Causal-Aware Generative Imputation for Automated Underwriting

Qian Li
School of Computer Science,
University of Technology Sydney
Qian.Li@uts.edu.au

Tri Dung Duong
School of Computer Science,
University of Technology Sydney
TriDung.Duong@student.uts.edu.au

Zhichao Wang
School of Electrical Engineering,
University of New South Wales
zchaoking@gmail.com

Shaowu Liu
Colonial First State,
Commonwealth Bank of Australia
Sean.Liu1@cba.com.au

Dingxian Wang
Ebay
diwang@ebay.com

Guandong Xu
School of Computer Science,
University of Technology Sydney
Guandong.Xu@uts.edu.au

ABSTRACT

Underwriting is an important process in insurance and is concerned with accepting individuals into insurance policy with tolerable claim risk. This process is tedious and labor intensive, which heavily relies on underwriters' domain knowledge and experience, thus is labor-intensive and prone to error. Machine learning models are recently applied to automate the underwriting process and thus to ease the burden on the underwriters as well as improve underwriting accuracy. However, observational data used for underwriting modelling is sparse and incomplete, due to the dynamic evolving nature (e.g., upgrade) of business information systems. Simply applying traditional supervised learning methods e.g., logistic regression or Gradient boosting on such highly incomplete data usually leads to the unsatisfactory underwriting result, thus requiring practical data imputation for training quality improvement. In this paper, rather than choosing off-the-shelf solutions tackling the complex data missing problem, we propose an innovative Generative Adversarial Nets (GAN) framework that can capture the missing pattern from a causal perspective. Specifically, we design a structural causal model to learn the causal relations underlying the missing pattern of data. Then, we devise a Causality-aware Generative network (CaGen) using the learned causal relationship prior to generating missing values, and correct the imputed values via the adversarial learning. We also show that CaGen significantly improves the underwriting prediction in real-world insurance applications.

KEYWORDS

data imputation, automated underwriting, causal-awareness, GANs

ACM Reference Format:
Qian Li, Tri Dung Duong, Zhichao Wang, Shaowu Liu, Dingxian Wang, and Guandong Xu. 2021. Causal-Aware Generative Imputation for Automated Underwriting. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3459637.3481900

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3481900

1 INTRODUCTION

Insurance fund is to create a pool of wealth such that an unfortunate loss incurred by the few can be compensated by the wealth of the many. Underwriting is an important process used by life insurer to assess the risk posed by customer applicants and decide to cover customers with the tolerable risk. For example, the risk in life insurance can be considered as the likelihood of an injury, sickness, disease, disability or mortality. A direct outcome of this process is the decision to accept or decline a policy for a customer, and if accepting, which additional restrictions they should apply, e.g., adding extra *loading* or *exclusion* of claim to protect insurers' profit.

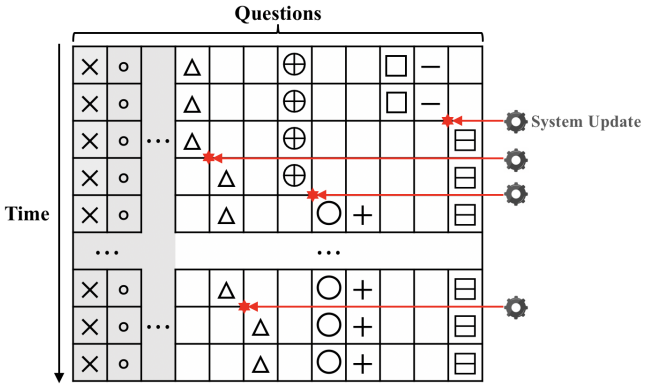


Figure 1: Due to system upgrades, the questionnaire may change (e.g., △), split (e.g., ⊕ = ○+), merge (e.g., □− = ⊟), or remain unchanged (e.g., gray information). For example, one question designed in 2020 is *When was heart disease first diagnosed*, but it has been split into two questions in 2021: *Have you been diagnosed as the heart disease* and *How long have you been*. Apparently, these two new questions of 2021 are not answered by customers of 2020, which results in the missing values in the questionnaire matrix.

Observational data used for underwriting is obtained from application questionnaires containing health, behavioral, and financial attributes. Human underwriters make a underwriting decision by

analyzing a large amount of data, which is labor intensive and prone to error. With recent advances in machine learning, manual underwriting in the insurance industry is desired to gradually transform into an automated process benefiting all parties involved [1, 21]. Nevertheless, existing machine learning approaches fail to perform satisfactory underwriting due to the sparse and incomplete data resulted from business system upgrades. As illustrated in Figure 1, since data used for underwriting are collected over years, some questions keep unchanged and fully observed over time, while other questions may change overtime due to the business system upgrades on questionnaires. The evolution of questionnaires over time leads to the sparsity and incompleteness issue of data, which poses a significant challenge for automated underwriting by machine learning.

Motivation. While it looks like a typical missing data problem, there is an essential difference: the missing information of a question is not simply absent at random (MCAR) or depends on the observed variables (MAR) but drifted to other columns, i.e., *information drift*. However, existing imputation algorithms relying the correlations between the missingness and the observed variables are required to assume the missing mechanism like MCAR or MAR. It is important to note that the missingness in *information drift* is more complicated than the scenario handled by existing imputation, as it does not depend on the observed variables instead unobserved ones. Because the old question(s) and new question(s) are not both available for every applicant, their underlying correlations can not be known from the observed data, leading to the failure of traditional correlation based imputation methods.

Contributions. Without strict assumption of a pre-specified missing mechanism, we aim to impute the missing data conditioned on *information drift* from a causal principle. Concretely, we consider generating a complete dataset conditioned on *information drift* as a causal process, where the causal relationships between the questions embedded in observations. By designing a novel causal structure learning, we can be fully aware of the generating pattern of observations, and then use this for the robust imputing performance. Our contributions are in three-folds:

- Fundamentally different from previous studies, our CaGen is the first deep learning method to impute missing data for automated underwriting from a causal view.
- We design a structural causal model which is capable of learning question-wise causal relations related to *information drift*, producing customized data generating patterns.
- Our CaGen unifies structural causal model and GAN framework, i.e., causality-aware generator and discriminator. Our generator aware of the causality-customized generating pattern, by playing an adversarial game with the discriminator, can significantly boost the underwriting accuracy.

2 RELATED WORK

This section reviews related work from two perspectives: automated insurance underwriting and missing data imputation.

Automated Underwriting. Automation of the underwriting process can assist in a number of ways and benefit all parties involved [25]. The current underwriting completion time frame of weeks can be reduced significantly with the assistance of automated decision making tools [1, 2]. In some instances rule-based expert systems have been crafted to identify and process these simple applications but they are complex and cumbersome to update in light of new information [22]. In addition, the use of machine learning and pattern recognition tools can assist the underwriter in increasing their knowledge base and identifying these complex relationships. Linear and non-linear relationships between features are found by advanced machine learning algorithms, such as ensemble methods, neural network [14, 15], support vector machine, and random forest. The use of machine learning and pattern recognition tools can assist the underwriter in increasing their knowledge base and identifying these complex relationships [5, 19].

