ESTIMATING THE CONTAMINATION FACTOR'S DISTRIBUTION IN UNSUPERVISED ANOMALY DETECTION

A PREPRINT

Lorenzo Perini
DTAI lab & Leuven.AI
Department of Computer Science
KU Leuven
B-3000 Leuven, Belgium
lorenzo.perini@kuleuven.be

Paul-Christian Bürkner
Cluster of Excellence SimTech
University of Stuttgart
Universitätsstr. 32, 70569 Stuttgart
paul.buerkner@gmail.com

Arto Klami
Department of Computer Science
University of Helsinki
P.O.Box 68, 00014 Helsinki, Finland
arto.klami@helsinki.fi

October 20, 2022

ABSTRACT

Anomaly detection methods identify examples that do not follow the expected behaviour, typically in an unsupervised fashion, by assigning real-valued anomaly scores to the examples based on various heuristics. These scores need to be transformed into actual predictions by thresholding, so that the proportion of examples marked as anomalies equals the expected proportion of anomalies, called contamination factor. Unfortunately, there are no good methods for estimating the contamination factor itself. We address this need from a Bayesian perspective, introducing a method for estimating the posterior distribution of the contamination factor of a given unlabeled dataset. We leverage on outputs of several anomaly detectors as a representation that already captures the basic notion of anomalousness and estimate the contamination using a specific mixture formulation. Empirically on 22 datasets, we show that the estimated distribution is well-calibrated and that setting the threshold using the posterior mean improves the anomaly detectors' performance over several alternative methods. All code is publicly available for full reproducibility.

Keywords Anomaly Detection · Unsupervised Learning · Bayesian Methods · Contamination Factor

1 Introduction

Anomaly detection [Chandola et al., 2009] is the task of automatically identifying examples that do not conform to the expected behaviour. The anomalies are often indicative of critical events such as intrusions in web networks [Malaiya et al., 2018, Xin et al., 2018], failures in the petroleum extraction [Martí et al., 2015], excessive usage of water [Vercruyssen et al., 2018], or breakdowns in wind turbines [Zaher et al., 2009] and gas turbines [Yan and Yu, 2019]. Such events have an associated high cost and detecting the anomalies in time avoids wasting resources.

Typically, anomaly detection is tackled from an unsupervised perspective [Maxion and Tan, 2000, Goldstein and Uchida, 2016, Zong et al., 2018, Perini et al., 2020a, Han et al., 2022] because labeled examples, especially anomalies, may be expensive and difficult to acquire (e.g., you do not want to voluntarily break the equipment simply to observe anomalous behaviours), or simply rare (e.g., you may need to inspect many examples before finding an anomalous one). Unsupervised anomaly detectors exploit data-driven heuristic assumptions to assign a real-valued score to each example denoting how anomalous it is. For instance, KNNO [Angiulli and Pizzuti, 2002] assigns the example's distance to its k-th nearest neighbour, while AUTOENCODER [Chen et al., 2018] uses the example's reconstruction error. Using such anomaly scores enables ranking the examples from most to least anomalous.

Converting the anomaly scores into discrete predictions would practically allow the user to flag the anomalies. Commonly, one chooses a predictive threshold and assigns the anomaly label to the examples with higher score, while all the remaining ones are considered normal. However, setting the threshold is a challenging task as it cannot be tuned (e.g., by maximizing the model performance) due to the absence of labels [Soenen et al., 2021]. One approach

is to set the threshold such that the proportion of scores above it matches the dataset's *contamination factor* γ, the expected proportion of anomalies. This allows perfect rankings to be transformed into perfect predictions. Usually, the contamination factor is assumed to be given by a domain expert, but in most of the real-world scenarios it is actually unknown.

Estimating the contamination factor is challenging. Existing works provide an estimate by either collecting some normal labels [Perini et al., 2020b] or by exploiting domain knowledge [Perini et al., 2022]. Moreover, one can apply statistical threshold estimators that directly set a threshold on the anomaly scores, and derive the contamination factor as proportion of scores higher than the threshold. For instance, the Modified Thompson Tau test thresholder (MTT) is based on the modified Thompson Tau test [Rengasamy et al., 2021], while the Inter-Quartile Region thresholder (IQR) sets the threshold as the third quartile plus 1.5 times the inter-quartile region [Bardet and Dimby, 2017]. We will later explain and evaluate several estimators in Section 4.

Transforming the scores using an incorrect estimate of the contamination factor (or, equivalently, an incorrect threshold) deteriorates the anomaly detector's performance [Fourure et al., 2021, Emmott et al., 2015] and reduces the trust in the detection system. If such an estimate was coupled with a measure of uncertainty, one could take into account this uncertainty to improve decisions.

We study the estimation of the contamination factor from a Bayesian perspective and propose γGMM, the first algorithm for estimating the contamination factor's (posterior) distribution in unlabeled anomaly detection setups. First, we use a set of unsupervised anomaly detectors to assign anomaly scores for all examples and use these scores as a new representation for the data. Second, we fit a Bayesian Gaussian Mixture model with a Dirichlet process prior (DPGMM) [Ferguson, 1973, Rasmussen, 1999, Görür and Edward Rasmussen, 2010] in this new space. If we knew which components contain the anomalies, we could derive the contamination factor's posterior distribution as the distribution of the sum of such components' weights. Because we do not know this, as a third step γGMM estimates the probability that the k most extreme components are jointly anomalous, and uses this information to construct the desired posterior. The method explained in detail in Section 3.

In summary, we make four contributions. First, we adopt a Bayesian perspective and introduce the problem of estimating the contamination factor's posterior distribution. Second, we propose an algorithm that is able to sample from this posterior. Third, we demonstrate experimentally that the implied uncertainty-aware predictions are well calibrated and that taking the posterior mean as point estimate of γ outperforms several other algorithms in common benchmarks. Finally, we show that using the posterior mean as a threshold improves the actual anomaly detection accuracy.

2 Preliminaries

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, and $X\colon \Omega \to \mathbb{R}^d$ a random variable, from which a dataset D containing N i.i.d. examples $x_1, \ldots, x_N \in \mathbb{R}^d$ is drawn. Assume that X has a distribution of the form $p(X) = (1-\gamma)\cdot p(X^{(1)}) + \gamma \cdot p(X^{(2)})$, where $X^{(1)}$ and $X^{(2)}$ are the random variables representing, respectively, normals and anomalies, and γ is the *contamination factor*, the proportion of anomalies. An (unsupervised) *anomaly detector* is a measurable function $f\colon \mathbb{R}^d \to \mathbb{R}$ that assigns real-valued anomaly scores $S = f(X)$ to the examples. Such anomaly scores follow the rule that *the higher the score, the more anomalous the example*.

A Gaussian mixture model (GMM) with K components [Roberts et al., 1998] is a generative model such that $p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$, where π_k is the mixing proportion, μ_k and Σ_k are the mean and covariance matrix of the component k. For finite mixtures we typically have a Dirichlet prior over $\pi = [\pi_1, \ldots, \pi_K]$, but Dirichlet Process (DP) priors allow treating also the number of components as unknown [Görür and Edward Rasmussen, 2010]. For both cases we need approximate inference to estimate the posterior of the model parameters.

3 Methodology

We tackle the problem: **Given** an unlabeled dataset D and a set of M unsupervised anomaly detectors; **Estimate** the (posterior) distribution of the contamination factor γ.

Learning from an unlabeled dataset has three key challenges. First, the absence of labels forces us to make relatively strong assumptions. Second, the anomaly detectors rely on different heuristics that may or may not hold, and their performance can hence vary significantly across datasets. Third, we need to be careful in introducing user-specified hyperparameters, because setting them properly may be as hard as directly specifying the contamination factor.

In this paper we propose γGMM, a novel approach that estimates the contamination factor's posterior distribution using a Bayesian approach. It consists of four steps, which we introduce on a conceptual level now and mathematically, in

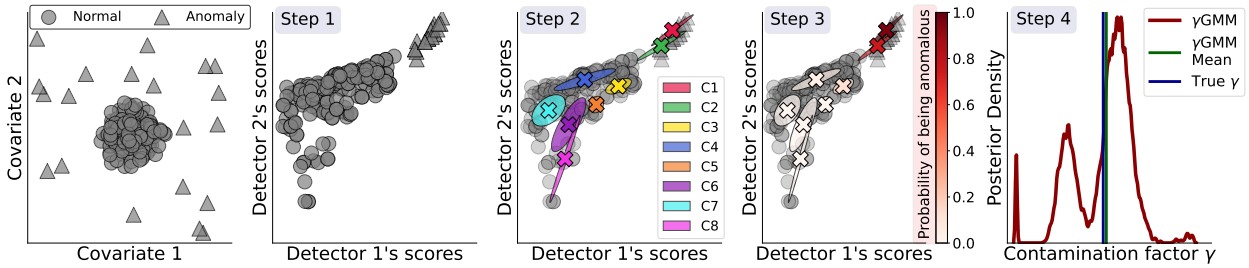

Figure 1: Illustration of the γGMM's four steps on a 2D toy dataset (left plot): we 1) map the 2D dataset into an $M = 2$ dimensional anomaly space, 2) fit a DPGMM model on it, 3) compute the components' probability of being anomalous (conditional, in the plot), 4) derive $\gamma|S$'s posterior. γGMM's mean is an accurate point estimate for the true value γ^*.

more detail, afterwards. See Figure 1 for an illustration.

