html#survey.

into the parameters (see in Fig. 3b) based on their parents in the structure.

BN learning is achieved by *structure learning*, which enables finding the best directed acyclic graph G that describes the data, followed by *parameter learning*, which ensures fitting the conditional probability tables to the data and to the learnt structure. To learn the structure, *constraint-based* methods search for conditional independencies between the features using various metrics (e.g., Mutual information) [48, 64] while *score-based* methods explore the space of possible structures and return the candidate that optimizes a scoring metric [11, 20, 26]. *Hybrid* methods combine the advantages of both methods with a local search of the conditional independencies and a global optimization of the score [41, 63, 70]. To perform parameters learning the statistical methods, such as Maximum likelihood Estimation [54], and Bayesian methods, such as maximum a posteriori estimation [30], are usually applied.

4.2 Scoring functions of Bayesian network

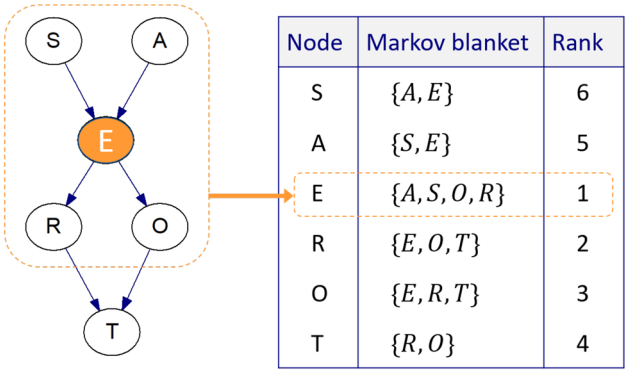
Generally, BN scores measure the degree of fitness between the structure and the input training data [4]. These scores are categorized into information-theoretic scores, which are scores derived from Log Likelihood measure [56], and Bayesian scores, which are derived from Bayesian Dirichlet Score [22]. The first family is based on Occam's razor: "Given two equally predictive theories, choose the simpler". Therefore, they provide more optimal complexity-fitness trade-off. The most known metrics are Bayesian Information Criterion (BIC) [61] and Akaike's Information Criterion (AIC) [60]. As for the second family of BN scoring functions, the maximization of the posterior probability is the key element to assess the structure's fitness to the data. The most known functions are K2 score [10] and Bayesian Dirichlet Equivalent score (BDe) [22]. Basically, the optimal BN (sub)structure entails a minimized score.

4.3 Markov blanket

Having a learnt BN, the robustness of its structure, globally, and its features, particularly, can be determined by Markov blanket [43]. This concept takes part in the organization of the directionality affectation task for each node for constraint-based structure learning.

Definition 3 (*Markov blanket*) Let a BN be a tuple $B = (G, \Theta)$, where $G = (V, E)$ is the structure and Θ denotes the parameters, the Markov blanket of a node $V_i \in V$ is any subset $M \subseteq V$ such that V_i is conditionally independent of the other nodes in M:

$$V_i \perp\!\!\!\perp V - M | M$$



Node	Markov blanket	Rank
S	$\{A, E\}$	6
A	$\{S, E\}$	5
E	$\{A, S, O, R\}$	1
R	$\{E, O, T\}$	2
O	$\{E, R, T\}$	3
T	$\{R, O\}$	4

Fig. 4 Example of Markov blanket of "Survey" Bayesian network

Example 2 In Fig. 4, we represent the list of Markov Blankets of all the nodes in Survey benchmark BN. For example, node E is holding a maximized informative characteristic in the network as it has the largest Markov Blanket, which is composed of A, S, O, R.

5 Proposed H2S Kernel

Lack of data, including records and/or labels, is the main challenge of learning models given large input datasets. Following our theoretical presentation (see Sect. 3), the probabilistic aspect of the Bayesian network offers a promising remedy to this problem. We propose to reckon on the ensemble learning approach to mitigate the lack of records' impact, on the one hand, and to optimize the records labeling, on the other hand. Evidently, in-depth studies of the ensemble learning's benefits [14, 58] have proven the merit of combining several learning algorithms in generalizing the learnt model. These algorithms include data perturbation by bootstrap sampling [55], genetic algorithms [23], and committee-based learning [7]. Inspired by the impressive results of the ensemble learning, we propose a new generic BN-based kernel, called H2S, for handling missingness in large data. We ensure that H2S kernel is extensible to deal with different types of lack of data underlying issues (see Fig. 1). As described in Fig. 5, the framework of H2S kernel is the succession of three main steps: multiple structure learning, weighted structures' fusion then the merged structure's parameters learning.

5.1 Multiple structure learning

In this step, we suggest applying various BN structures' learning algorithms in order to diversify the candidate models and to mitigate the impact of over-fitting. The obtainment of various structures, when applying algorithms adopting

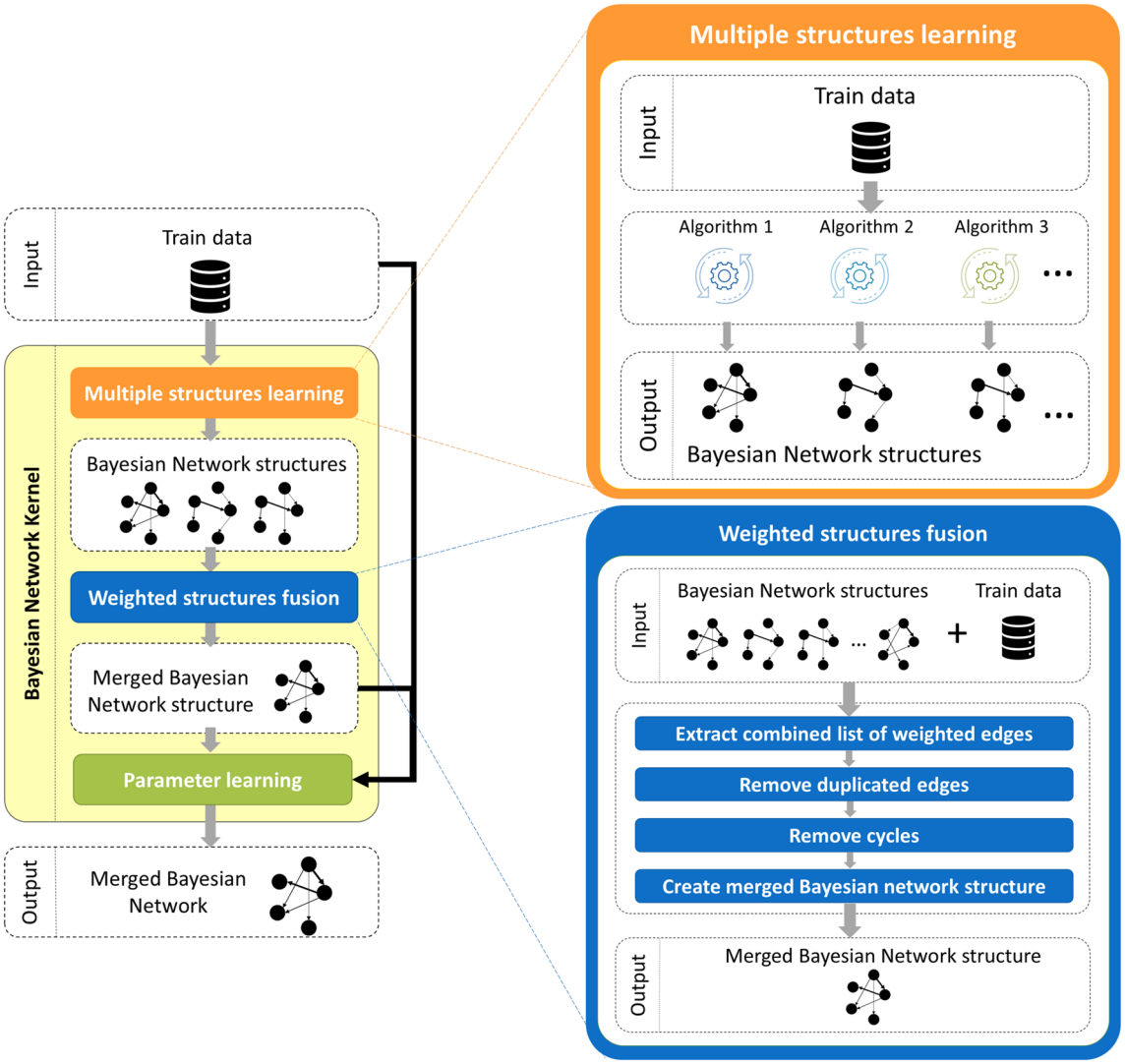


Fig. 5 Framework of H2S kernel

different strategies, is the highlight of our contribution. We aim at using the same training dataset for producing several "optimal" structures favoring the chances of spotting out prominent dependencies between pairs of features. Ideally, the combination of constraint-based methods, score-based methods and hybrid methods yields a global coverage of diverse candidate structures. Additionally, opting for more supervised alternatives, the BN-based classifiers, such as Tree Augmented Naïve Bayes (TAN) [18], insure a more class-based discriminative structure. Furthermore, the introduction of prior knowledge (i.e., priorly known dependencies between the features) potentially steers the learning to a more faithful-to-reality BN structure.

5.2 Weighted structures fusion

Having the set of the learnt structures, the second step of H2S kernel lines up to rigorously merge them into a global structure. The main characteristics of this latter involve the generic (vs. over-fitted) aspect of the model and the representativeness of the dependencies acquired from different backgrounds. To achieve this end, we aim at setting the seal on obtaining a directed acyclic graph while emphasizing on the edges' strengths, which are transmitted from the previous step. In the context of merging BNs learnt from the same training dataset, the main tendency focuses on simplistic methods (e.g., union and intersection of edges) [34], voting methods (e.g., majority vote) [44, 65] and optimization methods (e.g., add and/or remove arcs until convergence) [52]. Our vision for H2S kernel focuses on achieving a

compromise between the lightest structure (i.e., a structure with a less complexity) and the more informative structure. Therefore, we introduce "Weighted structure fusion" method, which entails four major phases, which are detailed in Algorithm 1.

Phase 1: Extract combined list of weighted edges. Obviously, the examination of the input multiple structures, which were obtained from the previous step of our H2S kernel, highlights their differences. Being learnt by algorithms having different strategies, each structure upholds its own importance's degree of the underlying edges. In other words, an important edge in a structure G_i may be less important in another structure G_j. Inspired by this reasoning, we propose to use BN scoring functions for quantifying the edges' degree of importance within a structure. We opt to use BN scoring functions (see Sect. 3.2) for assigning weights to each edge. Hence, the function *weighted_edges*(G_i, *score*) attributes the score value of an edge where the parameter "score" indicates the chosen BN scoring function. We recommend using equivalent scores in order to consider both possible directions of a given edge. Hence, BIC, AIC and K2 are more suitable as they are proven to be more practical [4]. Accordingly, the set E is composed of all the edges in all the considered BN structures.