Data Imputation. Methods for imputing missing data range from replacing missing values by the column average to complex imputations based on various statistical models. Successful statistical models for data imputation can be categorized as discriminate [9, 13**?**] or generative [8, 26]. Discriminative methods include MICE [4], MissForest [20], k-nearest neighbors (KNN), and matrix completion [10, 12, 17]. A drawback of NN methods is that their performance depends on k. For example, KNN impute typically performs well when k is between 5 and 10, but the performance deteriorates for larger values of k. Alternatively, generative methods for imputation include *MIDA* (*Multiple Imputation with Denoising Autoencoders*) [8], *GAIN* (*Generative Adversarial Imputation Nets*) [26], and *MIWAE* (*Missing Data Importance-Weighted Autoencoder*) [16]. Most of approaches for data imputation are based on [7] that makes assumptions about the underlying distribution and fails to generalize well for data of mixed modalities. Statistical methods for data imputation often provide useful theoretical properties but exhibit notable shortcomings: (1) they tend to make strong assumptions about the data distribution; (2) they fail to exploit the causal interactions between multiple observations and features, which limits their the ability to learn from samples with missing data across different distributions.

3 PROBLEM DEFINITION

In this section, we firstly introduce the problem formulation and notations. We have a dataset $\{(\mathbf{X}, Y)\}$ drawn from the questionnaires of life insurance application. The questionnaire covers a large range of information about each applicant including family and medical history, occupation details, finances as well as leisure activities. Let $\mathbf{X} \coloneqq [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n]$ be a collection set of samples, each sample $\boldsymbol{x}_i \in \mathbb{R}^d$ consists of the applicant's binary responses to d questions $\{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_d\}$. An exclusion outcome $y_i \in Y$ indicating the presence of particular exclusions for individual from making a specific claim due to information $\boldsymbol{x}_i$ gathered from the applicants questionnaire.

Recall that applicants can not provide answers to all d questions $\{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_d\}$ due to the system updates. Among them, n_o questions are fixed over time leading to n_o fully observed features denoted as $\{\boldsymbol{v}_o\}$, and $d - n_o$ questions changing as the system updates are partially observed features denoted as $\{\boldsymbol{v}_p\}$. In other words, V_p is the set of feature variables that are missing in at least one applicant record. In the missing data problem, we know exactly which entries in each sample are missing. Therefore, we can represent an incomplete data case as a pair of a partially-observed data vector

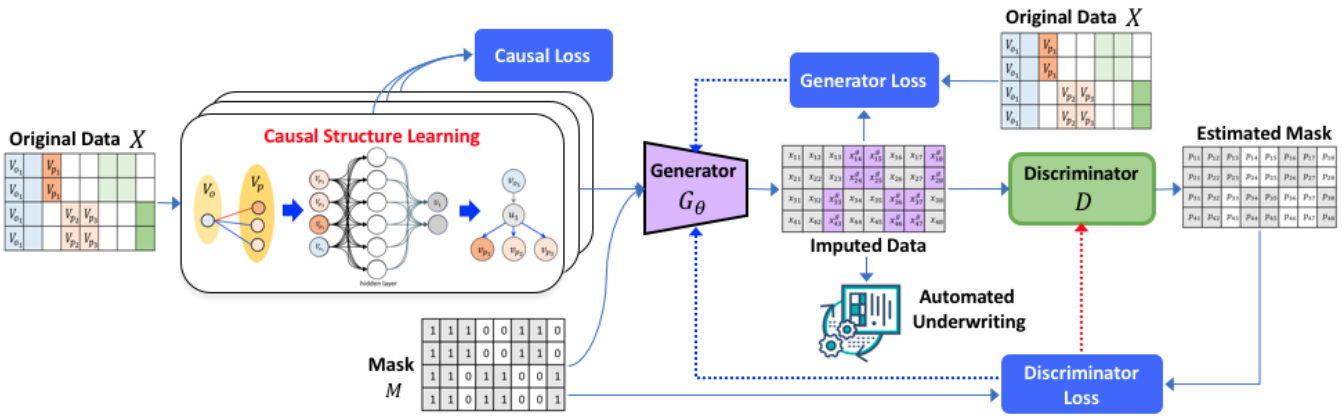


Figure 2: Overall architecture of our method for data imputation, including causal structure learning and generative network. The solid lines and dash lines are the inputs and backpropogations, respectively. The causal structure learning is to discover the underlying causality between fully observed $\{v_o\}$ and partial $\{v_p\}$, then the results are as the input of generative network for causality-aware imputation. Finally, the imputed data generated by our method will be applied for automated underwriting.

$\boldsymbol{x}_j \in \mathbb{R}^d$ and a corresponding mask $\boldsymbol{m}_j \in \{0,1\}^d$ indicating which entries in $\boldsymbol{x}_j$ are observed.

Based on the above descriptions, we consider two tasks: (1) feature imputation, where the goal is to predict the missing feature values $\boldsymbol{x}_j$ at $\boldsymbol{m}_j = 0$ for downstream processing. (2) downstream exclusion prediction, where the goal is to predict exclusion outcome y for a new applicant.

4 CAUSALITY-BASED ADVERSARIAL IMPUTATION

The overall framework of our data imputation framework is illustrated in Figure 2. We first give a brief overview of the proposed method, and then introduce three elements of the model in detail. At last, we introduce the causality adversarial learning to train the model in an end-to-end manner for imputation.

4.1 Overview

To systematically study the data imputation of underwriting task, we leverage causal learning to express the causal mechanism from partially observed data to latent cause confounder, and design a novel causality-aware GAN generator into the whole adversarial learning model as a competitor. The structure of our approach is illustrated in Figure 2. It includes three elements: a causal learning network, a causality-aware generator and a discriminator.

4.2 Causal Structural Learning

4.2.1 Motivation. Our fundamental step is to discovers the causal graph underneath the observed underwriting data with missing values. Using causal knowledge in causal graph, we can deduce the generating process of the whole dataset and impute the missing values via the observed values. This section exploits the idea that how to construct the causal graph from the observed data.

From a causal perspective, we argue that the dependencies among observed features are partially or completely due to the interactions among their corresponding latent confounders [6, 11]. Such a latent factor is defined as a *confounder* in causal inference, with the definition as below.

Definition 1 (Confounder). Variable $\boldsymbol{u}$ is a confounder of $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ if their causal structure is:

$$\boldsymbol{v}_i \leftarrow \boldsymbol{u} \rightarrow \boldsymbol{v}_j \tag{1}$$

$\boldsymbol{u}$ is called hidden confounder if $\boldsymbol{u} \notin \boldsymbol{v}$

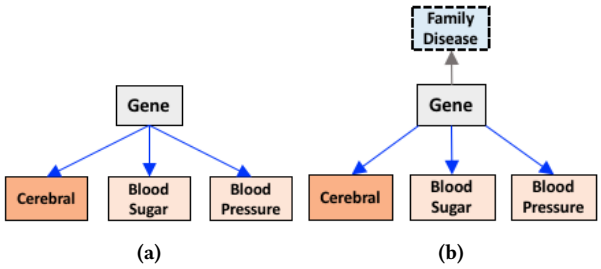


Figure 3: An example of confounder in underwriting data, where *cerebral* is split into *blood sugar* and *blood pressure* due to questionnaire updates: (a) Gene as a unobserved confounder is a common cause (indicated by blue arrows) of partially observed *cerebral* and *blood sugar* and *blood pressure*. (b) Fully observed *family disease* can provide proxy information (indicated by a grey arrow) to infer confounder *Gene*.

