# CONOUT: CONtextual OUTlier Detection with Multiple Contexts: Application to Ad Fraud

Meghanath M Y[1] Deepak Pai[2] Leman Akoglu[1]

[1] Carnegie Mellon University
Heinz College of Information Systems and Public Policy
{meghanam, lakoglu}@andrew.cmu.edu
[2] Adobe {dpai}@adobe.com

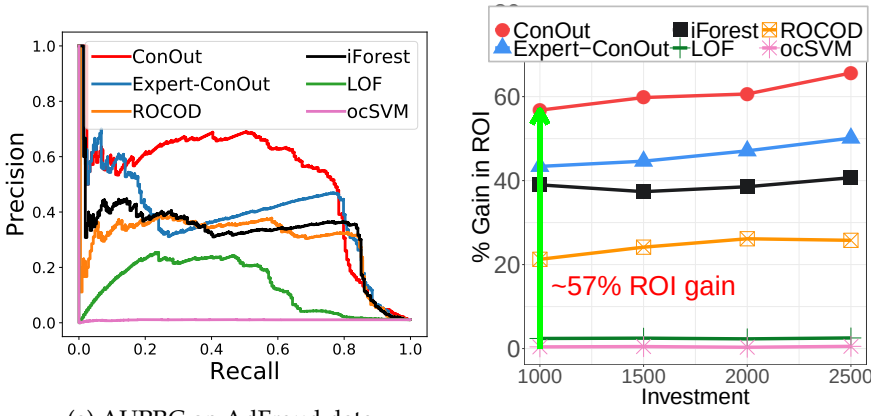
**Abstract.** Outlier detection has numerous applications in different domains. A family of techniques, called contextual outlier detectors, are based on a *single*, *user-specified* demarcation of data attributes into indicators and contexts. In this work, we propose CONOUT, a new contextual outlier detection technique that leverages *multiple* contexts that are *automatically* identified. Importantly, CONOUT is a *one-click* algorithm—it does not require any user-specified (hyper)parameters. Through experiments on various real-world data sets, we show that CONOUT outperforms existing baselines in detection accuracy. Further, we motivate and apply CONOUT to the advertisement domain to identify fraudulent publishers, where CONOUT not only improves detection but also provides statistically significant revenue gains to advertisers: a minimum of 57% compared to a naïve fraud detector; and ∼20% in revenue gains as well as ∼34% in mean average precision compared to its nearest competitor.

## 1 Introduction

Outlier detection is a fundamental data mining task and has important applications in medicine, finance and advertisement industry [1]. A family of methods in the outlier detection literature focuses on context based detection. A contextual outlier is defined as an instance whose behavior deviates markedly from instances that share similar contexts. The contextual outlier detection (COD) techniques are aimed to incorporate two main ideas. First, they avoid assigning a higher outlier score to instances that stand out in attributes that are not directly indicative of outlierness, called *contextual attributes*. Second, they aim to tease out the instances, whose behavior, defined by *indicator attributes*, deviates markedly only in sub-populations identified by similar contexts. This demarcation between the two types of attributes, namely contextual and indicator, has been recently shown to improve the detection performance of outlier detection techniques in various domains [9,13,17], and is an active area of interest.

**Motivating Application:** To exemplify the key ideas of COD, consider the domain of advertisement fraud where one of the primary goals is to identify publishers[3] that illicitly generate fake eyeballs to increase their revenue by a variety

[3] A publisher is an entity that provides real estate on their website to host ads.




(a) AUPRC on AdFraud data

(b) Gain in ROI for advertisers

Fig. 1: Proposed CONOUT achieves significant improvements in *both* (a) detection & (b) revenue gains compared to existing techniques. (See §3.1 for details)

of schemes. Advertisement fraud is well documented to have multiple mechanisms that generate these fake eyeballs [4]. Outlier detection has been used to detect fraudulent publishers in the advertisement industry (refer [14] for a survey). Publishers can be characterized by a long list of features[4] derived from ad request data as well as external sources (See e.g., Table 2). For ease of illustration, let us consider three attributes - average clicks on ads (`clicks`), average number of ad impressions served (`impressions`) and host country (`country`) of the publisher to identify fraudulent publishers.

*Why COD ? :* Which `country` a publisher belongs to can not be a direct indication of fraudulent activity. However, `country` could be used to infer `clicks` and `impressions`, hence could be used as a contextual attribute. Traditional detection techniques that consider `country` equally important to the other two variables would be adding additional noise to the model. On the other hand, if one were to not consider `country` altogether, the auxiliary information provided by considering the sub-populations created by `country` would not be incorporated. Hence, one would expect a model similar to [9,13], which treats `country` as a context, and `clicks` and `impressions` as indicators to perform better. However, in practice, given a long list of publisher features and the complex generating mechanism of ad fraud, the assumption of a *single*, *user-specified* demarcation of attributes in the existing COD techniques is questionable.

*Why multiple contexts ? :* Viewing `clicks` as an indicator attribute is justified since fraudulent publishers are expected to have higher clicks compared to other publishers. However, to avoid detection, illegitimate publishers often employ schemes to mimic legitimate publishers and camouflage their clicks, while feeding off the long tail impression revenue [4]. In such a scheme, `clicks` may no longer be indicative of fraudulent behavior directly but would serve as a context to assess the deviation in `impressions` served. This ambiguity of view-

[4] Throughout the paper, attributes and features are used interchangeably.

ing `clicks` as a context or an indicator attribute further debates the assumption of a *single*, *user-specified* demarcation and motivates the need for *multiple* contexts that current COD approaches do not explore. To better detect fraudulent publishers, an approach that incorporates both the contexts is required. Also, given the uncertainty of the role of different attributes, a data-driven identification of contexts and indicator attributes would be needed.

To address current limitations, we introduce a novel COD technique called CONOUT that does not rely on a pre-specified context, rather automatically identifies and incorporates multiple contexts. Our work is motivated and applied to the publisher fraud detection problem in the ad domain. It outperforms a list of existing techniques, including one with domain expert-specified context, in identifying fraudulent publishers (Figure 1a). CONOUT also achieves statistically significant revenue gains when compared to its competitors. In particular, CONOUT provides more than 57% gains (Figure 1b) in terms of return on investment (ROI) to the advertiser when compared to a naïve fraud detector and ∼20% gains when compared to its nearest competitor. We summarize our notable contributions as follows.

