Detecting Bad Answers in Survey Data through Unsupervised Machine Learning

Najeeb Moharram Jebreel, Rami Haffar, Ashneet Khandpur Singh, David Sánchez, Josep Domingo-Ferrer, and Alberto Blanco-Justicia

Universitat Rovira i Virgili,
Department of Computer Engineering and Mathematics,
CYBERCAT-Center for Cybersecurity Research of Catalonia,
UNESCO Chair in Data Privacy,
Av. Paros Catalans 26, 43007 Tarragona, Catalonia
E-mail: {najeebmoharramsalim.jebreel, rami.haffar, ashneet.singh, david.sanchez, josep.domingo, alberto.blanco}@urv.cat

Abstract. Surveys are one of the most common ways of collecting data on individuals. Such data are of great value for economic and social research. However, the quality of the decisions and research results based on survey data depends on the ability to detect and filter out bad answers. The most common source of bad data are the respondents, who might provide imprecise or fabricated answers due to several reasons. In this paper we present a method to sanitize survey data that relies on combining the classification outcomes of three unsupervised machine learning algorithms (DBSCAN, PCA and IForest) aimed at detecting bad answers. Empirical results on real data show that our approach is able to improve the detection of both completely and partially bad answers with respect to the results provided by each algorithm independently.

Keywords: Survey data, Data quality, Bad answers, Fabrication, Unsupervised machine learning.

1 Introduction

Survey data are central to many decision-making processes taking place at public and private institutions. They are also of paramount importance for research studies in fields such as economics, politics, social sciences or medicine. The quality of the decisions and research results based on survey data is, however, heavily dependent on the quality of the data themselves [1].

Experience shows that not all survey data are as good as desired. Errors during the survey design, such as choosing non-representative population samples or introducing unclear questions may lead to bad results [2]. Nonetheless, the most common source of bad data are the respondents, who might provide imprecise responses or fabricated answers when they are faced with sensitive questions, when they feel too much private information is requested or when they are not interested in the survey [3].

A literature survey in [4] shows that the percentage of fabricated survey data is typically below 5% for large-scale surveys with strict supervision, reaching up to 50% in small surveys with limited supervision (*e.g.* with inaccessible respondents). Regarding the effects of data fabrication, even small levels of fabrication may cause big effects. For example, in the German Socio-Economic Panel (SOEP) survey, removing 2.5% of the fabricated answers changed the effect of the years of education on gross wages by 80% [5]. The authors in [6] conclude that if roughly 1% of the answers are fabricated via, *e.g.* duplication, the probability of obtaining unbiased statistical estimates is of 41.6%; and if 10% or more of the answers are fabricated, the probability falls to 11.4%.

Given the importance of good data, effective methods are needed to sanitize survey data before conducting any meaningful empirical research or making decisions. Several methods have been proposed to detect bad answers in surveys, which involve re-interviewing respondents, recording and analyzing interviews with respondents, or analyzing statistical features of the responses [7]. However, the application of such verification checks incurs high costs.

Machine learning (ML) classification algorithms are a promising alternative. In particular, unsupervised ML algorithms, which extract patterns from unlabeled data with little human intervention, are widely used in anomaly detection; this includes detecting credit card fraud, cyberattacks, changes in medical conditions and unusual images [8]. In this work, we apply unsupervised ML algorithms to detect bad answers in survey data.

Although ML algorithms have been used to solve many regression and classification problems, no individual algorithm performs best under every circumstance. A common way to overcome this problem is to combine different algorithms [9]. Following this line, we propose aggregating a chosen set of ML algorithms to detect bad answers.

Contribution and organization of this paper

We introduce a new approach for detecting bad responses in survey data. We distinguish two kinds of bad responses: completely bad responses, in which the answers to all survey items have been fabricated in some way (*e.g.* by providing systematic random answers); and partially bad responses, where only part of the answers are fabricated (*e.g.* to hide sensitive features). Our proposal aggregates the results from three different unsupervised ML algorithms, namely linear-based, density-based and ensemble-based methods. With this approach, we aim to improve the detection of both completely and partially bad answers.

