The Influence of Cellwise Outliers on Propensity Score Matching

ERASMUS UNIVERSITY ROTTERDAM

Erasmus School of Economics

Master Thesis [Econometrics and Management Science]

Author: Dennis van de Noort

Student number: 494883

Supervisor: M. Zhelonkin

Second assessor: A. Naghi

May 1, 2023

Abstract

Propensity score matching is a popular tool in the field of causal inference and particular treatment evaluation. Unfortunately, it typically relies on ordinary least squares, which is highly sensitive to outliers. The current literature has sparsely investigated the influence of cellwise outliers on treatment evaluation methods. Therefore, in this paper, we study the performance of propensity score estimators when the data is contaminated with different sorts of cellwise outliers. A robust algorithm with several outlier detection methods and Multiple Imputation is proposed to deal with these outliers. We see that the robust algorithm with the DDC outlier detection method performs best when there is no correlation between the outliers. The MacroPCA outlier detection method performs best when the outliers are correlated. After applying the robust algorithm, the differences between the propensity score estimators disappear. The robust algorithm is applied to the datasets used in Canavire-Bacarreza et al. (2021) and reveals that outliers are present in the datasets. When conducting a sensitivity analysis on the LaLonde (1986) dataset, the classical estimator breaks down, but the robust estimators perform well.

Keywords: Cellwise outliers, Matching, Robust, Imputation.

Contents

1 Introduction

Causal inference is a fundamental concept in many fields, including economics and social sciences. It refers to the process of identifying the causal relationship between two or more variables. Causal inference is essential because it allows us to understand the impact of an intervention or treatment on an outcome of interest. In the field of economics, for example, policymakers often need to make decisions about the effectiveness of different policy interventions, such as tax policies or education programs. However, we cannot observe both the treated and untreated outcomes for the same individual, as an individual can only receive one treatment, which is the fundamental problem in causal inference. Propensity score matching is a popular tool for solving this problem.

The increase in data availability and computing power in recent years has allowed researchers to collect and analyze large amounts of data. While this has opened up new research opportunities, it has also brought new challenges, particularly in the presence of outliers. Classical econometric methods that assume normal behavior of the data may not be appropriate for causal inference, particularly when outliers are present. Outliers can distort the estimation of treatment effects and lead to biased conclusions. For example, estimation within propensity score matching usually relies on Ordinary Least Squares (OLS), which has a breakdown point of 0%. This means that the presence of a single outlier can cause OLS to break down. Therefore, researchers need to examine the presence of outliers and deal with them carefully, increasing the demand for robust estimation techniques.

Outliers used to be seen as rows or observations. Until Alqallaf et al. (2009) proposed a new concept of cellwise outliers. A new discussion erupted, instead of dealing with entire rows as outliers, only particular cells had to be dealt with. Several papers examined the influence of these cellwise outliers on propensity score matching. Canavire-Bacarreza et al. (2021) found that these cellwise outliers had a negative impact on the bias of the treatment effect estimation. Agostinelli et al. (2015) showed that their robust estimators perform well under both rowwise as cellwise outliers and mentions the need for a new generation of robust estimators.

Our research will focus on the influence of cellwise outliers on propensity score matching methods and propose a robust method. Using a simulation study, we will generate three sorts of outliers: bad leverage points, good leverage points, and vertical outliers. These outliers will be generated in the treatment group, in the control group, and in both. On top of that, we use two different contamination types: one where all outliers are generated independently and one where there is a correlation between the outliers. We use five different propensity score estimators: logistic regression, classification trees, neural networks, gradient boosting models, and random forests. We use three different outlier detection methods are used: Detecting Deviating Cells (Rousseeuw and Van den Bossche, 2018), MacroPCA (Hubert et al., 2019) and Detection-Imputation (Raymaekers and Rousseeuw, 2019). After finding the cellwise outliers, the values are imputed using Multiple Imputation. Finally, the results are analyzed based on the bias, the variance, and the coverage of the coefficients.

The results show that the logistic regression and neural network best estimate the treatment effect when the outlier detection methods and Multiple Imputation are not used. After applying the outlier

detection methods and Multiple Imputation, the differences between the propensity score estimators decreased to a point where there was no superior performer. The DDC method performs best when there is no correlation between the outliers. The MacroPCA method performs best when there is a correlation between the outliers. The DI method performs significantly worse than the other two methods. Simulation results show that using the outliers detection methods and Multiple Imputation substantially improves the performance of the propensity score estimators.

Our methodology is also applied to the datasets used in Canavire-Bacarreza et al. (2021), and a sensitivity analysis is conducted on the LaLonde (1986) dataset. The results for the datasets used in Canavire-Bacarreza et al. (2021) show that the classical estimators show a significant bias in estimating the treatment effect. When using our robust estimators, the estimated treatment effect is closer to the true experimental value. Compared to the simulation study, the DI-MI algorithm outperforms the other two algorithms for these datasets. When conducting the sensitivity analysis, we show that the classical estimator is sensitive to even the smallest amount of contamination, whereas the robust estimators keep performing well.

The remainder of this research is organized as follows. In Section 2, already existing literature on this topic and our contribution to the literature is discussed. After that, we present the methodology used in this paper. Furthermore, in Section 4, we present the framework for the simulation study. Next, Section 5 gives a description of the real datasets our methodology is applied to. Our main findings are presented in Section 6. At last, in Section 7, we summarize our results and provide a conclusion.

2 Literature

In this section, literature that is relevant to our research is described. The use of matching methods for causal inference dates back to the early 20th century when some researchers began recognizing the importance of controlling for confounding variables in observational studies. One of the earliest examples of matching can be found in the work of Cochran (1939), who matched cases and controls based on their ages and other demographic characteristics to investigate the association between smoking and lung cancer.

Matching methods continued to be developed over the following decades, with some researchers advocating for the use of matching to estimate treatment effects in observational studies. In the early 1970s, D. Rubin introduced the concept of the potential outcomes framework, which provided a rigorous theoretical foundation for estimating treatment effects in observational studies. Matching methods offer one way of achieving balance between treated and control units in the potential outcome framework. Then, in a series of papers in the 1970s and 1980s, D. Rubin developed the idea of propensity score matching, in which units are matched based on their probability of receiving the treatment (the propensity score) rather than on individual covariates. This approach has the advantage of being able to balance a large number of covariates simultaneously, and it can also be used to construct weighted estimators that give more weight to well-matched units (Rubin 1973; Rubin 1977; Rubin 1985; Rubin 2001).

However, it wasn't until the 1980s and 1990s that matching methods began to gain widespread use in the causal inference literature (LaLonde 1986; Rosenbaum 1984). Dehejia and Wahba (1999) proposed

the "propensity score with continuous treatment" approach, which extended the propensity score methodology to situations where the treatment variable was continuous rather than binary. This approach has also been extended to the case of multiple treatments (Imbens 2000) and competing risks (Austin and Fine 2019).

Estimating the propensity score by machine learning has only been around for 10-15 years. Before that, it was estimated by logistic or probit regression. Penning et al. (2018) gives an excellent overview of different ways to estimate the propensity scores using Classification And Regression Trees. Random forest was first proposed by Zhao et al. (2016). McCaffrey et al. (2004) proposed to use Generalized Boosted Regression Models. Not until 2018 were neural networks used to estimate the propensity scores (Kallus and Zhou 2018; Setoguchi et al. 2008).

Rosenbaum and Rubin (1983) were the first to mention that outliers can disproportionately impact the matching results, mainly when the number of treated units is small. They recommended diagnostic tests to identify and remove outliers before matching. Stuart (2010) examined the influence of outliers on propensity score matching and found that the presence of outliers can lead to biased estimates of the treatment effect. Both recommended identifying and removing those outliers.

Outliers were, until 2009, seen as rows or observations. Alqallaf et al. (2009) introduced the concept of cellwise outliers. The quality of the match can be affected by the presence of cellwise outliers in the data. These outliers can have a disproportionate influence on the matching process, leading to poor quality matches and biased estimates of treatment effects (Canavire-Bacarreza et al., 2021). However, it was not the first time it was spoken of cellwise outliers. Alfio Marazzi and Werner Stahel brought it up at ETH Zürich as an open problem. It was believed that the available tools were insufficient at that time.

Finding such cellwise outliers is quite a hurdle. Rousseeuw and Van den Bossche (2018) proposed a new method for detecting and handling cellwise outliers in data, named DDC. It is a classification method that uses the Minimum Covariance Determinant (MCD) estimator to identify cells that deviate from the majority of the data. The method is shown to be highly robust to the presence of outliers and performs well on a variety of datasets with different types of outliers.

Another cellwise outlier detection method was proposed by Raymaekers and Rousseeuw (2019), named the DI method. The DI method is a two-step method to handle cellwise outliers in a data matrix: detection and imputation. In the detection step, the cellHandler technique is used to flag outliers. The cellHandler technique is also proposed in their paper. In the imputation step, a robust estimator imputes the missing values in the detected cellwise outliers and updates the covariance matrix. The performance of the DI method is compared with several existing methods on both simulated and real data sets. The results show that the DI method performs well in outlier detection and imputation accuracy.

At last, Hubert et al. (2019) introduced MacroPCA. It is an all-in-one PCA method that can handle different types of data issues, including missing values, cellwise outliers, and row-wise outliers. The performance of MacroPCA is evaluated on simulated data and real-world datasets. The results show that MacroPCA outperforms other methods in outlier detection and is computationally efficient.

Several papers already examined the influence of cellwise outliers on matching methods. Canavire-Bacarreza et al. (2021) examined the relative performance of leading semi-parametric estimators of average

treatment effects in the presence of outliers. Their most important conclusions were: bad leverage points bias estimates of average treatment effects, and good leverage points in the control sample do not affect the estimates of treatment effects. We expand this methodology using different propensity score estimators and state-of-the-art outliers detection and imputation methods.

Austin (2014) examined different algorithms for forming pairs in matching. For example, nearest neighbor, optimal or caliper matching. They found that nearest neighbor and optimal matching induced the same balance in baseline covariates. Furthermore, matching with replacement did not perform superior to caliper matching without replacement. As a result of their conclusions, we only use optimal matching in our paper.

Gharibzadeh et al. (2018) compared different ways of estimating the propensity score. They found that the logistic regression model is efficient when correctly specified. It will also outperform any data adaptive method, like CART or GBM, for modeling the propensity score when the relationship between the propensity score and covariates is linear and additive. We differ from this paper in that we also examine the influence of outliers on these estimators.

3 Methodology

In this section, all the methods used for this research are explained. First, the potential outcome framework is described in Section 3.1. Next, the matching methods are defined in Section 3.2. Next, the propensity score estimators are explained in Section 3.3. Next, the outlier detection methods are described in Section 3.4. And at last, the Multiple Imputation method is explained in Section 3.5.

3.1 Potential Outcome Framework

To create a setup for the matching estimators, we rely on the idea of randomized experiments proposed by Neyman (1923). Rubin (1974) built upon Neyman's ideas and proposed the potential outcomes framework for causal inference. He defines causal effect as the difference of potential outcomes defined on the same observation. In the potential outcome framework, each observation has two potential outcomes for treatment, Y_i^0 and Y_i^1. Y_i^1 if the observation is treated and thus is assigned to the treatment group. Y_i^0 if the observation is not treated and is thus assigned to the control group. Each observation does either receive treatment T_i=1 or does not receive treatment T_i=0. Additionally, each observation has a set of covariates, which is not affected by the treatment. Therefore, for each observation, we observe $(Y_i, T_i \in \{0,1\}, X_i)$, where Y_i is the outcome: $Y_i = T_i Y_i^1 + (1 - T_i) Y_I^0$. Unfortunately, we cannot observe Y_i^0 and Y_i^1 simultaneously. To estimate the average treatment effect on the treated, we thus need to estimate the missing potential outcome for each observation assigned to the treatment group.

3.2 Matching Methods

The potential outcome (Y_i^0) is not observable when an observation is exposed to treatment. Semi-parametric treatment effect estimation methods, such as matching, impute this missing potential outcome

by finding other observations which are similar in terms of covariates but are not exposed to treatment. To consistently estimate and identify the treatment effect, the following assumptions have to be met:

Assumption 1 *(Unconfoundedness)*
$(Y_i^0, Y_i^1) \perp T \,|X.$

Assumption 2 *(Overlap)*
$0 < P(T{=}1|X) < 1.$

Assumption 3 *(Stable Unit Treatment Value Assumption)*
$Y_i = Y_i^t$, *if* $T_i = t.$

The unconfoundedness assumption states that treatment assignment is independent of potential outcomes given the observed covariates, which ensures that any differences in the outcome between the two groups can be attributed to the treatment effect. Next, the overlap assumption establishes a positive probability of receiving treatment for all X. In other words, there should be some similarity between the treated and control groups in terms of their observed covariate values. If there is no overlap, it becomes impossible to find pairs of treated and control group observations, making the matching methods ineffective. At last, the Stable Unit Treatment Value Assumption (SUTVA) states that the potential outcomes of one unit are not influenced by the treatment assignment to other units. Furthermore, for each unit, no different variations of each treatment level lead to different potential outcomes (Imbens and Rubin, 2016). See Imbens (2004) for a discussion on these assumptions.

When estimating the causal effect of a treatment, several different measures of effect can be used, including the average treatment effect (ATE), the average treatment effect on the treated (ATT), the local average treatment effect (LATE), the marginal treatment effect (MTE), and others. The LATE was first introduced by Angrist and Krueger (1991) and was used to estimate the impact of compulsory schooling on earnings by using a quarter of birth as an instrument for education. The MTE was introduced by Imbens and Rubin (1997) and measures the effect of treatment for individuals at the margin of treatment eligibility. The ATT is calculated as the difference between the average outcome for the treated group and the average outcome for the control group, where both groups consist only of individuals who received the treatment. Contrary to the ATE, which estimates the average treatment effect for those who did receive treatment and who did not. The ATT is generally considered a more informative measure of effect when the treatment is not universally received and there is potential for selection bias. In these situations, the ATT may provide a more accurate estimate of the effect of treatment. Furthermore, the LATE and MTE are more specific measures of effect that depend on the individual's compliance with the treatment assignment, which is not applicable in this study (Abadie and Imbens 2006, Angrist and Krueger 2001). In the context of the potential outcome framework, the ATT is defined as follows:

$$\tau = E(Y_i^1 - Y_i^0 | X_i, T_i = 1). \tag{1}$$

As mentioned, the matching estimators impute the missing potential outcome by finding observations with similar covariates and opposite treatment statuses. However, when the number of covariates grows, it

becomes impractical to match these observations on the covariates because of the curse of dimensionality. Therefore, it is necessary to convert these covariates into a scaler, p(x). The most common scaler is the propensity score, which was first introduced by Rosenbaum and Rubin (1983). The propensity score is defined as $\text{p}(X_i) \equiv P(T_i = 1|X_i)$. When conditioning on the propensity score, $X_i|p(X_i)$, the conditional distribution of both the treatment and control groups are equal. This is known as the balancing hypothesis and is stated in Assumption 4.

Assumption 4 *(Balancing Hypothesis)*
$(Y_i^0, Y_i^1) \perp T \mid p(X_i)$.

If all the assumptions are met, observations with the same propensity score have the same distribution of covariates independent of treatment status. The accomplishment of a balanced model depends on the estimation of the propensity score. There are multiple ways to estimate the propensity score. These are explained in the next section.

3.3 Propensity Score

We estimate the propensity score in five different ways in this paper. Namely, logistic regression, classification tree, neural network, gradient boosting model, and a random forest. The upcoming subsections will explain these methods in detail.

3.3.1 Logistic Regression

In this method, a logistic regression model is fitted with the treatment as the dependent variable and the covariates as the independent variables. The predicted probability from the logistic regression model for each individual can then be used as the estimated propensity score. The logistic regression model used to estimate the propensity score can be represented mathematically as follows:

$$\hat{p}(T = 1|X) = \frac{e^{X'B}}{(1 + e^{X'B})}, \tag{2}$$

where p(T=1|X) is the predicted probability of receiving the treatment (T=1) given the observed covariates X.

3.3.2 Classification Tree

Classification trees are commonly used to estimate propensity scores because they are simple, easy to interpret, and can handle a wide range of covariates. To fit a classification tree, we use a recursive partitioning algorithm that splits the sample into subsets based on the value of a selected covariate. Let's assume the j-th split is based on the value of X_j. The tree is represented by a set of binary decision rules of the form:

$$X_j \leq c_j, T = 1, \tag{3}$$

$$X_j > c_j, T = 0, \tag{4}$$

where c_j is the threshold value for the j-th split. The prediction for a new individual with covariates X is obtained by following the path through the tree that corresponds to the individual's covariate values. $\hat{Y}$ denotes the predicted treatment status. In a classification tree, the criterion used to split a leaf node into two child nodes is based on the reduction in a measure of impurity. Here we use the Gini impurity because it is computationally efficient and is less sensitive to outliers than other measures. The split aims to maximize the difference in the treatment status between the two child nodes so that the child nodes become more homogeneous concerning treatment status. Gini impurity measures the probability that a randomly chosen individual from the leaf node would be misclassified if we only use the class proportions in the leaf node to make a prediction. The Gini impurity for a node with k classes is defined as:

$$Gini = 1 - \sum_{k=1}^{K} p_k^2, \tag{5}$$

where K is the total number of classes and p_k is the proportion of individuals in the node with class k. A split that decreases the Gini impurity is preferred. The estimated propensity score for each i is calculated as the predicted probability of receiving treatment:

$$\hat{p}(X) = P(T = 1|X) = \hat{Y}. \tag{6}$$

3.3.3 Neural Network

A single-layer feed-forward neural network can be used to calculate propensity scores by training a binary classification model to predict the treatment status of an individual based on their observed covariates. This is achieved by training the network to learn the relationship between the observed covariates and the treatment status so that it can predict the probability of treatment for a new individual based on their covariate values.