- **Automatic context formation:** To identify contexts, we develop a unified measure grounded with concepts of statistical hypothesis tests to capture dependence between the attributes. The measure can handle mixed (type) attibutes and quantifies the similarity of sub-populations generated by attributes which we leverage to automate the context formation.
- **Context-incorporated detection algorithm:** CONOUT quantifies outlierness of an instance with reference to its sub-population specified by a given context. Rather than training a separate detector for each sub-population in a given context, we train a *single* density-based outlier detector and introduce a scheme for re-weighing neighbors of a given test instance (for density estimation) by their distance in the contextual space.
- **Incorporating multiple contexts:** CONOUT searches through multiple contexts to spot one in which a point deviates the most in its indicator attributes. In essence, CONOUT is an ensemble over contexts.
- **Parameter-free nature:** CONOUT relies on one parameter (a kernel bandwidth) to be specified. We introduce an unsupervised model selection procedure for tuning the parameter, as such, CONOUT requires no user input.
- **Application to ad fraud:** We provide an in depth case study of CONOUT in the ad domain. To showcase the advantage of deploying CONOUT versus competing detectors in making ad-placement decisions, we develop a cost benefit framework to assess the ROI (a metric more relevant to this domain than detection accuracy) gained by an advertiser.

**Reproducibility:** Implementation of CONOUT and public data used in experiments are open-sourced at `https://cmuconout.github.io/`

## 2 CONOUT for Outlier Detection with Multiple Contexts

**Notation** Consider an input data set $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ containing $n$ points in $d$ dimensions, where $\mathcal{F}$ denotes the feature set. A context $\mathcal{C}_p$ with $p$ attributes,

referred to as *contextual attributes*, is a subset of $\mathcal{F}$ and the corresponding *indicator attributes* are denoted by $\mathcal{I}_p = \mathcal{F} \setminus \mathcal{C}_p$, where $|\mathcal{C}_p| = p$, $|\mathcal{I}_p| = d - p$.

**Definition 1 (Sub-population).** *Given a context $\mathcal{C}_{p_k}$ and a point $\mathbf{x}_i$, its sub-population consists of objects similar to $\mathbf{x}_i$ in the context space, i.e. w.r.t. features in $\mathcal{C}_{p_k}$. If $\mathcal{C}_{p_k}$ is categorical, all points with the same categorical value as $\mathbf{x}_i$ belong to its sub-population. For a numerical context, a distance measure capturing the similarity in the context space is used to specify a sub-population.*

Our aim is to find $\boldsymbol{s} = \{s_1, s_2, ..., s_n\}$, where $s_i$ is the outlier score of instance $\mathbf{x}_i$ in $\mathcal{D}$ by automatically finding and incorporating multiple contexts. We break down the task into two sub problems.

*Problem 1 (Automatic Context Formation).* **Given** a dataset $\mathcal{D} \in \mathbb{R}^{n \times d}$, with feature set $\mathcal{F}$; **Find** a set of contexts $\mathcal{C} = \{\mathcal{C}_{p_1}, \mathcal{C}_{p_2}, \ldots, \mathcal{C}_{p_K}\}$ such that each $\mathcal{C}_{p_k}$ would act as a suitable frame of reference for set of indicator attributes $\mathcal{I} = \{\mathcal{I}_{p_1}, \mathcal{I}_{p_2}, \ldots, \mathcal{I}_{p_K}\}$, where $\mathcal{I}_{p_k} = \mathcal{F} \setminus \mathcal{C}_{p_k}$. (Details in §2.1)

*Problem 2 (Context-incorporated Outlier Detection (COD)).* **Given** a set of contexts $\mathcal{C}$ and corresponding indicator attributes $\mathcal{I}$; **Find** the outlier scores $\boldsymbol{s} = \{s_1, s_2, \ldots, s_n\}$ that incorporate the $K$ contexts in $\mathcal{C}$ such that $s_i$ is representative of the deviation of $\mathbf{x}_i$ in each $\mathcal{I}_{p_k}$ that share similar $\mathcal{C}_{p_k}$. (Details in §2.2)

Next we introduce CONOUT that addresses both problems stated above.

### 2.1 Automatic Context Formation

The underlying assumption of contextual detection is that objects sharing similar contextual attributes are expected to have similar indicator attributes [9]. The objects that share similar contextual attributes can be viewed as sub-populations of the whole data. For instance, earlier we used `country` (context) to create these sub-populations to assess `clicks` and `impressions` (indicators). These sub-populations are expected to have similar behavior and deviation from this behavior would indicate a contextual outlier. Assuming that the contexts are unknown and multiple, a naive way of forming contexts would be to consider all the subsets of the feature set $d$ resulting in $(2^d - 1)$ contexts.[5] However, this is computationally infeasible in high dimensions.

Alternatively, since the aim of contexts is to identify sub-populations, one could group the attributes which would result in similar sub-populations. Intuitively, a pair of highly dependent attributes would result in similar sub-populations. For example, two numerical features that have a similar rank ordering of instances would produce similar sub-populations when binned. As such, a measure of rank correlation can be used to capture dependence. However, many practical datasets often consist of *both* numerical and categorical attributes. To effectively handle *mixed* attributes, i.e. capture dependence between attribute pairs of mixed type, we develop a *unified* measure by leveraging statistical tests to quantify dependence between two samples. We then use

[5] We would need to omit the set which contains all the $d$ features since there would be no indicator attributes left to assess the deviation.

the measure to group the attributes into context groups that result in similar sub-populations. In particular, we set up hypothesis tests to handle combinations of categorical and numerical attributes, where the $p$ value ($p$-val) of the test would signify the dependence between a given pair of features. Depending on the types of the attribute pair, we calculate the dependence using the following tests.

