Quantifying the Confidence of Anomaly Detectors in Their Example-Wise Predictions

Lorenzo Perini[0000−0002−5929−9727] (🖂), Vincent Vercruyssen[0000−0003−3645−3135] (🖂), and Jesse Davis[0000−0002−3748−9263] (🖂)

DTAI Research Group & Leuven.AI, KU Leuven, Belgium
firstname.lastname@kuleuven.be

Abstract. Anomaly detection focuses on identifying examples in the data that somehow deviate from what is expected or typical. Algorithms for this task usually assign a score to each example that represents how anomalous the example is. Then, a threshold on the scores turns them into concrete predictions. However, each algorithm uses a different approach to assign the scores, which makes them difficult to interpret and can quickly erode a user's trust in the predictions. This paper introduces an approach for assessing the reliability of any anomaly detector's example-wise predictions. To do so, we propose a Bayesian approach for converting anomaly scores to probability estimates. This enables the anomaly detector to assign a confidence score to each prediction which captures its uncertainty in that prediction. We theoretically analyze the convergence behaviour of our confidence estimate. Empirically, we demonstrate the effectiveness of the framework in quantifying a detector's confidence in its predictions on a large benchmark of datasets.

Keywords: Anomaly detection · Interpretability · Confidence scores.

1 Introduction

Anomaly detection is a central task in data mining. It involves identifying portions of the data that do not correspond to expected normal behaviours. From a practical point of view, anomaly detection is important as anomalies often have significant costs in the real world. For example, fraudulent credit card transactions [2], retail store water leaks [18], or abnormal web traffic [15].

Typically, anomaly detection is tackled from an unsupervised perspective due to the costs and difficulties associated with acquiring labels for the anomaly class (e.g., you will not allow expensive equipment to breakdown simply to observe how it behaves in an anomalous state). The underlying assumption to these approaches is that anomalies are both (i) rare and (ii) somehow different from normal examples. Hence, anomalies may lie in low-density regions of the instance space or be far away from most other examples. The algorithms use these intuitions to assign a real-valued score to each example that denotes how anomalous an example is. Usually, these anomaly scores are converted to binary predictions (an example is normal or anomalous) by setting a threshold on the

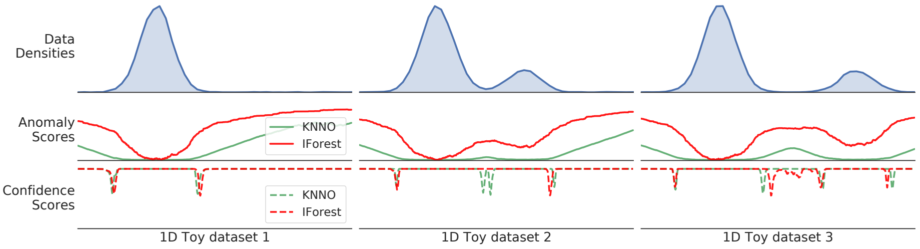

Fig. 1. Illustration of why confidence scores are important on three 1D toy datasets. The top plots show the data distributions under small perturbations. The middle plots show the anomaly scores assigned by Knno and iForest. These two models produce non-standard scores, which are difficult to interpret and compare. The bottom plots show the corresponding confidence scores computed using our method (see Section 3). Small changes in the data distribution affect anomaly scores and predictions. The confidence scores capture clearly where the models (dis)agree. The dips in the confidence scores correspond to a transition in the predicted label of the underlying model.

scores. However, the scores for many prominent approaches, such as Knno, iForest, and Ssdo are difficult for a human to interpret and compare. A natural way to address this issue is to transform the anomaly scores into a probability estimate. The standard approach is to *calibrate* the transformation such that for all examples that are predicted to have a $c\%$ probability of belonging to the anomaly class, $c\%$ of them should actually be anomalies. A user can now ignore any predictions with a low chance of being an anomaly. However, calibration requires labels which are generally not available in an anomaly detection setting.

The fact that anomalies are rare and unpredictable causes another issue. Hypothetically, even if we could collect multiple datasets, each one would contain distinct anomalies to which the anomaly detectors would assign different scores. Additionally, small perturbations in the training data might cause (large) differences in an example's anomaly score and, consequently, a different prediction. Consider the three one-dimensional toy datasets in Figure 1. The middle row plots show the continuous anomaly scores that Knno and iForest assign to each example in the distributions. These scores change as a result of small perturbations in the dataset, ultimately resulting in different predictions.

This paper tackles this challenge by providing a measure of how *uncertain* an anomaly detector's predictions are on an example-wise basis. The measure will allow a user to assess the reliability of anomaly detectors in different scenarios. We make the following four contributions. First, we propose a notion of a confidence measure that captures how consistent a model's prediction would be for that example if the training data were perturbed. Second, we propose ExCeeD (*EXample-wise ConfidEncE of anomaly Detectors*), an approach that is able to compute our confidence measure for any anomaly detector that produces a real-valued anomaly score. The method begins by transforming the anomaly scores to *outlier probabilities* using a Bayesian approach. Then, it uses these probabilities to derive the example-wise confidence scores. This is illustrated in the bottom plots of Figure 1, which show the computed confidence scores for Knno's and iForest's prediction that each example is anomalous. The scores show in an interpretable way where the algorithms disagree and where they are uncertain about their prediction. Third, we perform a theoretical analysis of the convergence behaviour of our confidence estimates. Fourth, we perform an extensive empirical evaluation on 21 benchmark datasets.