Phase 2: Remove duplicated edges. Evidently, E likely contains redundant edges, i.e., the same edge appears in a structure G_i and in another structure G_j with potentially different scores and different directions. Our algorithm proceeds through the definition of E' as the set of the weighted edges that will be included in the merged BN structure. To decide on the "best" edge to add in E', we propose to rely on the duplicated edges' weights and directions. We define two local variables s_1 and s_2 containing the sum of the edges' scores in both directions, respectively. The edge's direction with a minimized summed score is considered in the merged BN structure E'. Otherwise, i.e., both directions have the same score, the resulting arc (undirected edge) is added to E'.

Algorithm 1: Weighted structures fusion

input : $G = \{G_i, 1 \leq i \leq g\}$ such that g is the number of the learnt structures
output: G^F: merged structure of G
E: empty list of weighted edges
$V \in G$: set of vertices (features)
for $k = 1$ **to** g **do**
  $E_k = weighted\_edges(G_k, score)$
  $E = E \cup \{E_k : E_k = \{e_{ij} : 1 \leq i, j \leq |V|,$ i$\neq j\}\}$
end
E': empty list of weighted edges
foreach $(v_i \rightarrow v_j)$ *in* E **do**
  $s_1 = \sum score(e_{ij} = (v_i \rightarrow v_j))$
  $s_2 = \sum score(e_{ji} = (v_j \rightarrow v_i))$
  if $s_1 > s_2$ **then**
    $E' = E' \cup \{e_{ji}\}$
  else
    $E' = E' \cup \{e_{ij}\}$
  end
end
$E' = sort(E')$
$G' = create\_graph(E')$
$C = cycles(G')$
while $C \neq \emptyset$ **do**
  $E' = E' \backslash \{\hat{e}_{ij} : \hat{e}_{ij} = \arg\min_{e_{ij} \in C_k}(score(e_{ij}))\}$
end
if G' *is partially directed* **then**
  V': empty list of weighted nodes
  foreach $(v_i \in V)$ **do**
    $V' = V' \cup \{(V_i, |markov\_blanket(V_i)|)\}$
  end
  $V' = sort(V')$
  $G^F = partially\_directed\_structure(V', E')$
  $G^F = pdag2dag(G^F)$
else
  $G^F = G'$
end

Phase 3: Remove cycles E' contains distinct edges; however, this set does not necessarily ensure an acyclic resulting graph G'. Finding the cycles in a (partially) directed graph is a NP-complete problem [68]. Despite the polynomial attempts to reduce the complexity of this problem [35, 45], breaking a cycle in a BN structure is a challenging task. We propose to wisely choose the edge/arc that should be removed in order to yield a (partially) directed acyclic graph version G' based on the edges in E'. Our algorithm comes to rank the edges according to their scores. Subsequently, it examines each cycle $C_k \in E'$ and removes the corresponding arc/edge e_{ij} having a maximized weight from E':

$$\hat{e}_{ij} = \arg\min_{e_{ij} \in C_k} \left(score(e_{ij}) \right) \tag{2}$$

This procedure is repeated until breaking all the cycles in G'.

Phase 4: Create merged BN structure. The produced structure G' likely holds a partially directed acyclic graph. Using constraint-based structure learning algorithms, in "Multiple structure learning" step of H2S kernel, reinforces the likelihood of the partial directionality. They tend to identify a non-directed graph that represents existent conditional dependencies between the variables by the mean of conditional independence test. Arcs' orientation, called $pdag2dag(G)$ in Algorithm 1, is ensured by detecting *V-structures* and propagating the directions in a manner to obtain a directed acyclic graph. Consequently, we propose a rigorous method for setting the importance degree of each node in order to guarantee that $pdag2dag(G)$ starts with the most informative node in the graph G'. To quantify this aspect, we count upon the *Markov Blanket* associated to each node. Actually, blankets with a larger size imply that the corresponding feature (i.e., node) is holding more information. In the case where the same node is connected to multiple undirected arcs, choose the direct neighbor node having the larger Markov blanket. As a result, the obtained fully directed acyclic graph G^F represents the merged BN structure.

5.3 Parameter learning

As a final step of H2S kernel, we learn the *parameters* of the obtained merged structure $B^F = (G^F, \Theta^F)$ via Maximum likelihood Estimation [54] such that:

$$\Theta^F = \arg\min_{\Theta} \left(L(\theta | B^F, T) \right) \tag{3}$$

where the likelihood (L) function of two variables U and V is defined as follows:

$$L(U|V) = -P(U|V) \log(P(U|V)) \tag{4}$$

6 H2S-based systems for data generation

The fundamental goal of H2S kernel encloses learning a generalized BN while escaping overfitting from large input data characterized by a considerable rate of missing records/labels. To cope with the underlying challenges, we propose to extend H2S kernel into four versions of systems, with respect to our proposed meta-model and our suggested taxonomy (see Sect. 3). These systems are founded on the principle of *inference in BN*.

Definition 4 (*Inference in BN*) The inference in a BN is the estimation of the a-posteriori probability of an outcome $O \subset X$ given an evidence $E \subset X$ such that $E \cup O = \emptyset$ where:

- an outcome O is represented by a query on a (or multiple) missing value(s) of some feature(s) for a given record and
- an evidence E is represented by the certainty (the knowledge) about known value(s) of other feature(s). Therefore, $P(E) = 1$.

Following this definition, the inference consists in finding the configuration O^* of O in a way to maximize the posterior probability of O given E. Hence, O^* is called *Most Probable Configuration* and it is formally defined as follows:

$$O^* = \arg\min_{O} \left(P_B(O|E) \right) \tag{5}$$

where P_B is the probability function of the corresponding BN denoted by B.

The exact inference follows an iterative process; it starts by fixing the values of the evidence then it propagates their probability to the neighboring nodes by respecting the directionality of the edges and by applying the Bayes theorem when needed [46]. This algorithm is computationally demanding in larger BNs as the exact inference is a NP-hard problem [9]. Recent studies [32, 69, 79] are beginning to provide new insights into optimizing the approximated inference. In this paper we propose to adopt the *Approximate Forward Sampling* (AFS) [1].

With reference to our proposed taxonomy of "missing data" (see Fig. 1), we distinguish four different cases of study. In the following sub-sections, we describe our extensions and alterations to H2S kernel in order to tackle the different types of data missingness, namely PMD (Missing values) and CMD (Missing records).

6.1 H2S kernel extension for PMD

The ordinary cases of applying the inference on a BN, are classification and missing values imputation.

Data labeling. In the context of semi-supervised learning, the input dataset is subjugated to a considerable rate of unlabeled records. As explained in Sect. 2, guessing the missing labels theoretically and practically improves [71] the future exploitation of the data in both predictive and descriptive analyses. We consider this problem as a classification task, where the "test set" is not labeled. The inference process, therefore, consists in assigning the most probable class value to a given record. Formally, for each record, we aim at estimating O^* according to formula 5 such that $O = X_c$ is the unknown class value (i.e., the missing label) and $E = X_i : 1 \leq i \leq n$. Motivated by the severe

consequences of learning classifiers from a limited number of records, we opt for the synergy between H2S kernel brick-stones and the promising results of the committee-based classification. As illustrated in Fig. 6, we consider the labeled portion of the data as a training set.

We reproduce the same principle of "Multiple structure learning" step of H2S kernel while considering BN classifiers instead of ordinary BN. The unlabeled data is subsequently fed to each classifier in order to determine the values of the missing labels. We combine the decisions of the classifiers in a weighted vote fashion by following Algorithm 2. The final class of each unlabeled record takes the value $\hat{x}_{ck}$ having the maximized weights' sum of the classifiers voting for the corresponding x_{ck}.

Algorithm 2: Weighted vote

input : $C = \{C_i, 1 \leq i \leq k\}$: the set of the learnt (fitted) classifiers
$T = T_U \cup T_L$: with T_U is the unlabeled data and T_L is the labeled data
output: T': labeled data
foreach *Classifier* $C_i \in C$ **do**
  $w_i = score(C_i, T_L)$
end
$W = normalize(W)$
$A = \{a_j : 1 \leq j \leq |T_U|\}$: the set of the predicted labels
foreach *Unlabeled record* $t_j \in T_U$ **do**
  $S = \{(s, x_{ck}) : x_{ck} \in D_c, s = 0\}$
  foreach *Classifier* $C_i \in C$ **do**
    $(s, \hat{x}_{ck}) = (s + w_i, predict(C_i, t_j))$
  end
  $(x_c)_j = \arg\min_{(s,.) \in S}(s, x_{ck}))$
end

Missing values imputation

We propose to deal with the case where the records are labeled, yet, they have missing values of some features. Formally, for each record, we aim at estimating O^* according to formula 5 such that $O = \{X_i : X_i \in X - X_c\}$ is the set of features having missing values and $E = \{X_j : X_j \in X - O\}$. Our strategy is straightforward and it entails a sequential aspect (see Fig. 8a). The merged BN obtained from the direct application of H2S kernel on the input dataset is used as the input of AFS algorithm. The resulting dataset is the imputed version of the original train dataset.

6.2 H2S kernel extension for CMD

Data balancing (i.e., Class dependent over-sampling) and data generation from a BN (i.e., Class independent over-sampling) are the most popular applications for handling this type of missing data.

Data balancing Classes imbalance deteriorates the classification performance by increasing the risk of the model's overfitting in favor of the majority class. Therefore, we aim at over-sampling the original input dataset by generating new records based on the minority class(es). Formally, for each record, we estimate O^* according to formula 5 such that $O = \{X_i : 1 \leq i \leq n\}$ is the set of all features and $E = \{X_c \neq x_c^{maj} k\}$ encloses only the class variable where $x_c^{maj} k \in X_c$ denotes the majority class of the input dataset. To concretize the inference's principle, we ponder on the extensibility of H2S kernel's steps, combined with boosting strategy, for handling multi-classification tasks. As illustrated, in Fig. 7, the framework of H2S-based data balancing system proceeds into four major phases:

1. We start by sampling balanced subsets from the original input data.
2. We refer to H2S kernel's steps such that the "Multiple Structures Learning" is applied on each sample to produce several BN structures. Instead of merging the structures of each sample aside, we propose to fuse all the BN structures at once. This rectification aims at optimizing the running time and at minimizing the bias through learning different structures fitted with different samples. Accordingly, the parameters of the globally merged BN structure are learnt by the input imbalanced data. Hence, all the probability distributions in the resulting BN respect the original probability distributions of the variables.
3. We use AFS algorithm for generating new records while emphasizing on two specifications namely the minority considered class(es) and the desired sample size. Ideally, this size is equal to the majority class' cardinality.
4. Finally, we merge the generated data with the original input data to produce the output of our algorithm.