**Numerical-Numerical:** For a pair of numerical attributes, we use the non-parametric Spearman's rank correlation statistic that operates on the rank orderings of the two numerical features. Let us denote by $\mathbf{v}_f$ and $\mathbf{v}_{f'}$ two vectors with values for $n$ points in $\mathcal{D}$ of two arbitrary numerical features $f$ and $f'$ in $\mathcal{F}$. The test statistic $\rho$ is given by,

$$\rho = \frac{cov(\mathbf{r}_f, \mathbf{r}_{f'})}{\sigma_{\mathbf{r}_f} \sigma_{\mathbf{r}_{f'}}} \tag{1}$$

where $\mathbf{r}_f$ and $\mathbf{r}_{f'}$ correspond to vectors that hold the indices of points when ranked by values in $\mathbf{v}_f$ and $\mathbf{v}_{f'}$. Spearman's rank correlation assesses how well the relationship between two features can be described using a monotonic function. The test is widely used to quantify the dependence between both ordinal and continuous features. To determine the significance of the test, we employ a permutation test [8]. The permutation test works by randomly reshuffling the observed data to generate multiple samples. For each such permuted sample, the test statistic is computed. If one were to generate $B$ such permuted samples, the (null) distribution of the corresponding test statistics under no dependence could be used to obtain the $p$-val of the observed sample.[6]

**Numerical-Categorical:** For a pair of numerical and categorical features, we use the Kruskal-Wallis non-parameteric test statistic that operates on the rank orderings of the numerical feature within each group of the categorical feature. The test statistic $H$ is given by,

$$H = \frac{\sum_{c=1}^{a} n_c (\bar{r_c} - \bar{r})^2}{\sum_{c=1}^{a} \sum_{i=1}^{n_c} (r_{ic} - \bar{r})^2} \tag{2}$$

where $a$ is the arity of the categorical feature, $n_c$ is the number of data points of category $c$, $r_{ic}$ is the rank of observation $i$ of category $c$ with respect to the numerical feature, $\bar{r_c}$ is the average rank of all observations in category $c$, $\bar{r}$ is the average rank of all the observations. Under the null hypothesis of independence of the two samples, the Kruskal-Wallis statistic asymptotically follows a Chi-squared distribution with $(a-1)$ degrees of freedom, which is used to determine the significance of the test statistic. To account for the possibility of ties, we perform a permutation test as before.

**Categorical-Categorical:** For a pair of categorical features, we use the Chi-square statistic which quantifies the differences in the observed and expected frequency distribution of the two features. The statistic $\chi^2$ is given by,

$$\chi^2 = \sum_{c=1}^{a} \sum_{c'=1}^{a'} \frac{(O_{cc'} - E_{cc'})^2}{E_{cc'}} \tag{3}$$

[6] Typically a small set of random permutations of the observed data, $B$ = 400 [8] is sufficient to generate a reliable significance value.

where $a, a'$ are the arities of the two categorical features, $O_{cc'}$ and $E_{cc'}$ denote the observed and expected number of instances of type $c, c'$. Chi-square tests are widely used as a test of independence to assess whether unpaired observations on two samples, expressed in a contingency table, are independent of each other. The test statistic follows a Chi-squared distribution with $(a-1)(a'-1)$ degrees of freedom and is used to assess the significance of the test. To address the possibility of very few observations in a cell $O_{cc'}$, which could lead to inaccurate inference, we employ a permutation test to compute the $p$-val.

**Unified Measure:** The $p$-val of each of the tests listed above signifies the dependence between the pair of features. If a pair of features have a low $p$-val, one could reliably reject the null hypothesis of the two features being independent. Hence, to capture the dependence of any two features, we use $1-p$-val $\in [0,1]$ as our unified measure of dependence.

**Forming context groups:** Next, we perform a clustering of the features into context groups based on the unified (dependence) measure, using an algorithm that automatically decides the number of context groups, denoted by $G$, $G \leq d$. This is achieved either by employing X-means [11] or by performing a hierarchical clustering using the gap statistic [7] to automatically estimate the number of clusters. By construction, features in each context group are highly dependent and would result in similar sub-populations. Therefore, considering all $(2^G-1)$ combinations of the context groups would provide a sufficient and computationally efficient proxy to the naïve way of forming contexts based on all possible $(2^d-1)$ combinations of the original features.

This completes the formation of set of contexts $\mathcal{C} = \{\mathcal{C}_{p_1}, \mathcal{C}_{p_2}, \ldots, \mathcal{C}_{p_K}\}$, $K = (2^G-1)$, as stated in Problem 1. Next, we introduce our proposed detection algorithm that incorporates all these $K$ contexts.

### 2.2 Context-incorporated Outlier Detection

After generating multiple contexts, we aim to use them in assigning an outlier score $s_i$ to each instance $\mathbf{x}_i \in \mathcal{D}$. Given $(\mathcal{C}_{p_k}, \mathcal{I}_{p_k})$, a single (context, indicators) tuple, one could cluster the instances in the context space, to find sub-populations in the context and use a traditional outlier detection technique to assign scores to each sub-population based on the corresponding indicator attributes. The same heuristic could be extended to multiple contexts. There are two potential problems with this approach. First, clustering the data instances would involve non-trivial choices of the clustering algorithm (depending on the distribution of the data) and number of clusters.[7] Second, to assign the outlier scores, one would need to learn multiple models (corresponding to each sub-population in a context) which may be computationally expensive.

We address these issues by learning only a *single* model per context using the indicator attributes and weigh the scores of the model based on the similarity in the corresponding contextual attributes, completely avoiding the need to cluster the data instances into sub-populations. To achieve this, we propose a

[7] Note that, earlier we perform clustering in features rather than on data instances with a carefully constructed unified measure and are less prone to the issues mentioned.

modification to the outlier scoring mechanism of isolation forest [10] (iForest), a popular tree based outlier detection technique, which we briefly review next.

**iForest overview:** The core idea of iForest is to isolate a data point by recursively partitioning the feature space into random intervals. The recursive partitioning can be visualized as a binary search tree where the path of a tree from root to leaf node is a conjunction of multiple random feature splits. Broadly, iForest comprises of two phases - training and testing. In the training phase, multiple trees (denoted by $t$) are built by sampling a set of points (denoted by $\psi$) for building each tree. In the testing phase, an object traverses through trees built earlier until it reaches a leaf node. Intuitively, training instances in the same leaf with a point are its near-neighbors in the subspace specified by split-features from root to leaf (see Figure 2a for e.g. leaf node in an iTree), where the count of such neighbors serves as a crude estimate of density (the lower the count of such near-neighbors, the more likely that the point is an outlier). iForest outlier score for an instance is given by,

$$o(\mathbf{x}_i, \psi) = 2^{-\frac{E(h(\mathbf{x}_i))}{c(\psi)}} \tag{4}$$

where $h(\mathbf{x}_i)$ is a function of number of training instances in the leaf node to which $\mathbf{x}_i$ has traversed to, $E(h(\mathbf{x}_i))$ is the average of $h(\mathbf{x}_i)$ over multiple trees and $c(\psi)$ is a normalizing constant dependent on sample size $\psi$. A higher value of $o(\mathbf{x}_i, \psi)$ would indicate a higher chance of the instance being an outlier.

