MISFEAT: Feature Selection for Subgroups with Systematic Missing Data

Bar Genossar^{+}, Thinh On*, Md. Mouinul Islam*, Ben Eliav^{+}, Senjuti Basu Roy*, Avigdor Gal^{+}
Technion – Israel Institute of Technology^{+}, *New Jersey Institute of Technology**
sbargen@campus.technion.ac.il,to58@njit.edu,mi257@njit.edu,
ben.eliav@campus.technion.ac.il,senjutib@njit.edu,avigal@technion.ac.il

***Abstract*—We investigate the problem of selecting features for datasets that can be naturally partitioned into *subgroups* (*e.g.*, according to socio-demographic groups and age), each with its own dominant set of features. Within this subgroup-oriented framework, we address the challenge of *systematic missing data*, a scenario in which some feature values are missing for all tuples of a subgroup, due to flawed data integration, regulatory constraints, or privacy concerns. Feature selection is governed by finding *mutual Information*, a popular quantification of correlation, between features and a target variable. Our goal is to identify top-K feature subsets of some fixed size with the highest joint mutual information with a target variable. In the presence of systematic missing data, the closed form of mutual information could not simply be applied. We argue that in such a setting, leveraging relationships between available feature mutual information within a subgroup or across subgroups can assist inferring missing mutual information values. We propose a generalizable model based on *heterogeneous graph neural network* to identify interdependencies between feature-subgroup-target variable connections by modeling it as a multiplex graph, and employing information propagation between its nodes. We address two distinct scalability challenges related to training and propose principled solutions to tackle them. Through an extensive empirical evaluation, we demonstrate the efficacy of the proposed solutions both qualitatively and running time wise.**

I. Introduction and Motivation

Features, measurable properties of a phenomenon, are useful in data science for data exploration and model building. Feature selection, the process of choosing informative and discriminating features from a large set, is a key step in data science pipelines, focusing on retaining features that provide the greatest benefit for learning [1]–[3]. Mutual information (MI) is a model-agnostic measure that quantifies the dependency between two variables and has been widely used in feature selection [4]–[9]. Intuitively, features with higher MI values relative to the target variable are often more important.

In this work, we focus on feature selection using MI that is challenged by the presence of (1) distinct subgroups and (2) systematic missing data. A *subgroup* constitutes a group of records demonstrating identical characteristics pertinent to an application at hand. Subgroup analysis is common in user group analytics [10] and we illustrate the notion of subgroups next using medical cohort analytics [11], [12].

Example 1. *In medical cohort analysis, given a subgroup (cohort), experts seek answers to three typical questions [13], namely predicting some future status of patients in a cohort, interpreting a phenomenon using a cohort, and seeking similar cohorts. To effectively pursue these tasks, models are empowered by the most informative (for within subgroup analysis) and discriminating (for between subgroup analysis) features.*

Table I represents an instance of patient health records. Consider a task of predicting readmission (target variable) to a hospital within 30 days after initial patient discharge. Cohort analysis over this data excerpt may be based on a combination of age range and ethnic group. A possible such characterization (marked with different colors in the table) entails four subgroups: Asian aged 40 and below, Caucasian aged 40 and below, Asian above 40, and Caucasian above 40. In the context of feature selection, the goal is to identify for each of the subgroups predictors that exhibit high informativeness of being readmitted. Focusing on the Family History *feature, whenever its value is positive for the subgroups of 40 and below, the corresponding* Readmission *value is consistently "Yes," implying that* Family History *is a predictive feature of readmission within these subgroups. However, the same pattern does not hold for other subgroups. This nuanced subgroup-specific variation highlights the importance of tailoring feature selection to distinct subgroups.*

Missing data is a well-known phenomenon in data science, attributed to data issues such as measurement errors, manual data entry issues, data integration flaws and intentional non-responses. Missing data also complicates feature selection, due to the difficulty in assessing feature relevance with incomplete information [14]–[16]. While data can be missing at random, we are especially interested in systematic missingness [17], referring to a pattern that follows a discernible trend or mechanism. Systematic missingness often results from regulatory constraints. For example, privacy regulations might require the removal of sensitive attribute data (*e.g.*, gender). It can also stem from common domain practices, *e.g.*, medical guidelines often dictate routinely conducting different diagnostic tests for different age groups. These regulatory constraints and domain-specific practices often result in a complete absence of feature values for certain subgroups, while they remain available for others. In Table I, systematic missing data is illustrated in the complete absence of cholesterol levels data for sub-groups age 40 and below. Also, whenever data is collected and integrated from multiple jurisdictions, sensitive attributes data may be completely missing in some data sources.

TABLE I: Example Cardiovascular Health Dataset with Systematic Missing Data

Patient ID	Age	Ethnicity	Blood Pressure	Family History	Body Weight	Smoking	Cholesterol	Readmission
1	45	Asian	Normal	Yes	Overweight	Non-smoker	Normal	No
2	62	Caucasian	Null	No	Overweight	Smoker	Normal	Yes
3	35	Caucasian	Normal	Yes	Normal	Non-smoker	Null	Yes
4	50	Caucasian	Null	Yes	Obese	Non-smoker	High	Yes
5	30	Asian	Null	No	Normal	Smoker	Null	No
6	28	Caucasian	Null	Yes	Overweight	Non-smoker	Null	Yes
7	55	Asian	Hypertension	Yes	Overweight	Smoker	Low	No
8	28	Asian	Null	No	Normal	Smoker	Null	No
9	60	Caucasian	Null	Yes	Obese	Non-smoker	Normal	Yes
10	38	Asian	Null	Yes	Overweight	Smoker	Null	Yes
11	54	Asian	Prehypertension	Yes	Normal	Non-smoker	Null	Yes
12	30	Caucasian	Prehypertension	No	Underweight	Smoker	Null	No

We wish to identify feature subsets (of limited cardinality) *that are most informative to a target variable for each subgroup* in the presence of systematic missing data. Feature set informativeness is quantified by the joint MI between predicting features and a target variable, following [18]–[21]. We argue that this task could avoid the extra cost of collecting additional data, and without imputing missing values [19], [22]–[24], which may be ineffective. Instead, we propose to directly estimate feature subsets' MI that are possibly systematically missing in a sub-group, a stark departure from existing works and a fundamental novelty of our work.

Challenges. (1) Features often interrelate and inter-depend, yet their effects can manifest differently on the target variable across subgroups. With an exponential number of feature subsets, the first challenge involves *designing a generalizable model* to exploit feature interdependence across subgroups.
(2) The MI upward closure property [18] states that the joint MI of a feature set (with respect to a target variable) is never greater than that of any of its supersets. Therefore, the designed model should be cognizant of and benefit from the upward closure property when completing missing MI values of feature supersets and subsets. (3) Computing MI values over feature subsets requires enumeration over a power set, which may be prohibitively expensive both computing time and storage space wise. So, the challenge is to investigate the opportunity of sharing computation during training.

Contributions and structure. We design a multiplex graph [25]–[27] (Section IV-A) where features in each subgroup are represented by a distinct graph layer, and train a heterogeneous Graph Neural Network (GNN) [25], [28]–[30] over it (Section IV-C). We cast the MI estimation problem as a graph representation learning task [31], estimating MI scores for feature combinations. The trained model mostly obey the MI upward closure property, effectively increasing the model's predictive power and reducing the effort needed for exact MI computation. *The model is generalizable*, handling both systematic and random missing data, and could be trained on other model agnostic techniques (e.g., feature selection using Pearson correlation [32]). This novel model, to the best of our knowledge, has not been devised before.

Training the GNN is challenged by graph size, with a node cardinality that is the order of a power set of number of features for each of multiple subgroups (Section IV-B). We attend to two distinct computational opportunities. First, we exploit *the relationship between MI and joint entropy, and the chain rule of joint entropy* to demonstrate that the MI of any set of features over the power set could be expressed as a linear function of joint entropy and conditional entropy [33] over a set of constructs. If these constructs are precomputed and materialized, they could be reused repeatedly during training, allowing cost and extensive speed up. Next, we study a subproblem that determines which feature subsets to compute given a budget of MI calculations per layer. We reason that the selected subset should be the one that are uniform random sample of the distribution of MI of all nodes. We present a lazy random walk based algorithm that is guaranteed to converge to a unique stationary distribution producing uniform random samples over the search space (Section IV-B). In Section IV-D, we demonstrate a possible use of the trained model to obtain top-K feature sets with maximum MI score.

A thorough empirical analysis, conducted on both synthetic and real-world datasets, corroborates the efficacy of our approach. Specifically, we show: (1) robustness of the proposed model, which remains effective when a large number of features are systematically missing, compared to imputation based, neural network-based, and Markov blanket-based baselines; (2) efficiency of the sampling algorithm, overcoming a major computational bottleneck while satisfying the uniformity requirement; and (3) proposed solution scalability under varying parameters. The entire code is publicly available.[1]

Section II offers necessary background, followed by data model and problem definition (Section III). Related work is covered in Section VI and Section VII concludes the paper.

II. Preliminaries

In this section we introduce the necessary background on feature selection and MI (Section II-A), and GNNs (Section II-B). Table II provides a summary of notations.

A. Feature Selection and Mutual Information

A feature selection task involves a dataset $\mathbb{D} = \{(x_i, y_i) \mid 1 \leq i \leq n\}$, where x_i is the feature vector of the i-th tuple, y_i is its corresponding label (which may be discrete or continuous), and n is the number of tuples in the dataset. Let F denote the initial feature set. The objective of feature selection is to identify a subset of features $F^m = \{f_1, f_2, \ldots, f_m\} \subset F$

[1] https://github.com/BarGenossar/MISFEAT/

(of some fixed, application-dependent size m) that maximizes the predictive power of a machine learning model with respect to a predefined metric M. This metric can either depend on model performance or be model-agnostic, and is computed using $\mathbb{D}^m = \{(x_i^m, y_i) | 1 \leq i \leq n\}$, a vertical subset of $\mathbb{D}$ containing only the features in F^m, with x_i^m being the projection of the feature vector to the selected features in F^m.

