Attention GAN を用いたテーブルデータの欠測値補完

河越　淳† 董　于洋†† 野澤　拓磨†† 肖　川†

† 大阪大学大学院情報科学研究科　〒 565–0871 大阪府吹田市山田丘 1-5
†† 日本電気株式会社 データサイエンス研究所　〒 211–8666 神奈川県川崎市中原区下沼部 1753
E-mail: †{kawagoshi.jun,chuanx}@ist.osaka-u.ac.jp, ††{dongyuyang,nozawa-takuma}@nec.com

あらまし　欠測データはデータ分析において予測モデルの性能を低下させる一つの要因となっており，適切な欠測データ修正を行うことは，正しい予測結果を得るために重要である．欠測データ修正方法の一つとして，近年，GAN (Generative Adversarial Networks) を用いた修正方法が注目を浴びている．GAN を用いた修正方法では，欠測の存在する不完全データ行も学習に利用でき，既存手法よりも高い精度で欠測値補完が可能であるが，以下の 3 つの問題点を持つ．(1) ニューラルネットワークを利用することによる解釈の困難性，(2) カテゴリ値に対する生成能力の欠如 (3) 全ての列を 1 つのモデルで修正していることにより，列ごとの分布が考慮されていない点である．上記の欠点を解消するために，提案手法でははじめに，(1) 解釈性を得るために Attention を導入する．さらに，(2) カテゴリ値に対する生成能力を高めるために損失関数の変更を行う．最後に，(3) 列ごとの分布を学習させるために，モデルの作成を列ごとに行う．連続値・カテゴリ値の混在状況の異なるデータセットを用いて，既存手法との比較実験を行い，精度の向上を確認した．

キーワード　欠測データ，欠測値補完，GAN，Attention

1 はじめに

欠測データは，機械学習の研究者や実務者が直面する，値の一部が欠けてしまうというデータの問題である．データ処理時のヒューマンエラー，機器の故障によるマシンエラー，回答者の特定の質問に対する回答拒否，調査途中での被験者の脱落，データの結合など，様々な要因によって発生する [1] [2]．欠測データは，分析に用いる機械学習モデルの推定値にバイアスをもたらし，誤った結論に導く可能性がある．そのため，欠測データを適切に扱うことは必須となる．

欠測データは 3 つのタイプに分類される [3]．(1) 欠測が完全にランダムに発生する場合（MCAR），(2) 欠測が観測変数にのみ依存する場合（MAR），(3) 欠測が非観測変数に依存する場合（NMAR）である．欠測が MCAR である場合，欠測のあるサンプルを削除するという簡単な手法であっても，分析に用いる機械学習モデルにバイアスをもたらさない可能性がある [3]．しかしながら，このような厳密な MCAR の仮定は，現実のデータセットではほとんどあり得ず，欠測が発生したデータには，欠測が発生するだけのなんらかの理由があることが一般的である．そのため，欠測データを取り扱うための一般的な手法では欠測値補完に焦点を当てている．欠測値補完手法には，欠測値を標本平均（連続データの場合）や標本最頻値（離散データの場合）に置き換えるような，各データ列を他列とは独立して扱う単純な統計的アプローチから，入力データに対してモデルを学習し，推論によって欠測値補完を行う，より複雑な手法まで様々である．

一般的な，推論による欠測値補完は，識別モデルと生成モデルに分別される．識別モデルでは決定木 [4] [5]，k 近傍法 (KNN) [6]，回帰モデル (MICE) [7] などが利用されている．生成モデルでは，AutoEncoder [8]，GAN [9] などが利用されている．

これらの生成モデルは，大規模な高次元データから複雑な分布を効率的に学習し，学習に利用したデータの分布に従ったデータを生成することが可能となる [10]．そのため，近年，生成モデルを利用した欠測値補完に関する研究が盛んに行われている．