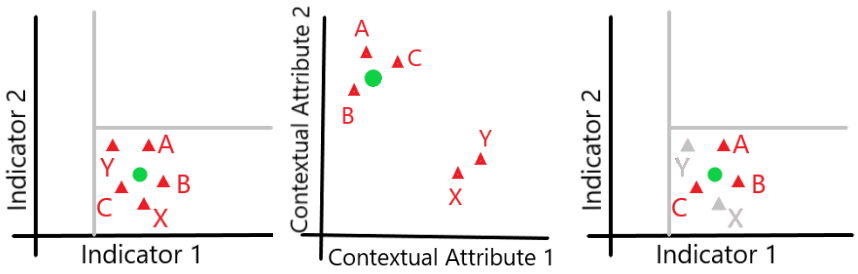
Next, we present how we modify Eq. 4 to quantify contextual outlierness of a point in CONOUT. To avoid cluttered notations, we consider a single context and later explain how to combine the scores from all $K$ contexts.

**Contextual weighing:** To incorporate a context into Eq. 4, we should account for the similarity of test instance and the tree neighbors in the contextual space. Intuitively, a neighbor that is dissimilar to the test instance in the contextual space should contribute less (to density) compared to a similar neighbor that belongs to the same sub-population. Formally, since $h(\mathbf{x}_i)$ is determined based on the leaf/neighbor counts, for a test instance $\mathbf{x}_i$ and a neighbor $\mathbf{x}_j$ that shares the same leaf in a tree, we need a smooth function $\phi(\mathbf{x}_i, \mathbf{x}_j)$ that returns 1 when $\mathbf{x}_i$ and $\mathbf{x}_j$ are identical and approaches 0 for the more dissimilar $\mathbf{x}_i$ and $\mathbf{x}_j$. We note that the radial basis function suits this purpose, given by,

$$\phi_\gamma(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma||\mathbf{x}_i - \mathbf{x}_j||) \in [0, 1] \,, \tag{5}$$

where $||\mathbf{x}_i - \mathbf{x}_j||$ denotes the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ in the contextual space, and $\gamma$ is the kernel bandwidth; a free parameter that controls the radius of influence. A higher $\gamma$ would impose a higher penalty to the contribution of the points farther in the context space, while a lower $\gamma$ would impose a relatively lesser penalty. $\gamma$ can also be seen as a parameter that controls the influence of the context on the indicators. When $\gamma = 0$, the influence of context vanishes and the modified outlier score would be equivalent to the score of an iForest solely based on the indicators. Then, the modified score of a point is written as

$$o(\mathbf{x}_i, \psi, \phi_\gamma) = 2^{-\frac{E(h(\mathbf{x}_i, \phi_\gamma))}{c(\psi)}} \tag{6}$$




(a) A tree in the iForest (b) Context space (c) Incorporating context

Fig. 2: **Toy Example:** To exemplify our *contextual weighing* scheme, consider the above example setup. In (a), we visualize the neighbors (in red) of the test instance (in green) for a single tree of iForest created by random splits on the indicator attributes. All the neighbors (A,B,C,X,Y) in the same leaf contribute equally to the outlier score of test instance irrespective of the alignment of the points in the context space (b). For the test instance to be a contextual outlier, points closer to it in context space (A,B,C) should contribute more, while points farther (X,Y) should contribute less (to density estimation). This is achieved using Eq. 5; weighing down the points farther from the test instance in context space (b) (X,Y greyed out in (c)) more than the points closer (A,B,C).

where this time $h(\mathbf{x}_i, \phi_\gamma)$ is a function of the *weighted* number of training instances in the same leaf as $\mathbf{x}_i$, where weights are obtained by Eq. 5.

In short, for a context $\mathcal{C}_{p_k}$, we train an iForest using the indicator attributes $\mathcal{I}_{p_k} = \mathcal{F} \setminus \mathcal{C}_{p_k}$. During the testing phase, for a point $\mathbf{x}_i$, we find the leaf it traverses to on each tree, use Eq. 5 to reweigh the points in each such leaf to estimate a weighted average count of neighbors $h(\mathbf{x}_i, \phi_\gamma)$. Then, we use Eq. 6 to assign an outlier score. We show an illustration of the idea in Figure 2.

We remark that the proposed modification could be integrated into other outlier detection techniques that are based on near neighbors of a test instance in arbitrary subspaces. For example, the outlier score of Half Space Trees [15] could also be modified in the stated fashion.

**Distance in the context space:** The distance $||\mathbf{x}_i - \mathbf{x}_j||$, input to the Eq. 5 should capture the similarity in the context space of the instance $\mathbf{x}_i$ to its neighbors in a given tree. Recall from §2.1 that a context could consist of attributes of mixed type. Here, we specify the distance computation in such a space.

For the attributes that are categorical in the context space, if $\mathbf{x}_i$ and $\mathbf{x}_j$ do not share the same value in *any* categorical attribute, then they do not belong to the same sub-population (for instance if they do not belong to the same `country`) and hence $\mathbf{x}_j$ should have no contribution to the density estimation at $\mathbf{x}_i$, that is, $\phi_\gamma(\mathbf{x}_i, \mathbf{x}_j) = 0$, irrespective of the numerical attributes. If $\mathbf{x}_i$ and $\mathbf{x}_j$ share the same value in *all* the categorical attributes, then we use the normalized Euclidean distance on the numerical attributes.

**Combining scores from multiple contexts:** In the testing phase, we compute the outlier score using Eq. 6, for all the $K$ contexts in $\mathcal{C}$ with their corresponding indicator attributes $\mathcal{I}$. This results in $K$ scores for each instance

$\mathbf{x}_i$. Since a larger score indicates a higher chance of being an outlier, we use the maximum of the scores across all contexts as our assembling scheme, i.e., $s_i = \max_{k\in\{1,\ldots,K\}} o_k(\mathbf{x}_i, \psi, \phi_\gamma)$. Taking the maximum achieves the purpose of teasing out a potential outlier which stands out in a particular context but is hidden in the rest of the contexts.