2 Related Work

2.1 Assigning Anomaly Scores

Several different assumptions underpin anomaly detectors, but all exploit the fact that anomalies are rare and different than normal examples. From a geometric perspective, this means that anomalies are far away from normal examples. From a statistical perspective, this means that anomalies will fall in a low-density region of the instance space. Although any model producing scores can be used, we briefly describe three canonical unsupervised anomaly detection algorithms.

kNNO assigns an anomaly score based on the k-distance, which is the distance between an example and its k'th nearest neighbor [14]. Examples far away from other examples get high scores indicating they are more anomalous.

iForest assigns an anomaly score based on how difficult it is to isolate an example from the rest of the data by iteratively splitting the data along random dimensions [9]. Examples in low-density regions get higher scores.

OCSVM assigns an anomaly score based on the signed distance to the surface of the hypersphere encapsulating the normal examples. Examples outside the sphere get high scores indicating their anomalousness [16].

Because each algorithm produces a score in a completely different way, cross-comparisons between the algorithms are difficult. Consider the example of Figure 1, where sometimes Knno predicts anomalies while iForest does not, even though the Knno scores are consistently lower.

2.2 From Anomaly Scores to Outlier Probabilities

A challenge with the anomaly scores produced by the aforementioned methods (as well as many others) is that it is difficult to interpret them. For example, understanding the k-distance requires context or domain knowledge (e.g., the number features, what constitutes a big distance, etc.). Therefore, a possible solution is to convert the anomaly score of an example to a probability estimate. The standard approach is to employ Platt scaling [13]:

$$\mathbb{P}(Y = 1 | S = s) = \frac{1}{1 + exp(\alpha \times s + \beta)}$$

where Y is the true class for an example with anomaly score s, and α and β are parameters that should be learned from the labeled data. Ideally, such a transformation should produce calibrated probability estimates. Intuitively, a probability $\mathbb{P}(Y = 1|S = s) = c$ is calibrated if out of all examples with this probability, about c percent of these examples are member of the positive class.

However, in most anomaly detection applications we lack labeled data with which to train such a calibration model. We typically only know the contamination factor γ which is the proportion of anomalies in the data. Hence, the standard approach is to calibrate the transformation such that γ percent of the examples have a probability > 0.5. Beyond the logistic calibration approach, there is a long literature of approaches [5,7,8,10,11,17] for ensuring that this property is obtained and we now briefly describe some prominent approaches. Isotonic Calibration [20] is a non-parametric form of regression in which the transformation function is chosen from the class of all non-decreasing functions. Beta Calibration [8] is based on the assumption that scores are Beta distributed class-wise and transforms them according to the likelihood rate.

In anomaly detection, three methods are widely used to get calibrated outlier probabilities. The linear and squashing methods map the scores to probabilities using respectively a linear and a sigmoid transformation [4]. The unify method assumes that scores are normally distributed and estimates the outlier probability through the Gaussian cumulative distribution function [6].

3 A Theoretical Framework for Assessing an Anomaly Detector's Example-Wise Confidence

Using an anomaly detector in practice requires converting its returned anomaly score into a hard prediction. Typically, this is done by setting a threshold λ on the scores. Then, any example x with a score $s > \lambda$ will be classified by the model as an anomaly. Standard approaches [9,14,16] set a threshold by analyzing the data used to train the model. Hence, perturbing the training data would lead to a different threshold being picked, which in turn would affect an example's predicted class.

To capture this potential uncertainty, we propose a notion of a detector's example-wise confidence in its predictions, which works with any anomaly detector producing a real-valued scores. Intuitively, we can think of the example-wise confidence as the probability that a detector's prediction would change if a different dataset was observed. More formally, we define it as follows.

Definition 1 (Example-wise Confidence). *Let $f\colon \mathbb{R}^d \to \mathbb{R}$ be a function that maps examples to anomaly scores. Given a dataset D such that $|D| = n$, the model's confidence in its prediction for an example x with anomaly score $s = f(x)$ can be defined as*

$$\mathscr{C}(\hat{Y})_x = \begin{cases} \mathbb{P}(\hat{Y} = 1 \,|\, s, n, \gamma, \hat{p}_s) & \text{if } \hat{Y} = 1 \\ 1 - \mathbb{P}(\hat{Y} = 1 \,|\, s, n, \gamma, \hat{p}_s) & \text{if } \hat{Y} = 0 \end{cases} \tag{1}$$

where $\hat{Y}$ is the class label predicted by the anomaly detector, $\hat{p}_s$ is the estimated outlier probability (i.e., the probability that the example belongs to the anomaly class), and γ is the expected proportion of positive examples.

From now on, when we use the term *confidence* we will refer to $\mathbb{P}(\hat{Y} = 1 \mid s, n, \gamma, \hat{p}_s)$, as the case when $\hat{Y} = 0$ is directly computable from the previous one. Hence, when $\hat{Y} = 1$, high values of $\mathbb{P}(\hat{Y} = 1 \mid s, n, \gamma, \hat{p}_s)$ indicate that model is confident in its prediction that the example is an anomaly. One would expect confidence values around 0.5 when an example is near the decision boundary, that is on the border between normal and abnormal behaviors. We estimate the confidence in two steps. First, we employ a Bayesian approach to estimate the distribution of anomaly scores. This allows us to derive an example's outlier probability $\hat{p}_s$. Second, we use the outlier probability to estimate the confidence of the anomaly detector by considering how the combination of the observed training set and contamination factor γ would be used to select the threshold λ for converting anomaly scores into predictions.