The main contributions of our work are:

- We show that unsupervised machine learning algorithms used to detect anomalies and fraud in categorical data can be effectively used to detect bad answers in survey data.
- We use three straightforward unsupervised machine learning methods, which have low computational cost, are easy to implement and have good performance.

– We show that the accuracy of detecting bad answers can be improved by aggregating the results from those three algorithms.

The results show that our approach detects completely bad answers with an overall accuracy over 96%, and a true positive rate (TPR) over 95%. Also, we obtain an overall accuracy over 80% and a true positive rate (TPR) over 75% in detecting partially bad answers.

The remainder of the paper is organized as follows. Section 2 discusses related works aimed at detecting fabricated survey data. Section 3 presents our approach and details the three ML algorithms we use. Section 4 describes and reports the results of the empirical experiments. Finally, Section 5 depicts the conclusions and proposes several lines of future research.

2 Related Work

The authors of [10] used Census Bureau Studies data to show the importance of re-interviewing respondents as a means of fabrication detection. In their study, most faked responses were detected through re-interviews, although some were detected due to anomalies in the data. Additionally, most of the cheating involved consisted of complete fabrications rather than partial fabrications of responses. In [11], the authors used data anomalies from the National Survey on Drug Use and Health in the US to determine suspicious behaviors. Specifically, they used the interview duration as a potential sign of fabrication allowing fraud detection in survey data.

Benford's law has been widely used to detect bad answers in surveys [12,13]. Benford's law involves examining the distribution of the leading (or left-most) digits of all the numbers reported on a survey form. By comparing against the expected distribution of the leading digits, one can identify unusual data, which may be fraudulent or generated by an error-prone process: suspect interviews are those in which the distribution of leading digits does not follow the expected distribution. Furthermore, [14] introduces an approach for detecting falsification in public opinion data. The proposed measure is the maximum percentage match statistic, which is the maximum percentage of questions on which each respondent matches any other respondent in the survey data.

Machine learning has been also employed to detect bad answers. The authors of [15] use cluster analysis on a data set that contains labels identifying the interviewers who fabricated entire interviews. They combine Benford-related features with other attributes (like item non-response rate) for which they expect differences between honest interviewers and cheaters. In cluster analysis, the classes of the groups of data points are not known to the algorithms, which means that the actual classes are not used for the classification task. In [16] the authors employ supervised and unsupervised classification to automatically detect interviewer fabrication in three real data sets containing both completely and partially bad answers. The authors report a better performance for supervised algorithms.

Each of the above methods above has its advantages and disadvantages. Re-interviewing respondents is effective to detect cheaters, but it requires substantial

time and personnel effort. Methods based on Benford's law are only suitable for numerical data, but survey data typically contain a mix of numerical, ordinal and categorical data. Supervised ML algorithms are undoubtedly effective if labeled data are available, but access to such data is a major challenge.

Unsupervised ML algorithms are a promising alternative, as they are easy to apply and do not require labeled data.

3 Methodology

In this work, we use three fast, effective unsupervised ML algorithms to detect and remove bad answers in surveys. Each of the considered algorithms is widely used to detect anomalies in multidimensional data points. Then we aggregate the decisions of the three algorithms to improve the accuracy of detecting bad answers.

We model survey data as microdata tables, where each row corresponds to a single respondent and each column to each of the questions in the survey. Before we apply any of the detection algorithms, the data are preprocessed. Numerical data are normalized and categorical data are one-hot encoded.

The methods used in this paper are chosen based on their performance in clustering and anomaly detection, and on their execution time. The first method is DBSCAN, a widely used density-based clustering algorithm. DBSCAN is described in Section 3.1. The second method is Principal Component Analysis (PCA), often used in exploratory data analysis and dimensionality reduction. PCA is described in Section 3.2. The third method is IForest, an ensemble-based model that works on the principle of isolating anomalies in datasets. The IForest model is described in Section 3.3.