**Choosing $\gamma$ :** Given the importance of $\gamma$ as a parameter to control the influence of a context, we vary $\gamma$ on a logarithmic grid between $10^{-3}$ to $10^3$ and employ an unsupervised model selection approach leveraging [6]. The idea is to convert the outlier scores for a certain $\gamma$ into calibrated probabilities assuming the posterior probabilities follow a logistic sigmoid function, $\sigma(s_i) = 1/(1+\exp(-(w_0 + w_1 s_i)))$. The parameters of the sigmoid function are estimated from the scores. This allows us to use the goodness of fit criterion across different models (corresponding to various $\gamma$) to choose the $\gamma$ that corresponds to the most calibrated probabilities. Formally, given $\boldsymbol{s}$, the outlier scores of the $n$ instances obtained using a certain $\gamma$, let us denote by $\boldsymbol{\ell}$ a binary *latent* vector corresponding to the unobserved labels of the $n$ instances, which takes the value $\ell_i = 1$ if the instance is an outlier and 0 otherwise. Then, the negative log likelihood function can be written as

$$LL(\boldsymbol{\ell}|\boldsymbol{s}) = \sum_{i=1}^{n} \left[\log(1+\exp\left(-w_0 - w_1 s_i\right)) + (1-\ell_i)(w_0 + w_1 s_i)\right] \qquad (7)$$

**EM initialization:** The model parameters could be estimated by employing the EM algorithm [5] that simultaneously estimates the unobserved labels $\ell_i$'s, and parameters $w_0$, $w_1$ of the sigmoid function. However, EM provides only a locally optimum solution for latent variable functions like Eq. 7, the quality of which strongly depends on the initialization. Thanks to the ranking provided by the scores $\boldsymbol{s}$, we can heuristically initialize our latent variable vector $\boldsymbol{\ell}$ where for a given threshold, the instances having a score higher than the threshold are initially labeled as outliers and the rest as inliers. This is a more informed initialization compared to a random one. Since we do not know the exact threshold, we try a few random thresholds in decreasing order of the scores, run EM multiple times and pick the solution that yields the highest likelihood.

For each $\gamma$, the maximum likelihood based on the fitted parameters is noted. We then choose the $\gamma$ that corresponds to the maximum likelihood of the different models. Since the number of parameters learnt in all the models is the same, this is equivalent to using a goodness of fit measure such as AIC/BIC. Note that the proposed unsupervised model selection criterion could be used for other outlier detection techniques that require user-specified (hyper)parameters. For example, the number of nearest neighbors $k$ required by most distance or density based methods (like LOF [3]) could be tuned in a similar fashion.

This completes the estimation of outlier scores $\boldsymbol{s}$, as stated in Problem 2, that incorporates the $K$ contexts generated from §2.1.

**Summary:** We conclude by outlining the detailed steps of CONOUT in Algo. 1. A high level abstraction of the CONOUT can be described as follows:

**Algorithm 1** CONOUT for Contextual Outlier Detection with Multiple Contexts

**Input:** unlabeled dataset $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ (no user-specified parameters)
**Output:** the outlier scores of $n$ instances $\boldsymbol{s} = \{s_1, s_2, \ldots, s_n\}$

Use $(1 - p\text{-val})$ based on appropriate test statistics in Eq.s 1, 2, 3 for clustering the features into context groups ▷ **Automatic Context Formation**
Generate contexts $\mathcal{C} = \{\mathcal{C}_{p_1}, \mathcal{C}_{p_2}, \ldots, \mathcal{C}_{p_K}\}$, $K = 2^G - 1$ by considering all possible combinations (except the full set) of the $G$ context groups
**for** each $\gamma$ in the logarithmic grid $10^{-3}$ to $10^3$ **do** ▷ **Model Selection**
  **for** each context $\mathcal{C}_{p_k}$ in $\mathcal{C}$ **do** ▷ **Context- incorporated Outlier Detection**
    Train an iForest using $\mathcal{I}_{p_k} = \mathcal{F} \setminus \mathcal{C}_{p_k}$
    **for** each instance $\mathbf{x}_i$ in $\mathcal{D}$ **do**
      Compute score $o_k(\mathbf{x}_i, \psi, \phi_\gamma)$ using the modified outlier score in Eq. 6
  **for** each instance $\mathbf{x}_i$ in $\mathcal{D}$ **do**
    Assign $s_i^\gamma = \max_{k \in \{1,\ldots,K\}} \; o_k(\mathbf{x}_i, \psi, \phi_\gamma)$, to form $\boldsymbol{s}^\gamma$
  Estimate the parameters in Eq. 7 via EM using $\boldsymbol{s}^\gamma$
**Return** the scores $\boldsymbol{s}^\gamma$ based on the $\gamma$ corresponding to the maximum loglikelihood

1. **Context Groups:** Use unified dependence measure $(1 - p\text{-val})$ to cluster the features into $G$ context groups, where $G$ is automatically chosen [11].
2. **Context Formation:** Form $K = 2^G - 1$ contexts by considering all possible combinations of the $G$ context groups.
3. **Training Detectors:** For each context $\mathcal{C}_{p_k}$, $k = 1, \ldots, K$, train an iForest on corresponding indicator attributes $\mathcal{I}_{p_k} = \mathcal{F} \setminus \mathcal{C}_{p_k}$.
4. **Outlier Scoring:** For each point $\mathbf{x}_i$ in $\mathcal{D}$, assign the outlier score to be the maximum across the $K$ contexts to form $\boldsymbol{s}$.

**One-click algorithm:** As evident from Algo. 1, CONOUT does not require the user to specify any input values other than feeding in the data set $\mathcal{D}$, which makes CONOUT a one-click algorithm that runs *parameter free*. We carefully choose $\gamma$, the only hyper parameter (kernel bandwidth) within CONOUT using the probabilistic unsupervised model selection scheme.

**Complexity analysis** To conclude, we analyze the time complexity of each of the above steps. In Step 1, we cluster the features based on dependence, quantified by the unified measure $(1 - p\text{-val})$ which takes $O(n)$ for a given pair.[8] We use the scalable X-means algorithm [11], which extends k-means to automatically find the number of clusters. Initially, we randomly pick $G$ features as the centroids. In the assignment step, we compute dependence of each feature on the $G$ designated centers, which is $O(ndG)$. In re-centering, we need to identify, within each current cluster, the feature with the largest total dependence to others in the cluster to be designated as the new center (note that we can *not* re-center by *averaging* the columns, as we work with mixed attributes). To this end, we randomly sample a few points and pick among those the one with the largest total dependence. This takes $O(nd)$ for all clusters. Overall complexity of clustering is $O(ndG)$ where $G$ is small. Having identified $G$ constant number of clusters, Step 2 takes $O(K)$ for constructing $K = 2^G - 1$ contexts.

[8] Since the number of simulations $B$ is constant, we omit this in the complexity.