Step 1. Because anomalies may not follow any particular pattern, γGMM maps the covariates X into an M dimensional anomaly space, where the dimensions correspond to the anomaly scores assigned by the M unsupervised anomaly detectors. Within each dimension of such a space, the evident pattern is that "the higher the more anomalous".

Step 2. We model the data in this space using a Dirichlet Process Gaussian Mixture Model (DPGMM) [Neal, 1992, Rasmussen, 1999, Görür and Edward Rasmussen, 2010]. We assume that each of the (potentially many) mixture components contains either only normals or only anomalies. If we knew which components contained anomalies, we could then easily derive γ's posterior as the sum of the mixing proportions π of the anomalous components. However, such information is not available in our setting.

Step 3. Thus, we estimate the probability of the most extreme k components being anomalous. This poses three challenges: (a) how to represent each component in M-dimensional space by a single value to sort them from the most to the least anomalous, (b) how to compute the probability that the kth component is anomalous given that the $k-1$th is such, and (c) how to derive the target probability that exactly k components are jointly anomalous.

Step 4. γGMM estimates the contamination factor's posterior by exploiting such a joint probability and the components' mixing proportions posterior.

In the following, we describe these steps in detail.

3.1 Representing data using anomaly scores

Learning from an unlabeled anomaly detection dataset has two major challenges. First, anomalies are rare and sparse events, which makes it hard to use common unsupervised methods like clustering [Breunig et al., 2000]. Second, making assumptions on the unlabeled data is challenging due to the absence of specific patterns in the anomalies, which makes it hard how to choose a proper anomaly detector.

Therefore, we use a set of M anomaly detectors to map the sample space into an M-dimensional score space $\mathbb{R}^M$, such that an example x gets a score s:

$$\mathbb{R}^d \ni x \to [f_1(x), f_2(x), \ldots, f_m(x)] = s \in \mathbb{R}^M.$$

This has two main effects: (1) it introduces an interpretable space where the evident pattern is that, within each dimension, higher scores are more likely to be anomalous, and (2) it accounts for multiple inductive biases by using multiple arbitrary anomaly detectors.

To make the dimensions comparable, we (independently for each dimension) linearly map the scores to the positive reals, apply the log transform to shorten heavy right tails, and normalize them to have zero mean and unit variance.

3.2 Modeling the density with DPGMM

We use mixture models as basis for quantifying the distribution of the contamination factor, relying on their ability to model proportions of examples using the mixture weights. For flexible modelling we use a Dirichlet Process Gaussian

Mixture Model (DPGMM) [Görür and Edward Rasmussen, 2010]

$$\begin{aligned} s_i &\sim \mathcal{N}(\tilde{\mu}_i, \tilde{\Sigma}_i) \\ (\tilde{\mu}_i, \tilde{\Sigma}_i) &\sim G \\ G &\sim DP(G_0, \alpha) \\ G_0 &= \mathcal{NIW}(m, \lambda, V, u) \end{aligned}$$

where G is a random distribution of the means μ_i and covariance matrices Σ_i, drawn from a DP with base distribution G_0. We use the explicit representation $G = \sum_{k=1}^{\infty} \pi_k \delta_{(\mu_k, \Sigma_k)}(\tilde{\mu}_i, \tilde{\Sigma}_i)$, where $\delta_{(\mu_k, \Sigma_k)}$ is the delta distribution at (μ_k, Σ_k) and π_k follow the stick-breaking distribution. We set G_0 as Normal Inverse Wishart [Nydick, 2012] with parameters m, λ, V, u common to all components.

The DP formulation allows us to refrain from specifying the number of components K. After fitting the model, we only consider the components with at least one observation assigned to them and propagate all the remaining density uniformly over the active components. Thus, for the following steps we can still proceed as if the model was a finite mixture with π following a Dirichlet distribution.

We use variational inference (VI; see Blei et al. [2017] for details) for approximating the posterior as VI is computationally efficient and sufficiently accurate for our purposes. Alternative methods (e.g., Markov Chain Monte Carlo Brooks et al. [2011]) could also be used but were not considered worth the additional computational effort here.

3.3 Estimating the components' anomalousness

We assume each mixture component only contains anomalous or normal samples, which is a reasonable assumption when K is large. If we knew which components contain anomalies, we could directly derive the posterior of the contamination factor γ as the sum of the mixing proportions π_k of those components. This is naturally not the case, but instead we need to estimate it in an unsupervised fashion.

More formally, we estimate the probability that k (out of all K) components are anomalous such that we can derive γ's posterior by averaging over all the values $0 \leq k \leq K$. We do this in three steps. Initially, we sort the components by degree of anomalousness, which comes natural from the representation we made in Step 1 (Sec. 3.1). Then, our insight is that the kth component c_k can be anomalous only if the $k-1$th is such. This points to the estimation of conditional probabilities, i.e., the probability of c_k being anomalous given that c_{k-1} is anomalous. Finally, the probability that exactly the first k components are anomalous can be obtained using basic rules of probability theory.

Assigning an ordering to the components. As initial step for computing the joint probability, we need to design an ordering map for the components, where the "top-right" component is the most anomalous one. We do this in a manner that accounts for the uncertainty of the components' parameters to rank high the components that can be reliably identified as anomalous: We want the means to be high but the variance low, to avoid the risk that also examples with low anomaly scores could belong to the component.

We construct the overall ranking using dimension-specific scores because our normalization cannot remove all statistical differences between the different detectors. Formally, let $r \colon \mathbb{R}^M \times \mathbb{R}^{M \times M} \to \mathbb{R}$ be the function of the mean μ_k and the covariance matrix Σ_k that assigns a representative value to the component k. We set r as

$$r\left(\mu_k^{(z)}, \Sigma_k^{(z)}\right) = \frac{1}{M} \sum_{j=1}^{M} \frac{\mu_k^{j\,(z)}}{1 + \sqrt{\Sigma_k^{j,j\,(z)}}}, \tag{1}$$

where $\mu_k^{(z)}$ and $\Sigma_k^{(z)}$ are samples from the parameters' posterior distributions of the kth component c_k. We obtain a representative value of the whole component by taking the expected value of r, i.e. through $\mathbb{E}[r(\mu_k, \Sigma_k)]$.

We add 1 to the component's standard deviation for two reasons. First, if a component contains examples with almost the same covariate values, the standard deviation would be close to 0 and the ratio would explode towards infinity, masking any effect of the mean. Second, adding 1 is reasonable because it is equal to the theoretical upper bound of the components variances, as they are normalized (Sec. 3.1).

Without loss on generality, from now on we assume that the components' index k is ordered based on their representative value such that the kth component c_k has higher value (i.e., more anomalous) than the $k+1$th component.

Estimating the probability that the kth component is anomalous. Because the components are sorted by anomalousness, our key insight is that *the kth component c_k can be anomalous only if the $k-1$th is anomalous*. Formally,

$$\mathbb{P}(c_k \text{ is anomalous} \mid c_{k-1} \textbf{ is } \text{anomalous}) > 0$$
$$\mathbb{P}(c_k \text{ is anomalous} \mid c_{k-1} \textbf{ is not } \text{anomalous}) = 0$$

for any $1 < k \leq K$. For the sake of brevity, we use $\mathbb{P}(c_k \mid c_{k-1})$ as $\mathbb{P}(c_k \text{ is anomalous} \mid c_{k-1} \text{ is anomalous})$.

Moreover, we assume $\mathbb{P}(c_1) = \mathbb{P}(c_1 \text{ is anomalous }) \in (0, 1)$. That is, we allow for the data to not have anomalies (< 1) but exclude certain knowledge of no anomalies (> 0). This is a sensible assumption because, if one knew for sure that no anomalies are in the data, then we trivially have $\gamma = 0$, whereas we still need to allow for the data to be free of anomalies if evidence suggests so.

We estimate the conditional probability as

$$\mathbb{P}(c_k|\ c_{k-1}) = \frac{1}{1 + e^{(\tau + \delta \cdot r(\mu_k, \Sigma_k))}}, \tag{2}$$

where τ and δ are the two hyperparameters of the sigmoid function, which will be carefully discussed in the next sections. Note that the principle itself is not restricted to this particular choice of functional form. One could apply any transformation that maps to $[0, 1]$, but the detailed derivations of the parameters would naturally be different.

