2019 IEEE 35th International Conference on Data Engineering (ICDE)

# Towards Explaining the Effects of Data Preprocessing on Machine Learning

Carlos Vladimiro González Zelaya
*School of Computing*
*Newcastle University*
*Newcastle upon Tyne, UK*
*c.v.gonzalez-zelaya2@newcastle.ac.uk*

***Abstract*—Ensuring the explainability of machine learning models is an active research topic, naturally associated with notions of algorithmic transparency and fairness. While most approaches focus on the problem of making the model itself explainable, we note that many of the decisions that affect the model's predictive behaviour are made during data preprocessing, and are encoded as specific data transformation steps as part of pre-learning pipelines. Our research explores metrics to quantify the effect of some of these steps. In this initial work we define a simple metric, which we call volatility, to measure the effect of including/excluding a specific step on predictions made by the resulting model. Using training set rebalancing as a concrete example, we report on early experiments on measuring volatility in two public benchmark datasets, Students' Academic Performance and German Credit, with the ultimate goal of identifying predictors for volatility that are independent of the dataset and of the specific preprocessing step.**

## I. Introduction

Since computers are entrusted with making sensible decisions that can have a big impact on people's lives, the issues of *fairness* and *transparency* in how machine learning (ML) algorithms work have become very relevant, and in recent years there's been a growing number of papers about such issues, e.g. [1, 2].

While transparency refers to understanding the way decisions are made, fairness involves having similar individuals obtain similar decision outcomes. Many different formal definitions of fairness have been proposed, see e.g. [3]. Broadly speaking, decisions can be class assignments, scores or ranking positions within a list. These two concepts are highly intertwined, since in order to be able to judge whether a decision was fair or not, understanding the way in which the decision was taken is fundamental.

### A. Research Hypothesis

This work starts from the observation that many of the decisions that affect a model's predictive behaviour are made during data preprocessing, and are implemented as data transformation steps as part of pipelines *prior* to using the data to train the model. We therefore focus not on explaining the model's behaviour, but rather, on techniques for explaining the changes in behaviour that occur when certain steps are added or removed from the processing. In order to do so, we have developed two metrics of these differences in prediction due to preprocessing, which we call *volatilities*.

Our research hypothesis is, then, that the way data is preprocessed for ML processing has noticeable effects, which can be measured and analysed in order to:

- Understand the sensitivity of outcomes of certain datapoints to different preprocessing techniques.
- Detect whether a preprocessing step is increasing or reducing bias for particularly vulnerable population groups.
- Increase the transparency of the whole ML process by analysing any of its specific steps, thus enabling better, fairer models to be created.

At the initial stage of this study we have defined two metrics of volatility, which apply to classification and regression models, respectively. These metrics quantify the difference in predictions for a datapoint with or without a certain preprocessing task applied to a dataset. We make two initial contributions. Firstly, we present a method to systematically compute volatility metrics, and secondly, we show the method in action on two separate use cases. We also suggest that it may be possible to predict which datapoints are more likely to be highly volatile.

### B. State of the Art

The notion of data preprocessing affecting the outcome of a ML task is widely accepted, yet not much work has been done on measuring this impact. One of the first papers to focus on the preprocessing stage of the ML process is [4], a survey on selecting relevant features to represent the data and drive the learning process. In a similar approach to ours, [5] measures the impact of various preprocessing tasks on performance metrics for certain classifiers. Similarly, [6] measures the way a list of particular preprocessing techniques affect the accuracy of document classification, and [7] presents a study in quantifying the impact of preprocessing over several aspects of text classification, such as accuracy, language and domain. More recently, [8] discuss different evaluation measures for feature selection, looking at performing this preprocessing step automatically through supervised, unsupervised and semi-supervised methods.

2375-026X/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDE.2019.00245

The main difference between our work and the papers cited on this section is the focus of research: ours is on which data are most sensitive to preprocessing perturbations, rather than analysing how classifiers' performances are affected by these methods.