Data generation

Along with CMD challenges, numerous recent applications demonstrate an insistent need of records' abundance due to the high cost of data acquisition. In this particular case, the lack of records strikes on all the classes. We aim to generate new records while preserving the statistical and probabilistic characteristics of the whole input data. Formally, for each record, we aim at estimating O^* according to formula 5 such that $O = X$ is the set of all features and $E = \emptyset$. To achieve this goal, we refer to H2S kernel for learning as

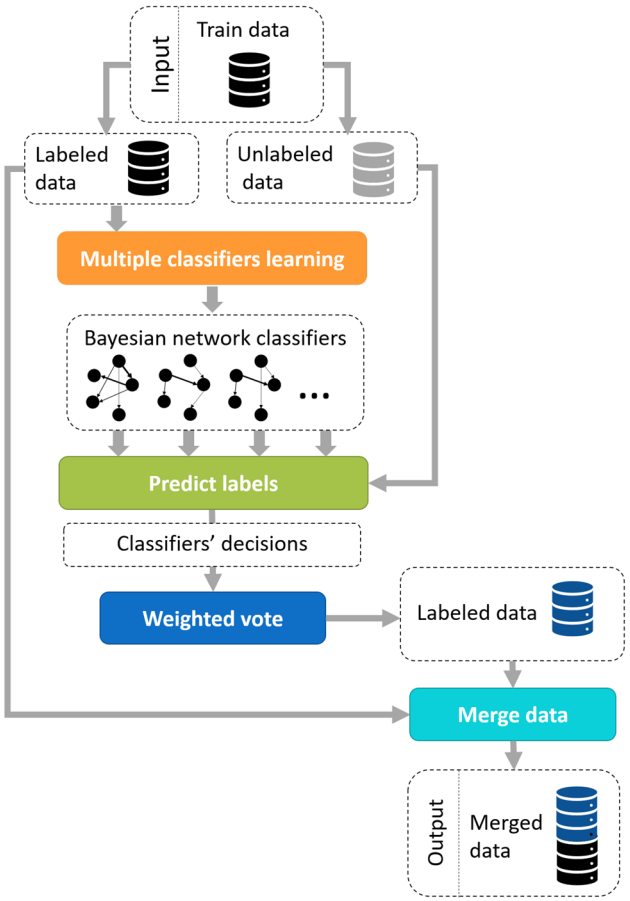


Fig. 6 Framework of data labeling system

more generic BN as possible. Subsequently, according to Fig. 8b, we apply AFS algorithm to generate the records. Contrary to the three above-mentioned solutions to deal with missing data, AFS algorithm is not initialized to an evidence ($E = \emptyset$). To set the starting node for generation and to deal with ties during inference, we propose to rely on our nodes' sorting procedure via Markov blanket size ranking.

7 Experimental analysis

To assess our H2S-based systems for dealing with data missingness, we propose to carry out a systemic experimental analysis over multiple real-world datasets. Therefore, we consider the evaluation of each H2S-based method for handling data missingness as an independent system according to the tackled issue (e.g., values imputation and record generation). Our experimental protocol provides a quantitative assessment that compromises between the properties and the performances of all the proposed H2S-based methods. Depending on the evaluated task, we report the adequate evaluation metrics.

7.1 Experimental protocol

7.1.1 Datasets

In this paper, we consider 12 datasets[3] collected from real-world applications. We consider a diversified collection of datasets with different layouts as described in Fig. 9. We present, in a detailed table (see Fig. 9a), the characteristics of these datasets including their names, their corresponding numbers of features, records and classes, as well as their Imbalance Ratio (IR) (the majority class' cardinality divided by the minority class' cardinality) and their Features-Records Ratio (FRR) (the count of records, T divided by the count of features, n). These ratios allow to quantify the degree of imbalance between the records' counts per class and the imbalance between the counts of features and records, respectively. We illustrate the distributions of these ratios in Fig. 9c, d. Additionally, to visualize the degree of imbalance between classes, we display in Fig. 9d the classes' proportions in each of the considered datasets. It convenes to highlight the fact that we split all the datasets into 70–30% train-test subsets. Thus, in the experiments of Sect. 6.2 use the same test set for assessment.

7.1.2 Compared algorithms

In our experimental analysis, we emphasize on the overfitting consequence of learning from small datasets. To evaluate our proposed H2S-based systems, we focus on assessing the efficiency of their output data in a classification task. Therefore, we consider classifiers that are greedy in terms of records and that implement different strategies. In our experimental study, we opted for the most simple and generic implementations in Weka software:[4]

- Multi-Layer Perceptron (MLP) is a feedforward artificial neural network presented as a fully connected network, which is composed of an input layer, an output layer and one (or many) hidden layer(s). To train a MLP classifier, the backpropagation algorithm updates the weights of the connections between the nodes in order to optimize a loss function [67]. A generous set of records allows more adequate weighting; hence, a better classification performance [40]. For our experimental analysis, we consider the default parameters of MLP classifier such that the learning rate is equal to 0.3, the epoch number is set to 500, the number of hidden layers is equal to $(n + |X_c|)/2$ and the Momentum is equal to 0.2.

[3] these datasets are available online via UCI Machine Learning repository https://archive.ics.uci.edu/ml/datasets.php.

[4] Weka is available at: https://waikato.github.io/weka-wiki/downloading_weka.

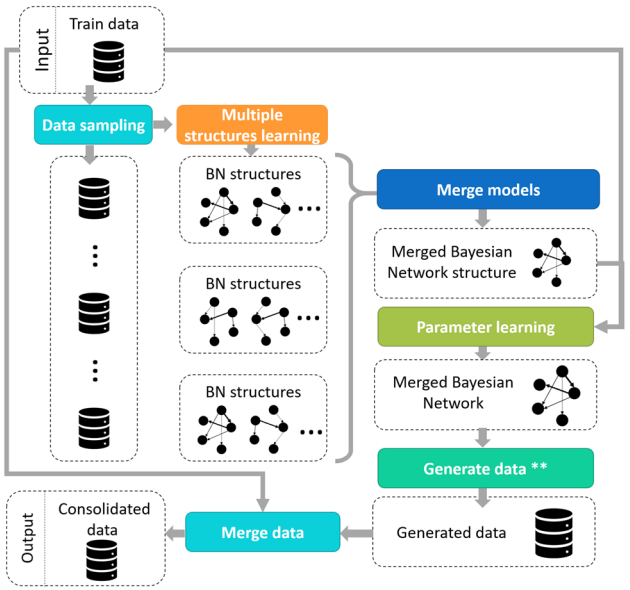


Fig. 7 Framework of H2S-based data balancing system

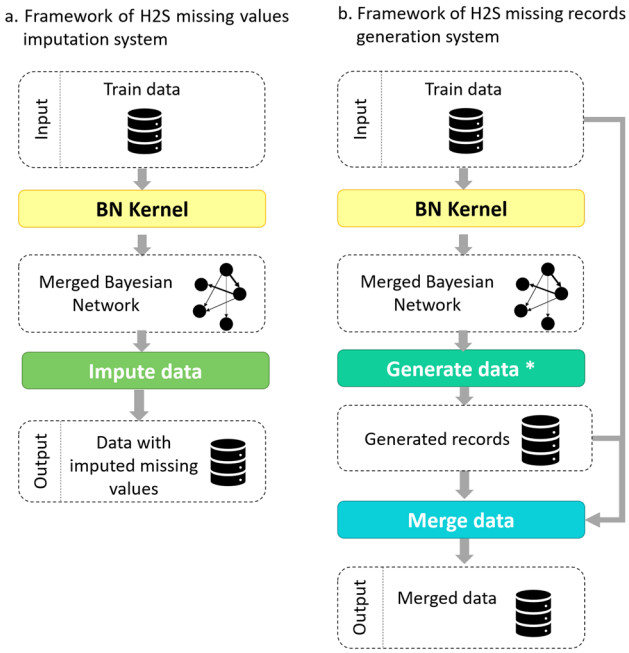


Fig. 8 Framework of unsupervised H2S-based systems

- Support Vector Machines (SVM) classifier aims at finding an "optimal" hyperplane, in n-dimensional space, that segments the data points into distinct classes. SVM is based on margin maximization and structural risk minimization. With sufficient data, SVM achieves a good generalization performance [72]. We employ SVM for Classification (SVC) using the radial kernel (RBF) with the parameter gamma of 0.1, cost of 0.1 and epsilon of 0.001.
- C4.5 is one of the most classic algorithms for learning a Decision Tree (DT) classifier [51]. This latter is represented through a hierarchical structure where the root represents the most segmenting feature, the leaves are the decision nodes (i.e., the classes) and the branches are the successions of tests to yield the classification. We consider J-48 implementation of C4.5 for learning an unpruned decision tree with a confidence factor of 0.25.

In the context of semi-supervised learning, our H2S-based labeling system is categorized among the self-training techniques. Accordingly, we propose to compare this contribution's performance to the following collective methods in the "collective" package[5] of Weka software:

- Collective EM (CEM) trains a base classifier on the labeled portion of the input data for annotating the unlabeled portion. Then, it runs an iterative routine for adjusting the labels with reference to the original distribution of the labeled data. In our experimental analysis, we used TAN as the base classifier, and we assumed that all the records are equally weighted ($q = 1$). For multi-class classification problems, we opted for the more generic version, called Collective Wrapper [71].
- Yet Another Two Stage Idea (YATSI) adopts the wrapper strategy, using a base classifier, combined with the nearest-neighbor-search of the unlabeled records [24]. We ensured that TAN is the used base classifier and Filtered Neighbor Search algorithm is used for finding the 10 nearest neighbors ($k = 10$).

Furthermore, we compare the performance of H2S-based systems for data sampling with the most known basic sampling methods. We used Random Over-Sampling Examples (ROSE) R package[6] for applying the following algorithms:

- Under-sampling reduces the number of records in the majority class(es) to obtain the same cardinality of the minority class.
- Hybrid sampling combines under-sampling and over-sampling to regulate the cardinalities of all the classes to k records per class (i.e., the parameter k is introduced by the user).
- SMOTE is an over-sampling method that replicates the principle of K-Nearest Neighbors to simulate the most similar records from k chosen instances of the minority class(es).

[5] The collective package is available online via: https://github.com/fracpete/collective-classification-weka-package.

