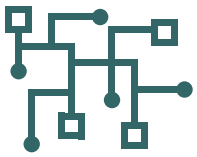

ISE

Industrial and
Systems Engineering

A Quantum-Inspired Hamiltonian Monte Carlo Method for Missing Data Imputation

DIDEM KOCHAN[1], ZHENG ZHANG[2], AND XIU YANG[1]

[1]Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA

[2]Department of Electrical Computer Engineering, University of California, Santa Barbara, CA, USA

ISE Technical Report 22T-009

LEHIGH
UNIVERSITY

A Quantum-Inspired Hamiltonian Monte Carlo Method for Missing Data Imputation

Didem Kochan*[1], Zheng Zhang†[2], and Xiu Yang ‡[3]

[1,3]Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA

[2]Department of Electrical & Computer Engineering, University of California, Santa Barbara, CA, USA

Abstract

We propose a hybrid technique combining Bayesian inference and quantum-inspired Hamiltonian Monte Carlo (QHMC) method for imputation of missing datasets. QHMC is an efficient way to sample from a broad class of distributions. Unlike the standard Hamiltonian Monte Carlo algorithm in which a particle has a fixed mass, QHMC allows a particle to have a random mass matrix with a probability distribution. Our data imputation method uses stochastic gradient optimization in QHMC to avoid calculating the full gradient on the entire dataset when evolving the Hamiltonian system. We combine the stochastic gradient QHMC and first order Langevin dynamics to obtain samples whose distribution converges to the posterior one. Comparing the performance of our method with existing imputation methods on several datasets, we show that the proposed algorithm improves the performance of data imputation in terms of accuracy and computational time.

keywords: quantum-inspired, Hamiltonian Monte Carlo, Bayesian inference, missing data.

1 Introduction

Having missing components in datasets is a problem that occurs in many empirical studies such as statistics, data mining and machine learning [12]. In practice, it is common that a large number of datasets suffer from noise, incompleteness or lack of training samples which decrease the performance of data analysis methods. In these scenarios, data augmentation and imputation techniques are usually employed to handle the missing data problem, and hence to improve the performance of computations and parameter estimations.

A considerable amount of research has been dedicated to developing missing data imputation approaches in the fields of data mining and statistics [28]. One of the simplest methods is listwise deletion, also called complete case analysis. In this approach, the latent variables are not considered and calculations are performed only through the observed variables of the dataset. Although this method can be implemented easily, it cannot represent the missing variables in the data, which may lead to failure of recognizing the characteristics of available and unavailable components [21]. Single [12] and multiple imputation techniques [4] offer to replace the latent variables with estimated values such as the mean of observed components. Those methods generate synthetic data that represent the characteristics of the original data, and they have their advantages and disadvantages depending on the type of the data and problem. They require self-efficient

*E-mail: dik318@lehigh.edu

†E-mail: zhengzhang@ece.ucsb.edu

‡E-mail: xiy518@lehigh.edu

estimations, otherwise the variance of the estimator might be inconsistent and considerably biased [24]. Hot-deck imputation is an improvement for the multiple imputation techniques, where the value of a similar case is borrowed to estimate the value of a latent variable [15, 20]. The hot-deck method aims to preserve the joint probability of observed data. Other examples of the improved imputation methods are k-nearest neighbor (kNN) [26], kernel-based imputation [28], regression-based imputation [27] and support vector machines [17]. Although they can be applicable to a broad class of cases, they all suffer from the bias issue which arises from deleting or replacing the data. Therefore, those techniques are not very well suited for especially high-dimensional datasets. Novel machine learning approaches, such as generative deep learning models (GANs) [9] are also widely used in handling missing data problems. In GANs framework, synthetic data is generated from scratch by feeding on random noise as input. Although GANs can provide realistic copies of the original data, they suffer from the problem of vanishing gradients which can make the training difficult and slow. Another issue of GANs is that the algorithm cannot guarantee convergence [25, 10].

On the other hand, Bayesian inference with a Markov Chain Monte Carlo (MCMC) approach can be an efficient approach for imputing latent variables and augment the generated data samples. A specific version of an MCMC, named Hamiltonian Monte Carlo (HMC) [5], offers a more practical approach to represent and analyze the cross-dimensional relations [18]. HMC method is a well-known powerful and efficient sampling algorithm for continuous distributions. It explores the posterior distribution using the Hamiltonian dynamics and random walk, and then generates samples for large scale datasets. In this way, it is possible to obtain multiple samples that can represent the distribution of the available data. These samples can be used for training the model and performing a learning process [22]. In particular, folded Hamiltonian Monte Carlo (FHMC) (2020) is an HMC-based method that handles missing data problems. This method uses the Hamiltonian dynamics to adapt posterior distribution, and process the cross-dimensional relations by applying a random walk procedure. It performs an HMC procedure for estimating the mean and the variance of the feature distribution. Then, another HMC (fold) is performed to obtain samples from posterior. The FHMC algorithm has been tested on high dimensional datasets and the results have shown that it can successfully impute the incomplete parts and augment the data. Although this technique provides an effective way to augment data samples and complete missing variables, it is efficient for small datasets [18].

In this paper, we propose a hybrid inference technique of Bayesian inference and quantum-inspired Hamiltonian Monte Carlo method (QHMC) [13], which is applicable to large datasets. More specifically, we implement the QHMC by stochastic optimization and first order Langevin dynamics to perform missing data imputations. We perform gradient approximations on randomly selected subsets of the dataset to avoid full gradient calculations, and use the approximated gradient information to generate parameter updates. Following the framework of Langevin dynamics, we inject noise into the parameter updates such that the parameters will converge to the posterior distribution. We apply our proposed algorithm and the original QHMC algorithm on various types of datasets: a synthetically generated Gaussian dataset, MNIST dataset and an adult income dataset which contains information about approximately $32,000$ people. The main contributions of this paper are: (i) we propose to use the QHMC method on missing data problems, (ii) we develop a modified version of QHMC to obtain the posterior distribution of the data by approximating the gradient information and impute the latent variables with that posterior distribution, and (iii) we improve the performance of data imputation using HMC-based method in terms of computational efficiency, especially for large datasets. Although the proposed algorithm is not only specific to the data imputation problems, this work is among the earliest applications of QHMC for data imputation.

