Catalogue no. 12-001-X
ISSN 1492-0921

Survey Methodology

Integrating probability and non-probability samples through deep learning-based mass imputation

by Sixia Chen, Chao Xu and James Cutler

Release date: December 23, 2025

Statistics Canada Statistique Canada

Canada

How to obtain more information

For information about this product or the wide range of services and data available from Statistics Canada, visit our website, www.statcan.gc.ca.

You can also contact us by

Email at infostats@statcan.gc.ca

Telephone, from Monday to Friday, 8:30 a.m. to 4:30 p.m., at the following numbers:

- Statistical Information Service 1-800-263-1136
- National telecommunications device for the hearing impaired 1-800-363-7629
- Fax line 1-514-283-9350

Standards of service to the public

Statistics Canada is committed to serving its clients in a prompt, reliable and courteous manner. To this end, the Agency has developed standards of service which its employees observe in serving its clients. To obtain a copy of these service standards, please contact Statistics Canada toll-free at 1-800-263-1136. The service standards are also published on www.statcan.gc.ca under "Contact us" > "Standards of service to the public."

Note of appreciation

Canada owes the success of its statistical system to a long-standing partnership between Statistics Canada, the citizens of Canada, its businesses, governments and other institutions. Accurate and timely statistical information could not be produced without their continued co-operation and goodwill.

Survey Methodology, December 2025
Vol. 51, No. 2, pp. 493-508
Statistics Canada, Catalogue No. 12-001-X

Integrating probability and non-probability samples through deep learning-based mass imputation

Sixia Chen, Chao Xu and James Cutler[1]

Abstract

Although probability samples have been regarded as the gold standard to collect information for population-based study, non-probability samples have been used frequently in practice due to low cost, convenience, and the lack of the sampling frame for the survey. Naïve estimates based on non-probability samples without any adjustments may be misleading due to selection bias. Recently, a valid data integration approach that includes mass imputation, propensity score weighting, and calibration has been used to improve the representativeness of non-probability samples. The effectiveness of the mass imputation approach depends on the underlying model assumptions. In this paper, we propose using deep learning for the mass imputation in the combining of probability and non-probability samples and compare it with several modern machine learning-based mass imputation approaches, including generalized additive modeling, regression tree, random forest, and XG-boosting. In the simulation study, deep learning-based approaches have been shown to be more robust and effective than other mass imputation approaches against the failure of underlying model assumptions under non-linearity scenarios.

Key Words: Data integration; Machine learning; Nonprobability sample; Selection bias; Variance estimation.

1. Introduction

Even though probability samples have been regarded as the gold standard to conduct population-based research, non-probability samples have been used frequently in the fields of public health, education, economics, and medicine due to the low cost and time efficiency or the lack of sampling frame information. For instance, in 2015, Pew Research Center (http://www.pewresearch.org) produced a dataset consisting of nine non-probability samples and a wide range of measurements, including demographics, public opinion, and health behavior. The Southern Plains Tribal Health Board conducted a survey in 2019 to collect health-related information for Native American adults by combining the data collection through American Indian Affairs and Social Media Platforms, see Chen, Campbell, Spain, Milligan and Snider (2022). Although non-probability samples are currently in high demand for obtaining timely information, naive estimates from non-probability samples without proper adjustment may suffer from selection bias, see Baker, Brick, Bates, Battaglia, Couper, Dever, Gile and Tourangeau (2013).

By combining the information from both probability and non-probability samples, data integration approaches have been shown to be effective in reducing the selection bias of using non-probability sample alone, see Rivers (2007), Bethlehem (2016), Elliott and Valliant (2017), Lohr and Raghunathan (2017), Beaumont (2020), Rao (2021), and Kim and Tam (2021) for discussions about data integration methods. In general, existing data integration methods can be categorized into three broad categories: calibration, propensity score weighting, and mass imputation. The calibration approach generates calibrated weights to

1. Sixia Chen and Chao Xu, Department of Biostatistics and Epidemiology, University of Oklahoma Health Sciences, Oklahoma City, OK, USA, 73104. E-mail: sixia-chen@ouhsc.edu; James Cutler, Southern Plains Tribal Health Board, Oklahoma City, OK, 73114.

align the distribution of calibrated variables in a non-probability sample with that of a probability sample, see DiSogra, Cobb, Chan and Dennis (2011) and Elliott and Valliant (2017) for more discussions. The validity of the calibration approach depends on the underlying model assumptions between the study variables of interest and covariate variables. Propensity score weighting approaches (Lee, 2006; Lee and Valliant, 2009; Chen, Li and Wu, 2020) model the propensity score for participating in the non-probability sample and estimate it by solving the estimating equations formed by using information from both probability and non-probability samples. The validity of the propensity score weighting approaches depends on the underlying propensity score model assumptions. To improve the robustness, Mass imputation approaches first build prediction models of outcome variables of interest and covariate variables by using the non-probability sample and generate imputed values of the outcome variables for all units in the probability sample for statistical estimation, see Rivers (2007), Kim, Park, Chen and Wu (2021), Chen, Yang and Kim (2022) for more detailed discussions. The validity of the mass imputation approaches depends on the underlying imputation model assumptions. To further improve the robustness of the above data integration approaches, the doubly robust approach (Chen et al., 2020) and the multiply robust approach (Chen and Haziza, 2022) have been proposed. Furthermore, machine learning based methods have been considered by Buelens, Burger and van den Brakel (2018), Castro-Martín, Rueda, Ferri-García and Hernando-Tamayo (2021), Castro-Martín, Rueda and Ferri-García (2020), Ferri-García and Rueda (2020), among others. However, none of them considered deep learning based methods for data integration.

