Anomaly Detection: How to Artificially Increase your F1-Score with a Biased Evaluation Protocol

Damien Fourure*[0000−0001−5085−0052],
Muhammad Usama Javaid*[0000−0001−9262−2250],
Nicolas Posocco*[0000−0002−1795−6039], and Simon Tihon*[0000−0002−3985−1967]

EURA NOVA, Mont-St-Guibert, Belgium
{firstname}.{lastname}@euranova.eu

Abstract. Anomaly detection is a widely explored domain in machine learning. Many models are proposed in the literature, and compared through different metrics measured on various datasets. The most popular metrics used to compare performances are F1-score, AUC and AVPR. In this paper, we show that F1-score and AVPR are highly sensitive to the contamination rate. One consequence is that it is possible to artificially increase their values by modifying the train-test split procedure. This leads to misleading comparisons between algorithms in the literature, especially when the evaluation protocol is not well detailed. Moreover, we show that the F1-score and the AVPR cannot be used to compare performances on different datasets as they do not reflect the intrinsic difficulty of modeling such data. Based on these observations, we claim that F1-score and AVPR should not be used as metrics for anomaly detection. We recommend a generic evaluation procedure for unsupervised anomaly detection, including the use of other metrics such as the AUC, which are more robust to arbitrary choices in the evaluation protocol.

Keywords: Anomaly detection · One-class classification · Contamination rate · Metrics

1 Introduction

Anomaly detection has been widely studied in the past few years, mostly for its immediate usability in real-world applications. Though there are multiple definitions of anomalies in the literature, most definitions agree on the fact that anomalies are data points which do not come from the main distribution. In the setting of unsupervised anomaly detection, the goal is to create a model which can distinguish anomalous samples from normal ones without being given such label at train time. In order to do so, most approaches follow a one-class classification framework, which models the normal data from the train set, and predicts as anomalous any point which does not fit this distribution of normal samples. Such prediction needs some prior knowledge provided through a contamination

* alphabetical order

rate on the test set, which is the ratio of anomalous data within. This ratio is used to build the model's decision rule.

In this setting, a lot of the literature uses the F1-score or the average precision (AVPR) to evaluate and compare models. In this paper we show that the evaluation protocol (train-test split and contamination rate estimation) has a direct influence on the contamination rate of the test set and the decision threshold, which in turn has a direct influence on these metrics. We highlight a comparability issue between results in different papers based on such evidence, and suggest an unbiased protocol to evaluate and compare unsupervised anomaly detection algorithms.

After an extensive study of the unsupervised anomaly detection field and of previous analyses of the evaluation methods (Section 2), we study the impact of the evaluation procedure on commonly used metrics (Section 3). Identified issues include a possibility to artificially increase the obtained scores and a non-comparability of the results over different datasets. Taking these into account, we suggest the use of a protocol leading to a better comparability in Section 4.

2 Related Work

Anomaly detection has been heavily dominated by unsupervised classification settings. One very popular approach in unsupervised anomaly detection is one-class classification, which refers to the setting where at train time, the model is given only normal samples to learn what the normal distribution is. The goal is to learn a scoring function to assign each data point an abnormality score. A threshold is then calculated from either a known or estimated contamination rate to turn scores into labels, samples with higher scores being considered as anomalies. In the literature different scoring functions have been used:

Proximity-based methods use heuristics based on distances between samples in some relevant space. These algorithms estimate the local density of data points through distances, and point out the most isolated ones. Legacy approaches include a simple distance to the Kth neighbour [2], Angle-Based Outlier Detection (ABOD) [11], which uses the variance over the angles between the different vectors to all pairs of points weighted by the distances between them, Local Outlier Factor (LOF) [3], which measures the local deviation of a given data point with respect to its neighbours, Connectivity-based Outlier Factor (COF) [23], which uses a ratio of averages of chaining distances with neighbours and Clustering-Based Local Outlier Factor (CBLOF) [10], which clusters the data and scores samples based on the size of the cluster they belong to and the distance to the closest big cluster. More recent approaches include DROCC [8], which makes the assumption that normal points lie on a well-sampled, locally linear low dimensional manifold and abnormal points lie at least at a certain distance from this manifold.

Reconstruction-based approaches use notions of reconstruction error to determine which data points are anomalous, the reconstruction of the densest parts of the distribution being easier to learn in general. In [17] for instance, the projection of each point on the main PCA axes is used to detect anomalies. As for [28], a GAN with a memory matrix is presented, each row containing a memorised latent vector with the objective to enclose all the normal data, in latent space, in between memorised vectors. The optimisation introduces a reconstruction error.

Representation-based approaches attempt to project the data in a space in which it is easy to identify outliers. Following this idea, One-Class SVM (OC-SVM) [22] uses a hypersphere to encompass all of the instances in the projection space. [12] proposed a neural network with robust subspace recovery layer. IDAGMM [13] presents an iterative algorithm based on an autoencoder and clustering, with the hypothesis that normal data points form a cluster with low variance. OneFlow [16], is a normalising-flow based method which aims at learning a minimum enclosing ball containing most of the data in the latent space, the optimisation ensuring that denser regions are projected close to the origin.

Adversarial scoring use the output of a discriminator as a proxy for abnormality, since it is precisely the goal of a discriminator to distinguish normal samples from other inputs. Driven by the motivation, an ensemble gan method is proposed in [9]. GANomaly [1] presents a conditional generative adversarial network with a encoder-decoder-encoder network to train better on normal images at training, and [27] presents Adversarially Learned Interface method with cycle consistency to ensure good reconstruction of normal data in one-class setting. [29] presents a gan network with autoencoder as generator for anomaly detection on images datasets.

Feature-level approaches try to detect anomalies at feature-level, and aggregate such information on each sample to produce an abnormality score at sample level. HBOS [7] assumes feature independence and calculates the degree of abnormality by building histograms. RVAE [5] uses a variational autencoder to introduce cell abnormality, which is converted into sample anomaly detection.

