SSC Annual Meeting, June 2025
Proceedings of the Survey Methods Section

Deep Neural Networks for Doubly Robust Estimation with Nonprobability Survey Samples

Yufang Dai[1] Shihua Luo [2] Wendy Luo [3] Zilin Wang [4] and Xuewen Lu [5]

ABSTRACT

The integration of probability and non-probability samples is an emerging research area in survey sampling. However, nonprobability samples often lack population representativeness, making it challenging to infer finite population characteristics. Traditional inverse propensity score weighting methods, typically reliant on parametric models, fail to capture the complex nonlinear relationships between auxiliary variables and participation rates. To address this, we propose a Deep Neural Network (DNN) model to estimate the propensity score, the weight parameters are optimized via the ADAM algorithm, enabling accurate propensity score estimation for nonprobability sample units. Using the DNN estimated propensity score, we construct a doubly robust estimator for the finite population mean. Simulation studies and empirical analyses demonstrate that our method outperforms traditional parametric models and provides a more reliable tool for statistical inference in complex survey settings.

KEY WORDS: ADAM algorithm; Deep neural networks; Doubly robust estimator; Probability and non-probability samples; Propensity score

RÉSUMÉ

L'intégration d'échantillons probabilistes et non probabilistes est un domaine de recherche émergent dans le domaine de l'échantillonnage pour les enquêtes. Cependant, les échantillons non probabilistes manquent souvent de représentativité de la population, ce qui rend difficile l'inférence des caractéristiques d'une population finie. Les méthodes traditionnelles de pondération par score de propension inverse, qui reposent généralement sur des modèles paramétriques, ne permettent pas de saisir les relations non linéaires complexes entre les variables auxiliaires et les taux de participation. Pour remédier à cela, nous proposons un modèle de réseau neuronal profond (DNN) pour estimer le score de propension. Les paramètres de pondération sont optimisés via l'algorithme ADAM, ce qui permet une estimation précise du score de propension pour les unités d'échantillonnage non probabilistes. À l'aide du score de propension estimé par le DNN, nous construisons un estimateur doublement robuste pour la moyenne de la population finie. Des études de simulation et des analyses empiriques démontrent que notre méthode surpasse les modèles paramétriques traditionnels et fournit un outil plus fiable pour l'inférence statistique dans des contextes d'enquête complexes.

MOTS CLÉS : Algorithme ADAM; Réseaux de neurones profonds; Estimateur doublement robuste; Échantillons probabilistes et non probabilistes; Score de propension

1 INTRODUCTION

Probability sampling is the mainstream method of data collection in official statistics and social research due to the perfect sampling design, following the random principle to select representative samples, and is regarded as the gold standard for finite population inference. However, changes in social structure and lifestyle make traditional probability sample collection face many challenges, such as low response rates, escalating costs, and

[1]Yufang Dai, Department of Mathematics and Statistics, University of Calgary, 2500 University Drive NW Calgary, Canada, T2N 1N4; School of Statistics and Data Science, Jiangxi University of Finance and Economics, Xinjian District, NO.169 Shuanggang East Ave, China, 330013, yufang.dai@ucalgary.ca
[2]Shihua Luo, School of Statistics and Data Science, Jiangxi University of Finance and Economics, Xinjian District, NO.169 Shuanggang East Ave, China, 330013, luoshihua@aliyun.com
[3]Wendy Luo, Biostatistics Division Centre for Global Health, University of Toronto, Canada, M5T 3M7, wendy.lou@utoronto.ca
[4]Zilin Wang, Department of Mathematics, Wilfrid Laurier university, 75 University Avenue West, Canada, N2L 3C5, zwang@wlu.ca
[5]Xuewen Lu, Department of Mathematics and Statistics, University of Calgary, 2500 University Drive NW Calgary, Canada, T2N 1N4, xlu@ucalgary.ca

time-consuming (Keiding and Louis, 2016). To address the challenges, an increasing number of studies have utilized multiple data sources such as web-based survey panels, satellite information, and mobile sensor data generated by the development of internet technology and the popularity of smart devices. Such data are collectively referred to as nonprobability samples, and Kitchin (2015) stated that nonprobability survey data serve as an alternative and complement to official statistical survey data. However, the presence of coverage errors and selectivity bias due to not satisfying the principle of random selection also poses additional challenges in inferring finite population characteristics. Since both sampling paradigms have certain shortcomings, in order to fully utilize the sample information, it is often considered to integrate probability samples and nonprobability samples to infer the population characteristics, so as to compensate for the limitations of each independent paradigm.