[6] The documentation of ROSE is found at: https://cran.r-project.org/web/packages/ROSE/ROSE.pdf.

a. Datasets' description

Name	# Feature	# Records	#Classes	IR	FRR
Adult	14	32561	2	3.1	2325.7
Bank	16	45211	2	3.8	2825.6
Breast Tissue	9	106	6	1	11.7
Car	6	1728	4	4.4	288
Dermatology	34	366	6	1.3	10.7
Flag	28	194	6	1.1	6.9
Gas	128	3600	6	1	28.1
Glass	9	214	6	2	23.7
Hepatitis	18	154	2	2	8.5
Nursery	8	12960	4	13.1	1620
Sonar	60	208	2	1	3.4
Weight	57	4024	5	2.7	70.5

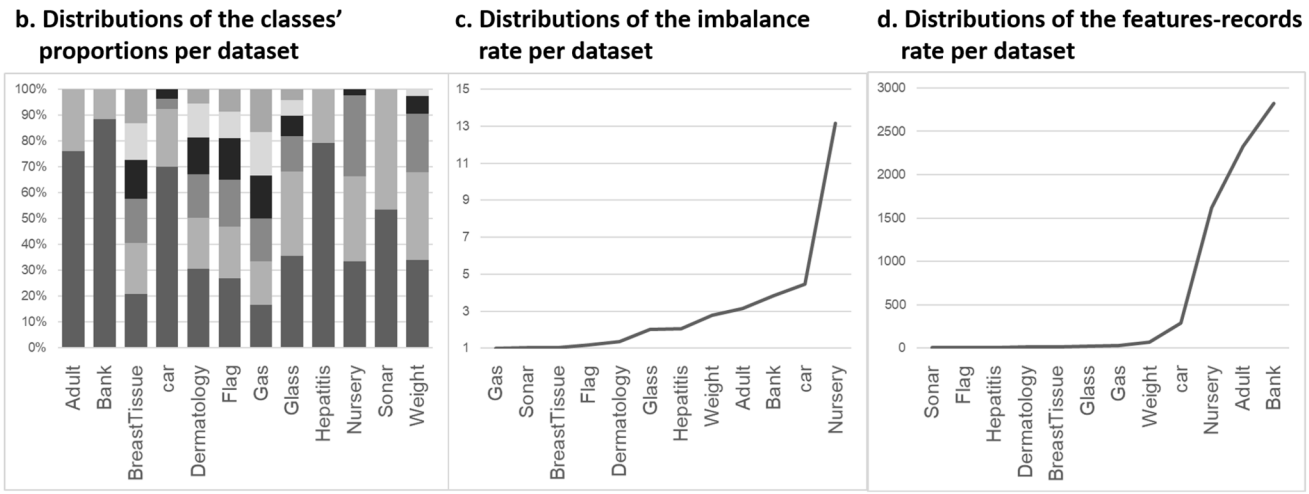


Fig. 9 Summary of datasets description

In the case of class-independent data generation, we include Over-sampling and SMOTE in our comparative study.

7.1.3 Assessment measures

We report the evaluation measures commonly used to assess the results of machine learning classification tasks. These measures usually consider that the classification is performed on a binary label (positive and negative classes) and they assume that the classes are equally important. However, in our experimental analysis, we deal with 8 out of 12 datasets with multi-class values where each class is perchance more important than the others. Accordingly, we adopt the weighted versions of the commonly used classification's evaluation measures. Mainly, these measures are based on the confusion matrix and they compare the actual label X_{ci} with the predicted label $\hat{X}_{ci}$. Thus, we denote:

- TP_i is the number of correctly predicted records such that $X_{ci} = \hat{X}_{ci}$ only for the class i.
- TN_i is the number of correctly predicted records such that $\hat{X}_{cj} = X_{cj}$ for all the classes $j \neq i$ with reference to the class i.
- FP_i the number of wrongly predicted records such that $\hat{X}_{ci} \neq X_{ci}$ only for the class i.
- FN_i the number of wrongly predicted records such that $\hat{X}_{cj} \neq X_{ci}$ for all the classes $j \neq i$ with reference to the class i.

Accordingly, the weighted Accuracy (see Eq. 6), is the ratio of correctly predicted records of the total number of records N:

$$A = \frac{1}{|X_c|}\sum_{i=1}^{|X_c|}\frac{1}{|X_{c_i}|} * \frac{\left(TP_i + TN_i\right)}{N} \tag{6}$$

And the weighted-average F-measure (see Eq. 7) is the harmonic average of the Recall (see Eq. 8) and Precision (see Eq. 9):

$$F = \frac{2 * R * P}{R + P} \tag{7}$$

such that:

$$R = \frac{1}{|X_c|}\sum_{i=1}^{|X_c|}\frac{1}{|X_{c_i}|} * \frac{TP_i}{\left(TP_i + FN_i\right)} \tag{8}$$

and

$$P = \frac{1}{|X_c|}\sum_{i=1}^{|X_c|}\frac{1}{|X_{c_i}|} * \frac{TP_i}{\left(TP_i + FP_i\right)} \tag{9}$$

Furthermore, we consider the Receiver Operator Characteristic (ROC) area and Precision Recall Curve (PRC) area, in our experimental analysis. For different classification thresholds, the former measures the tradeoff between the true positive rate and the false positive rate [15], while the latter measures the tradeoff between precision and recall [59]. ROC area considers the equal importance of all the classes and PRC area focuses on the minority classes. As we mostly consider multi-class classification, we report the macro average values of these metrics all over the possible class' values, in all our experiments.

7.2 Results and discussion

7.2.1 Values' imputation

We propose to deal with "the values imputation challenge" according to two perspectives: features' missing values and missing labels. In the former, we apply H2S-based system for data imputation and, in the latter, we apply H2S-based system for data labelling.

Data imputation To assess the ability of our H2S-based imputation method in restoring the missing values, we conduct a remove-reconstruct set of experiments over all the datasets of Fig. 9. Each experiment entails the following steps:

1. Remove a proportion of p random values (random features and random records) from the input data such that $p \in \{0.1, 0.2, ..., 0.5\}$.
2. Use the complete portion of the data to train:
   - Baseline model: a single BN learnt by Max-Min Hill Climbing (MMHC) [70].
   - H2S-based model: a merged BN learnt by H2S kernel.
3. Use the found models to differently impute the missing data.

We propose to compare the probability distribution of all the features, in the imputed versions, with their distribution in the original train dataset. For this objective, we compute the Mutual Information (MI) between the found distributions with the original features' distributions (by both H2S-based model and the baseline model) according to the following formula:

$$MI\left(X_i, X_i'\right) = \sum_{x_j \in X_i}\sum_{x_k' \in X_i'} log\frac{P\left(x_j, x_k'\right)}{P(x_j)P\left(x_k'\right)} \tag{10}$$

where $P(x_j, x_k')$ is the joint probability function of the features X_i and X_i' and $P(x_j)$ is the marginal probability of the feature X_i (idem for x_k' and X_i'). A large value of MI between two features implies the similarity between their probability distributions; contrariwise, small MI values imply the dissimilarity between (information loss among) the probability distributions.

In Fig. 10, we report the *average MI* of all the features in the imputed datasets (using the baseline system and H2S-based imputation system) with reference to the original features of the training datasets. Obviously, our contribution manages to minimize the information loss through the removal-imputation process for the considered datasets. This is explained by H2S-based method's higher average MI values, for the majority of the datasets, under different proportions of missing features' values. Nevertheless, the abundance of the records (i.e., datasets with extremely high FRR values) plays a favorable role in reconstructing the original features' probability distributions, independently from the used data imputation method. This observation explains the competitive results of both H2S-based method and the baseline method, when applied on Adults, Car, Nursery, Bank and Hepatitis datasets.

Particularly, the merit of the ensemble method (multiple structure learning by H2S kernel) for learning the merged BN structure is more pronounced for the datasets having smaller FRR values. Although in a such experimental setting the average MI dramatically decreases, H2S-based method showcases a sound ability in providing a better approximation of the original probability distribution of the features. This promising performance is more prominent when dealing with balanced datasets, i.e., the datasets having small

FRR values and IR ≈ 1 (see Gas, Dermatology, Sonar and Weight datasets' corresponding graphs in Fig. 10).

To examine the impact of the information loss on the predictive ability of the imputed data, we propose to feed the imputed versions of the input datasets to train MLP, SVM and DT classifiers. We report in Fig. 11, the average PRC area scores for different proportions of missing values computed over the whole collection of datasets. We recall that we consider the PRC area scoring because we deal with datasets having various IR values; hence, we take into consideration the multi-classification, the imbalance and the different class values' importance. We observe that on overall, our contribution significantly outperforms the baseline system by maximizing the predictive performance of 10 out of 12 datasets. This is more distinguished in the cases where the proportion of the missing values gets its higher values (see Bank, Car, Flag, Gas, Glass, Sonar, Nursery graphs in Fig. 11). On overall, H2S-based method of missing values imputation, not only conserves the probability distribution of the features, but also plays an important role in optimizing the predictive performance of the resultant imputed dataset.

Data labelling For the second perspective of our analysis, we propose to reproduce the case of the semi-supervised learning. To carry out our experimental study, we implement the following steps:

1. Split the records of each training dataset into two portions as unlabeled-labeled%: 10–90%, 20–80%, ..., 90–10%.
2. Remove the labels from the first portion of the data sets' split version.
3. Apply CEM, YATSI and H2S-based labeling system for estimating the removed class' value.
4. Use the originally labeled and the newly labeled versions of the data to train MLP, SVM and DT classifiers.
5. Use the same test dataset for evaluation.

To obtain reliable labeled portions of the training datasets (e.g., the 10% of the labeled records are sufficient for observing all the features' values), we propose to include only the datasets that have high FRR values namely Bank, Adult, Nursery, Car and Weight. With this assumption, we make sure that all the features' values are included in the evaluation process with reference to the IR values of the selected datasets. We report in Fig. 12 the obtained average accuracies for each labeling method and each dataset. We distinguish between the multi-class classification and binary classification (see Fig. 12a, b respectively). On the one hand, our contribution and CEM perform competitively for the datasets having higher FRR values and binary labels (i.e., Nursery, Adult and Bank. On the other hand, H2S-based system provides significantly labeled data that allows the optimization of the prediction performance; this is valid for multi-classification cases. Additionally, H2S maintains reliable performance for extremely small datasets including the datasets having small FRR, generally, and from 60 to 90% unlabeled portions of the considered benchmark datasets.

The results within Fig. 12 soundly prove the advantage of the probabilistic methods (CEM and H2S) handling the challenge of learning from small data. They aim at skewing the labeling model to the distribution of the input pseudo-labeled data. Moreover, they ensure a better between-classes discriminative power; hence, the achieved better results of multi-classification cases. Fundamentally, the ensemble methods provide an effective remedy to over-fitting; yet the ensemble strategy is the key to the optimized performance. Actually, the principle behind CEM and YATSI follows a vertical strategy, which initializes the labels and iteratively updates the confidence on the labels in a way to enhance a certain criterion (e.g., likelihood and accuracy). The main drawbacks of this strategy are the dependence to the labels' initialization (i.e., the algorithm progressively tries to improve the bias) and to the error propagation. Contrariwise, H2S-based labeling system follows a horizontal strategy as it immediately and indefinitely explores all the candidate structures. The BN structures' merging is benefic for escaping the overfitting and maintaining a faithful modelling of the input dataset.

7.2.2 Record's generation

We assess our contribution for tackling the challenge of missing records according to two perspectives: data balancing (i.e., data generation to compensate the imbalance in data) and data generation (i.e., data augmentation independently from the classification context).