3.1 Notation

Let $(\Omega, \Im, \mathbb{P})$ be a probability space, where Ω is the sample space, $\Im$ represents a σ-algebra over Ω and $\mathbb{P}$ is a probability measure. Let $X\colon \Omega \to \mathbb{R}^d$ be a multivariate real random variable with values in the feature space $\mathbb{R}^d$, and $Y\colon \Omega \to \{0,1\}$ be a random variable identifying the class label. Assume that D is an available dataset, which can be seen as an i.i.d. sample of size n drawn from the joint distribution of X and Y. An anomaly detection problem is the setting where there exists a function $f\colon \mathbb{R}^d \to \mathbb{R}$ that maps the feature points from the dataset D to a single real value called anomaly score. A common assumption is that the function f is measurable, so that $S = f(X)$ is a real random variable with anomaly scores as values. From now on, we will use the notation D_n referring to the dataset of scores $D_n = \{s_1, \ldots, s_n\} = \{f(x_1), \ldots, f(x_n)\}$, with $x_1, \ldots, x_n \in D$. Given an Anomaly Detector Γ, for any example x we define the *outlier probability of* x as the probability that x is anomalous according to its anomaly score $s = f(x)$

$$\mathbb{P}(Y = 1 | f(X) = f(x)) = \mathbb{P}(Y = 1 | S = s) \coloneqq \mathbb{P}(S \leq s), \tag{2}$$

where f is the function provided by Γ. Subsequently, the probability that one example is normal can be computed as

$$\mathbb{P}(Y = 0 | f(X) = f(x)) = 1 - \mathbb{P}(Y = 1 | S = s) = \mathbb{P}(S > s).$$

3.2 A Bayesian Approach for Assigning an Outlier Probability

Our goal is to infer the true class label based on the anomaly score. Formally, for any score $s \in \mathbb{R}$, we can model the example's true class given the score as the conditional random variable $Y|S = s$. Based on this framework, we can estimate an example's *outlier probability* (i.e., the probability it belongs to the anomaly class) as follows:

$$P_s \coloneqq \mathbb{P}((Y|S) = 1|s) = \mathbb{P}(Y = 1|S = s) = \mathbb{P}(S \leq s).$$

Because $Y|S = s$ takes values in the set $\{0, 1\}$, we model its outcome using a Bernoulli distribution:

$$Y|S = s \sim Bernoulli(P_s)$$

where P_s is the probability of success. If we knew S's distribution, we could compute $\mathbb{P}(S \leq s)$–the probability that an example belongs to the anomaly class–using the cumulative distribution function of S. Unfortunately, the distribution of S is usually unknown which makes it infeasible to directly approximate P_s.

Our solution is to take a Bayesian approach to this problem. The key insight is to measure the area of $\{S < s\}$ by drawing samples from the real distribution. We will view P_s as a random variable and assume a uniform prior. Theoretically, we can derive the probability that an example belongs to the anomaly class as follows. First, we draw one example a from the distribution of S, which simply entails drawing an example x from X and computing its anomaly score $a = f(x)$. Second, we record the event as a success (i.e., $b = 1$) if $a \leq s$ and as failure (i.e., $b = 0$) otherwise. We repeat the process n times and record the total number of successes as t and failures as $n - t$. In fact, the rate between successes and trials, corrected with other factors, will approximate the outlier probability as defined in formula 2. Thanks to Bayes's rule we can use the following theorem.

Theorem 2. *Assume that a random variable P_s follows a Beta distribution $\mathcal{B}eta(\alpha, \beta)$ as prior. Given the events $b_1, \ldots, b_n$, which are i.i.d. examples drawn from a Bernoulli random variable $Bernoulli(P_s)$, then the posterior distribution of P_s is still a Beta distribution with new parameters $\mathcal{B}eta\left(\alpha + t,\, \beta + n - t\right)$, where $t = \sum_{i=1}^{n} b_i$ is the number of successes.*

Proof. According to the hypotheses, the prior distribution of P_s is

$$\pi(q) = \frac{q^{\alpha-1}(1-q)^{\beta-1}}{\mathcal{B}(\alpha, \beta)},$$

where $\mathcal{B}(\alpha, \beta)$ is the Euler beta function. So, by using the Bayes's rule

$$\begin{aligned}\pi(q|b_1, \ldots, b_t) &= \frac{\pi(q) \cdot \mathbb{P}(b_1, \ldots, b_t|q)}{\int_0^1 \pi(r) \cdot \mathbb{P}(b_1, \ldots, b_t|r)\, dr} = \frac{\frac{q^{\alpha-1}(1-q)^{\beta-1}}{\mathcal{B}(\alpha,\beta)} \cdot q^t\, (1-q)^{n-t}}{\int_0^1 \frac{r^{\alpha-1}(1-r)^{\beta-1}}{\mathcal{B}(\alpha,\beta)} \cdot r^t\, (1-r)^{n-t}\, dr} \\ &= \frac{q^{\alpha+t-1}\, (1-q)^{\beta+n-t-1}}{\int_0^1 r^{\alpha+t-1}\, (1-r)^{\beta+n-t-1}\, dr} = \mathcal{B}eta\left(\alpha + t,\, \beta + n - t\right)\end{aligned}$$

where $t = \sum_{i=1}^{n} b_i$, $\pi(q|b_1, \ldots, b_n)$ is the posterior distribution of P_s after i.i.d. sampling n $Bernoulli(P_s)$ examples, and $\mathbb{P}(b_1, \ldots, b_n|q)$ is the likelihood. □

In our setting we assume that $P_s \sim \mathcal{B}eta(1, 1) = Unif(0, 1)$. As a result, the posterior distribution of P_s is still a Beta distribution

$$P_s|b_1, \ldots, b_n \sim \mathcal{B}eta\left(1 + t, 1 + n - t\right). \tag{3}$$

In order derive an estimate of the outlier probability from P_s, we take the expectation of P_s. Since the posterior distribution is known from (3), $\mathbb{E}[P_s]$ can be obtained as a function of the parameters:

$$\hat{p}_s \coloneqq \mathbb{E}[P_s] = \frac{1+t}{2+n}. \tag{4}$$

In practice we cannot sample from the true distribution and instead need to use the dataset D_n to infer the posterior distribution. Thus, when drawing an example, we are restricted to sampling from the dataset. This limits us to drawing n examples, that is, the total number of examples in the dataset. An additional consideration concerns the value of t. It represents the number of successes when sampling from $Y|S = s \sim Bernoulli(P_s)$. As a result, t is a practical approximation of the real percentage θ of successes times the number of trials n

$$t \approx \theta \cdot n. \tag{5}$$

The reason why we use t is to obtain a corrected estimate of the real parameter θ, which would be the exact probability value of $Y|S = s$ if it were known.