All of these categories are of course non-exclusive, and some approaches, as the very popular Isolation Forest [15], which uses the mean depth at which each sample is isolated in a forest of randomly built trees, do not fall in any of these. On the opposite, some recent methods combine multiple of such proxies for abnormality to reach better performances, each one using different hypotheses to model anomalies. For example [31] presents an end-to-end anomaly detection architecture. The model uses an autoencoder to perform dimensionality reduction to one or two dimensions and calculates several similarity errors, feeding then both latent representation and reconstruction errors to the gaussian mixture model. AnoGAN [21], which uses both a reconstruction error and a discriminator score to detect anomalies, also falls in this category.

Even if the original one-class setting requires data to be all normal at train time (which makes one-class approaches not strictly unsupervised) some approaches do not require clean data at train time, since they use what they learn about normal data to reduce as much as possible the impact of anomalies [13,16].

For all these settings, the main evaluation metrics used in the literature are the F1-score, the AUC (area under ROC curve) and the AVPR (average precision). The link between sensitivity, specificity and F1-score has been studied in [14], providing thresholding-related insights. In this work, we highlight the heterogeneity of current evaluation procedures in unsupervised anomaly detection performed in a one-class framework, would it be in terms of metrics or contamination-rate determination. For instance, many papers do not provide complete information about how the train-test splits are made [5,16]. For the same datasets, some papers re-inject the train anomalies in the test set [24,31,9] [1] and some others do not [31,30]. In some cases, it is not clear which contamination rate was used to compute the threshold [29,16,31,9,28], and some approaches prefer evaluating their model with multiple thresholds [12]. Different metrics are used to evaluate performences - F1-score [29,27,8,13,9,31], precision [27,9,31], recall [27,9,31], sensitivity [21], specificity [21], AUC [29,1,27,28,8,21,25,13,6,12,9,5], AVPR [12,13,5,25]. Finally many papers report directly the results from other papers and do not test the associated algorithms in their particular evaluation setting.

We show that all above-mentioned setup details have a direct impact on the F1-score and the AVPR. Since such heterogeneity leads to reproducibility and comparability issues, we suggest the use of an evaluation protocol with a robust metric which allows comparison.

3 Issues when Using F1-Score and AVPR Metrics

In this section, we analyse the sensitivity of the F1-score and AVPR metrics with respect to the contamination rate of the test set. First, we define the problem and different metrics and explain the impact of the estimation of the contamination rate. Then, we analyse the evolution of the metrics according to the true contamination rate of the test set. After having explained different evaluation protocols used in the literature, we show how they can be used to produce artificially good results using the F1-score and AVPR metrics. Finally, we show that these two metrics are also unsuitable for estimating the difficulty of datasets.

3.1 Formalism and Problem Statement

Consider a dataset $\mathbf{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\} \subset \mathbb{R}^d \times \{0, 1\}$, with $\mathbf{x}_i$ the d-dimensional samples and y_i the corresponding labels. We assume both classes are composed of i.i.d. samples. We also assume the normal class labeled 0 outnumbers the anomaly class labeled 1. Therefore, we choose the anomaly class as

[1] [31] do not publish their code but an unofficial implementation widely used (264 stars and 76 forks at the time of writing) is available at https://github.com/danieltan07/dagmm

	Actual Anomaly	Actual Normal
Predicted Anomaly	tp	fp
Predicted Normal	fn	tn

(a) Confusion Matrix

$$\textit{precision} = \frac{tp}{tp + fp} \tag{1}$$

$$\textit{recall} = \frac{tp}{tp + fn} \tag{2}$$

$$\textit{F1-score} = \frac{2}{\textit{precision}^{-1} + \textit{recall}^{-1}} \tag{3}$$

(b) Metrics based on binary predictions

Fig. 1: Metrics definitions.

positive class and use $^+$ to refer to it, while using $^-$ to refer to the normal class. This dataset is split into a train set $\mathbf{D}^{train} \subset \mathbf{D}$ and a test set $\mathbf{D}^{test} = \mathbf{D} \backslash \mathbf{D}^{train}$. Different procedures are used in the anomaly-detection community to perform this split, as detailed in Section 3.3. We denote N_t^+ (resp. N_t^-) the number of anomalous (resp. normal) samples in the test set.

We consider one-class classifiers, which are models learning an anomaly-score function f based only on clean samples $\mathbf{X}^{clean} = \{\mathbf{x}\ \forall(\mathbf{x}, y) \in \mathbf{D}^{train} \mid y = 0\}$. The anomaly-score function returns, for a given sample $\mathbf{x}$, an anomaly score $\hat{s} = f(\mathbf{x}) \in \mathbb{R}$ such that the higher the score, the more likely it is that $\mathbf{x}$ is an anomaly. We define $P^+(\hat{s})$ (resp. $P^-(\hat{s})$) the probability that an anomaly (resp. a clean sample) obtains an anomaly-score $\hat{s}$ with the trained model.

To get a binary prediction $\hat{y}$ for a sample $\mathbf{x}$ with anomaly score $\hat{s}$, we need to apply a threshold t to the anomaly score such that $\hat{y} = 1$ if $\hat{s} \geq t$ else $\hat{y} = 0$. Different ways to compute this threshold are used in the literature. A common approach is to choose it according to an estimation $\hat{\alpha}$ of the contamination rate α. The contamination rate is the proportion of anomalous samples in the dataset. It can be taken as domain knowledge, estimated on the train set or, for evaluation purposes only, on the test set directly.