1.1 Hamiltonian Monte Carlo

HMC method is a framework for sampling high dimensional continuous distributions. It introduces original and auxiliary variables to represent the movement of a particle in state space, where original variables x represent position, and auxiliary variables q represent the Gaussian momentum. In the HMC structure, the position is assumed to be independent of the momentum [2]. The method employs Hamiltonian dynamics to describe the evolution of the state (x, q). Specifically, this evolution is driven by the energy function $H(x, q)$,

and Hamiltonian equations [2]:

$$\frac{dx}{dt} = \frac{\partial H}{\partial q}, \tag{1}$$

$$\frac{dq}{dt} = -\frac{\partial H}{\partial x}. \tag{2}$$

Here, the energy of the system is $H(x, q) = U(x) + K(q)$, where U is the potential energy function and K is the kinetic energy function. The target density that we aim to sample is

$$P(x) = \frac{1}{Z} \exp(-U(x)),$$

where Z is the intractable normalizing constant. Since the HMC equations require computing derivatives, $U(x)$ needs to be differentiable. In Bayesian inference, the potential energy function U is the negative log of the posterior distribution (with prior p and log-likelihood l and a dataset X), which is defined as

$$U(x) = -\log[p(x)] - l(X|x).$$

The momentum q comes from a multivariate normal distribution with a positive-definite covariance matrix Σ, and its kinetic energy function is defined as

$$K(q) = \frac{1}{2} q^T \Sigma^{-1} q.$$

The sampling procedure of a standard HMC consists of two steps. In the first step, we move a particle along a constant energy surface which satisfies Hamiltonian dynamics given in equations (1) and (2), with a step size ϵ and number of iteration steps L. In the second step, we resample the momentum q while maintaining its position x to make a transition into another energy level. Since the leapfrog integration satisfies both volume preservation and reversibility, a standard HMC procedure uses leapfrog integration to update its state. Finally, a Metropolis-Hastings (MH) step is employed to accept or reject the proposal for (x, q), in order to correct any discretization error that might be caused by Leapfrog integrator [13, 2, 3].

1.2 Quantum Inspired Hamiltonian Monte Carlo

Although HMC is a popular framework to sample from high dimensional distributions, it has several limitations. One of these challenges is that it cannot perform well for discontiunous, non-smooth, spiky and multimodal distributions. QHMC, on the other hand, handles this problem by exploring various landscapes thanks to its time-varying mass and it can perform on such distributions [13]. Unlike the standard HMC algorithm in which a particle has a fixed mass, motivated by quantum mechanics, QHMC allows a particle to have a random mass matrix with a probability distribution.

In order to understand the quantum aspect of QHMC, we can consider a one-dimensional harmonic oscillator as an example provided in [13]. Let us suppose that we have a ball with fixed mass m attached to a spring at the origin. The stored force of the ball that pulls back the ball to the origin is $F = -kx$, where x is the displacement. In this type of a system, the ball oscillates with a time period $T = 2\pi\sqrt{\frac{m}{k}}$. In QHMC, the ball has a time-varying matrix, which means that the ball sometimes moves slowly and sometimes moves fast. This property is equivalent to having a varying time-scale, which helps explore different distribution landscapes with different time-scales. QHMC can quickly scan a broad but flat region with a small time period T, *i.e.*, small m, while it uses a larger T (or m) in a spiky region where it needs to consider every corner of the landscape [13]. Because of its ability to explore different distribution landscapes and property of satisfying HMC dynamics, QHMC is a variant of the Hamiltonian method suitable for sampling from spiky or multimodular distributions. The main idea is to set a time-varying mass matrix as a random variable associated with a probability distribution, which facilitates sampling from a spiky distribution efficiently.

Specifically, a stochastic process $M(t)$ is constructed for the mass, and at each time t, $M(t)$ is sampled from a distribution denoted as $P_M(M)$. The implementation of QHMC is straightforward, as it only adds one additional step of resampling the positive-definite mass matrix to the standard HMC algorithm [13]. Algorithm 1 presents the steps of the QHMC method with a mass-varying matrix. Note that in Algorithm 1, $P_M(M)$ is assumed to be independent of x and q. Although the choice of mass distribution is pretty flexible, in practice, simple choices can be quite useful. An example of a mass density function $P_M(M)$ with mean μ_m and variance σ_m^2 is $\log m \quad \mathsf{N}(\mu_m, \sigma_m^2)$, $M = mI$, where I is the identity matrix. After obtaining a realization of the mass distribution, QHMC approach simulates the following dynamical system:

$$d\begin{pmatrix} x \\ q \end{pmatrix} = dt \begin{pmatrix} M(t)^{-1}q \\ - \quad U(x) \end{pmatrix}.$$

In this work, we implement the QHMC method for data imputation. Here, we consider the missing variables as x in Algorithm 1. The method makes QHMC move to impute missing variables, and impute them by QHMC updates for x in Algorithm 1. To improve the efficiency of QHMC, we integrate the Stochastic Gradient Langevin Dynamics [23] (SGLD) approach with the QHMC method to avoid calculating the full gradient on the entire dataset. In this way, we can decrease the computational costs without sacrificing the accuracy of the imputations and parameter estimations, especially for large datasets.

Algorithm 1 Quantum Inspired Hamiltonian Monte Carlo (QHMC)

Input: Starting point x_0, step size ϵ, number of simulation steps L, mass distribution parameters μ_m and σ_m.

for $t = 1, 2, ...$ **do**
  Resample $M_t \quad P_M(M)$
  Resample $q_t \quad N(0, M_t)$
  $(x_0, q_0) = (x^{(t)}, q^{(t)})$
  $q_0 \leftarrow q_0 - \frac{\epsilon}{2} \quad U(x_0)$
  for $i = 1, 2, ..., L - 1$ **do**
    $x_i \leftarrow x_{i-1} + \epsilon M_t^{-1} q_{i-1}$
    $q_i \leftarrow q_{i-1} - \frac{\epsilon}{2} \quad U(x_i)$
  end for
  $x_L \leftarrow x_{L-1} + \epsilon M_t^{-1} q_{L-1}$
  $q_L \leftarrow q_{L-1} - \frac{\epsilon}{2} \quad U(x_L)$
  $(\hat{x}, \hat{q}) = (x_L, q_L)$
  MH step: $u \quad$ Uniform$[0, 1]$;
  $\rho = e^{-H(\hat{x},\hat{q}) + H(x^{(t)}, q^{(t)})}$;
  if $u < \min(1, \rho)$ **then**
    $(x^{(t+1)}, q^{(t+1)}) = (\hat{x}, \hat{q})$
  else
    $(x^{(t+1)}, q^{(t+1)} = (x^{(t)}, q^{(t)})$
  end if
end for
Output: $\{x^{(1)}, x^{(2)}, ...\}$