3.3 Deriving a Detector's Confidence in its Predictions

Although the second step of our framework works with any approach that converts anomaly scores into outlier probabilities, here $\hat{p}_s$ refers to the definition in Equation 4. Deriving the confidence value requires estimating the proportion of times that an example will be predicted as being anomalous by the chosen anomaly detection algorithm. This requires analyzing how to set the threshold λ for converting anomaly scores to predictions. Typically, anomaly detectors exploit the contamination factor γ to pick the threshold λ. Note that γ may be known from domain knowledge (e.g., historical anomaly rates) or it can be estimated from partially labeled data (c.f. [12]). There are two different scenarios:

$\gamma \in (0, 1)$**: The training set contains some anomalies.** Here, the standard approach is to compute the expected number of anomalies in the training set as $k = \gamma \times n$.[1] Then, it ranks the training examples by their anomaly scores and sets the threshold to be the value in position k.

$\gamma = 0$**: The training set contains only normal examples.** In this case, the threshold has to be equal to the maximum anomaly score in the training set. Picking a lower value would result in a false positive on the training data.

In both cases the chosen threshold depends on the distribution of anomaly scores in the training set. In turn, these scores depend on the available data sample. That is, if we drew another training set from the population, the chosen threshold may change. This leads to our key insight: the task of measuring the model's confidence can be formulated as estimating the probability that an example with score s will be classified as an anomaly based on a theoretical

[1] We assume that $k \in \mathbb{N}$, taking the floor function when needed.

sample D_n drawn from the population. Formally, given the training set size n and the contamination factor γ, we want to compute the probability that an example x with score $s = f(x)$ and outlier probability $\hat{p}_s$ will be classified as an anomaly when randomly drawing a training set of n examples from the population of scores. In practice, the confidence can be seen as the probability that the chosen threshold λ value will be less than or equal to the score s. This probability depends on our two cases for picking the threshold:

Contamination factor $\gamma \in (0, 1)$. In this case by drawing theoretically from the Bernoulli distribution of $Y|S = s$ with parameter P_s, we should get at least $n - k + 1$ successes to classify s as anomaly, where $k = \gamma \times n$ and "success" means that the drawn value is lower than s. As a result, our confidence is defined as:

$$\mathbb{P}(\hat{Y} = 1 \mid s, n, \gamma, \hat{p}_s) = \sum_{i=n(1-\gamma)+1}^{n} \binom{n}{i} \hat{p}_s^i (1 - \hat{p}_s)^{n-i} \tag{6}$$

where $\hat{p}_s = \mathbb{E}[P_s]$, which is estimated using our Bayesian approach for computing an example's outlier probability (see Equation 4). Hence, our confidence estimate explicitly relies on our outlier probability.

Contamination factor $\gamma = 0$. In this case, the threshold is the maximum score in the training set, $\lambda = \max\{s_i\}_{i=1}^n$. We need to compute the probability that an example with score s and outlier probability $\hat{p}_s$ will be classified as an anomaly when randomly drawing n examples from the normal population. It is quite similar to the previous case, with the only difference being that no failures[2] are allowed. So, when $\gamma = 0$ we need to denote the confidence as

$$\mathbb{P}(\hat{Y} = 1 \mid s, n, 0, \hat{p}_s) = (\hat{p}_s)^n \tag{7}$$

where again $\hat{p}_s = \mathbb{E}[P_s]$ comes from our Bayesian estimate of the example's outlier probability (see Equation 4.)

4 Convergence Analysis of our Confidence Estimate

This section analyzes the behaviour of our confidence estimate. In particular, given a fixed anomaly score s for a test example, we want investigate how our confidence in the model's prediction changes as the number of training examples tends towards infinity. We would expect that as the size of the training set increases, our confidence estimate should converge. Again, we analyze the two cases based on whether or not the training set contains any anomalies.

[2] A failure would correspond to an training example having a higher anomaly score than the chosen threshold. Given the assumption that all training examples are normal, this would indicate a false positive.

4.1 Convergence Analysis when $\gamma \in (0, 1)$

In this case, when we set the threshold based on a fixed dataset D_n, we can derive our confidence for a test example with score s by merging Eq. 4 and 6:

$$\mathbb{P}(\hat{Y} = 1 \mid s, n, \gamma, \hat{p}_s) = \sum_{i=n(1-\gamma)+1}^{n} \binom{n}{i} \left(\frac{1+t}{2+n}\right)^{i} \left(\frac{1+n-t}{2+n}\right)^{n-i} \tag{8}$$

where t represents the successes in the Bayesian learning phase (section 3.2).