Exisiting methods for data integration can be categorized into three types. The first approach is calibration weighting. Deville and Särndal (1992) first proposed the concept of calibration estimator, assuming that the total mean of auxiliary variables is known, the weight can be calculated using the known information of auxiliary variables. Wu and Sitter (2001) first proposed the concept of model calibration estimation by constructing a superpopulation model between study variables and auxiliary variables and applying full auxiliary information to calibration estimation.This technique forces the moments or the empirical distribution of auxiliary variables to be the same between the probability sample and the nonprobability sample, so that after calibration the weighted distribution in the nonprobability sample appears similar to that in the target population (DiSogra et al., 2011). Nevertheless, a critical limitation of this approach lies in its reliance on prior knowledge of population-level information, which is often unavailable in practical applications. The second approach is mass imputation, in the framework of mass imputation, the nonprobability sample is used as a training dataset, these models are subsequently applied to estimate the missing values for each unit in the probability sample. Rivers (2007) proposed using the value of the nearest neighbor for mass imputation, but did not discuss its properties theoretically. Kim et al. (2021) proposed using regression models for mass imputation and discussed its statistical properties, including consistent variance estimation. However, such a parametric mass imputation method is subject to model misspecification bias. The third approach is propensity score adjustment. In this approach, the probability of a unit being selected into the nonprobability sample, which is referred to as the propensity or sampling score, is modeled and estimated for all units in the nonprobability sample. Valliant and Dever (2011) estimated participation rates by fitting a logistic regression model to the combined nonprobability sample and probability sample. Sample weights for the probability sample were scaled by a constant so that the scaled probability sample was assumed to represent the complement of the nonprobability sample. Each unit in the nonprobability was assigned a weight of one. The results show that the sum of the scaled weights of the probability and nonprobability combined samples is an estimate of the population size, but the estimator is biased especially when the participation rate of the nonprobability sample is large. Chen et al. (2020) estimated the participation rate by manipulating the log-likelihood estimating equation. The resulting estimator is consistent and approximately unbiased regardless of the magnitude of participation rates. However, both mass imputation and propensity score adjustment rely on the settings of the model, model misspecification leads to biased estimation. In order to enhance the efficiency and robustness of estimation, Chen et al. (2020) proposed construction of doubly robust (DR) estimators of the finite population mean using the estimated propensity score as well as outcome regression model. While DR estimators guarantee consistency if at least one of the two models is correctly specified, they remain susceptible to bias when both models are misspecified. In their method, the relationship between Y and X is restricted to a prespecified linear functional form. However, parametric procedures may suffer from bias if the functional form is misspecified or if the vector X fails to include interactions or predictors. In contrast, nonparametric methods have the ability to capture nonlinear trends and tend to be robust. In the last decade, the interest in machine learning methods has been growing, and these data-driven methods provide flexible tools for obtaining accurate predictions.

In many application, the neural network has proven to be powerful for approximating complex function by providing accurate approximations of continuous functions (Leshno et al., 1993). Under some smoothness and structural assumptions, Schmidt-Hieber (2020) showed that DNN estimators may circumvent the curse of dimensionality and achieve the optimal minimax rate of convergence. With limited samples, however, a complex DNN can still lead to overfitting (Srivastava et al., 2014). Early stopping during training (Li et al., 2020), and adding dropout layers, have been proposed to address overfitting, but this method has not yet been applied in the propensity score adjustment method of statistical surveys.

To fill this gap, we propose a doubly robust estimation procedure that harness the representativeness of the probability sample and the outcome information in the nonprobability sample. Our major contributions are using

deep neural networks (DNNs) as a nonparametric approach to estimate propensity score and training the neural network using the ADAM optimization algorithm. The proposed method constructs a doubly robust estimator by combining the propensity score estimated via deep neural networks (DNN) with the outcome regression model's predictions. Crucially, this approach maintains strong estimation performance even under model misspecification, not only when the propensity score model is incorrectly specified but also in the more challenging scenario when both the propensity score and outcome regression models are misspecified. As a result, the method significantly enhances both the accuracy and robustness of population mean estimation.

The paper proceeds as follows. Section 2 introduces the basic setup of the problem and three estimators for estimating the finite population mean. Section 3 presents the proposed procedure for estimating the propensity scores and describes the computation ADAM algorithm for solving loss function and use the doubly robust estimator for estimating the finite population mean of the response variable. Section 4 reports simulation results that illustrate the finite sample performance of the method. Section 5 presents an application to analyze a nonprobability survey sample collected by the Pew Research Center(PRC) with auxiliary information from the Behavioural Risk Factor Surveillance System.

2 BASIC SET-UP

2.1 Notation: two samples

Let $\mathcal{FP} = \{1, 2, ..., N\}$ be the index set of units for the finite population with size N. For each unit i (where $i = 1, 2, ..., N$), there are corresponding values of the auxiliary variable x_i and the response variable y_i. Under the design-based framework, the set of finite population values $\mathcal{F}_N = (x_i, y_i), i \in \mathcal{FP}$ is treated as fixed. The parameter of interest is finite population mean $\mu_y = N^{-1} \sum_{i=1}^{N} y_i$ of the response variable.