2 Methods and Technical Solutions

Similar to the original version of HMC [2], the original QHMC algorithm requires the full gradient computation over each training sample. Hence, it can be costly when the training dataset is large. Stochastic gradient HMC [5] approach is an efficient method that builds on the HMC framework using stochastic gradient approximations to avoid the cost of full gradient calculation and introducing the second-order Langevin

dynamics to ensure convergence. Quantum Stochastic Gradient Nosé-Hoover Thermostat (QSGNHT) introduced in [13] approximates the true gradient with only a small batch of samples. The algorithm also considers the extra noise term arising from mini batch estimation and handles it by using the thermostat technique. Inspired by the stochastic HMC and QSGNHT, we propose a stochastic version of QHMC, in which we use the first order Langevin dynamics [23], rather than the thermostat technique. We aim at combining SGLD framework with QHMC and adapt the resulting technique for missing data environments. Our method combines the efficiency of QHMC in state space exploration with the computational efficiencies of SGLD. It would serve as a Bayesian sampling algorithm that can successfully and rapidly approximate the posterior for large-scale datasets.

2.1 Bayesian Framework

Bayesian inference, or Bayesian parameter estimation uses the initial distribution called prior distribution and the likelihood to estimate the posterior. The key idea is to update the prior knowledge using observation data to make the predictions more dependable [6]. In Bayesian literature, the distribution of interest is the conditional distribution of the unknown variable given the available variable y, denoted by $\pi(x) = p(x|y)$. Let x denote the unknown variable and y denote the observed variable. Then, the conditional distribution is defined as [1] $p(x|y) = \frac{p(x)p(y|x)}{p(y)}$, where $p(x|y)$ stands for the posterior probability density, $p(x)$ is the prior probability density function and $p(y|x)$ is the likelihood function. In Bayesian approach, $p(y)$ is assumed to be independent of x, the variable of interest. Hence, we can neglect $p(y)$ and rewrite the posterior probability as

$$p(x|y) \quad p(x)p(y|x), \tag{3}$$

meaning that the posterior distribution is proportional to the prior distribution and the likelihood function [8], [16].

In this work, we use this Bayesian framework for missing data analysis. Assume that $X = (X_{\text{miss}}, X_{\text{obs}})$ is the data matrix with incomplete parts, Y is the output vector, and θ is the parameter vector of a given model that describes the relation between X and Y. Our goal is to impute the missing variables X_{miss} in X, and estimate the parameter vector θ of the model. The conditional probability of missing variables can be written as

$$\begin{aligned}\pi(\theta, X_{\text{miss}}) &= p(\theta, X_{\text{miss}}|X_{\text{obs}}, Y) \\ &\quad p(\theta, X_{\text{miss}}, X_{\text{obs}}, Y) \\ &= p(\theta)\prod_{i=1}^{n} p(x_{\text{miss},i}, x_{\text{obs},i}|\theta) \times p(y_i|x_{\text{miss},i}, x_{\text{obs},i}, \theta).\end{aligned}$$

Under the assumption that X and θ are independent, we have

$$\begin{aligned}p(x_{\text{miss},i}, x_{\text{obs},i}|\theta) &= p(x_{\text{miss},i}, x_{\text{obs},i}) = p(x_i), \quad \text{and} \\ p(y_i|x_{\text{miss},i}, x_{\text{obs},i}, \theta) &= p(y_i|x_i, \theta).\end{aligned}$$

The imputation flow starts with this Bayesian learning and continues with an MCMC procedure for drawing samples generated by the Bayesian framework. A general imputation procedure at step t can be written as

$$\begin{aligned}X_{\text{miss}}^{(t+1)} \quad & \pi(X_{\text{miss}}|\theta) = p(X_{\text{miss}}|X_{\text{obs}}, \theta^{(t)}, Y) \\ \theta^{(t+1)} \quad & \pi(\theta|X_{\text{miss}}) = p(\theta|X_{\text{obs}}, X_{\text{miss}}^{(t+1)}, Y).\end{aligned}$$

More specifically, in SGLD-QHMC, the missing variables (*i.e.*, X_{miss}) in a randomly selected subset of the data are replaced by performing QHMC updates iteratively. QHMC updates first generate proposal values for missing variables, and after the MH step, the incomplete data is updated.

2.2 Stochastic Gradient Langevin Dynamics

SGLD algorithm is proposed by Welling and Teh (2011) [23] as an iterative sub-sampling based technique for Bayesian learning from large-scale datasets. SGLD is a combination of stochastic optimization in which gradient approximations are performed over small mini-batches and Langevin dynamics in which the gradient information generated by the first part is used to update unknown parameters of the model. Moreover, first order Langevin dynamics injects a noise parameter to the updates that ensures the updated parameters converge to the samples of full posterior distribution. The stochastic optimization part is responsible for providing an optimized approximation to the Markov chain. A standard SGLD sampler switches between stochastic optimization and a Bayesian inference method [23].

Let θ be the vector of parameters of a given learning model, and $X = (x_1, x_2, ..., x_n)$ be the random variable representing the training data. The posterior distribution of θ can be expressed as $p(\theta|X) \approx p(\theta)\prod_{i=1}^{n} p(x_i|\theta)$. Let $X^{(t)} = \{x_1^{(t)}, x_2^{(t)}, ..., x_m^{(t)}\}$ denote a subset of X with size m at step t. We can apply the stochastic optimization with the following updates [19]:

$$\theta^{(t+1)} = \theta^{(t)} + \Delta\theta^{(t)},$$
$$\Delta\theta^{(t)} = \frac{\epsilon^{(t)}}{2}\left(\log p(\theta^{(t)}) + \frac{n}{m}\sum_{i=1}^{m} \log p(x_i^{(t)}|\theta^{(t)}) \right),$$

where $\epsilon^{(t)}$ is the step size satisfying the following conditions to ensure convergence [23]:

$$\sum_{t=1}^{\infty} \epsilon^{(t)} = \infty, \quad \sum_{t=1}^{\infty} (\epsilon^{(t)})^2 < \infty. \tag{4}$$

After obtaining the estimate for $\Delta\theta^{(t)}$, we use Langevin dynamics to handle the uncertainty and data over-fitting problems by adding a Gaussian noise term $\eta^{(t)}$. This step finalizes the SGLD update as

$$\Delta\theta^{(t)} = \frac{\epsilon^{(t)}}{2}\left(\log p(\theta^{(t)}) + \frac{n}{m}\sum_{i=1}^{m} \log p(x_i^{(t)}|\theta^{(t)}) \right) + \eta^{(t)}, \tag{5}$$

with $\eta^{(t)} \quad \mathcal{N}(0, \epsilon^{(t)})$ and the step-sizes ϵ satisfy the conditions in (4). Typically, step-sizes can be set by the given formula $\epsilon^{(t)} = a(b+t)^{-\gamma}$ with $\gamma = 0.55$, and a and b are set such that $\epsilon^{(t)}$ decays from 0.01 to 0.0001 as the iteration number increases [23].

