Journal of Data Science 23 (2), 312–331
April 2025

DOI: 10.6339/25-JDS1186
Statistical Data Science

Impact of Data Perturbation for Statistical Disclosure Control on the Predictive Performance of Machine Learning Techniques

Thomas Johnson III[1] and Sayed A. Mostafa[1,*]

[1] *Department of Mathematics & Statistics, North Carolina A&T State University, Greensboro, NC, USA*

Abstract

The rapid accumulation and release of data have fueled research across various fields. While numerous methods exist for data collection and storage, data distribution presents challenges, as some datasets are restricted, and certain subsets may compromise privacy if released unaltered. Statistical disclosure control (SDC) aims to maximize data utility while minimizing the disclosure risk, i.e., the risk of individual identification. A key SDC method is data perturbation, with General Additive Data Perturbation (GADP) and Copula General Additive Data Perturbation (CGADP) being two prominent approaches. Both leverage multivariate normal distributions to generate synthetic data while preserving statistical properties of the original dataset. Given the increasing use of machine learning for data modeling, this study compares the performance of various machine learning models on GADP- and CGADP-perturbed data. Using Monte Carlo simulations with three data-generating models and a real dataset, we evaluate the predictive performance and robustness of ten machine learning techniques under data perturbation. Our findings provide insights into the machine learning techniques that perform robustly on GADP- and CGADP-perturbed datasets, extending previous research that primarily focused on simple statistics such as means, variances, and correlations.

Keywords *data confidentiality; data perturbation; machine learning; predictive modeling; statistical disclosure control*

1 Introduction

In the growing and evolving data environment, larger volumes of data are being generated and utilized for various purposes. While this presents many opportunities for analysis, inference, or construction of machine learning workflows, it can pose risks to the participants or members of a particular dataset. To be able to counteract potential threats of vulnerable information being taken from datasets organized by data producers, statistical disclosure control (SDC) methods have been developed (Willenborg and de Waal, 2001; Hoshino, 2020; Elliot and Domingo-Ferrer, 2018). SDC enhances dataset accessibility for researchers by applying techniques that prevent unauthorized actors from linking data subjects to their actual identities. Various SDC methods exist, each with its pros, cons, assumptions, and restrictions that can encourage, discourage, or restrict their usage in various scenarios (Elliot and Domingo-Ferrer, 2018; Hoshino, 2020). For tabular datasets, SDC methods are broadly classified as either perturbative or non-perturbative. Perturbative methods modify the actual values of sensitive variables, while non-perturbative methods employ suppression or recoding to obscure individual identities.

[*] Corresponding author. Email: sabdelmegeed@ncat.edu.

Received August 17, 2024; Accepted April 9, 2025

Two key measures are used to evaluate SDC methods: disclosure risk and data utility. *Disclosure risk* refers to the probability that a malicious actor could extract sensitive or identifying details from an SDC-protected dataset, and it must be minimized according to legal and ethical standards. *Data utility*, on the other hand, reflects the extent to which meaningful and accurate insights can still be derived from the protected dataset and should be maximized to maintain usability. The balance between these measures depends on the specific SDC method employed and the intended use of protected data.

Non-perturbative methods have been shown to reduce accuracy and increase disclosure risk, particularly when masking multivariate confidential numerical data (Willenborg and de Waal, 2001). Even for univariate confidential data, they often fail to guarantee accuracy preservation (e.g., Sarathy et al., 2002). Therefore, we focus on perturbation-based SDC methods. Precisely, the two SDC methods we focus on in this work are general additive data perturbation (GADP) and copula-based general additive data perturbation (CGADP). GADP, first proposed in Muralidhar et al. (1999), encompasses a broader family of methods that utilize additive noise to protect confidential, possibly multivariate, continuous data. A major drawback of GADP is that it relies on the assumption that the data being processed is normally distributed (Muralidhar et al., 1999). Sarathy et al. (2002) extended GADP to a far broader range of distributions by introducing the copula-based GADP (CGADP). Under CGADP, Gaussian copulas are used to transform all variables to a standard multivariate normal distribution, retaining their correlations to one another. GADP is then applied to the transformed variables before transforming them back to their original distributions.

Previous work on additive data perturbation tended to focus on simple statistical measures. For example, Muralidhar et al. (1999) examined the effects of several versions of additive data perturbation that can be considered special cases of GADP along with multiplicative data perturbation on simple statistics including means, standard deviations, and covariance matrices of banking data. Potential biases of the data perturbation methods were identified in the statistics calculated from the perturbed data. Similarly, Sarathy et al. (2002) studied CGADP in comparison to unperturbed data and GADP-perturbed data using the same banking data example. Attention was given to the correlation matrices of the involved variables and the means of variables for specified groupings. Overall, CGADP faired better at sustaining statistical details that required distribution-specific attributes or non-linear phenomena. Chu et al. (2019) used GADP and CGADP to perturb health data and examine their effects when estimating the means, standard deviations, and correlations of health-related variables. They found that GADP was more effective for maintaining the means and standard deviations, while CGADP was more effective at maintaining the correlation matrices. The difference in the extent of preservation of the means versus correlation matrices demonstrates that applying GADP or CGADP could lead to a decrease in data utility. To the best of our knowledge, no research has examined the effects of data perturbation using GADP and CGADP on the performance of machine learning (ML) methods for predictive modeling.

In this study, we aim to fill this gap by investigating the impacts of data perturbation for SDC on the predictive performance of popular ML techniques through an empirical comparative analysis. More specifically, using extensive Monte Carlo simulations and a real data application, we seek to determine which ML techniques are more robust/sensitive to data perturbation via GADP or CGADP, and if particular ML models favor one method over the other.

The remainder of this paper is organized as follows. Section 2 reviews the two general additive data perturbation methods investigated in this study. For completeness and context, Section 3 provides a brief overview of other statistical disclosure control methods, with a particular focus on synthetic data generation techniques. Section 4 describes the machine learning techniques evaluated under data perturbation. Section 5 outlines the simulation experiment setup and presents the simulation results, while Section 6 reports findings from a real data application. Finally, Section 7 concludes the paper with a discussion of key findings.

2 Additive Data Perturbation Methods

In this section, we describe two additive data perturbation methods for SDC. The general setting assumed in the paper is as follows. The dataset to be released consists of two sets of variables. The first set consists of p variables that are insensitive (nonconfidential) auxiliary variables and thus do not require any disclosure control. The second set consists of q confidential variables, i.e., possess vulnerable information, that should be masked before release to the public. Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ denote the data matrices for the nonconfidential variables and the confidential variables, respectively. Let the mean vectors for $\boldsymbol{X}$ and $\boldsymbol{Y}$ be μ_X and μ_Y, and the variance-covariance matrices for $\boldsymbol{X}$ and $\boldsymbol{Y}$ be $\boldsymbol{\Sigma}_{XX}$ and $\boldsymbol{\Sigma}_{YY}$, respectively. Finally, let the matrix $\boldsymbol{Z} = [\boldsymbol{X} \quad \boldsymbol{Y}]$ augment the data in $\boldsymbol{X}$ and $\boldsymbol{Y}$.

2.1 GADP and CGADP Algorithms

Under the above setup, we describe how the GADP and CGADP algorithms can be used to protect vulnerable information $\boldsymbol{Y}$. Since $\boldsymbol{Y}$ is confidential and cannot be released directly, both algorithms generate a set of variables $\tilde{\boldsymbol{Y}}$ which maintains the same statistical characteristics of $\boldsymbol{Y}$ with the privacy versus data utility trade-off managed by a tuning parameter ϑ.

Let $\mu_{\tilde{Y}}$ and $\boldsymbol{\Sigma}_{\tilde{Y}\tilde{Y}}$ denote the mean vector and variance-covariance matrix of $\tilde{\boldsymbol{Y}}$. Let $\boldsymbol{\Sigma}_{XY}$ denote the variance-covariance matrix of $\boldsymbol{X}$ and $\boldsymbol{Y}$, and $\boldsymbol{\Sigma}_{\tilde{Y}Z}$ denote the variance-covariance matrix of $\tilde{\boldsymbol{Y}}$ and $\boldsymbol{Z}$. The GADP algorithm assumes that the nonconfidential ($\boldsymbol{X}$), confidential ($\boldsymbol{Y}$), and perturbed ($\tilde{\boldsymbol{Y}}$) variables have a joint normal distribution (Muralidhar et al., 1999). Further, GADP puts the following conditions on the mean vectors and covariance matrices:

$$\{\mu_{\tilde{Y}} = \mu_Y,\ \boldsymbol{\Sigma}_{\tilde{Y}\tilde{Y}} = \boldsymbol{\Sigma}_{YY},\ \text{and } \boldsymbol{\Sigma}_{\tilde{Y}X} = \boldsymbol{\Sigma}_{YX}\}. \tag{1}$$

These specifications make the joint distributions of $(\boldsymbol{X}, \boldsymbol{Y})$ and $(\boldsymbol{X}, \tilde{\boldsymbol{Y}})$ multivariate normal with identical parameters (means and covariances). Therefore, GADP ensures that the characteristics of the variables in the original database are the same as those variables in the perturbed database. Moreover, by setting

$$\boldsymbol{\Sigma}_{Y\tilde{Y}} = \boldsymbol{\Sigma}_{YX}\boldsymbol{\Sigma}_{XX}^{-1}\boldsymbol{\Sigma}_{XY}, \tag{2}$$

GADP ensures that for any linear combination of variables, the proportion of variability explained is the same before and after perturbation (Muralidhar and Sarathy, 2003).