A single-layer feed-forward neural network is a type of artificial neural network that consists of a single layer of artificial neurons or nodes. The nodes in the network are connected by weighted connections, where the weight of each connection represents the strength of the relationship between the inputs and outputs. See Figure 1 for a visual representation.

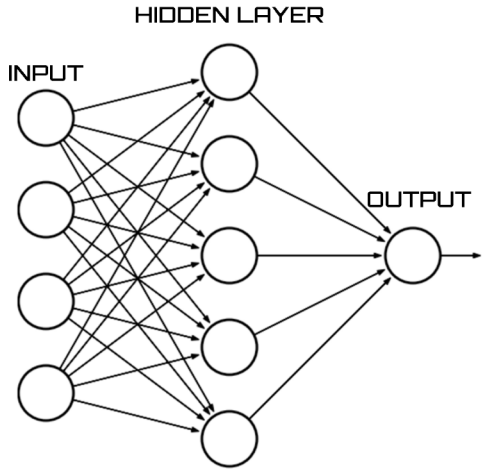


Figure 1: A single layer feed-forward neural network.

The nodes in the network process the input data for the network to produce the output. The processing of the input data starts with the input layer, where the input values are passed through the nodes in the network and weighted by the connections. The weighted inputs are then transformed using an activation function, which maps the inputs to a different range of values, typically between 0 and 1. In a single-layer feed-forward neural network, the activation function is typically a sigmoid function, which has an S-shaped curve and produces outputs between 0 and 1. The sigmoid activation function can be represented mathematically as:

$$\hat{e}_i = f(X_i; \theta) = \frac{1}{1 + e^{-z_i}}, \tag{7}$$

where $z_i = \theta_0 + \Sigma_{j=1}^{p} \theta_j X_{ij}$ is the weighted sum of the inputs for the i-th individual and $\theta = \theta_0, \theta_1, ..., \theta_p$ are the parameters of the model. The transformed inputs are then passed to the output layer, where the final output is produced by combining the transformed inputs from the activation function. The activation function is mathematically equivalent to the logistic regression in Equation (2).

The parameters of the model, θ, are learned by training the network on a set of training data, where the inputs and outputs are known. Next, the network is trained by minimizing a loss function, such as the cross-entropy loss, which measures the difference between the predicted and actual outputs for each training example. Finally, the optimization algorithm updates the model's parameters to minimize the loss function, and the training process is repeated until convergence.

Once the model has been trained, the estimated propensity scores for each individual can be obtained by evaluating the model on their observed covariates, X_i, and using the predicted probability of treatment as the estimated propensity score.

3.3.4 Gradient Boosting Model

In the context of a gradient boosting model (GBM), the propensity score can be estimated as follows:

Let X be a vector of covariates that describe the individual and T be a binary variable indicating treatment status. The goal is to estimate the conditional probability of treatment given covariates, denoted as *P(T=1/X)*. The gradient boosting model can be formulated as follows:

$$F(X) = f_1(X) + f_2(X) + ... + f_M(X), \tag{8}$$

where $f_m(X)$ are weak learner functions combined to form a strong predictor F(X). The functions $f_m(X)$ are decision trees chosen using 5-fold cross-validation. GBM works by training the weak learners sequentially on different parts of the training data, with each model trying to correct the mistakes of the previous model. The final prediction is then made by combining the weak learners' predictions, usually through a weighted sum. Training weak learners and combining their predictions is done iteratively until a desired level of accuracy is achieved or a maximum number of iterations is reached. The number of iterations determines the complexity of the model.

The estimated propensity score for individual i can then be calculated as:

$$\hat{e}_i = P(T = 1|X = x_i) = 1/(1 + \exp(-F(x_i))), \tag{9}$$

where x_i is the vector of covariates for individual i. The gradient boosting model can be trained using a maximum likelihood estimation approach to minimize the negative log-likelihood of the observed

treatment status given the estimated propensity scores. The optimization problem can be formulated as follows:

$$\min \sum -(T_i \log(\hat{e}_i) + (1 - T_i) \log(1 - \hat{e}_i)), \tag{10}$$

where T_i is the observed treatment status for individual i.

3.3.5 Random forest

Random forest uses an ensemble of decision trees for improved prediction accuracy. The concept is that multiple independent models produce better results collectively than individually. In classification tasks, each tree casts a "vote" for the final classification, and the one with the most votes is selected. The decision trees are created using the same method described in Section 3.3.2. After building the trees, the propensity score for each individual is calculated in each tree, then averaged to obtain the final propensity score. The algorithm is summarized in Appendix A (Zhao et al., 2016).

3.4 Outlier Detection Methods

This section provides a detailed explanation of the outlier detection methods used in our research. The Detecting Deviating Cells (DDC) method is explained in Section 3.4.1, the MacroPCA method is explained in Section 3.4.2, and the Detection-Imputation method is explained in Section 3.4.3.

3.4.1 Detecting Deviating Data Cells

Rousseeuw and Van den Bossche (2018) proposed the DDC method. In this paper, a summary of the DDC method is given. However, I refer to the original paper (Rousseeuw and Van den Bossche, 2018) for a more comprehensive and detailed explanation of the algorithm. The DDC method was the first method that considered correlations between variables to detect deviating cells in a multivariate sample. It has no problems with a large number of variables, and there are no limitations on the number of clean rows.

The method first does some prepossessing. It assumes that the data comes from a multivariate Gaussian distribution $N(\mu, \Sigma)$. The variables should therefore be numerical and take on more than a few values. It should be numerical data because it calculates the mean and standard deviation of the data, which are fundamental statistical parameters that require numerical values. Furthermore, the variables should take on more than a few values because the method is designed to detect small shifts or changes in the mean of a process. When the data takes on more than a few values, there is more significant variability in the data, making it easier to detect small shifts or changes in the mean. The method requires the data to be multivariate Gaussian which is necessary to calculate the control limits, which are used to determine if the process has shifted. If these assumptions are not met, the DDC method may not work correctly and produce biased results.

If the variables do not have a Gaussian distribution at their center, they could be transformed manually using a logarithm transformation or more generalized tools for transformations like the Box-Cox transformation or the Yeo-Johnson methods (Yeo and Johnson, 2000).

The method itself follows multiple steps. The first step is the standardization of the variables. Then, for each column j, the location and scale are robustly estimated under the assumption that the variables are centered. To robustly estimate the location and scale, we use the first step of an algorithm for M-estimators described on pages 39-41 in Maronna et al. (2006).

For each column j, we robustly estimate the location and scale using:

$$m_j = \text{robLoc}_i(x_{ij}) \text{ and } s_j = \text{robScale}_i(x_{ij} - m_j), \tag{11}$$

respectively. The functions robLoc and robScale can be found in Appendix B. Hereafter, we standardize X into Z by using the following formula:

$$z_{ij} = (x_{ij} - m_j)/s_j. \tag{12}$$

The second step is to perform univariate outlier detection on all variables formed by Equation (12). We define a new matrix U with entries defined as follows:

$$u_{ij} = \begin{cases} z_{ij}, & \text{if } |z_{ij}| \leq c, \\ \text{NA} & \text{if } |z_{ij}| > c. \end{cases} \tag{13}$$

As a result of the standardization presented in Equation (12), Equation (13) serves as a method to detect column-wise outliers. The cutoff value c is formulated as

$$c = \sqrt{\chi^2_{1,p}}, \tag{14}$$

the value of $\chi^2_{1,p}$ is determined as the p-th quantile of the chi-squared distribution with 1 degree of freedom, where the probability p is set to 0.99.

The third step is for the bivariate relations. If we have two variables, h and j, we calculate their correlation using the following formula:

$$\text{cor}_{jh} = \text{robCorr}_i(u_{ij}, u_{ih}), \tag{15}$$

where robCorr can be found in Appendix C. We will only utilize the correlation between variables j and h when

$$|\text{cor}_{jh}| \geq \text{corrlim}, \tag{16}$$

where corrlim is set to 0.5. Any variables j that fulfill Equation (16) for some $h \neq j$ will be referred to as "connected" and are deemed to contain valuable information about one another. These variables are sufficiently correlated to help predict each other. For these variable pairs, we also compute

$$b_{jh} = \text{robSlope}_i(u_{ij}|u_{hj}), \tag{17}$$

where robSlope is utilized to calculate the slope of a robust regression line without an intercept term. This line predicts the value of j from the value of h.

The fourth step is predicting values. We compute $\hat{z}_{ij}$ for all cells. For each variable j, we define the set H_j as the collection of all variables h that fulfill condition (16), which includes variable j itself. For all i, we then set

$$\hat{z}_{ij} = \text{G}\left(\{b_{jh}u_{ih}\ ;\ h \text{ in } H_j\}\right), \tag{18}$$

where G is a combination rule which ignores any NA values and will yield a zero result if no remaining values are left to combine. We use a weighted mean function for G with weights $w_{jh} = |\text{cor}_{jh}|$. Equation (18) offers the benefit of limiting the impact of an outlying cell, z_{ih}, on $\hat{z}_{ij}$. This is because $|u_{ih}|$ is constrained by the value of c, and as a result, can only influence a single term in the equation.

The fifth step is deshrinking. It is important to note that a prediction method, such as Equation (18), tends to shrink the scale of the entries, which is not desirable as this underestimates the variability. One potential solution would be to reduce the shrinkage applied to the individual terms, $b_{jh}u_{ih}$. However, this approach would not be effective, as these terms may have different signs for different h. To address this shrinkage issue, we apply the combination rule first, followed by a different approach to adjust the level of shrinkage as needed. Therefore, we replace $\hat{z}_{ij}$ by $a_j\hat{z}_{ij}$ where

$$a_j = \text{robSlope}_{i'}(z_{i'j}|\hat{z}_{i'j}). \tag{19}$$

The scaling factor, a_j, is determined by regressing the observed values of z_j on the shrunk predicted values of $\hat{z}_j$.

The sixth step is flagging the cellwise outliers. After calculating the predicted values, $\hat{z}_{ij}$, for all cells in fourth and fifth step, we proceed to compute the standardized cell residuals

$$r_{ij} = \frac{z_{ij} - \hat{z}_{ij}}{\text{robScale}_{i'}(z_{i'j} - \hat{z}_{i'j})}. \tag{20}$$

In each column, j, we flag any cells with $|r_{ij}| > c$, as defined in Equation (14), as anomalous.

Additionally, we construct an "imputed" matrix, $\boldsymbol{Z}_{imp}$, which is the same as the original data matrix, $\boldsymbol{Z}$, except that any deviating cells or missing values are replaced with their corresponding predicted values, $\hat{z}_{ij}$. We do not need this additional feature for the DDC method. However, in Section 3.4.2, we need these imputations, as the DDC method is a part of the MacroPCA method.

The seventh step is flagging the row-wise outliers. If we assume no outliers in multivariate Gaussian data under the null hypothesis, then the distribution of r_{ij} will be similar to a standard Gaussian distribution. Consequently, we can estimate the cumulative distribution function (CDF) of r_{ij}^2 by using the cdf F of χ_1^2. This brings us to the following:

$$T_i = \underset{j=1}{\overset{d}{\text{ave}}} F(r_{ij}^2). \tag{21}$$

After standardizing the T_i using Equation (12), we flag the rows i that exceed the cutoff value c from Equation (14).

The eighth and last step is converting the flagged cellwise outliers to NA values and removing the flagged row-wise outliers from the dataset. These NA values will be later converted to predicted values by Multiple Imputation. This will be explained in Section 3.5.

3.4.2 MacroPCA

The Missingness And Cellwise & Rowwise Outliers PCA (MacroPCA) method was proposed by Hubert et al. (2019). In this paper, a summary of the MacroPCA method is given. However, I refer to the original paper (Hubert et al., 2019) for a more comprehensive and detailed explanation of the algorithm.

It is the first PCA method that deals with both cellwise and row-wise outliers. Moreover, it can also handle missing values.

Assuming there are no outliers or missing values, the objective is to reduce the dimensionality of the data and represent it in a lower-dimensional space, that is,

$$X_{N,p} = 1_N \mu'_p + \mathcal{T}_{N,k}(\mathcal{P}_{p,k})' + \mathcal{E}_{N,p}, \tag{22}$$

where the data matrix is denoted as $X_{N,p}$. N is the number of rows and p the number of variables. 1_N is a column vector with all values equal to 1, μ_p is the location vector, $\mathcal{T}_{N,k}$ is the score matrix, $\mathcal{P}_{p,k}$ is the loadings matrix whose columns span the PCA subspace, and $\mathcal{E}_{N,p}$ is the error matrix. The reduced dimension, k, can range from 1 to p, but it is assumed to be small. The μ_p, $\mathcal{T}_{N,k}$ and $\mathcal{P}_{p,k}$ matrices are unknown, and their estimates will be denoted by m_p, $T_{N,p}$ and $P_{p,k}$.

There are two assumptions needed for this method. First, the data could contain missing values. We assume that these missing values are missing at random (MAR). This assumption implies that the pattern of missingness is dependent only on the observed data, not the unobserved data. This is needed because MacroPCA incorporates ICPCA (Gulrez and Al-Odienat 2015) and MROBPCA (Serneels and Verdonck 2008) that rely on that assumption. Second, the data could contain row-wise outliers. The current row-wise robust methods require that at least 50% of the rows are clean, so this assumption is taken over.

The MacroPCA method consists of two parts. The first part is the DDC method from Section 3.4.1. Its main goal is to identify cellwise outliers and also give an imputation of these outlying cells. The second part constructs the principal components. It follows the methodology of the ICPCA algorithm but uses a variant of the ROBPCA methods for fitting subspaces (Hubert et al., 2005). Throughout the method, they use the following two notations:

- the NA-imputed matrix $\ddot{X}$ only imputes the missing values of X.
- the cell-imputed matrix $\bar{X}$ imputes the missing values of X and has imputed values for outlying cells that do not belong to outlying rows.

Since we have yet to determine which cells and rows are outlying, these matrices will be updated during the method. The DDC method indicates the positions of the cellwise outliers in $I_{c,DDC}$ and flags the outlying rows in $I_{r,DDC}$.

The second part of the MacroPCA method starts by providing an initial indication of which rows are the least outlying. It uses the cell-imputed matrix $\bar{X}^{(0)}_{N,p}$ defined as follows:

1. In all rows, the missing values are replaced by the values $\ddot{x}_i^{(0)}$ as imputed by the DDC method.
2. In the h rows with the fewest cells flagged by DDC, but not in $I_{r,DDC}$, the flagged cells are also replaced by the imputed values of the DDC method.

Where h is determined as $0.5 \leq \alpha = h/n < 1$. This means we can withstand up to a fraction of $1 - \alpha$ outlying rows. To be safe, the default is $\alpha = 0.5$. The outlyingness for each row is calculated in the same

way as in ROBPCA:

$$\text{outl}(\bar{x}_i^{(0)}) = \max_{v \in B} \frac{|v'\bar{x}_i^{(0)} - m_{MCD}(v'\bar{x}_i^{(0)})|}{s_{MCD}(v'\bar{x}_i^{(0)})}, \tag{23}$$

where m_{MCD} and s_{MCD} are univariate MCD location and scale estimates, and the set B contains 250 directions through two data points (Rousseeuw and Leroy, 1987). Finally, the indices of the h rows with the lowest outlyingness and not belonging to $I_{r,DDC}$ are stored in the set H_0.

Next, the number of principal components is chosen. A new cell-imputed matrix is created, $\bar{X}_{N,p}^{(1)}$, which imputes the outlying cells in the rows of H_0 and imputes all NAs from the DDC method in the first part. After that, classical PCA is applied to the $\bar{x}_i^{(1)}$ with $i \in H_0$. From here, the subspace's appropriate dimension k can be derived.

Likewise, to the ICPCA method, this step involves an iterative process to estimate the k-dimensional subspace that fits the data. That is, for $\ddot{X}_{N,p}^{(s)}$ we update all of the imputations of missing cells, whereas for $\bar{X}_{N,p}^{s}$ we update the imputations of the outlying cells in the rows of H_0 as well as the missing cells in all rows. The superscript s represents the iteration, with a maximum of 20 or until convergence. After all iterations we have the NA-imputed matrix $\ddot{X}_{N,p}^{(s)}$ and the cell-imputed matrix $\bar{X}_{N,p}^{s}$ as well as the estimated center $m_p^{(s)}$ and the updated loading matrix $P_{p,k}^{(s)}$. See Hubert et al. (2019) for an extensive explanation of this iterative subspace estimation.

