A Comparative Analysis of Machine Learning Outlier Detection Methods for Robust Causal Inference

Adrian Enders

September 15, 2025

Abstract

We implement and compare novel outlier detection methods to remove outliers and see how they affect causal impact estimates. This study evaluates these models' accuracy in outlier detection and their impact on treatment effect estimation with simulated and real-world data. We highlight outlier solutions that minimize removing observations, while also preserving accuracy in causal impact analysis. Specifically, we find that ensemble approaches of outlier detection methods are the most effective at obtaining accurate treatment effect estimates, while also minimizing the removal of observations in high-dimensional data.

1 Introduction

1.1 Motivation

Outliers distort prediction models, but their impact extends to causal estimations as well. This vulnerability is especially pronounced in modern causal inference methods, such as those used at Amazon, because they use machine learning models that capture non-linear relationships. Our analysis indicates that these non-linear estimation methods can be significantly influenced by outliers. Additionally, with the prevalence of extreme events like COVID and tariffs, anomalies in datasets are becoming more common. Additionally, we examine how these models affect our treatment effect estimates.

This challenge becomes particularly difficult when using large datasets as it can be hard to spot outliers within a large feature space. In most cases, economists want a quick solution that can remove outliers without having to spend much time finetuning a model. The traditional approach is to trim observations univariately, column by column, at the 95^{th} percentile to remove outliers. However, this approach doesn't consider the multiple relationships between features and may trim too much data. For example, we might see that a customer may only have one feature that is abnormal despite having other normal characteristics. Additionally, if the data contains a cluster of customers with high consumption volume, they may get trimmed if the data contains mainly low-volume customers. In such cases, the right outlier method would identify the right data clusters for our data. Thus, we explore machine learning methods that handle outliers efficiently and consider all features jointly.

1.2 Literature Review

For our analysis we tested models found in Table 1 from Zhao et al. [1], the most comprehensive collection of outlier detection models. These models have different mechanisms for outlier detection,

but they are nonparametric and unsupervised learning models. This means the models learn outliers without using pre-defined outlier patterns in the data. The results will focus mainly on these kinds of models as in most cases we have no known patterns for our outliers. It is important to note that for the treatment effect estimates that these models remove observations from the treatment and control. This is deliberately done to mitigate either group from having outliers cause biased results.

This methodological choice stands in contrast to existing approaches in the literature. For instance, Bottmer et al. [2] discusses using sparse regression for large datasets with outliers. While their method efficiently manages large datasets, it stills fail to address multivariate conditional relationships between features. It uses the traditional approach discussed earlier and replaces outliers with expected values from the feature's estimated distribution. This is similar to using an expectation maximization algorithm to impute new data values for the outliers. However, we believe that augmenting the underlying data will lead to biased causal estimates. Given that our main goal is to reduce bias, we refrain from using approaches that augment data, rather those that omit outliers. This leads into work by Zanzing et al [3] which discusses creating a causal graph to explain which features are the root cause for certain outliers. This deals with the conditional relationships between features, however the method is computationally heavy, especially when we have to create a causal graph for 100+ features. Despite the plethora of outlier detection research, few methods have been applied to causal inference with large cross-sectional data.

Given the limitations of existing approaches, establishing a clear definition of outliers becomes critical for our analysis. We define outliers as individual observations that are the most statistically different from the rest of the observed data in a multivariate context. While outlier detection models may have different ways of determining statistical outliers, they follow a framework that measures and ranks how far an observation lies from other data. Outliers either differ from their local neighbors or are separate global points from the majority of the data. We employ several different multivariate outlier detection models to see which ones can remove bias from our treatment effect estimation the best.

To summarize the models tested, the table below describes model characteristics to help differentiate them. We note whether the models can handle multivariate data, use unsupervised methods, and if they can handle data with multiple conditional relationships (i.e. multiple data clusters). We also note the type of estimation method used, which can be trees, copulas, distributions, neural networks, or nearest-neighbors. Given the underlying structure of the data, choosing the appropriate method can offer advantages for a given use case. We further differentiate by the types of outliers that the models are designed to estimate and the number of hyper-parameters needed to run these models.

Table 1: Model Literature Summary

Model	Multivariate	Unsupervised	Ideal for Clusters	Outlier Detection Method	Outlier Detection Type	# Hyper-Parameters
IForest	Yes	Yes	Yes	Tree	Global	5
LOF	Yes	Yes	Yes	Tree	Local	6
COPOD	Yes	Yes	No	Copula	Global	1
MAD	No	Yes	No	Distribution	Global	2
XGBOD	Yes	No	Yes	Tree	Global	10+
AnoGAN	Yes	Yes	Yes	Neural Net	Global	10+
DeepSVDD	Yes	Yes	Yes	Neural Net	Global	10+
VAE	Yes	Yes	Yes	Neural Net	Global	10+
AutoEncoder	Yes	Yes	Yes	Neural Net	Global	10+
LUNAR	Yes	Yes	Yes	Neural Net	Global	10+
SUOD	Yes	Yes	Yes	Ensemble	Local/Global	10+
KNN	Yes	Yes	Yes	Nearest Neighbors	Local/Global	9