The mass imputation approaches based on parametric models are vulnerable to the failure of the underlying parametric model assumptions (Chen, Yang and Kim, 2022; Chen and Haziza, 2022). Even though the nonparametric mass imputation approach and multiply robust mass imputation approach have been developed, handling more complex linearity structure among predictors and high dimensionality still present a challenge. The literature shows that machine learning methods such as random forest (Breiman, 2001), XG-boosting (Chen and Guestrin, 2016) and deep learning (LeCun, Bengio and Hinton, 2015) are very effective in terms of prediction. In addition, XG-boosting and deep learning methods have been demonstrated to be very effective in imputation for missing values when the association among covariate variables is highly nonlinear or the dimension is high, see Yoon, Jordon and Schaar (2018), Gondara and Wang (2018), Dagdoug, Goga and Haziza (2023a) and Chen and Xu (2022), among others. Furthermore, deep learning methods have begun substantially outperforming other statistical learning methods in many areas, including speech recognition (Hinton, Deng, Yu, Dahl, Mohamed, Jaitly, Senior, Vanhoucke, Nguyen, Sainath and Kingsbury, 2012; Graves, Mohamed and Hinton, 2013), image analysis (Farabet, Couprie, Najman and LeCun, 2012; Krizhevsky, Sutskever and Hinton, 2012; Szegedy, Liu, Jia, Sermanet, Reed, Anguelov, Erhan, Vanhoucke and Rabinovich, 2015) and natural language processing (Collobert, Weston, Bottou, Karlen, Kavukcuoglu and Kuksa, 2011; Sarikaya, Hinton and Deoras, 2014). More recently, the asymptotic properties of deep learning- based statistical estimators, including nonparametric regression, survival analysis, and quantile regression, have been discussed in Bauer and Kohler (2019), Schmidt-Hieber (2020), Farrell, Liang and Misra (2021), Zhong, Mueller and Wang (2022), Padilla, Tansey and Chen (2020), among others. However, the development of deep learning-based mass imputation

approaches to combine probability and non-probability samples is lacking in the current literature. To fill this important research gap, we proposed using deep learning approaches for mass imputation and comparing its performance with other popular existing machine learning-based approaches (e.g., generalized additive modeling, regression tree, random forest, and XG-boosting) through extensive simulation study and real application. Our simulation study and real application show that the deep learning approach outperforms other methods in terms of balancing the bias and variance. Deep learning performs the best among all methods.

Our paper is organized as follows. Section 2 introduces the basic setups and mathematical notations of our research question. Deep learning-based mass imputation methods are discussed in Section 3. A Monte Carlo simulation study is conducted in Section 4 to compare different machine learning-based mass imputation methods. Section 5 contains a discussion of the findings and some future research areas.

2. Basic setups

Suppose we have a finite population $\mathcal{F}_N = \{(\mathbf{x}_i, y_i),\ i = 1,2,\ldots,N\}$ with population size N where $\mathbf{x} = (x_1, x_2, \ldots, x_p)$ is the covariate vector with dimension p and y is a scalar study variable of interest. The finite population $\mathcal{F}_N$ is a random sample from a super-population model

$$y_i = m(\mathbf{x}_i; \boldsymbol{\beta}^*) + \epsilon_i, \quad i = 1,2,\ldots,N, \tag{2.1}$$

where $m(\mathbf{x}_i; \boldsymbol{\beta}^*)$ is a unknown function with unknown true parameter $\boldsymbol{\beta}^*$ and the random error satisfies $E(\epsilon_i \mid \mathbf{x}_i) = 0$. Given the finite population $\mathcal{F}_N$, a probability sample S_A is selected by using some probability sampling design with first- and second-order inclusion probabilities as π_i and π_{ij} for $i \neq j$. Then, the corresponding design weight can be written as $w_i = \pi_i^{-1}$. Assume that we only observe the covariate vector $\mathbf{x}$ in the probability sample S_A. Furthermore, a non-probability sample S_B is selected from $\mathcal{F}_N$ with δ_i as the selection indicator such that $\delta_i = 1$ if unit i is selected and $\delta_i = 0$ otherwise. We assume the selection probability is unknown and only depends on $\mathbf{x}$ as follows:

$$\Pr(\delta = 1 \mid \mathbf{x}, y) = \Pr(\delta = 1 \mid \mathbf{x}). \tag{2.2}$$

Assume we observe both $\mathbf{x}$ and y in the non-probability sample S_B. This assumption closely mirrors real-life scenarios, as non-probability samples are customarily designed to address specific research questions of interest while also encompassing background demographic and socioeconomic information. In contrast, probability samples frequently involve established large-scale datasets such as the Behavioral Risk Factor Surveillance System (BRFSS), the National Health and Nutrition Examination Survey (NHANES), and the American Community Survey. These probability samples primarily emphasize the collection of comprehensive background demographic and socioeconomic data, in addition to other pertinent information related to the target population. Suppose the finite population parameter of interest $\boldsymbol{\theta}_N$ is defined as the unique solution of the estimating equation:

$$\mathbf{U}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}\mathbf{u}(\mathbf{x}_i, y_i; \boldsymbol{\theta}) = \mathbf{0}, \tag{2.3}$$

where $\mathbf{u}(\mathbf{x}, y; \boldsymbol{\theta})$ is a prespecified function of $\mathbf{x}$, y, and $\boldsymbol{\theta}$. For instance, $\mathbf{u}(\mathbf{x}, y; \boldsymbol{\theta}) = y - \theta$ corresponds to the the finite population mean $\theta_N = N^{-1}\sum_{i=1}^{N} y_i$. $\mathbf{u}(\mathbf{x}, y; \boldsymbol{\theta}) = I(y \le \theta_N) - \tau$ corresponds to the finite population 100τ -th percentile.