According to Definition 1, we assume there are a set of confounders that drive both the generating process of the observed features in the insurance data. Figure 3 gives a typical confounder example in real-life insurance applications, where the latent variable **Gene** causes a set of features (including **cerebral**, **blood pressure** and **blood sugar**). If we can estimate the unobserved confounder (e.g., **Gene**) from the proxy variable (e.g., **family disease**), then we can use the estimated confounder to infer the missing values

of its resulting features like **blood pressure**, **blood sugar**. Consequently, the identification of the structural causal model about how variables of interest interacting with each other through causal links provides a causal evidence to generate the missing values.

This motivates us to propose a causal modeling framework that captures these latent confounding variables, and utilizes the causal information in recovering missing data. The discovery of causal structure over the insurance data consists of two parts: 1) identification of the correlated features sharing one latent confounder. 2) estimation of the latent confounder used to infer the missing data.

4.2.2 Identification of Causal Skeleton. Our goal in this subsection is to infer structural confounder model that learns the causal relationships among variables in feature set $\mathbb{V}$ to best describes the generating procedure of observations.

Following the causality analysis theory [18, 24], we assume that latent confounder $\mathbf{U}$ is the always parent node of the features ($\{\boldsymbol{v}_o\}$, $\{\boldsymbol{v}_p\}$). With the assumption of the predefined causal direction, constructing the structural causal model becomes a easy task as it only requires to find the correlated features sharing the confounder. For example, as shown in Figure 4, $\boldsymbol{v}_{o_1}$ representing the *family disease* is a fully observed feature. Suppose we observe that two partially observed features $\boldsymbol{v}_{p_1}$ and $\boldsymbol{v}_{p_2}$ are highly correlated to $\boldsymbol{v}_{o_1}$, then we can infer that $\boldsymbol{v}_{o_1}, \boldsymbol{v}_{p_1}$ and $\boldsymbol{v}_{p_2}$ are within one structural confounder model. In particular, $\boldsymbol{v}_{o_1}$ is fully observed and can be used as the proxy variable to estimate the confounder $\mathbf{U}_1$. With the estimated $\mathbf{U}_1$ as the parent node, the missing values in the children nodes like $\boldsymbol{v}_{p_1}$ and $\boldsymbol{v}_{p_2}$ can be recovered as well.

Identifying the causal structure is transformed into finding the correlations among $\boldsymbol{v}_o$ and $\boldsymbol{v}_p$ to identify the nodes within one structural confounder model. A nature way to exploit correlations is computing the adjacency matrix $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{n_o \times (d-n_o)}$. We compute adjacency matrix A from a similarity matrix $\mathbf{S}$ of $\{\boldsymbol{v}_p\}$ and $\{\boldsymbol{v}_o\}$, where the entry $\mathbf{S}_{i,j}$ representing the distance between $\boldsymbol{v}_i \in \{\boldsymbol{v}_o\}$ and $\boldsymbol{v}_j \in \{\boldsymbol{v}_p\}$. The computation of distance is computed pairwise for all covariates using the Euclidean distance, which allows to use non-missing elements of both covariate vectors as follows.

$$\mathbf{S}_{ij} = dis\left(\boldsymbol{v}_i, \boldsymbol{v}_j\right) = \left[\frac{1}{n_{ij}} \sum_{k=1}^{d} \left|\boldsymbol{v}_{ik} - \boldsymbol{v}_{jk}\right|^2 \left(\boldsymbol{I}_{ik}\boldsymbol{I}_{jk}\right)\right]^{1/2} \tag{2}$$

where we have indicator $\boldsymbol{I}_{ik} = 1$ if $\boldsymbol{v}_{ik}$ is observed, and $n_{ij} = \sum_{k=1}^{d} \boldsymbol{I}_{ik}\boldsymbol{I}_{jk}$ denotes the number of observed components both in $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$. The distance results $\mathbf{S}$ after the pruning step by a threshold will be used as adjacency matrix

$$\mathbf{A}_{ij} = \begin{cases} 1, & \mathbf{S}_{ij} > \tau \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

for skeleton of the structural confounder model. If the entry $\mathbf{S}_{ij}$ in the distance matrix is above this threshold τ, we claim that fully observed feature $\boldsymbol{v}_i$ and partially observed $\boldsymbol{v}_j$ are connected and within the same structural confounder model. Note that threshold τ is defined as a percentile that is computed on each row of $\mathbf{S}$.

4.2.3 Confouder Learning. Having identifying the skeleton including feature variables as children nodes, we now discuss the how to estimate their corresponding parent confounders from the proxy variables. The increasing availability of fully observed features as proxy variables enables unobserved confounders to be inferred. For example, it may be difficult to obtain **Gen** directly, and fully observed features such as **family disease** can be considered as proxy variables instead to replace or infer the latent **Gene**.

Algorithm 1 Causal Structural Learning

Input: fully observed features $\{\boldsymbol{v}_{o_i}\}_{i=1}^{n_o}$, partially observed features $\{\boldsymbol{v}_{p_j}\}_{j=1}^{n_p}$, a threshold τ
Compute distance matrix $\mathbf{S}$ by Eq. (2)
if $\mathbf{S}_{ij} > \tau$ **then**
  Connect $\boldsymbol{v}_{o_i}$ with $\boldsymbol{v}_{p_j}$ by an edge
end if
Iterative connecting generates n_u seperate causal sketelons $\{\mathcal{G}_i\}_{1 \le i \le n_u}$
for each $\mathcal{G}_i$ **do**
  Initialize $\boldsymbol{u}_i$ as a d-dimensional random vector
  for each $\boldsymbol{v}_p, \boldsymbol{v}_o$ in $\mathcal{G}_i$ **do**
    Set $\boldsymbol{u}_i$ as the parent of $\boldsymbol{v}_p$ and $\boldsymbol{v}_o$
    Add directed edges $\boldsymbol{u}_i \rightarrow \boldsymbol{v}_p$ and $\boldsymbol{u}_i \rightarrow \boldsymbol{v}_o$
    Remove the edges between $\boldsymbol{v}_o$ and $\boldsymbol{v}_p$
  end for
  for K-steps **do**
    Update θ and $\boldsymbol{u}_i$ using stochastic gradient descent for (5)
  end for
end for
return causal graphs and confounders $\{\mathcal{G}_i, \boldsymbol{u}_i\}_{1 \le i \le n_u}$