PCA	Yes	Yes	Yes	SVD	Global	8
ECOD	Yes	Yes	No	Distribution	Global	1
CBLOF	Yes	Yes	Yes	Tree	Local/Global	8

All of these models can be found in PyOD or Zhao et al. [1]. While our initial investigation encompassed the full range of models presented in Table 1, we focus our detailed analysis on six models that represent diverse methodological approaches and demonstrated promising performance in our preliminary testing. These include Isolation Forest (IForest), Cluster-based Local Outlier Factor (CBLOF), Copula-Based Outlier Detection (COPOD), Empirical CDF Outlier Detection (ECOD), K-Nearest Neighbors (KNN), Principal Component Outlier Detection (PCA). We didn't include any neural network models because of computational efficiency concerns for scalability. More details on these models can be found in the Appendix (A1).

2 Methodology

2.1 Outlier Detection Models

After generating simulated data, as described below, we apply our selected outlier detection models. The outlier detection models require the us to specify the hypothesized proportion of outliers (contamination ratio) to predict in the dataset. In most cases, we don't know the true proportion of outliers and our results summarize the impact of selecting different contamination ratios. While the models have more tunable hyper-parameters, they were kept at their default configurations to be consistent with Zhao et al [1].

Lastly, we combine the results of all the models in an ensemble estimate by identifying which observations the models all agree on as an outlier. This means that we will not be selecting a specified contamination ratio of outliers, rather only the agreed upon outliers among all models. This is done to minimize data loss of potential non-outliers. However, this means the models within the ensemble require a higher contamination ratio to predict the desired proportion of outliers in the data.

2.2 Double Machine Learning (DML) Framework

To obtain our causal treatment effect estimates we use the DML framework by Chernozhukov et al. [6] and Python implementation from Bach et al. [7]. This is an ideal framework for estimating treatment effects for how commonly it is used. Specifically, we use Partial Linear Regression learners, LightGBM regressors, due to their ability to handle large datasets efficiently. This provides a basis for how outlier detection models affect causal inference estimates within current causal inference frameworks. More information on the DML methodology can be found in the Appendix (A2).

2.3 Univariate Data Model Recommendations

Although our study focuses on multivariate outlier detection, it is valuable to understand traditional univariate approaches as a baseline comparison. Univariate outlier methods identify outliers through trimming the top and bottom percentiles of the data. One can either use the Z-scores or Median Absolute Deviations (MAD) to rank observations. If the data is approximately normally distributed on can trim data by how many standard deviations an observation is from the mean (i.e. Z-scores). Alternatively, if the distribution is not normal it is recommended to use MAD, which is an observation's median absolute deviation from the median.

3 Data Generation and Descriptions

3.1 Simulating Data and Description

Building upon our conceptual framework for outlier detection, we now present a rigorous methodology for evaluating these models under controlled conditions. To produce our simulated data, we break the process into two parts. First, simulating causal data and then modifying this data by converting a percentage of the data into outliers.

We simulate 100,000 rows and 100 columns using the process from Bach et al [4]. This procedure generates a dataset with discrete and continuous kinds of variables. In our case we have 10 discrete and 90 continuous variables to mimic industry data. It creates a random causal graph and models the variables linearly according to the relations in the graph. The graph can contain common causes, instrumental variables, effect modifiers, and multiple treatments. For simplicity we keep a single treatment without these additional complexities.

Few studies have attempted outlier detection models on datasets as large as the one described. This is because some models tend to take quite a long time to run, especially neural networks. Thus, model comparison can become computationally difficult. This experimental setup allows us to address a critical gap in the literature by evaluating how different outlier detection approaches perform and scale when applied to datasets of realistic size for modern causal inference applications.

3.1.1 Generating Outliers

Second, to create outliers within our simulated data we select random observations to be either local or global outliers:

Local outliers refer to the anomalies that deviate from their local neighborhoods. Assuming observations are clustered together in some shape, local outliers would sit slightly outside these clusters. They may lie between clusters, but are not strictly larger or smaller than other observations. We use a Gaussian Mixture Model (GMM) model to generate feature covariances to be used to generate synthetic normal observations. We use a scaled covariance matrix Σ of these selected observations, $\Sigma = \alpha\Sigma,$ to generate local anomalies. In our case, the scaling parameter α = 5.

Global outliers are points that are strictly larger or smaller than all the observed data. These global outliers generated randomly from either two uniform distributions $Unif(1/a * \min(X^k), \min(X^k))$ and $Unif(\max(X^k), a * \max(X^k))$. Where k-th feature X^k and α = 1.1 controls the extremeness of anomalies. These outliers will be further away from true data compared to local outliers.

The ratio of outliers injected is evenly split to between these two outlier classes. The proportion of the data that is to become outliers is equal to a contamination ratio (i.e. 10%, 5%, 1%, or 0.1%). We define the outliers using the same procedures and parameters α as performed in Anomaly Detection Benchmark (ADBench) by Songqiao et al. [5] to keep consistent outlier class definitions.