Data balancing Pre-processing the data for balancing the classes is an extremely important step for training structural classifiers [12]. The empirical related studies [3, 5, 25] have proved that SVM, MLP and DT are extremely sensitive to imbalanced data. Therefore, we propose to apply the most known balancing techniques (see Sect. 2), along with H2S-based balancing system in order to analyze their impact on classification improvement. For this set of experiments, we consider the datasets having **high IR values**; thus, we use Dermatology, Glass, Hepatitis, Weight, Adult, Bank, Car and Nursery.

We report, in Fig. 13, the average Recall, Precision, F-measure, ROC area and PRC area obtained by the considered classifiers using all the compared balancing techniques. We consider that a successful balancing method is the one that yields a maximized value of each scoring function. The examination of the *Radar Charts*, in Fig. 13, demonstrates a competitive performance of all the considered methods for datasets with **higher FRR values** (i.e., Bank, Nursery and Weight). As for the remaining datasets, each method

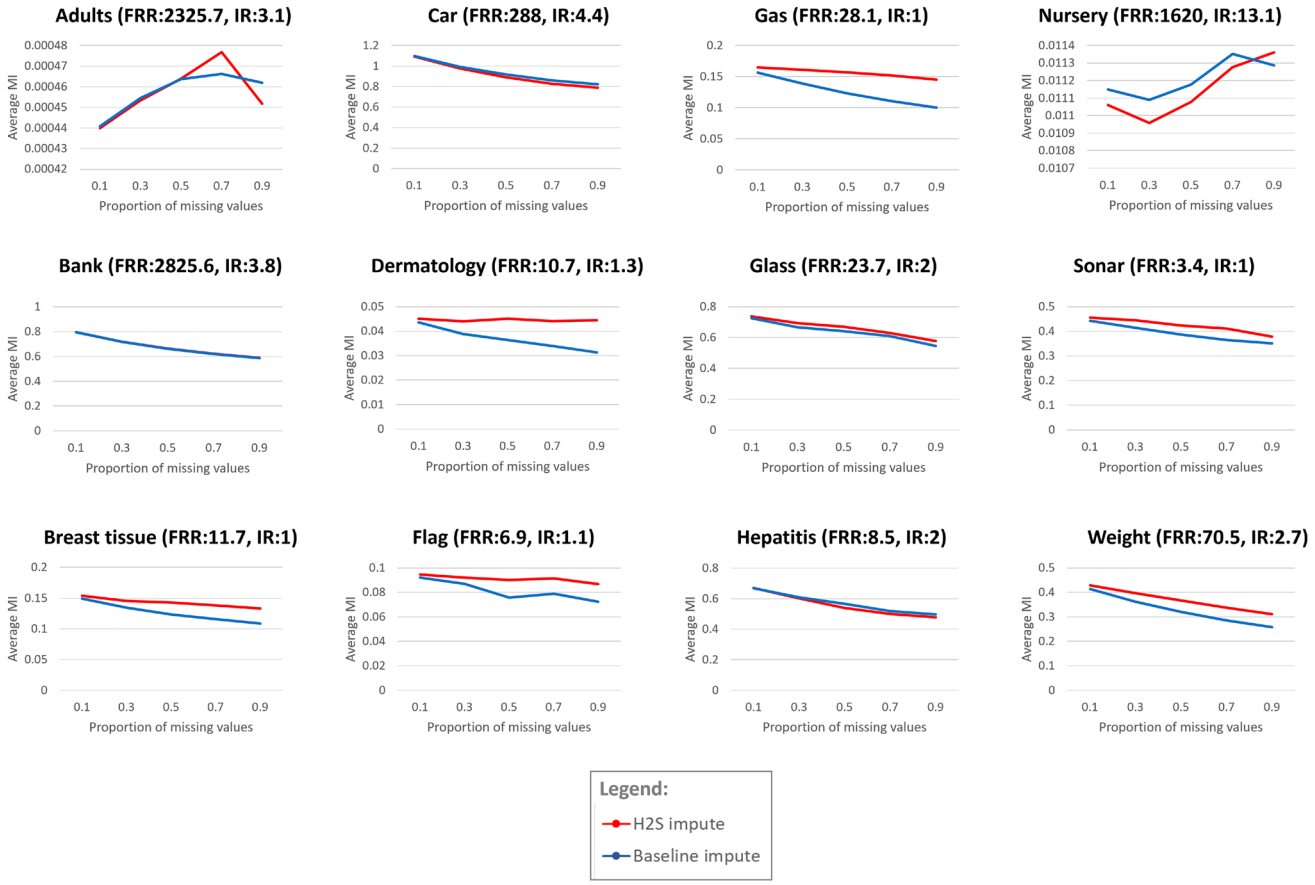


Fig. 10 Average mutual information between the original features' distributions and the imputed features' distributions as a function of the proportion of missing values in the training dataset

outstands in particular aspects. For example, the undersampling maximizes the Recall for Adult and Bank datasets while SMOTE maximizes the precision of Dermatology and Glass datasets. Meanwhile, our contribution ensures an optimized compromise between all the metrics, on the one hand, and an improved performance on the majority of the datasets, on the other hand. Notably, H2S-based balancing system manages to significantly optimize the PRC area. Such result is promising given the fact that this measure is considered as the most faithful assessment of the imbalanced data's classification [59].

Following our synthesis over the performance of the structural classifiers, we propose to dive into a more precise evaluation of the balancing methods' impact on each classifier (among SVM, MLP and DT). We report, in Fig. 14, the PRC areas of the balanced versions of all the considered datasets using all the balancing algorithms. We observe that all the balancing systems are competitive on the datasets having a high FRR (i.e., Nursery, Bank, Adult and Weight datasets). As for the extreme challenging dataset (i.e., Glass and Hepatitis datasets are having low FRR and high IR values), H2S method constitutes the most suitable balancing strategy for enhancing the predictive performance of SVM, MLP and DT. To summarize, our contribution maximized PRCs of 7 out of 8 datasets, for MLP classifier, 5 out of 8 datasets, for SVM, and 6 out of 8 datasets, for DT. Therefore, we conclude that our balancing system presents a reliable pre-processing step for MLP classifier, which is extremely greedy in terms of records.

Data generation To showcase the merit of data generation, we propose to augment the FRR of the considered benchmark datasets, which initially have **small FRR values**. Thus, we consider Sonar, Flag, Hepatitis, Dermatology, Breast Tissue and Glass datasets whose corresponding FRR values vary from 3.4 to 23.7. Our experimental protocol involves using Over-sampling, SMOTE and H2S-based system for records generation in order to achieve FRR values of 50, 100, 500 and 1000. We explore the impact of the different versions of these datasets, as compared to the original data, for training and evaluating MLP, SVM and DT classifiers. We report, in Fig. 15 the evolution of the weighted average F-measure values obtained by the considered generation methods per classifier for each dataset. We include, in each plot, the F-measure

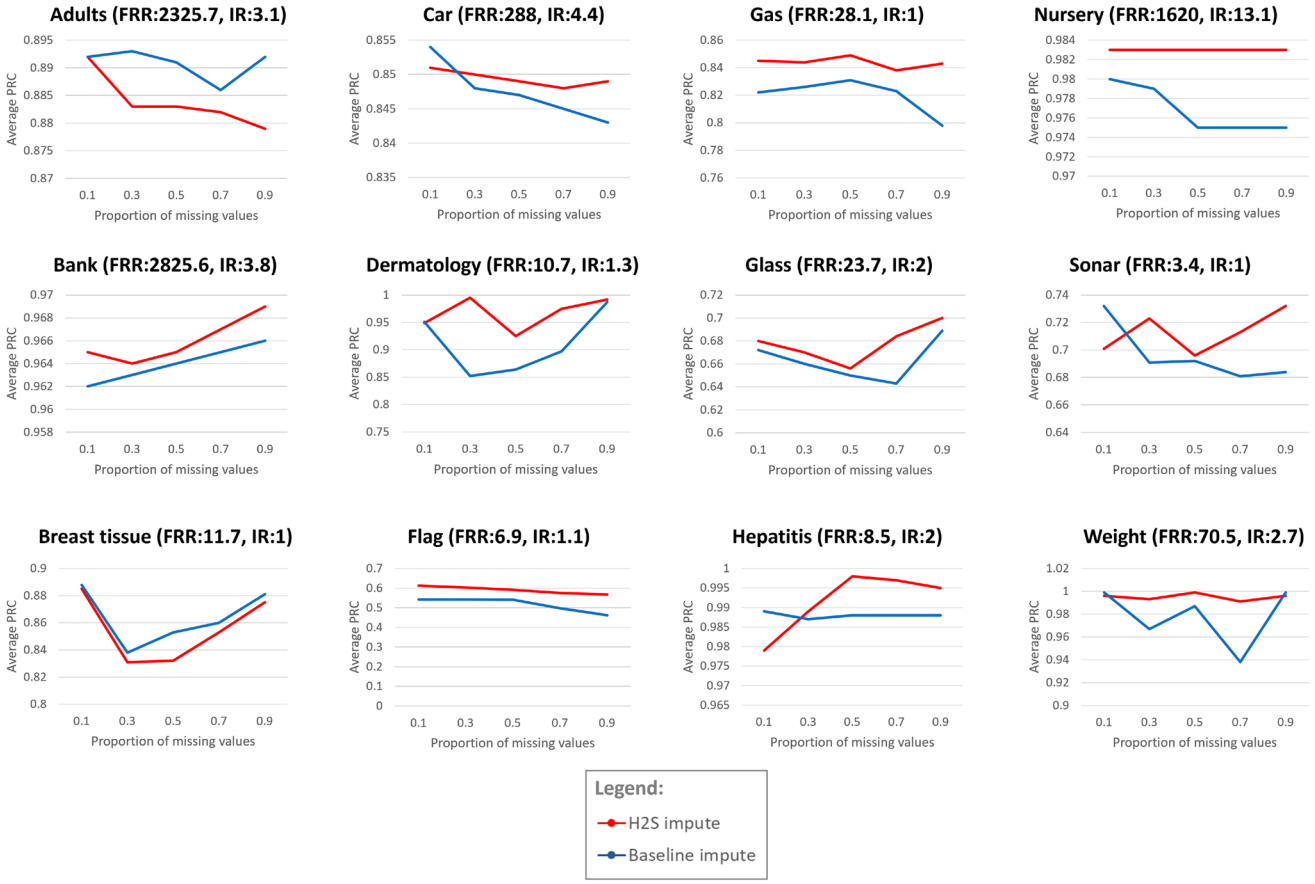


Fig. 11 Precision-Recall Curve (PRC) of baseline and H2S-based imputation methods as a function of proportion (p) of missing values per datasets

score obtained without applying any type of data generation (i.e., the point of the coordinates (1, .)). The obtained results, on all the datasets and all the classifiers, provide the empirical evidence of the extremely promising performance of H2S-based method for data generation. On overall, our contribution ensures that the generated data gradually improves the classification, as compared to the original data. It plays a crucial role in enhancing the classifiers performance with 5 out of 6 datasets for all the classifiers. Contrariwise, SMOTE showed a deceiving performance in 5 out of 6 datasets. This is explained by the fact that it uses the distance between the records for simulating the records; hence, it magnifies the overlap between the classes and it perturbates the discriminative performance of the classifiers. Being paired with MLP, SMOTE usually deteriorate the classification on all the datasets, except for Glass dataset. This is explained by the over-fitting impact. Independently from the balancing context, Over-sampling shows an enhanced performance as compared to SMOTE. Following all the obtained results, we conclude that the H2S-based generation system made the most from the ensemble learning and BN fusion in order to optimize the classification task while escaping the over-fitting.