## II. Methodology

In our initial exploration, and in the rest of the paper, we focus exclusively on supervised classification models. The *volatility* of a datapoint in a training or test set tells us how prone the datapoint is to changing its predicted label under two different data preprocessing schemes. Using two reference use cases, first we define and show how to calculate volatilities, then we begin to study the problem of predicting their distributions from features of the data, as well as from meta-features such as datapoints being outliers and the probability associated with label prediction.

**Categorical Volatility.** Given two classifiers $\hat{Y}_1$, $\hat{Y}_2$ and the ground truth $Y$, the *categorical volatility* of datapoint $p$ with respect to $\hat{Y}_1$, $\hat{Y}_2$ is given by:

$$CV_{\hat{Y}_1,\hat{Y}_2}(p) \coloneqq \begin{cases} 0 & \text{if } \hat{Y}_1(p) = \hat{Y}_2(p) = Y(p). \\ 1 & \text{if } \exists!\, i \in \{1,\ 2\} : \hat{Y}_i(p) = Y(p). \\ 2 & \text{if } \hat{Y}_1(p) = \hat{Y}_2(p) \neq Y(p). \\ 3 & \text{if } Y(p) \neq \hat{Y}_1(p) \neq \hat{Y}_2(p) \neq Y(p). \end{cases}$$

In plain words, categorical volatility compares two things at once: whether the classifiers are predicting the same thing and whether their predictions are correct with respect to the ground truth.

We have also defined a second—distance based—version of volatility, measuring how far apart two predictions are whenever this makes sense, e.g. when predicting for a numerical attribute, although a distance function may even be defined for non-numerical attributes. This second version, however, isn't *correctness-aware*.

**Numerical Volatility.** Given two predictors $\hat{Y}_1$, $\hat{Y}_2$ of a numerical attribute, the *numerical volatility* of datapoint $p$ with respect to $\hat{Y}_1$, $\hat{Y}_2$ is given by:

$$AV_{\hat{Y}_1,\hat{Y}_2}(p) \coloneqq d\left(\hat{Y}_1(p), \hat{Y}_2(p)\right)$$

for some distance function $d$.

Thus, numerical volatility measures how far apart two predictions for a datapoint are, irrespective of the ground truth $Y$. Building a correctness-aware version of absolute volatility is an interesting future challenge.

Given a dataset $D_{\text{raw}}$, a preprocessing task $T$ and a classifier $C$, the following workflow—shown in figure 1—computes volatility values over $D_{\text{raw}}$.

This workflow uses *leave-one-out* to single out each datapoint $p$ in turn, training a model on the remaining dataset, and using the model to predict a label for $p$. Formally:

1) Perform any basic cleanup and encoding over $D_{\text{raw}}$ to create a baseline dataset $D$.
2) For every[1] datapoint $p \in D$:
   a) Let $D_p = D \setminus \{p\}$ be a training set.
   b) Perform task $T$ on $D_p$, obtaining a second training set $D_p^T = T(D_p)$.
   c) Independently[2] train $\hat{Y}_p$ and $\hat{Y}_p^T$ from $D_p$ and $D_p^T$, respectively.
   d) Predict the labels for $p$, denoted $\hat{Y}_p(p)$ and $\hat{Y}_p^T(p)$, respectively.
   e) From these values, compute $CV_{\hat{Y}_p,\hat{Y}_p^T}(p)$ and $AV_{\hat{Y}_p,\hat{Y}_p^T}(p)$.

Volatility values provide two additional features for each datapoint, which can then be analysed in various ways, including:

- Performing exploratory analysis of the volatilities.
- Looking for correlations with the rest of the features.
- Predicting the volatilities using a transparent algorithm, in order to better understand which features might be more related to volatility.