In robust statistics, performing a re-weighting step after an initial estimate is common practice to enhance the statistical efficiency without significantly increasing the computational costs. Here the orthogonal distance of each $\bar{X}_i^s$ to the PCA subspace is used:

$$\overline{OD}_i = \|\bar{x}_i^{(s)} - \{m_p^{(s)} + (\bar{x}_i^{(s)} - m_p^{(s)})P_{p,k}^{(s)}(P_{p,k}^{(s)})'\}\|. \tag{24}$$

The orthogonal distances to the power 2/3 are roughly Gaussian except for the outliers (Hubert et al., 2005), so we compute the cutoff value:

$$c_{od} := \left\{ m_{MCD}(\bar{OD}_j^{2/3}) + s_{MCD}(\bar{OD}_j^{2/3})\Phi^{-1}(0.99) \right\}^{3/2}. \tag{25}$$

where m_{MCD} and s_{MCD} are the same as in Equation (23). All cases for which $\bar{OD}_i \leq c_{od}$ are considered non-outlying, and their indices are stored in H^*. Any index that is also in $I_{r,DDC}$ is removed from H^*. Applying classical PCA to the N^* rows in H^* yields a new center m_p^* and a new loading matrix $P_{p,k}^*$.

Next, we want a robust basis for the estimated subspace. We first project the N^* points of H^* onto the subspace, yielding

$$\bar{T}_{N^*,k} = (\bar{X}_{N^*,p} - 1_{N^*}m_p^{*'})P_{p,k}^*. \tag{26}$$

The center and scatter matrix of the scores $\bar{T}_{N^*,k}$ are estimated by the DetMCD method of Hubert et al. (2012). This results in the final center m_p and final loading $P_{p,k}$.

The last step is computing the scores, predicted values, and residuals. The scores of $\bar{X}_{N,p}$ as computes as $\bar{T}_{N,p} = (\bar{X}_{N,p} - 1_n m_d')P_{p,k}$ and the predictions of $\bar{X}_{N,p}$ as $\hat{\bar{X}}_{N,p} = 1_n m_d' + \bar{T}_{N,p}(P_{d,k})'$. This yield the difference matrix $\bar{X}_{N,p} - \hat{\bar{X}}_{N,p}$, which we then robustly scale by column, yielding the final standardized residual matrix $R_{N,p}$. Cells with $|r_{ij}| > \sqrt{\chi^2_{1,0.99}}$ are considered outliers, which is the same cutoff value as in Equation (14). The flagged outliers are converted to NA values which will be processed by the Multiple Imputation algorithm from Section 3.5.

3.4.3 Detection-Imputation Algorithm

The cellHandler technique was first introduced by Raymaekers and Rousseeuw (2019). It detects outlying cells by combining lasso regression with a step-wise application of constructed cutoff values. The cellHandler assumes that the covariance matrix is known. However, most of the time, that is not the case. That's why they also propose a Detection-Imputation (DI) method, which alternates between flagging outliers using the cellHandler technique and updating the covariance matrix. One condition is that this covariance matrix should be invertible. In this paper, a summary of the DI method is given. However, I refer to the original paper (Raymaekers and Rousseeuw, 2019) for a more comprehensive and detailed explanation of the algorithm.

The cellHandler technique starts by standardizing the columns using a robust univariate estimate of location and scale, resulting in p-variate observation denoted by z_i for i=1, ..., n. This ensures the result will be equivariant to shifting and rescaling. Next to do is to find the cells that are most likely contaminated. The squared Mahalanobis distance $MD^2(z, \mu, \Sigma) = (z - \mu)'\Sigma^{-1}(z - \mu)$ measures the distance between z and the uncontaminated distribution. This concept aims to decrease the Mahalanobis distance of z by altering only a small number of cells. Mathematically this will look like this:

$$MD^2(z - \delta, \mu, \Sigma) = \|\tilde{Y} - \tilde{X}\delta\|_2^2, \tag{27}$$

which is the objective function of a regression without an intercept with $\tilde{Y} := \Sigma^{-1/2}(z-\mu)$ and $\tilde{X} := \Sigma^{-1/2}$ with coefficient vector δ. For the proof, I refer to the original paper (Raymaekers and Rousseeuw 2019). Solving Equation (27) by OLS results in $\delta_{ls} = z - \mu$. However, this replaces the entire row instead of only the outlying cells. A natural solution to this problem is Lasso:

$$\|\tilde{Y} - \tilde{X}\delta\|_2^2+\lambda\|\delta\|_1, \tag{28}$$

where $\|\lambda\|_1$= $|\lambda_p|$+... + $|\lambda_p|$.

The current explanation is incomplete because we have to pay special attention to cells z_j that lie far away. Identifying such far marginal outliers z_j is a relatively simple task, as they have a high degree of univariate outlyingness: $O_j = |z_j - \mu_j|/\sqrt{\Sigma_{jj}}$. The λ_j in the penalty term is weighted with a factor w_j =min$(1, 1.5/O_j)$. This replaces $\|\lambda\|$ in Equation (28) by $\|W\lambda\|_1$ where $W := diag(w_1, ..., w_p)$. We transpose Equation (28) to

$$\|\tilde{Y} - \dot{X}\beta\|_2^2+\lambda\|\beta\|_1, \tag{29}$$

where $\dot{X} := \tilde{X}W^{-1}$ and $\beta := W\lambda$. Hereafter we minimize this equation and transform $\hat{\beta}$ back to $\hat{\lambda}$.

Because Lasso can also remove variables from a model, we use the LAR algorithm from Efron et al. (2004). This algorithm gives us a ranking of the cells from z, in the order from highest to lowest gradient. I refer to the original paper to explain how this algorithm works.

After k steps in the LAR algorithm, we have k candidate cells. These candidate cells are the cells that have the highest absolute correlation with the residual at each step of the LAR algorithm. The question is whether these k candidate cells are sufficient. In other words, can we update these candidate cells while keeping the other cells constant so that the remaining row behaves cleanly? On this notion,

the k candidate cells are edited to maximize the Gaussian likelihood given the remaining cells. This is shown by the following theorem:

Theorem 3.1 *Let the k-variate $\hat{\theta}_1$ be the OLS fit to the regression problem given by*

$$\underset{\theta}{argmin}\|\Sigma^{-1/2}(z-\mu)-(\Sigma^{-1/2})_{.1}\theta_1\|_2^2,$$

where $(\Sigma^{-1/2})_{.1}$ denotes the first k columns of the matrix $\Sigma^{-1/2}$. Then

$$z_1-\hat{\theta}_1=\mu_1+\Sigma_{12}\Sigma_{22}^{-1}(z_2-\mu_2).$$

Here the candidate cells are the first k entries of z, so we denote $z=[z_1,z_2]$,$\mu=[\mu_1,\mu_2]$, and Σ_{11} as the upper left submatrix of Σ of size kxk. The question now is, how many cells should we actually flag? For this, we use the following theorem:

Theorem 3.2 *for every $1\leq k\leq d$ we have:*

1. *The residual sum of squares $RSS_k=\|\Sigma^{-1/2}(z-\mu)-(\Sigma_{.1})\hat{\theta}_1\|_2^2$ of the OLS fit $\hat{\theta}_1$ to the first k cells in the pat hequal the squared partial Mahalanobis distance $MD^2(z_2,\mu_2,\Sigma_{22})=(z_2-\mu_2)'\Sigma_{22}^{-1}(z_2-\mu_2)$.*
2. *For Gaussian data, the difference between two subsequent RSS follows the χ^2 distribution with 1 degree of freedom, i.e., $\Delta_k:=RSS_{k-1}-RSS_k\sim\chi^2(1)$.*

For the proof, I refer to the original paper. The distributional assumption in the second part of Theorem 3.2 is unrealistic but forms a rule of thumb. Compare the δ_k to a cutoff q and flag the cells with $\delta_k>q$. This concludes the cellHandler technique, which is used for the DI method.

The DI method starts by standardizing the columns of the dataset as at the beginning of the cellHandler technique. Next, the initial estimators $\hat{\mu}^0$ and $\hat{\Sigma}^0$ are computed. The 2SGS estimator of Leung et al. (2017) is used. The DI method alternates between the D-step and the I-step.

D-step: Detecting outlying cells across all rows.

The D-step first applies the cellHandler method, which was described above, to each row z_i based on the estimates $\hat{\mu}^{t-1}$ and $\hat{\Sigma}^{t-1}$. We now have a ranking for each row of its cells z_{ij} based on the gradients. A non-increasing sequence of criterion values $C_{ij}:=max_{k\geq k(j)}\Delta_h$. If any cells are missing, they are put to infinity $C_{ih}:=\infty$. If too many cells are flagged, too much information is lost from a variable. Therefore, a maximum of 25% of the cells of a column can be flagged.

I-step: Re-estimate μ and Σ.

The I-step is the same as one step of the EM algorithm, which assumes that flagged cells are missing. Within each row, the flagged cells form one of the active sets evaluated by LAR in cellHandler, so the coefficient $\hat{\theta}_1$ from Theorom 3.1 is known. The E-step from the EM algorithm doesn't need any more computation. The $\hat{\mu}^t$ and $\hat{\Sigma}^2$ are computed similarly to the M-step. The iterative process stops when both $\hat{\mu}^t-\hat{\mu}^{t-1}$ and $\hat{\Sigma}^t-\hat{\Sigma}^{t-1}$ have reduced to a small value. Hereafter the cellHandler technique is applied one more time to the converged $\hat{\mu}^t$ and $\hat{\sigma}^t$. And at last, the DI method ends with the unstandardization of $\hat{\mu}$ and $\hat{\Sigma}$ using the univariate location and scale estimations derived from the original data columns and replaces the cellwise outliers detection by the last cellHandler application with NA.

3.5 Multiple Imputation

There are several ways to deal with missing data. The two most common are imputation and deletion. While deleting missing data can be a straightforward approach, it can lead to biased results due to the loss of information. This is because the remaining sample may not be representative, and the resulting analysis may not accurately reflect the true relationships between variables. Moreover, if the amount of missing data is substantial, it may lead to reduced statistical power, which can decrease the precision of the estimates. The main advantage of imputation is that it retains the full sample size, which can help maintain the sample's representativeness and increase the precision of the estimates. Additionally, imputation can also increase the accuracy of the analysis results by preserving the relationships between variables (Finney and DiStefano 2006; Sterne et al. 2009).

Multiple statistical methods to deal with missing values were reviewed in Newgard and Lewis (2015). Single imputation methods usually result in standard errors which are too small because it does not account for the uncertainty of missing values. Multiple Imputation (MI) is better for handling missing data. It creates multiple imputed datasets and appropriately combines them. The reason why the data is missing is of significant influence on the risk of bias. According to Rubin (1976), there are three types of missing data: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). When the data are MCAR, the randomness is entirely unrelated to the observed or unobserved data. Here MI is an appropriate method for handling missing data. When the data is MAR, MI can still be used. However, the imputation models may need to be more complex. When the data are MNAR, the data cannot be predicted based solely on the observed data, and one needs to specify a model for the missing data mechanism. MI can deal with MNAR. However, it assumes that this missing data model is correctly specified. The imputed data may still be biased when this assumption is not met.

This paper uses the Amelia package from Rstudio (RStudio Team, 2020). Amelia uses a unique bootstrapping approach called the EMB (Expectation-Maximization with Bootstrapping) algorithm to draw imputations for missing values. The approach involves applying the EM algorithm to several bootstrapped samples of the original incomplete data to obtain complete-data parameters. Amelia then uses these parameters to draw imputed values for each bootstrapped parameter set, replacing the missing values with these draws (Honaker et al., 2011).

Amelia's imputation model assumes that observed and unobserved data are multivariate normally distributed. While this assumption may be a simplistic approximation of the true data distribution, studies such as Schafer (1997) and Schafer and Olsen (1998) suggest that this model performs as well as more complex models, even when dealing with categorical or mixed data. The main challenge in imputation is that we can only observe a portion of the complete dataset, D_{obs}, and not the entire dataset, D. Therefore, Amelia assumes that the missing data are missing at random (MAR). Let M be the missingness matrix, where cells m_{ij} take a value of 1 if $d_{ij} \in D_{mis}$ (i.e., if the data point is missing) and 0 otherwise. In other words, M is a matrix that indicates which cells are missing in the dataset. Based on this, the MAR assumption can be defined as follows: the missingness pattern of D depends only on the observed values in D_{obs} and is not related to the unobserved values in D_{mis}:

$$p(M/D) = p(M/D_{obs}). \tag{30}$$

When dealing with missing data, MI focuses on the complete-data parameters, $\theta = (\mu, \Sigma)$. It is important to note that the observed data is represented by D_{obs} and M. Therefore, the likelihood of the observed data can be expressed as $p(D_{obs}, M|\theta)$. By assuming MAR, we can separate this likelihood expression into components that depend on the observed data and components that rely on the unobserved data, allowing for imputation of the missing values:

$$p(D_{obs}, M/\theta) = p(M/D_{obs})p(D_{obs}/\theta). \tag{31}$$

Since the focus is on the inference of the complete-data parameters, we can express the likelihood as:

$$L(\theta|D_{obs}) \propto p(D_{obs}|\theta), \tag{32}$$

We can rewrite the likelihood using the law of iterated expectations as:

$$p(D_{obs}|\theta) = \int p(D|\theta)dD_{mis}. \tag{33}$$

Given this likelihood and a flat prior on θ, we can derive the posterior as:

$$p(\theta|D_{obs}) \propto p(D_{obs}|\theta) = \int p(D|\theta)dD_{mis}. \tag{34}$$

The EMB algorithm combines the EM algorithm and bootstrapping to obtain samples from the posterior. For each sample, the data is bootstrapped to simulate estimation uncertainty, and then the EM algorithm is applied to find the posterior mode for the bootstrapped data. After obtaining multiple samples from the posterior of the complete-data parameters, imputations are created by drawing values of D_{mis} from its distribution, conditional on D_{obs} and the drawn values of θ. For more details on the EMB algorithm, see Honaker and King (2010).

Combining the results from multiple imputed datasets is accomplished using Rubin's rules. These rules provide a framework for averaging the estimates across the multiple imputed datasets while accounting for uncertainty and disagreement in the estimates. The resulting estimates are accompanied by standard errors that reflect the average uncertainty across the imputed datasets and the disagreement in the estimated values across the datasets. See Barnard and Rubin (1999) and Marshall et al. (2009) for an extensive explanation of Rubin's Rules.

The choice of MI above methods like k-Nearest Neighbours (k-NN) is because MI accounts for the extra uncertainty imputing values brings. MI also preserves the variability of the data by creating multiple imputed data sets. And at last, MI is more robust to missing data in the imputation model because it uses all available data to estimate the missing values (Van Buuren 2018).

4 Simulation Study

In this section, the simulation study conducted for our research is explained. This paper aims to examine whether the treatment evaluation methods are robust against cellwise outliers. Therefore it is of interest

to contaminate the data with cellwise outliers and examine the effect on the performance of the methods. Another way to analyze treatment evaluation methods' robustness is by relaxing their assumption. However, in this study, we use many estimators and methods. Relaxing their assumptions would be inconvenient because there are so many. That is why we evaluate the estimators and methods where all assumptions are met.

The simulation setup is as follows:

1. Generate the dataset.
2. Contaminate the dataset with outliers.
3. Calculate the propensity score using LR, CART, NN, GBM, and RF to the datasets from steps 1 and 2 and match them based on the optimal pair matching.
4. Apply MacroPCA, DDC, and DI to datasets from step 3 to detect cellwise outliers and replace them with NA.
5. Apply Multiple Imputation to datasets from step 4 to impute missing values
6. Calculate the propensity score using LR, CART, NN, GBM, and RF to the datasets from step 5 and match them based on the optimal pair matching.
7. Analyse results from steps 3 and 6.

The remainder of this section will look as follows: in Section 4.1, the model specification is given. Next, in Section 4.2, the different kinds of outliers and types of data contamination are explained. Next, in Section 4.3, non-parametric bootstrapping is explained. Last, in Section 4.4, the performance measures of the methods are presented.

4.1 Model

The Data Generating Process (DGP) is as follows:

$$T_i^* = X_i + \mu_i, \tag{35}$$

$$T_i = I(T_i^* > 0), \tag{36}$$

$$Y_i = \beta Z_i + \epsilon_i, \tag{37}$$

where $Z_i = (X_i, T_i)$. The error terms are i.i.d. drawn from a standard normal distribution, that is $\tau_i \sim N(0,1)$ and $\epsilon_i \sim N(0,1)$. All covariates are drawn from a multivariate normal distribution with mean zero such that they are i.i.d. with mean zero, that is, $x_i \sim N(0_k, \Sigma_k)$. We examine two types of underlying covariance matrices Σ_k to examine the robustness of the methods to multiple correlation structures. Type ALYZ refers to the covariance matrices randomly generated by Agostinelli et al. (2015), characterized by relatively low correlations. Type A09 is given by $\sum_{jh} := (-0.9)^{|j-h|}$ and contains both large and small correlations. The parameters are generated as follows: $\beta_i = 1.1^i + 2\dot{s}ign(-1.1^i)$. The *sign* function returns the signs of numeric elements. For positive numbers, 1 is returned, 0 is returned for zero,

and the value -1 is returned for negative numbers. A sample size of n=400 observations is used, and we use p=5 covariates for each observation. In total, there are S=100 simulation runs, which enables us to make claims about the robustness of the estimators we use. We don't use the subscript of the simulation run in the remainder of the paper.