Suppose a nonprobability sample S_A of size n_A is selected from $\mathcal{FP}$ by a self-selection mechanism. Let $\{(x_i, y_i), i \in S_A\}$ be the dataset from the nonprobability sample. Define the indicator variable for unit i in the sample S_A as $R_i = I(i \in S_A)$, that is, $R_i = 1$ if $i \in S_A$ and $R_i = 0$ if $i \notin S_A$, $i = 1, \ldots, N$. the underlying participation probability of nonprobability sample for a finite population unit i is defined as $\pi_i^A = E_q(R_i|x_i, y_i) = P_q(R_i = 1|x_i, y_i), i = 1, 2, ..., N$, where the subscript q refers to the propensity score model for the selection mechanism for the sample S_A. The corresponding implicit nonprobability sample weight is $w_i = 1/\pi_i^A$ for $i \in \mathcal{FP}$.

Consider a reference probability sample, denoted as S_B, independent of the nonprobability sample S_A. The probability sample S_B is assumed to be independently drawn from the same finite population under a known probability sampling design. Let $\{(x_i, d_i^B), i \in S_B\}$ be the dataset from the probability sample, where x_i is available in an existing survey, $d_i^B = 1/\pi_i^B$ are the survey weights and $\pi_i^B = P(i \in S_B)$ are the inclusion probabilities under the probability sampling design for the sample S_B. Note that the response variable y is not observed in the reference sample S_B.

Define the combined sample $S = S_A \cup S_B$, $n = n_A + n_B$. It is assumed that there are no overlapping elements between the two samples.

Table 1: Two sources of data

Sample	**Sampling weight** π^{-1}	**Covariate** X	**Study variable** Y
Nonprobability sample	?	$\surd$	$\surd$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
n_A	?	$\surd$	$\surd$
Probability sample	$\surd$	$\surd$	?
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$n_A + n_B$	$\surd$	$\surd$	?

[1]Sample A is a nonprobability sample, and sample B is a probability sample. '$\surd$' and '?' indicate observed and unobserved data respectively.

2.2 Existing estimators

In practice, the sampling score function π_i^A for nonprobability samples and the outcome mean function $m(x)$ for probability samples are typically unknown and need to be estimated through modeling. To address this issue, many researchers have modelled π_i^A and $m(x)$ via parametric specifications $\pi_i^A(x^\top\theta)$ and $m(x^\top\beta)$, where θ and β denote vectors for unknown model parameters. The statistical literature has developed multiple estimators for the population parameter μ_y, each employing distinct modeling frameworks and estimation methodologies. The following discussion examines several representative approaches, analyzing their respective theoretical properties and practical limitations within the context of survey sampling. Following Chen et al. (2020), we define three estimators of the population parameter μ_y as follows.

2.2.1 Outcome Regression Estimator

$$\hat{\mu}_{\text{REG}} = \frac{1}{\hat{N}^B}\sum_{i\in\mathcal{S}_B} d_i^B \hat{y}_i, \tag{1}$$

where $d_i^B = 1/\pi_i^B$ denotes the design weight for unit i in the probability samples S_B; $\hat{N}^B = \sum_{i\in S_B} d_i^B$ is the estimated population size using the probability sample; $\hat{y}_i$ is the predicted value of the interest variable for unit i, obtained from a fitted regression model.

Suppose that the finite population $(x_i, y_i), i \in U$ can be regarded as a random sample drawn from the underlying model

$$y_i = m(x_i) + \varepsilon_i, i = 1, \ldots, N,$$

where $m(x_i) = E(y_i|x_i)$, which can take a parametric form such as $m(x_i) = x_i^\top\beta$ or a specified non-linear parametric form. The error terms ε_i are independent with $E(\sigma_i) = 0$ and $V(\varepsilon_i) = v(x_i)\sigma^2$. The dataset from the nonprobability sample can be used to build the model. For the linear regression model $m(x_i) = x_i^\top\beta$ and the homogeneous variance structure $v(x_i) = 1$, the least square estimator β is given by

$$\hat{\beta} = (\sum_{i\in S_A} x_i x_i^\top)^{-1}(\sum_{i\in S_A} x_i y_i).$$

In probability sample, the predicted value for y_i with an associated x_i from the reference probability sample S_B is given by $\hat{y}_i = x_i^\top\hat{\beta}$. The validity of the regression estimator $\hat{\mu}_{REG}$ relies on a correct specification of $m(x^\top\beta)$ and the consistency of β. If $m(x^\top\beta)$ is misspecified or β is inconsistent, $\hat{\mu}_{REG}$ can be biased. In this paper, our model for $m(x)$ can be linear or non-linear parametric function, but the model for π^A is an unknown nonparametric function only.

2.2.2 Inverse Probability of Sampling Score Weighting Estimator

$$\hat{\mu}_{\text{IPW}} = \frac{1}{\hat{N}^A}\sum_{i\in\mathcal{S}_A} \frac{1}{\hat{\pi}_i^A} y_i, \tag{2}$$

where $\hat{N}^A = \sum_{i\in S_A} 1/\hat{\pi}_i^A$; y_i is the known value of the variable of interest in the nonprobability samples, $\hat{\pi}_i^A$ is the probability calculated from the logistic regression model, the specific estimation process is given in Section 3.1.