The answer i from every respondent in the survey (rows in our microdata table) is tested with each of these methods ($score_dbscan(i)$, $score_iforest(i)$, $score_pca(i)$), which results in a score either 1 if the answer is considered an outlier or 0 otherwise. The overall score of i is 1 if at least one of the methods scores it 1, in which case the respondent is classified as a cheater, and 0 otherwise. This process is formalized in Algorithm 1.

Algorithm 1: Bad survey data detection algorithm.

```
Input: Survey_data;
Output: Survey_without_cheaters_data;
numerical_data ← one_hot_encoding(Survey_data);
normalized_data ← normalize(numerical_data);
for i in normalized_data do
    score_dbscan(i) ← predict_dbscan(i);
    score_iforest(i) ← predict_iforest(i);
    score_pca(i) ← predict_pca(i);
    if score_dbscan(i) + score_iforest(i) + score_pca(i) ≥ 1 then
        Survey_Data ← Survey_Data.drop(i);
```

3.1 Density-based spatial clustering of applications with noise (DBSCAN)

DBSCAN [17] is a clustering algorithm that groups data points according to their density with respect to a distance parameter ϵ. DBSCAN identifies outlying points as those lying in low-density regions, *i.e.* points that are unreachable from other points when taking ϵ-distance jumps. It has a worst-case time complexity $\mathcal{O}(n^2)$.

Next, we summarize the working principles of the algorithm:

1. *Neighboring points* are those at a distance smaller than ϵ.
2. *Core points* are those points that have at least $MinPts$ neighbors.
3. A point is *directly reachable* if it is within distance ϵ from a core point.
4. A point is p *reachable* from q if there is a path of core points between p and q.
5. A core point p forms a cluster with all points reachable from it.
6. Points that are not reachable, and thus do not belong to any cluster, are *outliers*.

3.2 Principal Component Analysis (PCA)

Principal Component Analysis was first proposed in [18] and it is widely used for dimensionality reduction, data visualization and exploratory data analysis.

PCA is a linear transformation that takes the data points into an orthonormal coordinate system that has the principal components as its basis. The data in the new coordinate system have their components uncorrelated (*i.e.* the covariance or correlation matrix is diagonal). The record components are ordered with respect to their variance, with the first component having the highest variance and the sum of the variances contributed by each of the components being equal to that of the original data. The principal components can be obtained from an eigenvector decomposition of the covariance or correlation matrices of the data [19].

We use an outlier detection mechanism based on PCA, which was proposed in [20] to identify attacks or outliers in data sets. First, a robust estimator of the correlation matrix is obtained via multivariate trimming and the principal components are obtained. Two detection functions are used: the first one depends on *major* components of the transformed data points (those which explain about 50% of the variance) and is able to detect whether the values for single components of the original data points are outliers. The second one, based on *minor* components (components that contribute each less than 0.2 of the variance) is able to detect anomalies in the correlation structure of the original data points. If either of these functions are above a certain threshold, the original data point is considered an outlier.

3.3 Isolation Forest (IForest)

Isolation forests are unsupervised anomaly detection algorithms. Anomaly detection algorithms typically model the normal behavior of data points and look for points that do not follow this normal behavior. In contrast, isolation forests explicitly isolate outliers [21].

An isolation forest is a collection of isolation trees, which are data-induced random binary trees. Isolation trees contain external nodes, which are nodes with no children, and internal nodes, which are nodes with a test ($q < p$) and exactly two children. To build such trees, the data set is recursively split by choosing a random attribute q and a random split value p in the range of the chosen attribute until i) the tree reaches a predefined maximum height, ii) the data set contains a single element, or iii) all remaining data points are equal.

The working principle of isolation forests is that, given such random trees, outliers will be isolated (be assigned to an external node) with less splits on average than normal data points, that is, their height (distance from the root) will be smaller on average.