Under the above specifications, we can show that the conditional distribution of $\tilde{\boldsymbol{Y}}$ given $\boldsymbol{Z} = [\boldsymbol{X} \quad \boldsymbol{Y}]$ is also multivariate normal:

$$\tilde{\boldsymbol{Y}}|\boldsymbol{Z} = \boldsymbol{Z}_i \sim N\big(\mu_Y + \boldsymbol{\Sigma}_{\tilde{Y}Z}\boldsymbol{\Sigma}_{ZZ}^{-1}(\boldsymbol{Z}_i - \mu_Z), \boldsymbol{\Sigma}_{YY} - \boldsymbol{\Sigma}_{\tilde{Y}Z}\boldsymbol{\Sigma}_{ZZ}^{-1}\boldsymbol{\Sigma}_{Z\tilde{Y}}\big). \tag{3}$$

Note that $\boldsymbol{\Sigma}_{\tilde{Y}Z} = [\boldsymbol{\Sigma}_{\tilde{Y}X} \quad \boldsymbol{\Sigma}_{\tilde{Y}Y}]$, where $\boldsymbol{\Sigma}_{\tilde{Y}Y} = \vartheta^2\boldsymbol{\Sigma}_{YY}$. GADP randomly generates perturbed $\tilde{Y}_i$ value for each confidential Y_i value from the above conditional density. Finally, the perturbed database $[\boldsymbol{X} \quad \tilde{\boldsymbol{Y}}]$ is released to allow for statistical analysis.

Algorithm 1 GADP Algorithm.

```
Input submatrices $\boldsymbol{X}$ (nonconfidential attributes) and $\boldsymbol{Y}$ (confidential attributes).
Select $\vartheta$ according to the necessary data utility and privacy safeguards.
1. Calculate $\mu_X$, $\mu_Y$, $\boldsymbol{\Sigma}_{XX}$, and $\boldsymbol{\Sigma}_{YY}$.
2. Define $\boldsymbol{Z} = [\boldsymbol{X} \;\; \boldsymbol{Y}]$ and calculate $\mu_Z = [\mu_X \;\; \mu_Y]$ and $\boldsymbol{\Sigma}_{ZZ}$.
3. Set $\boldsymbol{\Sigma}_{\tilde{Y}X} = \boldsymbol{\Sigma}_{YX}$ and $\boldsymbol{\Sigma}_{\tilde{Y}Y} = \vartheta^2 \boldsymbol{\Sigma}_{YY}$.
4. Concatenate $\boldsymbol{\Sigma}_{\tilde{Y}X}$ and $\boldsymbol{\Sigma}_{\tilde{Y}Y}$ to form $\boldsymbol{\Sigma}_{\tilde{Y}Z} = [\boldsymbol{\Sigma}_{\tilde{Y}X} \; \boldsymbol{\Sigma}_{\tilde{Y}Y}]$.
5. For each confidential $Y_i$ value, randomly generate a value
       $\tilde{Y}_i | Z_i \sim N\big(\mu_Y + \boldsymbol{\Sigma}_{\tilde{Y}Z}\boldsymbol{\Sigma}_{ZZ}^{-1}(Z_i - \mu_Z), \boldsymbol{\Sigma}_{YY} - \boldsymbol{\Sigma}_{\tilde{Y}Z}\boldsymbol{\Sigma}_{ZZ}^{-1}\boldsymbol{\Sigma}_{Z\tilde{Y}}\big).$   (4)
6. Release the perturbed data $[\boldsymbol{X} \;\; \tilde{\boldsymbol{Y}}]$ for analysis.
```

The steps in the GADP algorithm are summarized in Algorithm 1.

Unlike the GDAP method, CGADP does not require all variables to be normally distributed. Therefore, the CGADP method can be effective for perturbing non-normal confidential variables while maintaining the monotonic correlation structure of the variables in the database. This is achieved through the use of copula functions, such as the Gaussian copula, as described below.

Let the matrices $\boldsymbol{X}$, $\boldsymbol{Y}$, $\tilde{\boldsymbol{Y}}$, and $\boldsymbol{Z}$ be as defined in the previous section. Further, let the marginal distribution of non-confidential variable X_j be denoted by F_j; $j = 1, 2, \ldots, p$. Similarly, the marginal distribution of confidential variables Y_k is denoted by G_k; $k = 1, 2, \ldots, q$. With this notation, we formulate CGADP in Algorithm 2.

The CGADP algorithm is motivated by the multivariate Gaussian copula given by

$$\boldsymbol{c}_{\boldsymbol{\rho}}(u) = \boldsymbol{\Phi}_{\boldsymbol{\rho}}^{k}\big(\boldsymbol{X}^*, \boldsymbol{Y}^*, \tilde{\boldsymbol{Y}}^*\big), \tag{5}$$

where $\boldsymbol{\Phi}_{\boldsymbol{\rho}}^{k}$ is the distribution function of a k-variate normal distribution with correlation matrix $\boldsymbol{\rho}$. The perturbed database $[\boldsymbol{X}\ \tilde{\boldsymbol{Y}}]$ maintains the correlation structure of the original database $[\boldsymbol{X}\ \boldsymbol{Y}]$ as well as the characteristics of the marginal distributions. However, CGADP has two major limitations. The first is that it can only be successful for variables that have monotonic relationships with one another (Chu et al., 2019; Sarathy et al., 2002). An additional concern with CGADP is the marginal distribution that each of the variables in $\boldsymbol{X}$ is identified to follow. If a different distribution than the actual family for any one variable is used, then the resulting application of CGADP can cause the values in $\tilde{\boldsymbol{Y}}$ to be generated from a different distribution than that of $\boldsymbol{Y}$ (Sarathy et al., 2002). This will not hinder the resulting safeguards provided by CGADP, but the same cannot be said about the utility of the data in $\tilde{\boldsymbol{Y}}$ (Sarathy et al., 2002).

The extended skew-t copula method was proposed by Chu et al. (2022) to better capture skewness and kurtosis as well as accommodate longer-tailed distributions. Another version of GADP, called Enhanced GADP (EGADP), was developed to better handle data perturbation on small datasets where the utility of GADP data and safeguards against disclosure risk could suffer (Muralidhar and Sarathy, 2005).

2.2 Theoretical Basis for Additive Data Perturbation

In this section, we summarize the theory underlying additive data perturbation to provide a foundation for parsing our empirical findings in later sections. We adopt the framework of Muralidhar and Sarathy (2003). Let $g(\cdot)$ denote the probability density function of the confidential

Algorithm 2 CGADP Algorithm.

```
Input submatrices $X$ (nonconfidential attributes) and $Y$ (confidential attributes).
Select $\vartheta$ according to the necessary data utility and privacy safeguards.
1. Identify the marginal distribution of each variable in $\boldsymbol{X}$ $(F_j;\ j = 1, 2, \ldots, p)$ and $\boldsymbol{Y}$ $(G_k;\ k = 1, 2, \ldots, q)$. Note that some or all of these distributions may be non-normal.
2. Transform the original variables in the database to have standard normal distribution:
       $X_j^* = \Phi^{-1}\big(F_j(X_j)\big);\ \ j = 1, 2, \ldots, p,$
       $Y_k^* = \Phi^{-1}\big(G_k(Y_k)\big);\ \ k = 1, 2, \ldots, q,$
   where $\Phi^{-1}(\cdot)$ is the inverse cumulative distribution function of a univariate standard normal random variable.
3. Compute the Spearman's rank order correlations among the variables in $\boldsymbol{Z} = [\boldsymbol{X}\ \ \boldsymbol{Y}]$ and store them in matrix $\boldsymbol{R}$. Note that $\boldsymbol{R}$ retains the same values whether the variables are transformed or not.
4. Obtain the Pearson's product moment correlations among the transformed variables in $\boldsymbol{Z}^* = [\boldsymbol{X}^*\ \ \boldsymbol{Y}^*]$ via the relationship
       $\rho_{j,k} = 2 \cdot \sin(\pi r_{j,k})/6,$
   where $r_{j,k}$ is element in the $j$-th row and $k$-th column of $\boldsymbol{R}$. Store these correlations in matrix $\boldsymbol{\rho}$. Since the transformed variables have a standard normal distribution, the correlation matrix $\boldsymbol{\rho}$ serves as the variance-covariance matrix, $\boldsymbol{\Sigma}_{Z^*Z^*}$.
5. Apply GADP (see Algorithm 1) to $\boldsymbol{Z}^* = [\boldsymbol{X}^*\ \ \boldsymbol{Y}^*]$ to generate $\tilde{Y}^*$, the perturbed version of $Y^*$.
6. Compute the final perturbed variables $\tilde{\boldsymbol{Y}}$ from $\tilde{\boldsymbol{Y}}^*$ via the back-transformation
       $\tilde{Y}_k = G_k^{-1}\big(\Phi\big(\tilde{Y}_k^*\big)\big);\ \ k = 1, 2, \ldots, q.$
7. Release the perturbed data $[\boldsymbol{X}\ \ \tilde{\boldsymbol{Y}}]$ for analysis.
```

data Y. The generation of perturbed data is carried out using the conditional distribution $g(Y|\boldsymbol{X})$ to obtain $\tilde{Y}$ that is independent of Y conditional on $\boldsymbol{X}$. Precisely, each data point i will have a corresponding perturbed data value $\tilde{y}_i$ obtained by

$$\tilde{y}_i \sim g(Y|\boldsymbol{X} = \boldsymbol{x}_i) \tag{6}$$

such that

$$g(Y, \tilde{Y}|\boldsymbol{X}) = g(Y|\boldsymbol{X})g(\tilde{Y}|\boldsymbol{X}). \tag{7}$$