In this study, we select feature subsets that maximize MI [4], [6], [34], commonly used as an indicator of the dependence between two random variables. MI is a model-agnostic feature importance assessment tool that quantifies the amount of information gain of one (or more) random variables when another random variable is observed. Higher MI values indicate greater feature importance. *Joint MI* of an m-size feature set F^m to a target variable Y is defined as

$$I(F^m; Y) = \sum_{\forall i \in f_1} \sum_{\forall j \in f_2} \cdots \sum_{\forall m \in f_m} \sum_{y \in \mathcal{Y}} P(i, j \ldots m, y) \log \frac{P(i, j \ldots m, y)}{P(i, j \ldots m) P(y)} \tag{1}$$

Symbol	Explanation
$\mathbb{D}$	Dataset
F	Feature set
$F' \subset F$	Feature subset for defining subgroups
$P = \{p_1, p_2, \ldots, p_{\|P\|}\}$	Complete set of minterm predicates over F'
$\mathbb{D}_i \subset \mathbb{D}$	A subgroup dataset
$F^S = F \setminus F' \subset F$	candidates for feature selection
$\mathcal{F}$	Powerset of F^S
F_i^-, F_i^+	Systematically missing features and complementary subset
$\mathcal{F}_i^-, \mathcal{F}_i^+$	Feature subset of subgroups with (without) systematically missing features
$F^m = \{f_1, f_2, \ldots, f_m\} \subset F^S$	Feature subset of size m
$\mathbb{D}^m$	Vertical subset of $\mathbb{D}$
F_i^{m*}	Feature subset with highest MI for $\mathbb{D}_i$
F_i^{mj}	j-th top feature subset with $\mathbb{D}_i$'s highest MI
$TopK_i^m$	m-size top-K feature subsets of subgroup $\mathbb{D}_i$
B_i	budget for subgroup $\mathbb{D}_i$
$\mathbb{G}_i = (\mathbb{V}_i, \mathbb{E}_i), \mathbb{G} = (\mathbb{V}, \mathbb{E})$	Subgroup feature lattice graph, multiple lattice graph
$level_{min}, level_{max}$	lower and upper bounds of sampling and prediction levels over $\mathbb{G}$

TABLE II: Table of notations

Example 2. *Using Table I we illustrate MI computation between* Family History *and the target variable* Readmission. *The domain of both features is* $\{Yes, No\}$. *The marginal distribution of* Family History *is* $P(Yes) = \frac{2}{3}$, $P(No) = \frac{1}{3}$, *and for* Readmission *is* $P(Yes) = \frac{7}{12}$, $P(No) = \frac{5}{12}$. *The joint distribution of the variables (ordered as before)* $P(Yes, Yes) = \frac{1}{2}$, $P(Yes, No) = \frac{1}{6}$, $P(No, Yes) = \frac{1}{12}$, $P(No, No) = \frac{1}{4}$. *Plugging values into Eq. 1 (with* $m = 1$*) yields:*

$$\begin{aligned} & I(\textit{Family History}; \textit{Readmission}) = \\ & \frac{1}{2} \log \frac{1/2}{(2/3) \cdot (7/12)} + \frac{1}{6} \log \frac{1/6}{(2/3) \cdot (5/12)} \\ & + \frac{1}{12} \log \frac{1/12}{(1/3) \cdot (7/12)} + \frac{1}{4} \log \frac{1/4}{(1/3) \cdot (5/12)} \simeq 0.23 \end{aligned}$$

The joint MI of {Family History, Body Weight} *and* Readmission *is computed using the joint distributions of* {Family History, Body Weight} *and* {Family History, Body Weight, Readmission}*, as well as* Readmission *marginal distribution.*

We conclude with introducing the *upward closure of MI* property, which becomes handy in addressing the computational challenges of this work. This property guarantees that the MI of a feature set F_1 is never larger than any of its superset F_2 ($F_1 \subseteq F_2$) [18], [35]:

$$I(F_1; Y) \leq I(F_2; Y) \tag{2}$$

B. Graph Neural Networks

Graphs serve as an effective tool for capturing relationships and interactions among data objects. GNNs leverage graph structures, serving as a complementary inference framework. They provide advanced capabilities for performing tasks, including node, link, and graph-level predictions, on such graphs. The fundamental premise of GNNs is the inherent interconnectedness of data objects, enabling information from one entity to influence another [25], [28], [29], [36].

Nodes in a GNN are instantiated with initial feature vectors by assigning attributes or embedding representations based on node properties or external knowledge. These vectors serve as a starting point for the iterative neural message passing mechanism of the GNN, propagating information across the graph. In each iteration, every node receives vector messages transmitted by its direct neighbors and aggregates them (*e.g.*, by using summation, mean, or max pooling) to form a new personal message (hidden representation) for itself. This updated message is sent to neighboring nodes in the subsequent iteration. Such message passing iteration occurs concurrently for all nodes, attributed as a *single GNN layer*.

A general formulation for updating the hidden representation of a node u through message passing in a heterogeneous graph structure can be described as follows.

$$\boldsymbol{h}_u^t = \text{AGGREGATE}\left(\boldsymbol{h}_u^{t-1}, \left\{\left\{\boldsymbol{h}_v^{t-1} \mid v \in \mathcal{N}_r(u)\right\} \forall r \in R\right\}\right) \tag{3}$$

where u is the node of interest, t is the iteration (GNN layer) number, R is a set of node types, $N_r(u)$ is node u's neighborhood with respect to node type $r \in R$, $\boldsymbol{h}_v^{t-1}$ is the previous layer hidden representation of the neighbor v, and AGGREGATE combines u's previous hidden representation with the representations of its neighbors, incorporating mathematical operations and learnable parameter matrix.

The overall architecture of a GNN consists of several layers stacked together. The output of the last layer consists of the latent representations of nodes (embeddings).

In a heterogeneous graph (see Section IV), with multiple node and edge types, conveying information from neighbors may be differ depending on their type [30]. The model captures the diverse structural characteristics of the graph by adaptively aggregating information from different neighbor types.

Typically, message passing and embeddings generation are performed over all nodes in the graph. For selecting F^m, we demonstrate a setting where embeddings are generated only for a subset of the graph, based on the graph structure (Section IV-A) and application needs (sections IV-C and IV-D).

III. Data Model and Problem Definition

Let $F' \subset F$ be a subset of features. We create subgroups of the dataset using F' by replacing $\mathbb{D}$ (Section II-A) with $\mathbb{D} = \bigcup_{i=1}^{|P|} \mathbb{D}_i$, where P is a complete set of minterm (conjunction of simple and negated simple) predicates over $\mathbb{D}$, p_i is a conjunctive predicate over the set of features F', and $\mathbb{D}_i = \sigma_{p_i}\mathbb{D}$ is a selection over $\mathbb{D}$ according to p_i ($1 \leq i \leq |P|$). Example 1 uses predicates such as "Age<40 AND Ethnicity='Asian'." The feature selection task is performed over $\mathcal{F} = 2^{F \setminus F'}$, the entire set of feature combinations, excluding the subgrouping features. We use $F^S = F \setminus F'$ to denote the feature subset that is used for the feature selection process.

Feature subsets, elements of $\mathcal{F}$, can be naturally organized in a lattice [18], a $|F^S|$-dimensional hypercube, where nodes represent feature subsets and an edge exists between two nodes with hamming distance of 1. Subsets in the lattice are arranged in a hierarchical manner based on the number of features included in a combination. The lowest level of the lattice consists of singleton nodes, each representing a single feature. Moving up to the next level, each node represents a pair of features built upon the singletons below. These pairs are formed by considering all possible combinations of two features. This pattern continues for higher levels of the lattice, with each level representing feature combinations of increasing size. As we move up the lattice, the combinations become more complex, incorporating more features from the dataset.

Definition 1. *Given a subgroup data fragment $\mathbb{D}_i$, we say that feature $f \in F^S$ is* systematically missing *in $\mathbb{D}_i$ if*

$$\forall t \in \mathbb{D}_i, t[f] = NULL \tag{4}$$

A feature may be systematically missing in one subgroup but not in others. In Example 1, feature "Cholesterol" is systematically missing only for subgroups defined by predicate "Age<40". We denote by F_i^- and $\mathcal{F}_i^-$ the set of systematically missing features and feature subsets that contain one or more systematically missing features for subgroup $\mathbb{D}_i$, respectively. F_i^+ and $\mathcal{F}_i^+$ are the complementary set, such that $F^S = F_i^- \cup F_i^+$ and $\mathcal{F} = \mathcal{F}_i^- \cup \mathcal{F}_i^+$.

For a subgroup $\mathbb{D}_i$ and a subset of features $\tilde{F} \in \mathcal{F}$, $MI_i^{\tilde{F}}$ is the empirical MI value of $\tilde{F}, Y$ with respect to $\mathbb{D}_i$, where Y is the target variable (*e.g.*, "Readmission" in Example 1). We restrict the selection process to a subset of features of a fixed size $m < |F^S|$, justified by computational complexity of MI for large feature sets. Let $\mathcal{F}^m = \{F^m \in \mathcal{F} \mid |F^m| = m\}$ denote the set of all subset features of size m (level m of the lattice with $|\mathcal{F}^m| = \binom{|F^S|}{m}$ nodes) and F_i^{m*} denote the feature subset that returns the highest empirical MI value for subgroup $\mathbb{D}_i$ of all feature subsets of size m. Therefore,

$$F_i^{m*} = \underset{F^m \in \mathcal{F}^m}{\arg\max}\, MI_i^{F^m} \tag{5}$$

for a subgroup $\mathbb{D}_i$. We denote F_i^{m*} over $\mathcal{F}$ by F_i^{m1} and recursively define F_i^{mj} (the j-th top feature subset).

Definition 2. *The m*-size top-K feature subsets *of subgroup $\mathbb{D}_i$ $TopK_i^m = \{F_i^{m1}, F_i^{m2}, \ldots, F_i^{mK}\}$ is defined recursively as follows.*

$$F_i^{m1} = \underset{F^m \in \mathcal{F}^m}{\arg\max}\, MI_i^{F^m} \tag{6}$$

For $1 < j \leq K$

$$F_i^{j} = \underset{F^m \in \mathcal{F}^m \setminus (\cup_{l=1}^{j-1} F_i^{ml})}{\arg\max} MI_i^{F^m} \tag{7}$$

Problem 1. *Given a feature set of size m and an integer k, for each subgroup $\mathbb{D}_i$, return $TopK_i^m = \{F_i^{m1}, F_i^{m2}, \ldots, F_i^{mK}\}$.*

We wish to identify Top-K sets, each with m features, for each subgroup $\mathbb{D}_i$. Using Examples 1 and 2, if $k = 2$ and $m = 3$, then the goal is to find top-2 sets, each with 3 features for the 4 different subgroups presented in Example 1.

A naïve solution to Problem 1 involves enumerating MI values of $\binom{|F^S|}{m}$ feature subsets of size m, and sort them for the top-K subsets of each subgroup. Such a computation is exponential in $|F^S|$ and therefore quickly becomes computationally expensive, as the number of **candidate** features increases, regardless of m. Moreover, whenever $TopK_i^m \cap \mathcal{F}_i^- \neq \emptyset$, we run into a problem of directly computing the MI of feature subsets due to systematic missing data. Therefore, we are forced to predict rather than compute the MI value of feature subsets that consist of features with systematic missing data. To summarize, systematic missing data, combined with an exponential search space, guide us towards reducing the number of MI computations, replacing computation with prediction even if a feature subset can be computed directly.

IV. Never Miss a Feature with MISFEAT

Equipped with a data model, we introduce MISFEAT, an efficient solution to Problem 1. We aim at retrieving $TopK_i^m$ through a hybrid approach that involves training a model to predict MI of feature subsets. We study this as a graph representation learning problem, framing it as an MI prediction (regression) task to estimate MI scores for feature subsets.