2.3 Proposed Method

In this SGLD version of the QHMC method, we select a random subset of the data at every iteration, and approximate the gradient over the selected subset. We update the missing variables using the gradient information and injecting a noise term. Adding Gaussian noise with an appropriate magnitude, which is balanced with the step size, can provide updates that will converge to the posterior distribution. Suppose we have n training samples, then the potential energy function in QHMC can be written as

$$U(x) = \frac{1}{n}\sum_{i=1}^{n} U_i(x),$$

where each $U_i(x)$ depends only on the i^{th} sample. We can perform the stochastic estimation of the gradient by the following:

$$\hat{U}(x) = \frac{1}{m}\sum_{i=1}^{m} \quad U_i(x),$$

where m is the batch size. We employ the Bayesian approximation for the gradient estimations, and perform the QHMC update for position x and momentum q as

$$x^{(t+1)} = x^{(t)} + \epsilon^{(t)} M_t^{-1} q^{(t)}, \tag{6}$$

$$q^{(t+1)} = q^{(t)} + \frac{\epsilon^{(t)}}{2} \left(\quad \log p(x^{(t)}) + \frac{n}{m} \sum_{i=1}^{m} \quad \log p(x_i^{(t)} | q^{(t)}) \right) + \eta^{(t)}, \tag{7}$$

where, by definition of HMC dynamics $p(x) = \exp(-U(x))$, and $\eta^{(t)} \quad N(0, \epsilon^{(t)})$ again is the noise term with step-size $\epsilon^{(t)}$. We summarize the SGLD-QHMC algorithm in Algorithm 2. Although the SGLD-QHMC algorithm can be useful for various problems and applications as the original QHMC, this paper will focus on the missing data application of the method. The updates in 6 and 7 are used for imputation of latent variables, which are the positions of x in this notation. Our method performs Bayesian learning combined with a stochastic version of QHMC to impute missing variables, and it differs from the existing missing data imputation algorithms in the literature, which delete or replace the data [28, 27, 17], or generate synthetic data from scratch [9], or perform full gradient calculations to simulate HMC dynamics which is efficient for only small datasets [18].

Algorithm 2 Stochastic Gradient Langevin Dynamics QHMC (SGLD-QHMC)

Input: Starting point x_0, step size ϵ, number of simulation steps L, mass distribution parameters μ_m and σ_m, subset size m.

for $t = 1, 2, ...$ **do**
  Resample $M_t \quad P_M(M)$;
  Resample $q^{(t)} \quad N(0, M_t)$;
  Select a random subset x such that the size of x is m;
  $(x_0, q_0) = (x^{(t)}, q^{(t)})$
  Approximate the gradient $\tilde{U}(x)$ by
  $\left(\quad \log p(x) + \frac{n}{m} \sum_{j=1}^{m} \quad \log p(x_j | q) \right)$
  $q_0 \leftarrow q_0 - \frac{\epsilon}{2} \quad \tilde{U}(x_0)$
  for $i = 1, 2, ..., L - 1$ **do**
    $x_i \leftarrow x_i + \epsilon M_t^{-1} q_{i-1}$

    $q_i \leftarrow q_{i-1} + \frac{\epsilon^{(t)}}{2} \quad \tilde{U}(x_i) + \eta^{(t)}$
  end for
  $x_L \leftarrow x_{L-1} + \epsilon M_t^{-1} q_{L-1}$
  $q_L \leftarrow q_{L-1} - \frac{\epsilon^{(t)}}{2} \quad \tilde{U}(x_L)$
  $(\hat{x}, \hat{q}) = (x_L, q_L)$
  MH step: un $\quad$ Uniform$[0, 1]$;
  $\rho = e^{-H(\hat{x}, \hat{q}) + H(x^{(t)}, q^{(t)})}$;
  if un $< \min(1, \rho)$ **then**
    $(x^{(t+1)}, q^{(t+1)}) = (\hat{x}, \hat{q})$
  else
    $(x^{(t+1)}, q^{(t+1)}) = (x^{(t)}, q^{(t)})$
  end if
end for
Output: $\{x^{(1)}, x^{(2)}, ...\}$

2.4 Theoretical Analysis of the Method

In this section, we will show that SGLD-QHMC generates the true posterior $p(x)$, via stochastic differential equations (SDE). We can define Langevin dynamics for a Hamiltonian system with diffusion factor A by the following SDE [7]

$$dx = qdt, \quad d(q) = - \quad U(x)dt - Aqdt + \sqrt{2A}dW, \tag{8}$$

where W is the Wiener process, and dW can be simply expressed as $N(0, dt)$. By rescaling time $t \leftarrow At$ and letting $A \rightarrow \infty$ we obtain Brownian dynamics

$$d(q) = - \quad U(x)dt + N(0, 2dt).$$

In SGLD, we use gradient approximations $\hat{U}(x)$ and perform updates combining stochastic gradient and Langevin dynamics with step-size $\epsilon^{(t)}$. Hence, the SGLD-QHMC implementation simulates the following system:

$$d\begin{pmatrix} x \\ q \end{pmatrix} = dt \begin{pmatrix} M(t)^{-1}q \\ - \quad \hat{U}(x) + N(0, 2D) \end{pmatrix}. \tag{9}$$