To uphold data utility and disclosure risk at their expected levels, some presumptions must be maintained. For data utility, the first condition is that the marginal distribution of the original data is maintained within the perturbed data, i.e., $g(\tilde{Y}) = g(Y)$. The second condition is related to the joint distribution of the confidential and non-confidential variables where it is expected that $g(\tilde{Y}, \boldsymbol{X}) = g(Y, \boldsymbol{X})$. Similarly, for maintaining the expected levels of disclosure risk, two assumptions must hold. The first point is that all relevant information about Y and $\boldsymbol{X}$ is contained in the conditional density $g(Y|\boldsymbol{X})$ and $\boldsymbol{X}$. When perturbed data $\tilde{Y}$ is disseminated, the conditional density can be expressed as $g(Y|\boldsymbol{X}, \tilde{Y})$, assuming $\tilde{Y}$ provides additional details about Y. Ideally, this is not the case, meaning the perturbed data should ensure

$g(Y|\boldsymbol{X}, \tilde{Y}) = g(Y|\boldsymbol{X})$, preventing $\tilde{Y}$ from revealing extra information about Y. This assumes that $\tilde{Y}$ and $\boldsymbol{X}$ retain sufficient information for inference. A key question is whether $\boldsymbol{X}$ still contributes meaningful information post-perturbation. Prior research examines how covariates influence estimation when incorporating external data sources (Estes et al., 2018). When transportability violations occur—meaning $Y|\boldsymbol{X}$ differs between datasets—estimator performance can degrade. This is particularly relevant to data perturbation, where any shift in $g(Y|\boldsymbol{X})$ between the perturbed and unperturbed datasets can impact the reliability of statistical methods. If perturbation alters this relationship too severely, analyses conducted on the perturbed dataset may deviate from those on the original, reducing data utility. Similarly, research on synthetic data generation highlights the importance of maintaining $g(Y|\boldsymbol{X})$ for accurate predictions (Gu et al., 2019). Artificial data generation methods rely on well-supported models, relevant external data, and imputation techniques to supplement a research dataset while minimizing distortions in the conditional distribution. In data perturbation, a similar challenge arises: ensuring that $g(Y|\boldsymbol{X})$ and $g(Y|\boldsymbol{X}, \tilde{Y})$ remain aligned to preserve analytical validity. Methods such as GADP and CGADP aim to maintain key statistical properties, including covariances and means, ensuring compatibility with machine learning models that depend on these features. Models benefiting from these preserved structures perform better under GADP and CGADP, whereas those requiring more intricate distributional properties may face limitations.

3 Related Methods

In addition to additive data perturbation, other statistical disclosure control methods aim to balance data privacy with data utility. One such approach is data swapping, which protects confidentiality by reordering dataset values rather than altering them. Data swapping can be implemented in various ways to switch the values of confidential variables while maintaining dataset structure. A well-known technique was introduced by Moore (1996), where the extent of value swaps is controlled by a parameter that directly affects both data utility and disclosure risk. Another approach, proposed by Carlson and Salabasis (2002), partitions the dataset into smaller subsets, swaps values within these subsets, and then recombines them to form a final perturbed dataset. While effective at reducing disclosure risk, these methods do not leverage the conditional distribution $g(Y|\boldsymbol{X})$, making it difficult to assess their impact on data utility within the data perturbation framework. Data shuffling, on the other hand, incorporates the distribution $g(Y|\boldsymbol{X})$ and can be studied within the framework of Muralidhar and Sarathy (2003). Unlike data swapping, which directly exchanges values, data shuffling reorders values based on the ranks of randomly generated variates rather than replacing them outright. This makes data shuffling more compatible with additive data perturbation techniques. Despite these advantages, data shuffling is prone to increased variance when applied to small datasets and struggles to capture non-monotonic associations between variables, reducing its effectiveness in certain scenarios (Muralidhar and Sarathy, 2006).

Another approach to statistical disclosure control is differential privacy, which injects noise into the data, often through the Laplace mechanism. Differential privacy can be applied at two levels: global sensitivity, which is primarily used in theoretical analyses, and local sensitivity, which is more practical for real-world implementations. The amount of noise added through the Laplace mechanism must be carefully calibrated to balance privacy protection and data utility (Hu et al., 2022a). Wang et al. (2016) proposed using the randomized response technique in the context of differential privacy, which scrambles Y independently of $\boldsymbol{X}$ and uses distributional properties of the scrambled values to facilitate accurate inference. Their empirical results suggest that randomized response outperforms the Laplace mechanism for categorical variables and network analysis tasks. Lately, Bayesian methods have been incorporated into the differential privacy framework to improve synthetic data generation (Hu et al., 2022b). For a recent review of differential privacy techniques and their implementation in the context of machine learning, we refer the reader to Blanco-Justicia et al. (2022). Similarly, a recent review of synthetic data generation methods can be found in Kokosi et al. (2022).

More recently, generative AI has been explored as a tool for supplementing datasets with synthetic data while maintaining privacy protections. The Syn framework, for example, integrates differential privacy to generate artificial datasets that retain the distributional properties of the original data (Shen et al., 2023). By reducing the degree of distortion introduced by earlier privacy-preserving methods, generative AI has the potential to improve data utility while still protecting against disclosure risks. However, ensuring that artificial data maintains statistical properties that align with the original dataset remains an ongoing challenge.

Each of these methods represents a different approach to balancing privacy protection with data utility. While additive data perturbation techniques focus on modifying individual values, data swapping and data shuffling attempt to preserve relationships while reordering values. Differential privacy introduces controlled noise, and Bayesian methods refine this approach by leveraging probabilistic modeling. Recent advances, such as generative AI, introduce new possibilities, but raise questions about the fidelity of synthetic data. These diverse methods highlight the complexity of statistical disclosure control and the need for a unified evaluation framework (Elliot and Domingo-Ferrer, 2018).

4 Predictive Machine Learning Techniques

In this investigation, we considered six popular machine learning techniques for predictive modeling as well as stacked ensembles of these techniques. In the following, we give a high-level description of each of these techniques and their implementation.

Linear Regression (LR): is an interpretable machine learning technique widely used for predictive modeling (Hastie et al., 2009). It assumes a linear relationship between the predictors and the response variable. While it serves as a useful baseline for evaluating predictive performance, its sensitivity to multicollinearity, outliers, and high-dimensional settings limits its applicability. In the context of data perturbation, LR provides insight into how distortions affect models reliant on explicit feature relationships. LR can be implemented using built-in R functions such as `lm` or `glm` (R Core Team, 2022).

Least Absolute Shrinkage and Selection Operator (LASSO) Regression: extends linear regression by applying L_1 regularization, which shrinks some coefficients to zero, effectively performing variable selection (Hastie et al., 2009). This makes LASSO well-suited for high-dimensional data, particularly when many predictors are correlated. Studying LASSO under data perturbation helps assess how regularization on perturbed data impacts predictive performance. LASSO is implemented in the `glmnet` package and its respective `glmnet` function available in R (Tay et al., 2023).

Support Vector Machines (SVMs): is a nonparametric method that constructs hyperplanes to minimize a chosen loss function, with kernel functions enabling flexibility in capturing nonlinear relationships (Li et al., 2006). Popular choices of the kernel function in SVMs include the linear kernel, the polynomial kernel, and the radial kernel (Lundell, 2023). Hereafter, we denote an SVM with a linear kernel as Linear_SVM, an SVM with a polynomial kernel by Poly_SVM, and an SVM with a radial kernel by Radial_SVM. Unlike LR, SVMs do not rely on explicit feature assumptions, making them potentially resilient to data perturbation effects. However, the performance of SVMs depends heavily on hyperparameter tuning, which might be further complicated by data perturbation. SVMs can be deployed using the `svm` and `tune.svm` functions from the `e1071` package in R (Meyer et al., 2023).

Neural Networks (NNETs): are highly flexible nonparametric predictive models that learn complex patterns through multiple layers of processing (Hastie et al., 2009), namely, an input layer, one or more hidden layers, and an output layer. NNETs are considered nonlinear models because they transform input data into increasingly abstract representations through a series of weighted connections and activation functions. Unlike linear models, NNETs can capture intricate dependencies between variables without requiring explicit feature engineering. For an in-depth introduction to neural networks, we refer the reader to Hastie et al. (2009, Ch. 11). NNETs are particularly relevant for studying the effects of data perturbation due to their adaptability and sensitivity to data alterations. Unlike simpler models such as LR or LASSO, NNETs can learn intricate feature interactions, making them more resistant to small perturbations. However, they are also highly sensitive to noise, especially in deep architectures with many parameters. If perturbation significantly alters the structure of the data, NNETs may overfit to the modified patterns, reducing their ability to generalize. Evaluating NNETs under different levels of perturbation provides insights into their robustness and the extent to which data distortions affect model stability. A single hidden layer NNET can be implemented in R utilizing the `nnet` function in the `nnet` package in R (Venables and Ripley, 2002).