3.2 Definition of the Metrics

Using the final prediction and the ground truth labels, we can count the *true positives* tp, *true negatives* tn, *false positives* fp and *false negatives* fn, as shown in Figure 1a. The *precision*, *recall* and *F1-score* are computed using these quantities as shown in the equations of Figure 1b. An example of these metrics applied with a varying contamination rate estimation $\hat{\alpha}$, inducing a varying threshold, is shown in Figure 2. It is interesting to note that, if the estimated contamination rate $\hat{\alpha}$ is equal to the true contamination rate α, we have *precision* = *recall* = *F1-score.* This can be easily explained: if the estimated contamination rate is the true contamination rate, the threshold is computed such that the number of samples predicted as anomalous is equal to the number of true anomalies in the set. Thus, if a normal sample is wrongly predicted as anomalous (i.e. is a false positive), it necessarily means that an anomalous sample has been predicted as normal (i.e. is a false negative). That is, $fp = fn$. Given the formulas of precision and recall

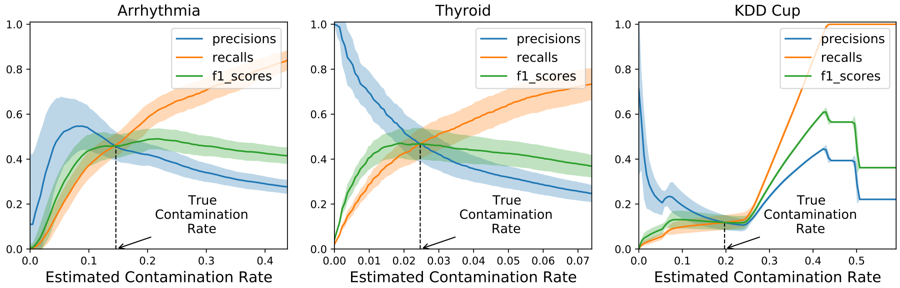


Fig. 2: Evolution of the *Precisions*, *Recalls* and *F1-scores* according to the estimated contamination rate on three different datasets. The curves are obtained using the Algorithm 1 introduced in Section 3.3.

(see equations of Figure 1b) we have $precision = recall$. As the F1-score is the harmonic mean of precision and recall, we have *precision* = *recall* = *F1-score*. Inversely, if this equality can be observed in reported results, it is safe to assume the estimation of the contamination rate is equal to the true contamination rate.

We also include the AUC and AVPR in our analysis. These metrics are obtained by analysing the results with different thresholds. The AUC is defined through the receiver-operator characteristic (ROC) curve, a curve of the true positive rate over the false positive rate for various thresholds. Therefore, we redefine tp, fp, fn and tn as functions depending on the threshold. The area under the ROC curve AUC, sometimes written $AUROC$, is the total area under this curve, that is:

$$AUC = \int_{t=-\infty}^{\infty} \frac{tp(t)}{tp(t) + fn(t)} \frac{d}{dt} \left(\frac{fp}{fp + tn} \right) \Big|_t dt. \tag{4}$$

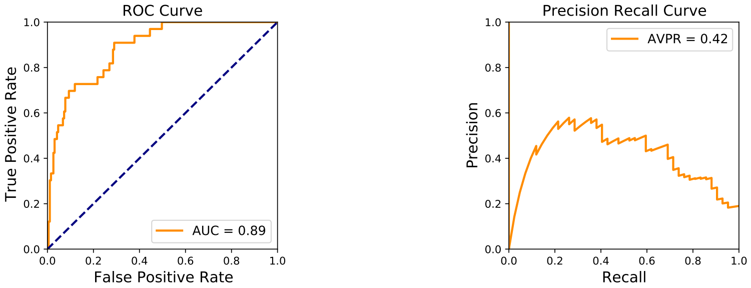


Fig. 3: Example of ROC Curve and Precision Recall Curve obtained on the Arrhythmia dataset. The scores are obtained using the Algorithm 1 introduced in Section 3.3.

Similarly, the AVPR is defined through the precision-recall (PR) curve, a curve of precision over recall for different thresholds. The area under this curve is referred to as the average precision (AVPR) metric, as it can be seen as a weighted average of the precision for different recall. We have

$$AVPR = \int_{t=-\infty}^{\infty} precision(t) \frac{d}{dt}(recall)\Big|_t dt. \tag{5}$$

An example of a ROC curve and a precision recall curve is given in Figure 3.

We show in this paper that the F1-score and AVPR metrics are highly sensitive to the true contamination rate of the test set. We show this sensitivity has a negative impact on the comparison of different classifiers or datasets, especially when using different protocols.

3.3 Evaluation Protocols: Theory vs Practice

Machine learning theory tells us that the evaluation of an algorithm should be done on a test set completely separated from the train set. Algorithm 1 presents the unbiased procedure to train and evaluate an anomaly detection model. A dataset (containing both normal and anomalous samples) is split into a train set and a test set. The anomalous samples from the train set are removed to get a clean set that is used to train a model. The train set is also used to compute the contamination rate and fix the threshold, for example using a threshold such that the train set has as many anomalies as predicted anomalies, i.e. $fp = fn$. This threshold is finally used on the predictions made on the new (unseen) samples composing the test set to measure the F1-score. The AUC and AVPR are computed using the predicted scores directly. Even though this procedure is theoretically the correct way to evaluate a model, it has a significant drawback in practice. The anomalous samples in the train set are used only to compute the threshold for the F1-score and are then thrown away. Because there are, by definition, few anomalies in a dataset, one could be tempted to use these samples in the test set. Indeed, as visible in Figure 4, the more anomalous samples we can use to evaluate a model, the more precise the evaluation.

To make full use of the anomalous samples, the procedure described in Algorithm 2 recycles the anomalous samples contained in the train set. The threshold is then computed on the test set as there are no anomalies left in the train set to estimate it. This leads to a situation where *precision* = *recall* = *F1-score* as described in Section 3.1. This recycling procedure makes sense in the context of anomaly detection as it obtains more precise results, and can be found in the literature [24,31].