3. Proposed method

In this section, we discuss deep learning-based mass imputation methods for estimating the finite population parameter $\boldsymbol{\theta}_N$ defined in the previous section. Following the framework of Bauer and Kohler (2019) and Schmidt-Hieber (2020), we consider the following L-Layer deep neural network (DNN) structure for modeling $m(\mathbf{x}; \boldsymbol{\beta}^*)$. The nodes in the first hidden layer are defined as

$$g_j^{(1)}(\mathbf{x}) = \sigma\left(\beta_{j0}^{(1)} + \sum_{k=1}^{p}\beta_{jk}^{(1)}x_k\right) \tag{3.1}$$

for $j = 1, 2, \ldots, d_1$, where $\sigma(t)$ is the activation function used to introduce the non-linearity. The nodes in the next $L-1$ layers are recursively defined as

$$g_j^{(l)}(\mathbf{x}) = \sigma\left(\beta_{j0}^{(l)} + \sum_{k=1}^{d_{l-1}}\beta_{jk}^{(l)}g_k^{(l-1)}(\mathbf{x})\right) \tag{3.2}$$

for $j = 1, 2, \ldots, d_l$ and $l = 2, 3, \ldots, L$, where the number of hidden layers L is called the depth of the DNN and the number of nodes d_l for $l = 1, 2, \ldots, L$ is called the width of each layer. A deep learning node is a fundamental computational unit equipped with one or more weighted input connections, a transfer function that amalgamates these inputs, and an output connection. These nodes are subsequently organized into layers to form a comprehensive neural network. A layer refers to a level of organization within a neural network. Neural networks are composed of layers of interconnected nodes or neurons. Each layer serves a specific purpose in processing and transforming input data. Then, the nodes in the final layer $g_1^{(L)}(\mathbf{x}), g_2^{(L)}(\mathbf{x}), \ldots, g_{d_L}^{(L)}(\mathbf{x})$ are used as basis functions for predicting y by using the following working model

$$E(y \mid \mathbf{x}) = \beta_0^{(L+1)} + \sum_{j=1}^{d_L}\beta_j^{(L+1)}g_j^{(L)}(\mathbf{x}) \triangleq \tilde{m}(\mathbf{x}; \tilde{\boldsymbol{\beta}}, L, \mathbf{d}, \sigma), \tag{3.3}$$

where $\tilde{\boldsymbol{\beta}}$ denotes the vector containing all the regression coefficients defined in (3.1), (3.2), and (3.3), L, $\mathbf{d} = (d_1, d_2, \ldots, d_L)^\top$, and σ are the tuning parameters that can be selected by using the K-fold cross validation approach. The most popular choice for the activation function σ includes the Rectified Linear Unit (relu) defined as $\sigma(t) = \max(t, 0)$ and Scaled Exponential Linear Unit (selu) defined as $\sigma(t) = \lambda t$ if $t > 0$, or $\lambda\alpha(e^t - 1)$ if $t \le 0$ with λ and α as some fixed approximate values. An example of a four-layer deep neural network is presented in Figure 3.1. For a review of different activation functions in deep neural

networks, see Rasamoelina, Adjailia and Sinčák (2020). For the given tuning parameters L^*, $\mathbf{d}^*$, and σ^*, the model fitting can be done by minimizing the following cost function with non-probability sample

$$Q(\tilde{\boldsymbol{\beta}}) = \frac{1}{n_B} \sum_{i \in S_B} \left\{ y_i - \tilde{m}(\mathbf{x}_i;\, \tilde{\boldsymbol{\beta}}, L^*, \mathbf{d}^*, \sigma^*) \right\}^2, \tag{3.4}$$

where n_B is the sample size for non-probability sample S_B and σ^* is the given activating function defined previously in equations (3.1) and (3.2). The above optimization procedure can be done numerically with stochastic gradient descent (Bottou, 2010) and backpropagation (LeCun et al., 2015), while guaranteeing reliable performance on unseen data (ensuring generalization). Denote the estimated regression coefficients through the above optimization procedure as $\hat{\tilde{\boldsymbol{\beta}}}$, then the mass imputed estimator $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}_N$ can be obtained by solving the following weighted estimating equation:

$$\hat{\mathbf{U}}(\boldsymbol{\theta}) = \sum_{i \in S_A} w_i \mathbf{u}(\mathbf{x}_i, y_i^*;\, \boldsymbol{\theta}) = \mathbf{0}, \tag{3.5}$$

where y_i^* is the imputed value for unit i in probability sample S_A and it can be written as $y_i^* = \tilde{m}(\mathbf{x}_i;\, \hat{\tilde{\boldsymbol{\beta}}}, L^*, \mathbf{d}^*, \sigma^*) + \epsilon_i^*$ with ϵ_i^* selected from the pool of residuals $\{\hat{\epsilon_i} = y_i - \tilde{m}(\mathbf{x}_i;\, \hat{\tilde{\boldsymbol{\beta}}}, L^*, \mathbf{d}^*, \sigma^*),\, i \in S_A\}$ by using the same sampling design as the original probability sample S_A. This is the so-called random imputation and it preserves the distribution of the imputed values, see Chen and Haziza (2019) for more detailed discussions. For estimating the finite population mean $\theta_N = N^{-1} \sum_{i=1}^N y_i$, one can use the following deterministic imputed estimator to reduce the variability of the imputed estimator:

$$\hat{\theta} = \frac{\sum_{i \in S_A} w_i \tilde{m}(\mathbf{x}_i;\, \hat{\tilde{\boldsymbol{\beta}}}, L^*, \mathbf{d}^*, \sigma^*)}{\sum_{i \in S_A} w_i}. \tag{3.6}$$

Under the regularity conditions specified in Farrell et al. (2021) and following their arguments and derivations, one can show the consistency of our proposed estimator $\hat{\theta}$. For estimating the finite population distribution function $\theta_N = N^{-1} \sum_{i=1}^N I(y_i \le t)$, the corresponding estimating equation is $\mathbf{u}(\mathbf{x}_i, y_i;\, \boldsymbol{\theta}) = I(y_i \le t) - \boldsymbol{\theta}$ and the mass imputed estimator can be written as:

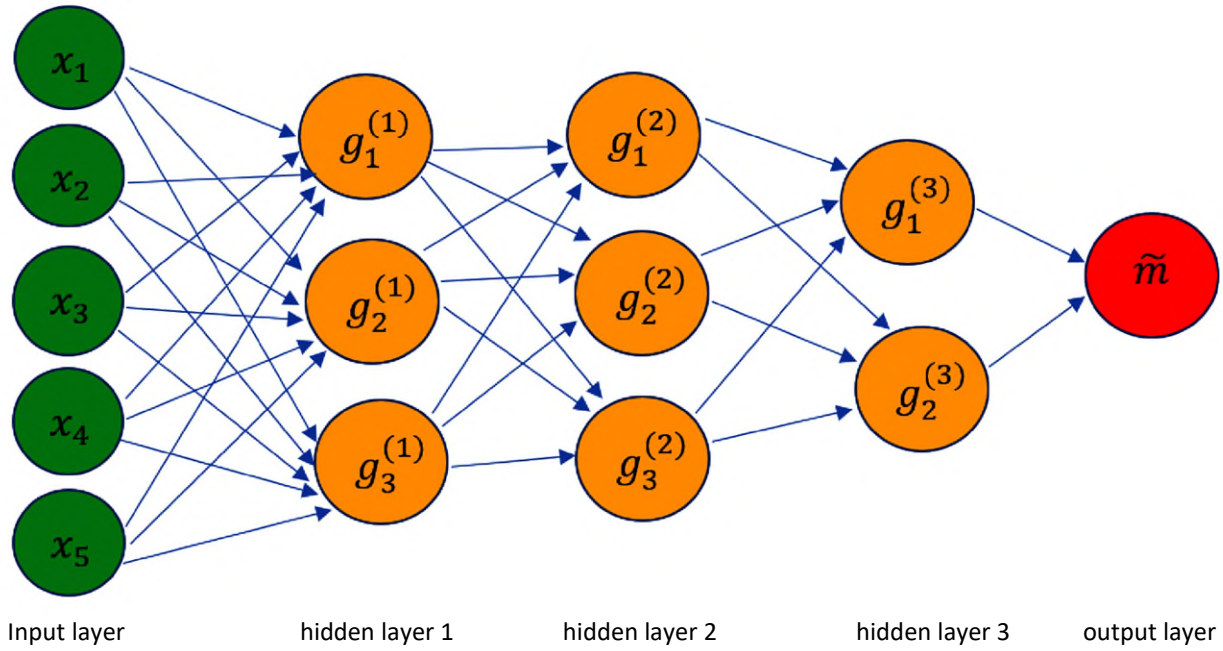
$$\hat{\theta} = \frac{\sum_{i \in S_A} w_i I(y_i^* \le t)}{\sum_{i \in S_A} w_i}. \tag{3.7}$$

For estimating the finite population 100τ -th percentile, the corresponding estimating equation is $\mathbf{u}(\mathbf{x}_i, y_i;\, \boldsymbol{\theta}) = I(y_i \le \theta) - \tau$ and the mass imputed estimator can be obtained by solving:

$$\hat{\mathbf{U}}(\boldsymbol{\theta}) = \sum_{i \in S_A} w_i \left\{ I(y_i^* \le \theta) - \tau \right\} = \mathbf{0}, \tag{3.8}$$

where y_i^* is the imputed value generated from random imputation.

Figure 3.1 A four-layer deep neural network



Remark 3.1 *Statistical inference based on (3.6) is quite challenging when the sample size n_B of non-probability sample is small. However, when the sample size n_B is large compared to n_A and under some regularity conditions, by using similar arguments as that in Yang, Kim and Hwang (2021) and Angelopoulos, Bates, Fannjiang, Jordan and Zrnic (2023), it can be shown that the randomness for estimating the predictive values based on non-probabiltiy sample can be safely ignored and we have*

$$\begin{aligned}\hat{\theta} &= \frac{\sum_{i\in S_A} w_i \tilde{m}(\mathbf{x}_i)}{\sum_{i\in S_A} w_i} \\ &= \frac{\sum_{i\in S_A} w_i m^*(\mathbf{x}_i)}{\sum_{i\in S_A} w_i} + \frac{\sum_{i\in S_A} w_i \left\{\tilde{m}(\mathbf{x}_i) - m^*(\mathbf{x}_i)\right\}}{\sum_{i\in S_A} w_i} \\ &= \frac{\sum_{i\in S_A} w_i m^*(\mathbf{x}_i)}{\sum_{i\in S_A} w_i} + o_p(n_A^{-1/2}),\end{aligned} \tag{3.9}$$

where $\tilde{m}(\mathbf{x}_i) = \tilde{m}(\mathbf{x}_i; \hat{\tilde{\boldsymbol{\beta}}}, L^*, \mathbf{d}^*, \sigma^*)$ and $m^*(\mathbf{x}_i)$ is the probability limit of $\tilde{m}(\mathbf{x}_i)$. Therefore, the asymptotic variance of $\hat{\theta}$ can be written as

$$\begin{aligned}\mathrm{Var}(\hat{\theta}) = &\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{\pi_{ij}-\pi_i\pi_j}{\pi_i\pi_j}\left\{m^*(\mathbf{x}_i)-\theta^*\right\}\left\{m^*(\mathbf{x}_j)-\theta^*\right\} \\ &+ o(n_A^{-1}),\end{aligned} \tag{3.10}$$

where θ^* is the probability limit of $\hat{\theta}$. Hence, the variance estimator can be written as

$$\hat{\text{Var}}(\hat{\theta}) = \frac{1}{\hat{N}^2}\sum_{i\in S_A}\sum_{i\in S_A}\frac{\pi_{ij}-\pi_i\pi_j}{\pi_{ij}\pi_i\pi_j}\left\{\tilde{m}(\mathbf{x}_i)-\hat{\theta}\right\}\left\{\tilde{m}(\mathbf{x}_j)-\hat{\theta}\right\}, \tag{3.11}$$

where $\hat{N}=\sum_{i\in S_A} w_i$.

4. Simulation studies

We generate $B=$ 200 Monte Carlo samples of finite populations with population size $N=$ 10,000 from the following three different super-population models:

(M1). $y_i = 1+x_{1,i}+x_{2,i}+x_{3,i}+x_{4,i}+\epsilon_i$ for $i=1,2,\ldots,N$, where $x_{1,i}$, $x_{2,i}$, $x_{3,i}$, $x_{4,i}$, and ϵ_i are mutually independent random variables generated from standard normal distribution.