Random Forests (RFs): leverage bagging, aka bootstrap aggregation, to aggregate multiple decision trees, improving stability and reducing variance (Breiman, 2001; Duroux and Scornet, 2018). They perform well with structured and unstructured data, capturing interactions and nonlinearities without requiring strict feature assumptions. Studying the impact of perturbation on RFs helps assess how ensemble methods mitigate distortions while maintaining predictive performance. RFs can be implemented using the `ranger` function in the `ranger` package in R (Wright and Ziegler, 2017).

Extreme Gradient Boosting Trees (XGBTrees): apply boosting to decision trees, sequentially improving predictions by weighting errors more heavily in successive iterations (Chen and Guestrin, 2016). This results in high predictive accuracy, but can also lead to overfitting, particularly when data perturbation alters the data distribution. Studying XGBTrees under data perturbation can highlight how boosting algorithms handle noise and whether they maintain generalization capabilities. XGBTrees are accessible from the `xgboost` package in R (Chen and Guestrin, 2016). For more details on XGBTrees, we refer the reader to Hastie et al. (2009) and Chen and Guestrin (2016).

Ensemble Techniques: In addition to the above modeling techniques, we also considered two types of stacked ensemble regressions. The first stacked ensemble is defined as the mean of the above eight predictive models' predictions with equal weights. It is identified as SE_equal. The second stacked ensemble is defined as a weighted mean of the above eight predictive models' predictions, where the weights are inversely proportional to the training mean squared error of each model. This implementation of the stacked ensemble is identified as SE_prop. The eight base models of the stacked ensembles are LR, LASSO, RF, XGBTrees, NNET, Linear_SVM, Radial_SVM, and Poly_SVM.

5 Simulation Experiments

In this section, we use extensive simulations to evaluate the impact of data perturbation, using GADP or CGADP, on the predictive performance of the various machine learning techniques described in the previous section.

5.1 Simulation Settings

As assumed throughout the paper, our simulations consider a database where the sensitive response variables $\boldsymbol{Y}$ are confidential attributes that must be subjected to some sort of perturbation before being released to the public while the predictors $\boldsymbol{X} = (X_1, X_2, \dots, X_p)$ are nonconfidential attributes that do not require any protection. We considered three data-generating models to evaluate the effects of data perturbation under different structural relationships between predictors and response variables. In all models, the predictor matrix $\boldsymbol{X}$ is generated from a multivariate normal distribution with mean zero, unit variance, and correlation ρ. In Models I and III, the normal predictors are transformed into a correlated uniform distribution using the CDF of the standard normal distribution. For each model, two response variables (Y_1 and Y_2) are generated with different noise levels: $\varepsilon_1 \sim N(0, 0.2)$ and $\varepsilon_2 \sim N(0, 0.4)$. Each model is examined under 10, 50, and 100 predictors, where variables without assigned coefficients act as noise predictors.

Model I (Linear Model): Adapted from McConville (2011), this model follows a linear structure:

$$Y_j = 0.841 + x_2 + 1.5x_4 + x_8 + \varepsilon_j,$$

where only three predictors contribute to the response variable, while the remaining predictors serve as noise. This model provides a baseline for assessing how well linear and non-linear machine learning methods perform under data perturbation.

Model II (Non-Linear Model with Threshold Effects): This model incorporates piecewise thresholding effects in the response function:

$$Y_j = 1 + 1.5x_1 - 0.5x_2 + 4 \cdot (x_4 < 0.5) - 2 \cdot (x_4 \geqslant 0.5) + 1.5 \cdot (x_5 < 0.25) + x_6 + \varepsilon_j,$$

where the impact of x_4 and x_5 depends on whether they fall above or below certain thresholds. This structure tests how data perturbation affects models that rely on non-linear relationships and categorical-like decision boundaries.

Model III (Smooth Non-Linear Model): Adapted from Duroux and Scornet (2018), this model incorporates smooth non-linear transformations of predictors:

$$Y_j = -\sin(2x_1) + x_2^2 + x_3 - \exp(-x_4) + \varepsilon_j,$$

where predictor effects are governed by sinusoidal, quadratic, and exponential functions. This model evaluates the performance of flexible machine learning techniques under data perturbation when the response surface is highly non-linear.

These models allow us to assess the impact of data perturbation on predictive performance under linear versus non-linear relationships, different predictor-to-response mappings, and varying levels of noise.

Under each of the above model configurations, we generated a training dataset of size $n_{\text{train}} = 500, 1000$ or 2000. The training set was then subjected to data perturbation using GADP and CGADP separately leading to two perturbed versions of the training set. The privacy parameter ϑ was varied ($\vartheta = 0.2, 0.4, 0.6, 0.8$) to enable examining the impact of varying the level of privacy versus data utility—noting that higher values of ϑ imply less privacy and higher data utility and vice versa. It should be noted that while data perturbation using GADP or CGADP was applied to both response variables (Y_j; $j = 1, 2$ resulting from the two levels of noise $\sigma = 0.2, 04$) simultaneously, each response variable was modeled and predicted separately. Each of the predictive modeling techniques described in Section 4 was fit to each of the three versions of the training set: i) the unperturbed training set, ii) the training set with GADP perturbation, and iii) the training set with CGADP perturbation. This process was repeated $m = 1000$ times. The trained models were then used to predict the response variable values in a single fixed testing set of size $n_{\text{test}} = 500$ and calculate the mean squared prediction error as follows:

$$\text{MSE}_k = \frac{1}{n_{\text{test}}} \sum_{i \in \text{test set}} \left(y_i - \hat{y}_i^{(k)}\right)^2,$$

where $k = 1, 2, \ldots, m$ denotes the simulation replication, y_i is the value of the response variable for the i-th observation in the test set without perturbation and $\hat{y}_i^{(k)}$ is the predicted value in the k-th simulation replication from a predictive model trained on i) the non-perturbed data, ii) the perturbed data using GADP, or iii) the perturbed data using CGADP. Two performance metrics were then obtained to assess the predictive performance of each predictive model. The first metric is the average test MSE (AMSE):

$$\text{AMSE} = \frac{1}{m} \sum_{k=1}^{m} \text{MSE}_k.$$

The second metric is the ratio between the model's AMSE when no data perturbation is applied and its AMSE under data perturbation calculated as

$$\text{AMSER} = \text{AMSE}_{\text{perturbed}} / \text{AMSE}_{\text{unperturbed}}.$$

Parameter tuning was performed for all models except linear regression. For LASSO, the penalty parameter was tuned across 100 λ values with the optimal λ selected via ten-fold cross-validation. The NNET models were trained using `maxit` = 200 with the two key hyperparameters `size` and `decay` tuned using five-fold cross-validation with the possible values being `size` = {3, 5, 10, 20} and `decay` = {0.001, 0.01, 0.1}. The RF models were trained using `num.trees` = 500 with the other tuning parameters `mtry` $= \{\lceil (1/3) \cdot p \rceil, \lceil (1/4) \cdot p \rceil\}$ and `min.node.size` $= \{5, 10\}$ chosen via five-fold cross-validation. The XGBTrees models were optimized using five-fold cross-validation with hyperparameter grids set as `nrounds` $= \{10, 20, 40\}$, `max.depth` $= \{2, 4\}$, `eta` $= \{0.2, 0.4, 0.6\}$, `gamma` $= \{0, 0.5, 1\}$, `colsample.bytree` $= \{0.2, 0.5, 1\}$, `min.child.weight` $= \{0.5, 1\}$, and `subsample` $= \{0.4, 0.5, 0.6\}$. The Radial_SVM models underwent ten-fold cross-validation to select the best `gamma` $= \{0.1, 1\}$ and `cost` $= \{0.1, 1\}$. The Linear_SVM models had their best `cost` chosen via ten-fold cross-validation from among $\{0.1, 1\}$. Similarly, the Poly_SVM models had their hyperparameters tuned using ten-fold cross-validation over the grids `degree` $= \{1, 2\}$, `cost` $= \{0.1, 1\}$, and `gamma` $= \{0.001, 0.1\}$. In all cases, the hyperparameter grids were determined to balance computational efficiency and model accuracy.

All computations were performed using R version 4.1.0 (R Core Team, 2022). The computations were run on a compute cluster with a compute node possessing AMD EPYC CPU with 256 cores and 530 GB memory.