To identify anomalies, data points are searched in the collection of isolation trees and their average height is computed. This average height is normalized to obtain an anomaly score s as follows:

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}},$$

where $E(h(x))$ is the average height of x in the collection of isolation trees and $c(n)$ is the average path length of unsuccessful searches in binary search trees, given in [22]. Data points with anomaly scores close to 1 are considered anomalous.

4 Experiments

4.1 Dataset

To test the performance of the three considered algorithms and their aggregated results, we used the Adult income data set [23]. This data set contains $48,842$ records, each of them reporting 6 continuous and 9 categorical demographic attributes on a certain individual. We viewed Adult as if it resulted from a survey, that is, as if each record represented the answer from an individual respondent.

After cleaning up the survey data from missing values we still had $45,222$ answers. Then, we simulated two types of bad answers: *completely bad answers* and *partially bad answers*. For completely bad answers, we randomly chose 20% of the records and replaced values in them with random answers generated from the respective domains of the 15 attributes. For the partially bad answers, we randomly selected 20% of the records; after that, we randomly chose 6 out of the 15 attributes in the selected records and replaced their values with random values from the corresponding attribute domains. We then attached a label to each record. The label value was set to 1 if the answer was bad or 0 if the

answer was good. Later, we used these labels to evaluate the performance of the methods.

For both completely and partially bad answers, we performed the same data preprocessing steps. First, we converted the categorical attributes into numerical using label encoding. After that, we applied min-max normalization for each attribute value, defined as

$$z_i = \frac{x_i - min(x)}{max(x) - min(x)},$$

where $x = (x_1, ..., x_n)$ and z_i is the i-th normalized value.

4.2 Performance metrics

Since we were dealing with a classification problem (to classify an answer as good or bad), we used five well-known evaluation metrics that have been used to assess the performance of classification methods [24]:

- **Accuracy** (ACC) represents the percentage of correctly classified answers compared with the total number of answers. We used the accuracy to assess the overall performance of each method. Accuracy is given by:

  $$ACC = \frac{TP + TN}{TP + TN + FP + FN} \times 100,$$

  where TP, TN, FP and FN stand, respectively, for number of true positives, number of true negatives, number of false positives and number of false negatives.
- **True positive rate** (TPR), a.k.a. detection rate or sensitivity, represents the percentage of correctly classified bad answers compared with the total number of bad answers. TPR is given by:

  $$TPR = \frac{TP}{TP + FN} \times 100.$$

- **True negative rate** (TNR), a.k.a. recall, represents the percentage of correctly classified good answers compared with the total number of good answers. TNR is given by:

  $$TNR = \frac{TN}{TN + FP} \times 100.$$

- **False positive rate** (FPR), a.k.a. fall out, represents the percentage of good answers incorrectly classified as bad answers compared with the total number of good answers. FPR is given by:

  $$FPR = \frac{FP}{FP + TN} \times 100.$$

- **False negative rate** (FNR) represents the percentage of bad answers incorrectly classified as good answers compared with the total number of bad answers. FNR is given by:

  $$FNR = \frac{FN}{FN + TP} \times 100.$$

4.3 Hyper-parameters

We used different hyper-parameter values with each method.

For DBSCAN applied to completely bad answer detection, we used $\epsilon = 0.515$ and a minimum number of answers required to form a cluster $MinPts = 300$. For DBSCAN applied to partially bad answer detection, we used stricter values $\epsilon = 0.42$ and $MinPts = 500$.

For IForest, we used the default parameters as described in [25]. Then, we filtered the scores obtained from the decision function of IForest using $\tau = 0.2$ for completely bad answers and $\tau = 0.1$ for partially bad answers.

For PCA, we used the default parameters specified in PyOD [26] with a number of components equal to 2. PyOD implements [20] to use PCA as an outlier detection tool.