3.2 Real Input Data Description

We use the Home Equity Line of Credit (HELOC) dataset, which originally comes from Explainable Machine Learning Challenge organized by FICO company. The data contains anonymized credit applications of HELOC credit lines, which are a type of loan, collateralized by a customer's property. There are 23 predictors in the dataset. Examining the simulated data results, we find that using an ensemble of outlier detection models is the ideal approach for accurate outlier detection and preserving observations.

4 Results

4.1 Simulated Data Results

Table 4: Treatment Effect estimates with True Contamination = 10%

Assumed Contamination Ratio	Model	Outlier Accuracy	Data Reduction Percentage	Treatment Effect Estimate	Confidence Interval Lower Bound	Confidence Interval Upper Bound
--	Truth	--	--	17.43	16.42	18.44
-	No Model	--	--	0.00	0.00	0.00
10%	IForest	100%	10%	17.61	16.54	18.69
10%	COPOD	100%	10%	17.61	16.54	18.69
10%	ECOD	100%	10%	17.61	16.54	18.69
10%	KNN	100%	10%	17.61	16.54	18.69
10%	CBLOF	100%	10%	17.61	16.54	18.69
10%	PCA	100%	10%	17.61	16.54	18.69
10%	Percentile95	100%	10%	17.61	16.54	18.69
10%	Ensemble	100%	10%	17.61	16.54	18.69
1%	IForest	91%	1%	0.00	0.00	0.00
1%	COPOD	91%	1%	0.00	0.00	0.00
1%	ECOD	91%	1%	0.00	0.00	0.00
1%	KNN	91%	1%	0.00	0.00	0.00
1%	CBLOF	91%	1%	0.00	0.00	0.00
1%	PCA	91%	1%	0.00	0.00	0.00
1%	Percentile95	100%	10%	17.61	16.54	18.69
1%	Ensemble	91%	1%	0.00	0.00	0.00

Table 5: Treatment Effect estimates with True Contamination = 1%

Assumed Contamination Ratio	Model	Outlier Accuracy	Data Reduction Percentage	Treatment Effect Estimate	Confidence Interval Lower Bound	Confidence Interval Upper Bound
--	Truth	--	--	17.43	16.42	18.44
-	No Model	--	--	-0.01	-0.02	0.00
10%	IForest	91%	10%	17.53	16.60	18.46
10%	COPOD	91%	10%	17.37	16.43	18.32
10%	ECOD	91%	10%	17.24	16.31	18.17
10%	KNN	91%	10%	17.30	16.36	18.24
10%	CBLOF	99%	0%	-0.01	-0.02	0.00
10%	PCA	91%	10%	17.06	16.12	18.00
10%	Percentile95	3%	98%	28.47	18.93	38.01
10%	Ensemble	92%	9%	16.93	15.99	17.87
1%	IForest	100%	1%	17.55	16.68	18.42

1%	COPOD	100%	1%	17.55	16.68	18.42
1%	ECOD	100%	1%	17.55	16.68	18.42
1%	KNN	100%	1%	17.55	16.68	18.42
1%	CBLOF	99%	0%	-0.01	-0.02	0.00
1%	PCA	100%	1%	17.55	16.68	18.42
1%	Percentile95	3%	98%	28.47	18.93	38.01
1%	Ensemble	100%	1%	17.55	16.68	18.42

Based on these results, selecting a higher assumed contamination ratio leads to more accurate treatment effect estimations. This is most apparent when the true contamination ratio is 10% in Table 4. One can see as we select a higher assumed contamination (10% vs 1%), that the treatment effect estimates shift closer towards the true estimate. Despite different outlier detection methods, the assumed contamination ratio plays a larger role in improving the estimation of the true treatment effect. One can see that selecting a contamination ratio greater than or equal to the true proportion of outliers allows us to estimate treatment effects accurately. This is likely because the models can select more potential outliers with higher assumed contamination ratio. In some cases, the treatment effect confidence intervals decrease, but these differences are relatively negligible given how wide the intervals are.

Despite different methodologies, these selected models are quite similar in their ability to detect outliers. This might be because the outliers are completely simulated, unlike in Zhao et al. [3]. This could make outliers much easier for the models to detect, since they come from a measurable distribution. Additionally, since our simulated outliers are very close to the true simulated data observations, they will have smaller effects on the treatment effect estimates. Thus, if outliers are not sufficiently far from true data or the data contains less than 1% of them, an outlier detection model will likely not influence treatment effect estimations. One can also see how the traditional approach (Percentile95) removes a lot of data in this scenario (98%) showing how it is less ideal for outliers such as these.

Given that the models are similar in outlier detection accuracy and some outliers may have little impact, we should use an ensemble approach. This is because it only selects the most likely outliers from amongst the models used. The results in Table 5 indicate that our ensemble approach is more accurate in selecting the outliers, but only when the assumed contamination is set high enough. For example, when we set the assumed contamination to 1%, when the true contamination ratio is 10%, the ensemble model performs the worst in terms of accuracy. Additionally, the ensemble significantly reduces how much data is removed. In Table 5 we can see that the ensemble removes 9% of data when the assumed contamination ratio is 10%. This allows the model to have greater accuracy as it does not have a fixed assumed contamination ratio. It also highlights that our models strongly agree outliers. Testing this approach on more real data may prove the ensemble approach to be more promising in terms of accuracy. This is because the simulated data may not capture all different kinds of potential outlier data patterns that these models can pick up.