(M2). $y_i = 1+x_{1,i}^2+x_{1,i}x_{2,i}x_{3,i}+x_{1,i}x_{2,i}+x_{1,i}x_{2,i}x_{3,i}x_{4,i}^2+\epsilon_i$ for $i=1,2,\ldots,N$, where $x_{1,i}$, $x_{2,i}$, $x_{3,i}$, $x_{4,i}$, and ϵ_i are mutually independent random variables generated from standard normal distribution.

(M3). Study variable y is generated from the following multi-layer models:

$$\begin{aligned}
a_{k,i}^{(1)} &= \log\left\{1+\exp\left(\alpha_{k0}^{(1)}+\sum_{r=1}^{20}\alpha_{kr}^{(1)}x_{r,i}\right)\right\},\ k=1,2,3,\\
a_{j,i}^{(2)} &= \log\left\{1+\exp\left(\alpha_{j0}^{(2)}+\sum_{k=1}^{3}\alpha_{jk}^{(2)}a_{k,i}^{(1)}\right)\right\},\ j=1,2,3,\\
y_i &= \beta_0+\sum_{j=1}^{3}\beta_j a_{j,i}^{(2)}+\epsilon_i, \qquad i=1,2,\ldots,N,
\end{aligned}$$

where ϵ_i is i.i.d sample from standard normal distribution. The coefficients $\alpha_{k0}^{(1)}$, $\alpha_{kr}^{(1)}$, $\alpha_{j0}^{(2)}$, $\alpha_{jk}^{(2)}$, β_0, and β_j are independently generated once from a uniform distribution with a range from -2 to 2 at the beginning of the simulation study and fixed for all Monte Carlo samples. $x_{r,i}$ for $r=1,2,\ldots,20$ are independently generated from a uniform distribution with range from -1 to 1.

Given the finite population, a probability sample S_A with sample size $n_A=$ 500 is selected by using (a). Simple random sampling without replacement sampling design (SRS); (b). Randomized systematic sampling with probability proportional to size (PPS) where the size variable $z_i = 0.5\psi_i+1$ with ψ_i generated from standard chi-square distribution with one degrees of freedom. For models (M1)-(M2), a non-probability sample S_B is selected from Bernoulli distribution with probability

$$p(\mathbf{x}_i) = \frac{\exp\left(c+x_{1,i}+x_{2,i}+x_{3,i}+x_{4,i}\right)}{1+\exp\left(c+x_{1,i}+x_{2,i}+x_{3,i}+x_{4,i}\right)},$$

where $\mathbf{x}_i=(x_{1,i},x_{2,i},x_{3,i},x_{4,i})$ and c is chosen to make the sample size n_B be about 500. For model (M3), a non-probability sample S_B with sample size n_B around 500 is selected from Bernoulli distribution with probability

$$p(\mathbf{x}_i) = \frac{\exp\left(c + \sum_{r=1}^{20} x_{r,i}\right)}{1 + \exp\left(c + \sum_{r=1}^{20} x_{r,i}\right)},$$

where $\mathbf{x}_i = (x_{1,i}, x_{2,i}, \ldots, x_{20,i})$, and c is chosen to make the sample size n_B be about 500. All parameters are chosen before the simulation study and we keep them the same for all Monte Carlo samples. Suppose we are interested in estimating the super-population mean (θ_0) of y. We compared the following methods: (1). Linear regression model (LM) with stepwise model selection based on AIC; (2). Generalized additive modeling (GAM); (3). Regression tree (RT); (4). Random forest (RF); (5). XG-boosting (XGB); (6). Our proposed deep learning (DL) methods with two activation functions (e.g., relu and selu) and three options of the number of layers (e.g., 3, 4, and 5 layers). Each layer contained 50 neurons or nodes. The linear activation was used in the outcome layer for continuous data. The corresponding estimate of each method was obtained by using

$$\hat{\theta} = \frac{\sum_{i \in S_A} w_i \hat{m}(\mathbf{x}_i)}{\sum_{i \in S_A} w_i}, \tag{4.1}$$

where $\hat{m}(\mathbf{x}_i)$ is the predictive value of subject $i \in S_A$ based on the corresponding method.

The machine learning methods RT, RF, and XGB were tuned using R packages `parsnip`, `workflows`, and `recipes`. The key hyper-parameters that we tuned for these methods are listed in Table 4.1. Hyper-parameters are machine learning parameters whose value control the learning process and determine the model performance. For more detailed information of those hyper-parameters, see Hastie, Tibshirani, Friedman and Friedman (2009). For each of these methods, 30 combination sets of tuning hyper-parameters were searched, based on 10-fold cross validation, to select the optimum set by minimizing the predictive root mean squared error (RMSE). Because we have no idea the search range of each hyper-parameter, so we first tried the default search range provided by the R packages on one simulated dataset. A refined search range can be set by visually checking the plots of RMSE versus each hyper-parameter. Then, we applied the refined search range for all simulation datasets to get the final tuning hyper-parameters. Given the tuning hyper-parameters, the parameter estimation model using these models was derived and fitted respectively.

In the context of deep learning, the learning rate is a hyper-parameter that determines the size of the steps taken during the optimization process. It represents the proportion by which the model's parameters are updated during each iteration of the optimization process. For the deep learning methods, we tuned the learning rate from (0.1, 0.01, 0.001, 0.0001, 0.00001) using a similar procedure. Based on 200 simulated datasets, we could set an optimum learning rate for all datasets by comparing the target function RMSE. The Python packages `tensorflow` and `keras` were used for the implementation. The validation split was 0.25. Epochs were set to 200. Epochs are iterations over the entire dataset during the training phase, where the model learns from the data and updates its parameters to minimize the loss function. Data were shuffled before the start of training. To further prevent over-fitting, the early stopping rule was applied if model performance in validation set could not improve after 15 epochs.