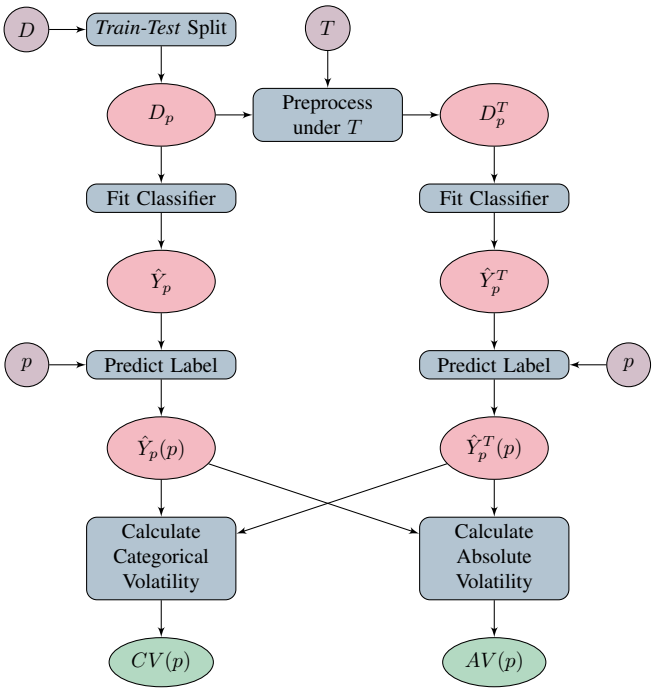


Figure 1: Calculating volatilities of datapoint $p \in D$ flowchart. On the train-test split we require $p \in$ *test*, $D_p =$ *train*.

[1] This step might be computationally expensive for large datasets. In such cases, analysing a test subset would work.

[2] Select the same classification algorithm, but tune each model's hyperparameters separately.

## III. Use Cases

In this section, we go through the workflow described in section II to measure how a preprocessing task can affect certain individuals of the population more than others.

We have analysed two different datasets: *Students' Academic Performance* [9, 10] and *German Credit* [11]. We will refer to these as the *academic* and *credit* datasets, respectively.

The *academic* dataset consists of 480 records of students' information stored in 17 features, 13 categorical and four numerical. The original aim of this dataset is to predict whether a student's markings will be *low, mid* or *high*.

The *credit* dataset includes 1000 records of a bank's clients that requested a credit. Each record has 21 features, including whether the client was *good* or *bad*, i.e. whether they posed a risk of not repaying their loan.

Categorical features were encoded into numbers, so that the *sklearn* functions could properly work. Binary features, as well as those with a natural ordering were encoded as integers starting from 0. Features without a natural ordering were one-hot encoded, as this is the standard way to deal with unordered categorical variables.

For both datasets, we performed balancing twice, separately over two different features: `Label`, and `Gender`. These balancings aimed at improving either the overall accuracy of a classifier (`Label`) or a particular subgroup's classfications (`Gender`).

On both datasets the `Gender` feature is unbalanced, having almost twice as many males (305) as it has females (175) on the *academic* dataset and more than twice as many males (690) as females (310) on the *credit* dataset. This proportion, though, is not representative of a real-life population which should roughly have the same number of males and females. Besides this fact, this feature could be considered *sensitive* for any fairness analysis, due to historical discrimination. Finally, even if a population is unbalanced, training a classifier over an imbalanced dataset can lead to unfair decisions for the minority groups [12]. For these three reasons, we decided to balance both `Gender` and `Label` in separate experiments.

We balanced our data in two different ways. The first—simpler—strategy was to undersample the majority classes through randomly dropping datapoints belonging to them until we have the same number of them as the minority class.

The second—more refined—strategy was to perform a *synthetic minority oversampling technique* (SMOTE) [13], in which synthetic datapoints are created for the minority class trying to follow the dataset's variable distributions as faithfully as possible. Specifically, SMOTE generates synthetic datapoints lying in the line segments between the nearest neighbours of the minority class individuals in the feature space.

Given the small size of our datasets, in order to make the most out of them we decided to implement the *leave-one-out* strategy described next for measuring all three classifiers' accuracies. Once we selected a dataset and a balancing feature, we split our data into $n$ different *train-test* pairs, each one leaving a single datapoint out of the *train* part, e.g. datapoints 2-480 were the training set for datapoint 1 on the *academic* dataset. We call the train parts of these pairs our *base* training sets.