Deriving the components' joint probability. Given the conditional probability $\mathbb{P}(c_k|\ c_{k-1})$, the joint probability $\mathbb{P}(\text{exact } c_1, \ldots, c_k)$ follows from simple steps. Taking inspiration from the sequential ordinal models [Bürkner and Vuorre, 2019], our insight is that exactly k components are jointly anomalous if and only if each of them is conditionally anomalous and the $k+1$th is not anomalous. Essentially,

$$\mathbb{P}(c_1, \ldots, c_k) = \mathbb{P}(c_1) \prod_{t=1}^{k-1} \mathbb{P}(c_{t+1}|c_t)(1 - \mathbb{P}(c_{k+1}|c_k)) \tag{3}$$

for any $k \leq K$, where $\mathbb{P}(c_{K+1}\ |c_K) = 0$ by convention.

3.4 Estimating the contamination factor's distribution

Given the joint probability that the first k components are anomalous (for $k \leq K$), the contamination factor γ's posterior distribution can be obtained as

$$p(\gamma|S) = \sum_{k=1}^{K} \mathbb{P}(c_1, \ldots, c_k) \cdot p\left(\sum_{j=1}^{k} \pi_j | S\right), \tag{4}$$

where $\pi_j|S$ is the posterior marginal distribution of the mixing proportion for the jth component. For any k, we have $p(\sum_{j=1}^{k} \pi_j|S) = \textsc{Beta}(\sum_{j=1}^{k} \alpha_j, \sum_{j=k+1}^{K} \alpha_j)$, where $(\pi_1|S, \ldots, \pi_K|S) \sim \textsc{Dir}(\alpha_1, \ldots, \alpha_K)$ [Lin, 2016].

Setting the sigmoid's hyperparameters τ and δ. Introducing new hyperparameters when the task is to estimate the contamination factor γ's posterior is risky because setting their value may be as difficult as directly providing a point estimate of γ. With regard to τ and δ, our key insight is that we can obtain them by asking the user two simple questions: (a) How likely is that no anomalies are in the data? (b) How likely is that a large amount of anomalies occurred, say, more than $t = 15\%$ of the data? Both of these values are supposed to be low. Let's call p_0 the first answer and p_{high} the second one. Formally,

$$p_0 = 1 - \mathbb{P}(c_1) = 1 - \frac{1}{1 + e^{(\tau + \delta \cdot r(\tilde{\mu}_1, \tilde{\Sigma}_1))}}$$
$$p_{\text{high}} = \mathbb{P}(\gamma \geq t|S) = \sum_{k=1}^{K} \mathbb{P}(c_1, \ldots, c_k) \mathbb{P}\left(\sum_{j=1}^{k} \pi_j \geq t | S\right)$$

One can use a numerical solver for non-linear equations with linear constraints to find the values of τ and δ that satisfy such constraints. The problem has a unique solution whenever $p_{\text{high}} \geq \mathbb{P}(\pi_1 \geq t)$. This holds almost always in our experimental cases, but in the Supplement we describe a procedure that can be used in the other scenarios. In the experiments, we show that changing the p_0 and p_{high} does not have a large impact on γ's posterior (see Q5, Sec. 4).

Sampling from γ's posterior distribution. Our estimate of the contamination factor's posterior $p(\gamma|S)$ does not have a simple closed form. However, we can sample from the distribution using a simple process. The DPGMM inference determines an approximation for $p(\pi, \mu, \Sigma|S)$ and all the quantities required for Equations 2, 3, 4 can be computed based on samples from the approximation. A description of the detailed sampling process is provided in the Supplement.

Additional technical details. Because our method uses the variational inference approximation, we run it 10 times and concatenate the samples to reduce the risk of biased distributions due to local minima. Moreover, after sorting the components, we set $\mathbb{P}(c_k|c_{k-1}) = 0$ for all $k > K' = \arg\max\{k\colon\ \mathbb{E}[\sum_{j=1}^{k} \pi_j] < 0.25\}$. This has the effect of setting an upper bound of 0.25 to the contamination factor γ. Because anomalies must be rare, we realistically assume that it is not possible to have more than 25% of them. Note that $\mathbb{E}[\pi_1] \geq 0.25$ cannot occur, as otherwise we could not set the hyperparameters p_0 and p_{high}.

4 Experiments

We empirically evaluate two aspects of our method: (a) whether it accurately estimates the contamination factor's posterior, and (b) how thresholding the scores using our method affects the anomaly detectors' performance. To this end, we address the following five experimental questions:

- Q1. Is the posterior estimate sharp and well-calibrated?
- Q2. How does γGMM compare to alternative threshold estimators?
- Q3. Does a better point estimate of γ improve the anomaly detector performance?
- Q4. What is the impact of the number of detectors M?
- Q5. How sensitive the method is to p_0 and p_{high}?

4.1 Experimental Setup.

Methods. We compare the sample mean of γGMM with 21 threshold estimators: *Kernel-based* (FGD [Qi et al., 2021] and AUCP [Ren et al., 2018]); *Curve-based* (EB [Friendly et al., 2013] and WIND [Jacobson et al., 2013]); *Normality-based* (ZSCORE [Bagdonavičius and Petkevičius, 2020], DSN [Amagata et al., 2021] and CHAU [Bol'shev and Ubaidullaeva, 1975]); *Regression-based* (CLF and REGR [Aggarwal, 2017]); *Filter-based* (FILTER [Hashemi et al., 2019] and HIST [Thanammal et al., 2014]); *Statistical test-based* (GESD [Alrawashdeh, 2021], MCST [Coin, 2008] and MTT [Rengasamy et al., 2021]); *Statistical moment-based* (BOOT [Martin and Roberts, 2006], KARCH [Afsari, 2011] and MAD [Archana and Pawar, 2015]); *Quantile-based* (IQR [Bardet and Dimby, 2017] and QMCD [Iouchtchenko et al., 2019]); *Transformation-based* (MOLL [Keyzer and Sonneveld, 1997] and YJ [Raymaekers and Rousseeuw, 2021]). We apply each threshold estimator to the univariate anomaly scores of each detector at a time. *We then average the contamination factors over the M detectors and use this value as the final point estimate for each dataset*. See the Supplement for more details.

Data. We carry out our study on 20 commonly used benchmark datasets and additionally 2 (proprietary) real tasks. The benchmark datasets contain semantically useful anomalies widely used in the literature [Campos et al., 2016]. The datasets vary in size, number of features, and true contamination factor. See Table 1 in the Supplement for details. For the real tasks, our experiments focus on preventing blade icing in wind turbines. We use two public wind turbine datasets, where sensors collect various measurements (e.g., wind speed, power energy, etc.) every 7 seconds for either 8 weeks (turbine 15) or 4 weeks (turbine 21). Following Zhang et al. [2018], we construct feature-vectors by taking the average over the time segment of one minute.

Metrics. We use three evaluation metrics to assess the performance of the methods. Contrary to all the threshold estimators, our method estimates the posterior of γ. Therefore, we measure the **probabilistic calibration** of γGMM's posterior using a QQ-plot with the x-axis representing the expected probabilities (i.e., assuming a perfectly calibrated distribution) and on the y-axis the empirical frequencies. That is, for $v \in [0, 0.5]$,

$$\text{Expected Prob.} = \mathbb{P}\left(\gamma^* \in [q(0.5 - v), q(0.5 + v)]\right) = 2v$$
$$\text{Empirical Freq.} = \frac{|\{\gamma \in [q(0.5 - v), q(0.5 + v)]\}|}{\#\text{experiments}},$$

where $q(u)$ is the quantile at the value u of our distribution, for $u \in [0, 1]$, and γ^* refers to the true dataset's contamination factor. For evaluating the point estimate of the methods, we use the mean absolute error (MAE) between the method's point estimate and the true value. Finally, we measure the impact of thresholding the scores using the methods' point estimate through the F_1 score, as common metrics like the Area Under the ROC curve and the Average Precision are not affected by different thresholds. Specifically, we measure the *relative deterioration* of the F_1 score:

$$F_1 \text{ deterioration} = \frac{F_1(f_m, D, \gamma^*) - F_1(f_m, D, \hat{\gamma})}{F_1(f_m, D, \hat{\gamma})}$$

where we compute the F_1 score on the dataset D using the anomaly detector f_m, and either the true value γ^* or an estimate $\hat{\gamma}$ to threshold the scores. The F_1 deterioration of a method is (mostly) negative, and the higher the better.

Setup. In the experiments we assume a transductive setting [Campos et al., 2016, Scott and Blanchard, 2008, Toron et al., 2022], where a dataset D is used both for training and testing. This is the typical setting of anomaly detection [Breunig et al., 2000, Schölkopf et al., 2001, Angiulli and Pizzuti, 2002, Liu et al., 2012, Goldstein and Dengel, 2012, Chen et al., 2018] because the absence of labels and patterns (for the anomaly class) avoids overfitting issues.