Algorithms 1 and 2 take as input any dataset and any trainable anomaly-score function. For the dataset, if not specified otherwise, we use the Arrhythmia and Thyroid datasets from the ODDS repository [20] and the Kddcup dataset from the UCI repository [4]. These datasets are often used in the anomaly-detection literature, and are therefore all indicated for our analysis. They have respectively 452, 3772 and 494020 samples, with a contamination rate of respectively 14.6%,

Algorithm 1: Theoretically unbiased evaluation protocol

```
Input:
    $\mathbf{D} \subset \mathbb{R}^d \times \{0,1\}$ a set of $N$ $d$-dimensional input samples and their
  corresponding labels (1 = anomaly, 0 = normal)
    $\beta$ the amount of data used for the test set
    $f$ a trainable anomaly-score function
Output:
    F1-score, AUC and AVPR
Procedure:
    $\mathbf{D}^{train}, \mathbf{D}^{test} = \text{split\_train\_test}(\mathbf{D}, \beta)$
    $\mathbf{X}^{clean} = \{\mathbf{x}\ \forall(\mathbf{x}, y) \in \mathbf{D}^{train} \mid y = 0\}$
    Normalise the data based on $\mathbf{X}^{clean}$ if necessary
    Train $f$ using $\mathbf{X}^{clean}$
    $\hat{\mathbf{s}}^{train} = \{(f(\mathbf{x}), y)\ \forall(\mathbf{x}, y) \in \mathbf{D}^{train}\}$
    Compute estimated contamination rate $\hat{\alpha} = \frac{|\{(\mathbf{x},y)\ \forall(\mathbf{x},y)\in\mathbf{D}^{train} | y=1\}|}{|\mathbf{D}^{train}|}$
    Compute threshold $t$ such that $\frac{|\{(\hat{s},y)\ \forall(\hat{s},y)\in\hat{\mathbf{s}}^{train} | \hat{s}\geq t\}|}{|\hat{\mathbf{s}}^{train}|} = \hat{\alpha}$
    $\hat{\mathbf{s}}^{test} = \{(f(\mathbf{x}), y)\ \forall(\mathbf{x}, y) \in \mathbf{D}^{test}\}$
    $\hat{\mathbf{y}}^{test} = \{(\hat{y}, y)\ \forall(\hat{s}, y) \in \hat{\mathbf{s}}^{test}\ \forall\hat{y} \in \{0,1\} \mid \hat{y} = 1 \text{ if } \hat{s} \geq t \text{ else } \hat{y} = 0\}$
    Compute F1-score using $\hat{\mathbf{y}}^{test}$
    Compute AUC and AVPR using $\hat{\mathbf{s}}^{test}$
```

Algorithm 2: *Recycling* evaluation protocol for anomaly detection

```
Input:
    $\mathbf{D} \subset \mathbb{R}^d \times \{0,1\}$ a set of $N$ $d$-dimensional input samples and their
  corresponding labels (1 = anomaly, 0 = normal)
    $\beta$ the amount of data used for the test set
    $f$ a trainable anomaly-score function
Output:
    F1-score, AUC and AVPR
Procedure:
    $\mathbf{D}^{train}, \mathbf{D}^{test} = \text{split\_train\_test}(\mathbf{D}, \beta)$
    $\mathbf{X}^{clean} = \{\mathbf{x}\ \forall(\mathbf{x}, y) \in \mathbf{D}^{train} \mid y = 0\}$
    Add $\{(\mathbf{x}, y)\ \forall(\mathbf{x}, y) \in \mathbf{D}^{train} \mid y = 1\}$ to $\mathbf{D}^{test}$
    Normalise the data based on $\mathbf{X}^{clean}$ if necessary
    Train $f$ using $\mathbf{X}^{clean}$
    $\hat{\mathbf{s}}^{test} = \{(f(\mathbf{x}), y)\ \forall(\mathbf{x}, y) \in \mathbf{D}^{test}\}$
    Compute contamination rate $\alpha = \frac{|\{(\mathbf{x},y)\ \forall(\mathbf{x},y)\in\mathbf{D}^{test} | y=1\}|}{|\mathbf{D}^{test}|}$
    Compute threshold $t$ such that $\frac{|\{(\hat{s},y)\ \forall(\hat{s},y)\in\hat{\mathbf{s}}^{test} | \hat{s}\geq t\}|}{|\hat{\mathbf{s}}^{test}|} = \alpha$
    $\hat{\mathbf{y}}^{test} = \{(\hat{y}, y)\ \forall(\hat{s}, y) \in \hat{\mathbf{s}}^{test}\ \forall\hat{y} \in \{0,1\} \mid \hat{y} = 1 \text{ if } \hat{s} \geq t \text{ else } \hat{y} = 0\}$
    Compute F1-score using $\hat{\mathbf{y}}^{test}$
    Compute AUC and AVPR using $\hat{\mathbf{s}}^{test}$
```

2.5% and 19.7%. For Kddcup, as done in the literature, the samples labeled as *"normal"* are considered as anomalous and, for computational reasons, only

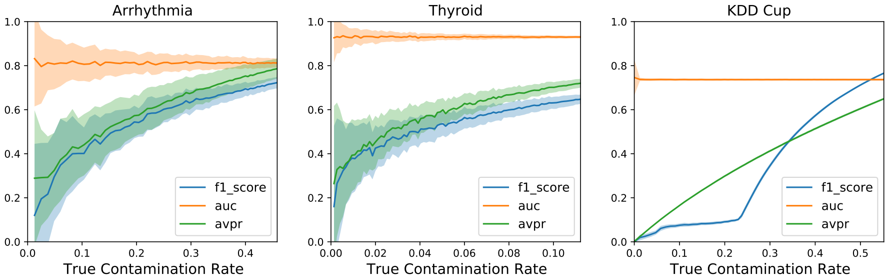


Fig. 4: F1-Score, AUC and AVPR versus the number of anomalies in the test set for three different datasets.