Some limitations arise when the true contamination is quite low, as selecting different contamination ratios has minimal effects on results. When the true contamination ratio becomes as low as 0.1% the results can tend to stray further from the true treatment effect values. Small amounts of outliers are hard for any model to differentiate from “true” data. For example, when the true contamination is 1%, IForest results change from positive to negative when going from 10% to 1% in assumed contamination. More results can be found in the Appendix (A4).

4.2 Real Data Results

Table 2: Treatment Effect Estimates using 10% Contamination Ratio

Model	Data Reduction Percentage	Treatment Effect Estimate	Confidence Interval Lower Bound	Confidence Interval Upper Bound
IForest	10%	0.20	-2.74	3.15
COPOD	10%	1.13	-2.13	4.39
ECOD	10%	1.04	-1.12	3.20
KNN	10%	2.08	-0.34	4.50
CBLOF	10%	1.79	-1.61	5.20
PCA	10%	0.78	-1.53	3.09
Percentile95	35.75%	0.26	-1.70	2.22
Ensemble	9.48%	0.87	-1.63	3.36
No Model	0%	4.44	0.53	8.35

Table 3: Treatment Effect Estimates using 1% Contamination Ratio

Model	Data Reduction Percentage	Treatment Effect Estimate	Confidence Interval Lower Bound	Confidence Interval Upper Bound
IForest	1%	1.11	-2.51	4.74
COPOD	1%	-0.84	-3.88	2.21
ECOD	1%	0.55	-2.24	3.34
KNN	1%	1.30	-1.30	3.91
CBLOF	1%	-0.04	-3.27	3.20
PCA	1%	1.22	-0.52	2.97
Percentile95	35.75%	0.26	-1.70	2.22
Ensemble	0.81%	1.25	-0.79	3.28
No Model	0%	4.44	0.53	8.35

The real data analysis reveals substantial impacts of outlier detection on treatment effect estimates. Without any outlier detection, the data shows a significant positive treatment effect of 4.44. However, this estimate changes dramatically when applying outlier detection methods.

When using a 10% contamination ratio (Table 2), all outlier detection models reduce the magnitude of the treatment effect substantially, with estimates ranging from 0.20 to 2.08. Most notably, all models' confidence intervals now include zero, suggesting that the original significant positive effect may have been driven by outliers. The ensemble approach removes slightly less data (9%) compared to individual models (10%) while producing a moderate estimate of 0.87.

With a 1% contamination ratio (Table 3), the results show more variation between models. Treatment effect estimates range from -0.84 (COPOD) to 1.30 (KNN), with some models even suggesting a direction change in the effect. The ensemble approach with 1% contamination removes only 0.81% of the data while producing an estimate of 1.25

The traditional Percentile95 approach appears particularly aggressive, removing 36% of the data regardless of the chosen contamination ratio. This substantial data reduction results in a relatively stable but potentially unreliable estimate of 0.26.

These results suggest that outliers may have been inflating the original treatment effect estimate, and their removal leads to more conservative conclusions about the treatment's impact. The ensemble

approach appears to offer a balanced solution, maintaining reasonable effect estimates while minimizing data reduction.

5 Conclusion

From the results, one can see that removing outliers has a significant impact on treatment effect estimates when the true percentage of outliers is high. Even with real data, the estimated impacts switch sign (positive to negative) after using any of the outlier detection models. While some models may perform better given different datasets, our results indicate that the Isolation Forest model is the best in terms of outlier accuracy and treatment effect estimation.

Despite the Isolation Forest model performing the best in terms of outlier detection accuracy, we recommend using the reduced ensemble method. This is because it minimizes data reduction and has similar outlier detection accuracy. However, if we assume too low of a contamination ratio, the ensemble approach tends to be not much better or even worse than using other models as seen in the simulated data results. Thus, an ensemble with a high assumed contamination ratio is ideal if we aren't sure of the true proportion of outliers. Given the difficulty of selecting the right model, the figure below highlights our strategy based on our findings.

Since underlying data patterns are hard to visualize for large datasets, we believe this ensemble framework is much more broadly applicable since it can capture outliers with varying patterns. It is possible that if too many models are introduced into the ensemble, the agreement between models can become too low and we won't be selecting enough the true outliers. Given that these models tend to lead to similar treatment effect estimates, the ensemble approach is recommended if preserving the original data is also a concern.

References

[1] Han, S., Hu, X., Huang, H., Jiang, M., & Zhao, Y. (2019). PyOD: A Python toolbox for scalable outlier detection. Journal of Machine Learning Research, 20(96), 1-7. doi: 10.48550/arXiv.1901.01588

[2] Bottmer, L., Croux, C., & Wilms, I. (2022). Sparse regression for large data sets with outliers. European Journal of Operational Research, 297(2), 782–794. doi: 10.1016/j.ejor.2021.05.049.