In step 3, training an iForest depends on the number of trees $t$ and sample size $\psi$ which is $O(t\psi \log \psi)$, where depth of each tree is $O(\log \psi)$. We train an iForest for each context, resulting in $O(Kt\psi \log \psi)$. Finally, assigning an outlier score to a point involves finding its leaf (in $O(\log \psi)$) in each tree, and calculating its distance (in the given context space, in $O(d)$) to the points sharing the same leaf (smaller than $\psi$ points in each leaf). This takes $O(Knt(\log \psi + d\psi))$ for $n$ points and $K$ contexts across $t$ trees. Overall complexity of steps 1–4 is $O(Kt[(n + \psi) \log \psi + nd\psi])$. Since $t$ and $\psi$ are constants, the complexity is linear in data size $n$ and $d$ as well as the number of generated contexts $K$.

## 3 Experiments

**Datasets** In this section, we empirically evaluate the efficacy of CONOUT. To this end, we provide an in depth case study of the practical applicability of CONOUT in ad domain by comparing it to multiple baselines. In addition, we also run CONOUT on multiple publicly available data sets from ODDS[9] and UCI repositories. A summary of the data sets ordered wrt outlier % is reported in Table 1. We consider both mixed and numeric-only attribute data sets to show the efficacy of CONOUT in handling different types of attributes.

Table 1: Data set summary

| Name | size $n$ | dim. $d$ | outliers (%) | type |
|---|---|---|---|---|
| AdFraud | 18,959 | 14 | 208 (1.09%) | Mixed |
| SatImage | 5,803 | 36 | 71 (1.20%) | Numeric |
| Pens | 6,870 | 16 | 156 (2.27%) | Numeric |
| Mammography | 11,183 | 6 | 260 (2.32%) | Numeric |
| Seismic | 2,584 | 18 | 170 (6.50%) | Mixed |
| Shuttle | 49,097 | 9 | 3,511 (7.21%) | Numeric |
| Income (Adult) | 48,842 | 14 | 7,841 (24.08%) | Mixed |
| Satellite | 6,435 | 36 | 2,036 (32.21%) | Numeric |

**Baselines** We compare CONOUT to the following state-of-the-art approaches in both traditional outlier and contextual detection literature.

- **ROCOD:** Robust contextual outlier detection [9] (ROCOD) combines both local and global effects in outlier detection. ROCOD requires a single context to be pre-specified. Such a context is not available for any of our data sets, as such, we assign all the categorical attributes in the mixed type data sets in Table 1 as contexts and use numerical ones as indicators. However, picking contexts in numeric data sets is non-trivial and requires domain expertise. Hence, we omit ROCOD from comparison on those data sets.
- **iForest:** iForest [10] is the isolation based tree ensemble detector discussed in §2.2. We set number of trees $t = 100$ and $\psi = 256$ as suggested in the paper.
- **LOF:** Local Outlier Factor (LOF) [3] compares the local density of each point to its neighbors. We vary $k$ (number of nearest neighbors) between 10 to 100 as suggested in [3] and pick the maximum outlier score.
- **ocSVM:** One class SVM (ocSVM) [12] is a popular outlier detection technique based on the principles of support vectors. We use the default ($\gamma = \frac{1}{n}$) radial kernel and $\nu = 0.5$ for our experiments as suggested in [12].
- **Expert-CONOUT:** Instead of generating multiple contexts in CONOUT, we use a *single* context that our industry collaborators hand-created for the AdFraud data set to investigate the efficacy of our context generation. For a fair comparison, we use the same context for ROCOD in the AdFraud data set.

[9] ODDS (Outlier Detection DataSets): http://odds.cs.stonybrook.edu

### 3.1 Case study : Ad Fraud Domain

CONOUT is motivated by the ad fraud problem, briefly, of identifying publishers that make revenue through a variety of illegitimate schemes. First, we provide a primer about the advertisement ecosystem in Figure 3.

Fig. 3: **Primer on advertisement ecosystem**: An advertiser (buyer) is an entity that manages ad campaigns of multiple brands, a publisher (seller) provides the real-estate for hosting ads. To establish the buyer/seller relationship, three key intermediaries are involved - Display Side Platforms (DSP), Supply Side Platforms (SSP) and an ad-exchange. Publishers earn revenue based on the number of views, clicks, or actions on ads, and have incentives to commit fraud. To mitigate this, DSPs maintain an up-to-date blacklist of publishers to avoid, while bidding on ad requests. The limitation of such a list is that a fraudulent publisher could switch between multiple websites to get through them. Hence, a robust data driven approach to identify fraudulent publishers is required.


Table 2: Publisher features extracted for AdFraud. N,O,C indicate numerical, ordinal and categorical features respectively. Fields with P are protected based on the agreement with the DSP. The mean features are aggregated over multiple ad requests of a publisher.

| Feature | Description | Type | Mean/Arity |
|---|---|---|---|
| **Features from the DSP (Display Side Platform) collaborator** | | | |
| `total_visitors` | # of users DSP has served ads | N | 2122.87 |
| `mean_revenue` | Mean revenue generated on ads | N | P |
| `mean_bid` | Mean bid amount placed by the DSP | N | P |
| `mean_cost` | Mean cost paid by the DSP. | N | P |
| `mean_conversions` | Mean conversions from ads | N | 0.002 |
| `mean_clicks` | Mean clicks on the ads placed | N | 1.247 |
| `unique_users` | #of unique users DSP has served ads | N | 1895.42 |
| **Features from public repositories (who.is and myip.ms)** | | | |
| `websites_before` | # of websites earlier hosted on publisher's IP | N | 90.77 |
| `websites_notworking` | # of websites hosted on IP that are not working | N | 52.072 |
| `websites_working` | # of websites hosted on IP that are working | N | 458.91 |
| `popularity` | Average # of visitors per day | O | 2,338,150 |
| `alexa_rank` | Alexa rank of the publisher | O | 570,403 |
| `host_country` | Hosting country of the publisher | C | 5 |
| `category` | Root IAB category of the publisher | C | 5 |

**Application** To detect fraudulent publishers, we partner with a large DSP. We build the publisher features from a snapshot of ad requests served by the DSP. Additionally, we collect data from publicly available repositories[10] which keep track of various network level information about publishers (See Table 2 for a full list). Our AdFraud data set contains a total of 18,959 unique publishers.

[10] https://www.whois.com/whois/, https://myip.ms/

We also have labels of each publisher based on their prior history of committing fraud, which allows us to assess the detection performance of CONOUT.

**Context Groups:** In Figure 4, we visualize the context groups formed by performing clustering using our proposed unified measure developed in §2.1. We note that the `popularity` of a publisher is grouped together with `alexa_rank`. This is intuitive since both these attributes would be highly dependent - a higher `popularity` would mean a lower `alexa_rank`. All the features related to monetary gains are grouped together. Ads with higher bid requests cost more to the advertiser and in return one could expect higher revenue, clicks and conversion. The `total_visitors` of a publisher and `unique_visitors` would be highly correlated and are grouped together. In total, we have $G = 6$ context groups forming $K = 63$ contexts.