10% are used. For the trainable anomaly-score function, if not specified otherwise, we use OC-SVM [22] with its default hyper-parameters, as implemented in `sklearn` [18]. We choose this model as it has proven its worth and is often used as a baseline in the literature. We run all our experiments 100 times to report meaningful means and standard deviations. The code to reproduce all our figures and results is available at https://github.com/euranova/F1-Score-is-Biased.

3.4 Metrics Sensitivity to the Contamination Rate of the Test Set

We analyse the effect of the contamination rate of the test set on the F1-score and AVPR metrics. To do so, we use a variant of Algorithm 2 with a 20-80 train-test split on the clean samples only. We then re-inject a varying number of anomalous samples in the test set, from none to all of them. Figure 4 shows that F1-score and AVPR improve as more anomalies are added to the test set. Because the train set is fixed, this clearly shows that the F1-score and AVPR metrics are biased by the amount of anomalous samples in the test set. This sensitivity can be analysed theoretically.

First, note that the contamination rate $\alpha = \frac{N_t^+}{N_t^+ + N_t^-} = \frac{N_t^+}{N_t^-} / \left(\frac{N_t^+}{N_t^-} + 1\right)$ is increasing with $\frac{N_t^+}{N_t^-}$. We start the analysis in a constant-threshold setting where the threshold t does not depend on the test set, e.g. as in Algorithm 1. In this setting, we can compute $p^- = \int_{\hat{s}=-\infty}^{t} P^-(\hat{s}) d\hat{s}$ the probability that the model classifies correctly a normal sample and $p^+ = \int_{\hat{s}=t}^{\infty} P^+(\hat{s}) d\hat{s}$ the probability that the model classifies correctly an anomalous sample (the recall). We observe that $tn = N_t^- * p^-$ and $fp = N_t^- * (1 - p^-)$ are directly proportional to N_t^-, while $tp = N_t^+ * p^+$ and $fn = N_t^+ * (1 - p^+)$ are directly proportional to N_t^+. As such, the recall p^+ does not depend on α while the precision $(= \frac{N_t^+ * p^+}{N_t^+ * p^+ + N_t^- * p^-} = \frac{\frac{N_t^+}{N_t^-} p^+}{\frac{N_t^+}{N_t^-} p^+ + p^-})$ increases with $\frac{N_t^+}{N_t^-}$ and therefore with α. This proves the AVPR increases with α as the only value changing in Equation 5 is the increasing precision. This also proves the F1-score with a fixed threshold is

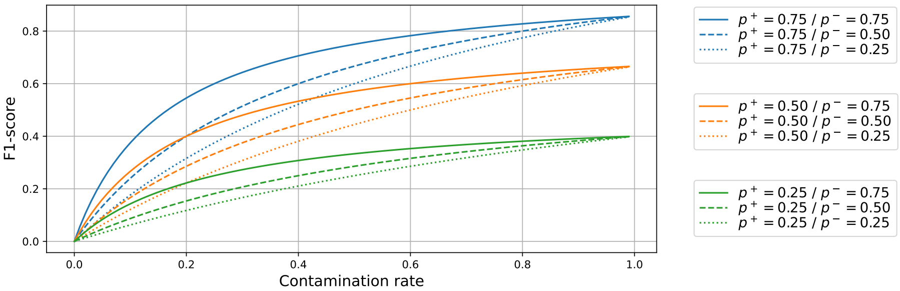


Fig. 5: Theoretical F1-score for varying contamination rates of the test set, anomaly-detection capabilities p^+ and normal-detection capabilities p^-.

increasing with α, as it is the harmonic mean of a constant and an increasing value. This theoretical variation of the F1-score is shown in Figure 5.

We now analyse the case where the threshold t for the F1-score is computed using the test set as done in Algorithm 2. As we use a perfect estimation of the contamination rate, we have *recall* = *precision* = *F1-score*. Let us analyse this quantity in the view of the recall and compare it to the constant-threshold setting. If we add an anomaly to the test set, there are two possibilities:

- It is at the right side of the threshold, hence the threshold stays constant as there are still as many samples detected as anomalies as there are anomalies.
- It is at the wrong side of the threshold. The threshold therefore decreases to include one more sample as a predicted anomaly. There are two possibilities:
  - This additional sample is an anomaly, in which case the recall increases, whereas it would have decreased in the constant-threshold setting.
  - This additional sample is a clean sample, in which case the recall decreases the same way it would have decreased in the constant-threshold setting.

Compared to the constant-threshold setting, the only difference is the case where the recall is better than expected thanks to the shift of the threshold. Therefore, adding anomalies increases the F1-score even more than in the constant-threshold setting, meaning the variable-threshold setting is even more biased by the contamination rate of the test set. More formally, if we add anomalies without changing the number of clean samples, the new threshold t' will be smaller (or equal in the case of a perfect classifier) than the old one t, as we want to select more samples as being anomalies. The recall, precision and F1-score therefore increase from $\int_{\hat{s}=t}^{\infty} P^{+}(\hat{s})d\hat{s}$ (i.e. p^+ in the previous demonstration) to $\int_{\hat{s}=t'}^{t} P^{+}(\hat{s})d\hat{s}+\int_{\hat{s}=t}^{\infty} P^{+}(\hat{s})d\hat{s}$, which is greater or equal as a probability is always positive. Thus, if the classifier is not a perfect classifier, the F1-score increases with the contamination rate of the test set.

This concludes our demonstration that both the AVPR and the F1-score metrics are biased by the contamination rate of the test set.

Table 1: Demonstration of the sensitivity of the metrics to the evaluation protocol. Optimal threshold is the threshold computed on the test set to obtain the best F1-score possible (unapplicable to AUC and AVPR).