Consider the common scenario where there are n_u latent confounders $\boldsymbol{u}_o \in \mathbf{U}$ that influence n_o fully observed features ($\boldsymbol{v}_i \in \mathbf{V}_o, 1 \le i \le n_o$). We initialize a set of n-dimensional random vectors $\{\boldsymbol{u}_1, \cdots, \boldsymbol{u}_{n_u}\}$, the i-th fully observed feature $\boldsymbol{v}_i$ conditioned on its cause $u_{o=pa(i)}$ can be represented by

$$\boldsymbol{v}_{i=1\ldots n_o} = f_\theta(\boldsymbol{u}_{o=pa(i)}) \tag{4}$$

where $o = pa(i)$ represents a parent (or cause) of i and $f_\theta(\cdot) : \mathcal{U} \rightarrow \mathcal{V}$ is a neural network parametrized by θ. Using the structure identified in Eq. (4), the confounder learning task then turns into finding a meaningful organization of $\boldsymbol{u}_o$, such that they can be mapped to their target observed features. Rather than using autoencoders [3], which is a pair of neural networks formed by an encoder and a decoder, we produce the latent confounder $\boldsymbol{u}$ by a parametric encoder f_θ, but learned freely in a non-parametric manner. Particularly, we seek to jointly learn the parameters θ and the optimal confounder $\boldsymbol{u}_{o=pa(i)}$ for each fully observed feature $\boldsymbol{v}_i$, by solving:

$$\min_{\theta \in \Theta} \frac{1}{n_o} \sum_{i=1}^{n_o} \left[\min_{\boldsymbol{u}_{o=pa(i)}} \mathcal{L}_C\left(f_\theta\left(\boldsymbol{u}_{o=pa(i)}\right), \boldsymbol{v}_i\right)\right] \tag{5}$$

4.2.4 Optimization. For any choice of differentiable generator, the objective (5) is differentiable with respect to $\boldsymbol{u}$, and $\boldsymbol{\theta}$. Therefore, we will learn $\boldsymbol{u}$ and $\boldsymbol{\theta}$ by Stochastic Gradient Descent (SGD). The gradient of (5) with respect to $\boldsymbol{u}$ canbe obtained by backpropagating the gradients through the generator function. We project each $\boldsymbol{u}$ back to the confounder space $\mathcal{U}$ after each update. We initialize $\boldsymbol{u}$ by sampling them from a Gaussian distribution.

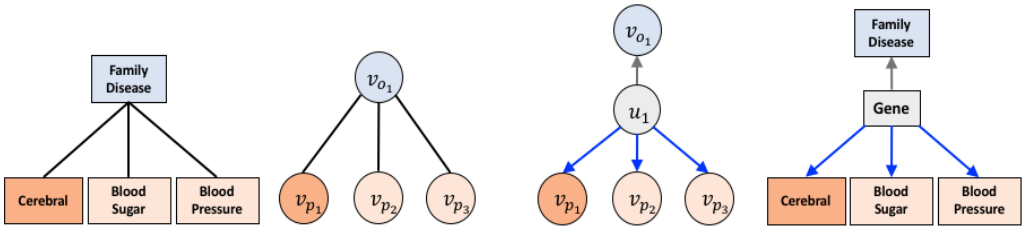


Figure 4: Causal learning process: (a) identifying causal sketelon by connecting fully observed feature (*family disease*) with partially observed features (*cerebral, blood sugar, blood pressure*) (b) constructing structural causal model with an estimated confounder u.

4.3 Adversarial Learning on Causal Structure

In this section, we develop the GAN based framework that utilizes the causal information underlying the causal structure model to impute the missing values.

4.3.1 Causality-based Generator for Imputation. The structural confounder model in Figure 4 depicts how children variables are generated by their parental confounder variables, which reveals the generation of observations in underwriting scenario. Based on causal structure model in Figure 4, we wish to design a causality-aware generator G to impute the missing values.

Based on the causal graph output by Algorithm 1, we wish to learn weights that quantify the dependency between confounders and feature variables. Hence, we design a novel generator G to learn W and further recover partially observed features in observations. In particularly, g_W generates $\tilde{\boldsymbol{v}}_i$, i.e., i-th feature variable with missing values, by a function of its parent $\boldsymbol{u}_{o=pa(i)}$, a weight $W_{o,i}$ and a noise variable ϵ_i:

$$\tilde{\boldsymbol{v}}_i = g_W(\boldsymbol{u}_{o=pa(i)}) \tag{6}$$

Based on the mask $\boldsymbol{I}$ indicating the missing position of $\boldsymbol{v}_i$, the imputed data $\hat{\boldsymbol{v}}^{(i)}$ for the i-th data can be expressed as

$$\hat{\boldsymbol{v}}_i = \boldsymbol{I} \odot \boldsymbol{v}_i + (1 - \boldsymbol{I}) \odot \tilde{\boldsymbol{v}}_i \tag{7}$$

That is, the vector obtained by taking the partial observation $\boldsymbol{v}_i$ and replacing each missing value with the corresponding value of $\tilde{\boldsymbol{v}}_i$.

Recall that each record $\boldsymbol{x}$ consists of an applicant's responses to questions/features $\{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_d\}$, that is, $\boldsymbol{v}_i$ is the i-th column of original data matrix $\mathbf{X}$. By replacing all partially observed $\boldsymbol{v}_i$ in $\mathbf{X}$ with $\hat{\boldsymbol{v}}_i$ imputed by Eq. (7), we can generate an imputed data $\hat{\mathbf{X}}$. In the following, we use the notation $\hat{\mathbf{x}} \in \mathbb{R}^d$ as one imputed record by generator g_W, i.e., one row of $\hat{\mathbf{X}}$.

4.3.2 Discriminator. As in the GAN framework, we introduce a discriminator, D, that will be used as an adversary to train G. The discriminator D of the GAN architecture receives $\hat{\boldsymbol{x}}$, and instead of trying to determine if each component from the output of the generator is either *completely* real or *completely* fake, the model attempts to distinguish if every component is either original (observed) or imputed (fake). In other words, the discriminator needs to be trained to maximize the probability of predicting the mask $\hat{\boldsymbol{m}}$ of $\hat{\boldsymbol{x}}$. Let $\boldsymbol{z} = (z_1, \cdots, z_d) \in \{0, 1\}^d$ and we define a hint vector

$$\boldsymbol{h} = \boldsymbol{z} \odot \hat{\boldsymbol{m}} + 0.5(1 - \boldsymbol{z}). \tag{8}$$

The hint vector $\boldsymbol{h}$ reveals partial information about the missingness of the original sample. Conditional on $\hat{\boldsymbol{x}}$ and $\boldsymbol{h}$, the optimal discriminator $D : \mathcal{X} \times \mathcal{H} \to [0, 1]^d$ outputs a d-dimensional vector. The k-th component of D is the probability of k-th component of $\boldsymbol{x}$ was observed, and can be represented by

$$D(\hat{\boldsymbol{x}}, \mathbf{h}) = \check{p}_k(\hat{m}_k = 1 | \hat{\boldsymbol{x}}, \boldsymbol{h}) = \frac{p(\hat{\boldsymbol{x}}, \mathbf{h}, \hat{m}_k = 1)}{p(\hat{\boldsymbol{x}}, \mathbf{h}, \hat{m}_k = 1) + p(\hat{\boldsymbol{x}}, \mathbf{h}, \hat{m}_k = 0)} \tag{9}$$

where $\hat{m}_k$ is k-th component of mask $\hat{\boldsymbol{m}}$.