2.2.3 Doubly Robust Estimator

$$\hat{\mu}_{\text{DR}} = \frac{1}{\hat{N}^A}\sum_{i\in\mathcal{S}_A} \frac{1}{\hat{\pi}_i^A}\{y_i - m(x_i, \hat{\beta})\} + \frac{1}{\hat{N}^B}\sum_{i\in\mathcal{S}_B} \frac{1}{\pi_i^B} m(x_i, \hat{\beta}), \tag{3}$$

where $\hat{N}^A = \sum_{i\in S_A} 1/\hat{\pi}_i^A$ and $\hat{N}^B = \sum_{i\in S_B} 1/\pi_i^B$ denote the estimated population sizes based on the nonprobability sample S_A (using estimated propensity scores) and the probability sample S_B (using known design weights), respectively. The doubly robust estimator $\hat{\mu}_{DR}$ possesses an important theoretical property: under fixed-dimensional covariates X, it remains consistent as long as either the propensity score model $\pi_i^A(x^\top\theta)$ or the outcome model $m(x^\top\beta)$ is correctly specified, without requiring both to hold simultaneously. This double robustness property enhances the estimator's reliability in practical applications where model misspecification may occur.

3 METHODOLOGY

In this section, we start from considering the hypothetical case where x_i is observed for all units in the finite population $\mathcal{FP}$ while y_i is only observed for the nonprobability sample S_A, and the logit function of the propensity score π^A is an unknown nonparametric function, not necessarily a linear function of the predictors, a comprehensive and meticulous procedure for the estimation of the population mean is presented. To begin with, a Deep Neural Network (DNN) model is used for the purpose of estimating the propensity score. The weight parameters within this model are optimized by means of the ADAM algorithm. This optimization process endows the model with the ability to conduct accurate propensity score estimation for nonprobability sample units. Finally, we discuss the construction of Doubly robust Deep Neural Network (DDNN) estimators for the finite population mean. These estimators are constructed by integrating the estimated propensity scores and the outcome regression model.

3.1 Estimation of Propensity Scores by DNN

Let's assume that the Deep Neural Network model for the propensity scores π_i, which is a function of x_i, i.e., $\pi_i = \pi(x_i)$ is given by

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = g(x_i), \quad \text{for } i \in \mathcal{FP}, \tag{4}$$

where $g(x_i)$ is an unknown real-valued nonparametric function, and x_i is a vector of covariates associated with the ith unit in the finite population. The population likelihood function for π_i is defined as

$$L(g) = \prod_{i=1}^{N} \{\pi_i\}^{R_i} \{1-\pi_i\}^{1-R_i}. \tag{5}$$

The maximum likelihood estimator of π_i is computed as $\hat{\pi}_i = 1/(1+e^{-\hat{g}(x)})$, where $\hat{g}(x)$ is the function that maximizes the log-likelihood function

$$\begin{aligned} \ell(g) &= \sum_{i=1}^{N} \left[\delta_i \log \pi_i + (1-\delta_i)\log(1-\pi_i)\right] \\ &= \sum_{i\in S_A} \log\frac{\pi_i}{1-\pi_i} + \sum_{i=1}^{N} \log(1-\pi_i). \end{aligned} \tag{6}$$

However, the log-likelihood function specified in (6) cannot be directly utilized in practice. This is because we do not have access to the covariate information x_i for all units in the finite population, and N is often unknown. To address this issue, instead of using $\ell(g)$, we compute $\hat{g}$ by maximizing the following pseudo-log-likelihood function

$$\ell^*(g) = \sum_{i\in S_A} \log\frac{\pi_i}{1-\pi_i} + \sum_{i\in S_B} d_i^B \log(1-\pi_i), \tag{7}$$

where the population total $\sum_{i=1}^{N}\log(1-\pi_i)$ in $\ell(g)$ is replaced by the Horvitz-Thompson (HT) estimator $\sum_{i\in S_B} d_i^B \log(1-\pi_i)$ using the reference sample S_B.

Under the DNN model for the propensity scores, the pseudo-log-likelihood function (7) can be further simplified as

$$\ell^*(g) = \sum_{i\in S_A} g(x_i) - \sum_{i\in S_B} d_i^B \log(1+e^{g(x_i)}). \tag{8}$$

Consequently, the loss function is defined as $-\ell^*(g)$. The optimal function $\hat{g}$ can be obtained by applying the ADAM optimization algorithm, which iteratively updates the parameters of the DNN model to maximize the pseudo-log-likelihood function. After obtaining $\hat{g}(x_i)$, we can get $\hat{\pi}_i^A$. According to the IPW estimator formula, we can obtain the DNN estimator as follows:

$$\hat{\mu}_{\text{DNN}} = \frac{1}{\hat{N}^A} \sum_{i\in\mathcal{S}_A} \frac{1}{\hat{\pi}_i^A} y_i. \tag{9}$$