8 Conclusions and future work

We present in this paper a new probabilistic approach to efficiently learn from small data. Between over-fitting and over-generalization, we aimed our vision to the ensemble learning combined with data-driven models' fusion. This is concretized by our new H2S kernel for learning generalized BN. To showcase the polyvalence of our kernel, we proposed to extend it and to enrich it in order to tackle different types of data missingness. Therefore, we suggest a taxonomy that categorizes missing values' cases, including features' values and classes' labels, and missing record's cases, including the lack of records and the imbalanced datasets. Following these applications contexts, we propose four H2S-based systems for data imputation, data labeling, data generation and data balancing. To emphasize on the impact of these probabilistic models on enhancing the usability of datasets, which are affected by

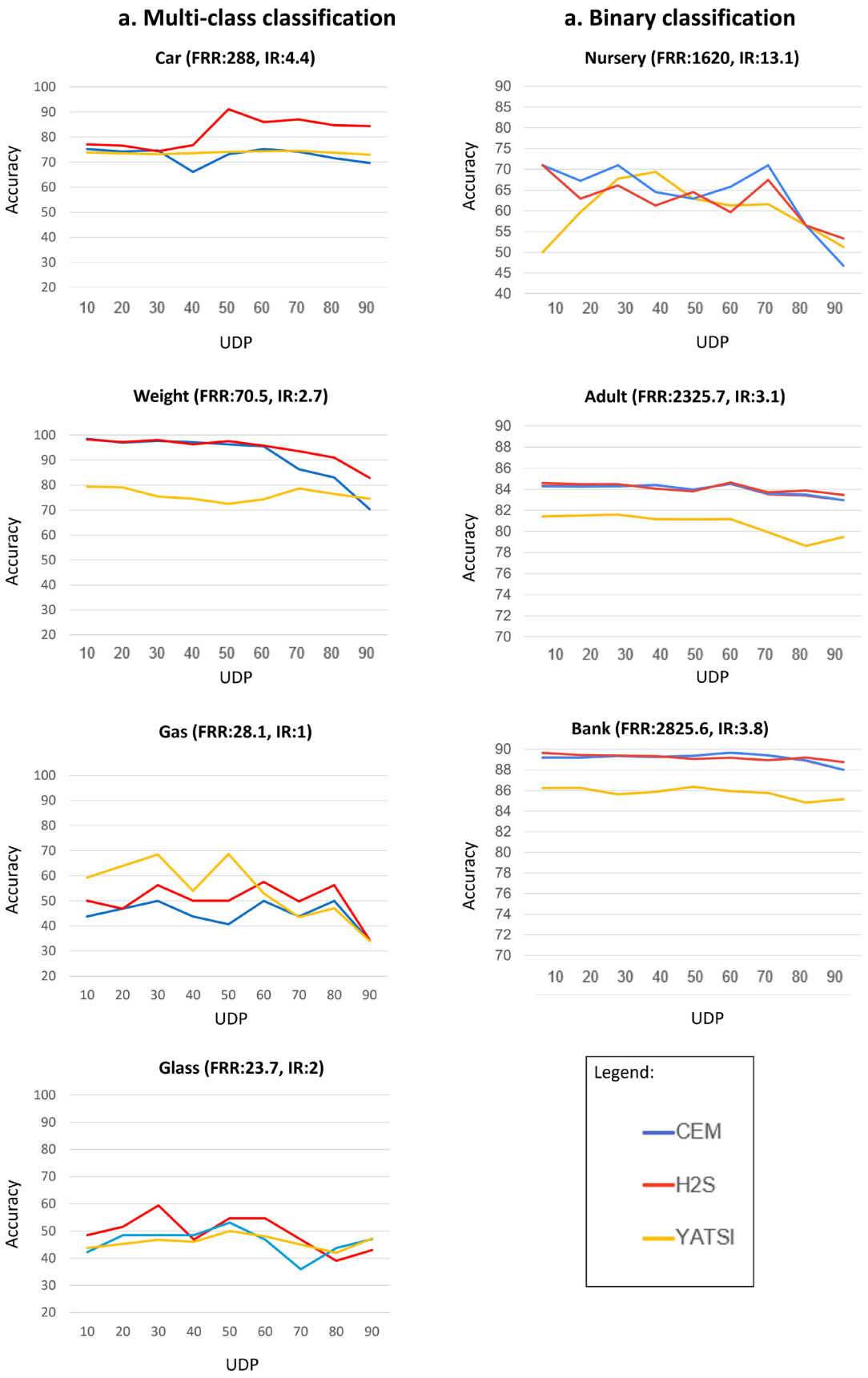


Fig. 12 Accuracy as a function of unlabeled data percentage

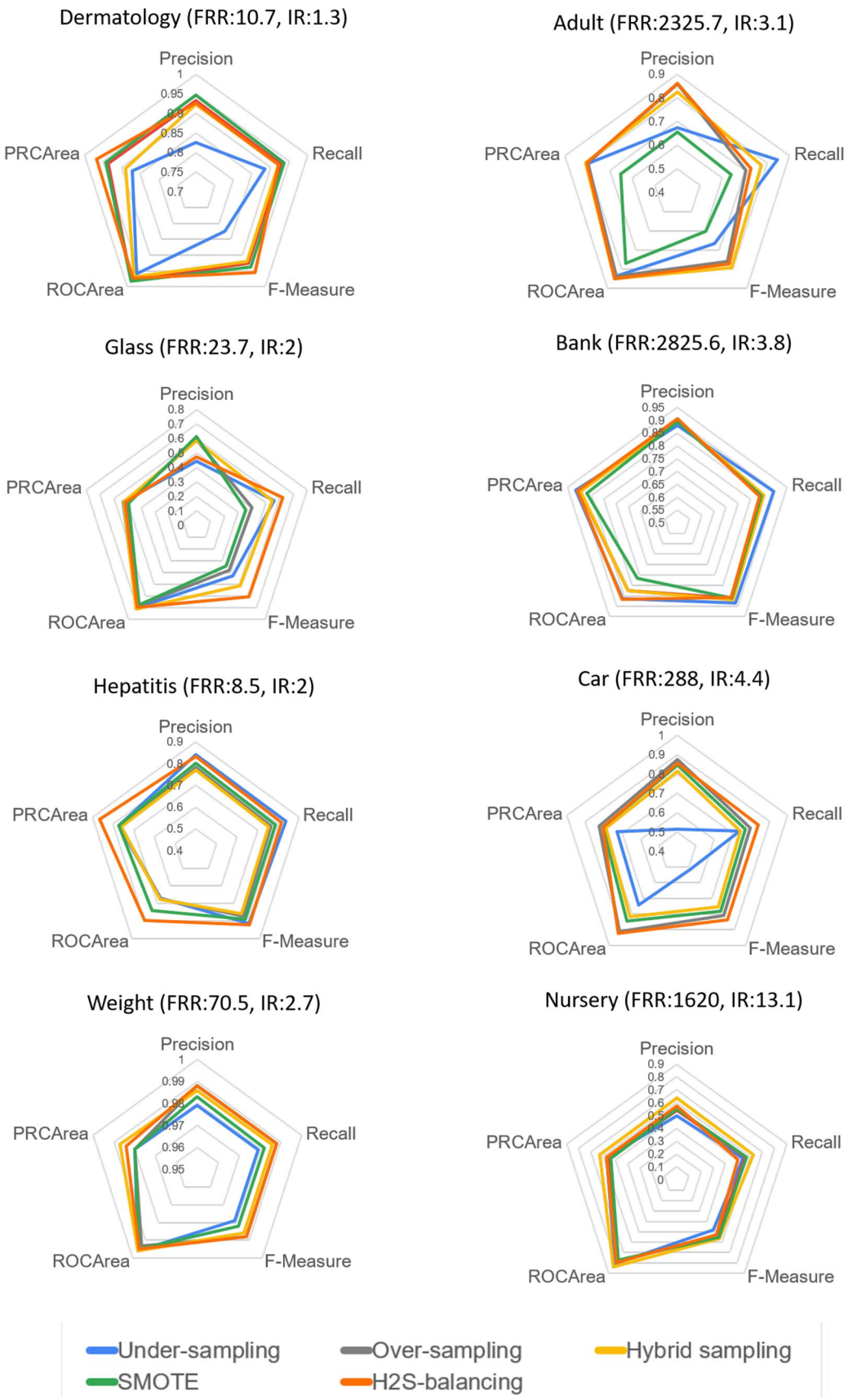


Fig. 13 Average Recall, Precision, F-measure, ROC area and PRC area obtained by the considered balancing methods per dataset