GAN を用いた手法である GAIN [9] では，生成器は欠測データとそれぞれの値が欠測か観測かを示すためのマスク行列を入力とし，欠測の予測値を出力とする．識別器は真の値と偽の値を区別することを目的とする．これにより生成器は，データの分布を学習し，高精度での欠測値補完が可能となる．しかしながら，GAIN を利用した方法では，以下の 3 つの問題点がある．(1) どのデータを利用して修正を行なったのかについて，理解し難いため，説明性が求められる際には利用ができない．(2) 損失関数として RMSE，BinaryCrossEntropy のみを利用しており，カテゴリ値に対する工夫がなされていないため，連続値やバイナリ値に対する精度が高い一方，カテゴリ値に対する精度には問題がある [11]．(3) 全ての列を 1 つのモデルで修正していることにより，列ごとの分布が考慮されていない．

上記の 3 つの問題点を解決するために，本稿では Attention GAN を用いた欠測値補完手法 (Attention GAIN) を提案する．提案手法では，列数と同数の生成器，識別器を利用し，それぞれの生成器は欠測データ行列とマスク行列を入力として，列データを出力する．生成器は，列ごとに Embedding を行い，Embedding 後のベクトルを Attention の入力とする．この Attention の重みを可視化することで，生成器がどの列を利用して，修正を行なったのかを理解することが可能となる．識別

器は欠測データ行列とマスク行列を入力として，列データを出力する．しかしながら，識別器に関しては，マスク行列自体が答えとなってしまうため，自列以外のマスク行列情報のみを利用する．損失関数は，GAIN [9] においては Vanilla GAN [12] と同様の損失関数を利用していたが，本研究では，Self-Attention GAN [13] と同様に Hinge Loss を利用し，学習の安定化のため Spectral Normalization を導入する．これにより，精度の向上が可能となる．また，簡単な方法であるが，カテゴリ値に対しては，ソフトマックス関数を利用することで，カテゴリ値に対する生成能力を向上させる．実験の結果，提案手法は既存手法と比較して，カテゴリ値，連続値ともに，より高精度に欠測値補完ができ，修正のために利用した列についての理解が可能となることを示した．

2 関 連 研 究

推論による欠測値補完手法は，一般的に識別モデルと生成モデルに分類される．

識別モデルは古くから研究されており，様々な手法が存在する [4] [14] [15]．k 近傍法を利用した手法 (KNN) [16]，連鎖方程式を利用した手法 (MICE) [7]，決定木を利用した手法 (Miss Forest，XGboost) [5] [17] が主に利用されている．MICE は欠測値補完によく利用される手法の一つであるが，欠測値が他の値と依存関係があるという強い仮定の下でのみ成り立つ手法であり，仮定が成り立たない場合には収束しないといった問題点がある．決定木を利用した手法は補間ベースの方法であるため，欠測が偏って生じている場合には，汎化性能が低下するといった問題点がある [18]．

生成モデルは近年盛んに研究をされており，データから分布を学習することで高精度の結果をもたらす．主に DAE [19] を利用したモデルと GAN を利用したモデルが存在する．

DAE を利用した欠測値補完は様々な研究が行われている [8] [20] [21] [22]．HI-VAE [8] は，この中で最も成功したモデルであり，欠測データを入力として，再構成誤差項をデータの観測部分に対してのみ行うことで，学習を可能としている．推論の際は，不完全なデータは入力前に任意の値（例えばゼロ）で埋める．この手法は，連続値とカテゴリ値の混在した高次元のデータであっても，修正を行うことが可能である．

GAN を利用した主な欠測値補完手法は，GAIN [9] である．GAIN では，生成器は欠測データとそれぞれの値が欠測か観測かを示すためのマスク行列を入力とし，欠測の予測値を出力とする．識別器は真の値と偽の値を区別することを目的とする．これにより生成器は，データの分布を学習し，高精度での欠測値補完が可能となる．その他にも，モデルを列ごとに分け，敵対的学習を利用した手法として IFGAN [23] も存在する