In order to show that the system described by Equation (9) has a unique and steady distribution $p(x)$ $\exp(-U(x))$, we will first review a result for a general continuous-time Markov process provided in [14]. A continuous Markov process can be expressed as an SDE:

$$dz = f(z)dt + \sqrt{2D(z)}dW(t), \tag{10}$$

where z represents a general vector, $f(z)$ is deterministic drift and $D(z)$ is the magnitude of the Wiener diffusion process. Then, $p_s(x)$ $\exp(-H(x))$ is a steady distribution of the dynamics in Equation (10), if $f(z)$ is restricted to the following form:

$$f(z) = -[D(z) + Q(z)] \quad H(z) + \Gamma(z), \quad \Gamma_i(z) = \sum_{j=1}^{d} \frac{\partial}{\partial z_j}(D_{ij}(z) + Q_{ij}(z)), \tag{11}$$

where $H(z) = U(x) + \frac{1}{2}q^T M(t)^{-1}q$ is the Hamiltonian of the system, $Q(z)$ is a skew-symmetric matrix and $D(z)$ is a positive semi-definite matrix. In practice, Equation (11) can be modified for the stochastic gradient variant of the sampler as [14]

$$f(z) = -\epsilon^{(t)}[D(z_t) + Q(z_t)] \quad \hat{H}(z_t) + \Gamma(z_t)] + N(0, \epsilon^{(t)}2D(z_t) - \epsilon^{(t)}B_t), \tag{12}$$

where B_t is the estimate of the variance of additional stochastic noise satisfying $2D(z_t) - \epsilon^{(t)}B_t \succeq 0$, *i.e.* positive semi-definite. The SGLD-QHMC system described by Equation (9) with decaying step-sizes ϵ satisfy the conditions in 4, but with a constant mass $M(t) = M$ has a unique and steady distribution which is proportional to $\exp(-U(x))$. The SGLD-QHMC system has the following update for q:

$$q^{(t+1)} = q^{(t)} + \epsilon^{(t)}D \quad \hat{U}(x^{(t)}) + N(0, 2\epsilon^{(t)}D),$$

which fits into the framework provided in Equations (10) and (12) by replacing

$$\begin{aligned} H(z) &= H(x, q) = U(x) + \frac{1}{2}q^T M^{-1}q, \\ Q(z) &= Q(x, q) = 0, \\ D(z) &= D(x, q) = D \text{ (with } B_t = 0). \end{aligned}$$

Now, we can show that the SGLD-QHMC framework with changing mass can provide a correct steady-state distribution that describes the aimed posterior distribution $p(x) \quad \exp(-U(x))$ applying Bayes rule. The joint probability density of (x, q, M) is $p(x, q, M) = p(x, q|M)P_M(M)$. We know from Lemma 2.4 that we have

$$p(x, q|M) \quad \exp\left(-U(x) - K(q)\right) = \exp\left(-U(x)\right) \exp\left(-\frac{1}{2} q^T M^{-1} q\right),$$

from which we can obtain the following:

$$p(x) = \int_q \int_M dq dM p(x, q, M) \quad \exp(-U(x)),$$

which shows that the marginal steady distribution approaches the true posterior distribution [13].

2.5 Discretization Error

We show that the difference between the exact solution and numerical solution generated by SGLD-QHMC updates is bounded. The SDE form of the exact solution can be written as in equation (8)

$$dq_t = - \quad U(x_t) + \Sigma_t dW_t, \tag{13}$$

where $U(x)$ is the potential energy function of the Hamiltonian system, W_t is the Wiener process and Σ_t is a diagonal covariance matrix. Then, the SGLD solution of the system is [11]

$$\hat{q}_{t+1} = \hat{q}_t - \epsilon^{(t)} \quad \hat{U}(x)dt + \sqrt{\epsilon^{(t)}}\hat{\Sigma}\xi_t,$$

where $\epsilon^{(t)}$ are stepsizes satisfying the conditions in 4, and ξ_t is the standard Gaussian distribution. Now let us define the semi-discretized solution $\{\hat{q}_t\}_{t \geq 0}$ as the linear interpolation of exact solutions $\{q_t\}_{t \geq 0}$ by integrating the following:

$$\hat{q}_t = \hat{q}_0 - \int_0^t \quad \hat{U}(x_s)ds + \int_0^t \hat{\Sigma} dW_s. \tag{14}$$

Theorem 1 *Let us define a function Φ_t^2 to quantify the error caused by the gradient approximation at iteration t, such that $\hat{U}(x_t) = \quad U(x_t) + \Phi_t^2$. Then, under the assumptions of dissipativity and smoothness of potential energy function $U(x)$ and having step-sizes satisfying conditions in 4, there exist constants C_1 and C_2 such that*

$$\mathsf{E}[||q_t - \hat{q}_t||^2] \leq C_1 \mathsf{E}[||\Phi_t^2||^2] + C_2 \epsilon^{(t)}. \tag{15}$$

By integrating (13) and subtracting (14) we obtain

$$q_t - \hat{q}_t = - \int_0^t \Big(\quad \hat{U}(x_s) - \quad U(x_s) \Big) ds + \int_0^t \left(\Sigma_s - \hat{\Sigma}_s \right) dW_s, \tag{16}$$

for any $0 \leq t \leq T$. Now, applying Cauchy-Schwarz inequality and then taking the expectation yield

$$\mathsf{E}[||q_t - \hat{q}_t||^2] \leq 2\mathsf{E}\left[\left|\left| \int_0^t (\quad \hat{U}(x_s) - \quad U(x_s)) ds \right|\right|^2 \right] + 2\mathsf{E}\left[\left|\left| \int_0^t (\Sigma_s - \hat{\Sigma}_s) ds \right|\right|^2 \right], \tag{17}$$

We will find upper bounds for the two terms on the right hand side of the inequality. Applying Cauchy-Schwarz inequality to the first and second terms we have

$$\mathsf{E}\left[\left|\left| \int_0^t \quad \hat{U}(x_s) - \quad U(x_s) ds \right|\right|^2 \right] \leq t\mathsf{E}\left[\int_0^t || \quad \hat{U}(x_s) - \quad U(x_s)||^2 ds \right], \tag{18}$$

$$\mathsf{E}\left[\left|\left| \int_0^t (\Sigma_s - \hat{\Sigma}_s) ds \right|\right|^2 \right] \leq t\mathsf{E}\left[\int_0^t ||(\Sigma_s - \hat{\Sigma}_s)||^2 ds \right], \tag{19}$$

where following the Burkholder-Davis-Gundy inequalities we can estimate the terms in (18) *and* (19)*. Combining the upper bounds for the terms and convergence property of step-size* $\epsilon^{(t)}$*, the theorem can be proved.*