For each *base* set, we performed the two balancing strategies described above (undersampling and SMOTE), resulting in an *under* and a *SMOTE* training sets, respectively. This way, each datapoint got three training sets from which classifiers could be trained to predict their `Label`, obtaining three different predictions. We then trained Random Forests individually for each of these training sets.

We used simple accuracy, defined as the proportion of correctly predicted datapoints to the complete dataset, to compare the performance across classifiers. These results are shown in Table I.

Table I: Accuracies for all three classifiers under different datasets and balancing features.

| | **Academic** | | **Credit** | |
|---|---|---|---|---|
| **Classifier** | **Gender** | **Label** | **Gender** | **Label** |
| Base | 0.81 | 0.81 | 0.77 | 0.76 |
| SMOTE | 0.81 | 0.79 | 0.76 | 0.77 |
| Under | 0.80 | 0.92 | 0.76 | 0.72 |

Having acquired volatility values for all our datapoints, the next logical step to take was to analyse how these new measurements correlated to the rest of the features. On our experiments, we compared the predictions obtained from training a random forest over the *base* unbalanced set and over both the undersampling and SMOTE balanced datasets.

The `Label` attributes we predicted for are categorical, but on the *academic* dataset it has a natural ordering (*Low*, *Mid*, *High*) which when encoded, e.g. (0, 1, 2), allow for evaluating both definitions of volatility. However, in future work we should also experiment over continuous output variables.

Table II displays the distributions for the different categories of volatility we obtained for both balancing strategies. As can be seen, it is much likelier that both classifiers predict the same thing correctly (between $70\%$ and $80\%$ of the time) than them making different predictions, one of them being right (between $5\%$ and $17\%$ of the time). Curiously, volatility 3 (both predictors different and wrong) occurred only once in all of our experiments. This might be due to there being a small number of label categories, hence the probability of having both predictors wrong being relatively low.

Interestingly, whenever two classifiers, $\{$*base, under*$\}$ or $\{$*base, SMOTE*$\}$, predicted different labels for a datapoint, it

Table II: Categorical volatility (CV) distributions for both datasets under both balancing labels and strategies.

| | | Gender Balancing | | Label Balancing | |
|---|---|---|---|---|---|
| | CV | SMOTE | Under | SMOTE | Under |
| Academic | 0 | 0.79 | 0.76 | 0.77 | 0.78 |
| | 1 | 0.05 | 0.09 | 0.06 | 0.17 |
| | 2 | 0.16 | 0.15 | 0.17 | 0.05 |
| | 3 | 0.00 | 0.00 | 0.00 | 0.00 |
| Credit | 0 | 0.74 | 0.72 | 0.73 | 0.61 |
| | 1 | 0.05 | 0.08 | 0.07 | 0.26 |
| | 2 | 0.21 | 0.19 | 0.20 | 0.13 |
| | 3 | 0.00 | 0.00 | 0.00 | 0.00 |

was never the case that the distance between predictions was more than 1 (e.g. *base* predicting 0 and *under* predicting 2 or vice versa). However, it is important to note that this is not necessarily always true. If there were more possible outcomes or they were spread farther appart, absolute volatility would have a wider range of possible values, and in the future we will experiment further with datasets having these characteristics.

As shown in Table III, on the *academic* dataset volatilities vary a lot depending on the `Label` we look at. For both balancing strategies, the probability of having volatility 1 doubles for individuals having *High* `Label` as opposed to individuals with *Low* `Label`. Meanwhile, volatility 2 probabilities are similar for the *Mid* and *High* labels, both being clearly larger than the probability of the *Low* label. Translated into the *academic* dataset's context, this means that *High* marked students are likelier to change classification due to either balancing.

On the other hand, when analysing the volatilities for males and females, we found that `Gender` balancing seems to affect both groups' volatilities in a very similar way. This is particularly interesting, since `Gender` is precisely the feature on which we performed our initial preprocessing, and it would seem natural to expect that females and males get treated differently after preprocessing, which clearly is not the case. Meanwhile, `Label` balancing causes volatility 1 on males to be four times as likely as volatility 1 on females, and volatility 2 appears to happen more often on females than on males. Based on this evidence, it appears that volatility is not a random characteristic of datapoints, but may be correlated to particular features of the data.