[3] Budhathoki, K., Minorics, L., Bloebaum, P., & Janzing, D. (2022). Causal structure-based root cause analysis of outliers. In Proceedings of the 39th International Conference on Machine Learning (pp. 2357–2369). PMLR. doi: 10.48550/arXiv.1912.02724.

[4] Bach, P., Blöbaum, P., Götz, P., Budhathoki, K., Mastakouri, A. A., & Janzing, D. (2022). DoWhy-GCM: An extension of DoWhy for causal inference in graphical causal models. arXiv preprint. doi:10.48550/arXiv.2206.06821.

[5] Han, S., Hu, X., Huang, H., Jiang, M., & Zhao, Y. (2022). ADBench: Anomaly detection benchmark. doi:10.48550/arXiv.2206.09426.

[6] Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W. & Robins, J. (2018), Double/debiased machine learning for treatment and structural parameters. The Econometrics Journal, 21: C1-C68, doi:10.1111/ectj.12097.

[7] Bach, P., Chernozhukov, V., Kurz, M. S., & Spindler, M. (2022). DoubleML – An object-oriented implementation of double machine learning in Python. Journal of Machine Learning Research, 23(53), 1-6. doi: 10.48550/arXiv.2104.03220

Appendix

A1 Outlier Model Descriptions

Below are descriptions of all the models used for presenting results for treatment effects. These models were presented for their variety in outlier detection methods.

A1.1 Isolation Forest (IForest)

The Isolation Forest 'isolates' observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature. Since recursive partitioning can be represented by a tree structure, the number of splits required to isolate a sample is equivalent to the path length from the root node to the terminating node. This path length, averaged over a forest of such random trees, is a measure of normality and our decision function. Random partitioning produces noticeably shorter paths for anomalies. Hence, when a forest of random trees collectively produces shorter path lengths for observations, they are highly likely to be anomalies.

A1.2 Copula Based Outlier Detection (COPOD)

COPOD is a parameter-free, highly interpretable outlier detection algorithm based on empirical copula models. The model first constructs the empirical copula or multivariate distribution, and then uses the fitted model to predict tail probabilities of each given data point to determine its level of "extremeness". Intuitively, we think of this as calculating an anomalous p-value for any given data point.

A1.3 Empirical CDF Outlier Detection (ECOD)

ECOD is a parameter-free, highly interpretable outlier detection algorithm based on empirical CDF functions of the feature space. Similar to COPOD, however it uses empirical CDFs instead of a Copula to identify outliers.

A1.4 K-Nearest Neighbors Outlier Detection (KNN)

For an observation, its distance to its kth nearest neighbor could be viewed as the outlying score. It could be viewed as a way to measure the density. Three KNN detectors are supported, use the distance to the kth neighbor as the outlier score, use the average of all k neighbors as the outlier and use the median of the distance to k neighbors as the outlier score. For this application we use the largest distance as the outlier score

A1.5 Cluster Based Local Outlier Factor (CBLOF)

The model calculates the outlier score based on cluster-based local outlier factor. It classifies the clusters into small clusters and large clusters using the parameters alpha and beta. The anomaly score is then calculated based on the size of the cluster the point belongs to as well as the distance to the nearest large cluster. However, to ensure stability, the model uses the largest cluster distance only.

A1.6 Principal Component Analysis (PCA)

Principal component analysis (PCA) can be used in detecting outliers. PCA is a linear dimensionality reduction using Singular Value Decomposition of the data to project it to a lower dimensional space. In this procedure, covariance matrix of the data can be decomposed to orthogonal vectors, called eigenvectors, associated with eigenvalues. The eigenvectors with high eigenvalues capture most of the variance in the data. Therefore, a low dimensional hyperplane constructed by k eigenvectors can capture most of the variance in the data. However, outliers are different from normal data points, which is more obvious on the hyperplane constructed by the eigenvectors with small eigenvalues. Therefore, outlier scores can be obtained as the sum of the projected distance of a sample on all eigenvectors.

A1.7 95th Percentile Outlier Detection (Percentile95)

This approach is the traditional way of removing outliers by removing data outside the 95th percentile in our data. For each column in our data, we select data that is more than two standard deviations away from the mean of the given column. If values are very close to the mean that would cause a large reduction of data and vice versa.

A2 Double Machine Learning Methodology

The Double (aka. Debiased) Machine learning (DML) method proposed by Chernozhukov et al. leverages the predictive power of modern Machine Learning (ML) methods in a principled causal estimation framework that is free of regularization bias asymptotically.

For treatment D, features X, we express the outcome Y as an additively separable function of D and arbitrary function of features X:

$$Y = D\beta + g(x) + \epsilon \quad (1)$$

DML's estimation strategy is motivated by writing out the residualized representation of Eq. (1) and its parts:

$$\tilde{Y} = Y - E(Y|X) \quad (2)$$

$$\tilde{D} = D - E(D|X) \quad (3)$$

$$\tilde{Y} = \tilde{D}\beta + \tilde{\epsilon} \quad (4)$$

We use ML models to estimate E(Y|X) and E(D|X). Specifically, LightGBM models for both estimations due to its speed and ability to handle large data.