3 Empirical Evaluation

We implemented our SGLD-QHMC algorithm and other baseline methods, namely, kNN, MH and FHMC on several datasets. We also included the performance of the original QHMC to compare it with SGLD-QHMC. In all of the experiments conducted with SGLD-QHMC, we randomly selected 40% of the data as the subset for SGLD. The result reported in each experiment is the averaged value of 100 independent trials. We started with a normally distributed dataset masked intentionally, and continued with MNIST dataset, which is a well-known benchmark dataset in machine learning, again masked intentionally. For these datasets, the performance of the different algorithms is measured by the difference between the predicted results and original ones. We were able to do this since we masked the synthetic datasets intentionally and have the information about the original datasets. We defined a distance metric as in [18] to evaluate the accuracy of the algorithms, and we included the execution times to compare computational efficiency. In the last experiments, we tested the algorithm on an adult income dataset which predicts if the annual income of an individual is greater than $\$50,000$ based on features such as education level, occupation and marital status. The dataset contains *missing features* in it, but has the binary result for every person. We imputed the dataset and used the generated data for predicting the income levels. We evaluated the performance by comparing estimated binary results with original ones.

3.1 Numerical Results for Synthetic Dataset

We generated a data matrix X to test the proposed approach on a normally distributed dataset with latent variables. The prior parameters, mean and covariance of the data are provided. Then another binary matrix A with the same size as the data matrix is generated in order to mask some variables in the dataset. Randomly located zero elements in A represent the latent variables in X, while one represents the observed variables. The sparsity of the mask matrix can be adjusted so that we can control the portion of incompleteness in the dataset. Thus, we have the following problem setup:

- Data matrix $X_{n\times d}$, where $n = 500,000$ and $d = 10$,
- Mask matrix $A_{n\times d}$, where $n = 500,000$ and $d = 10$,
- Observed data $X \circ A$,

where $\circ$ denotes the Hadamard product. We compare the imputation performances of MH, kNN, FHMC, QHMC and SGLD-QHMC algorithms according to the normalized root mean squared error (NRMSE) as distance metric defined as [18]

$$\text{NRMSE} = \sqrt{\text{mean}((X - X_{\text{est}})^2)/\text{var}(X)},$$

where X_{est} stands for the imputed version of X. Figure 1 presents the NRMSE values obtained by different algorithms with respect to the ratio of missing components. We can see that the kNN and MH algorithms yield higher NRMSE values, indicating that the distance between imputed and original variables are bigger than the distances obtained by FHMC, QHMC and SGLD-QHMC. When it comes to comparing FHMC and QHMC-based algorithms, we can observe that their prediction successes are close to each other. However, the execution times of those three algorithms are different as shown in Table 1 which lists the NRMSE values obtained by kNN, MH, FHMC, QHMC and SGLD-QHMC algorithms as the ratio of missing variables increases. Although we have similar success rates for the algorithms with QHMC attaining the highest for most of the cases, we can see a remarkable difference between the execution times. Thanks to the

subsampling and gradient approximation performed by SGLD-QHMC, we can save a significant amount of time with slightly sacrificing (or without sacrificing) the accuracy of imputations. In this test case, the execution time of SGLD-QHMC is about 30% shorter than that of QHMC, 40% shorter than that of FHMC and kNN, and more than 70% shorter than that of MH.

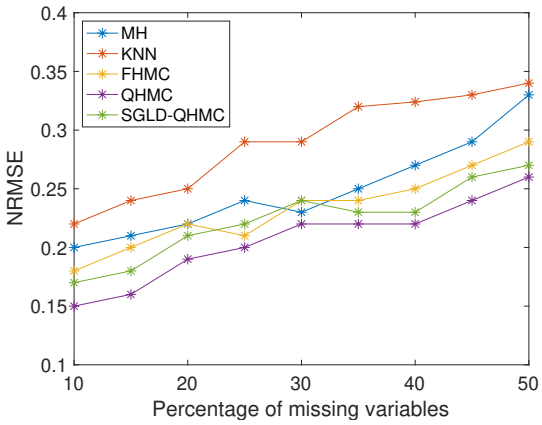

Figure 1: Comparison of algorithms on Gaussian dataset.

Table 1: Time comparison of the algorithms with their NRMSE values on Gaussian dataset.

Missing rate	Metric	FHMC	QHMC	SGLD-QHMC	kNN	MH
10%	NRMSE	0.18	0.15	0.17	0.22	0.20
	Time(sec)	55.3	49.2	**31.4**	60.7	175.6
20%	NRMSE	0.22	0.19	0.21	0.25	0.22
	Time(sec)	98.5	87.3	**64.3**	130.8	353.7
30%	NRMSE	0.24	0.22	0.24	0.28	0.23
	Time(sec)	163.1	144.6	**94.0**	175.2	631.6
40%	NRMSE	0.25	0.22	0.23	0.32	0.26
	Time(sec)	349.6	303.2	**198.7**	326.7	911.2

3.2 Numerical Results for MNIST dataset

In this section, we conduct a set of experiments with the MNIST dataset, which contains 60,000 images of handwritten digits with 28 × 28 pixels. We convert the data into a matrix of size $60,000 \times 784$, assuming that each pixel is a feature dimension. We mask the feature values using a random mask matrix, and obtain a missing dataset. Figure 2 shows the MNIST images reconstructed by SGLD-QHMC algorithm after randomly masking 20% of the pixels. We also compared quantitatively the imputation performances of kNN, MH, FHMC, QHMC and SGLD-QHMC algorithms in terms of the NRMSE. The comparisons of algorithms as the ratio of missing variables changes are shown in Figure 3 and Table 2. According to the results, we can clearly see that MH and kNN algorithms are outperformed by FHMC, QHMC and SGLD-QHMC algorithms. MH gives the highest NRMSE values with a longer execution time, while kNN gives the second highest NRMSE values with a shorter execution time than MH and FHMC. When it comes to comparing three HMC-based algorithms, although they yield similar prediction successes, QHMC and SGLD-QHMC provide the most accurate results. Moreover, SGLD-QHMC attains these NRMSE values in a shorter time than FHMC and QHMC, and the time saving is around 30%.