4.4 Results

Figure 2 shows the performance at detecting completely bad answers. We can see that both DBSCAN and IForest have a similar performance with an overall accuracy 96%, and with a slight advantage for DBSCAN regarding TPR and FPR: DBSCAN achieved TPR 91.3% and FPR 2.3%. PCA had a somewhat poorer performance.

Anyway, the aggregation of the three methods as detailed in Algorithm 1 results in a noticeable improvement. On the one hand, the overall accuracy is maintained at 96% like for DBSCAN and IForest. On the other hand, the TPR 95.3% outperformed the best score achieved by DBSCAN, which was 91.3%.

Figure 4 shows the performance at detecting partially bad answers. We can notice the good performance of both DBSCAN and IForest: DBSCAN achieved an overall accuracy 85.1% while IForest achieved 84.3%. However, IForest achieved the best TPR performance at 64.8%. We also see a poor performance of PCA at detecting partially bad answers: TPR was just 34.4%; however, PCA achieved a very good TNR 96%.

Again, we see a noticeable improvement when aggregating the three methods: the aggregation achieved TPR 72.98%, while IForest was the second best TPR performer with only 64.78%.

From the results, it is clear that aggregating the results of the three methods improves the detection rate of bad answers. The aggregation helps to complement the different strategies implemented by the three methods at detecting bad answers and overcome some of their limitations. It is also worth noting that the performance of detecting partially bad answers is noticeably lower than detecting completely bad answers. This is not surprising because in this latter case only 40% of the values of each answer were randomly generated. Consequently, some partially random answers might still be close to good answers and, therefore, would be considered good.

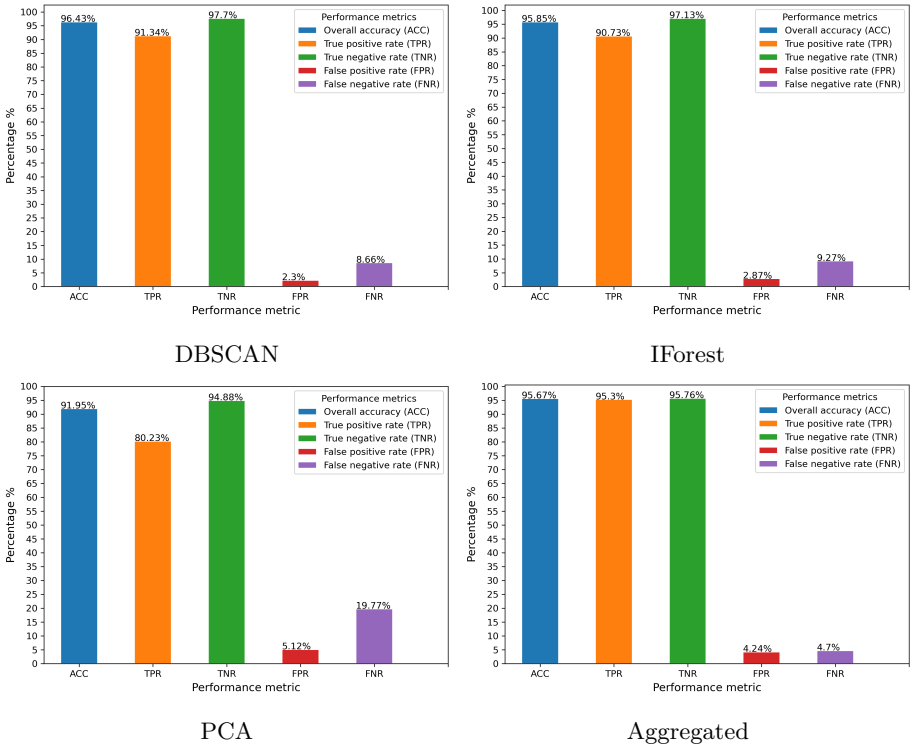


Fig. 2: Performance at detecting completely bad answers

5 Conclusions and future research

We have introduced a method to detect fabricated answers in surveys that aggregates the results from different unsupervised machine learning and anomaly detection algorithms. Our experiments show that the aggregated results improve the detection rate of bad answers with respect to the outcomes of each individual algorithm.