We now briefly introduce the relevant concepts of DNN, for further details, see Goodfellow et al. (2016). Let $\mathbb{N}_+$ be the set of all positive natural numbers, given $K \in \mathbb{N}_+$ and $p = (p_0, \ldots, p_K, p_{K+1})^\top \in \mathbb{N}_+^{K+2}$, a $(K+1)$-layer DNN with layer-width p is a composite function $g : \mathbb{R}^{p_0} \to \mathbb{R}^{p_{K+1}}$ recursively defined as

$$\begin{aligned} g(x) &= W_K g_K(x) + v_K, \\ g_K(x) &= \sigma\left(W_{K-1} g_{K-1}(x) + v_{K-1}\right), \ldots, \\ g_1(x) &= \sigma\left(W_0 x + v_0\right). \end{aligned} \tag{10}$$

Here, K denotes the depth of the network and vector p lists the width of each layer (p_0 is the dimension of the input variable, p_1,...,p_K are the dimensions of the K hidden layers, and p_{K+1} is the dimension of the output layer). The matrices $W_k \in \mathbb{R}^{p_{k+1} \times p_k}$ and vectors $v_k \in \mathbb{R}^{p_{k+1}}$ (for $k = 0, \ldots, K$) are the parameters of this DNN g. The matrix entries $(W_k)_{i,j}$ are the weight linking the jth neuron in layer k to the ith neuron in layer $k+1$, and the vector entries $(v_k)_i$ represent a shift term associated with the ith neuron in layer $k+1$. The function σ is a known deterministic activation function that performs a component-wise operation on vectors, that is, $\sigma\left[(x_1, \ldots, x_{p_k})^\top\right] = (\sigma(x_1), \ldots, \sigma(x_{p_k}))^\top$, which thus gives $g_k = (g_{k1}, \ldots, g_{kp_k})^\top : \mathbb{R}^{p_{k-1}} \to \mathbb{R}^{p_k}$ for $k = 1, \ldots, K$. Usually, a simple nonlinear function can be set as the activation function σ that connects adjacent layers. While many choices of activation function are considered in deep learning, the most popular one is the rectified linear unit (ReLU) in Nair and Hinton (2010): $\sigma(x) = \max\{x, 0\}$.

We consider a class of DNN:

$$\begin{aligned} \mathcal{G}(K, p) = \Big\{ & g : g \text{ is a DNN with } (K+1) \text{ layers and width vector } p, \\ & \max\left\{\|W_k\|_\infty, \|v_k\|_\infty\right\} \le 1, \text{ for all } k = 0, \ldots, K \Big\}, \end{aligned} \tag{11}$$

where $\|\cdot\|_\infty$ denotes the sup-norm of matrix or vector. Empirically the size of the learned matrices W_k and vectors v_k are rarely large when the size of initial matrices and vectors used to initialize stochastic gradient training are relatively small (as is typically the case). Thus we just consider the DNNs whose matrices and vectors are bounded by one. In practice, a deep feedforward network with fully-connected layers contains a huge number of parameters, which can lead to overfitting. This issue can be mitigated by pruning weights, which reduces the total number of nonzero parameters such that the network's layers are only sparsely connected (Han et al., 2015; Srinivas et al., 2017; Schmidt-Hieber, 2020). Following a similar methodology, we consider, for $s \in \mathbb{N}_+$ and $D > 0$, a class of sparse neural networks

$$\mathcal{G}(K, s, p, D) := \left\{ g \in \mathcal{G}(K, p) : \sum_{k=1}^{K} \{\|W_k\|_0 + \|v_k\|_0\} \le s,\ \|g\|_\infty \le D \right\}, \tag{12}$$

where $\|\cdot\|_0$ is the number of nonzero entries of matrix or vector, and $\|g\|_\infty$ is the sup-norm of function g.

To estimate the function g, we utilize the ADAM optimization algorithm to minimize the loss function $-\ell^*(g)$. The estimation is performed using the proposed DNN model, with a set of trainable parameters, including all the weight matrices and bias vectors, denoted as Θ. The ADAM algorithm, a variant of stochastic gradient descent, iteratively updates the parameter estimates by adaptively computing the first and second moments of the stochastic gradients derived from the empirical loss (Kingma and Ba, 2014).

For parameter initialization, the bias terms are set to zero, while the weights are initialized using the Xavier initialization scheme, which ensures stable gradient propagation during training (Nair and Hinton, 2010). To enhance numerical stability, a small constant $\epsilon_0 > 0$ is incorporated into the denominator. At each iteration, the adjustment of parameters is governed by the adaptive estimates of the gradient moments. Empirical observations suggest that convergence can often be achieved within a limited number of iterations. Furthermore, excessive iterations may induce overfitting in the DNN model. To mitigate this risk, early stopping is employed as a regularization strategy, which not only prevents overfitting but also promotes the consistency of the trained

network (Ji et al., 2021).