This leads to the question: how does the confidence about the class prediction for a score s behave as the number of training examples n goes towards $+\infty$? In order to formally analyze this, we rewrite Formula 8 as

$$\mathbb{P}(\hat{Y} = 1 \mid s, n, \gamma, \hat{p}_s) = F_T(n) - F_T(n - \gamma n)$$

where the sum in Equation 8 is the cumulative distribution of a binomial random variable $T \sim \mathcal{B}(n, \gamma n, \hat{p}_s)$ with n trials, γn successes, and probability $p = \hat{p}_s$. When n increases, the central limit theorem yields:

$$\mathbb{P}\left(\frac{T - n\,\hat{p}_s}{\sqrt{n\,\hat{p}_s(1-\hat{p}_s)}} \leq c\right) \to \Phi(c) \qquad \text{for } n \to +\infty,\ \forall c \in \mathbb{R}.$$

Consequently, assuming that n is large enough, we assert that, $\forall c \in \mathbb{R}$,

$$\mathbb{P}\left(T \leq c\right) = F_T(c) \approx \Phi\left(\frac{c - n\,\hat{p}_s + 0.5}{\sqrt{n\,\hat{p}_s(1-\hat{p}_s)}}\right),$$

where $+0.5$ is a correction due to the continuity of the Gaussian variable. Thus, the confidence can be approximated by the cumulative distribution function of a Gaussian variable T^* with mean $\mu = n \cdot \hat{p}_s + 0.5$ and variance $\sigma^2 = n \cdot \hat{p}_s(1 - \hat{p}_s)$,

$$T^* \sim \mathcal{N}(n\,\hat{p}_s + 0.5,\ n\,\hat{p}_s(1-\hat{p}_s)).$$

Therefore:

$$\begin{aligned} \mathbb{P}(\hat{Y} = 1 \mid s, n, \gamma, \hat{p}_s) &= F_T(n) - F_T(n - \gamma n) \\ &\approx \Phi\left(\frac{n - n\,\hat{p}_s + 0.5}{\sqrt{n\,\hat{p}_s(1-\hat{p}_s)}}\right) - \Phi\left(\frac{n(1-\gamma) - n\,\hat{p}_s + 0.5}{\sqrt{n\,\hat{p}_s(1-\hat{p}_s)}}\right) \\ &= \mathbb{P}\left(n(1-\gamma) \leq T^* \leq n\right) = \mathbb{P}\left((1-\gamma) \leq \frac{T^*}{n} \leq 1\right). \end{aligned}$$

Next, we analyze the behaviour of $\frac{T^*}{n}$ as $n \to \infty$ in order to interpret the final result. Since $n \in \mathbb{N}$, it still follows a normal distribution with new parameters:

$$\begin{aligned} \mathbb{E}\left[\frac{T^*}{n}\right] &= \frac{1}{n}\mathbb{E}[T^*] = \hat{p}_s - \frac{1}{2n} = \frac{1+t}{2+n} - \frac{1}{2n}; \\ \mathrm{Var}\left[\frac{T^*}{n}\right] &= \frac{1}{n^2}\mathrm{Var}[T^*] = \frac{\hat{p}_s(1-\hat{p}_s)}{n} = \frac{1+n-t}{n(2+n)}. \end{aligned}$$

Since the mean and the variance of $\frac{T^*}{n}$ are bounded (they both are decreasing sequences when n increases), the sequence of Gaussian random variables $\frac{T^*}{n}$ converges in distribution to a Gaussian random variable parameterized by the limit of the mean and the limit of the variance, respectively:

$$\lim_{n\to\infty} \mathbb{E}\left[\frac{T^*}{n}\right] = \theta, \quad \lim_{n\to\infty} \mathrm{Var}\left[\frac{T^*}{n}\right] = 0,$$

where θ is defined in Equation 5. Hence, when $n \to +\infty$, the limit random variable is normally distributed with mean θ and variance 0, which means the only value it assumes is θ, the true outlier probability (see the end of section 3.2). Formally, calling the limit degenerate random variable θ^*,

$$\lim_{n\to\infty} \mathbb{P}(\hat{Y} = 1 \,|\, s, n, \gamma, \hat{p}_s) = \mathbb{P}\left((1-\gamma) \leq \theta^* \leq 1\right).$$

Intuitively, this means that as the number of training examples goes to infinity the population is perfectly estimated and represented by the sample, which yields two cases. In the first case, the true outlier probability θ is greater than $1-\gamma$. Roughly speaking, the expected proportion of normal examples $(1-\gamma)$ is not high enough to yield a threshold value less than the considered score s. Hence, the confidence will be 1, because $\mathbb{P}\left((1-\gamma) \leq \theta^* \leq 1\right) = 1$ since θ^* takes the constant value θ and the inequalities are satisfied. In contrast, in the second case the inequalities are not respected, meaning that θ does not fall inside the interval $[1-\gamma, 1]$. Hence, in this scenario the proportion of normal examples is such that the value of the threshold must be greater than s and, as a result, the confidence is 0 when predicting class 1.

At the end, the limit of the confidence that s is predicted to be an anomaly is

$$\lim_{n\to\infty} \mathbb{P}(\hat{Y} = 1 \,|\, s, n, \gamma, \hat{p}_s) = \begin{cases} 1 & \text{if } \theta \geq 1-\gamma; \\ 0 & \text{if } \theta < 1-\gamma. \end{cases}$$

This corresponds to our intuition of what should occur when given an infinite number of training examples.

4.2 Convergence Analysis when $\gamma = 0$

In this analysis, the main hypothesis is that the training set only contains normal examples, which corresponds to learning a one-class model. Hence, only a representative sample of the normal class can be used to train the model.

The problem we tackle is: Given a true anomaly with score s^*, how confident will the model be in predicting that it belongs to the anomaly class? Our intuitions might be misleading in this case. In fact, since the contamination factor is 0, the threshold will be set as the highest observed score in the training set and the definition of confidence slightly changes. In this case no failures are allowed (i.e., all training examples must have a score less than the chosen threshold), once one training example has a score greater than s^*, this implies that the chosen threshold will be greater than s^* as well. In practice, if s^* is always greater than or equal to the anomaly score for each training example, the model will always predict that the test example is anomalous but its confidence might not be so high. The reason is simple: since the sample D_n represents the normal class, the model cannot learn from the anomalies. So, drawing a normal example with a high anomaly score is theoretically possible. Hence, for a fixed anomalous test example, we need to analyze how the model's confidence in its prediction for the example changes as the number of normal examples in the training increases.