4.3.3 Objective Function. The overall procedure is to impute all samples $\{\boldsymbol{x}_j\}_{1 \le j \le n}$ by simultaneously training a causality-aware generator G and a discriminator D. Hence, the overall loss is defined as the sum of discriminator loss $\mathcal{L}_D$, generator loss $\mathcal{L}_G$ and reconstructed loss $\mathcal{L}_{\text{rec}}$.

Particularly, G imputes the partially observed features of $\boldsymbol{x}$ by Eq. (7) to reconstruct the imputed $\hat{x}$. The discriminator D tries guess for every component of $\hat{x}$ if its variable value is either original or imputed. Namely, for every newly imputed $\hat{x}$, D is trained to maximize $\check{\boldsymbol{p}} = [\check{p}_1, \cdots, \check{p}_n]$ by computing $\check{p}_k$ in Eq. (9), i.e., the probability of correctly predicting the mask of $\hat{\boldsymbol{x}}$. Given the mask $\boldsymbol{m}$ of original $\boldsymbol{x}$, the discriminator loss $\mathcal{L}_D$ is defined as

$$\mathcal{L}_D(\boldsymbol{m}, \check{\boldsymbol{p}}, \boldsymbol{z}) = \sum_{k:\boldsymbol{z}_k=0} \Big[m_k \log(\check{p}_k) + (1 - m_k) \log(1 - \check{p}_k) \Big] \tag{10}$$

where m_k is the k-th component $\boldsymbol{m}$ and $\check{p}_k$ is in Eq. (9). Note that the outputs of discriminator D that depend on generator G are samples (i.e., indexed by k) corresponding to $\boldsymbol{z}_k = 0$. We design the discriminator loss (10) and optimize the discriminator D with a fixed generator G using mini-batches of size k_D to produce such kinds of outputs. For such a mini-batch, the discriminator D with parameters θ_D is trained to optimize:

$$\max_D \sum_{j=1}^{k_D} \mathcal{L}_D(\mathbf{m}_j, \check{\boldsymbol{p}}_j, \mathbf{z}_j) \tag{11}$$

G in fact outputs a value for the *entire* data vector including the observed components and missing components. We apply two loss functions $\mathcal{L}_G$ and $\mathcal{L}_{\text{rec}}$ to the missing components and the observed components, respectively. For the missing component indicating by $m_k = 0$, generator G outputs the k-th component of $\boldsymbol{x}$, denoted as x_k. The probability that recognizing x_k as observable one ($m_k = 1$) by D is $\check{p}_k$. D aims to minimize $\check{p}_k$ since it is simulated

Algorithm 2 CaGen: Causal-Aware Generative Imputation

Input: observed dataset $\{\mathbf{x}_j, \mathbf{m}_j\}_{j=1}^{n}$ with n_p partially features $\{\boldsymbol{v}_i\}_{i=1}^{n_p}$, causal graphs $\{\mathcal{G}_j, \boldsymbol{u}_j\}_{1\le j\le n_u}$, random binary vectors $\{\boldsymbol{z}_j \in \{0,1\}^d\}_{j=1}^{n}$, noise vectors $\{\mathbf{e}_j\}_{j=1}^{n}$

while not converge **do**

Step 1: Discriminator optimization

Draw k_D samples from the dataset $\{\mathbf{x}_j, \mathbf{m}_j\}_{j=1}^{n}$

Draw k_D i.i.d. samples from $\{\mathbf{z}_j\}_{j=1}^{n}$, $\{\mathbf{e}_j\}_{j=1}^{n}$

Construct data matrix $\mathbf{X}_D$ for $\{\mathbf{x}_j\}_{j=1}^{k_D}$

Construct the imputed matrix $\hat{\mathbf{X}}_D$ by imputing $\{\boldsymbol{v}_i\}_{i=1}^{n_p}$ via Eq. (7)

for $j = 1, \cdots, k_D$ **do**

Set $\hat{\boldsymbol{x}}_{\boldsymbol{j}} = \hat{\mathbf{X}}_D[j,:]$

$\mathbf{h}_j = \mathbf{z}_j \odot \mathbf{m}_j + 0.5(\mathbf{1} - \mathbf{z}_j)$

Compute $D(\hat{\mathbf{x}}_j, \mathbf{h}_j)$ by Eq. (9)

end for

Update D via optimizing (11).

Step 2: Generator optimization

Draw k_G samples from the dataset $\{\mathbf{x}_j, \mathbf{m}_j\}_{j=1}^{n}$

Draw k_G i.i.d. samples from $\{\mathbf{z}_j\}_{j=1}^{n}$, $\{\mathbf{e}_j\}_{j=1}^{n}$

Construct data matrix $\mathbf{X}_G$ for $\{\mathbf{x}_j\}_{j=1}^{k_G}$

Construct the imputed matrix $\hat{\mathbf{X}}_G$ by imputing $\{\boldsymbol{v}_i\}_{i=1}^{n_p}$ via Eq. (7)

for $j = 1, \cdots, k_G$ **do**

$\boldsymbol{h}_j = \boldsymbol{z}_j \odot \boldsymbol{m}_j + 0.5(\mathbf{1} - \boldsymbol{z}_j)$

Set $\hat{\boldsymbol{x}}_{\boldsymbol{j}} = \hat{\boldsymbol{X}}_G[j,:]$

Fixed D by computing $D(\hat{\mathbf{x}}_j, \mathbf{h}_j)$ via Eq. (9)

end for

Update G via optimizing (14)

end while

by G. Generator G aims to fool D, then G will maximize $\check{p}_k$, i.e., minimize $-\log(\check{p}_k)$. Therefore, the loss $\mathcal{L}_G$ needs to ensure that the imputed values for missing components ($m_k = 0$) successfully fool the discriminator (as defined by the minimax game), i.e.,

$$\mathcal{L}_G(\mathbf{m}, \check{\boldsymbol{p}}, \mathbf{z}) = -\sum_{k:z_k=0} (1 - m_k) \log(\check{p}_k) \tag{12}$$

In addition, we define a reconstruction loss function $\mathcal{L}_{\text{rec}}$ to ensure that the values outputted by G are close to the non-missing components. We consider the well-established cross entropy to calculate the reconstruction error for x_k, i.e., k-th observed components of $\boldsymbol{x}$:

$$\mathcal{L}_{\text{rec}}(\boldsymbol{x}, \hat{\boldsymbol{x}}, \boldsymbol{m}) = \sum_{k=1}^{d} m_k(-x_k \log(\hat{x}_k)) \tag{13}$$

We optimize the generator G using the newly updated discriminator D with mini-batches of size k_G. Regarding the loss function of the generator, we have

$$\min_G \sum_{j=1}^{k_G} \mathcal{L}_G(\mathbf{m}_j, \check{\boldsymbol{p}}_j, \mathbf{z}_j) + \alpha \mathcal{L}_{\text{rec}}(\boldsymbol{x}_j, \hat{\boldsymbol{x}}_j) \tag{14}$$

5 EXPERIMENTS

In this section, we report extensive experiments conducted on a large-scale industrial dataset to evaluate the effectiveness of the proposed data imputation methods for automated underwriting.