Algorithm 1: ADAM Algorithm

Input: Initial parameters $\theta^{(0)}$, learning rate γ, decay rates r_1, r_2, small constant ϵ_0, threshold l

Initialize $m^{(0)} \leftarrow 0$, $v^{(0)} \leftarrow 0$, $t \leftarrow 1$, $\boldsymbol{\Theta}^{(0)}$

while $\|\hat{\boldsymbol{\Theta}}^{(t)} - \hat{\boldsymbol{\Theta}}^{(t-1)}\|_2 > l$ **do**

  Compute the gradient $g_t \leftarrow \nabla_\theta f(\theta_{t-1})$;

  Update first moment: $m^{(t)} \leftarrow r_1 m^{(t-1)} + (1-r_1) g_t$;

  Update second moment: $v^{(t)} \leftarrow r_2 v^{(t-1)} + (1-r_2) g_t^2$;

  Bias correction: $\hat{m}^{(t)} \leftarrow \frac{m^{(t)}}{1-r_1^t}$;

  Bias correction: $\hat{v}^{(t)} \leftarrow \frac{v^{(t)}}{1-r_2^t}$;

  Update parameters: $\hat{\boldsymbol{\theta}}^{(t)} \leftarrow \hat{\boldsymbol{\theta}}^{(t-1)} - \gamma \frac{\hat{m}^{(t)}}{\sqrt{\hat{v}^{(t)}}+\epsilon_0}$;

  $t \leftarrow t+1$;

end

Output: Optimized parameters $\boldsymbol{\theta}^{(t)}$

3.2 Doubly Robust Estimation by DNN

By leveraging prediction approaches to derive the outcome variable y_i in the probability sample and employing DNN to estimate the propensity scores $\hat{\pi}_i^A$ in the nonprobability sample, we construct a DDNN estimator based on the doubly robust (DR) framework. The DDNN estimator given below provides a consistent and efficient approach for estimating the finite population mean.

$$\hat{\mu}_{\text{DDNN}} = \frac{1}{\hat{N}^A} \sum_{i \in \mathcal{S}_A} \frac{1}{\hat{\pi}_i^A} \{y_i - m(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}})\} + \frac{1}{\hat{N}^B} \sum_{i \in \mathcal{S}_B} \frac{1}{\pi_i^B} m(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}). \tag{13}$$

4 SIMULATION STUDIES

In the following simulation studies, a finite population of size $N = 20000$ is assumed, in which the response variable y and auxiliary variables x following the regression model

$$y_i = 2 + x_{1i} + x_{2i} + x_{3i} + x_{4i} + \sigma\varepsilon_i, \quad i = 1, \ldots, N,$$

where auxiliary variables $x_{1i} = z_{1i}$, $x_{2i} = z_{2i} + 0.3x_{1i}$, $x_{3i} = z_{3i} + 0.2(x_{1i} + x_{2i})$, $x_{4i} = z_{4i} + 0.1(x_{1i} + x_{2i} + x_{3i})$ with $z_{1i} \sim Bernoulli(0.5)$, $z_{2i} \sim Uniform(0, 2)$, $z_{3i} \sim Exponential(1)$, $z_{4i} \sim \chi^2(4)$. The error term ε_i are independent and identically distributed as $N(0, 1)$. The value of σ is chosen by controlling the correlation coefficient ρ between y and $x^\top\beta$ at 0.3, 0.5, 0.8. The true propensity score π_i^A for the nonprobability sample S_A follow the logistic regression model

$$\begin{aligned}\log\left(\frac{\pi_i^A}{1-\pi_i^A}\right) = \theta_0 &+ 0.05x_{1i}x_{2i} + 0.1x_{2i}^2 + 0.05x_{3i}x_{4i} \\ &+ 0.08\sin(0.3x_{3i}) + 0.05\ln(1 + x_{2i} + x_{4i}),\end{aligned}$$

which is a non-linear function of $x_i = (x_{1i}, x_{2i}, x_{3i}, x_{4i})^\top$, where θ_0 is chosen such that $\sum_{i=1}^N \pi_i^A = n_A$ with given target sample size n_A.

We consider two scenarios for model specifications: (i) the outcome regression model is correctly specified but the propensity score model is misspecified, denoted as "TF".The logistic regression model is misspecified as $\log(\pi_i^A/(1-\pi_i^A)) = \theta_0 + \theta_1 x_{1i} + \theta_2 x_{2i} + \theta_3 x_{3i} + \theta_4 x_{4i}$. (ii) Both models are misspecified, denoted as "FF". The regression model is misspecified as $y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i}$, with x_{4i} omitted from the model. The settings for the logistic regression model remain consistent with (i). For a given estimator $\hat{\mu}$, its performance is evaluated through the relative bias and the mean squared error computed as

$$\%RB = \frac{1}{B}\sum_{b=1}^{B} \frac{\hat{\mu}^{(b)} - \mu_y}{\mu_y} \times 100, \qquad MSE = \frac{1}{B}\sum_{b=1}^{B} (\hat{\mu}^{(b)} - \mu_y)^2,$$

where B is the number of simulation runs, in our study, $B = 500$.