5.2 Simulation Results

In this section, we present key results from the simulation experiments described above. Figures 1–8 display the average mean square error ratios (AMSER) for ten machine learning (ML) techniques under data perturbation relative to no perturbation. For brevity, we report results for $p = 10$ and $p = 100$, while results for $p = 50$, which follow similar patterns, are provided in the Supplementary Material. To illustrate how the noise level interacts with the data perturbation impacts on the predictive performance, we present results for Model I with error variances $\sigma_\varepsilon = 0.2$ and $\sigma_\varepsilon = 0.4$. Since Models II and III exhibit similar trends, we show AMSER graphs only for $\sigma_\varepsilon = 0.2$ and defer $\sigma_\varepsilon = 0.4$ to the Supplementary Material which also includes tabulations of the numerical values of the AMSE and AMSER for the best-achieving ML technique(s)

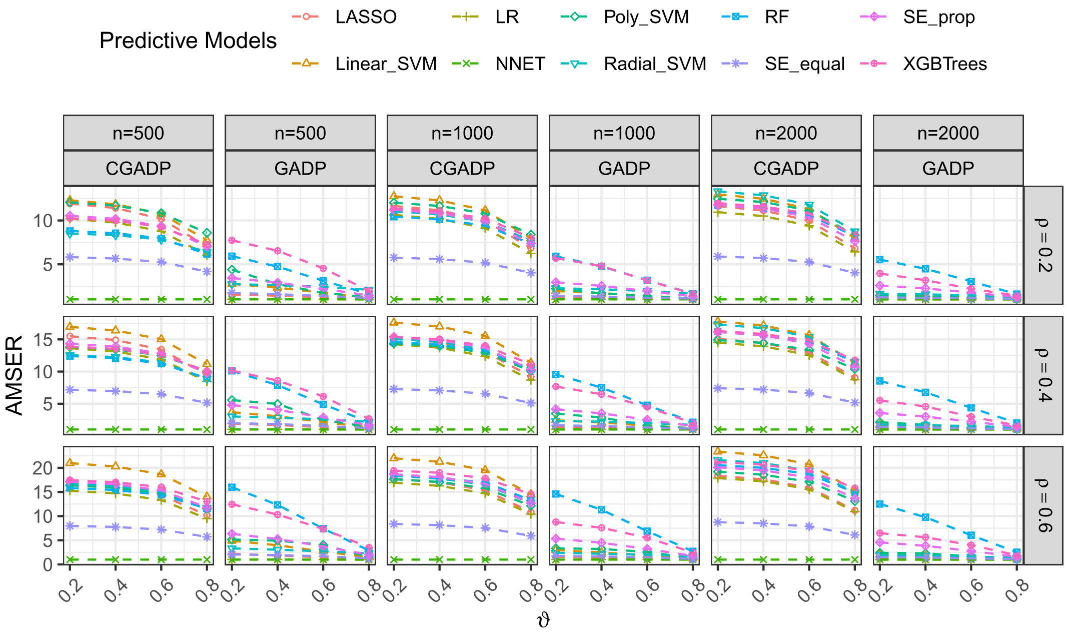

Figure 1: Average mean square error ratio (AMSER) for ten ML techniques under **Model I** with $p = 10$ and $\sigma_\varepsilon = 0.2$.

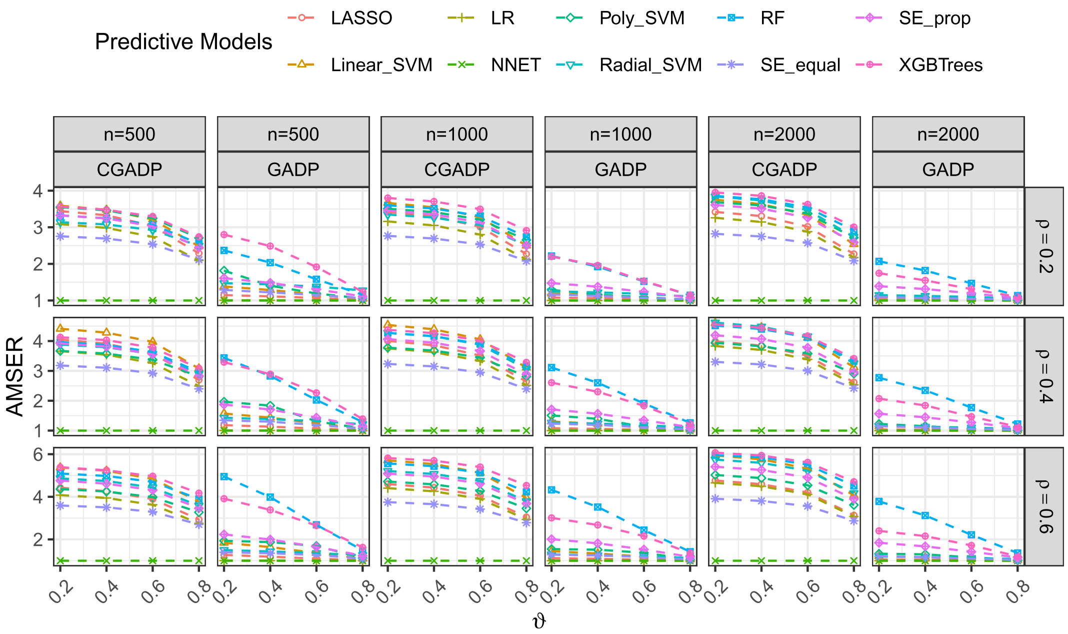

Figure 2: AMSER for ten ML techniques under **Model I** with $p = 10$ and $\sigma_\varepsilon = 0.4$.

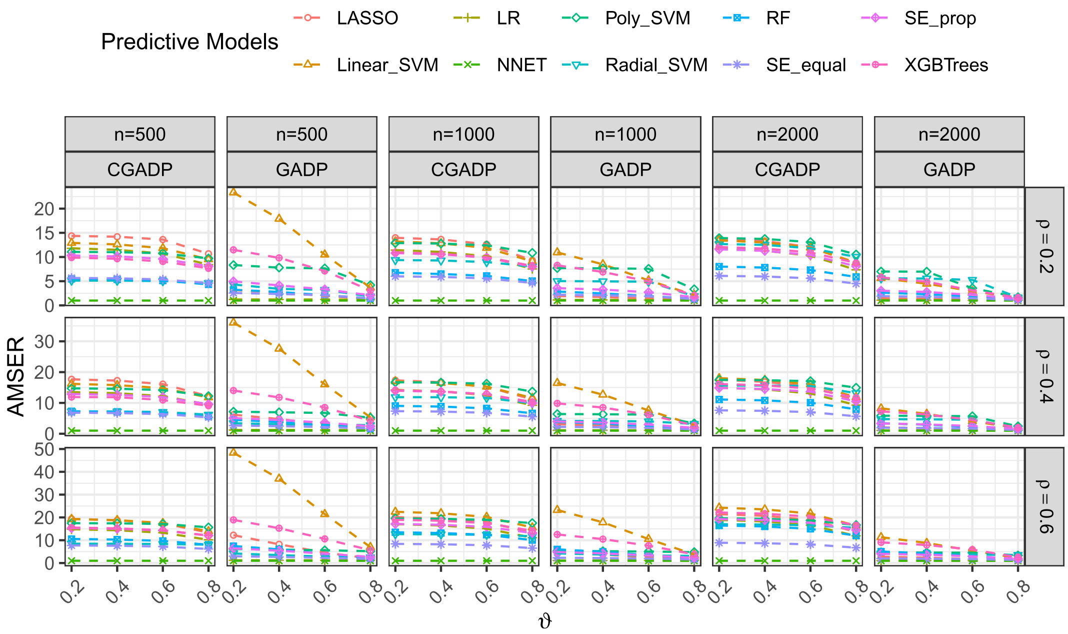

Figure 3: AMSER for ten ML techniques under **Model I** with $p = 100$ and $\sigma_\varepsilon = 0.2$.

under each scenario.

The results presented in Figures 1 to 8 illustrate the impact of General Additive Data Perturbation (GADP) and Copula-based GADP (CGADP) on the predictive performance of the ten ML techniques under variations in the number of predictors (p), correlation among predictors (ρ), sample size (n), and privacy protection parameter (ϑ). Overall, the results demonstrate that

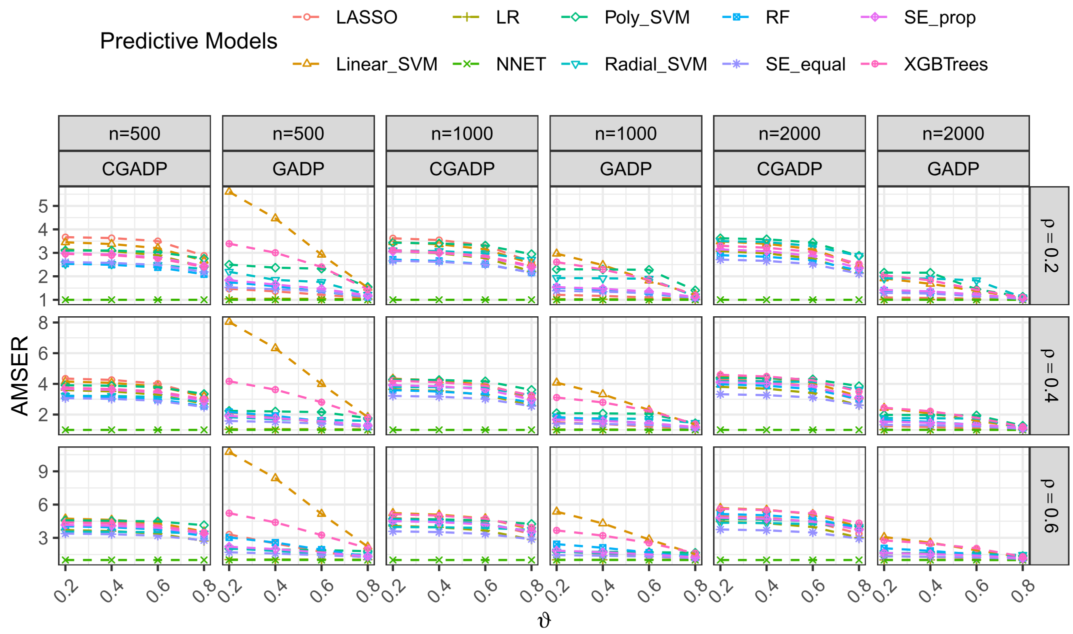

Figure 4: AMSER for ten ML techniques under **Model I** with $p = 100$ and $\sigma_\varepsilon = 0.4$.