A3 Additional Simulated Data Results

Table A4.1: True Contamination Ratio = 10%

Assumed Contamination Ratio	Model	Outlier Accuracy	Global Outlier Accuracy	Local Outlier Accuracy	Data Reduction Percentage	Treatment Effect Estimate	Confidence Interval Lower Bound	Confidence Interval Upper Bound
--	Truth	--	--	--	--	17.43	16.42	18.44
-	No Model	--	--	--	--	0.00	0.00	0.00
25%	IForest	85%	80%	80%	25%	17.34	16.09	18.58
25%	COPOD	85%	80%	80%	25%	17.37	16.17	18.56
25%	ECOD	85%	80%	80%	25%	17.80	16.55	19.04
25%	KNN	85%	80%	80%	25%	17.04	15.80	18.29
25%	CBLOF	85%	80%	80%	25%	17.52	16.26	18.77
25%	PCA	85%	80%	80%	25%	16.85	15.60	18.11
25%	Percentile95	100%	95%	95%	10%	17.61	16.54	18.69
25%	Ensemble	88%	83%	83%	22%	17.38	16.18	18.58

10%	IForest	100%	95%	95%	10%	17.61	16.54	18.69
10%	COPOD	100%	95%	95%	10%	17.61	16.54	18.69
10%	ECOD	100%	95%	95%	10%	17.61	16.54	18.69
10%	KNN	100%	95%	95%	10%	17.61	16.54	18.69
10%	CBLOF	100%	95%	95%	10%	17.61	16.54	18.69
10%	PCA	100%	95%	95%	10%	17.61	16.54	18.69
10%	Percentile95	100%	95%	95%	10%	17.61	16.54	18.69
10%	Ensemble	100%	95%	95%	10%	17.61	16.54	18.69
5%	IForest	95%	100%	90%	5%	0.00	0.00	0.00
5%	COPOD	95%	100%	90%	5%	0.00	0.00	0.00
5%	ECOD	95%	100%	90%	5%	0.00	0.00	0.00
5%	KNN	95%	100%	90%	5%	0.00	0.00	0.00
5%	CBLOF	95%	100%	90%	5%	0.00	0.00	0.00
5%	PCA	95%	100%	90%	5%	0.00	0.00	0.00
5%	Percentile95	100%	95%	95%	10%	17.61	16.54	18.69
5%	Ensemble	95%	100%	90%	5%	0.00	0.00	0.00
1%	IForest	91%	96%	94%	1%	0.00	0.00	0.00
1%	COPOD	91%	96%	94%	1%	0.00	0.00	0.00
1%	ECOD	91%	96%	94%	1%	0.00	0.00	0.00
1%	KNN	91%	96%	94%	1%	0.00	0.00	0.00
1%	CBLOF	91%	96%	94%	1%	0.00	0.00	0.00
1%	PCA	91%	96%	94%	1%	0.00	0.00	0.00
1%	Percentile95	100%	95%	95%	10%	17.61	16.54	18.69
1%	Ensemble	91%	96%	94%	1%	0.00	0.00	0.00
0.1%	IForest	90%	95%	95%	0.1%	0.00	0.00	0.00
0.1%	COPOD	90%	95%	95%	0.1%	0.00	0.00	0.00
0.1%	ECOD	90%	95%	95%	0.1%	0.00	0.00	0.00
0.1%	KNN	90%	95%	95%	0.1%	0.00	0.00	0.00
0.1%	CBLOF	90%	95%	95%	0.1%	0.00	0.00	0.00
0.1%	PCA	90%	95%	95%	0.1%	0.00	0.00	0.00
0.1%	Percentile95	100%	95%	95%	10%	17.61	16.54	18.69
0.1%	Ensemble	90%	95%	95%	0.1%	0.00	0.00	0.00

Table 2: True Contamination Ratio = 5%

Assumed Contamination Ratio	Model	Outlier Accuracy	Global Outlier Accuracy	Local Outlier Accuracy	Data Reduction Percentage	Treatment Effect Estimate	Confidence Interval Lower Bound	Confidence Interval Upper Bound
--	Truth	--	--	--	--	17.43	16.42	18.44
-	No Model	--	--	--	--	0.00	0.00	0.00