MISFEAT leverages a GNN to propagate information over a graph and capture the domain inherent structure and constraints (Section IV-A). The algorithm is performed in three main steps. First, we introduce a sampling mechanism over the graph structure, to reduce the MI computation complexity (Section IV-B). Once the MI values of the sampled nodes are computed, MISFEAT moves to the training phase, with the outcome of a model that captures the relationships between feature subsets in and among subgroups (Section IV-C). The training phase is followed by $TopK_i^m$ computation for all subgroups, a solution to Problem 1 (Section IV-D).

A. Feature Lattice Graph Construction

A single lattice graph encapsulates all feature subsets per subgroup, and their dependencies. Each feature subset within a subgroup is represented as a node in a graph, such that all feature subsets of a subgroup constitutes the set of nodes. The lattice provides a systematic way to explore the space of feature subsets, starting from individual features and gradually building up to larger subsets. The edges of this hierarchical structure aids in capturing the upward closure property of MI [18] by depicting interrelations between feature subsets. To facilitate information propagation across different subgroups, we connect different lattices. We employ a heterogeneous GNN trained on a node prediction task, specifically predicting the MI score of a feature subset. By leveraging this multiple lattice graph structure, our model captures latent dependencies between feature subsets, both within and across subgroups.

We begin with describing the construction of a single subgroup feature lattice graph (Section IV-A1), and continue with the generation of a multiple lattice graph (Section IV-A2).

*1) **Subgroup Feature Lattice Graph**:* In what follows, we use F_1 and F_2 to denote two feature subsets in $\mathcal{F}$.

Definition 3 (Subgroup feature lattice graph)**.** *Given a subgroup $\mathbb{D}_i$, the* subgroup feature lattice graph *is an undirected graph $\mathbb{G}_i = (\mathbb{V}_i, \mathbb{E}_i)$, where*

- *(nodes:) $\mathbb{V}_i = \mathcal{F} \setminus \emptyset$*
- *(edges:) $\mathbb{E}_i = \mathbb{E}_i^{Inter} \cup \mathbb{E}_i^{Intra}$ such that*
  - *(inter-level edges:) $(v_i, v_j) \in \mathbb{E}_i^{Inter}$ if (1) $\{v_i, v_j\} \subset \mathbb{V}_i$; (2) $v_i = F_1$; (3) $v_j = F_2$; (4) $|F_1| - |F_2| = 1$; and (5) $F_1 \subset F_2$*
  - *(intra-level edges:) $(v_i, v_j) \in \mathbb{E}_i^{Intra}$ if (1) $\{v_i, v_j\} \subset \mathbb{V}_i$; (2) $v_i = F_1$; (3) $v_j = F_2$; (4) $|F_1| = |F_2| = \ell$; (5) $|F_1 \cap F_2| = \ell - 1$; and (6) $|F_1 \cap F_2| > 0$*

We initiate a $1:1$ mapping between feature combinations and their respective representations using binary encoding. In this scheme, each element within the binary vector corresponds to a distinct feature in the dataset. When a feature is included in a particular combination, the corresponding element in the binary vector is assigned with a value of 1; otherwise, it retains a value of 0. With each element in the vector aligned to a specific feature, the dimensionality of these vectors is $|F^S|$.

This binary encoding offers a concise and informative feature combinations representation, facilitating efficient processing within the framework of GNN. It establishes an indispensable association between feature combinations and the hierarchical levels within the lattice. At each level, the number of 1's in a vector mirrors the level number, reflecting the hierarchical nature of the lattice and its alignment with the encoded feature combinations.

Example 3. *Figure 1 illustrates a lattice of the feature set: $F^S = \{f_0, f_1, f_2, f_3\}$. At the first level of the lattice, each binary representation contains a single* 1 *element, corresponding to the underlying feature that forms the singleton combination, while all other elements are* 0*. For example, the binary vector* 0001 *represents the combination $\{f_0\}$, and* 0100 *represents $\{f_2\}$. Moving up to the second level, each feature combination is now denoted by a vector with two 1's. For example, the combination $\{f_0, f_2\}$ is encoded by the vector* 0101*. On top of the lattice lies the vector* 1111 *that represents the feature combination of all available features.*

The lattice hierarchical structure is encoded in the subgroup feature lattice graph as *inter-level edges*. These edges, connecting subsumed feature subsets with cardinality difference of 1, ensure the learning abides by the upward closure of MI property (Section II).

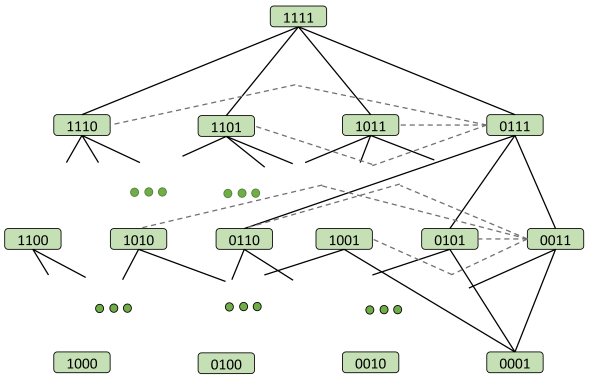


Fig. 1: An Illustration of a single, four level lattice graph.

In addition, the graph contains *intra-level edges*, connecting nodes in the same level of the lattice (cardinality difference of 0). These edges ensure that when the model learns about a feature subset, then other feature subsets that overlap with the former set benefit from this learning. We establish intra-level edges between feature subsets that differ by a single feature.

Example 4. *Consider, again, Figure 1. The gray dashed lines form a partial set of the intra-level edges. For example, the node represented by the vector* 0011 *is connected to all other feature combinations of size* 2 *that include either f_0 or f_1:* 0101, 1001, 0110, 1010*. The black solid lines are the inter-level edges. For illustration of the subsumption-based connection scheme, consider the node* 0111*, representing the feature combination f_0, f_1, f_2. This feature combination subsumes three feature combinations of size* 2*: $\{f_0, f_1\}, \{f_0, f_2\}, \{f_1, f_2\}$. Hence, the node* 0111 *is connected by a solid line to their corresponding nodes, represented by the vectors* 0011*,* 0101 *and* 0110*, respectively.*

Lemma 1. *The size of the subgroup feature lattice graph $\mathbb{G}_i = (\mathbb{V}_i, \mathbb{E}_i)$ is exponential to the number of features both in terms of the number of nodes $|\mathbb{V}_i|$ and the number of edges $|\mathbb{E}_i|$.*

Proof. Recall that each level in the lattice graph corresponds to the size of feature combinations reside in it, ranging from individual features to the full set. Specifically, at level ℓ, the number of nodes is determined by the binomial coefficient, $\binom{|F^S|}{\ell}$. Summing across all levels yields $2^{|F^S|}-1$ nodes, which is exponential to $|F^S|$.

An inter-level edge is created based on subsumption. A node with ℓ features is connected to each of its proper subsets of size $\ell - 1$. The number of such combinations is $\binom{\ell}{\ell-1} = \ell$. Therefore, the total number of inter-level nodes is $\sum_{\ell=2}^{|F^S|} \binom{|F^S|}{\ell} \cdot \ell$. Note that the summation starts from 2 since nodes of the first level are not connected by an inter-level edge to any other subsumed node by Definition 3. To reach a closed-form expression, we recall that $(1 + x)^{|F^S|} =$

$\sum_{\ell=0}^{|F^S|} \binom{|F^S|}{\ell} \cdot x^{\ell}$. Differentiating both sides w.r.t. x yields: $|F^S| \cdot (1+x)^{|F^S|-1} = \sum_{\ell=0}^{|F^S|} \binom{|F^S|}{\ell} \cdot \ell \cdot x^{\ell-1}$. After setting $x = 1$ we get: $|F^S| \cdot 2^{|F^S|-1} = \sum_{\ell=0}^{|F^S|} \binom{|F^S|}{\ell} \cdot \ell$. We note that the term with $k = 0$ does not contribute to the sum because it evaluates to 0. Since we are interested in the sum beginning from 2 we subtract the first element in the sum ($|F^S|$), obtained with $\ell = 1$, from both sides and get: $|F^S| \cdot 2^{|F^S|-1} - |F^S| = \sum_{\ell=2}^{|F^S|} \binom{|F^S|}{\ell} \cdot \ell$. The right-hand side is exactly the expression describing the number of inter-level edges. The left-hand side can be simplified and rewritten as $\frac{|F^S|}{2}\left(2^{|F^S|} - 2\right)$, which is exponential in $|F^S|$.

The number of intra-level edges is determined based on overlap. A node in ℓ, representing a certain combination, is connected to all other nodes within its level that share $\ell - 1$ common features with it, and differ only by a single feature. Every node of interest in level ℓ, contains $\binom{\ell}{\ell-1} = \ell$ combinations of feature subsets, each of size $\ell - 1$. Corresponding to each such combination of size $\ell - 1$, there are $|F^S| - \ell$ nodes that differ from the node of interest by a single feature. This logic dictates the following formula for the number of intra-level edges: $\sum_{\ell=2}^{|F^S|} \binom{|F^S|}{\ell} \cdot \frac{\ell \cdot (|F^S| - \ell)}{2}$, wherein the divisor 2 ensuring that each edge is not counted twice. Thus, the number of intra-level edges is also exponential to $|F^s|$. □

The proof of Lemma 1, offers an exact computation for the number of nodes and edges in a subgroup feature lattice graph, beyond substantiating the exponential nature of the graph. Our empirical evaluation (Section V) shows that the number of edges has a low impact on computation, as long as the overall graph size can be stored in memory as a whole. However, the number of nodes, and in particular the need to compute MI values for a large number of nodes, has a significant impact on the overall algorithmic solution performance. Consequently, we opt to limit the number of layers over which the proposed algorithm iterates (Section IV-C) and to sample nodes for MI computation (Section IV-B).

*2) **Multiple Lattice Graph**:* The subgroup feature lattice graph, $\mathbb{G}_i$, forms a key component in our learning framework, enabling flexibility in learning and prediction by treating identical feature subsets based on the contextual differences (subgroups). Next, we define a multiple lattice graph, generated by interconnecting the subgroup feature lattice graphs. The resulting structure is a multiplex graph, a special type of heterogeneous graph, allowing for the representation of a concept (in our case, feature subset) in different contexts (subgroups). Our multiplex graph incorporates all nodes and edges from the subgroup feature lattice graphs, complemented by a collection of inter-lattice edges.

Definition 4 (Multiple lattice graph)**.** $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, *where:*

- *(nodes:)* $\mathbb{V} = \bigcup_{i=1}^{|P|} \mathbb{V}_i$
- *(edges:)* $\mathbb{E} = \left(\bigcup_{i=1}^{|P|} \mathbb{E}_i\right) \cup \left(\bigcup_{\substack{i,j=1 \\ i \neq j}}^{|P|} \mathbb{E}_{i,j}\right)$ *such that* $(v_1, v_j) \in \mathbb{E}_{i,j}$ *if (1)* $v_i \in \mathbb{V}_i$*; (2)* $v_j \in \mathbb{V}_j$*; (3)* $v_i = F_1$*; (4)* $v_j = F_2$*; and (5)* $F_1 = F_2$

Pairs of lattices are connected by linking nodes of the same feature subsets. Therefore, between any two subgroup feature lattice graphs (and we have $\binom{|P|}{2}$ such pairs) we create at most $|\mathcal{F}| - 1$ edges. This approach ensures a comprehensive integration of information between diverse subgroups.