	Split procedure	Algo 1	Algo 2	Algo 2	Algo 2
	Test size	20%	20%	5%	5%
	Threshold	estimated	estimated	estimated	optimal
F1	arrhythmia	0.451(± 0.103)	0.715(± 0.025)	0.867(± 0.021)	0.888(± 0.012)
	kddcup	0.102(± 0.025)	0.762(± 0.004)	0.940(± 0.002)	0.971(± 0.001)
	thyroid	0.446(± 0.110)	0.647(± 0.022)	0.781(± 0.021)	0.803(± 0.017)
AVPR	arrhythmia	0.481(± 0.116)	0.770(± 0.041)	0.924(± 0.028)	0.924(± 0.029)
	kddcup	0.299(± 0.017)	0.653(± 0.015)	0.872(± 0.008)	0.873(± 0.007)
	thyroid	0.488(± 0.113)	0.719(± 0.020)	0.880(± 0.017)	0.881(± 0.017)
AUC	arrhythmia	0.809(± 0.065)	0.806(± 0.020)	0.803(± 0.042)	0.799(± 0.042)
	kddcup	0.736(± 0.007)	0.735(± 0.007)	0.735(± 0.011)	0.737(± 0.011)
	thyroid	0.935(± 0.027)	0.931(± 0.005)	0.929(± 0.009)	0.929(± 0.009)

3.5 How to Artificially Increase your F1-Score and AVPR

Combining the previous results and algorithms, we can define an algorithm to get an arbitrarily good F1-score or AVPR on any dataset. As shown in Section 3.4, the F1-score and AVPR are sensitive to the contamination rate of the test set. Using the Algorithm 2 from Section 3.3, we can make this contamination rate vary. To do so, we only have to modify β, the amount of data used for the test set. Indeed, it modifies the number of normal samples N_t^- in the test set while the number of anomalies N_t^+ stays the same. Pushed to the extreme, we can have near to no clean samples in the test set, resulting in a near-to-perfect F1-score and AVPR. This phenomenon is shown in Table 1. We can see that, by using the Algorithm 2 the F1-score increases for all three datasets. This is because the anomalous sample of the train set are re-injected and thus the contamination rate of the test set increases. Then, using 5% of the data for the test set instead of 20% increase again the F1-score and AVPR.

Another interesting observation is that fixing the threshold according to the contamination rate does not give the optimal F1-score [14]. In practice, using a threshold smaller than this one often results in a better F1-score, as visible in Figure 2 and shown in Figure 6. As a consequence, we can artificially increase the F1-scores even more by computing the optimal threshold. This is shown in the last two columns of Table 1.

This proves that, with the exact same model and seemingly identical metrics, the F1-score can be greater and greater. This clearly supports the importance of specifying in detail the train-test split used and the way the threshold is computed. We observe in the literature that this part of the evaluation protocols is often missing or unclear [29,31,9,5], and the reported results are therefore impossible to compare with. This is part of the reproducibility problem observed in the machine learning community. More importantly, some papers report re-

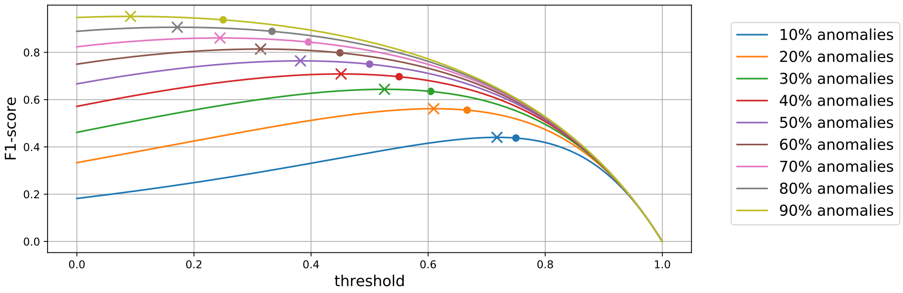


Fig. 6: Theoretical example of the evolution of the F1-score for different thresholds and contamination rates of the test set. The model used is a toy model having $P^{+}(\hat{s}) = 2 * \hat{s}$ and $P^{-}(\hat{s}) = 2 * (1 - \hat{s})$ for $0 \leq \hat{s} \leq 1$. Dots are the $fp = fn$ thresholds and crosses are the optimal thresholds.

sults computed using different evaluation protocols [24,9], leading to meaningless comparisons that are nonetheless used to draw arbitrary conclusions.

3.6 F1-Score Cannot Compare Datasets Difficulty

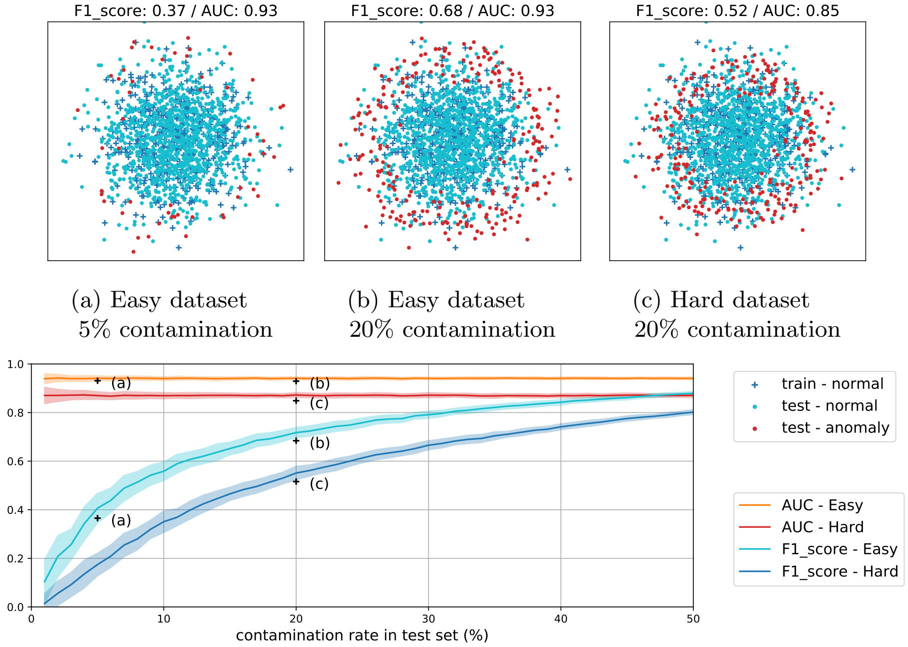
Another shortcoming of the F1-score and AVPR metrics is the comparison between datasets. One may be tempted to conclude that a dataset on which an approach has a higher F1-score is easier to model than another dataset with a lower score. However, this intuition is flawed when using these metrics as they strongly depend on the contamination rate of these datasets.