25%	IForest	80%	78%	78%	25%	17.03	16.02	18.05
25%	COPOD	80%	78%	78%	25%	18.11	17.17	19.06
25%	ECOD	80%	78%	78%	25%	17.79	16.80	18.79
25%	KNN	80%	78%	78%	25%	17.82	16.81	18.83
25%	CBLOF	80%	78%	78%	25%	17.79	16.79	18.79
25%	PCA	80%	78%	78%	25%	17.69	16.66	18.71
25%	Percentile95	22%	20%	20%	83%	21.92	19.45	24.38
25%	Ensemble	79%	77%	77%	26%	17.99	16.98	18.99
10%	IForest	95%	93%	93%	10%	17.98	17.09	18.86
10%	COPOD	95%	93%	93%	10%	18.03	17.15	18.90
10%	ECOD	95%	93%	93%	10%	17.90	17.02	18.78
10%	KNN	95%	93%	93%	10%	17.53	16.64	18.42
10%	CBLOF	95%	93%	93%	10%	17.38	16.49	18.28
10%	PCA	95%	93%	93%	10%	17.84	16.96	18.73
10%	Percentile95	22%	20%	20%	83%	21.92	19.45	24.38
10%	Ensemble	95%	93%	93%	10%	17.86	16.98	18.75
5%	IForest	100%	98%	98%	5%	17.99	17.12	18.85
5%	COPOD	100%	98%	98%	5%	17.99	17.12	18.85
5%	ECOD	100%	98%	98%	5%	17.99	17.12	18.85
5%	KNN	100%	98%	98%	5%	17.99	17.12	18.85
5%	CBLOF	100%	98%	98%	5%	17.99	17.12	18.85
5%	PCA	100%	98%	98%	5%	17.99	17.12	18.85
5%	Percentile95	22%	20%	20%	83%	21.92	19.45	24.38
5%	Ensemble	100%	98%	98%	5%	17.99	17.12	18.85
1%	IForest	96%	99%	97%	1%	0.00	0.00	0.00
1%	COPOD	96%	99%	97%	1%	0.00	0.00	0.00
1%	ECOD	96%	99%	97%	1%	0.00	0.00	0.00
1%	KNN	96%	99%	97%	1%	0.00	0.00	0.00
1%	CBLOF	96%	99%	97%	1%	0.00	0.00	0.00
1%	PCA	96%	99%	97%	1%	0.00	0.00	0.00
1%	Percentile95	22%	20%	20%	83%	21.92	19.45	24.38
1%	Ensemble	96%	99%	96%	1%	0.00	0.00	0.00
0.1%	IForest	95%	98%	97%	0.1%	0.00	0.00	0.00
0.1%	COPOD	95%	98%	97%	0.1%	0.00	0.00	0.00
0.1%	ECOD	95%	98%	97%	0.1%	0.00	0.00	0.00
0.1%	KNN	95%	98%	97%	0.1%	0.00	0.00	0.00
0.1%	CBLOF	95%	98%	97%	0.1%	0.00	0.00	0.00
0.1%	PCA	95%	98%	97%	0.1%	0.00	0.00	0.00
0.1%	Percentile95	22%	20%	20%	83%	21.92	19.45	24.38
0.1%	Ensemble	95%	98%	97%	0.1%	0.00	0.00	0.00

Table A4.3: True Contamination Ratio = 1%

Assumed Contamination Ratio	Model	Outlier Accuracy	Global Outlier Accuracy	Local Outlier Accuracy	Data Reduction Percentage	Treatment Effect Estimate	Confidence Interval Lower Bound	Confidence Interval Upper Bound
--	Truth	--	--	--	--	17.43	16.42	18.44
-	No Model	--	--	--	--	-0.01	-0.02	0.00
25%	IForest	76%	76%	76%	25%	17.27	16.24	18.31
25%	COPOD	76%	76%	76%	25%	17.06	15.98	18.15
25%	ECOD	76%	76%	76%	25%	17.52	16.47	18.57
25%	KNN	76%	76%	76%	25%	17.53	16.48	18.58
25%	CBLOF	99%	100%	100%	0%	-0.01	-0.02	0.00
25%	PCA	76%	76%	76%	25%	17.34	16.28	18.40
25%	Percentile95	3%	2%	2%	98%	28.47	18.93	38.01
25%	Ensemble	77%	76%	76%	24%	17.06	16.01	18.10
10%	IForest	91%	91%	91%	10%	17.53	16.60	18.46
10%	COPOD	91%	91%	91%	10%	17.37	16.43	18.32
10%	ECOD	91%	91%	91%	10%	17.24	16.31	18.17
10%	KNN	91%	91%	91%	10%	17.30	16.36	18.24
10%	CBLOF	99%	100%	100%	0%	-0.01	-0.02	0.00
10%	PCA	91%	91%	91%	10%	17.06	16.12	18.00
10%	Percentile95	3%	2%	2%	98%	28.47	18.93	38.01
10%	Ensemble	92%	91%	91%	9%	16.93	15.99	17.87
5%	IForest	96%	96%	96%	5%	17.30	16.41	18.20
5%	COPOD	96%	96%	96%	5%	17.46	16.54	18.38
5%	ECOD	96%	96%	96%	5%	17.27	16.34	18.19
5%	KNN	96%	96%	96%	5%	17.20	16.30	18.11
5%	CBLOF	99%	100%	100%	0%	-0.01	-0.02	0.00
5%	PCA	96%	96%	96%	5%	17.43	16.54	18.32
5%	Percentile95	3%	2%	2%	98%	28.47	18.93	38.01
5%	Ensemble	97%	96%	96%	4%	17.23	16.34	18.12
1%	IForest	100%	100%	100%	1%	17.55	16.68	18.42
1%	COPOD	100%	100%	100%	1%	17.55	16.68	18.42
1%	ECOD	100%	100%	100%	1%	17.55	16.68	18.42
1%	KNN	100%	100%	100%	1%	17.55	16.68	18.42
1%	CBLOF	99%	100%	100%	0%	-0.01	-0.02	0.00
1%	PCA	100%	100%	100%	1%	17.55	16.68	18.42
1%	Percentile95	3%	2%	2%	98%	28.47	18.93	38.01
1%	Ensemble	100%	100%	100%	1%	17.55	16.68	18.42
0.1%	IForest	99%	100%	99%	0.1%	-0.01	-0.02	0.00