4.2 Contamination

In this section, we describe how the data is contaminated. We consider three types of contamination setups. No contamination, independent contamination, and correlated contamination. Independent contamination is where all the outliers are generated independently. Correlated contamination is where two variables are always contaminated together. So either both or none are contaminated. The outliers for the remaining variables are generated independently. Furthermore, there are three types of outliers, which are shown in Figure 2. Vertical outliers are outliers in the error term. They are away from the bulk in the y-axis and far away from the regression line. It shows different behavior in the dependent variable but not in the explanatory variables. Good leverage points are away from the bulk in both the x-axis and y-axis but are close to the regression line. Bad leverage points are away from the bulk in the x-axis and are far away from the regression line. It shows different behavior in the explanatory variables but not necessarily in the dependent variables. At last, also the location of the outliers in the setup is important. We differentiate between three locations. In the treatment group (T), in the control group (C), and in both the treatment and control groups (T and C). Examining vertical outliers is impossible in the correlated contamination case, because we only contaminate one variable in this case. Therefore, we exclude this option from our research. Concluding, we will examine the performance of the estimators in 33 scenarios. The clean scenario and the contaminated scenario, the contamination scenarios are characterized by two different covariance matrix types, three different types of outliers (not including the vertical outliers in the correlated contamination), and three different locations.

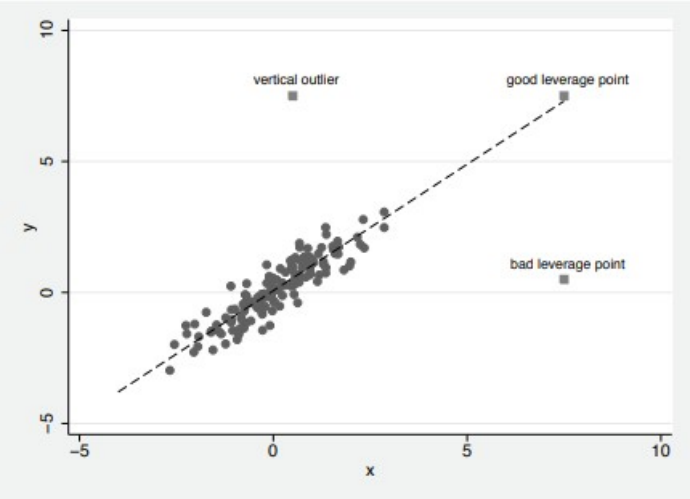


Figure 2: Different sorts of outliers in a simple linear regression.

Several different rowwise contamination models exist, but the Tukey-Huber contamination model

(THCM) is the most common one. Here it is assumed the data is generated from a clear distribution H with probability $1 - \epsilon > 0.5$ and from a random distribution G with probability ϵ.

$$\mathbf{X} = (1 - B)\mathbf{H} + B\mathbf{G}, \tag{38}$$

where $B \sim Bernoulli(\epsilon)$.

Under the THCM, the assumption is that an observation either comes from distribution H or distribution G. Methods developed under this assumption either accept the observation or completely remove it. But there is also the possibility that most of the covariates are clean, but only a few are outlying. Completely removing such an observation would be a loss of information. This resulted in the development of a cellwise outlier model. It was first published by Alqallaf et al. (2009). They propose that a covariate $\boldsymbol{X}$ is generated as follows:

$$\mathbf{X} = \mathbf{(I\text{-}B)W} + \mathbf{BZ}, \tag{39}$$

where $\boldsymbol{B}$ is a diagonal matrix. Its diagonal entries can only take the variables one or zero. This results in a random vector in which some covariates are contaminated, and others are clean. See Figure 3 for a visual comparison between row-wise (left) and cellwise (right) outliers (Raymaekers and Rousseeuw, 2023).

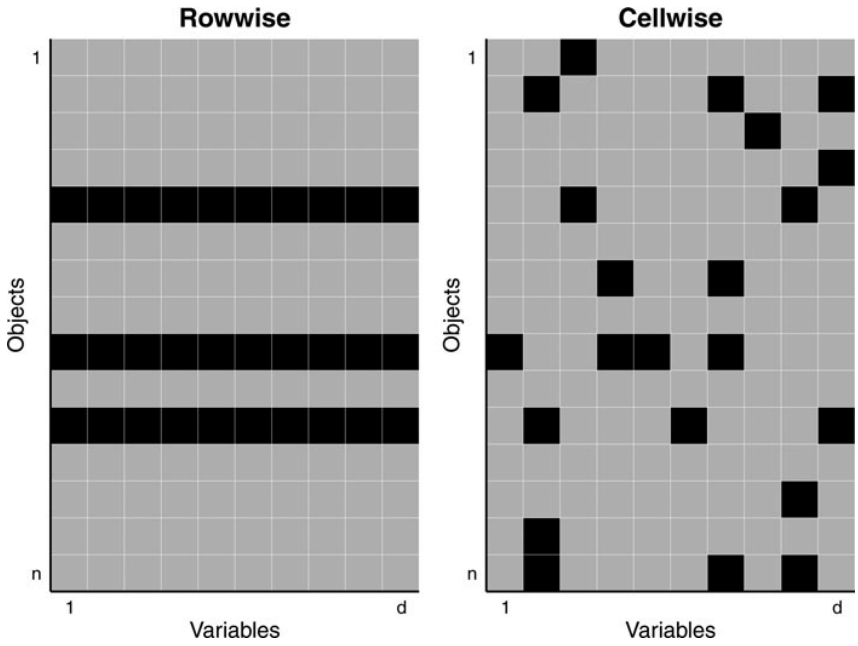


Figure 3: Rowwise outliers (left) and cellwise outliers (right).

When we contaminate our covariates, the matrix W will be the clean matrix as generated in Section 4.1, and the matrix Z will be the contaminated matrix. For the independent contamination of bad leverage points and good leverage points, we will create a binary matrix B where 10% of the matrix are ones. The matrix Z is calculated as follows: $Z = W + \boldsymbol{3}$, where $\mathbf{3}$ is a matrix full of 3's. When the bad leverage points are generated, we call the newly generated contamination matrix X^*. The new dependent variable Y^* is then generated as follows: $Y_i^* = \beta Z_i^* + \epsilon_i$, where $Z^* = (X^*, T)$. We use the same methods for the independent contamination of vertical outliers as for the good and bad leverage points. However,

we will contaminate with an increase of 10. For the vertical outliers, the number of columns is 1 instead of 5.

For the correlated contamination, the matrix Z will be the same as in the independent contamination case. For the binary matrix B, we will set 10% of the matrix to ones. However, the fourth and the fifth column are highly correlated. So if one of the entries for a given row is 1, the other entry is also 1. The same holds for zero. The first three rows are generated independently of each other.

In this study, we don't use binary variables for our covariates. This is because we examine the influence of outliers on estimating the treatment effect. Making an outlier out of a binary variable is not possible. As a result, we don't contaminate our treatment variable as well. Furthermore, we don't generate categorical variables for our covariates. This is for two reasons. First, outlier detection in categorical variables is often more challenging than in continuous variables. This is because categorical variables do not have a natural order or distance metric that can be used to define outliers in the same way as continuous variables. Second, the state-of-the-art outliers detection methods we use in this paper don't possess the option of calculating the distance between non-numerical variables.

4.3 Non-parametric Bootstrapping

In this study, we implement the non-parametric bootstrap before contaminating the dataset. This is because bootstrapping relies on resampling the original dataset to generate new datasets that are similar to the original dataset. This may not be true if there are many outliers. Outliers can have a disproportionate influence on the resampling process, leading to biased estimates of the statistic of interest.

The non-parametric bootstrap works as follows: We start by sampling with replacement from the original dataset $X = (x_1, ..., x_n)$, which results in $X_b = (x_{1b}, ..., x_{nb})$. Where $b = 1, ..., B$ with B the number of bootstrap replications. In this simulation study, we use B=100. Hereafter, we calculate the desired bootstrap statistics by $T_b = T(X_b)$. After completing all the replications, the estimates of the statistics are averages over all replication, that is, $\bar{T} = \frac{1}{B}\sum_{b=1}^{B} T_b$. The standard errors are also calculated over all replications by $\hat{\sigma}_{\bar{T}} = \sqrt{\frac{1}{(B-1)}\sum_{b=1}^{B}(T_b - \bar{T})}$.

Fortunately, only a few assumptions have to be accounted for: random sampling from the original dataset and independent and identically distributed observations. A disadvantage of the non-parametric bootstrap is the computational time, which can grow rapidly compared to other methods. Besides non-parametric bootstrap, the parametric bootstrap also exists. Here the assumption is made that the data comes from a known distribution, which is why it is unsuitable for this simulation study.

We use the following equations to link the bootstrap to the parameter estimates. We let $\hat{\beta}_{jbs}$ denote the j-th parameter in bootstrap run b in simulation run s, which we average over all bootstrap runs:

$$\hat{\beta}_{js} = \frac{1}{B}\sum_{b=1}^{B} \hat{\beta}_{jbs}. \tag{40}$$

To arrive at the final parameters, these also have to be averaged over all simulation runs:

$$\hat{\beta}_{j} = \frac{1}{S}\sum_{s=1}^{S} \hat{\beta}_{js}. \tag{41}$$

The standard errors of the parameter estimates over all simulation runs are calculated as follows:

$$\hat{\sigma}_{\beta_j} = \sqrt{\frac{1}{S-1}\sum_{s=1}^{S}(\hat{\beta}_{js} - \hat{\beta}_j)^2}. \tag{42}$$

In this paper, we use both Multiple Imputation and non-parametric Bootstrapping to improve the reliability of statistical inference. Multiple Imputation is used to reduce the bias of the estimates by creating multiple plausible imputations for the missing values. Non-parametric bootstrap can improve the variability and uncertainty in the estimates from a method by capturing the variation and randomness in the data, providing more robust and reliable estimates of population parameters and statistics.

To evaluate the performance of the methods, we need more evaluation measures than the estimated parameter and the standard error. The performance measures used to evaluate our methods are described in the next section.

4.4 Performance measures

In this section, we will discuss all the performance measures used to examine the performances of the methods used. Both bias and variance are of great importance when determining the accuracy and precision of a prediction. Bias refers to the degree to which a model's predictions differ from the true values. A high-bias model may consistently under or overestimate the true values, leading to reduced precision. Conversely, variance refers to the amount of variability in a model's predictions for different input values. A high variance model may produce significantly different predictions for similar input values, leading to overfitting and reduced accuracy (Hastie et al., 2009).

The bias is calculated as follows:

$$\text{Bias}_j = \frac{1}{S}\sum_{s=1}^{S}(\beta_{ijs} - \hat{\beta}_{js}), \tag{43}$$

where we sum over all simulation runs to end up with a final bias for all parameters.

The variance is calculated as follows:

$$\text{Variance}_j = \sigma_j^2 = \frac{1}{S}\frac{1}{n-1}\sum_{s=1}^{S}\sum_{i=1}^{n}(\beta_{ijs} - \hat{\beta}_{js})^2, \tag{44}$$

where $\hat{\beta}_{js}$ is the average for covariate j in simulation run s over all i. It is also important that the standard error is estimated accurately. That is where the coverage comes in handy. The coverage is defined as:

$$\text{Coverage} = \frac{1}{S}\sum_{s=1}^{S} I[\hat{\beta}_{is} - t_*\hat{\sigma}_{\beta_i} \leq \beta_i \leq \hat{\beta}_{is} + t_*\hat{\sigma}_{\beta_i}], \tag{45}$$

where t_* is the distribution's critical value for α=0.05. We will average over all parameters and simulations to end up with a single value. The parameter is expected to be in the interval in (1-α)% of the time. If that is the case, the standard errors are accurate. However, this only holds if the parameter estimates do not vary too much, which is true if the variance is low.

As we generate the outliers ourselves, we know the exact location of an outlier. DDC, MacroPCA, and DI all flag outliers, so we can evaluate these ODM on their outlier detection accuracy.

$$\text{OutDec} = \frac{1}{n_1}\sum_{i=1}^{n}\sum_{j=1}^{p}\text{Out}_{ij}, \tag{46}$$

where

$$\text{Out}_{ij} = \begin{cases} 1, \text{ if } x_{ij} \text{ is an outlier and correctly detected;} \\ 0, \text{ otherwise,} \end{cases} \tag{47}$$

where n_1 is the total number of outlying cells. At the end of the simulation, we average the metric over the simulation runs, resulting in the final outlier detection error.

5 Real Data

This section will use the datasets described in Canavire-Bacarreza et al. (2021). The findings in this paper are examined based on their robustness and correctness using the methodology from this paper. The datasets are collected from LaLonde (1986) and Dehejia and Wahba (1999). LaLonde (1986) demonstrated that traditional econometric methods could produce misleading results if they do not properly account for the selection bias problem. Matching would be a solution to this problem! Dehejia and Wahba (1999) and Dehejia and Wahba (2002) showed that applying propensity score matching was a good idea to deal with the selection bias. Their findings resulted in a low bias. Smith and Todd (2005) disagreed with this finding and concluded that matching was not the solution to the selection bias problem. Because matching produced a significant bias when applied to the full LaLonde sample. Dehejia (2005) argued that the issue was not with the matching method itself but rather with the accuracy of the propensity score estimation and the lack of covariate balance. Canavire-Bacarreza et al. (2021) suggests that the problem is with outliers in the data, which disrupt the balance in the covariates. This resulted in the rejection of the conclusions of Dehejia (2005) that the propensity score was not estimated correctly and for Smith and Todd (2005) that matching was not an appropriate solution. In this section, we aim to provide further empirical evidence that the issue of covariate imbalance in treatment effect estimation is attributed mainly to the presence of outliers. Moreover, using our methodology, we try to enhance the precision and accuracy of the ATT estimates in the presence of covariate imbalance.

We estimate the treatment effect on the treated based on LaLonde's full sample treatment group and Dehejia and Wahba's (DW's) subsample treatment group. For the comparison groups, we use the non-experimental comparison groups constructed by LaLonde (1986) from the Population Survey of Income Dynamics (PSID) and the Current Population Survey (CPS). The difference between LaLonde's and DW's samples is that DW excluded observations for which earnings data in 1974 were not obtainable.

The dependent variable is the real income in 1978 (*RE78*). The propensity score for the DW dataset is estimated using the following equation:

$$\begin{aligned} log(treat_i) = {} & \gamma_{1,emp} Age_i + \gamma_{2,emp} Age_i^2 + \gamma_{3,emp} Age_i^3 + \gamma_{4,emp} School_i + \gamma_{5,emp} School_i^2 \\ & + \gamma_{6,emp} Married_i + \gamma_{7,emp} NoDegree_i + \gamma_{8,emp} Black_i \\ & + \gamma_{9,emp} Hisp_i + \gamma_{10,emp} RE74_i + \gamma_{11,emp} RE75_i + \gamma_{12,emp} School_i \times RE74_i, \end{aligned} \tag{48}$$

where the variable names are self-explanatory. For the LaLonde dataset, the terms containing RE74 are

removed. To obtain an estimate of the treatment effect, the following equation is used:

$$RE78_i = \beta_{0,emp} + \beta_{1,emp}Age_i + \beta_{2,emp}Age_i^2 + \beta_{3,emp}Educ_i + \beta_{4,emp}Black_i + \beta_{5,emp}Hispan_i + \beta_{6,emp}Nodegree_i + \beta_{7,emp}RE75_i + \beta_{8,emp}Treat_i + \epsilon_i. \quad (49)$$

Hereafter, we will conduct a sensitivity analysis to assess the robustness of the proposed robust methods. We will use the same LaLonde treatment group as before, but we will now also use the LaLonde control group. The control group consists of 425 observations, making the dataset 722 observations. To estimate the propensity score, we use the following equation:

$$log(treat_i) = \gamma_{1,emp}Age_i + \gamma_{2,emp}Age_i^2 + \gamma_{3,emp}Age_i^3 + \gamma_{4,emp}School_i + \gamma_{5,emp}School_i^2 + \gamma_{6,emp}Married_i + \gamma_{7,emp}NoDegree_i + \gamma_{8,emp}Black_i + \gamma_{9,emp}Hisp_i + \gamma_{11,emp}RE75_i, \quad (50)$$

which is the same as Equation (48), but without *RE74*. We use Equation (49) to estimate the ATT. To do the sensitivity analysis, we alter just two observations. For observations 236 and 609, we alter the ethnicity to black.

Observation	RE78	Treat	Age	Educ	Black	Hispan	Married	Nodregree	RE75
236	16717.12	1	49	8	1	0	1	1	7285.95
609	30247.50	0	26	8	1	0	1	1	36941.27

Note that we only altered the ethnicity. This could hardly be an extreme case or a clear outlier. The results are discussed in Section 6.2

6 Results

In this section, the results of the simulation study and the results of the real data are examined. Section 6.1 presents the findings from the simulation study outlined in Section 4, examining several scenarios that vary in the location of the contamination and the type of contamination. Section 6.2 explores the outcomes derived from analyzing the datasets described in Section 5.