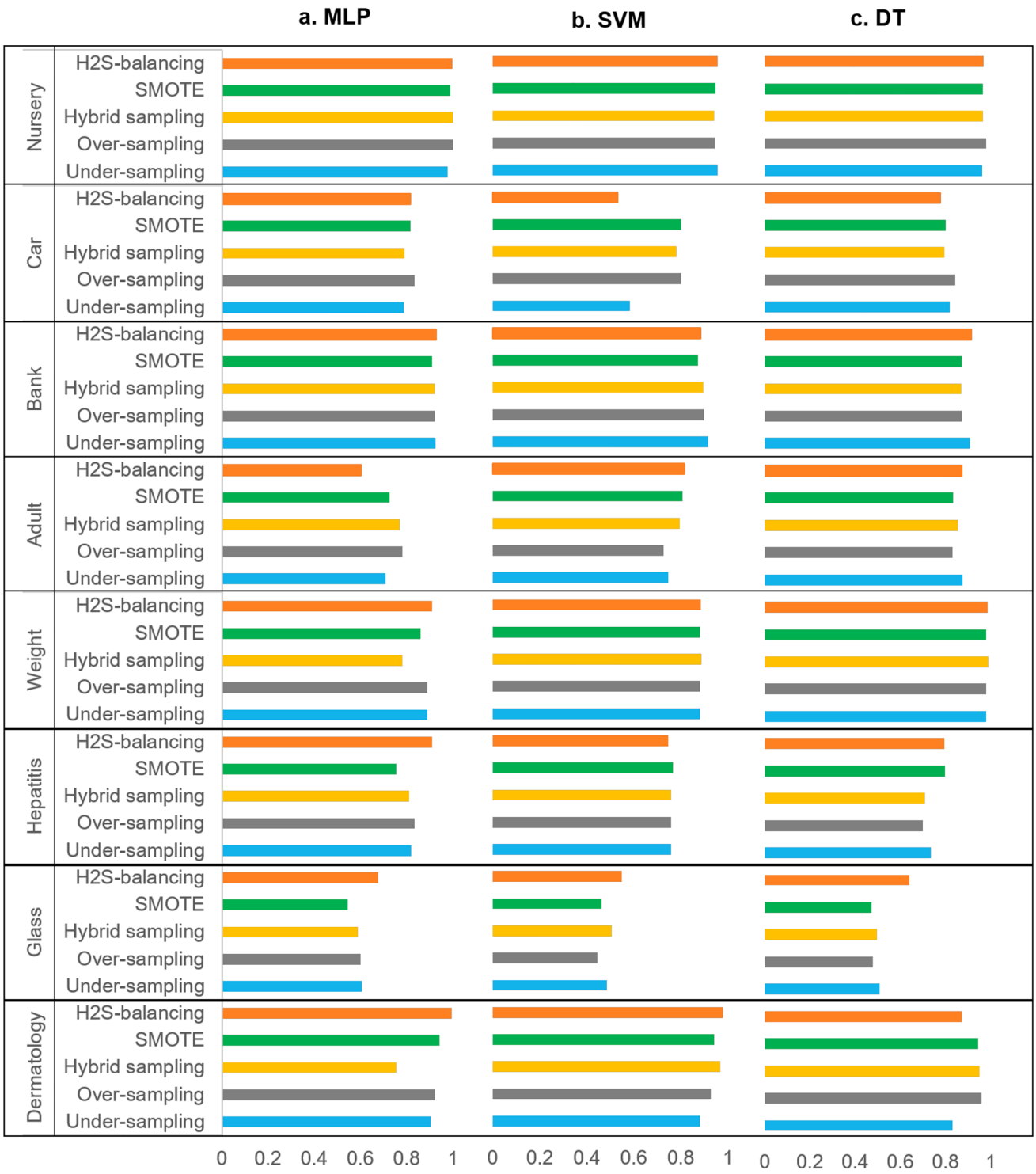


Fig. 14 PRC values obtained by the balancing methods per dataset per classifier

missingness, we carried out an exhaustive experimental study over diversified benchmark datasets. We made sure to include the most known related algorithms for comparison and to quantify the results using the most accurate evaluation metrics. Accordingly, our experimental analysis validated the merit of the probabilistic methods in the four applications. Particularly, it shed the light on the efficiency of H2S-based systems for providing reliable models learnt from the smallest datasets. At various experiments among our analysis, we pointed out the robustness

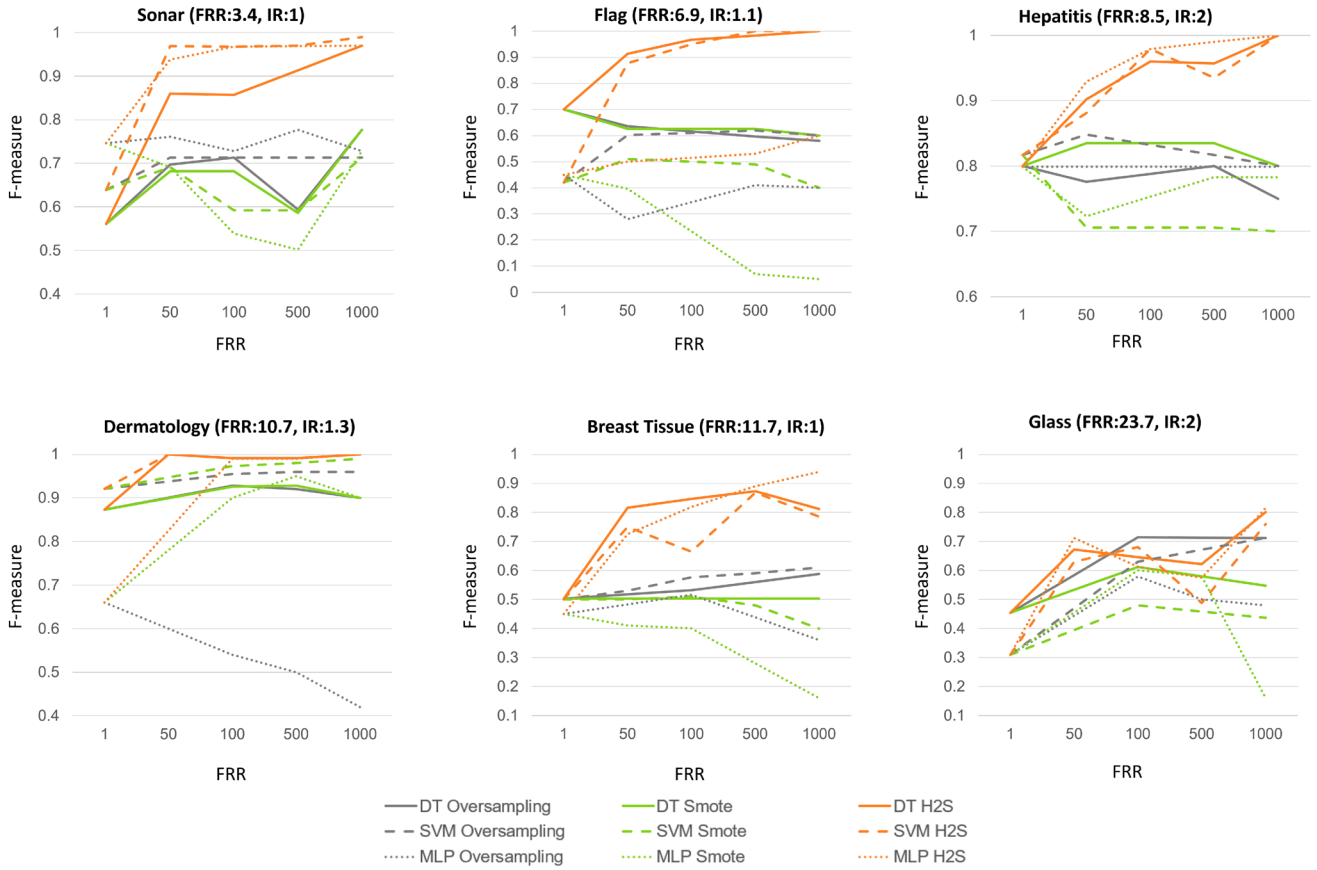


Fig. 15 Evolution of F-measure obtained by the considered generation methods per classifier for each dataset

of our contributions towards multi-classification tasks. The promising results of our H2S-based systems motivated us to propound a generic meta-model that allows the orchestration of our proposed modules. This meta-model potentially plays a crucial role in various applications, essentially, the medical decision support systems and social network analysis systems. Our future directions are oriented towards two major perspectives naming the consideration of multi-view data, which is important for profile matching, and the inclusion of the temporal aspect, which migrates the model to the dynamic BN.

References

1. Akeret J, Refregier A, Amara A, Seehars S, Hasner C (2015) Approximate Bayesian computation for forward modeling in cosmology. J Cosmol Astropart Phys 2015(08):043
2. Ben-David S, Lu T, Pál D, Sotáková M (2009) Learning low density separators. In: van Dyk D, Welling M (eds) Proceedings of the twelth international conference on artificial intelligence and statistics. Proceedings of Machine Learning Research, PMLR, Florida, USA, pp 25–32
3. Boonchuay K, Sinapiromsaran K, Lursinsap C (2017) Decision tree induction based on minority entropy for the class imbalance problem. Pattern Anal Appl 20(3):769–782
4. 2 Carvalho AM (2009) Scoring functions for learning Bayesian networks. Inesc-id Tec. Rep 1
5. Castro CL, Braga AP (2013) Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data. IEEE Trans Neural Netw Learn Syst 24(6):888–899
6. Chen T, Guestrin C (2016) Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd ACM sigkdd International Conference on knowledge discovery and data mining. Association for Computing Machinery, USA, pp 785–794. https://doi.org/10.1145/2939672.2939785
7. Chen Z, Lin T, Xia X, Xu H, Ding S (2018) A synthetic neighborhood generation based ensemble learning for the imbalanced data classification. Appl Intell 48(8):2441–2457
8. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P (2011) Natural language processing (almost) from scratch. J Mech Learn Res 12(ARTICLE):2493–2537
9. Cooper GF (1990) The computational complexity of probabilistic inference using Bayesian belief networks. Artif Intell 42(2–3):393–405
10. Cooper GF, Herskovits E (1992) A Bayesian method for the induction of probabilistic networks from data. Mach Learn 9(4):309–347
11. Correia AHC, Cussens J, de Campos C (2020) On pruning for score-based Bayesian network structure learning. In: The 23rd international conference on artificial intelligence and statistics