Wasserstein 距離を利用した WGAIN [24] も存在するが GAIN と比較して，それほど精度の向上がみられず，データセットにもよるが，GAIN の方が精度が高いことが多いため，本研究では比較手法として GAIN，IFGAN を利用する．提案手法に似た手法として，欠測データ補完に Self-Attention GAN [13] を利用した手法である SAGAIN [25] や Attention を利用した手法である AimNet [11] も存在するが，SAGAIN は Convolutional Neural Network を利用しており，テーブルデータのための手法ではなく，AimNet はテーブルデータに利用可能であるが，HoloClean [26] の補助のための手法であるため，比較手法としては利用しないこととする．

3 問 題 定 義

X を d 次元空間上の点とし，$X = (X_1,..., X_d)$ は連続値，またはカテゴリ値の確率変数であるとする．確率変数 X の分布を $P(X)$ とする．$M = (M_1,..., M_d)$ は 0 または 1 をとる確率変数であるとする．X をデータ行列，M をマスク行列と呼ぶこととする．

$i \in 1,..., d$ において，新たな空間 $\tilde{X}_i = X_i^{obs} \cup X_i^{mis}$ を定義する．$\tilde{X} = \tilde{X}_1,...,\tilde{X}_d \in X$ であり，次に従う．

$$\tilde{X}_i = \begin{cases} X_i^{obs} & M_i = 1 \\ X_i^{mis} & M_i = 0 \end{cases} \tag{1}$$

X^{obs} は観測値，X^{mis} は欠測値を表し，M は X が観測値か欠測値かを示す．また，本稿では，小文字は実現値を表す．

3.1 欠測値補完

X の実現値が $\tilde{x}^1,...,\tilde{x}^n$($n$ はサンプル数で i.i.d) である時，データセットを $D = (\tilde{x}^i, m^i)_{i=1}^n$ と定義する．本研究の目的は，各 $\tilde{x}^i$ の観測されない値を補完することである．サンプルデータの標本分布から母集団の確率分布を学習，モデル化し，データの分布に従って欠測の予測値を生成する．($\tilde{X} = \tilde{x}^i$ が与えられた際の $P(X|\tilde{X} = \tilde{x}^i)$ をモデル化する) 最適値を求めるだけではなく，データの分布をモデル化しようとすることで，複数のサンプルを取り出すことが可能となる．その結果，多重代入を行うことができ，欠測補完後のデータを用いて分析や予測を行った際に，標準誤差が過小評価されるという問題を回避することができる [3]．

4 提 案 手 法

本節では，$P(X|\tilde{X} = \tilde{x}^i)$ をモデル化するための提案手法について述べる．また，図 1 に提案手法の全体像を示す．

4.1 Generator

Generator は $\tilde{X}$，M，潜在変数 Z を入力とし，予測値 $\hat{X}_i (i \in 1,..., d)$ を出力とする．$Z = (Z_1,..., Z_d)$ は d 次元の潜在変数でそれぞれ正規分布に従う．このとき，$\hat{X}_i$，$\bar{X}_i$ を次のように定義する．

$$\hat{X}_i = G(X, M, (1 - M) \odot Z) \tag{2}$$

$$\bar{X}_i = M_i \odot X_i + (1 - M_i) \odot \hat{X}_i \tag{3}$$

$\odot$，1 はそれぞれ，アダマール積，全ての値が 1 の列ベクトルを表す．$\hat{X}_i$ は Generator の予測値である．$\hat{X}_i$ を欠測部分に代入し，補完を行う．$\bar{X}_i$ は欠測部分補完後の擬似完全データ