6.1 Simulation Study Results

This section investigates the impact of outliers on estimating treatment effects in various scenarios by utilizing different matching estimators and outlier detection methods. Table 1 illustrates the effectiveness of the estimators in estimating the average treatment effect on the treated (ATT) in both the absence and presence of outliers. It reports the bias observed across 100 simulations and 100 bootstrap replications using a sample size of 400 and 5 covariates. The second column corresponds to the clean case without any outliers. The third through fifth columns represent the presence of bad leverage points in the treatment group, control group, and both groups, respectively. The sixth through eighth columns display

Table 1: Simulated bias of Average Treatment Effect on the Treated using the A09 type correlation structure.

Panel A: Independent contamination		Bad Leverage Point			Good leverage point			Vertical outlier		
Estimators:	Clean	in T and C	in T	in C	in T and C	in T	in C	in T and C	in T	in C
LR	**-0.002**	0.062	**0.111**	0.063	0.008	-0.000	-0.001	**1.008**	1.012	**-0.011**
CART	0.002	0.091	0.127	0.011	0.007	-0.006	0.009	1.051	1.017	0.013
NN	-0.004	**0.048**	0.137	**-0.007**	**0.005**	**0.000**	**-0.001**	1.042	**1.003**	0.014
GBM	0.008	0.114	0.123	0.026	0.007	0.001	0.009	1.044	1.035	0.011
RF	0.009	0.101	0.131	0.012	0.012	0.001	0.015	1.049	1.032	0.020
Panel B: Correlated contamination										
LR	**-0.002**	0.046	0.072	0.065	-0.003	0.005	**-0.002**			
CART	0.002	0.053	0.075	0.024	0.003	**0.000**	0.004			
NN	-0.004	**0.043**	0.094	**0.013**	**-0.001**	0.006	0.003			
GBM	0.008	0.068	**0.070**	0.040	0.010	0.005	0.011			
RF	0.009	0.060	0.092	0.029	0.008	0.010	0.011			

the presence of good leverage points, while the ninth through eleventh columns indicate the presence of vertical outliers.

Table 1 showcases the outcomes for the bias of propensity score estimators, with bold numbers denoting the top-performing method for a specific metric. The results reveal that the LR and NN estimators exhibit superior performance over the other estimators across all metrics presented in Table 1. Nonetheless, we will proceed to analyze all the findings in detail.

First of all, in the absence of outliers, all the estimators demonstrate satisfactory performance, consistent with the findings reported by (Busso et al. 2009; Busso et al. 2014). However, when outliers are present, the estimates exhibit a more significant bias, mainly when the outliers are bad leverage points or vertical outliers. In contrast, good leverage points do not result in biased estimates.

Secondly, all estimates show some bias when bad leverage points are present. The size of the bias is dependent on the location of the outliers. When the outlier is located in the treatment group, it may be difficult to identify a suitable match. As a result, the resulting comparison group may be systematically different from the treatment group in ways not accounted for by the matching variables. This can lead to higher bias in the estimated treatment effect. When the bad leverage point is in the control group, the magnitude of the bias will be smaller because the outlying observations are less likely to be chosen as counterfactuals.

Third, the bias is extremely large when there are vertical outliers in the treatment group. Regression analysis assumes that the relationship between the predictor and outcome variables is linear. In the presence of vertical outliers, the linear relationship may no longer hold, and the estimator may be biased in trying to fit a straight line to the data. As a result, the estimated coefficients may be highly sensitive to vertical outliers, resulting in a significant bias.

Fourth, in the correlated contamination scenario, the LR estimator performs worse than the independent contamination scenario compared to the other estimators. A possible explanation for this could be that two variables are related in the correlated contamination scenario, and their relationship with the other variables may not be linear. In this case, LR may not be able to capture the non-linear relationship between the predictors and the outcome, resulting in higher bias. Neural networks, for example, are

better suited to capturing non-linear relationships, so their bias may not be as affected by the non-linear relationship between the predictors and the outcome.

Table 2 showcases the outcomes for the variance of propensity score estimators for the ATT, with bold numbers denoting the top-performing method for a specific metric.

Table 2: Simulated variance of Average Treatment Effect on the Treated using the A09 type correlation structure.

Panel A: Independent contamination		Bad Leverage Point			Good leverage point			Vertical outlier		
Estimators:	Clean	in T and C	in T	in C	in T and C	in T	in C	in T and C	in T	in C
LR	**0.095**	**0.347**	**0.271**	**0.221**	**0.104**	**0.114**	0.095	**0.307**	**0.223**	**0.230**
CART	0.096	0.355	0.290	0.239	0.106	0.116	0.095	0.311	0.225	0.233
NN	0.096	0.352	0.298	0.243	0.105	0.116	**0.095**	0.310	0.224	0.232
GBM	0.095	0.351	0.280	0.232	0.105	0.114	0.095	0.309	0.224	0.232
RF	0.096	0.363	0.295	0.243	0.106	0.115	0.095	0.309	0.224	0.231
Panel B: Correlated contamination										
LR	**0.095**	**0.237**	**0.203**	**0.172**	**0.103**	**0.110**	0.095			
CART	0.096	0.241	0.215	0.184	0.104	0.112	0.095			
NN	0.096	0.240	0.221	0.188	0.104	0.112	0.095			
GBM	0.095	0.239	0.208	0.181	0.103	0.110	**0.094**			
RF	0.096	0.243	0.219	0.188	0.104	0.111	0.095			

The first thing that is noticeable from Table 2 is that the difference in variances between the estimators is small. Nevertheless, the LR estimators outperform the other estimators in almost every scenario. A combination of the lowest bias and the lowest variance would suggest that the LR estimator would be the most suitable when dealing with outliers.

When there are good leverage points, the magnitude of the variance does not seem to increase by much. An explanation for this could be that these good leverage points follow the data pattern and are influential, which can help give accurate predictions. Bad leverage points and vertical outliers refer to observations that do not follow the data and are influential. As a result, they can shift the relationship between the explanatory variables and the dependent variable, increasing variance.

Table 3 showcases the outcomes for the coverage of propensity score estimators for the ATT, with bold numbers denoting the top-performing method for a specific metric. Ideally, this value would be close to 95%, indicating that the standard errors are accurate.

Table 3: Simulated coverage of Average Treatment Effect on the Treated using the A09 type correlation structure.

Panel A: Independent contamination		Bad Leverage Point			Good leverage point			Vertical outlier		
Estimators:	Clean	in T and C	in T	in C	in T and C	in T	in C	in T and C	in T	in C
LR	**95.207**	97.099	**96.946**	99.018	93.098	95.249	**92.644**	5.254	0.176	**100.000**
CART	92.582	96.315	96.419	99.423	91.180	93.133	88.507	5.318	1.449	**100.000**
NN	93.708	**98.497**	94.771	**100.000**	**93.273**	**95.650**	89.416	4.459	**1.591**	99.340
GBM	92.969	96.964	96.618	98.102	91.736	93.772	89.792	**5.460**	0.504	99.944
RF	94.558	96.697	94.314	**100.000**	93.001	93.651	90.978	3.301	0.684	99.387
Panel B: Correlated contamination										
LR	**95.207**	92.300	89.731	**94.337**	93.030	92.771	88.078			
CART	92.582	93.704	89.908	93.508	93.741	92.345	**90.970**			
NN	93.708	**94.607**	90.684	92.473	93.903	93.505	89.530			
GBM	92.969	91.138	**92.529**	91.781	**94.246**	92.991	88.359			
RF	94.558	89.242	92.350	92.073	93.894	**93.669**	90.680			

The coverage percentage of the LR estimator in the clean scenario in Table 3 is close to the ideal 95% value. The other estimator lies further away. Since we employ the bootstrap method, any deviations in coverage may result from a limited number of bootstrap replications, a restricted number of simulation runs, or a combination of both. Despite these small deviations, we acknowledge that computational constraints prevent us from investigating these cases for a larger number of bootstrap replicates and/or simulation runs. The coverage depends on both the parameter estimates and the variance, as the method involves computing confidence intervals. A large variance or significant variation in parameter estimates may lead to incorrect conclusions as the confidence intervals become excessively large and unrealistic. This is also the case when vertical outliers are present in the treatment group. The large bias from Table 1 now results in an unrealistic and unreliable coverage.

Table 4 shows the outcomes of the simulation study when implementing the outlier detection methods in combination with Multiple Imputation, now denoted as ODM-MI. The second column indicates a scenario in which no outliers are present, and the ODM-MI algorithm is applied to assess its performance under such circumstances. If the bias is low, it implies that utilizing the ODM-MI algorithm does not impact the treatment effect estimates if no outliers exist in the data. The remaining columns follow the same interpretation as Table 1.

First of all, as you can see in column 2 in Table 4, the DDC and MacroPCA methods, combined with Multiple Imputation, perform well in a scenario without outliers. The DI method performs worse in terms of bias. Where the bias of DDC and MacroPCA is close to 0 in most scenarios, the bias of DI is further away. Secondly, in Table 1, the LR and NN estimators were superior to the others. After applying the ODM-MI algorithm, there is no estimator that performs better overall. The contamination in the data may have introduced bias or increased variability in the different propensity score estimators, making LR and NN perform better than the others. However, by applying the ODM-MI algorithm, the contaminated values have been identified and imputed, reducing the impact of the contamination on the estimators. As a result, the different propensity score estimators have become more similar in their performance, as the differences caused by the contamination have been reduced. Third, the ODM-MI algorithm reduces the bias significantly in the scenario where vertical outliers are present in the treatment group. It is reduced from close to 1 to close to 0. Next, the bias for the bad leverage points is also reduced to a reasonable magnitude. Fourth, for almost all estimators, the bias is the highest when the outlier is present in the treatment group. This is a logical conclusion since when an outlier exists in the treatment group, it must be included in the matching process, resulting in a bias in the estimation of treatment effects. Fifth, when you compare the independent contamination bias with the correlated contamination bias, it is noticeable that the ODM-MI algorithm also succeeds in locating the correlated contaminated variables and correctly imputes them.

Table 5 showcases the outcomes for the variance of propensity score estimators of the ATT after applying the ODM-MI algorithm. In this table, there are no bold numbers for the best-performing estimators. This is because the performance of the estimators is very similar, with negligible differences between them.

The results in Table 5 show considerably different results than in Table 2. Table 2 shows that the

Table 4: Simulated bias of Average Treatment Effect on the Treated using the A09 type correlation structure after applying the outlier detection methods and the Multiple Imputation.

Panel A: Independent contamination		Bad Leverage Point			Good leverage point			Vertical outlier		
Estimators:	Clean	in T and C	in T	in C	in T and C	in T	in C	in T and C	in T	in C
DDC										
LR	0.006	-0.006	-0.025	0.008	0.007	-0.001	-0.004	**0.062**	0.008	**0.008**
CART	**-0.000**	-0.005	-0.027	0.007	0.003	-0.007	-0.007	0.069	0.004	0.013
NN	0.002	**-0.004**	-0.025	0.011	0.009	**-0.001**	-0.013	0.075	0.006	0.014
GBM	0.007	-0.004	**-0.024**	0.008	0.008	-0.001	**-0.004**	0.073	0.007	0.017
RF	0.002	-0.011	-0.026	**0.006**	**0.001**	-0.003	-0.013	0.070	**0.003**	0.011
DI										
LR	-0.072	-0.073	-0.098	-0.071	**-0.004**	**-0.007**	-0.006	0.057	**0.015**	-0.033
CART	-0.072	-0.075	-0.099	-0.071	-0.016	-0.011	-0.009	0.059	0.026	-0.027
NN	-0.069	-0.073	-0.099	-0.071	-0.008	-0.012	**-0.001**	**0.057**	0.026	-0.028
GBM	**-0.068**	**-0.072**	**-0.096**	**-0.065**	-0.005	-0.011	-0.002	0.062	0.027	-0.027
RF	-0.069	-0.072	-0.099	-0.069	-0.005	-0.009	-0.017	0.076	0.030	**-0.026**
MacroPCA										
LR	-0.029	-0.006	-0.048	-0.013	**0.002**	**0.003**	**-0.008**	-0.018	-0.028	-0.026
CART	-0.024	-0.003	-0.045	-0.011	0.015	0.004	-0.023	0.004	-0.021	-0.012
NN	-0.027	-0.002	-0.046	-0.010	0.008	0.010	-0.023	**-0.001**	-0.027	-0.018
GBM	-0.022	**0.001**	**-0.044**	-0.004	0.004	0.013	-0.025	0.001	**-0.020**	-0.014
RF	**-0.022**	0.004	-0.045	**-0.003**	0.004	0.011	-0.017	0.004	-0.020	**-0.011**
Panel A: Correlated contamination										
DDC										
LR	0.006	-0.005	-0.019	0.006	0.003	-0.009	-0.012			
CART	**-0.000**	-0.007	-0.019	**0.003**	**0.001**	-0.012	**-0.001**			
NN	0.002	**-0.002**	-0.015	0.009	0.001	-0.012	0.001			
GBM	0.007	-0.003	**-0.013**	0.004	0.005	**-0.008**	-0.005			
RF	0.002	-0.008	-0.020	0.003	0.004	-0.012	-0.006			
DI										
LR	-0.072	-0.080	-0.083	-0.072	-0.006	-0.006	-0.013			
CART	-0.072	**-0.079**	-0.086	-0.073	**-0.001**	-0.007	-0.005			
NN	-0.069	-0.081	-0.085	-0.075	-0.002	-0.003	-0.005			
GBM	**-0.068**	-0.081	-0.085	**-0.071**	-0.008	-0.008	**-0.002**			
RF	-0.069	-0.083	**-0.080**	-0.073	-0.006	**-0.001**	-0.012			
MacroPCA										
LR	-0.029	-0.013	-0.035	-0.014	**0.001**	-0.034	-0.028			
CART	-0.024	-0.010	-0.035	-0.006	0.011	**-0.005**	-0.005			
NN	-0.027	-0.008	-0.037	-0.008	0.003	-0.034	-0.003			
GBM	**-0.022**	-0.004	**-0.031**	-0.005	0.006	-0.031	-0.007			
RF	-0.022	**-0.003**	-0.031	**-0.003**	0.002	-0.020	**-0.000**			

variances were relatively constant for the estimators. However, they varied significantly across the outliers (bad leverage point, good leverage point, and vertical outlier) and where the outlier occurred (T&C, T and C). After applying ODM-MI, the variances for the bad leverage points are the same as in the clean scenario. The good leverage points in the treatment and control sample show larger variances, but not as high as before applying ODM-MI. The results from the independent and correlated contamination scenarios are comparable, which means that the ODM-MI algorithm can also find the outliers in the contaminated scenario and accurately predict them. The DI outlier detection method, combined with MI, has a high bias and a low variance, suggesting that DI does not capture the complexity of the underlying data when the dataset is contaminated. While the following result has been found, its confirmation and further details will be presented at the end of this section by the findings of Table 7.

Table 6 showcases the outcomes for the variance of propensity score estimators, with bold numbers denoting the top-performing method for a specific metric.

Table 5: Simulated variance of Average Treatment Effect on the Treated using the A09 type correlation structure after applying the outlier detection methods and the Multiple Imputation.

Panel A: Independent contamination		Bad Leverage Point			Good leverage point			Vertical outlier		
Estimators:	Clean	in T and C	in T	in C	in T and C	in T	in C	in T and C	in T	in C
DDC										
LR	0.010	0.012	0.012	0.010	0.129	0.032	0.031	0.046	0.011	0.026
CART	0.010	0.013	0.012	0.011	0.135	0.035	0.032	0.047	0.011	0.027
NN	0.011	0.012	0.012	0.010	0.130	0.034	0.031	0.047	0.012	0.026
GBM	0.010	0.012	0.012	0.010	0.129	0.033	0.031	0.046	0.011	0.026
RF	0.010	0.012	0.012	0.010	0.129	0.034	0.031	0.046	0.011	0.026
DI										
LR	0.010	0.012	0.012	0.011	0.086	0.020	0.034	0.044	0.013	0.032
CART	0.011	0.012	0.013	0.011	0.089	0.022	0.036	0.045	0.014	0.033
NN	0.011	0.012	0.012	0.011	0.087	0.021	0.035	0.043	0.013	0.032
GBM	0.010	0.012	0.012	0.010	0.087	0.021	0.035	0.043	0.013	0.033
RF	0.011	0.012	0.012	0.011	0.088	0.021	0.035	0.044	0.013	0.032
MacroPCA										
LR	0.010	0.014	0.015	0.010	0.121	0.022	0.016	0.034	0.011	0.021
CART	0.010	0.015	0.015	0.011	0.125	0.023	0.018	0.035	0.011	0.022
NN	0.010	0.014	0.015	0.010	0.120	0.022	0.017	0.035	0.011	0.022
GBM	0.010	0.014	0.015	0.010	0.119	0.022	0.017	0.034	0.011	0.022
RF	0.010	0.014	0.015	0.010	0.125	0.023	0.017	0.035	0.011	0.022
Panel A: Correlated contamination										
DDC										
LR	0.010	0.012	0.012	0.010	0.073	0.024	0.020			
CART	0.010	0.013	0.013	0.011	0.075	0.026	0.023			
NN	0.011	0.012	0.012	0.010	0.075	0.025	0.021			
GBM	0.010	0.012	0.012	0.010	0.073	0.024	0.021			
RF	0.010	0.012	0.012	0.010	0.075	0.025	0.021			
DI										
LR	0.010	0.012	0.012	0.010	0.062	0.020	0.026			
CART	0.011	0.013	0.013	0.011	0.065	0.021	0.028			
NN	0.011	0.012	0.012	0.010	0.064	0.020	0.026			
GBM	0.010	0.012	0.012	0.011	0.064	0.020	0.026			
RF	0.011	0.012	0.012	0.010	0.066	0.020	0.027			
MacroPCA										
LR	0.010	0.013	0.013	0.010	0.055	0.016	0.013			
CART	0.010	0.014	0.014	0.011	0.057	0.017	0.015			
NN	0.010	0.013	0.014	0.010	0.055	0.016	0.014			
GBM	0.010	0.013	0.014	0.010	0.055	0.016	0.014			
RF	0.010	0.013	0.014	0.010	0.057	0.016	0.013			

Table 6 shows us the coverage after applying ODM-MI. Again it is noticeable that DI performs worse. Because the coverage is a combination of the bias and the square root of the variance, the scenarios where the coverage is low, for example, a good leverage point in the treatment sample when using DI, is explained by the fact that the bias and/or the variance is high. Unfortunately, the coverage is lower than the ideal value of 95% in almost all scenarios.