Table 4.1
List of tuning parameters involved in different approaches

Method	Parameters	Definition
RT	tree_depth	maximum depth of the tree
	min_n	minimum number of data points in a node
	cost_complexity	cost/complexity parameter
RF	mtry	number of selected predictors
	trees	number of trees contained in the ensemble
	min_n	minimum number of data points in a node
XGB	tree_depth	maximum depth of the tree
	trees	number of trees contained in the ensemble
	learn_rate	learning rate
	mtry	number of selected predictors
	min_n	minimum number of data points in a node

Note: RF =random forest; RT = regression tree; XGB = extreme gradient boosting.

We compare the performance of different estimators by using Monte Carlo relative bias (RB), relative standard error (RSE), and relative root mean squared error (RRMSE) since they were used in existing literature (Chen and Xu, 2023; Dagdoug, Goga and Haziza, 2023b; Chen and Haziza, 2023) as effective measures to compare the performance of estimators in Monte Carlo simulation study. Smaller values of those measures indicate better precision and accuracy. They can be calculated as following.

$$\text{RB} = \frac{B^{-1}\sum_{k=1}^{B}\hat{\theta}_k - \theta_0}{|\theta_0|}, \tag{4.2}$$

$$\text{RSE} = \frac{\left\{(B-1)^{-1}\sum_{k=1}^{B}(\hat{\theta}_k - \bar{\hat{\theta}})^2\right\}^{1/2}}{|\theta_0|}, \tag{4.3}$$

and

$$\text{RRMSE} = (\text{RB}^2 + \text{RSE}^2)^{1/2}, \tag{4.4}$$

where $\hat{\theta}_k$ is the estimator based on the k -th Monte Carlo sample, B is the number of Monte Carlo samples defined previously, and $\bar{\hat{\theta}} = B^{-1}\sum_{k=1}^{B}\hat{\theta}_k$. Note that $\hat{\theta}_k$ denotes any one of the six estimators considered in the simulation study.

The results are presented in Table 4.2 (SRS) and Table 4.3 (PPS). Under a linear regression model (M1), LM, GAM, and DL methods have smaller RB, RSE, and RRMSE than do other methods since tree-based methods have extrapolation issue such that they can not predict out-of-range values in training data, see Hengl, Nussbaum, Wright, Heuvelink and Gräler (2018). In our simulated data, the ranges of covariate variable x for S_A and training data S_B are quite different. This is due to the magnitude of selection bias introduced by the probability mechanism of the non-probability sample. A more recent study has proved that, with appropriate training data, deep neural networks can extrapolate linear target functions (Xu, Zhang, Li, Du, Kawarabayashi and Jegelka, 2020), which is our setup. Under model (M1), LM and GAM methods have the smallest RB, RSE, and RRMSE since they used the correct model assumptions. DL method with using three layer and relu or selu activation function has comparable results with that of LM and GAM in

terms of RB. Under nonlinear models (M2) and (M3), DL methods are significantly better than are other methods in terms of producing smaller relative bias, and relative root mean squared error. Tree-based methods still have extrapolation issue. LM did not perform well due to the model misspecification. GAM had large biases since it did not account for interactions between auxiliary variables. For (M1), DL method with using three layer and selu activation function has the best result under SRS. DL method with using three layer and relu activation function has the best result under PPS. For (M2), DL method with using five layer, relu and selu activation functions have the best results under SRS. DL method with using three layer, relu activation function and five layer, selu activation function have the best results under PPS. For (M3), all DL methods have comparable results under both SRS and PPS. In summary, our proposed DL methods show results that are better than or comparable to those of all other methods, and the performance under all scenarios is stable. There are some variations in terms of performance for DL methods with different activation functions and Layers. When the underlying model is linear, LM and GAM had the best performance. When the underlying model is non-linear with interaction terms or multi-layer model structure. LM and GAM may produce large bias, DL had the best performance. Tree based methods including RF, RT, and XGB may not perform well when there is large discrepancy between the ranges of covariate variables in both samples. The computational code for the simulation study can be accessed at the following link: https://github.com/xu1912/imputationDL.

Table 4.2
Relative bias (RB) (%), Relative standard error (RSE) (%), and Relative root mean squared error (RRMSE) (%) of different methods under models (M1)-(M3) and simple random sampling without replacement

Model	Method	RB	RSE	RRMSE
	LM	0.18	11.94	11.94
	GAM	0.31	12.63	12.64
	RF	86.93	12.15	87.77
	RT	95.44	18.56	97.23
	XGB	34.06	16.49	37.84
M1	DL 3-layer relu	-0.89	18.97	18.99
	DL 3-layer selu	0.57	14.51	14.52
	DL 4-layer relu	-3.45	17.59	17.92
	DL 4-layer selu	4.14	15.81	16.35
	DL 5-layer relu	4.89	22.61	23.13
	DL 5-layer selu	8.02	13.87	16.02
	LM	-133.36	28.23	136.31
	GAM	-135.63	41.61	141.87
	RF	-25.29	20.64	32.65
	RT	-24.85	24.75	35.07
	XGB	-15.42	19.40	24.78
M2	DL 3-layer relu	-1.19	13.37	13.42
	DL 3-layer selu	3.07	13.60	13.94
	DL 4-layer relu	2.84	13.99	14.27
	DL 4-layer selu	-3.74	24.12	24.41
	DL 5-layer relu	0.69	12.91	12.93
	DL 5-layer selu	-0.40	12.68	12.69

Note: DL = deep learning; GAM = generalized additive modeling; LM = linear regression model; RF =random forest; RT = regression tree; XGB = extreme gradient boosting.