For each dataset, we proceed as follows: (i) use a set of M anomaly detectors to assign the anomaly scores S to each observation in the dataset D; (ii) map each anomaly score $s \in S$ to $\log(s - \min(S) + 0.01)$ and normalize them to have mean equal to 0 and standard deviation equal to 1; (iii) either use our method to estimate the contamination factor's posterior and extract the posterior mean as point estimate $\hat{\gamma}$, or use one of the threshold estimators to directly obtain a point estimate $\hat{\gamma}$ of the contamination factor (see methods paragraph above); (iv) evaluate the point estimates using the mean absolute error (MAE) between such estimate and the true value γ^*; (v) use the contamination factor's point estimate to threshold the anomaly scores of each of the M anomaly detectors f_m (individually); (vi) finally, we measure the F_1 score and compute the relative deterioration.

Hyperparameters, anomaly detectors and priors. Our method introduces two new hyperparameters: p_0 and p_{high} that we both set to 0.01 as default value.

We use 10 anomaly detectors with different inductive bias [Soenen et al., 2021]: KNN [Angiulli and Pizzuti, 2002], IFOREST [Liu et al., 2012], LOF [Breunig et al., 2000], OCSVM [Green and Richardson, 2001], AE [Chen et al., 2018], VAE [Kingma and Welling, 2013], LSCP [Zhao et al., 2019a], HBOS [Goldstein and Dengel, 2012], LODA [Pevnỳ, 2016], and COPOD [Li et al., 2020]. See the Supplement for additional details. We use the python library PyOD [Zhao et al., 2019b] for the code. The threshold estimators are implemented in the Python library PYTHRESH[1] with default hyperparameters.

For the variational inference of DPGMM we use the implementation in SKLEARN, with the Stick-breking representation [Dunson and Park, 2008], with 100 as upper bound of the number of components K. The parameter priors are set to the default values: means are set to 0, while the covariance matrices are set to identities of appropriate dimension. We opt for such (in our context) weakly-informative priors because sensible prior knowledge of the DPGMM hyperparameters is hard to come by in practice.

4.2 Experimental Results

Q1. Does our method estimate a sharp and well-calibrated posterior of γ? Figure 2 shows the contamination factor γ's posterior estimated by our method in 10 representative datasets (see Figure 8 in the Supplement for all the plots). In the top five plots, the distribution looks accurate as γ's true value (blue line) is close to the posterior mean (i.e., the expected value, the green line). On the contrary, the five plots below represent the five worst cases: although γ's true value sometimes falls on low density regions (Arrhythmia and Shuttle datasets), in many cases it would be quite likely to sample the true value from our posterior (KDDCup99, Parkinson, Glass datasets), which makes the density still quite reliable. Figure 3 shows the calibration plot. The posterior is well calibrated as it is very close to the dashed black line indicating perfectly calibrated distribution. The empirical frequencies deviate from the real probabilities by less than 5% (dark shadow grey) in more than 76% of the cases, while never deviating by more than 10% (light shadow grey).

Q2. How does γGMM compare to the threshold estimators? We take γGMM's posterior mean as our best point estimate of γ and compare such value to the point estimates obtained from the threshold estimators. Figure 4 illustrates the ordered MAE (mean $\pm$ std.) between the methods' estimate and the true γ. On average, γGMM obtains a MAE of 0.026 that is 20% lower than the best runner-up MTT and 27% lower than the third best method QMCD (MAE of 0.033

[1]Link: `https://github.com/KulikDM/pythresh`.

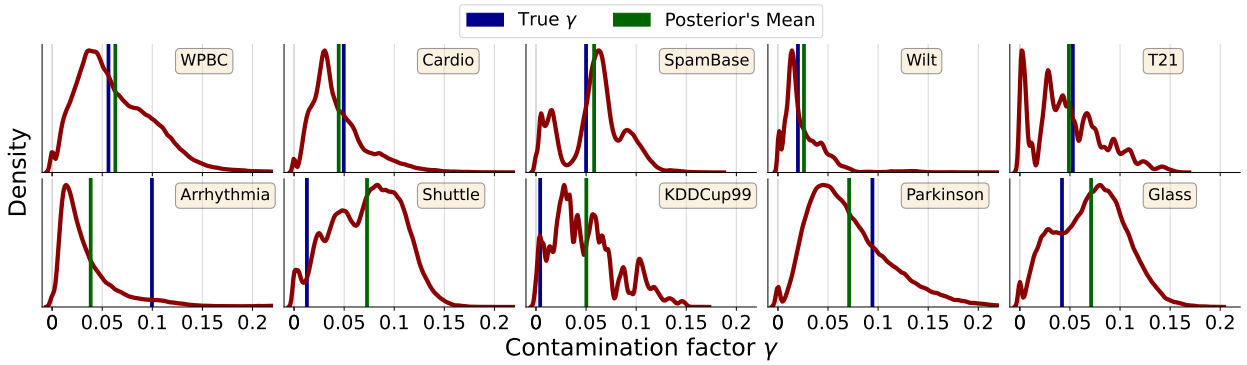

Figure 2: Illustration of how γGMM estimates γ's posterior on 10 representative datasets. On the top, the density has the sample mean close to the true value of γ, while on the bottom, the true value of γ falls far from our point estimate.

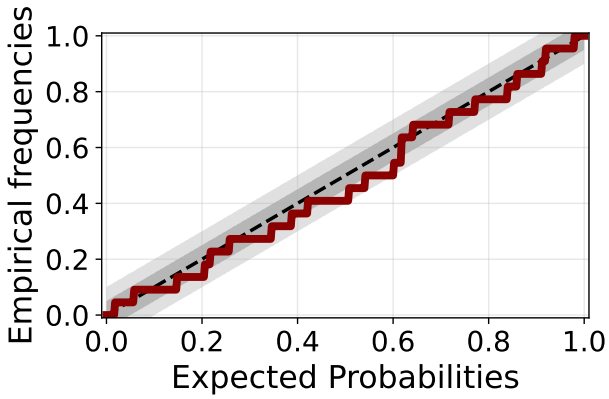

Figure 3: QQ-plot showing how calibrated γGMM's estimate of the distribution is. The black dashed line illustrates the perfect calibration scenario. Dark and light grey shades indicate a deviation of 5% and 10% from the black line.

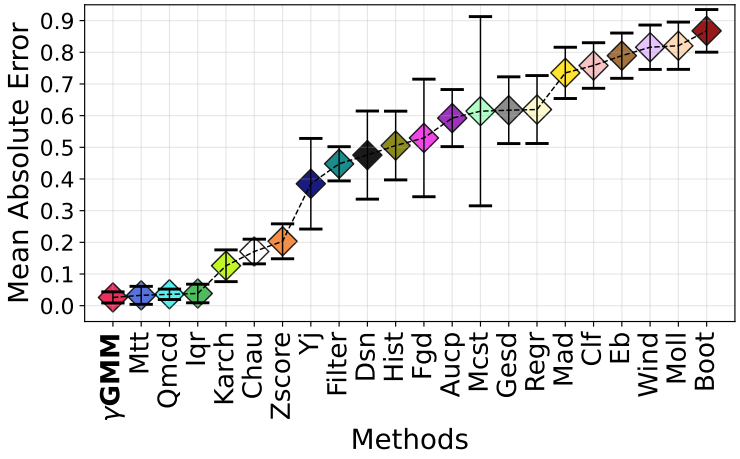

Figure 4: Average MAE ($\pm$ std.) of γGMM's sample mean compared to the other methods. Our method has the lowest (better) average, which is 20% lower than the runner-up.

and 0.036). For each experiment, we rank the methods from the best (position 1, lowest MAE) to the worst (position 22, greatest MAE). Our method has the best average rank (2.13 ± 1.04). Moreover, γGMM ranks first 8 times ($\approx 36\%$ of the cases), and for 13 times ($\approx 60\%$ of the cases) it is in the top two. The next best method, MTT, ranks first in 6 cases with average rank of 2.30 ± 1.10.

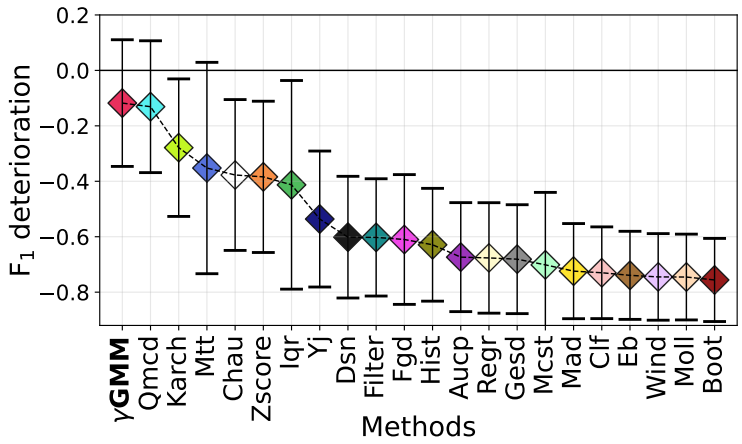

Figure 5: F_1 deterioration (mean $\pm$ std) for each method, where the higher the better. γGMM ranks as best method, obtaining $\approx 10\%$ higher average than QMCD.