We also looked at possible correlations of our engineered features with each other, with both our features of interest (`Gender` and `Label`), as well as with whether a datapoint is an outlier or not, which was determined via the Local Outlier Factor algorithm [14].

Performing a chi-squared test over every datapoint's values, we found a strong ($p < 0.05$) significance on both volatilities being correlated, and a slightly weaker ($p < 0.10$) significance on the datapoint's `Label` being correlated with volatility. Neither `Gender` nor `Outlier` turned out to be significantly correlated with volatility. Full results for our chi-squared tests for SMOTE volatility may be read on table IV.

Table III: Volatility frequencies for undersampling volatility by `Label` and `Gender` under both balancings for the *academic* dataset.

| | Gender Balancing | | | Label Balancing | | |
|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 0 | 1 | 2 |
| **Label** | | | | | | |
| Low | 0.88 | 0.11 | 0.01 | 0.83 | 0.06 | 0.10 |
| Mid | 0.75 | 0.16 | 0.09 | 0.76 | 0.09 | 0.15 |
| High | 0.74 | 0.23 | 0.03 | 0.69 | 0.13 | 0.18 |
| **Gender** | | | | | | |
| Female | 0.77 | 0.15 | 0.07 | 0.78 | 0.03 | 0.18 |
| Male | 0.79 | 0.18 | 0.04 | 0.74 | 0.12 | 0.13 |

Table IV: Chi-squared test $p$ values on possible correlation of SMOTE volatility with Undersampling Volatility, `Gender`, `Label` and `Outlier`. Values smaller than 0.05 indicate a significant correlation.

| | Academic | | Credit | |
|---|---|---|---|---|
| **Feature** | **Gender** | **Label** | **Gender** | **Label** |
| Under Vol. | $< 0.001$ | $< 0.001$ | $< 0.001$ | $< 0.001$ |
| Gender | 0.25 | 0.89 | 0.10 | 0.79 |
| Label | 0.03 | 0.09 | $< 0.001$ | $< 0.001$ |
| Outlier | 0.13 | 0.16 | 0.70 | 0.86 |

## IV. Predicting Volatility

Adopting the *use ML to understand ML* principle, we then trained classifiers to predict volatility. As our training set, we utilised our original datasets with the additional two categorical volatility columns we had created (one for SMOTE, one for undersampling) as our prediction labels. After trying different classifiers, we decided to use decision trees for our predictions, since their accuracy was consistent with other classifiers we tried, and they were far more interpretable.

Having fit the volatility-predicting decision trees, we computed the Gini feature importances for all classifiers. These features were considered then for performing additional analyses on them. Table V presents the five most important features for each decision tree on the *academic* dataset. Interestingly, the lists agree on several features, with some of them appearing on every single list, e.g. `SaudiArabia`. Again, this suggests that this particular subgroup of the population may be more sensitive to data balancing than other subgroups. Interestingly too, `Label` appears on three out of the four lists as the most important feature, both when doing `Gender` and `Label` balancing.

From what we have seen in our case studies, we can say that even though we didn't find a unique cause for

Table V: Top five Gini importances for volatility predictions for both balancing features and both balancing strategies.

| | SMOTE | | Undersampling | |
|---|---|---|---|---|
| | **Feature** | **Score** | **Feature** | **Score** |
| **Gender Balancing** | Label | 0.19 | Label | 0.22 |
| | RaisedHands | 0.18 | RaisedHands | 0.20 |
| | SaudiArabia | 0.14 | SaudiArabia | 0.11 |
| | AnnouncementsView | 0.13 | VisitedResources | 0.11 |
| | English | 0.09 | Kuwait | 0.07 |
| **Label Balancing** | Label | 0.21 | VisitedResources | 0.33 |
| | SaudiArabia | 0.13 | RaisedHands | 0.18 |
| | Kuwait | 0.10 | SaudiArabia | 0.09 |
| | VisitedResources | 0.09 | section_B | 0.08 |
| | RaisedHands | 0.08 | Kuwait | 0.07 |

volatility, and some expected results, such as `Gender` being a major factor in explaining volatility didn't turn out to be true, calculating it turned out to be very helpful in finding particularities of our data, such as the `SaudiArabia = 1` group being so different from the rest in terms of the rest of the features.