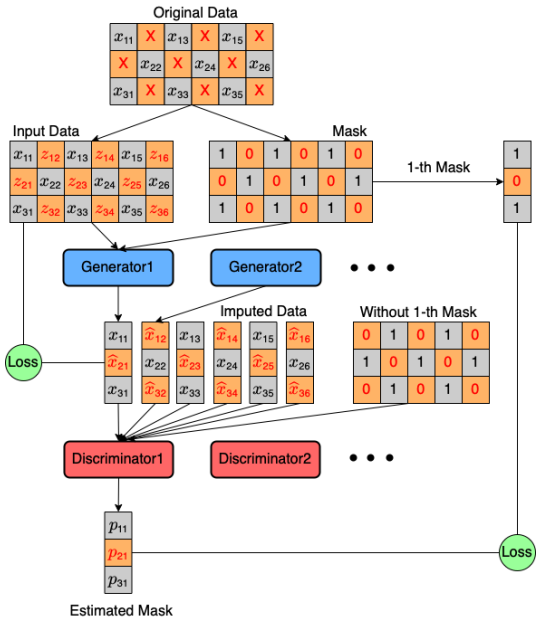

図 1　Attention GAIN のアーキテクチャ (1 列目の例)

列である.

4.2　Discriminator

Discriminator は $\bar{X}$，$M_{\tilde{i}}$($\tilde{i}$ は i 列を除く全ての列を表す) を入力とし，予測値 P_i を出力とする．Discriminator はマスク行列自体が答えとなってしまうため，i 列目のマスク列は入力から外す．このとき，P_i を次のように定義する．

$$P_i = D(\bar{X}, M_{\tilde{i}}) \tag{4}$$

P_i は入力された，i 列目の値が本物か偽物かの予測値を表す (本物であれば 1，偽物であれば 0 を出力).

4.3　損 失 関 数

Attention GAIN では，損失関数として Hinge ロスを使用する．これによって Discriminator が強くなりすぎることを防ぎ，安定した学習を行えるようにする．また，Generator は観測値に対して，連続値は GAIN と同様 Mean Squared Error Loss を使用し，カテゴリ値は Cross Entropy Loss を使用する．

$$\begin{aligned} L_{D_i} =& \mathbb{E}[M^T min(0, -1 + D(\bar{X}, M_{\tilde{i}}))] \\ & - \mathbb{E}[(1-M)^T min(0, -1 - D(\bar{X}, M_{\tilde{i}}))] \end{aligned} \tag{5}$$

$$L_{G_i}^{mis} = -\ \mathbb{E}[(1-M)^T D(\bar{X}, M_{\tilde{i}})] \tag{6}$$

$$L_{G_i}^{obs} = \begin{cases} \mathbb{E}[M^T (X_i - G(X, M, (1-M) \odot Z)^2)] & \text{連続値} \\ \mathbb{E}[M^T \sum -X_i \log G(X, M, (1-M) \odot Z)] & \text{カテゴリ値} \end{cases} \tag{7}$$

4.4　G・D モデルのアーキテクチャ

本節では，GAIN に Attention を導入するための，アーキテクチャについて説明する．

Attention [27] は，自然言語処理において，大きな貢献をした手法であり，その重みにより入力値それぞれに，どれだけ重みを付けたかを知ることができる．そのため，Attention を用いてブラックボックスであるモデルを解釈しようとする研究が盛んに行われている．提案手法では，Generator と Discriminator に Attention を利用し，その重みを確認することで，Generator が主にどの列を使用して修正を行なっているのかを理解可能にする．Attention を導入するため，本提案では Generator と Discriminator を Embbeding Layer，Attention Layer，Prediction Layer に分けて，作成した．また，図 2 に Generator モデルのアーキテクチャを示す．