0.1%	COPOD	99%	100%	99%	0.1%	-0.01	-0.02	0.00
0.1%	ECOD	99%	100%	99%	0.1%	-0.01	-0.02	0.00
0.1%	KNN	99%	100%	99%	0.1%	-0.01	-0.02	0.00
0.1%	CBLOF	99%	100%	100%	0%	-0.01	-0.02	0.00
0.1%	PCA	99%	100%	99%	0.1%	-0.01	-0.02	0.00
0.1%	Percentile95	3%	2%	2%	98.2%	28.47	18.93	38.01
0.1%	Ensemble	99%	100%	99%	0.1%	-0.01	-0.02	0.00

Table A4.4: True Contamination Ratio = 0.1%

Assumed Contamination Ratio	Model	Outlier Accuracy	Global Outlier Accuracy	Local Outlier Accuracy	Data Reduction Percentage	Treatment Effect Estimate	Confidence Interval Lower Bound	Confidence Interval Upper Bound
--	Truth	--	--	--	--	17.43	16.42	18.44
-	No Model	--	--	--	--	-0.07	-0.32	0.18
25%	IForest	75%	75%	75%	25%	17.87	16.62	19.13
25%	COPOD	75%	75%	75%	25%	17.21	15.90	18.52
25%	ECOD	75%	75%	75%	25%	17.63	16.37	18.89
25%	KNN	75%	75%	75%	25%	17.90	16.65	19.15
25%	CBLOF	100%	100%	100%	0%	-0.07	-0.32	0.18
25%	PCA	75%	75%	75%	25%	17.58	16.34	18.83
25%	Percentile95	1%	1%	1%	99%	30.74	11.29	50.19
25%	Ensemble	76%	76%	76%	24%	16.97	15.71	18.22
10%	IForest	90%	90%	90%	10%	18.04	16.95	19.14
10%	COPOD	90%	90%	90%	10%	17.81	16.69	18.94
10%	ECOD	90%	90%	90%	10%	17.91	16.81	19.01
10%	KNN	90%	90%	90%	10%	17.83	16.73	18.94
10%	CBLOF	100%	100%	100%	0%	-0.07	-0.32	0.18
10%	PCA	90%	90%	90%	10%	18.09	16.99	19.19
10%	Percentile95	1%	1%	1%	99%	30.74	11.29	50.19
10%	Ensemble	91%	91%	91%	9%	17.63	16.53	18.73
5%	IForest	95%	95%	95%	5%	18.03	16.96	19.10
5%	COPOD	95%	95%	95%	5%	17.76	16.70	18.82
5%	ECOD	95%	95%	95%	5%	18.11	17.07	19.15
5%	KNN	95%	95%	95%	5%	17.82	16.76	18.88
5%	CBLOF	100%	100%	100%	0%	-0.07	-0.32	0.18
5%	PCA	95%	95%	95%	5%	17.97	16.91	19.04
5%	Percentile95	1%	1%	1%	99%	30.74	11.29	50.19
5%	Ensemble	96%	96%	96%	4%	17.62	16.56	18.67
1%	IForest	99%	99%	99%	1%	17.89	16.86	18.91

1%	COPOD	99%	99%	99%	1%	18.23	17.20	19.27
1%	ECOD	99%	99%	99%	1%	17.96	16.93	18.98
1%	KNN	99%	99%	99%	1%	17.67	16.65	18.69
1%	CBLOF	100%	100%	100%	0%	-0.07	-0.32	0.18
1%	PCA	99%	99%	99%	1%	18.24	17.23	19.25
1%	Percentile95	1%	1%	1%	99%	30.74	11.29	50.19
1%	Ensemble	99%	99%	99%	0.8%	18.00	16.98	19.02
0.1%	IForest	100%	100%	100%	0.1%	18.04	17.03	19.05
0.1%	COPOD	100%	100%	100%	0.1%	18.04	17.03	19.05
0.1%	ECOD	100%	100%	100%	0.1%	18.04	17.03	19.05
0.1%	KNN	100%	100%	100%	0.1%	18.04	17.03	19.05
0.1%	CBLOF	100%	100%	100%	0%	-0.07	-0.32	0.18
0.1%	PCA	100%	100%	100%	0.1%	18.04	17.03	19.05
0.1%	Percentile95	1%	1%	1%	99%	30.74	11.29	50.19
0.1%	Ensemble	100%	100%	100%	0.1%	18.04	17.03	19.05