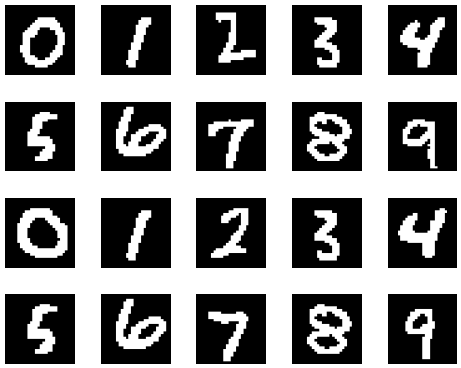

Figure 2: Image reconstruction on MNIST dataset using the SGLD-QHMC for data imputation after randomly dropping 20% pixels as missing values. Upper rows are ground truth, bottom rows are imputed images by SGLD-QHMC.

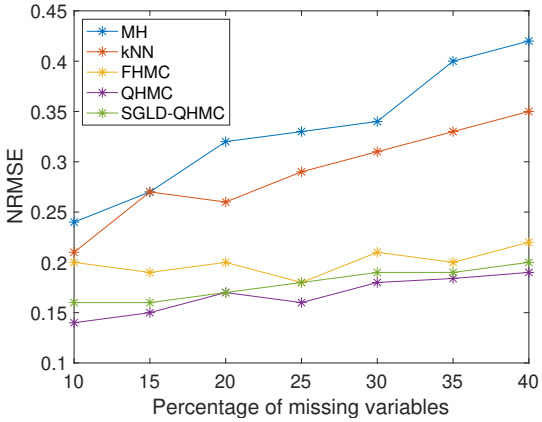

Figure 3: Comparison of the algorithms on the MNIST dataset.

3.3 Numerical Results for Adult Dataset

We analyze a real adult dataset that predicts whether the annual income level of a person exceeds $\$50K$ based on personal details such as education level, sex, marital status, current occupation and native country. The dataset contains the information of $32,561$ people for 5 different categories, and each row of the data matrix contains one person's features for these categories. Corresponding binary labels represent that the income of an adult exceeds $\$50K$ per year if it is one, and zero otherwise. There are approximately 10% missing features in the rows and we indicate them with the response indicator matrix A. The categorical data is fit with the logistic regression model so that we make our derivations using the logistic regression function. Similar to the synthetic data example, $X = (X_{\text{miss}}, X_{\text{obs}})$ denotes $n \times d$ data matrix with $n = 32,561$ and $d = 5$, Y denotes the corresponding binary outcome, and $\theta = (\theta_1, \theta_2, ..., \theta_d)^{\text{T}}$ is the parameter vector for the logistic regression model. We compared the performances of kNN, MH, FHMC, QHMC and SGLD-QHMC algorithms in terms of prediction accuracy. The steps for calculating the prediction accuracy are given in the following:

- The dataset is partitioned into training and test sets.

Table 2: Time comparison of the algorithms with NRMSE values on MNIST dataset.

Missing rate	Metric	FHMC	QHMC	SGLD-QHMC	kNN	MH
10%	NRMSE	0.20	0.14	0.16	0.21	0.24
	Time(sec)	1776.3	1608.1	**1280.4**	1540.3	2650.4
20%	NRMSE	0.20	0.17	0.17	0.26	0.32
	Time(sec)	2270.1	2054.2	**1502.0**	1873.4	2883.2
30%	NRMSE	0.21	0.18	0.19	0.31	0.34
	Time(sec)	2912.6	2690.0	**1978.5**	2577.1	3126.2
40%	NRMSE	0.22	0.19	0.20	0.35	0.42
	Time(sec)	3211.3	2989.3	**2245.7**	2910.2	3469.4

- The missing parts in the training set are imputed by the algorithms, and parameter estimations are performed.
- Using the estimated parameter vector θ and variables in the test set, the expectation of likelihood function to the observed variables is calculated by

$$\mathsf{E}_p(y_i) \approx \frac{1}{N}\sum_{i=1}^{N}\frac{1}{1+e^{-x_i\theta^{(i)}}},$$

  where N is the number of iterations after burn-in period, and $\mathsf{E}_p(y_i)$ is the prediction probability that determines the predicted value for y_i, the binary result of the logistic regression function.
- Predicted value is assigned to one, if the prediction probability is greater than 0.5, and assigned to zero otherwise.
- The prediction accuracy is calculated by taking the ratio of the number of matching results to all actual results in the test set.

Figure 4 shows the prediction accuracy for these algorithms with respect to the number of iterations. It illustrates that the obtained prediction success rates are around 80% for HMC-based methods with relatively fewer iterations, while kNN and MH require a higher number of iterations to attain that success rate. Moreover, QHMC and SGLD-QHMC can attain 90% prediction success as the number of iterations increases, while other methods stay around 80%. Table 3 shows the execution time of the algorithms. We can see that SGLD-QHMC is more tolerant to the number of iterations than the original QHMC, and it provides higher accuracy than FHMC, kNN and MH within a shorter amount of time.

Table 3: Time and prediction accuracy (PA) comparison of algorithms according to number of iterations on adult dataset.

Number of Iterations	Metric	FHMC	QHMC	SGLD-QHMC	kNN	MH
1×10^5	PA	0.78	0.80	0.79	0.73	0.75
	Time(sec)	475.0	325.3	**247.6**	372.0	613.3
2×10^5	PA	0.81	0.85	0.82	0.75	0.79
	Time(sec)	1102.2	918.3	**500.3**	945.3	1713.7
5×10^5	PA	0.84	0.91	0.91	0.82	0.82
	Time(sec)	3176.9	2956.2	**1283.3**	2285.2	4002.1

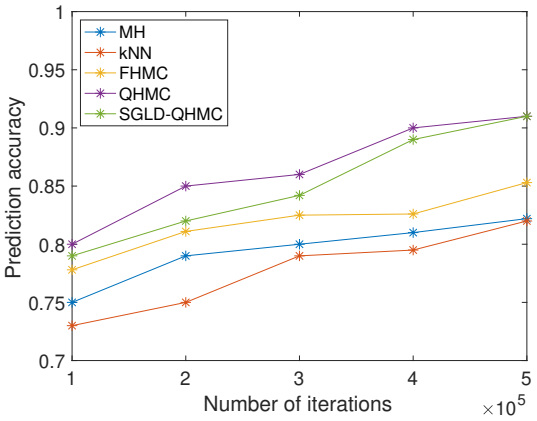

Figure 4: Comparison of the algorithms on adult dataset.