Q3. Does a better contamination improve the anomaly detectors' performance? We use γGMM's posterior mean as point estimate to measure the F_1 score of the anomaly detectors, because sampling from the distribution would not make a fair comparison against the other methods that can only provide a point estimate.

Anomaly detectors that fail to rank the samples accurately perform poorly even when using the correct contamination factor. Since our focus is studying the effect of the contamination factor, we compare F_1 scores only over the detectors that work well for each of the datasets. For each dataset D, we use as set of detectors those achieving the greatest F_1 score using the true contamination factor, i.e. $\arg\max_{f_m} \{F_1(f_m, D, \gamma^*)\}$. The Supplement contains the list of detectors used for each experiment.

Figure 5 shows the average ($\pm$ std.) deterioration for each of the methods. On average, γGMM has the best F_1 deterioration (-0.117 ± 0.228) that is around 10% better than the runner-up QMCD (-0.131 ± 0.238), and 58% better than the next best KARCH (-0.279 ± 0.248). For 25% of the cases we get higher F_1 score with γGMM than when using the true γ^*. This is due to the (still incorrect) ranks made by the detectors, which achieve better performance with slightly incorrect contamination factors. Supplement provides further details on how the methods perform in terms of false alarms and false negatives.

Q4. What is the impact of the number M of detectors on the contamination factor's posterior? In the previous experiments, we used $M = 10$ detectors. To evaluate the effect of M, we repeated the experiments for random choices of 3, 5, or 7 detectors, with 10 replicates using different detectors for each case. Figure 6 shows that the calibration suffers if using fewer detectors, but already with $M = 5$ the method works fairly well. The variance of the results (over repeated experiments) also increases for lower M.

Q5. Impact of the hyperparameters p_0 and p_{high}. We evaluate the impact of the hyperparameters p_0 and p_{high} by running the experiments with much smaller and larger values than the default (0.01): we vary, one at the time, p_0, $p_{\text{high}} \in [0.0001, 0.001, 0.05, 0.01]$ and keep the other value set as default. Figure 7 shows the QQ-plot for p_0 (left) and p_{high} (right). In both cases, smaller hyperparameters lead to slightly under-estimated expected probabilities. Overall, our method is robust to different values of p_0, while p_{high} affects the calibration slightly more. We also compare the resulting 8 variants of γGMM in terms of MAE. Overall, the posterior means produce similar values than our default setting, obtaining an MAE that varies from 0.252 ($p_{\text{high}} = 0.001$, the best) to 0.32 ($p_0 = 0.0001$, the worst).

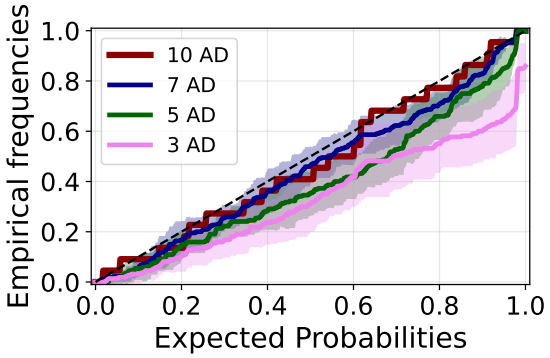

Figure 6: QQ-plot comparing the calibration curves of γGMM when a different number M of detectors is used. The colored shades report the uncertainty obtained by sampling randomly the chosen amount of detectors in a set of 10 detectors. The plot shows that the higher the number of detectors, the more calibrated the distribution.

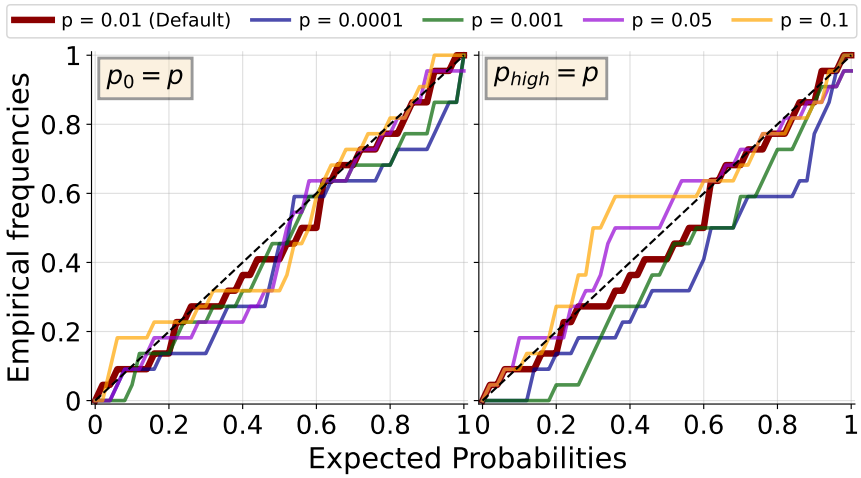

Figure 7: QQ-plot showing how calibrated γGMM's posterior mean would be if we varied p_0 (left) and p_{high} (right). While p_0 does not have a large impact on the method, the empirical frequencies slightly under (over) estimate the expected probabilities for low (high) values of p_{high}.

5 Conclusion

The literature on anomaly detection has, correctly, focused on unsupervised algorithms, but largely ignored practical challenges in their application. The algorithms are evaluated on performance metrics focusing on the ranking of the samples (e.g., AUC), and the ultimate choice of detecting the actual anomalies by thresholding the predictions is left for the practitioners. They lack good means for thresholding and thus often resort to using labels for such goal. This largely defeats the point of using unsupervised methods.

We presented the first practical method for estimating the posterior distribution of the contamination factor γ in a completely unsupervised manner. We empirically demonstrated on 22 datasets that our mean estimates effectively solve the question of where to threshold the predictions. We outperform all 21 comparison methods and show that the gap in detection accuracy between our estimate and the ground truth (available for these benchmark datasets) is small.

Besides solving the practical question of thresholding the predictions, we seek to transform the mindset of the anomaly detection community by presenting a fully probabilistic solution for the problem. Especially in unsupervised settings it would be completely unreasonable to expect the contamination factor could be identified exactly, but rather we need to characterize its uncertainty. However, we are not aware of any previous works even attempting this. As shown in Fig. 2,

the implied posterior distribution of γ may not only be wide but often also multi-modal. Communicating these aspects to the practitioner is critical so that they can e.g. use additional domain knowledge to interpret the possible alternatives. We showed that our estimates have near perfect calibration over the broad range of datasets used here and hence can be relied on in practical use.

On a first impression, the success of our method in solving this challenging and seemingly ill-posed problem may seem surprising. However, it can be attributed to a careful choice of strong inductive biases built into the underlying probabilistic model. We argue that all of the following elements are necessary, each substantially contributing to the overall success: (i) representing the data in the space of anomaly detector scores defines a meaning for the dimensions and allows borrowing inductive biases of arbitrary detector algorithms, (ii) the mixture model encodes a natural clustering assumption for both the normal samples and the anomalies, (iii) the ordering used for determining the final distribution incorporates both the location and shape of the mixture components in a carefully balanced manner, and (iv) the transformation from the ordering to probabilities is robustly parameterized via just two intuitive hyperparameters, enabling use of the same defaults for all cases.

Acknowledgments

This work was done during LP's research visit to University of Helsinki, funded by the Gustave Boël - Sofina Fellowship (grant V407821N). Moreover, this work is supported by (LP) the FWO-Vlaanderen aspirant grant 1166222N, (PB) the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2075 – 390740016, (AK) the Academy of Finland (grants 313125 and 336019), the Flagship program Finnish Center for Artificial Intelligence (FCAI), and the Finnish-American Research and Innovation Accelerator (FARIA).

References

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

Ritesh K Malaiya, Donghwoon Kwon, Jinoh Kim, Sang C Suh, Hyunjoo Kim, and Ikkyun Kim. An empirical evaluation of deep learning for network anomaly detection. In *2018 International Conference on Computing, Networking and Communications (ICNC)*, pages 893–898. IEEE, 2018.

Yang Xin, Lingshuang Kong, Zhi Liu, Yuling Chen, Yanmiao Li, Hongliang Zhu, Mingcheng Gao, Haixia Hou, and Chunhua Wang. Machine learning and deep learning methods for cybersecurity. *Ieee access*, 6:35365–35381, 2018.

Luis Martí, Nayat Sanchez-Pi, José Manuel Molina, and Ana Cristina Bicharra Garcia. Anomaly detection based on sensor data in petroleum industry applications. *Sensors*, 15(2):2774–2797, 2015.

Vincent Vercruyssen, Wannes Meert, Gust Verbruggen, Koen Maes, Ruben Baumer, and Jesse Davis. Semi-supervised anomaly detection with an application to water analytics. In *2018 ieee international conference on data mining (icdm)*, volume 2018, pages 527–536. IEEE, 2018.