Despite the promising results, the choice of the hyperparameters remains a major challenge. The value of these parameters depends on several factors like the domain of the survey and the size of the answers. More experiments will be needed to analyze the influence of the parameters on the results and, particularly, on their aggregation.

As future work, we plan to investigate other aggregation strategies and unsupervised ML methods. We will also consider specific cases of data fabrication, such as answer duplication, for which specific methods have been proposed.

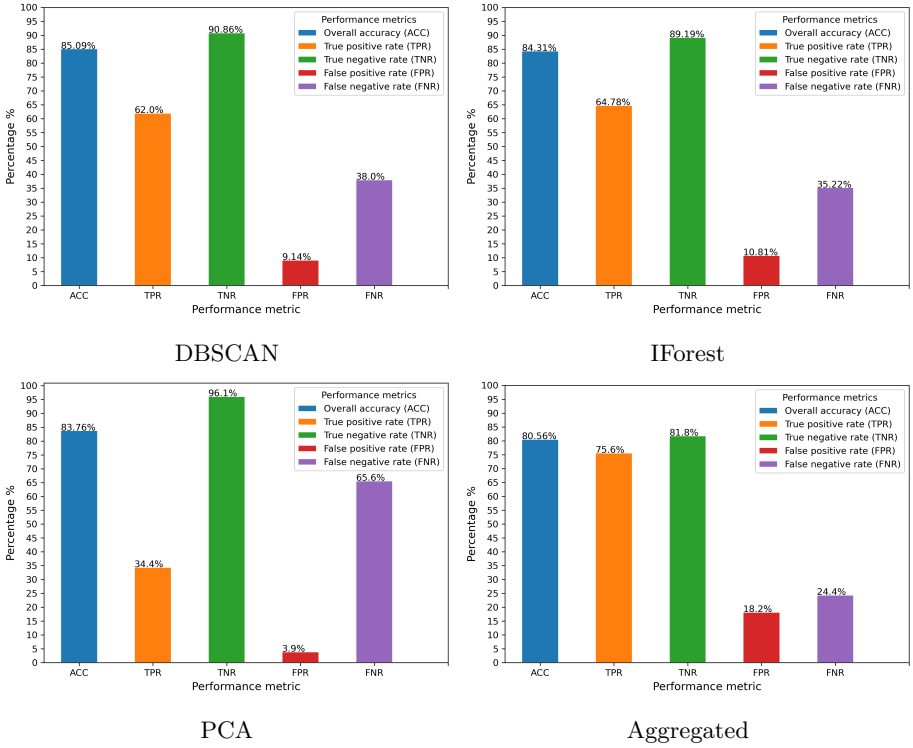


Fig. 4: Performance at detecting partially fabricated answers

Acknowledgments and disclaimer

We acknowledge support from: European Commission (project H2020-871042 "SoBigData++"), Government of Catalonia (ICREA Acadèmia Prize to Josep Domingo-Ferrer and grant 2017 SGR 705) and Spanish Government (projects RTI2018-095094-B-C21 and TIN2016-80250-R). The authors are with the UNESCO Chair in Data Privacy, but their views here are not necessarily shared by UNESCO.

References

1. P. P. Biemer and L. E. Lyberg, *Introduction to Survey Quality*. John Wiley & Sons, 2003.
2. P. Winker, "Assuring the quality of survey data: Incentives, detection and documentation of deviant behavior," *Statistical Journal of the IAOS*, vol. 32, no. 3, pp. 295–303, 2016.
3. F. Rogers and M. Richarme, "The honesty of online survey respondents: Lessons learned and prescriptive remedies," Decision Analyst, Inc White Papers, 2009. https://www.decisionanalyst.com/whitepapers/onlinerespondents/