5.1 Dataset and Preprocessing

Before we present the comparison results of baseline imputation methods, we first analyze the characteristics of the dataset used in the experiment. The datasets in this paper are from a leading life insurance company in Australia, which consists of 10-years of longitudinal application records of 119K applicants. As can be seen in Figure 5, the response rates to questions are considerably low, most applicants fill out less than 10 percent of the entire questionnaire, due to the *information drift* of the questionnaire. This results in sparse feature vectors for the majority of applicants. Similar to the sparse feature vectors, the data exhibits the sparsity in relation to the application of *exclusions*, resulting in extreme class imbalances when predicting *exclusions*. Though there are over one thousand different *exclusions* applied in the data set, many of them are extremely rare occurred, and thus removed in experiments. Consequently, we keep the most 53 frequently *exclusions* of diseases along with 149 features condensing the applicant's medical and lifestyle information filled in the questionnaire [1].

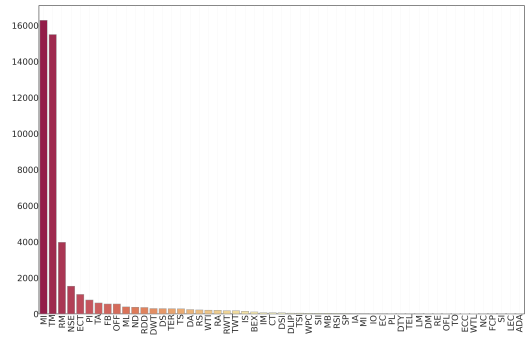


Figure 5: Histogram of records in each class (i.e. *exclusion*) in descending order.

5.2 Setup

Baseline methods. Our CaGen algorithm is compared against seven baseline methods in two categories: 1) discriminative methods including *MICE*(*Multiple Imputation by Chained Equations*) [4], missForest [20], k-nearest neighbors (KNN) [23], *Matrix Completion* [9, 13, 17], and 2) generative methods including *MIDA* (*Multiple Imputation with Denoising Autoencoders*) [8], *GAIN* (*Generative Adversarial Imputation Nets*) [26], and *MIWAE* (*Missing Data Importance-Weighted Autoencoder*) [16]. Since there is no ground-truth for missing values of underwriting data in practice, we turn to evaluate the imputation performance by the underwriting prediction accuracy on the imputed dataset. Following existing underwriting work in insurance sector, we use the Logistic Regression and Gradient Boosting Classifier as the underwriting predictors on the imputed data.

Implementation. All models are trained with 80% of the original data and the rest 20% is used to evaluate the performance of the imputation methods. For kNN, Matrix, MICE and missForest, we

[1]The data used in this research does not involve any Personal Identifiable Information (PII). A sample dataset from the real dataset will be released for research purposes.

use the implementation from the library fancy impute [2], for GAIN, MIDA, we implement with Pytorch, and implement MIWAE with Tensorflow. We implement our method using Tensorflow with the Adam optimizer. Particularly, we implement f_θ in Eq. (5) using a multilayer perceptron (MLP) with 1-dimension latent space and 128 hidden neurons. For the downstream underwriting prediction task, we use the same settings of logistic regression classifier and Gradient Boosting Classifier for all imputed methods.

Hyper-parameter tuning. We set $k = 5, 10, 20$ in kNN and rank $r = 50, 100, 1000$ in matrix completion with the maximum number of iteration of 1000. The parameters of the other compared methods were set as what were used by their authors. We perform a grid search over trade-off parameter $\alpha_1, \alpha_2 \in [10, 50, 100, 200]$, batch size $s \in [32, 64, 128, 512, 1024]$, and learning rate $\eta \in [1e-6, 1e-5, 1e-4, 1e-3, 1e-2]$, resulting in the optimal values $\alpha_1 = 500$, $\alpha_1 = 100$, $s = 124$ and $\eta = 1e-4$. The maximum epoch is set as 10000, and an early stopping strategy is performed. For the logistic regression classifier, we use the maximum epoch as 10000. For the Gradient Boosting Classifier, we use 1000 estimators with maximum depth at 5 and learning rate at 0.1.

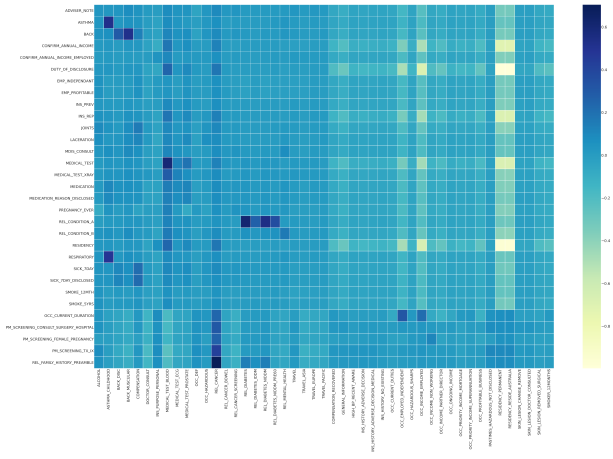
Metrics. To make a complete comparison with the previous work on imputation, we report the results of our proposed model on a variety of metrics, including macro- and micro-averaged F1, Precision, Recall and AUC (area under the ROC curve). A macro-average computes the metric independently for each class and then take the average (hence treating all classes equally), while a micro-average computes a simple average over all classes.

5.3 Results

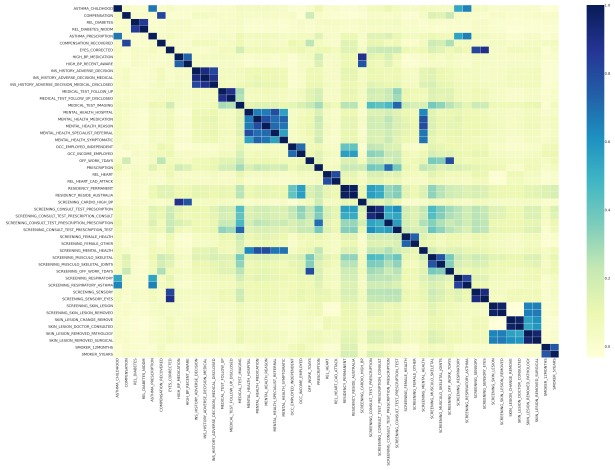
In this section, our goal is to classify the data after imputation into different correct classes in terms of *exclusions*.

5.3.1 Causality analysis. We further explore the correlations between fully observed features and partially observed features produced by Eq. (2) of causal sketelon identification. The result is shown as heatmap in Figure 6 (a) where the fully observed features with thresholds above τ in Eq. (3) are demonstrated. Each cell is colored from dark green to yellow where dark green indicating the observed feature is high relevant to the partially observed feature. For example, the heatmap shows that the fully observed **Asthma-Childhood** (2^{nd} column) is correlated to partially observed features including **Respiratory** (2^{nd} row) and **Asthma** (22^{th} row). **Asthma-Childhood** indicates that persons developing **Asthma** or **Respiratory** disease are more likely attributed to a genetic cause rather than environment causes. In other words, **Asthma-Childhood** provides proxy information to estimate confounder (i.e., **genetic cause**) that is not explicitly shown. The estimated confounder as the cause of partially features (e.g., **Asthma** and **Respiratory**) thus can be further used to impute the missing values for these partially features.