4 Significance and Impact

In this work, we propose a stochastic version of the QHMC method, and implement the method for missing data problems. The proposed method obtains the posterior distribution using the Bayesian approach. We integrate the SGLD framework with QHMC, where the gradient approximation is performed on a randomly selected subset of the dataset in every iteration. Our experiments have demonstrated that using the gradient estimates rather than exact sampling is an efficient way to impute missing variables, especially for large-scale datasets. We compared our results with other types of imputation methods including kNN, MH and FHMC, also original QHMC, and we found out that the SGLD-QHMC method provides better results. The experiments have evaluated the performance of the algorithms in three aspects: distance metric (NRMSE), prediction accuracy (PA) and execution time. In the first set of experiments using synthetic data, the results have shown that SGLD-QHMC provides slightly better NRMSE values than FHMC results and slightly worse NRMSE values than QHMC results, even when it uses 40% of the dataset. Since SGLD-QHMC uses only 40% of the samples to approximate gradients, it requires shorter time in the computation while maintaining good accuracy. The experiments on MNIST data show that MH and kNN are outperformed by HMC-based methods in terms of both NRMSE and execution time, while within the HMC-based methods, SGLD-QHMC is the most efficient. In the last set of experiments in which a real life adult dataset is used, we have shown that the SGLD-QHMC technique can estimate whether the income of a person is higher than $\$50K$ per year with a success rate of 90%, which is the same success rate attained by QHMC in a longer time. Moreover, the success rate of SGLD-QHMC might be increased by using a larger portion of the dataset and it can still perform faster than FHMC and QHMC.

Overall, the work has demonstrated that both QHMC and the proposed algorithm are efficient for missing data imputation. Imputing missing datasets using our approach outperforms the other improved multiple imputation methods such as kNN, and MCMC-based imputation methods such as MH and FHMC. Although FHMC is an efficient imputation method, SGLD-QHMC can provide the same or higher accuracy within a shorter amount of time. Unlike FHMC or other imputation methods that suffer from bias, SGLD-QHMC is more tolerant to larger datasets. Moreover, the results on MNIST and adult datasets show that SGLD-QHMC is promising to be generalized to various applications.

References

[1] H. D.-G. Acquah. Bayesian logistic regression modelling via Markov chain Monte Carlo algorithm. 2013.

[2] A. Barbu and S.-C. Zhu. *Monte Carlo Methods*, volume 35. Springer, 2020.

[3] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng. *Handbook of Markov chain Monte Carlo*. CRC press, 2011.

[4] Y. C Yuan. Multiple imputation for missing data: Concepts and new development. *SAS Institute Inc.*, January 2005.

[5] T. Chen, E. Fox, and C. Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In *International conference on machine learning*, pages 1683–1691. PMLR, 2014.

[6] C. Davidson-Pilon. *Bayesian methods for hackers: probabilistic programming and Bayesian inference*. Addison-Wesley Professional, 2015.

[7] N. Ding, Y. Fang, R. Babbush, C. Chen, R. D. Skeel, and H. Neven. Bayesian sampling using stochastic gradient thermostats. *Advances in neural information processing systems*, 27, 2014.

[8] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian data analysis*. Tyler & Francis Group, 2014.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[10] I. K. C. Joshi and L. Nolan. Generative adversarial networks (gans) for synthetic dataset generation with binary classes, 2019.

[11] G. Lin, Y. Wang, and Z. Zhang. Multi-variance replica exchange stochastic gradient MCMC for inverse and forward Bayesian physics-informed neural network. *arXiv preprint arXiv:2107.06330*, 2021.

[12] R. Little and D. Rubin. *Statistical Analysis with Missing Data, Second Edition*. Wiley, Hoboken, NJ, 2002. ISBN 9780471183860.

[13] Z. Liu and Z. Zhang. Quantum-inspired Hamiltonian Monte Carlo for Bayesian sampling. *arXiv preprint arXiv:1912.01937*, 2020.

[14] Y.-A. Ma, T. Chen, and E. Fox. A complete recipe for stochastic gradient MCMC. *Advances in neural information processing systems*, 28, 2015.

[15] T. A. Myers. Goodbye, listwise deletion: Presenting hot deck imputation as an easy and effective tool for handling missing data. *Communication methods and measures*, 5(4):297–310, 2011.

[16] R. B. O'hara, E. Arjas, H. Toivonen, and I. Hanski. Bayesian analysis of metapopulation data. *Wiley 2002, on behalf of the Ecological Society of America*, pages 1–4, 2002.

[17] K. Pelckmans, J. De Brabanter, J. A. Suykens, and B. De Moor. Handling missing values in support vector machine classifiers. *Neural Networks*, 18(5-6):684–692, 2005.

[18] N. Pourshahrokhi, S. Kouchaki, K. M. Kober, C. Miaskowski, and P. Barnaghi. A Hamiltonian Monte Carlo model for imputation and augmentation of healthcare data. *arXiv preprint arXiv:2103.02349*, 2021.

[19] H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[20] P. L. Roth, F. S. Switzer III, and D. M. Switzer. Missing data in multiple item scales: A Monte Carlo analysis of missing data techniques. *Organizational research methods*, 2(3):211–232, 1999.

[21] M. Soley-Bori. Dealing with missing data: Key assumptions and methods for applied analysis. *Boston University*, 23:20, 2013.

[22] Z. Wang, S. Mohamed, and N. Freitas. Adaptive Hamiltonian and Riemann manifold Monte Carlo. In *International conference on machine learning*, pages 1462–1470. PMLR, 2013.

[23] M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.

[24] Y. Yang, J. Kim, and I.-H. Cho. Parallel fractional hot deck imputation and variance estimation for big incomplete data curing. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[25] J. Yoon, J. Jordon, and M. Schaar. Gain: Missing data imputation using generative adversarial nets. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5689–5698. PMLR, 2018.

[26] S. Zhang. Nearest neighbor selection for iteratively kNN imputation. *Journal of Systems and Software*, 85(11):2541–2552, 2012.

[27] S. Zhang, Z. Qin, C. X. Ling, and S. Sheng. " missing is useful": Missing values in cost-sensitive decision trees. *IEEE transactions on knowledge and data engineering*, 17(12):1689–1693, 2005.

[28] S. Zhang, Z. Jin, and X. Zhu. Missing data imputation by utilizing information within incomplete instances. *Journal of Systems and Software*, 84(3):452–459, 2011.