4. S. Bredl, N. Storfinger, and N. Menold, "A literature review of methods to detect fabricated survey data," Discussion Paper, Tech. Rep., 2011. https://www.econstor.eu/handle/10419/74449
5. J.-P. Schraepler and G. G. Wagner, "Identification, characteristics and impact of faked interviews in surveys: An analysis by means of genuine fakes in the raw data of SOEP," IZA Discussion Paper no. 969, 2003. http://hdl.handle.net/10419/20205
6. F. Sarracino and M. Mikucka, "Estimation bias due to duplicated observations: a Monte Carlo simulation," Munich Personal RePEc Archive-MPRA, 2016.
7. C. Schwemmer, *Identification of fabricated interviews in surveys – An analysis using the Socio-Economic Panel (SOEP)*, Working Paper, Sep. 2015. https://www.researchgate.net/publication/318888495_Identification_of_fabricated_interviews_in_surveys_-_An_analysis_using_the_Socio-Economic_Panel_SOEP
8. V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009.
9. T. G. Dietterich *et al.*, "Ensemble learning," *The Handbook of Brain Theory and Neural Networks*, vol. 2, pp. 110–125, 2002.
10. I. D. Schreiner, J. Newbrough, and K. Pennie, *Interviewer falsification in Census Bureau surveys.* US Census Bureau [custodian], 1988.
11. J. Murphy, R. Baxter, J. Eyerman, D. Cunningham, and J. Kennet, "A system for detecting interviewer falsification," in *American Association for Public Opinion Research 59th Annual Conference*, 2004, pp. 4968–4975.
12. S. Bredl, P. Winker, and K. Kötschau, "A statistical approach to detect interviewer falsification of survey data," *Survey Methodology*, vol. 38, no. 1, pp. 1–10, 2012.
13. C. Schäfer, J.-P. Schräpler, K.-R. Müller, and G. G. Wagner, "Automatic identification of faked and fraudulent interviews in surveys by two different methods," DIW Discussion Papers, Tech. Rep., 2004. http://hdl.handle.net/10419/18293
14. N. Kuriakose and M. Robbins, "Falsification in survey research: Detecting near-duplicate observations," in *Annual Meeting of the American Political Science Association*, San Francisco, CA, September, 2015, pp. 3–6.
15. S. Bredl, P. Winker, and K. Kötschau, "A statistical approach to detect cheating interviewers," Discussion Paper, Tech. Rep., 2008. https://www.econstor.eu/handle/10419/39808
16. B. Birnbaum, *Algorithmic Approaches to Detecting Interviewer Fabrication in Surveys,* Ph.D. dissertation, University of Washington, 2012. https://digital.lib.washington.edu/researchworks/handle/1773/22011
17. M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "Density-based spatial clustering of applications with noise," in *Int. Conf. Knowledge Discovery and Data Mining-KDD'96*, 1996, pp. 226-231.
18. K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.
19. S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37–52, 1987.
20. M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang, *A novel anomaly detection scheme based on principal component classifier*, University of Miami Coral Gables, Dept. of Electrical and Computer Engineering, Tech. Rep., 2003. https://apps.dtic.mil/dtic/tr/fulltext/u2/a465712.pdf
21. F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining.* IEEE, 2008, pp. 413–422.
22. B. Preis, *Data Structures and Algorithms with Object-Oriented Design Patterns in Java,* John Wiley&Sons, 1999.

23. D. Dua and C. Graff, "Uci machine learning repository," Irvine CA: University of California, School of Information and Computer Science, 2019.
24. C. E. Metz, "Basic principles of ROC analysis,", *Seminars in Nuclear Medicine*, vol. VIII, no. 4, pp. 283–298, 1978.
25. F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 1, pp. 1–39, 2012.
26. Y. Zhao, Z. Nasrullah, and Z. Li, "Pyod: A python toolbox for scalable outlier detection," *Journal of Machine Learning Research*, vol. 20, no. 96, pp. 1–7, 2019.