Figure 6 (b) is the heatmap generated by traditional correlation analysis, where 51 observed features that are highly correlated. This figure indicates that traditional correlation analysis tends to produce spurious relationships that lead to unstable imputation results across different distributions. For example, **Asthma-Childhood** is



(a) x-axis: partially observed features, y-axis: fully observed features.



(b) x-axis and y-axis: fully observed features.

Figure 6: Heatmaps of features.

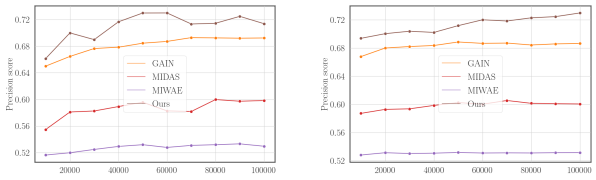
highly correlated with **Asthma-Prescription**. Traditional correlation analysis are more likely to deduce that **Asthma-Prescription** is also correlated with **Asthma** so as to impute the corresponding missing values. However, **Asthma-Prescription** is the result (not the cause/parent node) of **Asthma**, which thus can not reveal the true generating process of **Asthma** with a causal explanation. Concretely, if an applicant doesn't have **Asthma-Prescription**, we can not guarantee that this applicant has been cured of **Asthma**.

5.3.2 Underwriting prediction. We report the following performance averaged over 10 random repetitions with 5-cross validations. Table 1 reports comparison performance of all methods in terms of eight metrics. Note that our underwriting data is highly imbalanced where the distribution of records across class labels is not equal as shown in Figure 5. As can be seen from the table, the micro-average results are preferable than macro-average results for the class imbalance problem. More importantly, our method significantly outperforms each baseline on average. Our method outperforms the best baseline method (i.e. MIDA) in underwriting

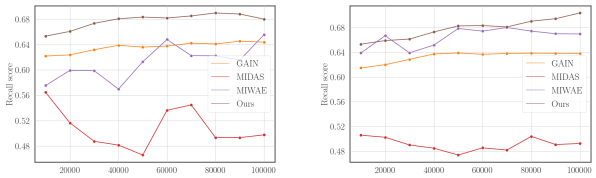
[2] https://pypi.org/project/fancyimpute/

Table 1: Underwriting prediction on the real-world dataset imputed by eight methods.

		Logistic Regression								Gradient Boosting Classifier							
		GAIN	Matrix	KNN	MICE	missforest	MIDA	MIWAE	Ours	GAIN	Matrix	KNN	MICE	missforest	MIDA	MIWAE	Ours
AUC	Marco	0.600	0.617	0.618	0.617	0.533	0.621	0.500	0.702	0.636	0.679	0.682	0.681	0.555	0.681	0.500	0.716
	Micro	0.805	0.821	0.823	0.823	0.734	0.823	0.854	0.875	0.807	0.824	0.824	0.824	0.732	0.823	0.843	0.891
F1	Marco	0.293	0.324	0.326	0.324	0.140	0.329	0.092	0.375	0.369	0.448	0.456	0.454	0.201	0.456	0.091	0.481
	Micro	0.664	0.687	0.69	0.689	0.550	0.690	0.627	0.716	0.664	0.686	0.685	0.686	0.542	0.685	0.619	0.733
Precision	Marco	0.409	0.430	0.429	0.434	0.275	0.433	0.070	0.452	0.482	0.544	0.546	0.544	0.328	0.548	0.070	0.582
	Micro	0.692	0.705	0.707	0.706	0.613	0.707	0.525	0.722	0.686	0.695	0.694	0.695	0.594	0.695	0.526	0.758
Recall	Marco	0.251	0.285	0.287	0.285	0.125	0.292	0.133	0.315	0.322	0.408	0.415	0.413	0.171	0.412	0.129	0.439
	Micro	0.638	0.67	0.674	0.673	0.499	0.673	0.778	0.814	0.644	0.677	0.677	0.678	0.498	0.675	0.753	0.786

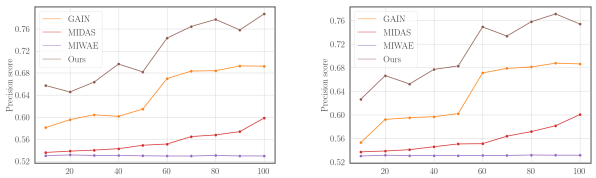


(a) Gradient Boosting Classifier. **(b) Logistic Regression.**

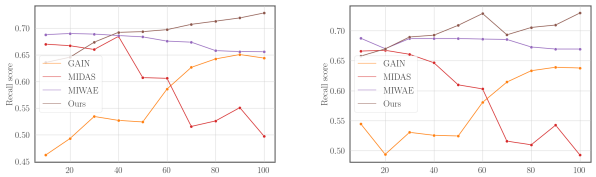


(c) Gradient Boosting Classifier. **(d) Logistic Regression.**

Figure 7: Underwriting prediction under different samples.



(a) Gradient Boosting Classifier. **(b) Logistic Regression.**

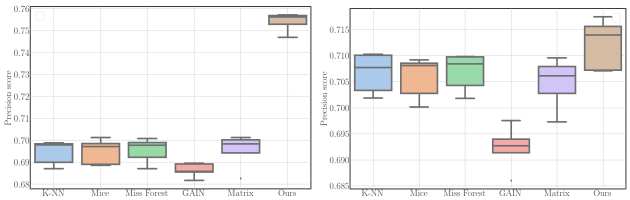


(c) Gradient Boosting Classifier. **(d) Logistic Regression.**

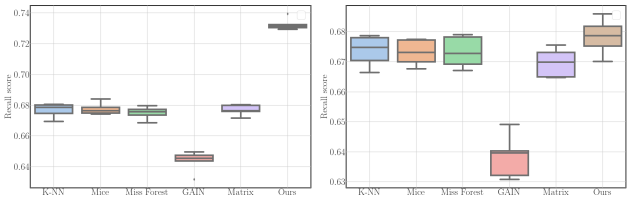
Figure 8: Underwriting prediction under different features.

prediction by 8% in terms of AUC and 10% in terms of F1. In particular, our method beats the other GAN-based method (i.e. GAIN) on all the metrics. Compared with GAN-based method, which uses an MLP as the generative model, the causality-aware generator designed in our method is able to explicitly model the observation generating process for imputing missing feature values. This further validates the effectiveness of the proposed causality-aware generator. Furthermore, MIWAE achieves the worst performance. We can also see that auto-encoder algorithms are not suitable for such data.

Robustness against different samples and features. We choose four generative methods including MIDA, GAIN and MIWAE as the baselines (with Micro settings) to present the imputation performance under different settings of samples and features. In Figure 7 and Figure 8, as expected, the performance of every model degrades when the proportion of training samples decreases (consequently discarding useful information in the observational data). Interestingly, comparing traditional generative baselines and our method, we can see that considering causal relationships will have a positive effect on the performance of data imputation, thus on the performance of underwriting.



(a) Gradient Boosting Classifier. **(b) Logistic Regression.**



(c) Gradient Boosting Classifier. **(d) Logistic Regression.**

Figure 9: Underwriting prediction on new observations.