Figure 7 highlights the dataset comparison problem. Figure 7d shows the F1-score and AUC obtained on two toy datasets, an easy one (with a big radius) and a hard one (with a small radius). We show that we can obtain a better F1-score on a hard dataset (Figure 7c) than on an easy dataset (Figure 7a) just by changing the contamination rate. With an equal contamination rate (Figure 7b) we can see that the easy dataset is indeed easier to model.

This situation also appears in real-world datasets. Indeed, in Table 1 with Algorithm 1, the *kdd cup* dataset appears harder than *arrhythmia* and *thyroid* as it obtains a worse F1-score. However, if we compare them with Algorithm 2, the *kdd cup* dataset obtains better results than the other two. The comparison of the datasets difficulty is inconsistent and therefore unreliable.

4 Call for Action

Given the instability shown in Section 3, we suggest the anomaly-detection community to use the evaluation protocol described in Algorithm 2 but using only the AUC metric. Other approaches could be adopted, but this one will give better comparability between reported results and these results will have lower variances.



(a) Easy dataset
5% contamination

(b) Easy dataset
20% contamination

(c) Hard dataset
20% contamination

(d) Performances comparison between easy toy dataset and hard toy dataset

Fig. 7: Analysis of the dataset comparison through different metrics. We randomly draw normal samples from a standard gaussian distribution, and anomalous samples from a noisy a around the mean. By varying the radius of the circle - 2.5 for the *easy* case, 2.1 for the *hard* one - we change the difficulty of the dataset. The greater the radius, the easier it is to separate both distributions. A simple gaussian is used as model.

4.1 Use AUC

We have demonstrated in Section 3 how the F1-score and AVPR metrics can be tricky to use and lead to wrong conclusions, slowing down the research in the field. To avoid these pitfalls, we recommend using the AUC metric. First of all, AUC is not sensitive to the contamination rate of the test set, as shown in Figure 4. This can be proven by developing Equation 4:

$$AUC = \int_{t=-\infty}^{\infty} \frac{tp(t)}{tp(t)+fn(t)} \frac{d}{dt}\left(\frac{fp}{fp+tn}\right)\Big|_t dt \tag{6}$$

$$= \int_{t=-\infty}^{\infty} \int_{\hat{s}=t}^{\infty} P^+(\hat{s}) d\hat{s} \frac{d}{dt}\left(\int_{\hat{s}'=-\infty}^{t} P^-(\hat{s}') d\hat{s}'\right)\Big|_t dt \tag{7}$$

$$= \int_{\{(\hat{s},t)\in\mathbb{R}^2 | \hat{s}\geq t\}} P^+(\hat{s}) P^-(t)\ d\hat{s}\ dt \tag{8}$$

which depends only on the model properties (P^+ and P^-) and not on the test set. This independence prevents most of the problems identified in the previous section. As illustrated in Figure 7, datasets are more comparable using AUC. Moreover, Table 1 highlights the stability of the AUC.

Additionally, there is no need to define a threshold when using AUC. This is a good thing as the choice of a threshold can prevent comparability. Indeed, most of the proposed models in the literature [29,19,16,31,9] do not include a way to train a threshold. Therefore, arbitrary thresholds are used to compute the F1-score. The way to arbitrarily choose this threshold can vary from one paper to the other and lead to incomparable results. Even worse, this threshold could depend on the test set, such as the one producing $fp = fn$, thus having results biased by the contamination rate of the test set. This is not a problem with the AUC as it does not need a threshold.

Finally, another source of non-comparability is the choice of the positive class. Some may choose the *normal* class as positive [8,19] and other the *anomaly* class as positive [12,26,9]. AUC has the advantage of being independent of the choice of which class is seen as positive, as long as the scores are negated accordingly. Indeed, Equation 8 is symmetric between P^+ and P^- up to the $\hat{s} \geq t$ part which is solved by negating the scores.

All in all, AUC is insensitive to many arbitrary choices in the evaluation protocol. It results in a better comparability between the different reported results.

4.2 Do not Waste Anomalous Samples

As, by definition, anomalous samples are rare, it is important to re-inject them in the test set, as described in Algorithm 2. Indeed, by using more anomalous samples in the test set, the variance in the metrics is lower.

As shown in Table 1, when using AUC, Algorithm 2 gives the same mean result than Algorithm 1, but with a better precision (lower standard deviation). This is easily explained by the fact that there are more anomalies in the test set, increasing the applicability of the law of large numbers. This increased precision can be useful to obtain significant results rather than random-looking ones. Algorithm 2 can be used as long as the metric used is not biased by the contamination rate of the test set. It is therefore compatible with the AUC metric.