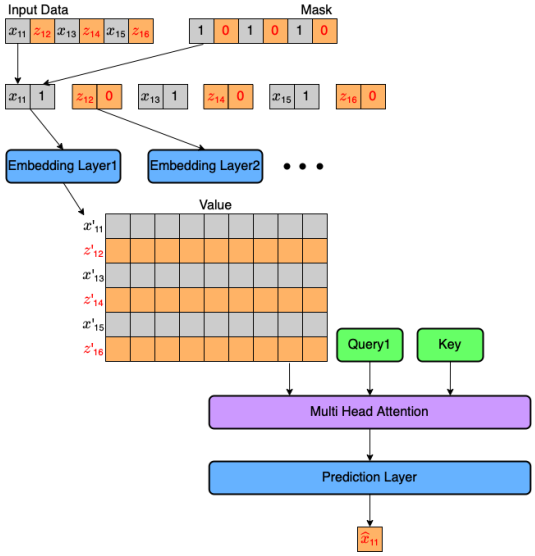

図 2　Generator のアーキテクチャ (1 列目の例)

4.4.1　Embedding Layer

Embedding Layer は Attention への入力のために，入力値を k 次元空間に変換する．Embedding Layer への入力は，連続値では実数値，カテゴリ値は One-Hot Encoding を行なった後の値とした．

3 層のニューラルネットワークで構成されており，Generator, Discriminator ともに，学習の安定化のために Spectral Normalization を適応している．i 列目の Discriminator は，j= i の時，自列である j 列目の Mask 列を Embedding Layer の入力にしない．($i, j \in 1,..., d$)

$$X_{jG_i}^{emb} = EL(X_j, M_j, (1-M_j) \odot Z_j) \tag{8}$$

$$X_{jD_i}^{emb} = \begin{cases} EL(X_j, M_j, (1-M_j) \odot Z_j) & j \neq i \\ EL(X_j, (1-M_j) \odot Z_j) & j = i \end{cases} \tag{9}$$

$$X_{G_i}^{emb} = (X_{1G_i}^{emb}, ..., X_{dG_i}^{emb}) \tag{10}$$

EL は Embedding Layer を表す．$X_{jG_i}^{emb}$ は G_i の，j 列目の入力の Embedding 後の k 次元ベクトルを表す．$X_{jD_i}^{emb}$ は D_i の，j 列目の入力の Embedding 後の k 次元ベクトルを表す．

4.4.2 Attention Layer

Attention Layer では，Multi Head Attention [27] を利用している．Multi Head Attention は，Query，Key，Value を入力とし，必要な情報に重みをつけた値を出力とする．

テーブルデータでは，i 列の値を予測するために利用する列はどのサンプルであっても同じであると考えられるため，Query と Key は k 次元ベクトルの固定値 (事前に作成しておく) とし，必要な列を利用するように，重みの学習を行う．

$$Q = K \tag{11}$$

$$V = X_{G_i}^{emb} \tag{12}$$

$$\begin{aligned} &MultiHead(Q_i, K, V) = Concat(head_1, ..., head_h)W^o \\ &\quad where \quad head_i = Attention(Q_i W_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{13}$$

Q，K は Query，Key であり，V は Value である．Value は Embedding Layer の出力とする．

4.4.3 Prediction Layer

Prediction Layer では，$MultiHead(Q_i, K, V)$ を入力とし，Generator は欠測の予測値，Discriminator は入力値が真である確率を出力する．

3 層のニューラルネットワークで構成されており，学習の安定化のため最終層以外は，Spectral Normalization を適応している．

Generator において，3 層のニューラルネットワークを通した後の出力は，連続値では 1 次元ベクトルであり，予測値をそのまま出力するが，カテゴリ値に対しては，k 次元ベクトルを出力し，ターゲットベクトルとの間の内積を計算することで類似度を測る．その後，内積に対してソフトマックスを適用し，カテゴリの各値に対する予測確率を生成し，Prediction Layer の出力とする．