Generalization on new observations. We further investigate the generalization capability of our method. Concretely, previous comparison results are produced by first imputing all missing values and then spliting the imputed data into training and testing. This section adopts a different setting by first spliting the original data into training and testing, and then examines whether a trained imputation model can successfully achieve superior prediction performance on new observations with missing values.

Figure 9 shows the performance of five baseline methods summarized in boxplots, excluding MIDA and MIWAE. Since the lengths of all boxes are small, to present more information of boxplots, we fixed the range of y-axis approximately at [0.63, 0.76]. This leads to the exclusion of MIDA and MIWAE, as MIDA and MIWAE perform below the average range. We can see that applying Gradient Boosting classifier on our imputed data yields average 10% higher prediction accuracy on compared with baselines without being retrained, indicating that our imputed model generalizes well to unseen observations. Interestingly, MIWAE performs nearly the best among the baselines in previous subsections, whereas performs below the average for new observations. This is mainly because test data in this subsection are new observations without imputation, it may have distributions different from the training dataset. Similarly, baseline methods merely handle correlations rather than causality, and thus have difficulties transferring the correlations knowledge in the training data to new missing data with a different distribution. In contrast, our model does not have the issues since the proposed causality-aware generator can identify causality underlying observations to learn the cause variables (i.e., confounder) of the unobserved features.

6 CONCLUSIONS AND FUTURE WORK

In this work, we tackled the learning problem of incomplete data for automated underwriting task. We find that the ignorance of causal relationships may produce bias in the generation of features and the outcome, and hence complicate the data imputation process. This motivates us to propose a causality-aware GAN framework to generate high quality missing data for underwriting task. Specifically, we first design a causal learning module to uncover the underlying causal graph resulting in missing data pattern of underwriting observations. Then, we design a causality-aware generative network for GAN to initialize the missing values following the causal relationships, and followed by an iterative process of updating imputed values via an adversarial learning. Numerical experiments with real world datasets show that our method significantly outperforms state-of-the-art imputation techniques. Future work will construct the complete causal graph on the non-stationary answers with the assistance of insurance prior knowledge among a large-scale set of variables.

ACKNOWLEDGMENTS

This work is sponsored by Australian Research Council (ARC) with No. 10.13039/501100000923.

REFERENCES

[1] Rhys Biddle, Shaowu Liu, Peter Tilocca, and Guandong Xu. 2018. Automated underwriting in life insurance: Predictions and optimisation. In *Australasian Database Conference*. Springer, 135–146.

[2] PP Bonisone, Raj Subbu, and Kareem S Aggour. 2002. Evolutionary optimization of fuzzy decision systems for automated insurance underwriting. In *2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No. 02CH37291)*, Vol. 2. IEEE, 1003–1008.

[3] Hervé Bourlard and Yves Kamp. 1988. Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics* 59, 4 (1988), 291–294.

[4] S van Buuren and Karin Groothuis-Oudshoorn. 2011. mice: Multivariate imputation by chained equations in R. *Journal of statistical software* (2011), 1–68.

[5] Hongxu Chen, Yicong Li, Xiangguo Sun, Guandong Xu, and Hongzhi Yin. 2021. Temporal meta-path guided explainable recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 1056–1064.

[6] Tri Dung Duong, Qian Li, and Guandong Xu. 2021. Stochastic Intervention for Causal Effect Estimation. *arXiv preprint arXiv:2105.12898* (2021).

[7] Pedro J García-Laencina, José-Luis Sancho-Gómez, and Aníbal R Figueiras-Vidal. 2010. Pattern classification with missing data: a review. *Neural Computing and Applications* 19, 2 (2010), 263–282.

[8] Lovedeep Gondara and Ke Wang. 2018. Mida: Multiple imputation using denoising autoencoders. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 260–272.

[9] Qian Li, Wenjia Niu, Gang Li, Yanan Cao, Jianlong Tan, and Li Guo. 2015. Lingo: linearized grassmannian optimization for nuclear norm minimization. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 801–809.

[10] Qian Li, Wenjia Niu, Gang Li, Jianlong Tan, Gang Xiong, and Li Guo. 2016. Riemannian optimization with subspace tracking for low-rank recovery. In *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 3280–3287.

[11] Qian Li, Xiangmeng Wang, and Guandong Xu. 2021. Be Causal: De-biasing Social Network Confounding in Recommendation. *arXiv preprint arXiv:2105.07775* (2021).

[12] Qian Li and Zhichao Wang. 2017. Riemannian submanifold tracking on low-rank algebraic variety. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[13] Qian Li, Zhichao Wang, Gang Li, Yanan Cao, Gang Xiong, and Li Guo. 2017. Learning robust low-rank approximation for crowdsourcing on Riemannian manifold. *Procedia Computer Science* 108 (2017), 285–294.

[14] Xueyan Liu, Bo Yang, Hechang Chen, Katarzyna Musial, Hongxu Chen, Yang Li, and Wanli Zuo. 2021. A Scalable Redefined Stochastic Blockmodel. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15, 3 (2021), 1–28.

[15] Xueyan Liu, Bo Yang, Wenzhuo Song, Katarzyna Musial, Wanli Zuo, Hongxu Chen, and Hongzhi Yin. 2021. A block-based generative model for attributed network embedding. *World Wide Web* (2021), 1–26.

[16] Pierre-Alexandre Mattei and Jes Frellsen. 2019. MIWAE: Deep generative modelling and imputation of incomplete data sets. In *International Conference on Machine Learning*. 4413–4423.

[17] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. 2010. Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research* 11 (2010), 2287–2322.

[18] Judea Pearl et al. 2009. Causal inference in statistics: An overview. *Statistics surveys* 3 (2009), 96–146.

[19] Swati Sachan, Jian-Bo Yang, Dong-Ling Xu, David Eraso Benavides, and Yang Li. 2020. An explainable AI decision-support-system to automate loan underwriting. *Expert Systems with Applications* 144 (2020), 113100.

[20] Daniel J Stekhoven and Peter Bühlmann. 2012. MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics* 28, 1 (2012), 112–118.

[21] Zhenchao Sun, Hongzhi Yin, Hongxu Chen, Tong Chen, Lizhen Cui, and Fan Yang. 2020. Disease Prediction via Graph Neural Networks. *IEEE Journal of Biomedical and Health Informatics* 25, 3 (2020), 818–826.

[22] Yi Tan and Guo-Ji Zhang. 2005. The application of machine learning algorithm in underwriting process. In *2005 International Conference on Machine Learning and Cybernetics*, Vol. 6. IEEE, 3523–3527.

[23] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. 2001. Missing value estimation methods for DNA microarrays. *Bioinformatics* 17, 6 (2001), 520–525.

[24] Guandong Xu, Tri Dung Duong, Qian Li, Shaowu Liu, and Xianzhi Wang. 2020. Causality Learning: A New Perspective for Interpretable Machine Learning. *arXiv preprint arXiv:2006.16789* (2020).

[25] Weizhong Yan and Piero P Bonissone. 2006. Designing a Neural Network Decision System for Automated Insurance Underwriting. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. IEEE, 2106–2113.

[26] Jinsung Yoon, James Jordon, and Mihaela Schaar. 2018. GAIN: Missing Data Imputation using Generative Adversarial Nets. In *International Conference on Machine Learning*. 5689–5698.