5 Conclusion

The literature in the field of anomaly detection lacks precision in describing evaluation protocols. Because of the sensitivity of the F1-score and AVPR metrics to the contamination rate of the test set, this results in a reproducibility issue of the proposed works as well as a comparison problem between said works. Moreover, we observe that some works do the subtle mistake of comparing results produced with different evaluation protocols and draw arbitrary conclusions from it. To solve this problem, we suggest the anomaly-detection community to use the AUC, which is insensitive to most arbitrary choices in the evaluation protocol. Moreover, we propose to use a *recycling* algorithm (Algorithm 2) for the train-test split to make the most of anomalies in each dataset. These two actions will result in more comparable and more precise results across research teams.

References

1. Akcay, S., Atapour-Abarghouei, A., Breckon, T.P.: Ganomaly: Semi-supervised anomaly detection via adversarial training. In: Asian conference on computer vision. pp. 622–637. Springer (2018)
2. Angiulli, F., Pizzuti, C.: Fast outlier detection in high dimensional spaces. In: European conference on principles of data mining and knowledge discovery. pp. 15–27. Springer (2002)
3. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD international conference on Management of data. pp. 93–104 (2000)
4. Dua, D., Graff, C.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
5. Eduardo, S., Nazábal, A., Williams, C.K., Sutton, C.: Robust variational autoencoders for outlier detection and repair of mixed-type data. In: International Conference on Artificial Intelligence and Statistics. pp. 4056–4066. PMLR (2020)
6. Ergen, T., Kozat, S.S.: Unsupervised anomaly detection with lstm neural networks. IEEE transactions on neural networks and learning systems **31**(8), 3127–3141 (2019)
7. Goldstein, M., Dengel, A.: Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. KI-2012: Poster and Demo Track pp. 59–63 (2012)
8. Goyal, S., Raghunathan, A., Jain, M., Simhadri, H.V., Jain, P.: Drocc: Deep robust one-class classification. In: International Conference on Machine Learning. pp. 3711–3721. PMLR (2020)
9. Han, X., Chen, X., Liu, L.P.: Gan ensemble for anomaly detection. arXiv preprint arXiv:2012.07988 (2020)
10. He, Z., Xu, X., Deng, S.: Discovering cluster-based local outliers. Pattern Recognition Letters **24**(9-10), 1641–1650 (2003)
11. Kriegel, H.P., Schubert, M., Zimek, A.: Angle-based outlier detection in high-dimensional data. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 444–452 (08 2008). https://doi.org/10.1145/1401890.1401946
12. Lai, C.H., Zou, D., Lerman, G.: Robust subspace recovery layer for unsupervised anomaly detection. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=rylb3eBtwr
13. Li, T., Wang, Z., Liu, S., Lin, W.Y.: Deep unsupervised anomaly detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 3636–3645 (January 2021)
14. Lipton, Z.C., Elkan, C., Naryanaswamy, B.: Optimal thresholding of classifiers to maximize f1 measure. In: Calders, T., Esposito, F., Hüllermeier, E., Meo, R. (eds.) Machine Learning and Knowledge Discovery in Databases. ECML PKDD. pp. 225–239. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)

15. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining. p. 413422. ICDM '08, IEEE Computer Society, USA (2008). https://doi.org/10.1109/ICDM.2008.17, https://doi.org/10.1109/ICDM.2008.17
16. ukasz Maziarka, mieja, M., Sendera, M., ukasz Struski, Tabor, J., Spurek, P.: Flow-based anomaly detection (2020)
17. Parra, L., Deco, G., Miesbach, S.: Statistical independence and novelty detection with information preserving nonlinear maps. Neural Computation **8** (02 1997). https://doi.org/10.1162/neco.1996.8.2.260
18. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
19. Perera, P., Nallapati, R., Xiang, B.: Ocgan: One-class novelty detection using gans with constrained latent representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
20. Rayana, S.: ODDS library (2016), http://odds.cs.stonybrook.edu
21. Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: International conference on information processing in medical imaging. pp. 146–157. Springer (2017)
22. Schölkopf, B., Williamson, R.C., Smola, A.J., Shawe-Taylor, J., Platt, J.C., et al.: Support vector method for novelty detection. In: NIPS. vol. 12, pp. 582–588. Citeseer (1999)
23. Tang, J., Chen, Z., Fu, A.W.c., Cheung, D.W.: Enhancing effectiveness of outlier detections for low density patterns. In: Chen, M.S., Yu, P.S., Liu, B. (eds.) Advances in Knowledge Discovery and Data Mining. pp. 535–548. Springer Berlin Heidelberg, Berlin, Heidelberg (2002)
24. Wang, J., Sun, S., Yu, Y.: Multivariate triangular quantile maps for novelty detection. In: Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019)
25. Wang, S., Zeng, Y., Liu, X., Zhu, E., Yin, J., Xu, C., Kloft, M.: Effective end-to-end unsupervised outlier detection via inlier priority of discriminative network. In: NeurIPS. pp. 5960–5973 (2019)
26. Xu, X., Liu, H., Yao, M.: Recent progress of anomaly detection. Complexity **2019**, 1–11 (01 2019). https://doi.org/10.1155/2019/2686378
27. Yang, Z., Bozchalooi, I.S., Darve, E.: Regularized cycle consistent generative adversarial network for anomaly detection (2020)
28. Yang, Z., Zhang, T., Bozchalooi, I.S., Darve, E.: Memory augmented generative adversarial networks for anomaly detection (2020)
29. Zaigham Zaheer, M., Lee, J.H., Astrid, M., Lee, S.I.: Old is gold: Redefining the adversarially learned one-class classifier training paradigm. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14171–14181 (2020). https://doi.org/10.1109/CVPR42600.2020.01419
30. Zhai, S., Cheng, Y., Lu, W., Zhang, Z.: Deep structured energy based models for anomaly detection. In: International Conference on Machine Learning. pp. 1100–1109. PMLR (2016)
31. Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D., Chen, H.: Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In: International Conference on Learning Representations (2018)