ターゲットベクトルは，事前に生成しておく必要がある．

5 実　　験

本節では提案手法の有効性について，実データを用いて評価を行う．

5.1 実験設定

本実験では，既存手法との精度の比較を行う，精度の評価指標としては，連続値に対しては RMSE，カテゴリ値に対しては Accuracy を使用する，データセットには，連続値，カテゴリ値の混在したデータである，Credit データセット，連続値のみのデータである Spam データセット，カテゴリ値のみのデータである Letter データセットを利用する [28] [29] [30]．これら 3 つのデータセットは欠測のないデータであるため，欠測率 30%で MCAR の欠測を発生させ，欠測部分の実際の値をテストデータとして，実験を行なった．ミニバッチサイズは 256 で，収束が見られた 50 エポック時の結果を載せる．

$$RMSE = \sqrt{\frac{(1-M)^T(X - G(X, M, Z))^2}{\sum 1 - M}} \tag{14}$$

5.2 データセット

Credit データセットはクレジット利用可能枠，請求額，支払額といった連続値と性別，最終学歴，支払い遅延情報といったカテゴリ値の混在したデータであり，14 の連続値列と 9 のカテゴリ値列が存在する．

Spam データセットはメールがスパムかどうかを判断するための様々な情報が含まれたデータであり，57 の連続値列が存在する．

Letter データセットは英字 26 文字の大文字の構造を記述したデータであり，16 のカテゴリ値列が存在する．

5.3 実験結果

実験結果を表 1，表 2 に示す．実験の結果，提案手法は GAIN と比較して，RMSE では 0.0783 から 0.0562 の 28%，Accuracy では 0.551 から 0.743 と 34%もの精度向上が見られた．その他の手法と比較しても，提案手法の精度が高いことが確認できる．

表 1　各アルゴリズムに対する精度の比較 (RMSE)

アルゴリズム	Credit	Spam
Mean Imputation	0.0687	0.0563
MICE	0.0573	0.0572
GAIN	0.0783	**0.0554**
IFGAN	0.0679	0.0563
HI-VAE	35587.0	2.3980
Attention GAIN	**0.0562**	0.0619
Attention GAIN w/o L_D	0.0573	0.606
Attention GAIN w/o L_Obs	0.1532	0.2979

表 2　各アルゴリズムに対する精度の比較 (Accuracy)

アルゴリズム	Credit	Letter
Mean Imputation	0.534	0.220
MICE	0.662	0.269
GAIN	0.551	0.233
IFGAN	0.582	0.281
HI-VAE	0.664	0.301
Attention GAIN	**0.743**	**0.298**
Attention GAIN w/o L_D	0.742	0.296
Attention GAIN w/o L_Obs	0.433	0.028

しかしながら，Discriminator を使用せず，Generator のみで学習を行なった場合 (Attention GAIN w/o L_D) も同様の精度を達成することが可能であった．さらに，敵対的学習のみで学習を行なった際 (Attention GAIN w/o L_Obs) には RMSE が 0.1532，Accuracy が 0.433 と悪化しており，敵対的学習の恩恵を受けているとは言えない．すなわち，可視化のために Attention を利用したが，Attention が精度の向上にも貢献したのではないかと考えられる．