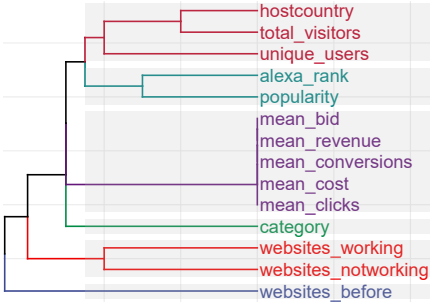



Fig. 4: Context Groups of AdFraud

**Detection Performance:** In Figure 1a, we compare AUPRC (Area Under the Precision-Recall Curve) of CONOUT with the baselines listed earlier averaged across five independent runs. We notice that CONOUT significantly outperforms all the baselines in detecting fraudulent publishers. Interestingly, Expert-CONOUT (CONOUT with a context specified by experts from our partner DSP) is inferior to CONOUT with automatically generated contexts, demonstrating the utility of multiple contexts and the sub-optimality of a hand-made context. ROCOD with the same pre-specified context also performs poorly in comparison to CONOUT further supporting the incorporation of multiple contexts. The non-contextual techniques ocSVM and LOF perform poorly, followed by iForest indicating the importance of contexts in the ad fraud domain.

**Cost-Benefit Analysis:** AUPRC measures performance assuming all the fraudulent publishers are equally important to detect. However, we note that as different publishers employ different schemes, detecting the *right* set of fraudulent publishers becomes more important. For instance, a fraudulent publisher that targets costly bids would deplete the advertiser's budget quicker, returning higher gains when caught. Therefore, we use a second evaluation metric more relevant to the ad domain, which compares the benefits an advertiser obtains by employing a given detector in the bidding mechanism.

Specifically, we perform a cost benefit analysis, computing the advertiser's return on investment (ROI). This is done using a simulation mirroring the ad buying ecosystem outlined in Algo. 2, where the advertiser decides whether or not to bid on an ad request (denoted by *bidbit*) from a given publisher based on its score by the employed detector (the higher the score/risk $s_i$, the more likely the advertiser *not* to bid. The simulation incorporates the importance of a publisher based on the the number of requests made (`total_visitors`), cost to the advertiser (`mean_cost`) and revenue (`mean_revenue`). The more $s_i$ reflects the true labels of publishers, the more revenue the advertiser makes due to de-

cisions on correct estimates, and the more they lose (to fake eyeballs) otherwise.

**Algorithm 2** Advertiser Cost-Benefit Analysis

**Input:** $\{s_1^1, s_2^1, \ldots, s_n^1\}, \ldots \{s_1^M, s_2^M, \ldots, s_n^M\}$, fraudulent scores of $M$ competing detectors, Investment (Budgets) $\boldsymbol{b} = \{b^1, b^2, \ldots, b^M\}$, initially all equal to bgt.
**Output:** $ROI = \{roi^1, roi^2, \ldots, roi^M\}$, Return on Investments, all set to zero initially.

**while** any of $b^l \in \boldsymbol{b} > 0$ **do** ▷ *Until all the budgets are depleted*
  Simulate an ad request by choosing a publisher $i$ based on $total\_visitors_i$
  **for** each detector $l$ **do**
    Based on $s_i^l$, decide $bidbit^l$, ▷ *Decide whether or not to bid on the request*
    **if** $bidbit^l = 1$ and $i$'s true label is benign **then** ▷ *Budget spent, revenue gained*
      $b^l = b^l - \text{mean\_cost}_i$, $roi^l = roi^l + \text{mean\_revenue}_i/\text{bgt}$
    **else if** $bidbit^l = 1$ and $i$'s true label is fraudulent **then**
      $b^l = b^l - \text{mean\_cost}_i$ ▷ *Budget spent lost due to fraudulent scheme*
    **else** ▷ $bidbit^l = 0$, *i.e., DSP does not bid due to high risk (outlier score)*
      $b^l, roi^l$ stay the same ▷ *No budget spent, no revenue received*
**Return** $ROI \leftarrow \{roi^1, roi^2, \ldots, roi^M\}$

Table 3: Comparison of relative % gains in ROI with varying budgets.

| **budget** | CONOUT | **p-value wrt naïve (wrt Exp-CONOUT)** | **Expert-CONOUT** | **ROCOD** | **iForest** | **LOF** | **ocSVM** |
|---|---|---|---|---|---|---|---|
| 500 | **NA (24.58)** | 2e-5 (2e-5) | NA | NA | NA | NA | NA |
| 1000 | **56.71 (19.55)** | 1e-7 (4e-5) | 43.38 | 21.26 | 38.98 | 2.40 | 0.40 |
| 1500 | **59.78 (19.63)** | 2e-6 (1e-4) | 44.62 | 24.15 | 37.36 | 2.47 | 0.47 |
| 2000 | **60.63 (25.65)** | 2e-8 (2e-5) | 47.08 | 26.45 | 38.54 | 2.31 | 0.31 |
| 2500 | **65.62 (21.07)** | 1e-6 (4e-4) | 50.11 | 25.78 | 40.69 | 2.53 | 0.53 |

We perform the simulation for 10,000 times with varying initial bgt values. In Table 3, we report the relative % gain—the difference of mean ROI of a method with that of a naïve detector (that bids 98.91 times out of 100—based on the outlier % in AdFraud dataset) divided by mean ROI of naïve detector. We observe that CONOUT outperforms all the baselines and achieves a minimum of ∼57% relative gain against a naïve detector. NA's at budget 500 are due to zero ROI obtained using a naïve detector. We also report the relative gain of CONOUT over its closest competitor, Expert-CONOUT (in brackets), where we use a single context provided by our industry collaborators. CONOUT provides a minimum of ∼20% relative gain in ROI when compared to Expert-CONOUT.

Additionally, to assess the significance of the differences in CONOUT's ROI over the naïve detector as well as Expert-CONOUT (its closest competitor), we employ a one tailed paired two sample $t$-test and report the corresponding p-values in Table 3. The lower p-values indicate that CONOUT achieves statistically significant gains in both cases. Moreover, the estimated ROI of CONOUT is consistent, where returns increase as investment/bgt increases. We remark that LOF, ocSVM and the naïve detector yielded no return with a bgt of 500. Given the low returns in ad domain, this is expected. On the contrary, this further highlights the improved gains obtained by CONOUT even at low budget.