Table 4.2 (continued)
Relative bias (RB) (%), Relative standard error (RSE) (%), and Relative root mean squared error (RRMSE) (%) of different methods under models (M1)-(M3) and simple random sampling without replacement

Model	Method	RB	RSE	RRMSE
	LM	15.88	6.54	17.18
	GAM	15.25	6.41	16.55
	RF	11.72	4.57	12.58
	RT	10.49	6.81	12.51
	XGB	13.17	5.52	14.28
M3	DL 3-layer relu	1.06	4.98	5.09
	DL 3-layer selu	5.42	4.91	7.31
	DL 4-layer relu	2.10	5.45	5.85
	DL 4-layer selu	1.43	4.71	4.92
	DL 5-layer relu	1.39	4.99	5.18
	DL 5-layer selu	1.32	4.90	5.08

Note: DL = deep learning; GAM = generalized additive modeling; LM = linear regression model; RF =random forest; RT = regression tree; XGB = extreme gradient boosting.

Table 4.3
Relative bias (RB) (%), Relative standard error (RSE) (%), and Relative root mean squared error (RRMSE) (%) of different methods under models (M1)-(M3) and randomized systematic sampling with probability proportional to size

Model	Method	RB	RSE	RRMSE
	LM	-0.46	12.37	12.38
	GAM	-0.35	12.46	12.47
	RF	85.47	11.56	86.25
	RT	93.69	18.06	95.41
	XGB	33.66	17.18	37.79
M1	DL 3-layer relu	-0.44	17.74	17.75
	DL 3-layer selu	-3.76	18.13	18.51
	DL 4-layer relu	1.52	17.63	17.69
	DL 4-layer selu	-2.88	17.80	18.03
	DL 5-layer relu	-2.16	18.82	18.94
	DL 5-layer selu	-1.86	17.81	17.91
	LM	-132.00	25.46	134.43
	GAM	-132.82	42.09	139.33
	RF	-23.53	19.90	30.82
	RT	-24.18	30.11	38.62
	XGB	-12.35	18.17	21.97
M2	DL 3-layer relu	0.14	11.09	11.09
	DL 3-layer selu	-4.26	11.34	12.11
	DL 4-layer relu	-3.41	12.94	13.38
	DL 4-layer selu	3.30	10.82	11.31
	DL 5-layer relu	-5.23	11.84	12.94
	DL 5-layer selu	-0.38	11.23	11.24
	LM	16.35	8.40	18.38
	GAM	14.52	8.29	16.72
	RF	45.94	6.26	46.36
	RT	50.08	10.36	51.14
	XGB	27.46	7.71	28.52
M3	DL 3-layer relu	6.92	8.94	11.30
	DL 3-layer selu	4.35	10.61	11.46
	DL 4-layer relu	10.66	10.14	14.71
	DL 4-layer selu	8.90	12.14	15.05
	DL 5-layer relu	5.73	10.39	11.87
	DL 5-layer selu	3.03	10.40	10.83

Note: DL = deep learning; GAM = generalized additive modeling; LM = linear regression model; RF =random forest; RT = regression tree; XGB = extreme gradient boosting.

5. Discussion

In this paper, we compared deep learning-based mass imputation methods with some existing machine learning-based mass imputation methods for data integration. Simulation study showed the benefits of our proposed deep learning-based methods in terms of reducing the Monte Carlo bias, standard error, and mean squared error. Specifically, deep learning methods outperformed other machine learning methods dramatically with a highly non-linear hierarchical model structure. In terms of computational considerations, deep learning methods require more time for tuning, but this may not be a big issue with modern computational power. In our simulation study, it took 5 minutes to complete 200 simulations for model 3, which is the most complicated model, at learning rate of 0.1. It took 27 minutes to complete 200 simulations for model 3, when the learning rate was 0.00001. We used NVIDIA Quadro P600 GPU for the deep learning. The P600 GPU was released 7 years ago. With much faster computation speed provided by new GPUs, the computation time for running deep learning models is expected to be trivial. Once an optimized deep learning model is derived, it would be fast to apply it to new data. The key consideration for deep learning could still be preventing the over-fitting, while early stopping, random drop out, and high validation split could help. Obtaining the optimal selection of tuning parameters including the number of layers and the number of nodes remains as an open research question. Statistical inference with deep learning-based mass imputation with small sample size of non-probability sample is quite challenging and we plan to pursue it as a future research topic. In addition, more empirical evidence using real data examples is needed for future research.

Acknowledgements

Dr. Chen was supported by ASA/NSF/Census Research Fellowship with the U.S. Bureau of the Census (Contract Number-1333LB24P00000032), the National Institute on Minority Health and Health Disparities at National Institutes of Health (1R21MD014658-01A1) and the Oklahoma Shared Clinical and Translational Resources (U54GM104938) with an Institutional Development Award from National Institute of General Medical Sciences. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health and US Census. Part of the computing for this project was performed at the OU Supercomputing Center for Education & Research (OSCER) at the University of Oklahoma (OU).

References

Angelopoulos, A.N., Bates, S., Fannjiang, C., Jordan, M.I. and Zrnic, T. (2023). Prediction-powered inference. *Science*, 382(6671), 669-674.

Baker, R., Brick, J.M., Bates, N.A., Battaglia, M., Couper, M.P., Dever, J.A., Gile, K.J. and Tourangeau, R. (2013). Summary report of the AAPOR task force on non-probability sampling. *Journal of Survey Statistics and Methodology*, 1(2), 90-143.

Bauer, B. and Kohler, M. (2019). On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *The Annals of Statistics*, 47(4), 2261-2285.

Beaumont, J.-F. (2020). Are probability surveys bound to disappear for the production of official statistics? *Survey Methodology*, 46(1), 1-28. Paper available at https://www150.statcan.gc.ca/n1/en/pub/12-001-x/2020001/article/00001-eng.pdf.

Bethlehem, J. (2016). Solving the nonresponse problem with sample matching? *Social Science Computer Review*, 34(1), 59-77.

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. *Proceedings of COMPSTAT'2010*, Springer, 177-186.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32.

Buelens, B., Burger, J. and van den Brakel, J.A. (2018). Comparing inference methods for non-probability samples. *International Statistical Review*, 86(2), 322-343.

Castro-Martín, L., Rueda, M.d.M. and Ferri-García, R. (2020). Inference from non-probability surveys with statistical matching and propensity score adjustment using modern prediction techniques. *Mathematics*, 8(6), 879.