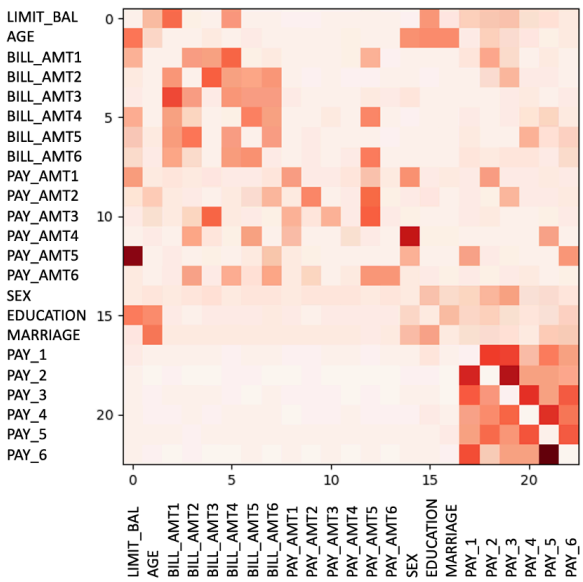

図 3　Attention の可視化 (Generator) 縦軸が Query 横軸が Key

5.4 可　視　化

提案手法における Generator の Attention の重みを可視化したものを図 3 に示す．Attention を可視化した結果，PAY_6(6ヶ月前の支払い歴，支払いが遅延したかどうかの情報) は，AGE，BILL_AMT_6(6ヶ月前の請求額)，PAY_1-5(PAY_3 を除く) を主に使用して修正を行なっていることがわかる．支払いが遅延するかどうかは，請求額や過去に支払いの遅延をしたことがあるのかに依存すると考えられるため，必要な情報を利用していると考えられる．これらの結果は，Attention を導入し，可視化を行なったことで，どの列を参照して，修正を行なっているのかについて理解することが可能となったことを示している．

6 おわりに

本報告では，欠測値補完ための Attention GAN を利用した手法を提案した．本手法を用いることで，生成モデルがどの列を利用して予測を行なっているのかを可視化することが可能となる．さらに，アーキテクチャーと損失関数を変更したことにより既存手法である GAIN よりも連続値，カテゴリ値ともに精度が向上することを示した．

今後の課題として以下に取り組みたい．本手法では，全体として GAIN と比較して精度の向上が可能となったが，提案手法において，Discriminator を使用せず，Generator のみで学習を行なった場合も同様の結果を得ることが出来た．この結果から，敵対的学習の恩恵を受けていないと考えられるが，他のデータセットにおいては Generator のみで学習を行うよりも精度が向上する可能性がある．そのため，他のデータセットでも精度の確認をしていきたい．また今回，一部の列ではモード崩壊が生じており，同じような値しか生成されないことがあった．特に，データの分布が小さい値に大きく偏っている列に対して，モード崩壊の現象が見られた．GAN の入力データ分布が偏っている際に，分布の偏りが悪化する方向に学習が進むことが原因であると考えられる [31]．従って，入力データの分布が偏っている際の生成器へのデータの入力の仕方や損失関数について検討したい．

文　献

[1] Bhavisha Suthar, Hemant M. Patel, Ankur Goswami, and M. tech. Scholar. A survey: Classification of imputation methods in data mining. 2012.

[2] Rima Houari, Ahcène Bounceur, Abdelkamel Tari, and M. Tahar Kecha. Handling missing data problems with sampling methods. *2014 International Conference on Advanced Networking Distributed Systems and Applications*, pp. 99–104, 2014.

[3] DONALD B. RUBIN. Inference and missing data. *Biometrika*, Vol. 63, No. 3, pp. 581–592, 12 1976.

[4] Antonio D'Ambrosio, Massimo Aria, and Roberta Siciliano. Accurate tree-based missing data imputation and data fusion within the statistical learning paradigm. *Journal of Classification*, Vol. 29, pp. 227–258, 2012.

[5] Daniel J. Stekhoven and Peter Bühlmann. MissForest—nonparametric missing value imputation for mixed-type data. *Bioinformatics*, Vol. 28, No. 1, pp. 112–118, 10 2011.

[6] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B. Altman. Missing value estimation methods for DNA microarrays . *Bioinformatics*, Vol. 17, No. 6, pp. 520–525, 06 2001.

[7] Stef van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, Vol. 45, No. 3, p. 1–67, 2011.

[8] Alfredo Nazábal, Pablo M. Olmos, Zoubin Ghahramani, and Isabel Valera. Handling incomplete heterogeneous data using vaes. *Pattern Recognition*, Vol. 107, p. 107501, 2020.

[9] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. Gain: Missing data imputation using generative adversarial nets, 2018.