The fact that both balancing techniques produced similar volatilities, even though they are very different in nature (one randomly deleting datapoints before fitting, the other generating synthetic ones) is also very interesting. As expected, volatilities turned out to be correlated significantly with each other, and their agreement rate across the `Label` categories was substantial.

## V. Conclusion

For these particular cases, we couldn't make much use of numerical volatility, because its outcomes were redundant with the ones in categorical volatility.

The approach of using a second classifier to explain the volatility of the first one seems to work quite well. Specifically, the usage of decision trees allowed us to detect which features were worth inspecting with greater detail.

In short, we believe that volatility can be part of a basic toolbox of helper methods which serve the double purpose of making better, fairer predictions, as well as understanding the implications of data preprocessing into predictions, thus making a more transparent overall process.

Our procedure being data and model agnostic, we intend to keep studying volatility over different datasets and preprocessing tasks to look for common trends. Having already looked at how the analysed datasets react to balancing, we may as well try other preprocessing approaches such as normalisation of numerical features and encodings or feature subsetting.

We also suspect there might be a correlation between volatility and the certainty a classifier has on its predictions, e.g. the more certain it is about a specific datapoint's label, the less volatile such datapoint should be.

It will be particularly important to try out larger datasets for both classification and regression problems, so that we may also analyse the behaviour of absolute volatility. We will also investigate on ways to reduce both classifier accuracy and closeness to a second classifier into a single number.

Finally, we wish to develop a framework—in the form of *Jupyter* notebooks—that generalises and automates our workflow, and make it available to the general audience, so that volatility may be obtained by the users given any classifier, dataset and predicting label of choice. This would follow the trend of IBM's recently released *AI Fairness 360* toolkit, which allows anyone to perform fairness and transparency analysis over any dataset and classifier.

## References


[1] A. Chouldechova, "Fair prediction with disparate impact: A study of bias in recidivism prediction instruments," *Big data*, vol. 5, no. 2, pp. 153–163, 2017.

[2] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?: Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 2016, pp. 1135–1144.

[3] M. J. Kusner, J. Loftus, C. Russell, and R. Silva, "Counterfactual Fairness," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4066–4076.

[4] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial Intelligence*, no. 97, pp. 245–271, 1997.

[5] S. F. Crone, S. Lessmann, and R. Stahlbock, "The impact of preprocessing on data mining: An evaluation of classifier sensitivity in direct marketing," *European Journal of Operational Research*, vol. 173, no. 3, pp. 781–800, 2006.

[6] C. A. Gonçalves, C. T. Gonçalves, R. Camacho, and E. C. Oliveira, "The impact of Pre-Processing on the Classification of MEDLINE Documents," *Pattern Recognition in Information Systems, Proceedings of the 10th International Workshop on Pattern Recognition in Information Systems, PRIS 2010, In conjunction with ICEIS 2010*, p. 10, 2010.

[7] A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification," *Information Processing and Management*, vol. 50, no. 1, pp. 104–112, 2014.

[8] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, 2018.

[9] E. A. Amrieh, T. Hamtini, and I. Aljarah, "Preprocessing and analyzing educational data set using x-api for improving student's performance," in *Applied Electrical Engineering and Computing Technologies (AEECT), 2015 IEEE Jordan Conference on*. IEEE, 2015, pp. 1–5.

[10] ——, "Mining educational data to predict student's academic performance using ensemble methods," *International Journal of Database Theory and Application*, vol. 9, no. 8, pp. 119–136, 2016.

[11] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[12] S. Hido, H. Kashima, and Y. Takahashi, "Roughly balanced bagging for imbalanced data," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 2, no. 5-6, pp. 412–426, 2009.

[13] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[14] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying Density-Based Local Outliers," *SIGMOD '00 Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104, 2000.