Theorem 3. *Given a dataset D_n with n scores and given a score s^* such that $s < s^*$ $\forall s \in D_n$, fixed $\gamma = 0$, the expected rate of anomalies in D_n, then*

$$\mathbb{P}(\hat{Y} = 1 \,|\, s^*, n, \gamma = 0, \hat{p}_{s^*}) \longrightarrow \frac{1}{e} \approx 0.368 \quad \textit{for } n \to +\infty.$$

Proof. Assuming that D_n contains no anomalies, we get $t = n$ successes. This yields an outlier probability of:

$$\hat{p}_{s^*} = \mathbb{E}[P_{s^*}] = \frac{1+t}{2+n} = \frac{1+n}{2+n}.$$

Then, using the estimated probability that one score is less than or equal to s^*, we can compute the confidence using the hypothesis that the contamination factor in the training set is 0, meaning that no failures are allowed:

$$\mathbb{P}(\hat{Y} = 1 \,|\, s^*, n, \gamma = 0, \hat{p}_{s^*}) = (\hat{p}_{s^*})^n\,.$$

In fact, if we drew a score greater than s^* from the training set, then the threshold would be greater than s^* (predicted class equal to 0). Let's now analyze the limit:

$$\lim_{n\to+\infty} \mathbb{P}(\hat{Y} = 1 \,|\, s^*, n, \gamma = 0, \hat{p}_{s^*}) = \lim_{n\to+\infty} (\hat{p}_{s^*})^n = \lim_{n\to+\infty} \left(\frac{1+n}{2+n}\right)^n$$
$$= \lim_{n\to+\infty} \left(\frac{2+n-1}{2+n}\right)^n = \lim_{n\to+\infty} \left[\left(1 + \frac{-1}{2+n}\right)^{2+n} \cdot \left(\frac{1+n}{2+n}\right)^{-2}\right] = \frac{1}{e},$$

where the first factor is a notable limit and converges to $\frac{1}{e}$, whereas the second term converges to 1 because of the rate of polynomials of degree 1. □

This can be understood as follows. While the outlier probability for a true anomaly goes to 1 when $n \to +\infty$, the number of normal examples in the training data also increases. Thus, we are more likely to observe unlikely events, i.e., the training set containing a normal example with a high anomaly score.

5 Experiments

The goal of our empirical evaluation is to: (1) intuitively illustrate how our confidence score works; (2) evaluate the quality of our confidence scores; and (3) assess the effect of using our Bayesian approach for converting anomalies scores to outlier probabilities on the quality of the confidence scores.

Table 1. The 21 benchmark anomaly detection datasets from [1].

Dataset	# Examples	# Vars	γ	Dataset	# Examples	# Vars	γ
ALOI	12384	27	0.030	PenDigits	9868	16	0.002
Annthyroid	7129	21	0.075	Pima	625	8	0.200
Arrhythmia	450	259	0.457	Shuttle	1013	9	0.013
Cardiotocography	1734	21	0.050	Spambase	3160	57	0.200
Glass	214	7	0.042	Stamps	340	9	0.091
HeartDisease	270	13	0.444	Waveform	3443	21	0.029
Hepatitis	80	19	0.163	WBC	454	9	0.022
Ionosphere	351	32	0.359	WDBC	367	30	0.027
Lymphography	148	19	0.040	Wilt	4655	5	0.020
PageBlocks	5473	10	0.102	WPBC	198	33	0.237
Parkinson	60	22	0.200				

5.1 Experimental Setup

Our experimental goal is to evaluate ExCeeD's ability to recover the example-wise confidences of an anomaly detector as opposed to evaluating predictive performance or quality of calibration. Hence, metrics like the AUC or Brier score are not suitable for assessing how small perturbations in the training data affect the example-wise predictions of the detector. For example, AUC would treat models that flipped the positions of two anomalies (normals) in the ranking produced by each model as being equivalently performant. In contrast, we are explicitly interested in understanding the number and magnitude of such flips. Moreover, our approach for converting the anomaly score to an outlier probability is a monotone function and hence does not affect the AUC of the model. We consider a confidence score to be good if it accurately captures the consistency with which a detector predicts the same label for an example. Hence, we expect a detector to predict for all examples subject to a confidence value of $\mathscr{C}(\hat{Y})_x$ the same label $\mathscr{C}(\hat{Y})_x$-percent of the time when retraining the detector multiple times with slightly perturbed training datasets. Therefore, we propose a novel method for evaluating an anomaly detector's example-wise confidence as an indication of how consistently it predicts the same label for that example. The method (1) draws 1000 sub-samples from the training data with the size of each sub-sample randomly selected in $[0.2 \cdot n, n]$, (2) trains an anomaly detector on each sub-sample, (3) uses each detector to predict the class labels of every example in the test set, and finally (4) computes for each test set example x the frequency F_x with which the detector predicted the same class.

We carryout our study on a benchmark consisting of 21 standard anomaly detection datasets from [1]. The datasets vary in size, number of features, and proportion of anomalies (Table 1). Given a benchmark dataset, we can now evaluate our confidence scores as follows. First, we split the dataset into training and test sets with stratified 5-fold cross-validation. Then, we take the class-weighted average of the L^2 differences between the confidence score $\mathscr{C}(\hat{Y})_x$ and the earlier computed frequency F_x of each test set example x, yielding:

$$error(\mathscr{C}, F) = \frac{1}{2|T_N|} \sum_{x \in T_N} \left(\mathscr{C}(\hat{Y})_x - F_x\right)^2 + \frac{1}{2|T_A|} \sum_{x \in T_A} \left(\mathscr{C}(\hat{Y})_x - F_x\right)^2$$

Table 2. Comparison of ExCeeD with the baselines. The table shows: the weighted average $error(\mathscr{C}, F)$ rank ± standard deviation (SD) of each method; the weighted average $error(\mathscr{C}, F)$ ± SD of each method (computed as in [3]); and the number of times ExCeeD wins (lower error), draws, and loses (higher error) against each baseline.