ASAE Zaher, SDJ McArthur, DG Infield, and Y Patel. Online wind turbine fault detection through automated scada data analysis. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, 12(6):574–593, 2009.

Weizhong Yan and Lijie Yu. On accurate and reliable anomaly detection for gas turbine combustors: A deep learning approach. *arXiv preprint arXiv:1908.09238*, 2019.

Roy A Maxion and Kymie MC Tan. Benchmarking anomaly-based detection systems. In *Proceeding International Conference on Dependable Systems and Networks. DSN 2000*, pages 623–630. IEEE, 2000.

Markus Goldstein and Seiichi Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4):e0152173, 2016.

Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*, 2018.

Lorenzo Perini, Vincent Vercruyssen, and Jesse Davis. Quantifying the confidence of anomaly detectors in their example-wise predictions. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 227–243. Springer, 2020a.

Songqiao Han, Xiyang Hu, Hailiang Huang, Mingqi Jiang, and Yue Zhao. Adbench: Anomaly detection benchmark. *arXiv preprint arXiv:2206.09426*, 2022.

Fabrizio Angiulli and Clara Pizzuti. Fast outlier detection in high dimensional spaces. In *European conference on principles of data mining and knowledge discovery*, pages 15–27. Springer, 2002.

Zhaomin Chen, Chai Kiat Yeo, Bu Sung Lee, and Chiew Tong Lau. Autoencoder-based network anomaly detection. In *2018 Wireless telecommunications symposium (WTS)*, pages 1–5. IEEE, 2018.

Jonas Soenen, Elia Van Wolputte, Lorenzo Perini, Vincent Vercruyssen, Wannes Meert, Jesse Davis, and Hendrik Blockeel. The effect of hyperparameter tuning on the comparative evaluation of unsupervised anomaly detection methods. In *Proceedings of the KDD*, volume 21, pages 1–9, 2021.

Lorenzo Perini, Vincent Vercruyssen, and Jesse Davis. Class prior estimation in active positive and unlabeled learning. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence and the 17th Pacific Rim International Conference on Artificial Intelligence (IJCAI-PRICAI 2020)*, pages 2915–2921. IJCAI-PRICAI, 2020b.

Lorenzo Perini, Vincent Vercruyssen, and Jesse Davis. Transferring the contamination factor between anomaly detection domains by shape similarity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4128–4136, 2022.

Divish Rengasamy, Benjamin C Rothwell, and Grazziela P Figueredo. Towards a more reliable interpretation of machine learning outputs for safety-critical systems using feature importance fusion. *Applied Sciences*, 11(24):11854, 2021.

Jean-Marc Bardet and Solohaja-Faniaha Dimby. A new non-parametric detector of univariate outliers for distributions with unbounded support. *Extremes*, 20(4):751–775, 2017.

Damien Fourure, Muhammad Usama Javaid, Nicolas Posocco, and Simon Tihon. Anomaly detection: how to artificially increase your f1-score with a biased evaluation protocol. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 3–18. Springer, 2021.

Andrew Emmott, Shubhomoy Das, Thomas Dietterich, Alan Fern, and Weng-Keen Wong. A meta-analysis of the anomaly detection problem. *arXiv preprint arXiv:1503.01158*, 2015.

Thomas S Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973.

Carl Rasmussen. The infinite gaussian mixture model. *Advances in neural information processing systems*, 12, 1999.

Dilan Görür and Carl Edward Rasmussen. Dirichlet process gaussian mixture models: Choice of the base distribution. *Journal of Computer Science and Technology*, 25(4):653–664, 2010.

Stephen J Roberts, Dirk Husmeier, Iead Rezek, and William Penny. Bayesian approaches to gaussian mixture modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1133–1142, 1998.

Radford M Neal. Bayesian mixture modeling. In *Maximum Entropy and Bayesian Methods*, pages 197–211. Springer, 1992.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.

Steven W Nydick. The wishart and inverse wishart distributions. *Electronic Journal of Statistics*, 6(1-19), 2012.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of Markov Chain Monte Carlo*. CRC press, 2011.

Paul-Christian Bürkner and Matti Vuorre. Ordinal regression models in psychology: A tutorial. *Advances in Methods and Practices in Psychological Science*, 2(1):77–101, 2019.

Jiayu Lin. On the dirichlet distribution. *Department of Mathematics and Statistics, Queens University*, 2016.

Zhuang Qi, Dazhi Jiang, and Xiaming Chen. Iterative gradient descent for outlier detection. *International Journal of Wavelets, Multiresolution and Information Processing*, 19(04):2150004, 2021.

Ke Ren, Haichuan Yang, Yu Zhao, Wu Chen, Mingshan Xue, Hongyu Miao, Shuai Huang, and Ji Liu. A robust auc maximization framework with simultaneous outlier detection and feature selection for positive-unlabeled classification. *IEEE transactions on neural networks and learning systems*, 30(10):3072–3083, 2018.

Michael Friendly, Georges Monette, and John Fox. Elliptical insights: understanding statistical methods through elliptical geometry. *Statistical Science*, 28(1):1–39, 2013.

Alec Jacobson, Ladislav Kavan, and Olga Sorkine-Hornung. Robust inside-outside segmentation using generalized winding numbers. *ACM Transactions on Graphics (TOG)*, 32(4):1–12, 2013.

Vilijandas Bagdonavičius and Linas Petkevičius. Multiple outlier detection tests for parametric models. *Mathematics*, 8 (12):2156, 2020.

Daichi Amagata, Makoto Onizuka, and Takahiro Hara. Fast and exact outlier detection in metric spaces: a proximity graph-based approach. In *Proceedings of the 2021 International Conference on Management of Data*, pages 36–48, 2021.

LN Bol'shev and M Ubaidullaeva. Chauvenet's test in the classical theory of errors. *Theory of Probability & Its Applications*, 19(4):683–692, 1975.

Charu C Aggarwal. An introduction to outlier analysis. In *Outlier analysis*, pages 1–34. Springer, 2017.

Navid Hashemi, Eduardo Verdugo German, Jonatan Pena Ramirez, and Justin Ruths. Filtering approaches for dealing with noise in anomaly detection. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 5356–5361. IEEE, 2019.

KK Thanammal, RR Vijayalakshmi, S Arumugaperumal, and JS Jayasudha. Effective histogram thresholding techniques for natural images using segmentation. *Journal of Image and Graphics*, 2(2):113–116, 2014.

Mufda Jameel Alrawashdeh. An adjusted grubbs' and generalized extreme studentized deviation. *Demonstratio Mathematica*, 54(1):548–557, 2021.

Daniele Coin. Testing normality in the presence of outliers. *Statistical Methods and Applications*, 17(1):3–12, 2008.

Michael A Martin and Steven Roberts. An evaluation of bootstrap methods for outlier detection in least squares regression. *Journal of Applied Statistics*, 33(7):703–720, 2006.

Bijan Afsari. Riemannian l^p center of mass: existence, uniqueness, and convexity. *Proceedings of the American Mathematical Society*, 139(2):655–673, 2011.

N Archana and SS Pawar. Periodicity detection of outlier sequences using constraint based pattern tree with mad. *International Journal of Advanced Studies in Computers, Science and Engineering*, 4(6):34, 2015.

Dmitri Iouchtchenko, Neil Raymond, Pierre-Nicholas Roy, and Marcel Nooijen. Deterministic and quasi-random sampling of optimized gaussian mixture distributions for vibronic monte carlo. *arXiv preprint arXiv:1912.11594*, 2019.

Michiel A Keyzer and BGJS Sonneveld. Using the mollifier method to characterize datasets and models: the case of the universal soil loss equation. *ITC Journal*, 3(4):263–272, 1997.

Jakob Raymaekers and Peter J Rousseeuw. Transforming variables to central normality. *Machine Learning*, pages 1–23, 2021.

Guilherme O Campos, Arthur Zimek, Jörg Sander, Ricardo JGB Campello, Barbora Micenková, Erich Schubert, Ira Assent, and Michael E Houle. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data mining and knowledge discovery*, 30(4):891–927, 2016.

Lijun Zhang, Kai Liu, Yufeng Wang, and Zachary Bosire Omariba. Ice detection model of wind turbine blades based on random forest classifier. *Energies*, 11(10):2548, 2018.

Clayton Scott and Gilles Blanchard. Transductive anomaly detection. Technical report, Tech. Rep., 2008, http://www. eecs. umich. edu/cscott, 2008.

Najiba Toron, Janaina Mourão-Miranda, and John Shawe-Taylor. Transductgan: a transductive adversarial model for novelty detection. *arXiv e-prints*, pages arXiv–2203, 2022.

Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(1):1–39, 2012.

Markus Goldstein and Andreas Dengel. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: poster and demo track*, 9, 2012.

Peter J Green and Sylvia Richardson. Modelling heterogeneity with and without the dirichlet process. *Scandinavian journal of statistics*, 28(2):355–375, 2001.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Yue Zhao, Zain Nasrullah, Maciej K Hryniewicki, and Zheng Li. Lscp: Locally selective combination in parallel outlier ensembles. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 585–593. SIAM, 2019a.