Table 2: Simulated %RB and MSE of the estimators of μ_y ($n_A = 500$, $n_B = 1000$)

Models	Estimator	$\rho = 0.30$		$\rho = 0.50$		$\rho = 0.80$	
		%RB	MSE	%RB	MSE	%RB	MSE
TF	$\hat{\mu}_A$	72.89	46.31	74.67	47.97	75.89	49.15
	$\hat{\mu}_{\text{REG}}$	-1.21	0.81	-0.67	0.25	-0.32	0.06
	$\hat{\mu}_{\text{IPW}}$	-5.77	18.35	-5.87	7.47	-5.92	3.80
	$\hat{\mu}_{\text{DR}}$	0.97	13.82	0.73	4.06	0.26	0.81
	$\hat{\mu}_{\text{DNN}}$	1.24	2.90	2.10	1.70	2.76	1.37
	$\hat{\mu}_{\text{DDNN}}$	-0.90	2.27	-0.46	0.62	-0.22	0.15
FF	$\hat{\mu}_A$	72.86	46.28	74.66	47.97	75.89	49.15
	$\hat{\mu}_{\text{REG}}$	77.18	51.96	79.06	53.81	80.35	55.12
	$\hat{\mu}_{\text{IPW}}$	-6.32	17.79	-5.80	7.48	-5.92	3.80
	$\hat{\mu}_{\text{DR}}$	-23.75	18.82	-24.21	10.07	-24.84	7.05
	$\hat{\mu}_{\text{DNN}}$	2.00	3.00	2.77	1.77	2.88	1.31
	$\hat{\mu}_{\text{DDNN}}$	-0.21	2.68	-0.26	1.49	0.17	0.88

The simulation results for $B = 500$, $n_A = 500$ and $n_B = 1000$ are reprot in Table 2. Major observations from Table 2 can be summarized as follows. (1). Under the correctly specified regression model and a misspecified logistic regression model ("TF"), the naive estimator $\hat{\mu}_A = \bar{y}$ exhibits significant positive bias and high MSE across all ρ values. In terms of bias, the $\hat{\mu}_{REG}$ estimator demonstrates excellent performance under this setting, with near-negligible bias and the lowest MSE among all estimators. In contrast, the $\hat{\mu}_{IPW}$ estimator suffers from substantial negative bias and relatively high MSE, due to its exclusive reliance on the misspecified propensity score model. The $\hat{\mu}_{DNN}$ estimator achieves improved performance relative to $\hat{\mu}_{IPW}$ estimator, benefitting from the flexibility of DNN in approximating complex functional relationships. Notably, the $\hat{\mu}_{DR}$ and $\hat{\mu}_{DDNN}$ estimator achieve nearly unbiased results, as only one of the two underlying models is misspecified. (2). Under the misspecified regression model and misspecified logistic regression model ("FF"), the performance of both the $\hat{\mu}_{REG}$ and $\hat{\mu}_{IPW}$ estimator deteriorates significantly, exhibiting substantial bias and increased variability. The $\hat{\mu}_{DR}$ estimator, which relies on the validity of at least one of the two models, also fails to yield accurate estimates in this setting, as the double robustness property no longer holds. Notably, the $\hat{\mu}_{DDNN}$ estimator demonstrates relatively improved performance. This can be attributed to the flexibility of deep neural networks in approximating complex, unknown functions, particularly when estimating the propensity scores for nonprobability samples. In this context, the DNN component may effectively capture latent structures within the data, thereby partially compensating for model misspecification and resulting in more accurate and stable estimates compared to the conventional methods.

In summary, the $\hat{\mu}_{DDNN}$ estimator exhibits consistently favorable performance not only under the "TF" scenario, where only one of the nuisance models is misspecified, but also under the more challenging "FF" setting, where both the outcome and propensity score models are misspecified. While traditional estimators such as $\hat{\mu}_{REG}$, $\hat{\mu}_{IPW}$, and $\hat{\mu}_{DR}$ suffer significant performance loss under full model misspecification, $\hat{\mu}_{DDNN}$ remains robust due to the expressive capacity of deep neural networks to approximate complex functional relationships. This suggests that $\hat{\mu}_{DDNN}$ not only inherits the double robustness property when applicable, but also offers an additional layer of protection against misspecification through flexible, data-adaptive modeling, thereby providing a promising alternative in practical applications where model assumptions may be violated.

5 APPLICATION

In this section, we illustrate the application of the proposed method using a dataset collected by the Pew Research Center (PRC) in 2015 (*http://www.pewresearch.org*). This dataset comprises nine nonprobability samples obtained from eight different vendors, each employing distinct and largely undocumented strategies for panel recruitment, sampling, participant incentives, and related procedures. For analytical purposes, we aggregate these into a single nonprobability sample with a total sample size of $n_A = 9301$, hereafter referred to as the PRC dataset. To provide auxiliary population-level information, we incorporate a reference probability sample from the 2015 Behavioral Risk Factor Surveillance System (BRFSS) survey (*https://www.cdc.gov/brfss/index.html*), which contains 441,456 respondents.