Method	Weighted avg. $error(\mathscr{C}, F)$ rank	# times ExCeeD			Weighted avg. $error(\mathscr{C}, F)$
	± SD of each method	Wins	Loses	Draws	± SD $\times 10^2$ of each method
ExCeeD	$\mathbf{1.429 \pm 0.844}$	-	-	-	$\mathbf{1.972 \pm 2.637}$
Baseline	2.270 ± 0.641	**52**	3	8	2.679 ± 2.931
ExCeeD-Unify	4.103 ± 2.040	**55**	2	6	15.13 ± 16.29
Isotonic	5.151 ± 1.482	**59**	2	2	17.92 ± 13.68
Beta	5.421 ± 1.285	**62**	0	1	23.13 ± 29.65
Logistic	5.532 ± 1.525	**62**	0	1	23.89 ± 29.69
ExCeeD-m	5.937 ± 2.709	**59**	1	3	24.95 ± 37.38
ExCeeD-Linear	6.802 ± 1.350	**61**	2	0	31.47 ± 23.15
ExCeeD-Squash	8.357 ± 1.271	**61**	2	0	45.97 ± 36.67

where T_N are the true test set normals and T_A the true test set anomalies. We take a class-weighted average because the large class-imbalances that characterize anomaly detection datasets, would otherwise skew the final error. We report averages over the folds. The underlying anomaly detector is either Knno, iForest, or Ocsvm. This results in $21 \times 3 = 63$ experiments for each method. We compare 9 approaches, which can be divided into three categories:

Our method. **ExCeeD** as introduced in Section 3.[3]

Naive baselines. **ExCeeD-m**, this is ExCeeD with a different prior distribution $\mathcal{B}eta(\gamma \cdot m, m \cdot (1 - \gamma))$ where γ is the contamination factor and m is such that $\gamma \cdot m = 10$ (suggested in [19]). A **Baseline** approach which assumes the confidence scores to be equal to the model's predictions.

Outlier probability methods. The unify [6], linear, and squash methods for estimating the outlier probabilities $\hat{p}_s$ from anomaly scores. These probabilities are *not* confidence scores. To obtain true confidence scores, we have to combine each of these methods with the second step of our framework, yielding **ExCeeD-Unify**, **ExCeeD-Linear**, and **ExCeeD-Squash**.

Calibration methods. The **Logistic** [13], **Isotonic** [20], and **Beta** [8] methods to empirically estimate calibration frequencies. Although these methods do not compute confidence scores, probabilistic predictions are often (incorrectly) interpreted as such and therefore included for completeness.

5.2 Experimental Results

To illustrate the intuition beyond our approach, Figure 2 compares how the confidence scores computed by the different methods evolve as we gradually move an example from the large normal central cluster to the small anomaly cluster in the bottom right (using Knno). For ExCeeD, the confidence scores start out high when the example is close to the cluster of normal points and the prediction is 0 (i.e., normal). The confidence gradually decreases as the example

[3] Implementation available at: https://github.com/Lorenzo-Perini/Confidence_AD.

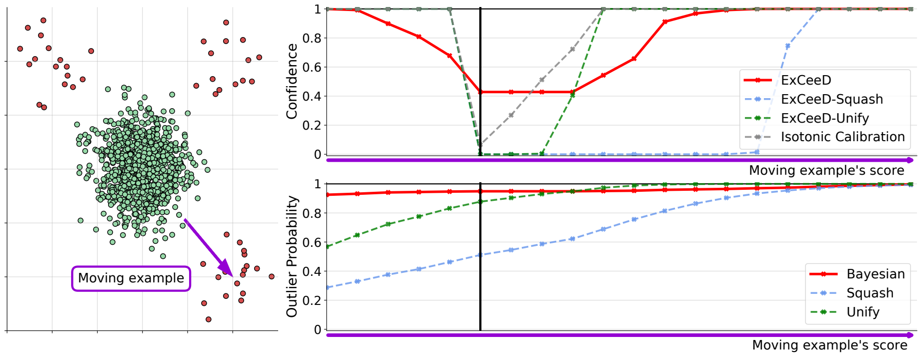

Fig. 2. Illustration of how moving an example along the purple arrow in the dataset (left plot) affects its outlier probability (bottom-right plot) and its confidence score derived from this probability (top-right plot). The underlying anomaly detector is Knno. Left of the vertical black line, the detector predicts that the example belongs to the normal class. Because at first the example is embedded in the cluster of normal examples (the green points in the dataset), the initial confidence score is high. However, the detector's confidence in its prediction decreases as the example moves away from the normal points. Finally, it increases again when the example nears the anomalies (the red points) and the example is predicted to be an anomaly. ExCeeD's confidence score captures our intuitions that the prediction should be confident (i.e., confidence score near 1.0) when the example is very obviously either normal or anomalous and uncertain (i.e., confidence score is near 0.5) when the example is equidistant from the normal and anomalous examples.

moves away from the normal cluster, eventually reaching about 50% when it is halfway between the normal and abnormal clusters. Once the example is far enough away from the normal cluster, the confidence increases again as the model changes its prediction and becomes more certain that the example is anomalous. Using ExCeed with its Bayesian outlier probability clearly captures the gradual change in confidence we would intuitively expect in this scenario.

Question 1: Does ExCeeD produce good confidence scores? Table 2 summarizes the comparison between ExCeed and the baselines in terms of $error(\mathscr{C}, F)$. Our method outperforms all baselines and has the lowest average error rank over the 63 experiments. It also achieves lower errors in at least 54 of the 63 experiments compared to every other method and achieves the lowest weighted average error. When the results are split out per underlying anomaly detector (Table 3), ExCeed still outperforms all baselines, winning against each baseline at least 20, 18 and 13 out of 21 times when the detectors are, respectively, Knno, iForest and Ocsvm.