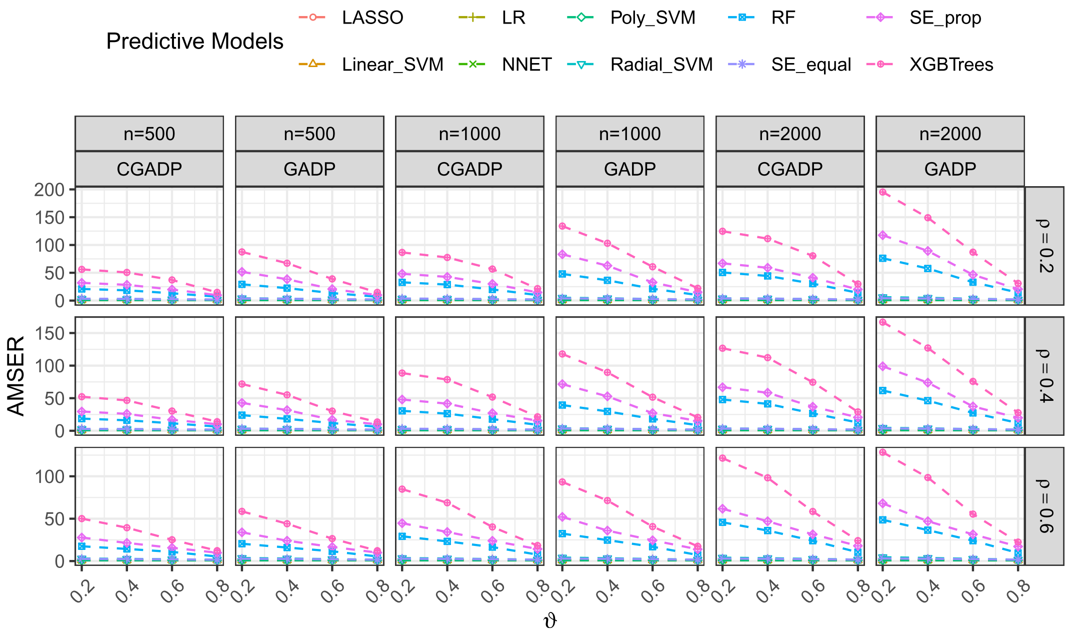

Figure 5: AMSER for ten ML techniques under **Model II** with $p = 10$ and $\sigma_\varepsilon = 0.2$.

perturbation affects models differently, with parametric models like Linear Regression (LR) and LASSO showing more resilience under GADP, while flexible models such as Random Forests (RF) and XGBTrees overfit.

For Model I, which follows a linear structure, Figures 1 and 2 reveal that LR, LASSO, and Linear_SVM maintain the lowest AMSER, confirming that linear models perform well when

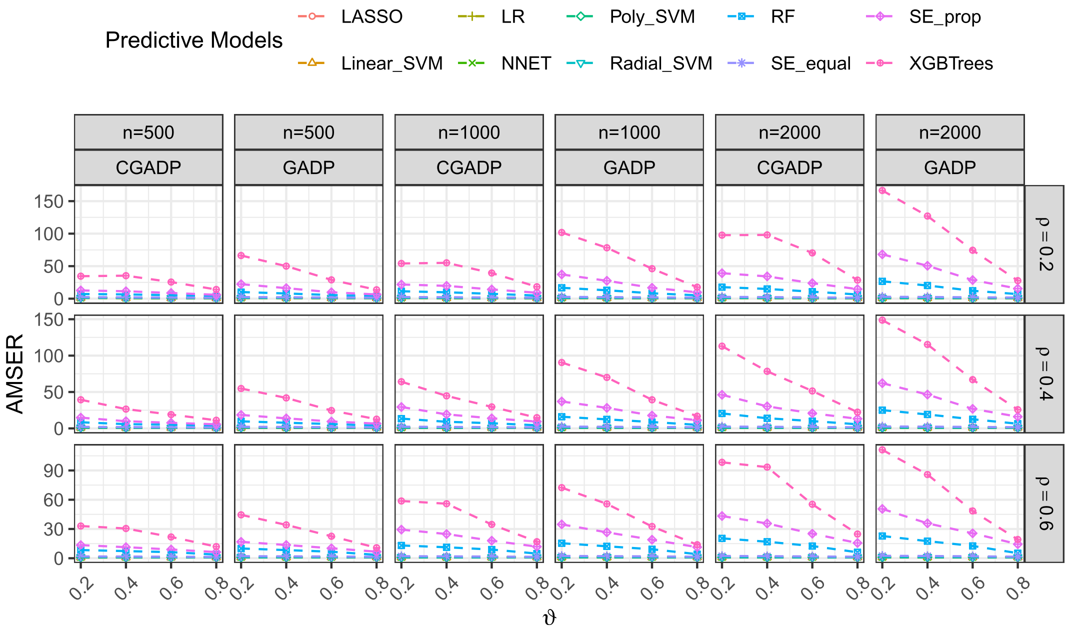

Figure 6: AMSER for ten ML techniques under **Model II** with $p = 100$ and $\sigma_\varepsilon = 0.2$.

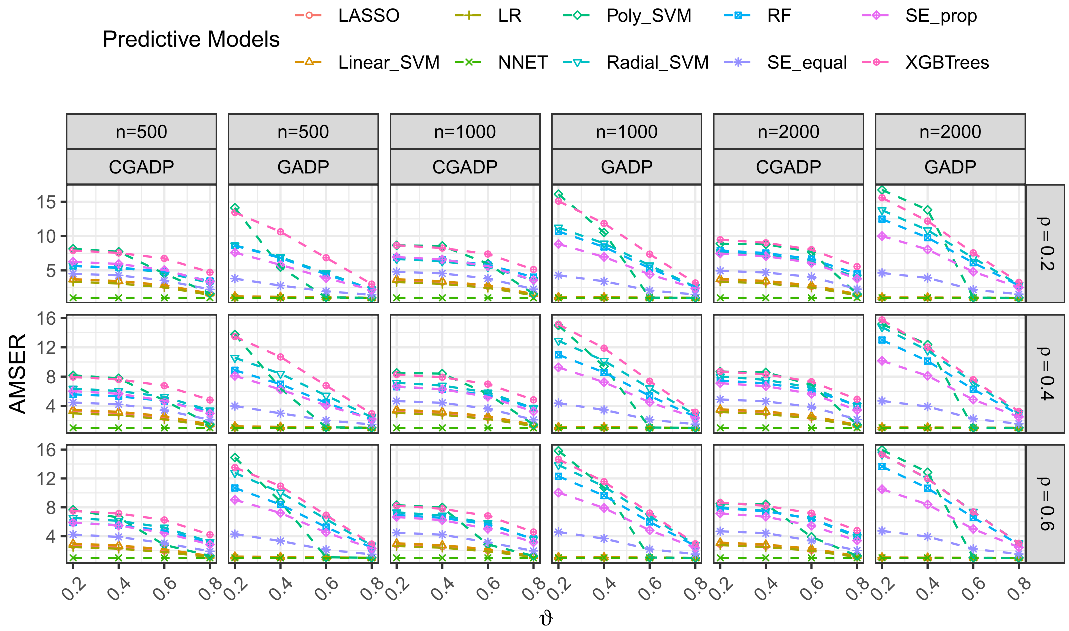

Figure 7: AMSER for ten ML techniques under **Model III** with $p = 10$ and $\sigma_\varepsilon = 0.2$.

data perturbation is primarily based on preserving means and covariances. In contrast, RF and XGBTrees exhibit high AMSER, indicating significant degradation in performance due to their increased sensitivity to noise introduced by perturbation. When ϑ increases, meaning less privacy protection is applied, AMSER decreases across all models, but the reduction is more pronounced for parametric models that rely on the stability of means and covariances. SE_equal, which as-

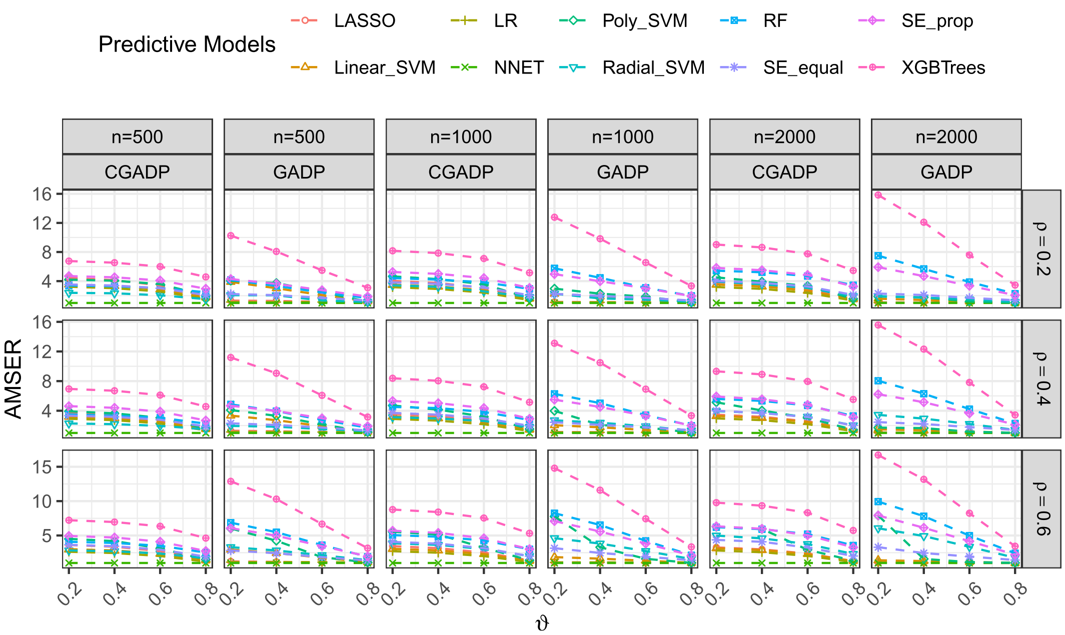

Figure 8: AMSER for ten ML techniques under **Model III** with $p = 100$ and $\sigma_\varepsilon = 0.2$.

signs equal weights to predictions from all models, outperforms SE_prop, which weights models proportionally based on performance. The latter suffers from an overreliance on flexible models such as RF and XGBTrees, which tend to overfit the perturbed data, leading to suboptimal performance.