Tomáš Pevnỳ. Loda: Lightweight on-line detector of anomalies. *Machine Learning*, 102(2):275–304, 2016.

Zheng Li, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu. Copod: copula-based outlier detection. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 1118–1123. IEEE, 2020.

Yue Zhao, Zain Nasrullah, and Zheng Li. Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20:1–7, 2019b.

David B Dunson and Ju-Hyun Park. Kernel stick-breaking processes. *Biometrika*, 95(2):307–323, 2008.

Supplementary Materials

This Supplement contains additional details about our method and the experiments.

6 Methodology

In this section we further explain (1) how to find the optimal values of our hyperparameters δ and τ, and (2) how to collect samples from our estimate of $\gamma|S$ posterior.

1. Finding the values of δ and τ. We introduce two new hyperparameters δ and τ that have the role of properly calibrating the sigmoid function. By setting the following equations:

$$p_0 = 1 - \mathbb{P}(c_1) = 1 - \frac{1}{1 + e^{(\tau + \delta \cdot r(\tilde{\mu}_1, \tilde{\Sigma}_1))}}$$

$$p_{\text{high}} = \mathbb{P}(\gamma \geq t|S) = \sum_{k=1}^{K} \mathbb{P}(c_1, \ldots, c_k) \mathbb{P}\left(\sum_{j=1}^{k} \pi_j \geq t|S\right)$$

we aim at finding the unique (optimal) value of δ and τ. We solve the optimization problem using the least square optimizer as implemented in SKLEARN.[2]

However, while we can always find some values such that the first condition holds (on p_0), the second one can be satisfied only if $p_{\text{high}} \geq \mathbb{P}(\pi_1 \geq t)$. Experimentally π_1 has almost always low values, which means $\mathbb{P}(\pi_1 \geq t) = 0$ for $t = 0.15$. However, in case the such a constraint cannot be satisfied, we keep running again the variational inference method (with different starting point) for the DPGMM until the constraint on p_{high} holds. If this cannot happen or does not happen within 100 iterations, we reject the possibility of too high contamination factors and just set it to 0.

2. Sampling algorithm. After creating the M-dimensional space using the M anomaly detectors, we fit the DPGMM model and obtain an approximation for the posterior $p(\pi, \mu, \Sigma|S)$. Then we derive a sample from $p(\gamma|S)$ in four steps by repeating the next operations for all $k \leq K$. First, we draw a sample $\pi_k^{(z)}, \mu_k^{(z)}, \Sigma_k^{(z)}$ from π_k (Dirichlet), μ_k (Normal), Σ_k (Inverse Wishart). Second, we transform $\pi_k^{(z)}$ by taking the cumulative sum and obtain a sample $\sum_{j=1}^{k} \pi_j^{(z)}$. Third, we pass $\mu_k^{(z)}$ and $\Sigma_k^{(z)}$ through the sigmoid function (Eq.2) to get the conditional probabilities $\mathbb{P}(c_k \mid c_{k-1})$, and transform them into the exact joint probabilities $\mathbb{P}(c_1, \ldots, c_k)$ using the equation 3. Finally, we multiply the samples following the Formula 4 and obtain a sample $\gamma^{(z)}$ from $p(\gamma|S)$.

7 Experiments

In this section we explain the threshold estimators and the ten anomaly detectors used by γGMM to assign the anomaly scores. Moreover, we provide additional details on the datasets used, and we extend the results of Q1,Q2 and Q3.

Threshold estimators. In statistics there are several methods that, given mono-dimensional scores belonging to two distributions, set a threshold to split them. These methods can be applied in our setting to estimate the contamination factor as the proportion of examples greater than the set threshold. We cluster these statistical methods in 9 groups:
1. Kernel-based. FGD [Qi et al., 2021] and AUCP [Ren et al., 2018] both use the kernel density estimator to estimate the score density; FGD exploits the inflection points of the density's first derivative, while AUCP uses the percentage of the total kernel density estimator's AUC to set the threshold;
2. Curve-based. EB [Friendly et al., 2013] creates elliptical boundaries by generating pseudo-random eccentricities, while WIND [Jacobson et al., 2013] is based on the topological winding number with respect to the origin;
3. Normality-based. ZSCORE [Bagdonavičius and Petkevičius, 2020] exploits the Z-scores, DSN [Amagata et al., 2021] measures the distance shift from a normal distribution, and CHAU [Bol'shev and Ubaidullaeva, 1975] follows the Chauvenet's criterion before using the Z-score;
4. Regression-based. CLF and REGR [Aggarwal, 2017] are two regression models that separate the anomalies based on the y-intercept value;
5. Filter-based. FILTER [Hashemi et al., 2019], and HIST [Thanammal et al., 2014] use, respectively, the wiener filter and the Otsu's method to filter out the anomalous scores;

[2] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least_squares.html`

6. Statistical test-based. GESD [Alrawashdeh, 2021], MCST [Coin, 2008] and MTT [Rengasamy et al., 2021] are based on, respectively, the generalized extreme studentized, the Shapiro-Wilk, and the modified Thompson Tau statistical tests;
7. Statistical moment-based. BOOT [Martin and Roberts, 2006] derives the confidence interval through the two sided bias-corrected and accelerated bootstrap; KARCH [Afsari, 2011] and MAD [Archana and Pawar, 2015] are based on means and standard deviations, i.e., respectively, the Karcher mean plus one standard deviation, and the mean plus the median absolute deviation over the standard deviation;
8. Quantile-based. IQR [Bardet and Dimby, 2017] and QMCD [Iouchtchenko et al., 2019] set the threshold based on quantiles, i.e., respectively, the third quartile Q_3 plus 1.5 times the inter-quartile region $|Q_3 - Q_1|$, and the quantile of one minus the Quasi-Monte Carlo discreprancy;
9. Transformation-based. MOLL [Keyzer and Sonneveld, 1997] smooths the scores through the Friedrichs' mollifier, while YJ [Raymaekers and Rousseeuw, 2021] applies the Yeo-Johnson monotonic transformations.

Anomaly Detectors. We use 10 anomaly detectors with different inductive biases: KNN [Angiulli and Pizzuti, 2002] assumes that the anomalies are far away from normals, IFOREST [Liu et al., 2012] assumes that the anomalies are easier to isolate, LOF [Breunig et al., 2000] exploits the examples' density, OCSVM [Green and Richardson, 2001] encapsulates the data into a multi-dimensional hypersphere, AE [Chen et al., 2018] and VAE [Kingma and Welling, 2013] use the reconstruction error as anomaly score function in a, respectively, deterministic and probabilistic perspective, LSCP [Zhao et al., 2019a] is an ensemble method that selects competent detectors locally, HBOS [Goldstein and Dengel, 2012] calculates the degree of anomalousness by building histograms, LODA [Pevnỳ, 2016] is an ensemble of weak detectors that build histograms on randomly generated projected spaces, and COPOD [Li et al., 2020] is a copula based method. All these methods are implemented in the python library PyOD [Zhao et al., 2019b].

Table 1: Properties of the 22 datasets used. For each dataset, we report the number examples, the number of original covariates, and the ground-truth contamination factor.

Dataset	# Examples (N)	# Covariates	True γ^*
ALOI	12384	27	0.0304
Annthyroid	7129	21	0.0749
Arrhythmia	271	259	0.0996
Cardiotocography	1734	21	0.0496
Glass	214	7	0.0421
InternetAds	1682	1555	0.0499
KDDCup99	48113	40	0.0042
Lymphography	148	47	0.0405
PageBlocks	5473	10	0.1023
Parkinson	53	22	0.0943
PenDigits	9868	16	0.0020
Pima	526	8	0.0494
Shuttle	1013	9	0.0128
SpamBase	2661	57	0.0500
Stamps	340	9	0.0912
T15	42125	10	0.0668
T21	18509	10	0.0529
WBC	223	9	0.0448
WDBC	367	30	0.0272
WPBC	160	33	0.0562
Waveform	3443	21	0.0290
Wilt	4655	5	0.0200

Data. Table 1 shows the details of the used datasets. The datasets vary in terms of number of examples (from around 50 to more than 48000), number of covariates (from 5 to more than 1500) and the contamination factor (from around 0.004 to more than 0.10). Note that even the highest contamination factor is around 0.10, confirming the general assumption of anomalies being rare.