Table 7 shows the outlier detection performance of the outlier detection methods. Where the outlier detection methods are in the first column. The table looks the same as the previous tables, except there is no clean scenario column.

The expectations from the previous results are confirmed in Table 7. The DI method is worse at finding outliers than the other two methods. When the data is independently contaminated, the DDC

Table 6: Simulated coverage of Average Treatment Effect on the Treated using the A09 type correlation structure after applying the outlier detection methods and the Multiple Imputation.

Panel A: Independent contamination		Bad Leverage Point			Good leverage point			Vertical outlier		
Estimators:	Clean	in T and C	in T	in C	in T and C	in T	in C	in T and C	in T	in C
DDC										
LR	88.842	**95.280**	86.230	91.007	**94.048**	91.569	96.173	**96.951**	87.860	97.319
CART	89.497	93.969	**87.466**	89.093	92.215	91.251	**96.451**	96.305	88.878	96.114
NN	86.242	94.874	85.113	90.215	93.539	**98.529**	93.474	96.261	86.334	96.248
GBM	87.504	93.247	85.986	89.077	92.882	99.520	94.038	96.190	87.040	96.547
RF	**90.563**	93.045	85.005	**91.819**	93.594	90.412	96.186	96.819	**91.695**	**98.460**
DI										
LR	79.645	83.740	79.464	**84.541**	76.974	81.836	81.364	**91.655**	87.928	89.780
CART	80.438	83.080	78.113	84.338	**78.631**	**85.853**	**88.138**	90.590	87.553	87.139
NN	80.923	83.205	76.563	80.097	73.395	83.889	85.333	90.495	86.254	89.502
GBM	**83.328**	82.914	**81.874**	83.539	77.546	83.768	87.385	91.324	86.919	88.141
RF	82.756	**86.823**	79.950	83.025	76.224	84.200	84.263	94.193	**87.985**	**90.373**
MacroPCA										
LR	89.583	91.752	92.854	91.743	90.526	89.571	90.748	96.929	90.688	95.895
CART	**90.492**	91.585	92.056	91.517	**98.156**	88.219	93.543	95.167	90.346	95.217
NN	89.726	**94.535**	**93.071**	91.721	94.777	87.297	94.047	96.194	90.325	**96.152**
GBM	88.626	92.916	91.427	**93.018**	95.011	90.674	**95.096**	96.142	91.237	95.309
RF	88.383	93.530	91.110	92.501	96.567	**94.860**	93.865	**96.938**	**93.405**	94.190
Panel A: Correlated contamination										
DDC										
LR	88.842	88.881	89.480	92.602	89.523	93.879	**96.185**			
CART	89.497	88.231	**91.610**	92.111	90.860	**96.058**	95.817			
NN	86.242	88.960	90.380	90.449	88.649	94.162	94.756			
GBM	87.504	86.435	88.950	89.572	**91.627**	93.531	95.593			
RF	**90.563**	**89.147**	90.466	**94.077**	87.036	94.681	96.067			
DI										
LR	79.645	81.239	82.741	81.332	71.016	77.389	86.175			
CART	80.438	83.082	**84.693**	82.595	71.637	72.609	88.822			
NN	80.923	81.541	84.064	79.252	71.015	**79.509**	88.315			
GBM	**83.328**	**84.916**	81.865	80.333	69.671	72.823	88.357			
RF	82.756	78.949	82.243	**84.078**	**72.256**	73.144	**89.482**			
MacroPCA										
LR	89.583	**96.913**	**93.835**	91.208	**95.093**	91.031	94.418			
CART	**90.492**	94.310	93.374	92.267	92.759	**91.523**	94.235			
NN	89.726	96.279	92.405	92.318	93.670	90.453	94.072			
GBM	88.626	95.198	92.032	90.795	93.895	90.071	92.551			
RF	88.383	95.446	93.359	**93.353**	92.861	90.314	**95.626**			

Table 7: Simulated outlier detection performance using the A09 type correlation structure.

Panel A: Independent contamination	Bad Leverage Point			Good leverage point			Vertical outlier		
Outlier Detection Methods:	in T and C	in T	in C	in T and C	in T	in C	in T and C	in T	in C
DDC	0.981	**0.995**	**0.994**	**0.962**	**0.992**	**0.991**	0.612	0.996	0.605
DI	**0.897**	0.895	0.894	0.877	0.866	0.884	0.153	0.232	0.107
MacroPCA	0.923	0.992	0.977	0.910	0.988	0.976	**0.727**	**1.000**	**0.705**
Panel B: Correlated contamination									
DDC	0.979	0.995	0.985	**0.967**	0.987	0.987			
DI	0.897	0.895	0.890	0.870	0.857	0.878			
MacroPCA	**0.980**	**0.997**	**0.995**	0.959	**0.994**	**0.990**			

method best detects outliers. When the data is correlated contaminated, the MacroPCA method is better at finding outliers. The outlier detection ratio is also related to outcomes in the previous tables. When the ratio is low, the ODM-MI algorithms perform worse. This highlights the importance of high-quality outlier detection methods.

To conclude, the DDC method is best at finding outliers when the contamination is independent.

The MacroPCA method is best at finding outliers when the contamination is correlated. It depends on the data structure which method is more suitable. The LR and NN estimators are the most accurate propensity score estimators without the ODM-MI algorithm. There does not stand out an estimator in terms of variance.

In this section, we use the A09 correlation structure. The results for the treatment variable with the ALYZ correlation structure can be found in Appendix D and are comparable to the results in Table (1)-(7). The results from this section are only for the coefficient of the ATT variable. We also had five covariates in this simulation study. The results for these variables are not included in this paper due to space constraints but are available upon request.

6.2 Real Data Results

In this section, we examine the datasets described in Section 5. We begin by applying the ODM-MI algorithm to the data used in Canavire-Bacarreza et al. (2021). Table 8 shows the ATT for LaLonde's and DW's treatment samples in column 1. The comparison groups are PSID and CPS and are given in column 2. Column 3 gives us the experimental ATT presented in the original paper from LaLonde (1986). The 4th column gives us the estimated ATT using only propensity score matching. Finally, columns 5-7 give us the ATT using the ODM-MI algorithm. The first thing that is noticeable from Table

Table 8: Treatment effect estimates of the LaLonde and DW samples using the outlier detection methods and Multiple Imputation.

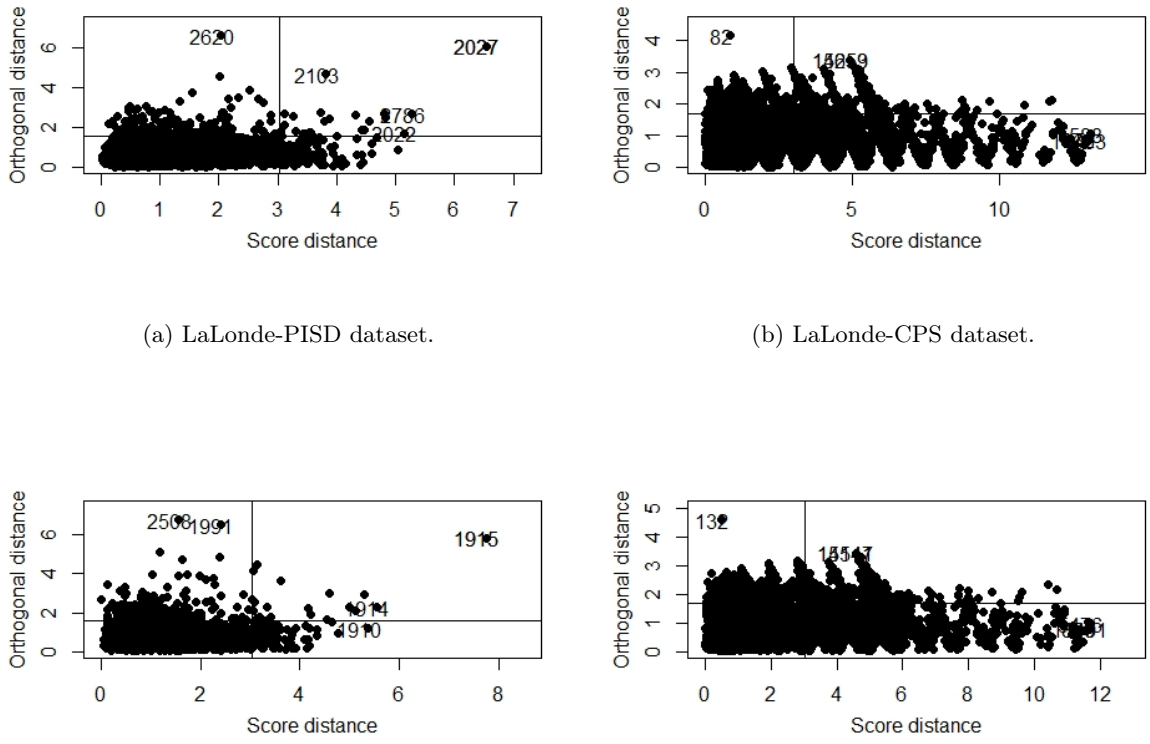
Treatment Group	Comparison Group	Experimental ATT	Estimated ATT	DDC-MI ATT	DI-MI ATT	MacroPCA-MI ATT
LaLonde [297]	PSID [2490]	886	-2857	-1436	-2519	-1545
LaLonde [297]	CPS [15992]	886	-710	-63	-538	-341
Dehejia & Wahba [185]	PSID [2490]	1794	793	693	1732	1013
Dehejia & Wahba [185]	CPS [15992]	1794	803	998	1346	1173

8 is that the estimated ATT without ODM-MI is highly biased. Secondly, when applying the ODM-MI algorithms, a significant improvement is made, especially for the CPS dataset. This suggests that there is a large number of outliers present in the data. The results from the complete regression, including standard errors, for all datasets are given in Appendix D. These tables show that all parameters change in magnitude and, for some outlier detection methods, also in significance when applying the ODM-MI algorithm. This is another reason to suspect that these datasets are contaminated with outliers.

Figure 4 shows us the distance-distance plots for all datasets calculated using the MacroPCA method. The figure reveals the presence of some outliers, which are the observations above the cutoff of the orthogonal distance and score distance. Especially for the CPS control group dataset, the number of outliers is large. This would also suggest why the ODM-MI algorithm significantly improves the results for the datasets.

In summary, our analysis of this dataset suggests that the standard propensity score matching method

is inadequate and that the robust method is preferable. However, it should be noted that the original conclusion drawn from the outcomes in LaLonde (1986), Dehejia and Wahba (1999), and Dehejia and Wahba (2002) were shown to be incorrect, as discussed in Canavire-Bacarreza et al. (2021). These results suggest that the rejection of the original conclusion could have been achieved by using robust propensity score matching instead.



(a) LaLonde-PISD dataset.

(b) LaLonde-CPS dataset.

(c) DW-PISD dataset.

(d) DW-CPS dataset.

Figure 4: Distance-Distance plot for various datasets.

In Table 9, the results for the sensitivity analysis on the LaLonde (1986) dataset are shown. The classical estimators' results are displayed in columns 2 and 3. And the results for the robust estimators are shown in columns 4-9. In columns 2 and 3, the treatment and black variables become insignificant. On top of that, the variables Age and Hispanic change sign. This shows that the classical estimator is sensitive to the smallest amount of contamination. For the robust estimators, the only difference is the significance of the treatment variable for the MacroPCA-MI algorithm. Overall the robust methods are a bit affected in magnitude, but not quite as much as the classical one.

Table 9: Regression results for the robust propensity score estimator in the clean and contaminated case.

	Classic		*DDC-MI*		*DI-MI*		*MacroPCA-MI*	
Dependent variable: RE78	Clean	Contaminated	Clean	Contaminated	Clean	Contaminated	Clean	Contaminated
Constant	3,103.17	2,148.34	13,389.94**	12,310.10**	10,628.57*	10,601.21*	3,583.64	6,207.71
	(3,914.87)	(4,114.95)	(5,296.40)	(5,243.51)	(5,597.14)	(5,732.06)	(4,546.33)	(4,448.48)
Age	−35.34	2.94	−469.09	−304.90	−214.76	−374.31	−147.85	−261.57
	(228.75)	(241.14)	(413.84)	(404.98)	(442.77)	(450.15)	(281.40)	(277.77)
Age squared	1.21	0.68	8.87	5.61	2.99	6.71	3.00	5.05
	(3.81)	(4.04)	(8.42)	(8.23)	(9.12)	(9.30)	(5.18)	(5.14)
Education	283.23	276.81	−64.33	−123.95	−143.72	−314.13	375.73*	251.38
	(197.03)	(204.49)	(185.02)	(189.01)	(195.03)	(192.24)	(213.54)	(210.03)
Black	−1,669.81*	−926.48	−1,885.76***	−2,297.67***	−1,830.76***	−1,880.63***	−1,819.02**	−1,820.14**
	(853.58)	(907.57)	(607.77)	(610.56)	(598.28)	(610.11)	(746.89)	(715.69)
Hispanic	−82.75	190.71	−703.20	−812.53	−928.09	−982.92	−104.03	−869.93
	(1,128.71)	(1,200.96)	(840.44)	(840.83)	(800.77)	(811.55)	(1,034.27)	(988.47)
No degree	−246.46	−105.20	−849.68	−903.40	−681.08	−960.01	310.49	126.39
	(789.20)	(814.56)	(619.73)	(620.18)	(591.87)	(587.42)	(725.91)	(710.40)
RE75	0.19***	0.18***	0.01	0.14	0.12***	0.11***	0.50***	0.60***
	(0.05)	(0.05)	(0.20)	(0.21)	(0.03)	(0.03)	(0.12)	(0.11)
Treatment	953.47*	693.39	212.70	217.13	597.76*	269.11	435.00	806.81*
	(505.87)	(524.37)	(363.00)	(362.88)	(347.22)	(346.47)	(439.68)	(434.90)
Observations	594	594	594	594	594	594	594	594
R^2	0.06	0.04	0.03	0.04	0.06	0.05	0.07	0.08
Adjusted R^2	0.04	0.03	0.02	0.03	0.04	0.03	0.05	0.06
Residual Std. Error (df = 585)	6,160.40	6,385.81	4,418.17	4,417.46	4,218.19	4,215.49	5,347.99	5,292.61
F Statistic (df = 8; 585)	4.38***	2.95***	2.30**	3.07***	4.32***	3.67***	5.12***	6.01***

Note: *p<0.1; **p<0.05; ***p<0.01

7 Conclusion

This paper aimed to examine the influence of outliers on propensity score matching and provide a robust method. To solve this question, we performed a simulation study. We used five different propensity score estimators, used two sorts of contamination types: independent and correlated, used three different kinds of outliers: good leverage points, bad leverage points, and vertical outliers, and let them occur in three different places: in the treatment group, in the control group and both, and use two correlation structures for the variables. For the robust method, we used DDC, DI, and MacroPCA to find the outliers and impute them using Multiple Imputation. After applying the robust method, the matching procedure, and the estimation of the treatment effect using Ordinary Least Squares, the performance of the methods was evaluated based on the bias, variance, and coverage of the parameters.

The simulation study showed that the outliers affected the propensity score matching estimators. Especially the bad leverage points biased the estimates. This is because these kinds of outliers completely distort the distribution used to find good counterfactuals and create suitable matches. Vertical outliers also show this property when they are located in the treatment group. When located in the control group, the outlying values are not used for matching, thus not influencing the bias. Overall, the neural network outperforms the other estimators regarding the performance measures. These differences mitigate when the ODM-MI algorithm is applied. The estimators now perform almost equally. The DDC and MacroPCA methods outperform the DI method. These methods significantly reduce the influence of the outliers to

the point where the bias is nearly removed. The DDC performs better in the independent contamination scenario, and the MacroPCA outperforms the other methods in the correlated contamination scenario. The correlation structures of the variables used in the simulation do not influence the simulation study results.