As the number of predictors increases from 10 to 100, Figures 3 and 4 show that LR and LASSO remain robust, while nonparametric models experience a further decline in predictive accuracy. Radial_SVM and Poly_SVM, which were more stable in the lower-dimensional setting, display increased AMSER in the high-dimensional case, likely due to difficulties in tuning hyperparameters on perturbed datasets. RF and XGBTrees continue to perform poorly, suggesting that the introduction of more predictors exacerbates their vulnerability to perturbation noise. Another key observation is that increasing correlation (ρ) leads to an increase in AMSER for most models, likely because correlated features amplify the distortion caused by data perturbation, making it harder for models to generalize effectively.

Model II introduces threshold-based non-linearity, which presents additional challenges for predictive models, as illustrated in Figures 5 and 6. The added complexity reduces the performance of all models compared to Model I, but the effect is particularly evident for RF and XGBTrees, which experience the most significant decline. These tree-based models, which typically excel in capturing thresholded relationships under no perturbation, exhibit substantially higher AMSE under data perturbation. This suggests that perturbation disrupts the thresholded structure, inflating AMSE of these tree-based models and consequently inflating their AMSER. NNET consistently exhibits an AMSER close to 1, suggesting that neural networks fail to adjust effectively to perturbed data across different scenarios. SE_equal continues to outperform SE_prop, reinforcing the observation that equal weighting prevents excessive reliance on overfitting-prone models.

For Model III, which introduces smooth non-linear transformations, Figures 7 and 8 highlight a shift in relative performance among the models. Radial_SVM and RF perform better in this setting compared to their results in Models I and II, benefiting from their ability to capture non-linearity. However, AMSER remains high for XGBTrees, suggesting that its boosting mechanism amplifies errors in the presence of perturbation. LR and LASSO, which performed strongly in Model I, now show noticeable performance deterioration, reflecting the difficulty of linear methods in modeling complex functional relationships. NNET continues to have AMSER close to 1, meaning its prediction errors are consistently large for both perturbed and unperturbed datasets, further reinforcing its poor adaptability to data perturbation. The results also indicate that CGADP imposes greater challenges than GADP, particularly for parametric models like LR, LASSO, and Linear_SVM, as it introduces noise that disrupts the marginal distributions of features rather than merely perturbing statistical moments.

Across all models and scenarios, increasing sample size (n) generally improves performance by reducing AMSER, but the extent of improvement varies by model. While larger sample sizes help LR, LASSO, and SVMs maintain stability, they do not fully mitigate the overfitting issues observed in RF and XGBTrees. Higher values of ϑ, which correspond to weaker privacy protection, lead to a universal reduction in AMSER, but the impact is more significant for models that rely on underlying statistical properties, such as LR and LASSO. The key takeaway from these results is that GADP favors linear models, which align with its preservation of means and covariances, while CGADP presents a greater challenge for parametric models by disrupting the underlying distributions. Among flexible models, Radial_SVM and RF demonstrate some resilience in high-dimensional settings, but XGBTrees and NNET consistently underperform across all scenarios, making them unsuitable choices for prediction under data perturbation.

6 Real Data Application

In this section, we examine the impact of data perturbation on the predictive performance of various modeling techniques using data from the 2015 Consumer Expenditure (CE) survey conducted by the U.S. Bureau of Labor Statistics. The publicly available CE data, also included in the `rpms` package in R (Toth, 2021), contains information on consumer unit characteristics, assets, and expenditures across 47 variables for $N = 68{,}415$ consumer units. Several key variables are relevant to our analysis. FINCBTAX represents pre-tax earnings accumulated by respondents over the 12 months preceding the survey. SALARYX measures total earnings before deductions within the same period. TOTXEST estimates the amount of tax paid, while TOTEXPCQ captures total expenditures for the economic quarter in which the survey was conducted. AGE indicates the primary earner's age in years, and SEX records the respondent's gender (male or female).

Initial data processing involved removing outliers using the $(\mathrm{Q}_1 - 3\cdot\mathrm{IQR}, \mathrm{Q}_3 + 3\cdot\mathrm{IQR})$-rule for FINCBTAX, SALARYX, TOTXEST, and TOTEXPCQ. The TOTXEST, TOTEXPCQ, and FINCBTAX variables were right-shifted to align with the following pre-specified distributions for CGADP:

- AGE: mixture normal distribution.
- SEX: binomial distribution.
- TOTXEST: approximately log-normal distribution.
- TOTEXPCQ: approximately log-normal distribution.
- FINCBTAX: approximately log-normal distribution.
- SALARYX: approximately normal distribution.

After preprocessing, a stratified simple random sample of 1,000 males and 1,000 females was drawn, resulting in a final dataset of 2,000 observations. For each simulation run, a simple random sample was split 50:50 into a training set (1,000 observations) and a testing set (1,000 observations). The predictors included AGE, SEX, TOTXEST, and TOTEXPCQ, while FINCBTAX and SALARYX served as response variables subject to perturbation. Two sets of predictive models were trained: one using perturbed training data and one using non-perturbed training data. Regardless of how the models were trained, their performance was evaluated on the actual values of FINCBTAX and SALARYX in the testing set. This setup mirrors the simulation experiments described in Section 5. The model parameter tuning was performed exactly as described in Section 5. The AMSE and AMSER were calculated using the same expressions in Section 5. The results are summarized in Tables 1 and 2.

Generally, Linear Regression (LR) and LASSO consistently achieve the lowest AMSER across all scenarios, making them the most robust models under data perturbation. LR had the best predictive performance (lowest AMSE) under GADP and LASSO had the best predictive performance under CGADP. The Linear_SVM and Poly_SVM also perform relatively well, with AMSER values lower than those observed for tree-based models such as XGBTrees and Random Forests (RF), which tend to suffer from increased error inflation due to overfitting on the perturbed datasets. SE_equal outperforms SE_prop, likely because SE_prop assigns greater weight to models like XGBTrees and RF, which overfit under perturbation.

Table 1: AMSER of ten predictive modeling techniques under GADP and CGADP of the CE data.

ϑ	Response	Metric	LASSO	LR	Linear_SVM	NNET	Poly_SVM	RF	Radial_SVM	SE_equal	SE_prop	XGBTrees
0.2	FINCBTAX	GADP	1.01	1.01	1.06	1.00	1.16	6.97	3.79	1.44	3.70	5.80
		CGADP	2.00	2.00	2.32	1.00	2.34	4.80	3.46	1.93	3.81	4.88
	SALARYX	GADP	1.01	1.00	1.03	1.00	1.07	5.46	3.23	1.74	3.08	4.76
		CGADP	1.55	1.55	1.74	1.00	1.74	4.04	2.80	2.06	3.19	3.90
0.4	FINCBTAX	GADP	1.00	1.00	1.03	1.00	1.04	5.49	3.14	1.35	3.13	4.68
		CGADP	1.88	1.88	2.18	1.00	2.19	4.52	3.28	1.85	3.61	4.64
	SALARYX	GADP	1.00	1.00	1.01	1.00	1.02	4.38	2.71	1.60	2.64	3.84
		CGADP	1.51	1.51	1.71	1.00	1.71	3.94	2.74	2.02	3.12	3.80
0.6	FINCBTAX	GADP	0.99	0.99	1.01	1.00	1.01	3.50	2.25	1.23	2.33	3.10
		CGADP	1.63	1.63	1.89	1.00	1.91	3.87	2.87	1.67	3.18	4.05
	SALARYX	GADP	1.00	1.00	1.00	1.00	1.00	2.91	2.00	1.41	2.04	2.64
		CGADP	1.44	1.45	1.64	1.00	1.64	3.69	2.62	1.92	2.96	3.56
0.8	FINCBTAX	GADP	0.99	0.99	0.99	1.00	0.99	1.72	1.39	1.10	1.47	1.64
		CGADP	1.27	1.27	1.40	1.00	1.41	2.48	2.02	1.32	2.18	2.83
	SALARYX	GADP	1.00	1.00	1.00	1.00	1.00	1.61	1.32	1.18	1.41	1.50
		CGADP	1.29	1.29	1.41	1.00	1.41	2.87	2.15	1.64	2.30	2.75

Table 2: Best performing predictive modeling technique (lowest AMSE) under GADP and CGADP of the CE data. All values have been divided by 10^8 for readability.