$\mathbb{G}$ is a heterogeneous graph, representing a comprehensive information integration between diverse subgroups. The graph encompasses various types of edges, reflecting the different relationships between subgroups. The inter-lattice edges correspond to connections between nodes from different subgroup pairs. It is important to note that each subgroup pair results in a unique edge type within this set. Conversely, the in-lattice edges signify relationships within individual subgroups, with each subgroup inducing its distinct edge type within this set.

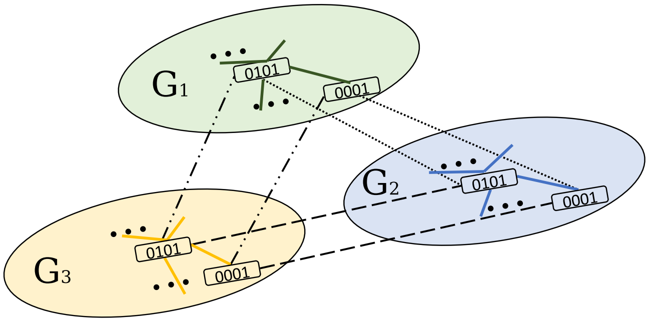


Fig. 2: An Illustration of a multiple lattice graph structure.

Example 5. *Figure 2 illustrates a heterogeneous graph constructed from three subgroups, each represented by a separate lattice graph ($\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_3$). In this example, we consider a simplified scenario where only two nodes, namely 0001 and 0101, are depicted for each lattice. The interconnections between the lattices signify the relationships between different subgroups. Each connection is denoted by a different edge shape, reflecting the diversity among edge types, which varies based on the subgroup pairs involved. A node is directly connected to all other nodes representing the same feature combination across the remaining lattice graphs. Solid lines characterize the in-lattice edges, with each line uniquely colored to denote its lattice origin, thus illustrating the distinct edge types represented within each lattice.*

B. Efficiency Opportunities

Recall that the size of the subgroup feature lattice graph $\mathbb{G}_i = (\mathbb{V}_i, \mathbb{E}_i)$ as well as the multiple lattice graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, are exponential to the feature set size (Lemma 1). Even though done once, training such a gigantic heterogeneous network is prohibitively expensive both computationally and storage space-wise. We study two distinct efficiency opportunities in the training process, as discussed next.

1) Pre-computing and Sharing Computation of MI in $\mathbb{G}_i$*:* As Figure 1 illustrates, each subgroup feature lattice graph represents the power set of features, each being a node. A naïve implementation involves running an exponential number of computations, where for a node with m features there is a need for a nested loop of size $m+1$ for computing MI with the target variable (Eq. 1).

We observe that for each $\mathbb{G}_i$, inter-level edges connect nodes that are related through subsumption. For example, node represented by the vector 0011 is a superset of nodes 0001 and 0010. We therefore argue that the naïve implementation contains redundant computations that can be eliminated by exploiting relationship between joint entropy and MI and the chain rule of joint entropy [33].

MI of a set of x random variables (in our case, $x-1$ features and one target variable) could be represented as follows, where $H(\cdot)$ represents the joint entropy [33] of the random variables involved in the calculation.

$$\begin{aligned} I(A_1, A_2, \ldots, A_x) = & \Sigma_{i=1}^{x} H(A_i) - \Sigma_{1\leq i<j\leq x} H(A_i, A_j) + \\ & \Sigma_{1\leq i<j<k\leq x} H(A_i, A_j, A_k) + \ldots + \\ & (-1)^{x+1} H(A_1, A_2, \ldots, A_x) \end{aligned} \tag{8}$$

Basically, Eq. 8 demonstrates that the MI of x random variables could be expressed as a linear relationship of joint entropy involving all possible subsets of these x variables. Then, the chain rule of joint entropy states

$$\begin{aligned} H(A_1, A_2, \ldots, A_x) = & H(A_1, A_2, \ldots, A_{x-1}) + \\ & H(A_x | A_1, A_2, \ldots, A_{x-1}) \end{aligned} \tag{9}$$

Using Eq. 9, $H(A_1, A_2) = H(A_1) + H(A_2|A_1)$. The generalized form says that the joint entropy of a set of x random variables could be computed by simply adding two constructs: a) joint entropy of any of those $x-1$ random variables, and b) the conditional entropy of the remaining extra variable, conditioned on the same subset of $x-1$ variables. The order of the random variables does not matter in this process.

Given x features, we pre-compute a set of 2^x constructs of joint and conditional entropy: x constructs for x entropy values of the individual features, $\binom{x}{2}$ conditional entropy constructs needed to compute joint entropy of all size 2 feature set, *e.g.*, $H(A_2|A1), H(A_3|A_1), \ldots, H(A_x|A_1)$, because $H(A_iA_j) = H(A_i) + H(A_j|A_i)$, $\binom{x}{3}$ conditional entropy constructs needed to compute joint entropy of all size 3 feature set, *e.g.*, $H(A_3|A_1, A_2)$, $\ldots$, $\binom{x}{x-1}$ conditional entropy constructs needed to compute joint entropy of all size $(x-1)$ feature set. The training process simply adds and subtracts (Eq. 8) these pre-computed constructs to compute MI of any feature set with the target variable.

To understand the extent of computational saving, consider some level $i+1$ of the lattice. To compute MI of a feature set of size $i+1$, we reuse pre-computed constructs of conditional entropy of size i, adding to it the joint entropy of feature set of size i (which was computed in the previous step) using a single addition operation. This is in contrast to a naïve computation that requires repeated computation of conditional entropy.

2) Sampling Subgroup Feature Lattice Graph: Given a subgroup feature lattice graph $\mathbb{G}_i = (\mathbb{V}_i, \mathbb{E}_i)$, we next study how to identify a set $\tilde{\mathbb{V}}_i \subset \mathbb{V}_i$ of nodes of size B_i (a budget parameter). With infinite computing resources and capabilities, the different feature subsets present in $\mathbb{V}_i$ would have produced a distribution of MI ($PDF_{MI}(\mathbb{V}_i)$) with the target variable Y. Ideally, if computation and storage were not bottlenecks, one would retain the whole $\mathbb{G}_i = (\mathbb{V}_i, \mathbb{E}_i)$. Under a budgetary constraint, one natural goal of sampling is thus to retain those $\tilde{\mathbb{V}}_i$ such that if one creates a MI distribution using $\tilde{\mathbb{V}}_i$ that distribution should be as close as possible to the MI distribution of $\mathbb{G}_i = (\mathbb{V}_i, \mathbb{E}_i)$. However, computing the MI distribution of $\mathbb{G}_i = (\mathbb{V}_i, \mathbb{E}_i)$ is infeasible in the first place. We therefore wish to generate a set of sampled nodes such that the distribution of their MIs ($PDF_{MI}(\tilde{\mathbb{V}}_i)$) is a uniform representative of $\mathbb{G}_i$ and design a highly efficient solution that gives theoretical guarantees towards that.

Problem 2 (Sampled subgraph node selection)**.** *Given a budget B_i for subgroup $\mathbb{D}_i$ identify $\tilde{\mathbb{V}}_i$ of size B_i such that $PDF_{MI}(\tilde{\mathbb{V}}_i)$ is a uniform random sample of $PDF_{MI}(\mathbb{V}_i)$.*

Algorithm 1 Sampling Algorithm RANDWALK

```
Require: $B_i$
1: $\tilde{\mathbb{V}}_i = v_1$ /* $v_1$ is generated by setting each $j \in F_i^+$ to 1
   or 0 uniformly at random and independently of all others.
   All bits in $F_i^-$ are assigned to 0 */
2: $c = 2$
3: while $c \leq B_i$ do
4:     With probability $1/2$, goto 8
5:     With probability $1/2$, create $v_c$ from $v_{c-1}$ by
6:      flipping the $j$-th bit, chosen uniformly from $F_i^+$
7:     $\tilde{\mathbb{V}}_i = \tilde{\mathbb{V}}_i \cup v_c$
8:     If $|\tilde{\mathbb{V}}_i| = c$ then $c = c + 1$
9: end while
```

RANDWALK (Algorithm 1) samples using a lazy random walk [37] on the feature $|F^S|$-dimensional hypercube, a walk that starts at a random node in the hypercube that is computable, that is a node that does not contain features with systematically missing data. Recall the binary encoding of a feature subset (Section IV-A1). We generate an initial node representation by setting each of the F_i^+ feature values to 1 or 0 uniformly at random and independently of all others (Line 1). Then, the random walk works iteratively until the budget is consumed. With $1/2$ probability it stays at the same node (and the same level) of the hypercube as it is currently (Line 4). This step is needed to avoid cycles and to ensure convergence. With probability $1/2$, it chooses a feature in F_i^+ and flips its current value, allowing the walk to go to one of the neighboring computable nodes, moving either one level up or one level down (lines 5 and 6). If this subset has not been seen before, it is added to the sample (Line 7). The process ends when B_i samples are collected.

RANDWALK takes exactly $|F^S|$ time to decide the first sample. Then, the choice of finding each subsequent sample takes constant time, but it does not guarantee a new subset. Therefore, running time of RANDWALK can be bounded from below by $\Omega(|F^S| + B_i)$.

Lemma 2. RANDWALK *produces a uniform random sample of $PDF_{MI}(\mathbb{V}_i)$.*

Proof. (Sketch.) It could be shown that RANDWALK induces a Markov chain that is aperiodic and irreducible [38], [39], converging to its unique stationary distribution, which is known to produce a uniform distribution over the $|F^S|$ dimensional hypercube [38], [39]. RANDWALK hence produces a uniform random sample. □

C. Training a Model using $\mathbb{G}$

Each subgroup feature lattice graph $\mathbb{G}_i$ in the multiplex graph $\mathbb{G}$ dictates a different node type (see Section IV-A2). This heterogeneity is reflected in our message passing scheme wherein distinct parameter matrices are learned for distinct node type pairs, allowing for specialized information propagation within the multiplex graph.

We adapt GraphSage [40] for heterogeneous graphs. The aggregated neighborhood message of $v_i \in \mathbb{V}_i$ at layer t is

$$\boldsymbol{h}^t_{N(v_i)} = \left(\frac{\boldsymbol{W}^t_i}{|N_i(v_i)|} \sum_{u_i \in N_i(v_i)} \boldsymbol{h}^{t-1}_{u_i} + \sum_{\substack{j=1 \\ j \neq i}}^{|P|} \boldsymbol{W}^t_{j,i} \boldsymbol{h}^{t-1}_{v_j} \right) \tag{10}$$

where $N_i(v_i)$ denotes the neighboring nodes of v_i within the same subgroup lattice graph $\mathbb{G}_i$. Both $\boldsymbol{W}^t_i$ and $\boldsymbol{W}^t_{j,i}$ are learnable weight matrices of the t-th GNN layer. The former refers to message aggregation from neighbors within $\mathbb{G}_i$, while the latter to message aggregation from a corresponding node v_j of $\mathbb{G}_j$.