Castro-Martín, L., Rueda, M.d.M., Ferri-García, R. and Hernando-Tamayo, C. (2021). On the use of gradient boosting methods to improve the estimation with data obtained with self-selection procedures. *Mathematics*, 9(23), 2991.

Chen, S., Campbell, J., Spain, E., Milligan, A. and Snider, C. (2022). Improving the representativeness of the tribal behavioral risk factor surveillance system through data integration, submitted.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785-794.

Chen, S. and Haziza, D. (2019). Recent developments in dealing with item non-response in surveys: A critical review. *International Statistical Review*, 87, S192-S218.

Chen, S. and Haziza, D. (2022). A general multiply robust framework for combining probability and non-probability samples in surveys. *Scandinavian Journal of Statistics*.

Chen, S. and Haziza, D. (2023). General purpose multiply robust data integration procedures for handling nonprobability samples. *Scandinavian Journal of Statistics*, 50(2), 697-724.

Chen, Y., Li, P. and Wu, C. (2020). Doubly robust inference with nonprobability survey samples. *Journal of the American Statistical Association*, 115(532), 2011-2021.

Chen, S. and Xu, C. (2022). Handling high-dimensional data with missing values by modern machine learning techniques. *Journal of Applied Statistics*, 1-19.

Chen, S. and Xu, C. (2023). Handling high-dimensional data with missing values by modern machine learning techniques. *Journal of Applied Statistics*, 50(3), 786-804.

Chen, S., Yang, S. and Kim, J.K. (2022). Nonparametric mass imputation for data integration. *Journal of Survey Statistics and Methodology*, 10(1), 1-24.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 2493-2537.

Dagdoug, M., Goga, C. and Haziza, D. (2023a). Imputation procedures in surveys using nonparametric and machine learning methods: An empirical comparison. *Journal of Survey Statistics and Methodology*, 11(1), 141-188.

Dagdoug, M., Goga, C. and Haziza, D. (2023b). Model-assisted estimation through random forests in finite population sampling. *Journal of the American Statistical Association*, 118(542), 1234-1251.

DiSogra, C., Cobb, C., Chan, E. and Dennis, J.M. (2011). Calibrating non-probability internet samples with probability samples using early adopter characteristics. *Joint Statistical Meetings (JSM), Survey Research Methods*, 4501-4515.

Elliott, M.R. and Valliant, R. (2017). Inference for nonprobability samples. *Statistical Science*, 32(2), 249-264.

Farabet, C., Couprie, C., Najman, L. and LeCun, Y. (2012). Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 1915-1929.

Farrell, M.H., Liang, T. and Misra, S. (2021). Deep neural networks for estimation and inference. *Econometrica*, 89(1), 181-213.

Ferri-García, R. and Rueda, M.d.M. (2020). Propensity score adjustment using machine learning classification algorithms to control selection bias in online surveys. *PloS One*, 15(4), e0231500.

Gondara, L. and Wang, K. (2018). MIDA: Multiple imputation using denoising autoencoders. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, 260-272.

Graves, A., Mohamed, A.-r. and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 6645-6649.

Hastie, T., Tibshirani, R., Friedman, J.H. and Friedman, J.H. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, 2.

Hengl, T., Nussbaum, M., Wright, M.N., Heuvelink, G.B. and Gräler, B. (2018). Random forest as a generic framework for predictive modeling of spatial and spatio-temporal variables. *PeerJ*, 6, e5518.

Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N. and Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6), 82-97.

Kim, J.K., Park, S., Chen, Y. and Wu, C. (2021). Combining non-probability and probability survey samples through mass imputation. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 184(3), 941-963.

Kim, J.-K. and Tam, S.-M. (2021). Data integration by combining big data and survey sample data for finite population inference. *International Statistical Review*, 89(2), 382-401.

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25.

LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.

Lee, S. (2006). Propensity score adjustment as a weighting scheme for volunteer panel web surveys. *Journal of Official Statistics*, 22(2), 329.

Lee, S. and Valliant, R. (2009). Estimation for volunteer panel web surveys using propensity score adjustment and calibration adjustment. *Sociological Methods & Research*, 37(3), 319-343.

Lohr, S.L. and Raghunathan, T.E. (2017). Combining survey data with other data sources. *Statistical Science*, 32(2), 293-312.

Padilla, O., Tansey, W. and Chen, Y. (2020). Quantile regression with relu networks: Estimators and minimax rates. *arXiv Preprint arXiv:2010.08236*, 13.

Rao, J.N.K. (2021). On making valid inferences by integrating data from surveys and other sources. *Sankhyā B*, 83(1), 242-272.

Rasamoelina, A.D., Adjailia, F. and Sinčák, P. (2020). A review of activation function for artificial neural network. *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, IEEE, 281-286.

Rivers, D. (2007). Sampling for web surveys. *Joint Statistical Meetings*, 4.

Sarikaya, R., Hinton, G.E. and Deoras, A. (2014). Application of deep belief networks for natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4), 778-784.

Schmidt-Hieber, J. (2020). Nonparametric regression using deep neural networks with relu activation function. *The Annals of Statistics*, 48(4), 1875-1897.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1-9.

Xu, K., Zhang, M., Li, J., Du, S.S., Kawarabayashi, K.-i. and Jegelka, S. (2020). How neural networks extrapolate: From feedforward to graph neural networks. *arXiv Preprint arXiv:2009.11848*.

Yang, S., Kim, J.K. and Hwang, Y. (2021). Integration of data from probability surveys and big found data for finite population inference using mass imputation. *Survey Methodology*, 47(1), 29-58. Paper available at https://www150.statcan.gc.ca/n1/en/pub/12-001-x/2021001/article/00004-eng.pdf.

Yoon, J., Jordon, J. and Schaar, M. (2018). Gain: Missing data imputation using generative adversarial nets. *International Conference on Machine Learning*, PMLR, 5689-5698.

Zhong, Q., Mueller, J. and Wang, J.-L. (2022). Deep learning for the partially linear cox model. *The Annals of Statistics*, 50(3), 1348-1375.