Q1-Q2. γGMM's estimated distribution. Figure 8 shows our estimate of $\gamma|S$'s posterior on the 22 used datasets. Moreover, Table 2 shows the MAE between γGMM's sample mean and the true value γ^* on a per-dataset basis. With respect to the true value γ^* and out of 22 experiments, the sample mean is:

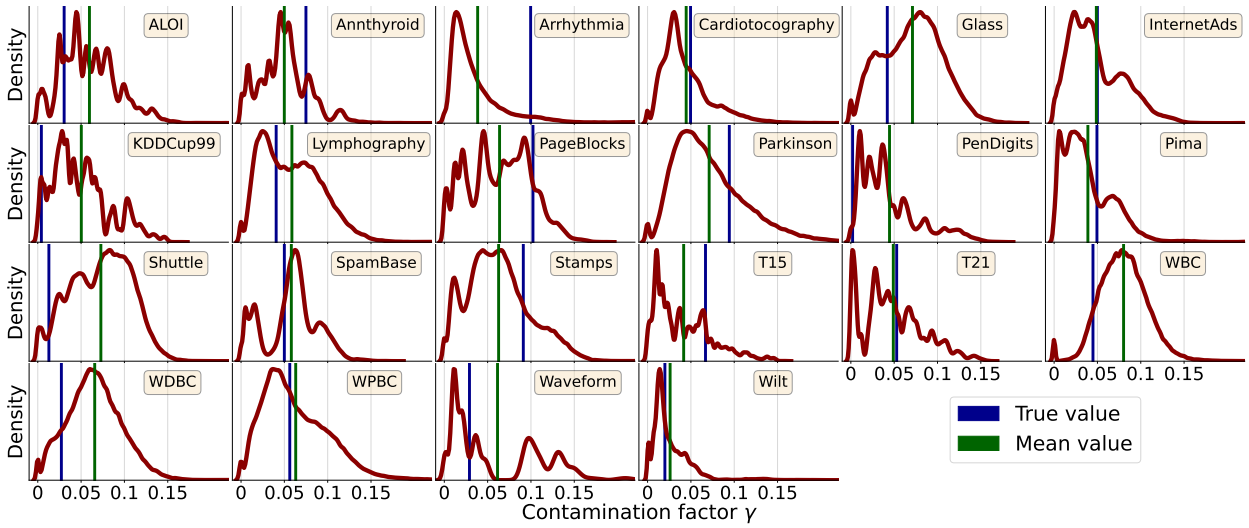

Figure 8: Illustration of how γGMM estimates γ's posterior on the 22 datasets.

Table 2: Mean Absolute Error (MAE) between the true contamination factor and γGMM's sample mean for the 22 datasets.

Dataset	γGMM's sample mean	True γ^*	MAE
ALOI	0.0596	0.0304	0.0292
Annthyroid	0.0499	0.0749	0.0250
Arrhythmia	0.0385	0.0996	0.0611
Cardiotocography	0.0446	0.0496	0.0050
Glass	0.0711	0.0421	0.0290
InternetAds	0.0487	0.0499	0.0012
KDDCup99	0.0502	0.0042	0.0460
Lymphography	0.0587	0.0405	0.0182
PageBlocks	0.0638	0.1023	0.0385
Parkinson	0.0711	0.0943	0.0232
PenDigits	0.0446	0.0020	0.0426
Pima	0.0390	0.0494	0.0104
Shuttle	0.0728	0.0128	0.0600
SpamBase	0.0580	0.0500	0.0080
Stamps	0.0627	0.0912	0.0285
T15	0.0417	0.0668	0.0251
T21	0.0490	0.0529	0.0039
WBC	0.0802	0.0448	0.0354
WDBC	0.0657	0.0272	0.0385
WPBC	0.0631	0.0562	0.0069
Waveform	0.0614	0.0290	0.0324
Wilt	0.0260	0.0200	0.0060

- a *good estimate* (i.e., MAE ≤ 0.01) for 7 datasets (Cardiotocography, InternetAds, Pima, SpamBase, T21, WPBC, Wilt);
- a *slightly imprecise estimate* (i.e., $0.01 <$ MAE ≤ 0.03) for 7 datasets (ALOI, Annthyroid, Glass, Lymphography, Parkinson, Stamps and T15);
- a *not-optimal estimate* (i.e., $0.03 <$ MAE ≤ 0.05) for 6 datasets (KDDCup99, PageBlocks, PenDigits, WBC, WDBC, and Waveform);
- a *bad estimate* (MAE > 0.05) for just two datasets (Arrhythmia, and Shuttle).

This shows, again, that the estimated distribution is well-calibrated.

Table 3: List of detectors with the greatest F_1 score when using the true contamination factor to set the threshold. For each dataset, we use such subset of detectors to compute the deterioration.

Dataset	Anomaly Detectors
ALOI	KNN
Annthyroid	HBOS
Arrhythmia	IForest-HBOS-COPOD
Cardiotocography	KNN
Glass	LOF
InternetAds	LSCP
KDDCup99	COPOD
Lymphography	KNN-LOF-OCSVM-HBOS
PageBlocks	LOF
Parkinson	LSCP-HBOS-COPOD-LSCP-HBOS-COPOD
PenDigits	KNN-IForest-LOF-OCSVM-LSCP-Ae-VAE-HBOS-LODA-COPOD
Pima	IForest
Shuttle	KNN-OCSVM-Ae-VAE-HBOS-KNN-OCSVM-Ae-VAE-HBOS
SpamBase	LSCP
Stamps	LSCP
T15	OCSVM
T21	OCSVM
WBC	KNN-LOF-OCSVM-LODA-COPOD
WDBC	KNN-LOF-OCSVM-LSCP-Ae-VAE-LODA-COPOD
WPBC	OCSVM
Waveform	OCSVM
Wilt	LOF

Q3. Selecting the anomaly detectors to compute the F_1 score. Because we aim at studying the effect of the contamination factor on the detectors' performance, we compare the F_1 scores only over the detectors that work well for each of the dataset. For each dataset D, we use as set of detectors those achieving the greatest F_1 score using the true contamination factor, i.e. $\arg\max_{f_m} \{F_1(f_m, D, \gamma^*)\}$. This means that, for each dataset, we (1) use each detector separately to make predictions using the true contamination factor γ^*, (2) measure their F_1 score, (3) keep those detectors that obtains the greatest F_1, and (4) use them to compute the F_1 deterioration using the point-estimates of the contamination factor. Table 3 lists the detectors used for each dataset to compute the F_1 deterioration. Observe that sometimes only a single detector obtains the greatest F_1 score, while sometimes several detectors get the same F_1 score.

Q3. False alarms and false negatives. Finally, Table 4 shows the false alarm (false positive) rate and the false negative rate. The majority of the threshold estimators provide extremely high estimates for the contamination factor, shown here as extremely low false negative rates, but they would be useless in practice because of their high false alarm rate. In fact, this metric is important as false alarms result in real costs for the company (e.g., turning the wind turbine off to wait until the ice on the blades melts down), while reducing trust in the detection system. Our method reduces the false alarm rate compared to most of the baselines, including QMCD and KARCH that achieve the second and third best F_1 scores on average. On the other hand, IQR and MTT have the lowest false positive rates, due to the fact that they often underestimate the contamination factor as supported by the false negative table.

Table 4: Mean and standard deviation of the false alarm rate (left) and false negative rate (right) obtained by using each method's γ estimate to set the threshold (the lower the better). Regarding the false alarms, γGMM has the third best mean and outperforms QMCD and KARCH, which are the second and third best baseline when measuring the F_1 score. On the other hand, γGMM obtains higher false negative rates than almost all the competitors, due to the fact that the threshold estimators overestimate the true contamination factor.

False Alarm Rate	
Method	Mean $\pm$ std.
IQR	0.009 ± 0.008
MTT	0.027 ± 0.024
γGMM	**0.042±0.015**
QMCD	0.059 ± 0.018
KARCH	0.147 ± 0.047
CHAU	0.190 ± 0.035
ZSCORE	0.221 ± 0.050
YJ	0.390 ± 0.139
FILTER	0.454 ± 0.054
DSN	0.477 ± 0.134
HIST	0.513 ± 0.100
FGD	0.533 ± 0.183
AUCP	0.591 ± 0.088
MCST	0.611 ± 0.287
GESD	0.616 ± 0.106
REGR	0.643 ± 0.105
MAD	0.731 ± 0.083
CLF	0.757 ± 0.077
EB	0.785 ± 0.077
WIND	0.809 ± 0.076
MOLL	0.816 ± 0.082
BOOT	0.862 ± 0.079

False Negative Rate	
Method	Mean $\pm$ std
BOOT	0.001 ± 0.002
WIND	0.001 ± 0.002
MOLL	0.001 ± 0.002
EB	0.001 ± 0.003
MAD	0.002 ± 0.003
CLF	0.002 ± 0.003
GESD	0.003 ± 0.005
REGR	0.003 ± 0.005
AUCP	0.004 ± 0.006
HIST	0.006 ± 0.008
FGD	0.007 ± 0.011
DSN	0.007 ± 0.009
FILTER	0.007 ± 0.009
MCST	0.007 ± 0.012
YJ	0.009 ± 0.010
ZSCORE	0.017 ± 0.015
CHAU	0.019 ± 0.016
KARCH	0.021 ± 0.018
QMCD	0.034 ± 0.024
γGMM	**0.036 $\pm$ 0.025**
MTT	0.042 ± 0.028
IQR	0.044 ± 0.029