To yield the final node representation of v_i at layer t, the aggregated neighborhood message is concatenated to the representation of v_i from the previous layer, and the resulted concatenated vector is then multiplied with another trainable weight matrix as follows:

$$\boldsymbol{h}^t_{v_i} = \sigma\left(\boldsymbol{W}^t_{conc} \cdot \left[\boldsymbol{h}^{t-1}_{v_i} \mid\mid \boldsymbol{h}^t_{N(v_i)} \right] \right) \tag{11}$$

where σ is an activation function (*e.g.,* ReLU), $||$ is concatenation operator and $\boldsymbol{W}^t_{conc}$ is a fully connected layer.

The final node representation, derived after a predefined number of message passing iterations, is fed into a regression head, implemented as a fully connected neural network with *Mean Squared Error* (MSE) loss function, which predicts node MI scores.

Unlike inductive learning, where the model is trained on a subset of nodes and is expected to generalize to unseen nodes in the same graph or similar graphs, MISFEAT's transductive learning involves leveraging information from the entire graph (possibly restricted, for efficiency purposes, to levels between $level_{min}$ and $level_{max}$) during training to make predictions for all nodes, including those not seen during training [25]. In our case, only some nodes are labeled with an MI score due to either systematic missing data or sampling policy (Section IV-B). Labeled nodes serve as input to the MSE loss computation, yet unlabeled nodes also participate in the message passing process.

A separate GNN model is generated for each subgroup. This strategy entails utilizing the entire multiplex graph structure for each model but restricting the loss computation to nodes belonging to a single subgroup at a time. The rationale behind this strategy is to allow each GNN model to focus solely on the specific characteristics and relationships within its corresponding subgroup, while still being influenced by nodes from other subgroups. By training separate models for each subgroup, we aim to enhance the model's ability to capture the nuanced patterns and dependencies unique to each subgroup, ultimately leading to more accurate predictions.

Algorithm 2 GNN Training Epoch

```
Require: $\mathbb{G}$: A multiple lattice graph, with $\tilde{\mathbb{V}} = \bigcup_{i=1}^{|P|} \tilde{\mathbb{V}}_i$
    nodes labeled with ground truth MI values $\tilde{\boldsymbol{MI}}^{GT}$
Require: $\boldsymbol{\Theta}$: Model parameters (weight matrices)
Require: $T$: Number of GNN layers
 1: for $i = 1$ to $|P|$ do
 2:     for $t = 1$ to $T$ do
 3:         for each node $v_i \in \mathbb{V}$ do
 4:             Compute $\boldsymbol{h}^t_{v_i}$ using Eq. (10) and Eq. (11)
 5:         end for
 6:     end for
 7:     for each node $v_i \in \tilde{\mathbb{V}}_i$ do
 8:         $\hat{\boldsymbol{MI}}_{v_i} = \boldsymbol{W}^{out}_i \boldsymbol{h}^T_{v_i}$
 9:     end for
10:     $\mathcal{L} = \frac{1}{|\tilde{\mathbb{V}}_i|} \left\| \hat{\boldsymbol{MI}}_{v_i} - \tilde{\boldsymbol{MI}}^{GT}_{v_i} \right\|_2^2$
11:     Update $\boldsymbol{\Theta}_i$ based on the gradient of $\mathcal{L}$
12: end for
```

A simplified sketch of a single training epoch is depicted in Algorithm 2 for illustration purposes. The computation of revised node representations for **all** nodes in the graph (assume $level_{min} = 1$ and $level_{max} = |F^S|$) through message passing is performed in Line 4. Then, we use these representations to predict the MI values for nodes in $\tilde{\mathbb{V}}_i$, for which we have computed the ground truth MI values (Line 8). The average loss over these predictions is computed in Line 10, followed by an update of model parameters (Line 11). It is noteworthy that one should not infer any computational complexity based on Algorithm 2, as all computations occur synchronously and the use of loops is given for clarity of presentation.

D. Inferencing $TopK^m_i$

MISFEAT, trained on $\mathbb{G}$, serves as a robust tool for predicting MI scores for feature subsets. However. our analysis extends beyond, offering insights into the intricate relationships among feature subsets within. Leveraging the trained model, we flexibly estimate MI scores for uncomputed nodes within subgroups. Consequently, when presented with a subgroup $\mathbb{D}_i$, combination size m, and integer K, our objective is to retrieve $TopK^m_i$. This is done by retaining only nodes positioned at the m-th level at $\mathbb{G}_i$ and sorting them in a descending order with respect to their predicted MI scores.

V. Empirical Evaluation

In this section, we test MISFEAT with real-world and synthetic datasets. We evaluate its efficacy by systematically varying core parameters, and compare it with multiple baselines of different kinds – imputation-based, neural network-based, and Markov blanket after non-trivial adaptation.

Our experiments reveal four consistent observations that showcase the efficacy of our proposed solution. (1) MISFEAT outperforms the baselines consistently and exhibits robustness to significantly missing data compared to the baselines. (2) Our proposed sampling strategy is both effective and efficient in capturing interdependencies among the feature sets and subgroups. (3) Computing MI of all possible feature subsets is a leading computational bottleneck that MISFEAT tactically overcomes and scales under different varying parameters. (4) MISFEAT learns the problem semantics and in particular conserve to a large degree the upward closure property.

Experimental setup (Section V-A) is followed by an efficacy study of MISFEAT, considering multiple baselines (Section V-B). Section V-C delineates the effectiveness of our proposed sampling strategy, followed by an analysis of the upward closure property (Section V-D). Scalability analysis (Section V-E) concludes the empirical study.

A. Experimental Setup

We next discuss benchmark datasets (Section V-A1), implementation details (Section V-A2), baseline methods (Section V-A3), and evaluation metrics (Section V-A4).

1) Datasets: We use three real-world and two synthetic datasets. The key metadata summary of the datasets is presented in Table III.

Real-world datasets. We now describe the three publicly available datasets used in our experiments.

`Employee Attrition [41]` predicts employee attrition (whether the employee stayed at work or left), based on (mostly) categorical features related to the employee's profession history (*e.g.*, number of promotions, company tenure), personal circumstances (*e.g.*, marital status, work-life balance), and job-related aspects (*e.g.*, job satisfaction, job role, company size). Three continuous features, namely *Years at Company, Monthly Income* and *Distance from Home* were discretized with the following range scheme: *Years at Company* : $\{\leq\ 3, 4-6, 7-10, 11-20, 20\ <\}$, *Monthly Income (USD)* : $\{\leq\ 3,000, 3,001-5,000, 5,001-8,000, 8,001-10,000, 10,000\ <\}$, *Distance from Home (miles)* : $\{\leq 3, 4{-}6, 7{-}10, 11{-}20, 20 <\}$. We use age and gender as criteria for subgroup split, forming eight of them. The age range splits are: $\{\leq 25,\ 25-40\ 40-50,\ 50+\}$.

`Mobile [42]` classifies price ranges of different mobile phones using features like battery power, blue tooth capability, memory size, depth, weight and screen size. Subgroups are created by separating mobile phones with single and dual sims.

`Loan` [43] classifies whether an individual will default on a loan payment using features like loan amount, interest rate, employment duration, home ownership, and payment plan. Subgroups are created based on loan grades, a measure of customer's financial credibility.

TABLE III: Metadata of the processed datasets

Datasets	# records	$\lvert F^S \rvert$	# subgroups
Attrition	59,598	19	8
Mobile	2,000	15	2
Loan	67,463	15	3
$SD1$	50,000	15	4
$SD2$	50,000	20	4

We discretize continuous variables through binning, a common practice in the MI literature [44], [45]. For simplicity, we also discretize categorical features having more than 9 distinct values. For systematic missing data, we randomly select a subset of features within each subgroup, using missingness probability p. This method ensures our ability to evaluate the performance of MISFEAT against a valid ground truth. We ensure that every feature is present in at least one subgroup, and each subgroup contains at least one missing feature.

Synthetic datasets. we use synthetic datasets to control specific characteristics relevant to the studied problem. we use synthetic datasets to control specific characteristics of the data relevant to the studied problem. We predefine relevant features, through the use of digital logic to determine target variable values based on existing features. This logic, defined through Boolean algebra, introduces both linear and nonlinear relationships among feature subsets and target variables, and among different feature subsets. To accommodate complex scenarios beyond binary values, we perform these logical operations bitwise. For example, to allow four possible feature values we use two binary bits such that, for example, $01\ XOR\ 11 = 10$ and $01\ AND\ 11 = 01$.

Following [46], [47], we define four types of features, as follows. *Relevant* features are those incorporated into the logical formula, directly influencing the definition of the target variable. *Correlated* features are generated by randomly modifying the target variable's value at a predefined rate. *Redundant* features are created as logical functions of relevant features. Lastly, *irrelevant* features are randomly generated and hold no indicative significance with the target variable. The complexity level and distinctiveness between feature subsets in terms of MI are influenced by both the logical formula and the distribution of the different feature types.

Subgroups are created by randomly partitioning the dataset into vertical fragments. Within each subgroup, feature values are randomly generated from a uniform distribution. Additionally, each subgroup uses a distinct random noise injection drawn from a normal distribution, involving value flips across its features. To simulate datasets with systematically missing data, we randomly select a set of features within each subgroup to make each of them empty with probability p. We employ three logical formulas to form two collections of hyperparameters ($SD1$ and $SD2$), differing in number of features, tuples in the dataset, random noise injection, *etc*. A comprehensive description of the hyperparameter settings and logical formulas is provided in our publicly available repository.[1]

2) Implementation Details: The experiments were executed on a Linux machine with NVIDIA A100 GPU. The GNN was implemented with PyTorch Geometric [48]. Our entire code is publicly available in a GitHub repository.[1]

We employ a heterogeneous variant of GraphSage [40] with two layers and a hidden representation dimensionality of 128. Training lasts for 1000 epochs, utilizing the *Adam* optimizer [49] with a learning rate of 0.001 and weight decay of $5e-4$. Model parameters are selected based on their performance on a validation set, which consists of 20% of the training data randomly sampled.

3) Baselines: We use four baselines, as follows.

(1) KNN [50]. An imputation based baseline, considering the entire feature space of every record and using hamming distance [51] to compute the mode of the feature value of k-nearest neighbors (k is an input parameter of the algorithm) whenever imputation is needed. Imputed values are used with the closed form of MI to quantify feature importance.

(2) Markov Blanket A non-trivial adaptation of [23], integrating missing data imputation inside Markov Blanket (MB) Learning. We first fill missing values using K-Nearest Neighbors (KNN) imputation. Then, using [23] we find the Markov Blanket (MB) of the target variable, which is a set of the most relevant features. The aforementioned two steps repeat until the MB no longer changes. Finally, the top features selected by MB are combined to create sets of size m by maximizing joint MI, from where the best K results that have the highest joint MI are retained.