In addition to our simulation study, our methodology was also applied to real datasets. When applying our robust propensity score matching methods to the datasets used in Canavire-Bacarreza et al. (2021), we found that the data was contaminated with outliers. The robust estimators' usage moved the treatment effect's prediction to the true experimental value. Remarkably, the DI method outperforms the DDC and MacroPCA methods using this dataset. This puts even more emphasis on the importance of the used dataset when performing a robustness analysis. When conducting sensitivity analysis on the LaLonde (1986) dataset, the results show that the classical estimator is sensitive to even the smallest amount of contamination. The robust estimators are affected but return the same significance for all but one variable.

The present paper has some limitations, particularly in the simulation study. Although we used sufficient bootstrap replications and simulation runs, increasing them could lead to more accurate results. Furthermore, we did not explore high-dimensional datasets, which are becoming the standard with the current data availability.

The outliers contamination scenarios we used in this paper were relatively simple. An interesting direction would be examining whether the robust counterpart would also capture complex/realistic contamination scenarios.

References

Abadie, A. and Imbens, G. W. (2006). Large sample properties of matching estimators for average treatment effects. *Econometrica*, 74(1):235–267.

Agostinelli, C., Leung, A., Yohai, V. J., and Zamar, R. H. (2015). Robust estimation of multivariate location and scatter in the presence of cellwise and casewise contamination. *Test*, 24(3):441–461.

Alqallaf, F., Van Aelst, S., Yohai, V. J., and Zamar, R. H. (2009). Propagation of outliers in multivariate data. *The Annals of Statistics*, pages 311–331.

Angrist, J. D. and Krueger, A. B. (1991). Does compulsory school attendance affect schooling and earnings? *The Quarterly Journal of Economics*, 106(4):979–1014.

Angrist, J. D. and Krueger, A. B. (2001). Instrumental variables and the search for identification: From supply and demand to natural experiments. *Journal of Economic Perspectives*, 15(4):69–85.

Austin, P. C. (2014). A comparison of 12 algorithms for matching on the propensity score. *Statistics in medicine*, 33(6):1057–1069.

Austin, P. C. and Fine, J. P. (2019). Propensity-score matching with competing risks in survival analysis. *Statistics in medicine*, 38(5):751–777.

Barnard, J. and Rubin, D. B. (1999). Small-sample degrees of freedom with multiple imputation. *Biometrika*, 86(4):948–955.

Busso, M., DiNardo, J., and McCrary, J. (2009). Finite sample properties of semiparametric estimators of average treatment effects. *forthcoming in the Journal of Business and Economic Statistics*.

Busso, M., DiNardo, J., and McCrary, J. (2014). New evidence on the finite sample properties of propensity score reweighting and matching estimators. *Review of Economics and Statistics*, 96(5):885–897.

Canavire-Bacarreza, G., Castro Peñarrieta, L., and Ugarte Ontiveros, D. (2021). Outliers in semiparametric estimation of treatment effects. *Econometrics*, 9(2):19.

Cochran, W. G. (1939). The use of the analysis of variance in enumeration by sampling. *Journal of the American Statistical Association*, 34(207):492–510.

Dehejia, R. (2005). Practical propensity score matching: a reply to smith and todd. *Journal of econometrics*, 125(1-2):355–364.

Dehejia, R. H. and Wahba, S. (1999). Causal effects in nonexperimental studies: Reevaluating the evaluation of training programs. *Journal of the American statistical Association*, 94(448):1053–1062.

Dehejia, R. H. and Wahba, S. (2002). Propensity score-matching methods for nonexperimental causal studies. *Review of Economics and statistics*, 84(1):151–161.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *The Annals Of Statistics*, 32(2):407–499.

Finney, S. J. and DiStefano, C. (2006). Non-normal and categorical data in structural equation modeling. *Structural equation modeling: A second course*, 10(6):269–314.

Gharibzadeh, S., Mansournia, M. A., Rahimiforoushani, A., Alizadeh, A., Amouzegar, A., Mehrabani-Zeinabad, K., and Mohammad, K. (2018). Comparing different propensity score estimation methods for estimating the marginal causal effect through standardization to propensity scores. *Communications in Statistics-Simulation and Computation*, 47(4):964–976.

Gulrez, T. and Al-Odienat, A. (2015). A new perspective on principal component analysis using inverse covariance. *International Arab Journal of Information Technology (IAJIT)*, 12(1).

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Honaker, J. and King, G. (2010). What to do about missing values in time-series cross-section data. *American journal of political science*, 54(2):561–581.

Honaker, J., King, G., and Blackwell, M. (2011). Amelia ii: A program for missing data. *Journal of statistical software*, 45:1–47.

Hubert, M., Rousseeuw, P. J., and Van den Bossche, W. (2019). Macropca: An all-in-one pca method allowing for missing values as well as cellwise and rowwise outliers. *Technometrics*, 61(4):459–473.

Hubert, M., Rousseeuw, P. J., and Vanden Branden, K. (2005). Robpca: a new approach to robust principal component analysis. *Technometrics*, 47(1):64–79.

Hubert, M., Rousseeuw, P. J., and Verdonck, T. (2012). A deterministic algorithm for robust location and scatter. *Journal of Computational and Graphical Statistics*, 21(3):618–637.

Imbens, G. W. (2000). The role of the propensity score in estimating dose-response functions. *Biometrika*, 87(3):706–710.

Imbens, G. W. (2004). Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and statistics*, 86(1):4–29.

Imbens, G. W. and Rubin, D. B. (1997). Estimating outcome distributions for compliers in instrumental variables models. *The Review of Economic Studies*, 64(4):555–574.

Imbens, G. W. and Rubin, D. B. (2016). *Causal inference for statistics, social, and biomedical sciences: An introduction.* Taylor & Francis.

Kallus, N. and Zhou, A. (2018). Policy evaluation and optimization with continuous treatments. In *International conference on artificial intelligence and statistics*, pages 1243–1251. PMLR.

LaLonde, R. J. (1986). Evaluating the econometric evaluations of training programs with experimental data. *The American economic review*, pages 604–620.

Leung, A., Yohai, V., and Zamar, R. (2017). Multivariate location and scatter matrix estimation under cellwise and casewise contamination. *Computational Statistics & Data Analysis*, 111:59–76.

Maronna, R., Martin, R. D., Yohai, V., and Salibián-Barrera, M. (2006). Robust statistics: Theory and practice.

Marshall, A., Altman, D. G., Holder, R. L., and Royston, P. (2009). Combining estimates of interest in prognostic modelling studies after multiple imputation: Current practice and guidelines. *BMC Medical Research Methodology*, 9:57.

McCaffrey, D. F., Ridgeway, G., and Morral, A. R. (2004). Propensity score estimation with boosted regression for evaluating causal effects in observational studies. *Psychological methods*, 9(4):403.

Newgard, C. D. and Lewis, R. J. (2015). Missing data: how to best account for what is not known. *Jama*, 314(9):940–941.

Neyman, J. (1923). On the application of probability theory to agricultural experiments. essay on principles. *Ann. Agricultural Sciences*, pages 1–51.

Penning, de Vries, B. B., van Smeden, M., and Groenwold, R. H. (2018). Propensity score estimation using classification and regression trees in the presence of missing covariate data. *Epidemiologic Methods*, 7(1):20170020.

Raymaekers, J. and Rousseeuw, P. J. (2019). Handling cellwise outliers by sparse regression and robust covariance. *arXiv preprint arXiv:1912.12446*.

Raymaekers, J. and Rousseeuw, P. J. (2023). Challenges of cellwise outliers. *arXiv preprint arXiv:2302.02156*.

Rosenbaum, P. R. (1984). The consequences of adjustment for a concomitant variable that has been affected by the treatment. *Journal of the Royal Statistical Society: Series A (General)*, 147(5):656–666.

Rosenbaum, P. R. and Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55.

Rousseeuw, P. and Leroy, A. (1987). Robust regression and outlier detection. new york: John wiley& sons.

Rousseeuw, P. J. and Van den Bossche, W. (2018). Detecting deviating data cells. *Technometrics*, 60(2):135–145.

RStudio Team (2020). *RStudio: Integrated Development Environment for R*. RStudio, PBC., Boston, MA.

Rubin, D. B. (1973). Matching to remove bias in observational studies. *Biometrics*, pages 159–183.

Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3):581–592.

Rubin, D. B. (1977). Assignment to treatment group on the basis of a covariate. *Journal of Educational Statistics*, 2(1):1–26.

Rubin, D. B. (1985). Matching to remove bias in observational studies. *Biometrics*, 41(1):159–183.

Rubin, D. B. (2001). Using propensity scores to help design observational studies: Application to the tobacco litigation. *Health Services and Outcomes Research Methodology*, 2(3-4):169–188.

Schafer, J. L. (1997). *Analysis of incomplete multivariate data*. CRC press.

Schafer, J. L. and Olsen, M. K. (1998). Multiple imputation for multivariate missing-data problems: A data analyst's perspective. *Multivariate behavioral research*, 33(4):545–571.

Serneels, S. and Verdonck, T. (2008). Principal component analysis for data containing outliers and missing elements. *Computational Statistics & Data Analysis*, 52(3):1712–1727.

Setoguchi, S., Schneeweiss, S., Brookhart, M. A., Glynn, R. J., and Cook, E. F. (2008). Evaluating uses of data mining techniques in propensity score estimation: a simulation study. *Pharmacoepidemiology and drug safety*, 17(6):546–555.

Smith, J. A. and Todd, P. E. (2005). Does matching overcome lalonde's critique of nonexperimental estimators? *Journal of econometrics*, 125(1-2):305–353.

Sterne, J. A., White, I. R., Carlin, J. B., Spratt, M., Royston, P., Kenward, M. G., Wood, A. M., and Carpenter, J. R. (2009). Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. *Bmj*, 338.

Stuart, E. A. (2010). Matching methods for causal inference: A review and a look forward. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 25(1):1.

Van Buuren, S. (2018). *Flexible imputation of missing data*. CRC press.

Yeo, I.-K. and Johnson, R. A. (2000). A new family of power transformations to improve normality or symmetry. *Biometrika*, 87(4):954–959.

Zhao, P., Su, X., Ge, T., and Fan, J. (2016). Propensity score and proximity matching using random forest. *Contemporary clinical trials*, 47:85–92.

Appendix A

Algorithm for calculating propensity score and proximity matrix based on random forest.

Suppose that the total sample size, including treated and control subjects, is n and that the total number of trees in the random forest is B. Let $(S_1, \ldots, S_n)$ denote the propensity score vector.

Initialize: set $S_i = 0$ for $i = 1, \ldots, n$.

For $b = 1, \ldots, B$, do

- Using all data, grow a tree with treatment as output and all other covariates as inputs. At each split, search over m randomly selected inputs. No pruning.
- Propensity score calculation:
  - Compute the percentage of treated observations s_i for the terminal node to which the ith subject belongs.
  - Update: $S_i = S_i + s_i$

End do.

Average: $S_i = S_i/B$

Appendix B

For the location, we use Tukey's biweight function given by:

$$W(t) = \left\{1 - \left(\frac{t}{c}\right)^2\right\}^2 I(|t| \leq l), \tag{51}$$

where l=3 is a tuning constant. Given a univariate dataset $X = \{x_1, ..., x_n\}$, we start from the initial estimates for scale and location, m_1 and s_1 respectively:

$$m_1 = \underset{i=1}{\overset{n}{\text{med}}}(x_i), \tag{52}$$

$$s_1 = \underset{i=1}{\overset{n}{\text{med}}}|x_i - m_1|, \tag{53}$$

where the function med(X) is the median of X. Subsequently, we calculate the robust location estimate as follows:

$$\text{robLoc}(X) = (\sum_{i=1}^{n} w_i x_i)/(\sum_{i=1}^{n} w_i), \tag{54}$$

where the weights are given by $w_i = W((x_i - m_1)/s_1)$.

To estimate the robust scale, we assume that X has already been centered by subtracting robLoc(X), which means we only need to focus on the deviations from zero. Starting from the initial estimate $s_2 = \text{med}_i(|x_i|)$, we then compute the scale estimate

$$s = \text{robScale}(X) = s_2 \sqrt{\frac{1}{\delta} \underset{i=1}{\overset{n}{\text{ave}}} \rho\left(\frac{y_i}{s_2}\right)}, \tag{55}$$

where the function ave(X) stands for the average of X and the constant δ=0.845 ensures consistency for Gaussian data. The function $\rho(t)$ is defined as: $\rho(t) = \min(t^2, b^2)$, where b=2.5.

Appendix C

A robust correlation measure for bivariate relations, where the computation is performed for all i in which neither u_{ij} nor u_{ih} is NA.

The subsequent methods are bivariate, meaning they involve two data columns, referred to as j and h. To compute the correlation, we begin with an initial estimate:

$$\hat{\rho}_{jh} = ((robScale_i(z_{ij} + z_{ih}))^2 - (robScale_i(z_{ij} - z_{ih}))^2)/4, \tag{56}$$

where the estimate is limited to a range between -1 and 1. The estimated correlation coefficient ρ^{jh} represents a tolerance ellipse around the point (0,0) with the same coverage probability p as described in equation 14. Then robCorr is defined as the basic product-moment correlation of the data points (z_{ij}, z_{ih}) that fall within the ellipse.

Regarding the slope, we assume that the columns have already been centered, but they don't need to be normalized. The initial estimate for the slope is:

$$b_{jh} = med_{i=1}^{n}(\frac{z_{ij}}{z_{ih}}), \tag{57}$$

where any fractions with a zero denominator are removed before proceeding, for each value of i, we calculate the raw residual r_{ijh} as the difference between z_{ij} and the product of b_{jh} and z_{ih}.

Lastly, we compute the ordinary least-squares regression line without an intercept on the data points for which the absolute value of the raw residual (r_{ijh}) is less than or equal to c times the robust scale, where c is a constant described in equation 14. We then define robSlope as the slope of the regression equation.

$$|r_{ijh}| \leq c \times robScale_{i'}(r_{i'jh}). \tag{58}$$

Appendix D

Table 10: Regression results from the LaLonde-PSID dataset.

	Intercept	Age	Age^2	Educ	Black	Hispan	Nodegree	RE75	Treat
Clean	-7524.05	855.73**	-14.63***	360.79	279.56	1979.44	-242.55	0.27***	-2856.80***
	5751.10	342.40	5.54	266.80	970.59	1480.10	1066.52	0.07	774.40
DDC	4192.17	202.18	-4.66	-20.72	-519.80	1181.29	-1063.51	0.67***	-1436.07
	4073.72	231.40	3.74	195.17	647.46	975.68	744.48	0.07	586.30
DI	-1992.20	253.28	-4.72	547.72**	609.53	2549.97**	-656.72	0.37***	-2518.90
	5098.75	298.04	4.90	268.16	778.42	1160.78	908.71	0.06	614.78
MacroPCA	-1465.79	536.97**	-9.73	73.67	-202.60	1188.01	-828.04	0.55***	-1545.20
	4052.24	248.25	4.03	202.42	748.64	1056.90	755.69	0.06	556.96

Table 11: Regression results from the LaLonde-CPS dataset.

	Intercept	Age	Age^2	Educ	Black	Hispan	Nodegree	RE75	Treat
Clean	-4011.88	319.14	-4.75	351.05*	625.91	2139.84*	616.28	0.37***	-710.35
	4531.46	274.55	4.71	208.02	915.94	1247.64	854.37	0.06	550.55
DDC	8116.41*	-237.47	3.01	-37.99	-966.93	-702.87	-272.89	0.80***	-63.42
	3860.54	184.86	3.03	214.90	900.19	1161.04	730.89	0.05	387.83
DI	4074.77	-31.97	0.45	240.59	-784.17	-342.97	-404.32	0.39***	-538.36
	4537.89	251.65	4.16	221.66	930.34	1220.60	919.14	0.07	556.72
MacroPCA	6156.08*	-106.77	1.10	102.52	-1194.32	-815.57	-200.79	0.61***	-340.78
	3605.44	219.23	3.74	217.83	883.76	1325.67	769.88	0.06	480.74

Table 12: Regression results from the Dehejia and Wahba-PSID dataset.

	Intercept	Age	Age^2	Educ	Black	Hispan	Nodegree	RE74	RE75	Treat
Clean	-9766.25	616.28	-10.28*	447.97*	1428.77	2323.64	-79.09	0.24**	0.24*	792.94
	6089.55	376.86	6.13	257.90	1018.96	1761.77	1106.29	0.09	0.12	809.17
DDC	2782.07	67.06	-1.94	76.86	-375.88	309.24	-561.44	0.18	0.57***	692.79
	4682.65	302.75	5.03	231.59	741.45	1342.11	887.79	0.11	0.10	612.36
DI	4339.99	-90.87	0.50	109.43	-795.00	-431.51	-655.21	0.91***	-0.12	1732.24***
	4936.64	303.20	4.87	199.98	838.19	1362.53	863.74	0.07	0.09	612.74
MacroPCA	-285.85	227.81**	-4.29	140.27**	-84.57	426.32	-856.47	0.13	0.55***	1013.33
	4465.36	270.53	4.40	204.58	861.60	1429.52	860.68	0.09	0.12	652.27

Table 13: Regression results from the Dehejia and Wahba-CPS dataset.