[10] Chongxuan Li, Jun Zhu, and Bo Zhang. Max-margin deep generative models for (semi-)supervised learning, 2016.

[11] Richard Wu, Aoqian Zhang, Ihab Ilyas, and Theodoros Rekatsinas. Attention-based learning for missing data imputation in holoclean. In I. Dhillon, D. Papailiopoulos, and V. Sze, editors, *Proceedings of Machine Learning and Systems*, Vol. 2, pp. 307–325, 2020.

[12] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[13] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks, 2019.

[14] R.J.A. Little and D.B. Rubin. *Statistical analysis with missing data*. Wiley series in probability and mathematical statistics. Probability and mathematical statistics. Wiley, 2002.

[15] Alireza Farhangfar, Lukasz Kurgan, and Jennifer Dy. Impact of imputation of missing values on classification error for discrete data. *Pattern Recognition*, Vol. 41, No. 12, pp. 3692–3705, 2008.

[16] P. Jonsson and C. Wohlin. An evaluation of k-nearest neighbour imputation using likert data. In *10th International Symposium on Software Metrics, 2004. Proceedings.*, pp. 108–118, 2004.

[17] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, p. 785–794, New York, NY, USA, 2016. Association for Computing Machinery.

[18] Mikhail Belkin, Daniel Hsu, and Partha Mitra. Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate, 2018.

[19] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, p. 1096–1103, New York, NY, USA, 2008. Association for Computing Machinery.

[20] Lovedeep Gondara and Ke Wang. Mida: Multiple imputation using denoising autoencoders, 2018.

[21] John T. McCoy, Steve Kroon, and Lidia Auret. Variational autoencoders for missing data imputation with application to a simulated milling circuit. *IFAC-PapersOnLine*, Vol. 51, No. 21, pp. 141–146, 2018. 5th IFAC Workshop on Mining, Mineral and Metal Processing MMM 2018.

[22] Seunghyoung Ryu, Minsoo Kim, and Hongseok Kim. Denoising autoencoder-based missing value imputation for smart meters. *IEEE Access*, Vol. 8, pp. 40656–40666, 2020.

[23] Wei Qiu, Yangsibo Huang, and Quanzheng Li. IFGAN: missing value imputation using feature-specific generative adversarial networks. *CoRR*, Vol. abs/2012.12581, , 2020.

[24] Magda Friedjungová, Daniel Vašata, Maksym Balatsko, and Marcel Jiřina. Missing features reconstruction using a wasserstein generative adversarial imputation network. In Valeria V. Krzhizhanovskaya, Gábor Závodszky, Michael H. Lees, Jack J. Dongarra, Peter M. A. Sloot, Sérgio Brissos, and João Teixeira, editors, *Computational Science – ICCS 2020*, pp. 225–239, Cham, 2020. Springer International Publishing.

[25] Weibin Zhang, Pulin Zhang, Yinghao Yu, Xiying Li, Salvatore Antonio Biancardo, and Junyi Zhang. Missing data repairs for traffic flow with self-attention generative adversarial imputation net. *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2021.

[26] Theodoros Rekatsinas, Xu Chu, Ihab F. Ilyas, and Christopher Ré. Holoclean: Holistic data repairs with probabilistic inference. *CoRR*, Vol. abs/1702.00820, , 2017.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[28] I-Cheng Yeh and Che hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, Vol. 36, No. 2, Part 1, pp. 2473–2480, 2009.

[29] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[30] Peter W. Frey, David J. Slate, and Tom Dietterich. Letter recognition using holland-style adaptive classifiers. In *Machine Learning*, 1991.

[31] Niharika Jain, Alberto Olmo Hernandez, Sailik Sengupta, Lydia Manikonda, and Subbarao Kambhampati. Imperfect imaganation: Implications of gans exacerbating biases on facial data augmentation and snapchat selfie lenses. *CoRR*, Vol. abs/2001.09528, , 2020.