(3) MLP employs a fully-connected neural network using the binary representation of feature combinations (Section IV-A1) as inputs. The network uses two fully-connected layers with hidden dimension of 64 each. The model is trained against each subgroup separately, with the same parameter selection approach as MISFEAT (Section V-A2).

(4) Arbitrary performs a uniform random selection of samples from $\mathcal{F}$, to be compared against RANDWALK.

Evaluation is conducted with varying probability p (0.2 and 0.5) of features with systematic missing data, sampling rate $B \in \{0.25, 0.5, 0.75, 1.0\}$ per subgroup, $m = 3$ (number of designated features in a set), and $K \in \{5, 10\}$.

4) Evaluation Measures: We use two rank-based measures to evaluate the effectiveness of the proposed solutions, namely nDCG@K and precision@K. Sampling effectiveness is measured using ℓ_1 norm of total variation distance [52], as defined in Eq. 7. We test on each subgroup separately and compute the average across all subgroups. The reported results are computed over three different seeds. The test set for each subgroup consists of the feature subsets that include at least one missing feature, alongside feature subsets that have not been sampled (Section V-C).

Given our special MI-based ranking system, we adjust the measures to our settings, as follows.

Definition 5 (MI-based Normalized Discounted Cumulative Gain)**.** *Given a set of feature combinations of m-size:*

$$\text{nDCG@}K = \frac{\sum_{i=1}^{K} \frac{\mathbb{1}_{\text{rank}_{\text{pred}}[i] \in \text{TopK}^m}}{\log_2(i+1)}}{\sum_{i=1}^{K} \frac{1}{\log_2(i+1)}} \tag{12}$$

where $rank_{pred}$ is a descending order of predicted ranks and $\mathbb{1}_{rank_{pred}[i] \, \in \, TopK^m}$ is an indicator, assigned 1 if the i-th predicted rank is in the ground truth top-k and 0 otherwise.

Definition 6 (MI-based Precision)**.**

$$\text{precision@}K = \frac{rank_{pred}[:K] \cap TopK^m}{K} \tag{13}$$

Definition 7 (ℓ_1 norm of total variation distance between two distributions)**.** *Given two probability distribution function S and P defined on event space $\mathcal{E}$, ℓ_1 norm of total variation distance between S and P is defined as follows.*

$$\delta_{\ell_1}(S, P) = \sum_{e \in \mathcal{E}} |S(e) - P(e)| \tag{14}$$

B. MISFEAT *vs. Baselines*

Comparisons are performed using all datasets with no sampling strategy for MISFEAT. In this work, we consider a setting and problem definition that, to the best of our knowledge, are not present in the literature. Hence, to facilitate comparisons, we modified existing methods to serve as baselines. For all datasets, MISFEAT outperforms the baselines according to both effectiveness measures, with multiple K values.

Effectiveness. Table IV shows the test results on the real-world and synthetic datasets, with a probability $p = 0.2$ for a feature having systematic missing data. For the real-world datasets MISFEAT consistently outperforms all baselines in both nDCG@K and precision@K. As for the synthetic datasets, which are rendered more difficult, MISFEAT showcases superior performance over the baseline, with few exceptions. MLP exhibits the worst performance among the baselines. As the input for MLP is initialized exactly as MISFEAT, its inferior results indicate that without capturing the underlying dependencies between features, addressing the missing data challenge as a learning task is unfeasible. Markov Blanket is better than MLP but worse than KNN. From these experimental analyses a clear message prevails - our proposed solution MISFEAT outperforms all the baselines. Among the baseline solutions, KNN turns out to be the most effective one. The rest of the comparison in the experiment section therefore is conducted between KNN and MISFEAT.

Evaluating Robustness. In this experiment, we compare MISFEAT and KNN with increasing p (likelihood of a feature to have systematically missing data). As expected, both MISFEAT and the baselines including KNN perform worse as p increases. We display effectiveness on the real-world datasets and the average relative drop Δ for 10% increase in p. For nDCG the average drop is calculated as: $\Delta = 1/3 \times \{\frac{nDCG@p=0.2 - nDCG@p=0.3}{nDCG@p=0.2} + \frac{nDCG@p=0.3 - nDCG@p=0.4}{nDCG@p=0.3} + \frac{nDCG@p=0.4 - nDCG@p=0.5}{nDCG@p=0.4}\}$. The average drop in precision is

TABLE IV: Effectiveness comparison of MISFEAT and the baselines, demonstrating that MISFEAT exhibits higher nDCG and precision compared to the baselines for almost all K.

Dataset	Metrics	Algorithm			
		MISFEAT	KNN	Markov	MLP
Attrition	nDCG@5	0.39	0.29	0.34	0.04
	nDCG@10	0.53	0.34	0.39	0.07
	precision@5	0.36	0.25	0.33	0.04
	precision@10	0.49	0.29	0.37	0.08
Mobile	nDCG@5	0.64	0.53	0.05	0.06
	nDCG@10	0.74	0.60	0.07	0.19
	precision@5	0.64	0.47	0.04	0.07
	precision@10	0.67	0.53	0.07	0.20
Loan	nDCG@5	0.56	0.45	0.23	0.02
	nDCG@10	0.64	0.54	0.34	0.02
	precision@5	0.51	0.42	0.18	0.02
	precision@10	0.59	0.52	0.30	0.02
SD1	nDCG@5	0.58	0.45	0.47	0.02
	nDCG@10	0.68	0.54	0.50	0.05
	precision@5	0.52	0.39	0.40	0.02
	precision@10	0.61	0.47	0.42	0.06
SD2	nDCG@5	0.52	0.58	0.54	0.03
	nDCG@10	0.64	0.61	0.59	0.06
	precision@5	0.47	0.56	0.47	0.03
	precision@10	0.57	0.54	0.55	0.06

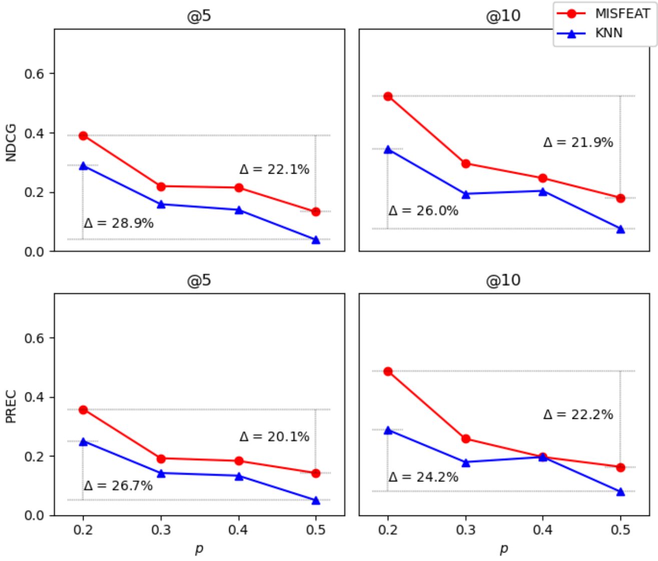


Fig. 3: (Attrition dataset) nDCG and Precision with increasing p. MISFEAT is consistently more effective with smaller Δ values.

calculated analogously. As can be seen in Figure 3, MISFEAT is more robust to high values of p, reflected by its lower Δ in both measures. This highlights the effectiveness of MISFEAT in mitigating the impact of systematically missing data, utilizing its message passing mechanism across subgroups to capture feature set dependencies and reuce the adverse effect of missingness. Figures 4 and 5, demonstrate similar trends, showing that MISFEAT is more robust to high values of p. The drop is higher for both algorithms in the Loan dataset, although MISFEAT always outperforms KNN. The difference can be attributed to lower correlation across subgroups for this dataset.

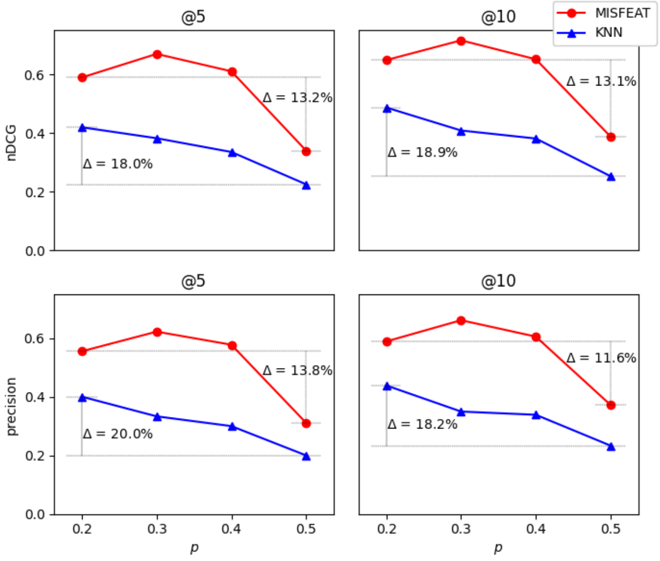


Fig. 4: (Mobile dataset) nDCG and Precision with increasing p. MISFEAT is consistently more effective with smaller Δ values.

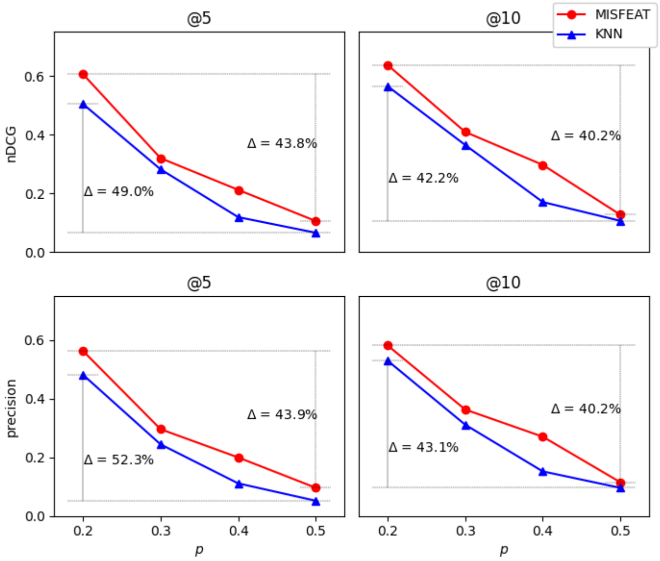


Fig. 5: (Loan dataset) nDCG and Precision with increasing p. For both algorithms, Δ is higher (although MISFEAT performs better) due to low correlation of features across subgroups.

C. Sampling Analysis

The goal of RANDWALK is to produce a uniform random sample from the distribution of all possible feature sets (population). We evaluate the effectiveness of RANDWALK vs Arbitrary on this regard by measuring the ℓ_1 norm of total variation distance of MI between RANDWALK and population and that of Arbitrary and population.