ϑ	Response	GADP		CGADP	
0.2	FINCBTAX	LR	5.06	LR	10.04
	SALARYX	LR	3.57	LASSO	5.52
0.4	FINCBTAX	LR	5.03	LR	9.43
	SALARYX	LR	3.56	LASSO	5.40
0.6	FINCBTAX	LR	4.99	LR	8.21
	SALARYX	LR	3.55	LASSO	5.15
0.8	FINCBTAX	LR	4.98	LASSO	6.37
	SALARYX	LR	3.55	LASSO	4.60

As ϑ increases, the models generally perform better under GADP, despite the non-normality of the CE data. This trend suggests that the reduced privacy protection allows for better retention of statistical properties essential for accurate predictions. Notably, NNET had poor performance under no perturbation and maintained an AMSER of approximately 1 across all conditions, confirming its poor performance on the perturbed data. The Radial_SVM and Poly_SVM achieve lower AMSER values compared to tree-based models, implying that their flexibility allows them to mitigate some of the perturbation effects. However, the increased adaptability of XGBTrees and RF appears to be a disadvantage under GADP and CGADP, leading to worse AMSER values, particularly when ϑ is low. Overall, LR and LASSO emerge as the best-performing predictive models across both perturbation methods. If predictive accuracy is the primary concern under GADP, LR is the optimal choice. However, under CGADP, LASSO is the preferred model, particularly for higher values of ϑ.

7 Discussion

This study examined the impact of data perturbation using GADP and CGADP on various predictive machine learning techniques. Extensive simulations demonstrated that simpler parametric models, such as LR and LASSO, consistently perform well under both perturbation methods due to their reliance on means and covariance structures, which GADP and CGADP aim to preserve. The SVM model with linear kernel also shows promise as a viable alternative. Nonparametric models, including tree-based methods and neural networks, can be effective when the privacy parameter ϑ is large, allowing for higher data utility. However, these models tend to overfit perturbed data, leading to reduced performance. The choice of ϑ should be guided by the dataset's characteristics, regulatory and ethical considerations, and empirical validation of data utility and disclosure risk. GADP is appropriate for data that follows a multivariate normal distribution, whereas CGADP is preferable for non-normal data as it can better preserve distributional characteristics. Conducting post-implementation experiments on CGADP-perturbed data is recommended to assess whether the chosen marginal distributions effectively retain key statistical properties. Ensuring that GADP and CGADP provide the expected level of data utility and privacy protection before disseminating perturbed datasets is crucial for maintaining analytical reliability and protecting sensitive information.

Future research could explore additional machine learning models to further understand performance variations under data perturbation. Moreover, broader comparisons with alternative SDC methods, such as data shuffling and coarsening, could provide deeper insights into their trade-offs in data utility and disclosure risk. Evaluating how different SDC methods impact the predictive performance of various machine learning models is crucial given the growing concerns about data privacy in predictive modeling.

A key limitation of this study is the scope of hyperparameter tuning. Computational constraints prevented an exhaustive exploration of hyperparameter configurations, which could have improved the performance of some models. The SVMs, in particular, incurred high computational costs due to cross-validation, while ensemble methods like XGBTrees and RF required significant runtime for model construction. Future studies could focus on optimizing hyperparameter selection and comparing different predictive model configurations to enhance performance under GADP and CGADP. Additionally, the extensive computational demands highlight the need for sufficient resources when employing complex models. The original simulations required over a year to run on a combination of a 12-core desktop and a 6-core laptop CPU. Increasing the number of predictors or dataset size would further extend computation time. While LR and LASSO scale efficiently, ensemble and SVM models demand more resources, emphasizing the importance of computing power in large-scale applications.

Supplementary Material

The supplementary material includes the following: (1) README: a brief explanation of the supplementary material; (2) a detailed description of the predictive machine learning techniques compared in this paper and additional simulation results; and (3) R code files.

References

Blanco-Justicia A, Sánchez D, Domingo-Ferrer J, Muralidhar K (2022). A critical review on the use (and misuse) of differential privacy in machine learning. *ACM Computing Surveys*, 55(8): 1–16.

Breiman L (2001). Random forests. *Machine Learning*, 45(1): 5–32. https://doi.org/10.1023/A:1010933404324

Carlson M, Salabasis M (2002). A data-swapping technique using ranks—a method for disclosure control. *Research in Official Statistics*, 6(2): 35–64.

Chen T, Guestrin C (2016). XGBoost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Chu AM, Ip CY, Lam BS, So MK (2022). Statistical disclosure control for continuous variables using an extended skew-t copula. *Applied Stochastic Models in Business and Industry*, 38(1): 96–115. https://doi.org/10.1002/asmb.2650

Chu AM, Lam BS, Tiwari A, So MK (2019). An empirical study of applying statistical disclosure control methods to public health research. *International Journal of Environmental Research and Public Health*, 16(22): 4519. https://doi.org/10.3390/ijerph16224519

Duroux R, Scornet E (2018). Impact of subsampling and tree depth on random forests. *ESAIM: Probability and Statistics*, 22: 96–128. https://doi.org/10.1051/ps/2018008

Elliot M, Domingo-Ferrer J (2018). The future of statistical disclosure control. *CoRR*, abs/1812.09204.

Estes JP, Mukherjee B, Taylor JM (2018). Empirical Bayes estimation and prediction using summary-level information from external big data sources adjusting for violations of transportability. *Statistics in Biosciences*, 10: 568–586. https://doi.org/10.1007/s12561-018-9217-4

Gu T, Taylor JM, Cheng W, Mukherjee B (2019). Synthetic data method to incorporate external information into a current study. *Canadian Journal of Statistics*, 47(4): 580–603. https://doi.org/10.1002/cjs.11513

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

Hoshino N (2020). A firm foundation for statistical disclosure control. *Japanese Journal of Statistics and Data Science*, 3: 721–746. https://doi.org/10.1007/s42081-020-00086-9

Hu J, Drechsler J, Kim HJ (2022a). Accuracy gains from privacy amplification through sampling for differential privacy. *Journal of Survey Statistics and Methodology*, 10(3): 688–719. https://doi.org/10.1093/jssam/smac012

Hu J, Savitsky TD, Williams MR (2022b). Private tabular survey data products through synthetic microdata generation. *Journal of Survey Statistics and Methodology*, 10(3): 720–752. https://doi.org/10.1093/jssam/smac001

Kokosi T, De Stavola B, Mitra R, Frayling L, Doherty A, Dove I, et al. (2022). An overview of synthetic administrative data for research. *International Journal of Population Data Science*, 7(1). https://doi.org/10.23889/ijpds.v7i1.1727

Li B, Li X, Zhao Z (2006). Novel algorithm for constructing support vector machine regression ensemble. *Journal of Systems Engineering and Electronics*, 17(3): 541–545. https://doi.org/10.1016/S1004-4132(06)60093-5

Lundell JF (2023). Tuning support vector machines and boosted trees using optimization algorithms. *arXiv preprint arXiv:2303.07400*.

McConville K (2011). Improved estimation for complex surveys using modern regression techniques. Ph.D. thesis, Colorado State University.

Meyer D, Dimitriadou E, Hornik K, Weingessel A, Leisch F (2023). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group. (Formerly: E1071), TU Wien*. R package version 1.7-14.

Moore R (1996). Controlled data swapping for masking public use microdata sets. *US Census Bureau Research Report*, 96(04).

Muralidhar K, Parsa R, Sarathy R (1999). A general additive data perturbation method for database security. *Management Science*, 45(10): 1399–1415. https://doi.org/10.1287/mnsc.45.10.1399

Muralidhar K, Sarathy R (2003). A theoretical basis for perturbation methods. *Statistics and Computing*, 13: 329–335. https://doi.org/10.1023/A:1025610705286

Muralidhar K, Sarathy R (2005). An enhanced data perturbation approach for small data sets. *Decision Sciences*, 36(3): 513–529. https://doi.org/10.1111/j.1540-5414.2005.00082.x

Muralidhar K, Sarathy R (2006). Data shuffling—a new masking approach for numerical data. *Management Science*, 52(5): 658–670. https://doi.org/10.1287/mnsc.1050.0503

R Core Team (2022). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Sarathy R, Muralidhar K, Parsa R (2002). Perturbing nonnormal confidential attributes: The copula approach. *Management Science*, 48(12): 1613–1627. https://doi.org/10.1287/mnsc.48.12.1613.439

Shen X, Liu Y, Shen R (2023). Boosting data analytics with synthetic volume expansion. arXiv preprint: https://arxiv.org/abs/2310.17848.

Tay JK, Narasimhan B, Hastie T (2023). Elastic net regularization paths for all generalized linear models. *Journal of Statistical Software*, 106(1): 1–31. https://doi.org/10.18637/jss.v106.i01

Toth D (2021). *rpms: Recursive Partitioning for Modeling Survey Data*. R package version 0.5.1.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. Springer, New York. ISBN 0-387-95457-0.

Wang Y, Wu X, Hu D (2016). Using randomized response for differential privacy preserving data collection. In: Themis Palpanas and Kostas Stefanidis (Eds.), *Proceedings of the Workshops of the EDBT/ICDT 2016 Joint Conference*, volume 1558, 0090–6778.

Willenborg L, de Waal T (2001). *Elements of Statistical Disclosure Control*. Springer, New York.

Wright MN, Ziegler A (2017). ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, 77(1): 1–17. https://doi.org/10.18637/jss.v077.i01