We employ the proposed method to estimate the population means of seven outcome variables, as detailed in Table 3. The first six outcomes are binary, while the final one is continuous. Table 3 presents the estimation results based on a reduced set of covariates that are available in both datasets. We use these covariates include age, gender, race, origin, region, marital status, employment status, and education(high school or less , bachelor's degree and above). The variable "age" is treated as a continuous variable in the analysis. Table 3 presents the estimated population means for various response variables using different methods. Overall, $\hat{\mu}_{DDNN}$ demonstrates remarkable stability and consistency across all measures, and the results of the $\hat{\mu}_{DDNN}$ estimator and the $\hat{\mu}_{REG}$ estimator are very close. Compared with the conventional $\hat{\mu}_{IPW}$ estimator, $\hat{\mu}_{DDNN}$ avoids extremely low estimates observed in certain variables, e.g., in 'Talked with neighbors frequently', $\hat{\mu}_{IPW} = 39.88$ vs. $\hat{\mu}_{DDNN} = 44.83$. In addition, it also avoids extremely high estimations in some variables, e.g., in 'Expressed opinions at a government level', $\hat{\mu}_{IPW} = 28.39$ vs. $\hat{\mu}_{DDNN} = 22.34$. These results indicate that $\hat{\mu}_{DDNN}$ effectively leverages both flexibility and robustness when handling survey data with complex nonlinear structures, producing superior performance in population mean estimation.

Table 3: Estimated population mean of y

Response variable y	$\hat{\mu}_A$	$\hat{\mu}_{REG}$	$\hat{\mu}_{IPW}$	$\hat{\mu}_{DR}$	$\hat{\mu}_{DNN}$	$\hat{\mu}_{DDNN}$
%Talked with neighbors frequently	46.36	44.97	39.88	43.99	46.12	44.83
%Tended to trust neighbors	58.66	52.16	52.85	51.66	48.21	52.07
%Expressed opinions at a government level	26.90	21.27	28.39	28.79	26.78	22.34
%Voted local elections	74.60	64.46	64.31	66.70	74.47	64.05
%Participated in school groups	20.83	18.78	16.40	15.44	17.12	19.90
%Participated in service organizations	13.98	12.27	15.72	15.30	12.45	12.08
Days had at least one drink last month	5.25	4.45	4.34	4.74	5.33	4.37

REFERENCES

Anthony, M., & Bartlett, P. L. (2009). *Neural network learning: Theoretical foundations*. Cambridge University Press, Cambridge.

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

Deville, J.-C., & Särndal, C.-E. (1992). Calibration estimators in survey sampling. *Journal of the American Statistical Association*, 87, 376–382.

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*, pp. 4501-4515.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press, Cambridge, MA.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics* (pp. 249–256).

Han, S., Pool, J., Tran, J., & Dally, W. (2015). Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems* (Vol. 28, pp. 1135–1143).

Ji, Z., Li, J., & Telgarsky, M. (2021). Early-stopped neural networks are consistent. *Advances in Neural Information Processing Systems*, 34, 1805–1817.

Keiding, N., & Louis, T. A. (2016). Perils and potentials of self-selected entry to epidemiological studies and surveys. *Journal of the Royal Statistical Society: Series A*, 179, 319–376.

Kitchin, R. (2015). The opportunities, challenges and risks of big data for official statistics. *Statistical Journal of the IAOS*, 31, 471–481.

Kim, J. K., Park, S., Chen, Y., & 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, 941–963.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint* arXiv:1412.6980.

Leshno, M., Lin, V. Y., Pinkus, A., & Schocken, S. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6, 861–867.

Li, M., Soltanolkotabi, M., & Oymak, S. (2020). Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *Advances in Neural Information Processing Systems* (pp. 4313–4324).

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)* (pp. 807–814).

Rivers, D. (2007). Sampling for web surveys. *American Statistical Association Proceedings*, 4, 1320.

Schmidt-Hieber, J. (2020). Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of Statistics*, 48, 1916–1921.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15, 1929–1958.

Srinivas, S., Subramanya, A., & Venkatesh Babu, R. (2017). Training sparse neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 138–145).

Valliant, R., & Dever, J. A. (2011). Estimating propensity adjustments for volunteer web surveys. *Sociological Methods & Research*, 40, 105–137.

Van der Vaart, A. W. (2000). *Asymptotic statistics*. Cambridge University Press, Cambridge.

Van der Vaart, A. W., & Wellner, J. A. (1996). *Weak convergence and empirical processes: With applications to statistics*. Springer, New York.

Wu, C., & Sitter, R. R. (2001). A model-calibration approach to using complete auxiliary information from survey data. *Journal of the American Statistical Association*, 96, 185–193.

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