	Intercept	Age	Age^2	Educ	Black	Hispan	Nodegree	RE74	RE75	Treat
Clean	-2649.52	263.66	-4.42	475.47*	-1851.08	-658.23	57.53	0.02	0.48***	803.36*
	5460.49	328.39	5.44	265.31	1240.11	2011.78	1158.51	0.11	0.15	437.31
DDC	7852.44**	-307.79	4.11	90.58	-1556.77	-1260.71	-35.09	0.20*	0.56***	998.01*
	3954.29	218.70	3.83	246.31	775.85	1238.13	966.79	0.11	0.14	528.94
DI	1259.85	-209.75	2.63	621.22	-1314.18	671.76	15.14	0.38***	0.30**	1345.53*
	7149.47	291.21	4.72	380.50	1036.59	1558.55	1109.80	0.11	0.14	698.32
MacroPCA	4602.95	-251.48	3.17	343.82	-985.45	-243.30	46.93	0.02	0.64***	1173.27**
	4141.72	250.15	4.20	240.46	951.57	1792.75	931.55	0.13	0.15	566.12

.

Appendix E

Table 14: Simulated bias of Average Treatment Effect on the Treated using the ALYZ type correlation structure.

Panel A: Independent contamination		Bad Leverage Point			Good leverage point			Vertical outlier		
Estimators:	Clean	in T and C	in T	in C	in T and C	in T	in C	in T and C	in T	in C
LR	**0.001**	**-0.188**	-0.686	0.284	-0.003	-0.007	0.004	**1.046**	1.028	**0.022**
CART	0.004	-0.203	-0.692	0.278	0.003	-0.015	0.007	1.070	**1.015**	0.055
NN	0.007	-0.207	-0.713	0.290	-0.003	-0.006	**0.003**	1.054	1.023	0.057
GBM	-0.001	-0.216	**-0.680**	**0.251**	**-0.001**	-0.003	-0.003	1.050	1.031	0.032
RF	0.004	-0.276	-0.737	0.282	-0.002	**-0.002**	0.005	1.070	1.018	0.051
Panel B: Correlated contamination										
LR	**0.001**	0.030	**-0.454**	0.302	**-0.008**	-0.005	**-0.001**			
CART	0.004	**-0.004**	-0.455	0.320	-0.010	-0.004	0.006			
NN	0.007	-0.006	-0.485	0.335	-0.014	-0.004	0.004			
GBM	-0.001	-0.045	-0.466	**0.279**	-0.016	-0.004	-0.002			
RF	0.004	-0.041	-0.506	0.309	-0.009	**0.002**	0.004			

Table 15: Simulated variance of Average Treatment Effect on the Treated using the ALYZ type correlation structure.

Panel A: Independent contamination		Bad Leverage Point			Good leverage point			Vertical outlier		
Estimators:	Clean	in T and C	in T	in C	in T and C	in T	in C	in T and C	in T	in C
LR	**0.096**	**0.348**	**0.270**	0.263	**0.105**	**0.109**	**0.096**	**0.310**	**0.227**	**0.233**
CART	0.098	0.362	0.291	0.266	0.108	0.112	0.097	0.313	0.228	0.237
NN	0.098	0.362	0.298	0.268	0.108	0.112	0.097	0.313	0.229	0.239
GBM	0.097	0.354	0.274	**0.262**	0.106	0.109	0.097	0.311	0.228	0.235
RF	0.097	0.368	0.292	0.268	0.108	0.110	0.097	0.312	0.227	0.236
Panel B: Correlated contamination										
LR	**0.096**	**0.300**	**0.238**	**0.235**	**0.104**	**0.108**	**0.096**			
CART	0.098	0.313	0.258	0.239	0.106	0.110	0.097			
NN	0.098	0.313	0.265	0.243	0.106	0.111	0.097			
GBM	0.097	0.305	0.242	0.237	0.105	0.108	0.096			
RF	0.097	0.315	0.259	0.243	0.106	0.109	0.096			

Table 16: Simulated coverage of Average Treatment Effect on the Treated using the ALYZ type correlation structure.

Panel A: Independent contamination		Bad Leverage Point			Good leverage point			Vertical outlier		
Estimators:	Clean	in T and C	in T	in C	in T and C	in T	in C	in T and C	in T	in C
LR	91.365	89.087	32.586	81.808	87.983	83.537	86.770	5.002	**5.239**	**100.000**
CART	89.480	88.586	**35.825**	83.613	88.357	83.151	**90.971**	4.302	2.094	**100.000**
NN	**95.149**	**89.480**	33.382	**84.986**	**88.942**	**86.285**	90.051	**5.137**	2.603	**100.000**
GBM	89.487	87.289	33.969	84.232	86.785	83.514	89.926	4.966	3.318	**100.000**
RF	94.169	86.778	30.257	82.816	88.212	82.410	89.780	3.362	1.665	**100.000**
Panel B: Correlated contamination										
LR	91.365	**93.580**	50.892	79.599	91.673	90.355	88.184			
CART	89.480	92.932	**53.497**	77.319	91.199	87.338	90.378			
NN	**95.149**	92.309	51.291	75.591	90.647	86.416	**90.604**			
GBM	89.487	92.413	50.006	**79.612**	91.204	89.213	90.268			
RF	94.169	90.714	50.448	77.845	**92.425**	**91.596**	89.532			

Table 17: Simulated bias of Average Treatment Effect on the Treated using the ALYZ type correlation structure after applying the outlier detection methods and the Multiple Imputation.

Panel A: Independent contamination		Bad Leverage Point			Good leverage point			Vertical outlier		
Estimators:	Clean	in T and C	in T	in C	in T and C	in T	in C	in T and C	in T	in C
DDC										
LR	-0.031	**-0.108**	-0.038	**-0.018**	-0.062	-0.018	**-0.041**	**0.034**	0.092	-0.020
CART	-0.030	-0.113	-0.037	-0.023	**-0.059**	-0.019	-0.045	0.047	0.091	**-0.008**
NN	**-0.030**	-0.113	**-0.034**	-0.018	-0.064	**-0.011**	-0.044	0.040	0.095	-0.013
GBM	-0.032	-0.113	-0.038	-0.023	-0.060	-0.015	-0.047	0.038	**0.090**	-0.017
RF	-0.030	-0.117	-0.036	-0.019	-0.073	-0.024	-0.044	0.047	0.097	-0.013
DI										
LR	-0.003	-0.081	-0.080	**-0.067**	-0.090	-0.030	**-0.135**	-0.040	-0.036	-0.102
CART	-0.004	-0.078	-0.076	-0.072	**-0.010**	-0.024	-0.139	-0.037	-0.028	-0.101
NN	0.004	**-0.071**	**-0.070**	-0.067	-0.050	**-0.018**	-0.138	**-0.034**	**-0.020**	**-0.095**
GBM	**-0.001**	-0.078	-0.080	-0.077	-0.040	-0.027	-0.137	-0.039	-0.034	-0.103
RF	-0.004	-0.081	-0.080	-0.076	-0.027	-0.047	-0.147	-0.037	-0.032	-0.107
MacroPCA										
LR	-0.003	**-0.127**	**-0.071**	-0.031	-0.016	-0.018	-0.042	**0.073**	**0.030**	-0.037
CART	-0.004	-0.146	-0.074	-0.033	**-0.014**	-0.013	**-0.042**	0.102	0.032	**-0.019**
NN	0.003	-0.140	-0.072	**-0.025**	-0.060	**-0.011**	-0.043	0.102	0.043	-0.020
GBM	**-0.000**	-0.143	-0.073	-0.033	-0.018	-0.020	-0.046	0.090	0.034	-0.031
RF	-0.006	-0.168	-0.085	-0.033	-0.028	-0.027	-0.054	0.091	0.032	-0.035
Panel A: Correlated contamination										
DDC										
LR	-0.031	**-0.051**	-0.020	-0.007	-0.099	-0.142	**-0.041**			
CART	-0.030	-0.057	-0.015	-0.008	-0.104	**-0.133**	-0.048			
NN	-0.030	-0.056	-0.014	-0.009	-0.100	0.133	-0.050			
GBM	-0.032	-0.057	-0.018	-0.010	**-0.098**	-0.140	-0.051			
RF	**-0.030**	-0.052	**-0.004**	**-0.005**	-0.100	-0.133	-0.048			
DI										
LR	-0.003	-0.078	-0.079	-0.074	**-0.048**	-0.032	-0.123			
CART	-0.004	-0.074	-0.082	-0.074	-0.052	-0.032	**-0.122**			
NN	0.004	**-0.072**	**-0.071**	**-0.064**	-0.048	**-0.023**	-0.116			
GBM	**-0.001**	-0.080	-0.081	-0.073	-0.055	-0.033	-0.125			
RF	-0.004	-0.082	-0.083	-0.074	-0.073	-0.047	-0.125			
MacroPCA										
LR	-0.003	**-0.107**	**-0.045**	-0.029	-0.171	-0.186	-0.035			
CART	-0.004	-0.118	-0.055	-0.021	**-0.163**	-0.180	-0.035			
NN	0.003	-0.108	-0.051	**-0.016**	-0.164	**-0.177**	**-0.031**			
GBM	**-0.000**	-0.110	-0.053	-0.027	-0.171	-0.184	-0.037			
RF	-0.006	-0.135	-0.060	-0.022	-0.182	-0.192	-0.037			

Table 18: Simulated variance of Average Treatment Effect on the Treated using the ALYZ type correlation structure after applying the outlier detection methods and the Multiple Imputation.

Panel A: Independent contamination		Bad Leverage Point			Good leverage point			Vertical outlier		
Estimators:	Clean	in T and C	in T	in C	in T and C	in T	in C	in T and C	in T	in C
DDC										
LR	0.010	0.040	0.019	0.016	0.026	0.020	0.018	0.052	0.016	0.029
CART	0.011	0.043	0.021	0.018	0.028	0.023	0.019	0.055	0.018	0.030
NN	0.011	0.042	0.021	0.017	0.027	0.022	0.018	0.053	0.017	0.030
GBM	0.010	0.040	0.019	0.017	0.027	0.020	0.018	0.053	0.017	0.030
RF	0.011	0.042	0.020	0.017	0.028	0.022	0.019	0.053	0.017	0.030
DI										
LR	0.009	0.014	0.013	0.011	0.041	0.019	0.020	0.026	0.012	0.021
CART	0.010	0.016	0.014	0.013	0.044	0.021	0.023	0.027	0.013	0.021
NN	0.010	0.015	0.013	0.012	0.042	0.020	0.021	0.027	0.013	0.021
GBM	0.010	0.015	0.013	0.012	0.041	0.019	0.021	0.026	0.012	0.021
RF	0.010	0.015	0.013	0.012	0.043	0.020	0.021	0.027	0.013	0.021
MacroPCA										
LR	0.009	0.039	0.021	0.013	0.019	0.017	0.014	0.043	0.014	0.026
CART	0.010	0.046	0.024	0.015	0.021	0.019	0.016	0.045	0.014	0.027
NN	0.010	0.044	0.024	0.014	0.020	0.019	0.014	0.044	0.014	0.027
GBM	0.010	0.041	0.022	0.014	0.020	0.018	0.015	0.043	0.014	0.026
RF	0.010	0.043	0.024	0.014	0.020	0.019	0.015	0.045	0.015	0.027
Panel A: Correlated contamination										
DDC										
LR	0.010	0.040	0.018	0.017	0.022	0.019	0.016			
CART	0.011	0.045	0.021	0.018	0.025	0.023	0.018			
NN	0.011	0.042	0.019	0.017	0.025	0.021	0.017			
GBM	0.010	0.041	0.019	0.017	0.023	0.020	0.017			
RF	0.011	0.042	0.020	0.017	0.024	0.021	0.017			
DI										
LR	0.009	0.015	0.013	0.011	0.032	0.017	0.018			
CART	0.010	0.016	0.014	0.013	0.036	0.020	0.019			
NN	0.010	0.015	0.013	0.012	0.035	0.019	0.018			
GBM	0.010	0.015	0.013	0.012	0.033	0.018	0.018			
RF	0.010	0.016	0.013	0.012	0.034	0.018	0.018			
MacroPCA										
LR	0.009	0.039	0.020	0.012	0.016	0.015	0.013			
CART	0.010	0.045	0.025	0.014	0.018	0.018	0.014			
NN	0.010	0.042	0.024	0.013	0.018	0.017	0.013			
GBM	0.010	0.040	0.021	0.013	0.017	0.016	0.013			
RF	0.010	0.044	0.024	0.013	0.018	0.017	0.013			

Table 19: Simulated coverage of Average Treatment Effect on the Treated using the ALYZ type correlation structure after applying the outlier detection methods and the Multiple Imputation.

Panel A: Independent contamination		Bad Leverage Point			Good leverage point			Vertical outlier		
Estimators:	Clean	in T and C	in T	in C	in T and C	in T	in C	in T and C	in T	in C
DDC										
LR	85.751	85.094	76.649	90.121	84.916	81.992	95.452	88.494	81.831	97.485
CART	85.465	**85.673**	79.112	90.254	86.710	85.548	95.879	**89.002**	81.130	**98.636**
NN	**87.504**	84.445	**80.085**	**93.077**	86.359	**87.187**	**96.780**	87.999	85.128	97.405
GBM	87.001	83.915	77.425	90.354	**87.144**	83.351	93.969	87.391	83.789	97.635
RF	86.393	84.951	79.437	90.380	81.749	83.427	93.823	87.411	**85.365**	95.435
DI										
LR	91.801	87.508	80.948	**87.655**	70.169	80.070	**88.157**	93.521	85.260	86.716
CART	91.643	**89.101**	**85.400**	86.177	70.430	80.063	87.428	94.749	86.577	**90.717**
NN	92.113	87.960	84.195	86.071	71.066	**80.449**	87.296	94.267	87.053	87.457
GBM	90.666	88.516	83.419	85.024	72.539	80.176	84.347	93.364	**89.177**	88.854
RF	**92.541**	88.461	84.373	84.785	**79.808**	80.404	84.511	**95.335**	85.338	86.334
MacroPCA										
LR	91.596	**91.315**	73.495	90.793	80.631	**89.464**	94.049	88.615	82.485	94.566
CART	91.376	88.891	73.341	91.193	85.619	82.120	94.981	88.477	82.987	94.176
NN	**91.779**	88.323	72.601	**91.866**	87.563	87.733	**96.785**	87.035	**85.046**	94.289
GBM	90.042	88.647	**77.226**	90.341	82.249	83.380	92.947	87.993	84.790	**94.704**
RF	92.715	86.039	71.518	89.880	**88.854**	87.961	94.075	**89.801**	84.476	94.051
Panel A: Correlated contamination										
DDC										
LR	85.751	**93.672**	93.642	91.257	72.684	84.968	92.017			
CART	85.465	93.101	93.798	93.583	**76.630**	88.903	**95.183**			
NN	**87.504**	91.902	94.401	94.048	71.708	84.689	93.968			
GBM	87.001	91.099	**95.275**	**94.167**	73.673	86.630	89.453			
RF	86.393	91.754	95.146	93.127	71.655	**89.535**	93.191			
DI										
LR	91.442	86.266	81.016	81.719	80.775	84.199	82.870			
CART	91.277	**88.791**	84.371	84.710	87.218	86.906	**84.640**			
NN	92.186	87.682	**84.396**	84.826	87.848	**89.190**	83.073			
GBM	90.242	85.408	81.108	**86.182**	83.800	85.759	83.159			
RF	**92.532**	87.427	84.357	84.401	**89.615**	84.789	80.505			
MacroPCA										
LR	91.596	89.445	84.574	89.258	74.457	80.721	90.279			
CART	91.376	90.162	83.395	91.507	**79.210**	80.539	**91.687**			
NN	91.779	**91.818**	85.831	**92.829**	78.798	82.744	90.816			
GBM	90.042	88.500	84.151	89.110	75.303	**89.052**	89.616			
RF	**92.715**	87.527	**86.106**	91.604	73.024	83.731	88.585			

Table 20: ALYZ independent contamination outlier detection

Panel A: Independent contamination	Bad Leverage Point			Good leverage point			Vertical outlier		
Outlier Detection Methods:	in T and C	in T	in C	in T and C	in T	in C	in T and C	in T	in C
DDC	**0.878**	**0.932**	0.889	**0.887**	0.771	0.719	0.831	0.896	**0.893**
DI	0.805	0.850	**0.893**	0.766	0.815	0.803	0.638	0.708	0.781
MacroPCA	0.804	0.746	0.753	0.856	**0.824**	**0.860**	**0.873**	**0.942**	0.838
Panel B: Correlated contamination									
DDC	0.870	0.830	0.780	**0.785**	0.768	**0.801**			
DI	**0.904**	**0.944**	**0.892**	0.751	**0.810**	0.799			
MacroPCA	0.803	0.735	0.749	0.748	0.617	0.744			