Figure 6 illustrates a comparative analysis of RANDWALK and Arbitrary in terms of ℓ_1 norm of total variation distance on Mobile, Loan, and Employee Attrition datasets. Per Definition 7, S represents the distribution of

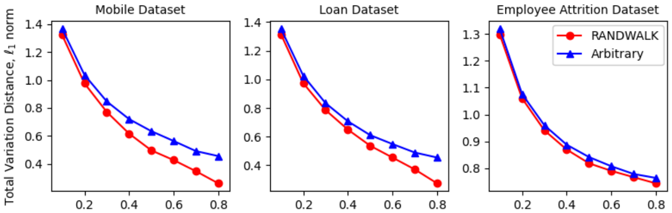


Fig. 6: ℓ_1 norm of total variation distance of MI between RANDWALK and population distribution vs. that of `Arbitrary` and population distribution varying sampling budget B. RANDWALK consistently shows lower variation distance.

MI for sampled nodes, while P represents the distribution of MI for the entire lattice. Evidently, RANDWALK outperforms `Arbitrary` in all datasets with lower δ_{ℓ_1} across different budgets B, hence reliably represents the characteristics of the distribution of MI across all nodes in the lattice.

D. *The Upward Closure Property with* MISFEAT

The upward closure property states that the MI of a smaller feature subset is never larger than that of any of its supersets (Section II-A). In the context of our work, this property is manifested by the lattice structure, where inter-level edges (Section IV-A1) indicate the inclusion relationship between subsets of subsequent levels in the lattice. When all MI scores are available, every node has a higher MI score than its neighbors from the level below reflecting the upward closure property. MISFEAT utilizes the graph structure and learns its topology via the message passing mechanism.

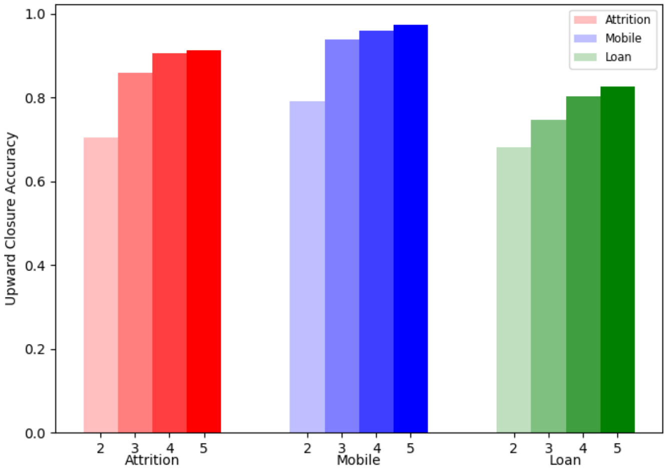


Fig. 7: Upward closure accuracy scores for the real-world data sets over increasing number of levels

We compute the upward closure prediction accuracy as a ratio of the number of times a node was predicted with a higher MI than its neighbor from a lower level to the total number of edges connecting between subsequent levels in the lattice. The empirical results for MISFEAT are shown in Figure 7, where levels are represented by distinct bar, and a group of four adjacent bars refers to the same dataset. As can be seen, MISFEAT gradually improves its performance and feature subsets become more inclusive as the level number increases. This highlights the model's ability to leverage the lattice's structural properties and the relationships between feature subsets.

E. *Scalability of* MISFEAT

To identify computational bottlenecks and study the efficiency of MISFEAT, we compare it against `KNN`, an imputation-based baseline. To ensure a fair comparison between the two approaches, we compare the time required by `KNN` to impute missing data and compute MI over $\mathcal{F}^-$ (feature subsets with missing features) against the training and inference time of MISFEAT. Note that MISFEAT eliminates the needs for MI computation of $\mathcal{F}^-$ during inference by leveraging the trained GNN model.

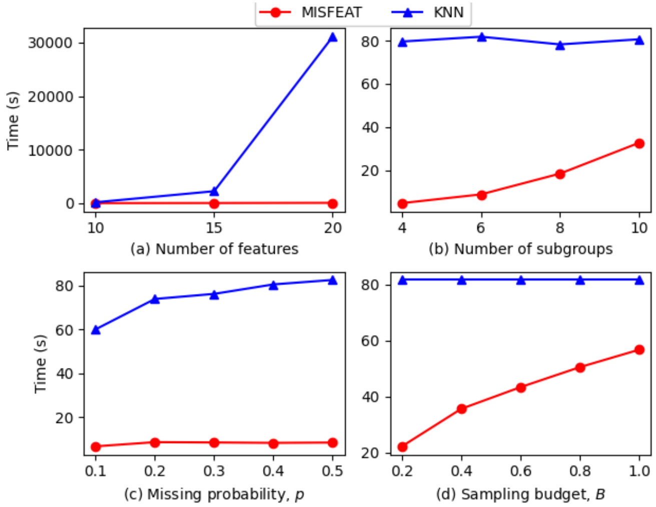
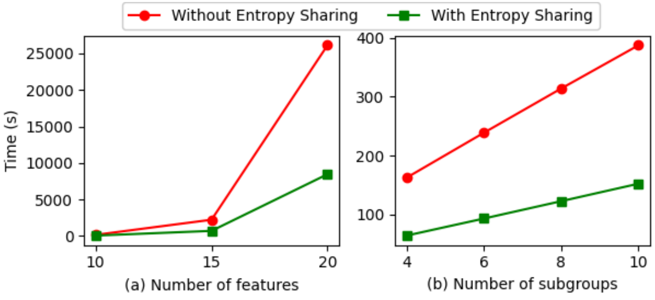


Fig. 8: Execution time: MISFEAT vs. `KNN` by varying different parameters, demonstrating that MISFEAT scales well.

We evaluate scalability of the algorithms using a synthetic dataset of $50,000$ records. In each experiment, we vary the value of a single parameter while keeping the others constant (60% sampling strategy, missing probability $p = 0.2$, $|F^S| = 10$, and 4 subgroups). Figure 8 presents the execution time comparison of MISFEAT and `KNN` by varying each of the four parameters. Figure 8a focuses on varying the number of features. Since `KNN` inference requires computing MI after imputation, it is susceptible to the exponential growth of feature subsets when the number of features increases. MISFEAT, on the other hand, benefits from fast training and inference using GNN, resulting in a significant speedup compared to `KNN`. In Figure 8b we vary the number of subgroups. Since `KNN` does not differentiate between subgroups, its execution time remains stable as their number increases. MISFEAT, on the other hand, demonstrates its scalability and effectively handles a large number of subgroups. Figure 8c compares varying values of missing probability p. As p increases, the size of $\mathcal{F}^-$ grows, requiring KNN to spend more time on imputating MI values for $\mathcal{F}^-$. In contrast, MISFEAT benefit from its inferencing mechanism and increasing p does not add to its overhead. Finally, Figure 8d shows the results across increasing sampling budget, B. With a lower budget, MISFEAT requires less time for training whereas the execution time of KNN is unaffected by the budget, as expected. For MISFEAT, reducing B from 1.0 to 0.2 results in 3× speedup on execution time.



(a) Number of features

(b) Number of subgroups

Fig. 9: MI pre-computation with and without entropy sharing.

To speed up the process of pre-computing MI, we use entropy sharing when computing MI values over the lattice, using $\Theta(2^{|\mathcal{F}|})$ storage space (see IV-B1). Figure 9 illustrate the effectiveness of this approach by varying the number of features (Figure 9a) and the number of subgroups (Figure 9b). Entropy sharing reduces significantly the MI computation time with increasing number of features and subgroups.

We can conclude that exhaustive enumeration of MI is computationally infeasible and MISFEAT scales robustly across varying numbers of subgroups, missing probabilities, and sampling budgets.

VI. Related Work

There are three common methods for feature selection. Filtering uses statistical measures to rank and select features [53], [54]. Wrapper methods utilize predictive performance of a specific learning algorithm [55], [56]. Finally, embedded methods, such as LASSO, integrate feature selection directly into the model training process to enhance model generalization [57]–[59]. No related work, to the best of our knowledge, studies feature selection for systematic missingness considering different subgroups. We next analyze the related work on the use of MI for feature selection, feature selection with missing data, and the use of deep learning for the task.

Feature selection & MI. Existing feature selection algorithms are typically categorizes into four main groups [1]: similarity-based, information-theoretical-based, sparse-learning-based, and statistical-based methods. MI is a model agnostic filtering-based information-theoretic approach that is widely used in feature selection [8], [18], [34], [60]. It quantifies the dependency between variables, thereby assisting in selecting the most informative features for predicting the target variable [8], [34], [60]. [61] [61] introduce a comprehensive framework that utilizes MI to achieve optimal feature selection by maximizing dependency on the target variable, enhancing relevance of the features, and minimizing redundancy among them. *We borrow inspiration from these prior works and consider MI for selecting important feature sets.*

Feature selection & missing data. Meesad and Hengpraprohm [50] propose the use of k-NN based missing value imputation to improve feature selection.

Yu et al. [23] introduce a novel framework for causal feature selection with missing data, integrating multiple imputation and Markov blanket learning to enhance both data imputation and causal discovery in Bayesian networks. The proposed graphical model integrates multiple imputations (*not typically suitable for systematic missing data*) and Markov blanket learning to enhance both data imputation and causal discovery of features. This work is different from ours in two main aspects. First, *we use prediction rather than imputation*. Second, *no support for subgroups or systematic computation of top-K feature subsets, each with m features, is provided*. We non-trivially adapt this solution to serve as a baseline.

Xue et al. [62] have developed a multi-objective approach to feature selection that effectively handles missing data in classification tasks, optimizing for both feature relevance and robustness of the selection process. Zhu et al. [63] have developed a method for multi-label feature selection that effectively addresses the challenge of missing labels, ensuring the robustness and accuracy of feature selection in complex multi-label environments. Prior work has studied a method for feature selection with missing data using MI estimators that directly estimate MI from incomplete datasets, bypassing the need for imputation and preserving the integrity of the original data distribution [19]. Qian and Shu [24] studied MI-based feature selection method for incomplete data that combines tolerance information granules and a forward greedy strategy for efficiency purposes. *Other than the k-NN based approach (implemented as a baseline with inferior performance), none of these techniques handle systematic missing data to produce top-K feature sets with a predefined number of features.*

Feature selection & Deep learning. Gradient-based methods, such as DeepLIFT, deduce feature significance through changes in gradients observed during back-propagation across network layers [64]. Lu et al. [65] present DeepPINK, a methodology that employs filter technique to enhance the reproducibility of feature selection in deep neural networks, focusing on reliable identification of significant features across different datasets. Furthermore, methods like Integrated Gradients offer alternative techniques by attributing the prediction of a neural network to its inputs, thereby providing a deeper understanding of feature importance, which complements these methods in complex model architectures [66]. *These works are not filtering-based and cannot produce top-K feature sets based on MI.* Belghazi et al [67] introduce MINE, estimating MI using a neural network-based discriminator. While their work also tackles computational challenges using sampling *it is not aimed to deal with missing data.*

VII. Conclusions

We introduced MISFEAT, a GNN-based framework for feature selection considering MI with systematic missing data for datasets with distinct subgroups. The proposed model is generalizable. It can handle both systematic and random missing data and can be extended to handle multiple model agnostic feature selection measures. The proposed model is based on lattice organization of feature subsets, benefiting from the MI upward closure property, which it learns well, and targeted sampling of MI computation of a limited number of feature subsets, using multiple efficiency opportunities in training the model. Through a thorough empirical analysis, of both real-world and synthetic datasets, we demonstrate the effectiveness of MISFEATand its different components. In particular, we demonstrate that the efficiency opportunities attain significant speedup. We also show the designed solution accurately predicts missing MI values, even under severe sampling budget limitations. Also, the top-K feature subsets correlate well with the true ranking of feature subsets, offering a useful decision-making mechanism for applications in domains such as medical informatics, where data gathering may be costly or otherwise restricted by regulatory bodies.