Question 2: Does the Bayesian approach to estimate outlier probabilities contribute to better confidence scores? Our confidence score can be computed from any outlier probability measure. To evaluate specifically how our proposed Bayesian approach for estimating the outlier probabilities contributes

Table 3. Comparison of ExCeeD with the baselines, split out per anomaly detector (Knno, iForest, and Ocsvm). The table presents the number of times ExCeeD wins (lower error), draws, and loses (higher error) vs. each baseline.

Method	**Knno: # of**			**iForest: # of**			**Ocsvm: # of**		
	Wins	**Draws**	**Losses**	**Wins**	**Draws**	**Losses**	**Wins**	**Draws**	**Losses**
ExCeeD	**-**	-	-	-	-	-	-	-	-
Baseline	**21**	0	0	**18**	0	3	**13**	3	5
ExCeeD-Unify	**21**	0	0	**18**	0	3	**16**	2	3
Isotonic	**20**	0	1	**21**	0	0	**18**	2	1
Beta	**21**	0	0	**21**	0	0	**20**	0	1
Logistic	**21**	0	0	**21**	0	0	**20**	0	1
ExCeeD-m	**20**	0	1	**19**	0	2	**20**	1	0
ExCeeD-Linear	**21**	0	0	**21**	0	0	**19**	2	0
ExCeeD-Squash	**21**	0	0	**21**	0	0	**19**	2	0

to the confidence scores, we simply compare ExCeeD with the ExCeeD-Unify, ExCeeD-Linear, and ExCeeD-Squash baselines. The results are summarized in Tables 2 and 3. The Bayesian subroutine of ExCeeD to compute the outlier probabilities outperforms the three baselines by a substantial margin, obtaining lower errors in respectively 55, 61, and 61 out of 63 experiments, indicating its effectiveness.

6 Conclusions

We proposed a method to estimate the confidence of anomaly detectors in their example-wise class predictions. We first formally defined the confidence as the probability that example-wise predictions change due to perturbations in the training set. Then, we introduced a method that estimates the confidence using a two step approach. First, we estimate smooth outlier probabilities using a Bayesian approach. Second, we use the estimated outlier probabilities to derive a confidence score on an example-by-example basis. A large experimental comparison shows that our approach can recover confidence scores matching empirical frequencies.

Acknowledgements

This work is supported by the Flemish government under the "*Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen*" programme (JD, LP, VV), FWO (G0D8819N to JD), and KU Leuven Research Fund (C14/17/07 to JD).

References

1. Campos, G.O., Zimek, A., Sander, J., Campello, R.J., Micenková, B., Schubert, E., Assent, I., Houle, M.E.: On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. Data Mining and Knowledge Discovery **30**(4), 891–927 (2016)

2. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM computing surveys (CSUR) **41**(3), 1–58 (2009)
3. Demšar, J.: Statistical comparisons of classifiers over multiple datasets. Journal of Machine learning research **7**(Jan), 1–30 (2006)
4. Gao, J., Tan, P.N.: Converting output scores from outlier detection algorithms into probability estimates. In: Proceedings of Sixth IEEE International Conference on Data Mining. pp. 212–221. IEEE (2006)
5. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: Proceedings of the 34th International Conference on Machine Learning-. pp. 1321–1330 (2017)
6. Kriegel, H.P., Kroger, P., Schubert, E., Zimek, A.: Interpreting and unifying outlier scores. In: Proceedings of the 2011 SIAM International Conference on Data Mining. pp. 13–24. SIAM (2011)
7. Kull, M., Nieto, M.P., Kängsepp, M., Silva Filho, T., Song, H., Flach, P.: Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In: Advances in Neural Information Processing Systems (2019)
8. Kull, M., Silva Filho, T.M., Flach, P., et al.: Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. Electronic Journal of Statistics **11**(2), 5052–5080 (2017)
9. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: Proceeding of 2008 Eighth IEEE International Conference on Data Mining. pp. 413–422. IEEE (2008)
10. Naeini, M.P., Cooper, G., Hauskrecht, M.: Obtaining well calibrated probabilities using Bayesian binning. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
11. Perello-Nieto, M., Telmo De Menezes Filho, E.S., Kull, M., Flach, P.: Background check: A general technique to build more reliable and versatile classifiers. In: Proceedings of 16th IEEE International Conference on Data Mining. IEEE (2016)
12. Perini, L., Vercruyssen, V., Davis, J.: 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)
13. Platt, J., et al.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in large margin classifiers (1999)
14. Ramaswamy, S., Rastogi, R., Shim, K.: Efficient algorithms for mining outliers from large datasets. In: Proceedings of the 2000 ACM SIGMOD international conference on Management of data. pp. 427–438 (2000)
15. Robberechts, P., Bosteels, M., Davis, J., Meert, W.: Query log analysis: Detecting anomalies in dns traffic at a tld resolver. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 55–67. Springer (2018)
16. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural computation (2001)
17. Vaicenavicius, J., Widmann, D., Andersson, C., Lindsten, F., Roll, J., Schön, T.B.: Evaluating model calibration in classification. arXiv:1902.06977 (2019)
18. Vercruyssen, V., Wannes, M., Gust, V., Koen, M., Ruben, B., Jesse, D.: Semi-supervised anomaly detection with an application to water analytics. In: Proceedings of 18th IEEE International Conference on Data Mining. pp. 527–536. IEEE (2018)
19. Zadrozny, B., Elkan, C.: Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In: Proceedings of ICML. pp. 609–616 (2001)
20. Zadrozny, B., Elkan, C.: Transforming classifier scores into accurate multiclass probability estimates. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 694–699 (2002)