### 3.2 CONOUT on Public Datasets

Table 4: AUPRC on Mixed Datasets

| Method/ Data | CONOUT | ROCOD | iForest | LOF | ocSVM |
|---|---|---|---|---|---|
| AdFraud | **0.5138** | 0.3011 | 0.3270 | 0.1344 | 0.0108 |
| Seismic | **0.9180** | 0.8083 | 0.8970 | 0.9007 | 0.9011 |
| Income (Adult) | **0.5812** | 0.5604 | 0.3128 | 0.2760 | 0.2377 |

Table 5: AUPRC on Numeric Datasets

| Method/ Data | CONOUT | iForest | LOF | ocSVM |
|---|---|---|---|---|
| Pens | **0.3574** | 0.3193 | 0.0499 | 0.1413 |
| SatImage | **0.9442** | 0.9011 | 0.3525 | 0.1101 |
| Mammography | 0.1902 | **0.1925** | 0.1425 | 0.1806 |
| Satellite | **0.6862** | 0.6557 | 0.4037 | 0.3819 |
| Shuttle | **0.9911** | 0.9793 | 0.2481 | 0.4331 |

Next, we compare the detection performance of CONOUT on various data sets with ground truth outliers listed in Table 1. In Tables 4 and 5, we report the mean average precision (averaged over five independent runs) of competing methods on mixed attribute and numeric attribute data sets respectively. CONOUT consistently outperforms the baselines. The importance of incorporating multiple contexts is evident in the Seismic and Income data sets, where baselines considering single context (ROCOD[11]) or no context (iForest, LOF, ocSVM) perform worse compared to CONOUT. Results are similar on numeric data sets. Here, we do not show ROCOD as it requires a pre-specified context, which is not available nor easy to set. CONOUT outperforms the no-context baselines highlighting the potential benefits of incorporating multiple contexts.

## 4 Related Work

Outlier detection has been extensively studied in the literature [1]. Contextual outlier detection (COD) is notably different and is the focus of our work. COD has been studied in [9,13,17]. The method developed in [17] applies to spatio-temporal data where the contexts comprise of spatial, temporal or spatio-temporal attributes. Direct applicability of such techniques to other types of data is not obvious.

Song et al. [13] takes a generative approach to model the relation between context and indicators. Both are modeled separately as a mixture of multiple Gaussian components. Next, a mapping function between the Gaussian components is learned using EM to incorporate the intuition of similar contexts generating similar indicators. Liang et al. [9] tackles the problem of sparsity in the context space by proposing an ensemble of local and global estimation of the indicators. The local estimates are obtained using a kNN regression where context is used to find the neighbors. The global estimates are obtained via a linear or a non-linear regression using both indicators and contexts. All the above techniques assume that there is a *single*, *user-given* context.

Wang et al. [16] propose a graph based method to find contextual neighbors without the need of a demarcation. However, the contexts here are defined as a set of instances rather than a set of attributes which is different from our problem. Angiulli et al. [2] considers the problem of characterizing outliers in a *labeled* data set by automatically finding context attributes and a *single* indicator attribute to explain a group of outliers—which is not a detection technique.

[11] We assign the categorical attributes as contexts in Seismic and Income data sets.

## 5 Conclusion

We introduced CONOUT for contextual outlier detection addressing the problem of automatically finding and incorporating multiple contexts while handling mixed type attributes. In summary, we make the following contributions.

- **Automatic context formation**, by developing a unified measure that can handle mixed type attributes;
- **Leveraging multiple contexts**, by proposing a context-incorporated detection algorithm that is assembled over multiple contexts;
- **Parameter-free nature**, by tuning its (one) hyperparameter via an unsupervised model selection criterion, that makes CONOUT a *one-click* algorithm.

Through experiments on real-world data sets, we showed the effectiveness of CONOUT over existing techniques in detection performance. We motivated and applied CONOUT to the ad domain where CONOUT not only improves detection but also provides statistically significant revenue gains to advertisers.

**Acknowledgments** : This research is sponsored by Adobe University Marketing Research Award , NSF CAREER 1452425 and IIS 1408287. Any conclusions expressed in this material do not necessarily reflect the views expressed by the funding parties.

## References

1. C. C. Aggarwal. *Outlier Analysis*. Springer, 2013.
2. F. Angiulli, F. Fassetti, and L. Palopoli. Discovering characterizations of the behavior of anomalous subpopulations. *IEEE TKDE*, 25(6):1280–1292, 2013.
3. M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: identifying density-based local outliers. In *SIGMOD*, volume 29, pages 93–104. ACM, 2000.
4. V. Dave, S. Guha, and Y. Zhang. Viceroi: Catching click-spam in search ad networks. In *SIGSAC*, pages 765–776. ACM, 2013.
5. A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. of Royal Stat. Society*, 39:1–38, 1977.
6. J. Gao and P.-N. Tan. Converting output scores from outlier detection algorithms into probability estimates. In *ICDM*, pages 212–221. IEEE, 2006.
7. T. Hastie, R. Tibshirani, and G. Walther. Estimating the number of data clusters via the gap statistic. *J. of Royal Stat. Society B*, 63:411–423, 2001.
8. E. L. Lehmann and J. P. Romano. *Testing statistical hypotheses*. Springer, 2006.
9. J. Liang and S. Parthasarathy. Robust contextual outlier detection: Where context meets sparsity. In *CIKM*, pages 2167–2172. ACM, 2016.
10. F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *ICDM*. IEEE, 2008.
11. D. Pelleg, A. W. Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML*, pages 727–734, 2000.
12. B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt. Support vector method for novelty detection. In *NIPS*, pages 582–588, 2000.
13. X. Song, M. Wu, C. Jermaine, and S. Ranka. Conditional anomaly detection. *IEEE TKDE*, 19(5):631–645, 2007.
14. N. Spirin and J. Han. Survey on web spam detection: principles and algorithms. *ACM SIGKDD Explorations Newsletter*, 13(2):50–64, 2012.
15. S. C. Tan, K. M. Ting, and T. F. Liu. Fast anomaly detection for streaming data. In *IJCAI*, volume 22, page 1511, 2011.
16. X. Wang and I. Davidson. Discovering contexts and contextual outliers using random walks in graphs. In *ICDM*, pages 1034–1039. IEEE, 2009.
17. G. Zheng, S. L. Brantley, T. Lauvaux, and Z. Li. Contextual spatial outlier detection with metric learning. In *KDD*, pages 2161–2170. ACM, 2017.