{AISTATS}, Proceedings of Machine Learning Research, vol 108. PMLR, pp 2709–2718
12. Domingues I, Amorim JP, Abreu PH, Duarte H, Santos J (2018) Evaluation of oversampling data balancing techniques in the context of ordinal classification. In: 2018 International Joint Conference on neural networks (IJCNN). IEEE, Brazil, pp 1–8. https://doi.org/10.1109/IJCNN.2018.8489599
13. Dópido I, Li J, Marpu PR, Plaza A, Dias JMB, Benediktsson JA (2013) Semisupervised self-learning for hyperspectral image classification. IEEE Trans Geosci Remote Sens 51(7):4032–4044
14. Džeroski S, Panov P, Ženko B (2009) Ensemble methods in machine learning. In: Encyclopedia of Complexity and Systems Science. Springer, New York, NY, pp 5317–5325. NY. https://doi.org/10.1007/978-0-387-30440-3_315
15. Fawcett T (2004) Roc graphs: notes and practical considerations for researchers. Mach Learn 31(1):1–38
16. Feng W, Huang W, Ren J (2018) Class imbalance ensemble learning based on the margin theory. Appl Sci 8(5):815
17. Fernández A, Garcia S, Herrera F, Chawla NV (2018) Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. J Artif Intell Res 61:863–905
18. François O, Leray P (2006) Learning the tree augmented Naive Bayes classifier from incomplete datasets. In: Third European workshop on probabilistic graphical models, 12–15 September, Prague, Czech Republic. Electronic Proceedings, pp 91–98
19. Friedman N, Geiger D, Goldszmidt M (1997) Bayesian network classifiers. Mach Learn 29(2):131–163
20. Gámez JA, Mateo JL, Puerta JM (2011) Learning Bayesian networks by hill climbing: efficient methods based on progressive restriction of the neighborhood. Data Min Knowl Disc 22(1):106–148
21. Han H, Wang WY, Mao BH (2005) Borderline-smote: a new over-sampling method in imbalanced data sets learning. In: Huang DS, Zhang XP, Huang GB (eds) Advances in Intelligent Computing. International Conference on intelligent computing (ICIC). Springer, Berlin, Heidelberg, pp 878–887. https://doi.org/10.1007/11538059_91
22. Heckerman D, Geiger D, Chickering DM (1995) Learning Bayesian networks: the combination of knowledge and statistical data. Mach Learn 20(3):197–243
23. Huang Y, Gao Y, Gan Y, Ye M (2021) A new financial data forecasting model using genetic algorithm and long short-term memory network. Neurocomputing 425:207–218
24. Imam N, Issac B, Jacob SM (2019) A semi-supervised learning approach for tackling twitter spam drift. Int J Comput Intell Appl 18(02):1950010
25. Imam T, Ting KM, Kamruzzaman J (2006) z-SVM: An SVM for improved classification of imbalanced data. In: Sattar A, Kang, Bh (eds) Advances in artificial intelligence, 19th Australian joint conference on artificial intelligence, Hobart, Australia. Springer, Berlin, Heidelberg, pp 264–273. https://doi.org/10.1007/11941439_30
26. Janžura M, Nielsen J (2006) A simulated annealing-based method for learning Bayesian networks from statistical data. Int J Intell Syst 21(3):335–348
27. Kang H (2013) The prevention and handling of the missing data. Korean J Anesthesiol 64(5):402
28. Kim J, Tae D, Seok J (2020) A survey of missing data imputation using generative adversarial networks. In: 2020 International Conference on artificial intelligence in information and communication (ICAIIC). IEEE, Fukuoka, Japan, pp 454–456. https://doi.org/10.1109/ICAIIC48513.2020.9065044
29. Kraaijeveld P, Druzdzel MJ, Onisko A, Wasyluk H (2005) Genierate: an interactive generator of diagnostic Bayesian network models. In: Proc. 16th Int. Workshop Principles Diagnosis. Citeseer, pp 175–180
30. Kramer SC, Sorenson HW (1988) Bayesian parameter estimation. IEEE Trans Autom Control 33(2):217–222
31. Lateh MA, Muda AK, Yusof ZIM, Muda NA, Azmi MS (2017) Handling a small dataset problem in prediction model by employ artificial data generation approach: a review. J Phys Conf Ser 892:012016
32. Li H, Jin G, Zhou J, Zb ZHOU, Dq LI (2008) Survey of Bayesian network inference algorithms. Syst Eng Eclectron 30(5):935–939
33. Little RJ, Rubin DB (2019) Statistical analysis with missing data, vol 793. Wiley, Hoboken
34. Liu F, Tian F, Zhu Q (2007) Bayesian network structure ensemble learning. In: Alhajj R, Gao H, Li J, Li X, Zaïane OR (eds) Advanced Data Mining and Applications. ADMA 2007 Springer, Berlin, Heidelberg, pp 454–465. https://doi.org/10.1007/978-3-540-73871-8_42
35. Liu H, Wang J (2006) A new way to enumerate cycles in graph. In: Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/ICIW). IEEE, Guadeloupe, French Caribbean, p 57. https://doi.org/10.1109/AICT-ICIW.2006.22
36. Longadge R, Dongre S (2013) Class imbalance problem in data mining review. arXiv preprint arXiv:1305.1707
37. Mack C, Su Z, Westreich D (2018) Managing missing data in patient registries: addendum to registries for evaluating patient outcomes: A user's Gguide, Third Edition [Internet]. Agency for healthcare research and quality (US), Rockville (MD), Report No.: 17(18)-EHC015-EF
38. Mallapragada PK, Jin R, Jain AK, Liu Y (2008) Semiboost: boosting for semi-supervised learning. IEEE Trans Pattern Anal Mach Intell 31(11):2000–2014
39. Marlin B (2008) Missing data problems in machine learning. Ph.D. thesis
40. Marqués AI, García V, Sánchez JS (2012) Exploring the behaviour of base classifiers in credit scoring ensembles. Expert Syst Appl 39(11):10244–10250
41. Martins MS, El Yafrani M, Delgado M, Lüders R, Santana R, Siqueira HV, Akcay HG, Ahiod B (2021) Analysis of Bayesian network learning techniques for a hybrid multi-objective Bayesian estimation of distribution algorithm: a case study on mnk landscape. J Heuristics 27(4):549–573
42. Mirza M, Osindero S (2014) Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784
43. Neapolitan RE, Jiang X (2010) Probabilistic methods for financial and marketing informatics. Elsevier, Amsterdam
44. Njah H, Jamoussi S (2015) Weighted ensemble learning of Bayesian network for gene regulatory networks. Neurocomputing 150:404–416
45. Paton K (1969) An algorithm for finding a fundamental set of cycles of a graph. Commun ACM 12(9):514–518
46. Pearl J (1988) Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann Publishers Inc., San Francisco
47. Pearl J (2014) Probabilistic reasoning in intelligent systems: networks of plausible inference. Elsevier, Amsterdam
48. Pellet JP, Elisseeff A (2008) Using Markov blankets for causal structure learning. J Mach Learn Res 9(7):1295–1342. https://doi.org/10.5555/1390681.1442776
49. Pérez-Miñana E (2016) Improving ecosystem services modelling: Insights from a Bayesian network tools review. Environ Model Softw 85:184–201
50. Qi GJ, Luo J (2020) Small data challenges in big data era: a survey of recent progress on unsupervised and semi-supervised methods. IEEE Trans Pattern Anal Mach Intell 44(4):2168–2187. https://doi.org/10.1109/TPAMI.2020.3031898

51. Quinlan JR (1986) Induction of decision trees. Mach Learn 1(1):81–106
52. Ramanan N, Natarajan S (2020) Causal learning from predictive modeling for observational data. Front Big Data 3:34
53. Rancoita PM, Zaffalon M, Zucca E, Bertoni F, De Campos CP (2016) Bayesian network data imputation with application to survival tree analysis. Comput Stat Data Anal 93:373–387
54. Redner RA, Walker HF (1984) Mixture densities, maximum likelihood and the EM algorithm. SIAM Rev 26(2):195–239
55. Rekha G, Reddy VK, Tyagi AK, Nair MM (2020) Distance-based bootstrap sampling in bagging for imbalanced data-set. In: 2020 International Conference on emerging trends in information technology and engineering (ic-ETITE). IEEE, Vellore, India, pp 1–6. https://doi.org/10.1109/ic-ETITE47903.2020.345
56. Rissanen J (1999) Hypothesis selection and testing by the mdl principle. Comput J 42(4):260–269
57. Rosenberg C, Hebert M, Schneiderman H (2005) Semi-supervised self-training of object detection models. In: 2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05), pp 29–36. https://doi.org/10.1109/ACVMOT.2005.107
58. Sagi O, Rokach L (2018) Ensemble learning: a survey. Wiley Interdiscipl Rev Data Min Knowl Discov 8(4):e1249
59. Saito T, Rehmsmeier M (2015) The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. PLoS One 10(3):e0118432
60. Sakamoto Y, Ishiguro M (1986) Akaike information criterion statistics, vol 81. D. Reidel, Dordrecht, p 26853 (**10.5555**)
61. Schwarz G et al (1978) Estimating the dimension of a model. Ann Stat 6(2):461–464
62. Scutari M (2009) Learning Bayesian networks with the bnlearn r package. arXiv preprint arXiv:0908.3817
63. Scutari M, Lebre S (2013) Bayesian network constraint-based structure learning algorithms: parallel and optimised implementations in the bnlearn R Package. http://arxiv.org/abs/1406.7648
64. Spirtes P, Glymour CN, Scheines R, Heckerman D (2000) Causation, prediction, and search. MIT Press, Cambridge
65. Tang Y, Wang Y, Cooper KM, Li L (2014) Towards big data Bayesian network learning-an ensemble learning based approach. In: 2014 IEEE International Congress on big data. IEEE, Anchorage, AK, USA, pp 355–357. https://doi.org/10.1109/BigData.Congress.2014.58
66. Tanha J, van Someren M, Afsarmanesh H (2017) Semi-supervised self-training for decision tree classifiers. Int J Mach Learn Cybern 8(1):355–370
67. Taud H, Mas JF (2018) Multilayer perceptron (mlp). In: Camacho Olmedo MT, Paegelow M, Mas JF, Escobar F (eds) Geomatic approaches for modeling land change scenarios. Springer, Cham, pp 451–455. https://doi.org/10.1007/978-3-319-60801-3_27
68. Thomassen C (1985) Even cycles in directed graphs. Eur J Comb 6(1):85–89
69. Tong Y, Tien I (2017) Algorithms for Bayesian network modeling, inference, and reliability assessment for multistate flow networks. J Comput Civ Eng 31(5):04017051
70. Tsamardinos I, Brown LE, Aliferis CF (2006) The max-min hill-climbing Bayesian network structure learning algorithm. Mach Learn 65(1):31–78
71. Van Engelen JE, Hoos HH (2020) A survey on semi-supervised learning. Mach Learn 109(2):373–440
72. Vapnik V, Guyon I, Hastie T (1995) Support vector machines. Mach Learn 20(3):273–297
73. Vilardell M, Buxó M, Clèries R, Martínez JM, Garcia G, Ameijide A, Font R, Civit S, Marcos-Gragera R, Vilardell ML et al (2020) Missing data imputation and synthetic data simulation through modeling graphical probabilistic dependencies between variables (modgraprodep): an application to breast cancer survival. Artif Intell Med 107:101875
74. Xu L, Schuurmans D (2005) Unsupervised and semi-supervised multi-class support vector machines. In: AAAI, vol. 40, p. 50
75. Yap BW, Abd Rani K, Abd Rahman HA, Fong S, Khairudin Z, Abdullah NN (2014) An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets. In: Proceedings of the first international conference on advanced data and information engineering (DaEng-2013). Springer, pp 13–22
76. Yarowsky D (1995) Unsupervised word sense disambiguation rivaling supervised methods. In: 33rd Annual Meeting of the Association for computational linguistics, MIT, Cambridge, Massachusetts, USA, pp 189–196. https://doi.org/10.3115/981658.981684
77. Yaslan Y, Cataltepe Z (2010) Co-training with relevant random subspaces. Neurocomputing 73(10–12):1652–1661
78. Yoon J, Jordon J, Schaar M (2018) Gain: Missing data imputation using generative adversarial nets. In: Proceedings ofthe 35th International Conference on Machine Learning (ICML). PMLR, Stockholm, Sweden, pp 5675–5684
79. Yu J, Smith VA, Wang PP, Hartemink AJ, Jarvis ED (2002) Using Bayesian network inference algorithms to recover molecular genetic regulatory networks. In: International Conference on systems biology, vol 2002
80. Yu S, Krishnapuram B, Rosales R, Rao RB (2011) Bayesian co-training. J Mach Learn Res 12:2649–2680
81. Zheng W, Jin M (2020) The effects of class imbalance and training data size on classifier learning: an empirical study. SN Comput Sci 1(2):1–13
82. Zhou D, Bousquet O, Lal TN, Weston J, Schölkopf B (2004) Learning with local and global consistency. In: Thrun S, Saul L, Schölkopf B (eds) Advances in neural information processing systems, vol 16. MIT Press
83. Zhu X, Lafferty J (2005) Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In: Proceedings of the 22nd International Conference on machine learning, pp 1052–1059. https://doi.org/10.1145/1102351.1102484