References

[1] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM computing surveys (CSUR)*, vol. 50, no. 6, pp. 1–45, 2017.

[2] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[3] M. Dash and H. Liu, "Feature selection for classification," *Intelligent data analysis*, vol. 1, no. 1-4, pp. 131–156, 1997.

[4] X. Chen and S. Wang, "Efficient approximate algorithms for empirical entropy and mutual information," in *Proceedings of the 2021 ACM SIGMOD international conference on Management of data*, 2021, pp. 274–286.

[5] J. R. Vergara and P. A. Estévez, "A review of feature selection methods based on mutual information," *Neural computing and applications*, vol. 24, pp. 175–186, 2014.

[6] C. Pascoal, M. R. Oliveira, A. Pacheco, and R. Valadas, "Theoretical evaluation of feature selection methods based on mutual information," *Neurocomputing*, vol. 226, pp. 168–181, 2017.

[7] J. Huang, Y. Cai, and X. Xu, "A hybrid genetic algorithm for feature selection wrapper based on mutual information," *Pattern Recognition Letters*, vol. 28, no. 13, pp. 1825–1844, 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865507001754

[8] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Transactions on neural networks*, vol. 5, no. 4, pp. 537–550, 1994.

[9] F. Fleuret, "Fast binary feature selection with conditional mutual information." *Journal of Machine learning research*, vol. 5, no. 9, 2004.

[10] B. Omidvar-Tehrani and S. Amer-Yahia, "User group analytics survey and research opportunities," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 10, pp. 2040–2059, 2020. [Online]. Available: https://doi.org/10.1109/TKDE.2019.2913651

[11] A. Munshi, V. Sharma, and S. Sharma, "Chapter 10 - lessons learned from cohort studies, and hospital-based studies and their implications in precision medicine," in *Progress and Challenges in Precision Medicine*, M. Verma and D. Barh, Eds. Academic Press, 2017, pp. 187–207. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128094112000106

[12] X. Sun, J. P. Ioannidis, T. Agoritsas, A. C. Alba, and G. Guyatt, "How to use a subgroup analysis: users' guide to the medical literature," *Jama*, vol. 311, no. 4, pp. 405–411, 2014.

[13] B. Omidvar-Tehrani, S. Amer-Yahia, and L. V. S. Lakshmanan, "Cohort analytics: efficiency and applicability," *VLDB J.*, vol. 29, no. 6, pp. 1527–1550, 2020. [Online]. Available: https://doi.org/10.1007/s00778-020-00625-6

[14] A. N. Baraldi and C. K. Enders, "An introduction to modern missing data analyses," *Journal of school psychology*, vol. 48, no. 1, pp. 5–37, 2010.

[15] X. Zhu, J. Yang, C. Zhang, and S. Zhang, "Efficient utilization of missing data in cost-sensitive learning," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 33, no. 6, pp. 2425–2436, 2021.

[16] R. J. Little and D. B. Rubin, *Statistical analysis with missing data*. John Wiley & Sons, 2019, vol. 793.

[17] D. A. Newman, "Missing data: Five practical guidelines," *Organizational Research Methods*, vol. 17, no. 4, pp. 372–411, 2014.

[18] M. A. Salam, M. E. Koone, S. Thirumuruganathan, G. Das, and S. Basu Roy, "A human-in-the-loop attribute design framework for classification," in *The World Wide Web Conference*, 2019, pp. 1612–1622.

[19] G. Doquire and M. Verleysen, "Feature selection with missing data using mutual information estimators," *Neurocomputing*, vol. 90, pp. 3–11, 2012.

[20] M. Vollmer, I. Rutter, and K. Böhm, "On complexity and efficiency of mutual information estimation on static and dynamic data." in *EDBT*, 2018, pp. 49–60.

[21] E. Schubert, A. Koos, T. Emrich, A. Züfle, K. A. Schmid, and A. Zimek, "A framework for clustering uncertain data," *Proceedings of the VLDB Endowment*, vol. 8, no. 12, pp. 1976–1979, 2015.

[22] U. Pujianto, A. P. Wibawa, M. I. Akbar *et al.*, "K-nearest neighbor (k-nn) based missing data imputation," in *2019 5th International Conference on Science in Information Technology (ICSITech)*. IEEE, 2019, pp. 83–88.

[23] K. Yu, Y. Yang, and W. Ding, "Causal feature selection with missing data," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 16, no. 4, pp. 1–24, 2022.

[24] W. Qian and W. Shu, "Mutual information criterion for feature selection from incomplete data," *Neurocomputing*, vol. 168, pp. 210–220, 2015.

[25] W. L. Hamilton, *Graph representation learning*. Morgan & Claypool Publishers, 2020.

[26] B. Genossar, R. Shraga, and A. Gal, "Flexer: Flexible entity resolution for multiple intents," *Proceedings of the ACM on Management of Data*, vol. 1, no. 1, pp. 1–27, 2023.

[27] P. Yu, C. Fu, Y. Yu, C. Huang, Z. Zhao, and J. Dong, "Multiplex heterogeneous graph convolutional network," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 2377–2387.

[28] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[29] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI open*, vol. 1, pp. 57–81, 2020.

[30] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*. Springer, 2018, pp. 593–607.

[31] Y. He, Y. Zhang, S. Gurukar, and S. Parthasarathy, "Webmile: democratizing network representation learning at scale," *Proceedings of the VLDB Endowment*, vol. 15, no. 12, 2022.

[32] I. Cohen, Y. Huang, J. Chen, J. Benesty, J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," *Noise reduction in speech processing*, pp. 1–4, 2009.

[33] T. M. Cover, J. A. Thomas *et al.*, "Entropy, relative entropy and mutual information," *Elements of information theory*, vol. 2, no. 1, pp. 12–13, 1991.

[34] J. Vergara and P. Estevez, "A review of feature selection methods based on mutual information," *Neural Computing and Applications*, vol. 24, 01 2014.

[35] S. Brin, R. Motwani, and C. Silverstein, "Beyond market baskets: Generalizing association rules to correlations," in *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, 1997, pp. 265–276.

[36] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[37] D. A. Levin and Y. Peres, *Markov chains and mixing times*. American Mathematical Soc., 2017, vol. 107.

[38] P. Brémaud, *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*. Springer Science & Business Media, 2013, vol. 31.

[39] J. R. Norris, *Markov chains*. Cambridge university press, 1998, no. 2.

[40] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[41] U. Zia, "Employee attrition classification dataset." [Online]. Available: https://www.kaggle.com/datasets/stealthtechnologies/employee-attrition-dataset/data

[42] K. Contributors, "Kaggle mobile phone classification dataset." [Online]. Available: https://www.kaggle.com/datasets/iabhishekofficial/mobile-price-classification/data

[43] H. Sai, "Kaggle loan default prediction dataset." [Online]. Available: https://www.kaggle.com/datasets/hemanthsai7/loandefault

[44] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *Neural Networks, IEEE Transactions on*, vol. 5, pp. 537 – 550, 08 1994.

[45] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[46] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "A review of feature selection methods on synthetic data," *Knowledge and information systems*, vol. 34, pp. 483–519, 2013.

[47] F. Kamalov, H. Sulieman, and A. K. Cherukuri, "Synthetic data for feature selection," in *International Conference on Artificial Intelligence and Soft Computing*. Springer, 2023, pp. 353–365.

[48] M. Fey and J. E. Lenssen, "Fast graph representation learning with pytorch geometric," *arXiv preprint arXiv:1903.02428*, 2019.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[50] P. Meesad and K. Hengpraprohm, "Combination of knn-based feature selection and knnbased missing-value imputation of microarray data," *2008 3rd International Conference on Innovative Computing Information and Control*, pp. 341–341, 2008. [Online]. Available: https://api.semanticscholar.org/CorpusID:18993863

[51] J. Han, M. Kamber, and J. Pei, "Data mining concepts and techniques third edition," *University of Illinois at Urbana-Champaign Micheline Kamber Jian Pei Simon Fraser University*, 2012.

[52] J. Chung, P. Kannappan, C. T. Ng, and P. Sahoo, "Measures of distance between probability distributions," *Journal of mathematical analysis and applications*, vol. 138, no. 1, pp. 280–292, 1989.

[53] D. Koller and M. Sahami, "Toward optimal feature selection," in *International Conference on Machine Learning*, 1996. [Online]. Available: https://api.semanticscholar.org/CorpusID:1455429

[54] E. Hancer, B. Xue, and M. Zhang, "Differential evolution for filter feature selection based on information theory and feature ranking," *Know.-Based Syst.*, vol. 140, no. C, p. 103–119, jan 2018. [Online]. Available: https://doi.org/10.1016/j.knosys.2017.10.028

[55] M. M. Mafarja and S. M. Mirjalili, "Whale optimization approaches for wrapper feature selection," *Appl. Soft Comput.*, vol. 62, pp. 441–453, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:44611337

[56] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," in *Machine Learning Proceedings 1994*, W. W. Cohen and H. Hirsh, Eds. San Francisco (CA): Morgan Kaufmann, 1994, pp. 121–129. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9781558603356500234

[57] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996. [Online]. Available: http://www.jstor.org/stable/2346178

[58] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005. [Online]. Available: http://www.jstor.org/stable/3647580

[59] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006. [Online]. Available: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2005.00532.x

[60] G. Brown, A. Pocock, M.-J. Zhao, and M. Luján, "Conditional likelihood maximisation: A unifying framework for information theoretic feature selection," *Journal of Machine Learning Research*, vol. 13, no. 2, pp. 27–66, 2012. [Online]. Available: http://jmlr.org/papers/v13/brown12a.html

[61] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[62] Y. Xue, Y. Tang, X. Xu, J. Liang, and F. Neri, "Multi-objective feature selection with missing data in classification," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 2, pp. 355–364, 2022.

[63] P. Zhu, Q. Xu, Q. Hu, C. Zhang, and H. Zhao, "Multi-label feature selection with missing labels," *Pattern Recognition*, vol. 74, pp. 488–502, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320317303886

[64] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," 04 2017.

[65] Y. Y. Lu, Y. Fan, J. Lv, and W. S. Noble, "Deeppink: reproducible feature selection in deep neural networks," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18. Red Hook, NY, USA: Curran Associates Inc., 2018, p. 8690–8700.

[66] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17. JMLR.org, 2017, p. 3319–3328.

[67] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, "Mutual information neural estimation," in *International conference on machine learning*. PMLR, 2018, pp. 531–540.