

# Transferring the Contamination Factor between Anomaly Detection Domains by Shape Similarity

Lorenzo Perini, Vincent Vercruyssen, Jesse Davis

KU Leuven, Dept. of Computer Science, DTAI & Leuven.AI, B-3000 Leuven, Belgium  
lorenzo.perini@kuleuven.be, vincent.vercruyssen@kuleuven.be, jesse.davis@kuleuven.be

## Abstract

Anomaly detection attempts to find examples in a dataset that do not conform to the expected behavior. Algorithms for this task assign an anomaly score to each example representing its degree of anomalousness. Setting a threshold on the anomaly scores enables converting these scores into a discrete prediction for each example. Setting an appropriate threshold is challenging in practice since anomaly detection is often treated as an unsupervised problem. A common approach is to set the threshold based on the dataset's contamination factor, i.e., the proportion of anomalous examples in the data. While the contamination factor may be known based on domain knowledge, it is often necessary to estimate it by labeling data. However, many anomaly detection problems involve monitoring multiple related, yet slightly different entities (e.g., a fleet of machines). Then, estimating the contamination factor for each dataset separately by labeling data would be extremely time-consuming. Therefore, this paper introduces a method for *transferring the known contamination factor* from one dataset (the source domain) to a related dataset where it is unknown (the target domain). Our approach does not require labeled target data and is based on modeling the shape of the distribution of the anomaly scores in both domains. We theoretically analyze how our method behaves when the (biased) target domain anomaly score distribution converges to its true one. Empirically, our method outperforms several baselines on real-world datasets.

## Introduction

Anomaly detection (Chandola, Banerjee, and Kumar 2009) is a data mining task that aims to automatically identify examples in a dataset that do not correspond to typical or expected behavior. This is a significant and important problem because anomalies often represent unwanted behaviors such as excess water usage (Vercruyssen et al. 2018), abnormal web traffic (Robberechts et al. 2018), or malfunctions in unmanned aerial vehicles (Khan et al. 2019) that have an associated cost. Anomaly detection is particularly challenging as one often lacks labeled examples, especially for the anomalies, because collecting them may be infeasible (e.g., intentionally break equipment to observe anomalies) or because anomalies are scarce (e.g., one often has to inspect 100s of examples before

encountering an anomaly) among other reasons. Hence, classic approaches (Breunig et al. 2000; Goldstein and Dengel 2012; He, Xu, and Deng 2003; Kriegel et al. 2009; Li et al. 2020; Liu, Ting, and Zhou 2008; Pevný 2016; Ramaswamy, Rastogi, and Shim 2000; Zhao et al. 2019) treat anomaly detection as an unsupervised problem. They exploit heuristic intuitions that the anomalies in a dataset are both rare and somehow different from the normal examples in order to assign a real-valued score to each example denoting how anomalous it is. This enables ranking the examples from most to least anomalous. The practical question is deciding how many and which anomalies to flag to a user by converting such a ranking into discrete predictions. While this is a complicated issue, the common approach would be to use a dataset's *contamination factor*, that is the expected proportion of anomalies in the dataset (Perini, Vercruyssen, and Davis 2020b), to set a threshold on the anomaly scores such that the proportion of examples with an anomaly score greater than the chosen threshold equals the contamination factor (Bandaragoda et al. 2018; Bergman and Hoshen 2020; Vikram et al. 2020). Examples with an anomaly score below the threshold are considered normal, those with a score larger than the threshold are the anomalies. The contamination factor is usually assumed to be known (e.g., from domain knowledge). The alternative is to estimate it by sampling and labeling some data (Perini, Vercruyssen, and Davis 2020a).

However, real-world anomaly detection tasks often involve monitoring a *fleet of related entities* such as machines (Randall 2011), windmill farms (Zhao et al. 2018) or retail stores (Vercruyssen et al. 2018). While the entities' behaviors are related in such cases, there are important differences that will affect the collected data. For example, windmill-specific properties (e.g., orientation, size, location) or store-specific properties (e.g., size, services, or opening hours) will affect the observed data. Consequently, how many anomalies are present will vary from entity to entity. Given that such tasks may involve monitoring 100s of entities, estimating the contamination factor for each one separately by labeling data would be too onerous. Thus, an interesting avenue to explore is whether it is possible to *transfer* a known contamination factor from data about one entity (the source domain) to data collected from another similar entity (the target domain). If this were possible, it would significantly decrease the labeling burden, as one would no longer need to collect labels for

all entities.

This paper proposes TRADE (*transferring the contamination factor between anomaly detection domains by shape similarity*), the first algorithm for transferring the known contamination factor from a source domain to a target domain where it is unknown. TRADE’s key assumption is that if the distributions over the anomaly scores of the normal examples computed by a given anomaly detection algorithm, are similar in shape in both the source and target domain, the target anomaly score threshold can be derived from the (known) source threshold. First, we use the known source contamination factor to construct a proper distribution over the normal examples in the source domain. Then, we find a threshold on the target domain anomaly scores that makes the distribution over the anomaly scores of the resulting “normal” target examples as similar as possible to the earlier-derived source distribution. This is constructed as an optimization problem. Finally, we use the resulting threshold to infer the target domain’s contamination factor. We theoretically analyze our approach and prove that the estimated target contamination factor converges to its true value when the distribution of the target scores becomes closer to their ground-truth distribution. Empirically, we performed an extensive evaluation on 206 source-target pairs arising from three real-world domains: detecting anomalous water usage in retail stores, detecting blade icing on windmills, and detecting botnets on IoT traffic data. We find that TRADE outperforms multiple competitors.

## Related Work

A first related research line looks at combining transfer learning with anomaly detection in different application domains. For instance, time series anomaly detection (Wen and Keyes 2019), detecting dangerous aircraft test flight actions (Xiong et al. 2018), hyperspectral image anomaly detection (Li, Wu, and Du 2017), or video anomaly detection (Bansod and Nandedkar 2019; Liu et al. 2020). Some authors focus on instance-transfer for anomaly detection (Vercruyssen, Meert, and Davis 2017, 2020), others on feature-based transfer (Kumagai, Iwata, and Fujiwara 2019; Yamaguchi, Koizumi, and Harada 2019), or model-based transfer (Wang et al. 2019; Idé, Phan, and Kalagnanam 2017; Du et al. 2013). The goal is almost always to improve a target model using source domain label information, i.e., deriving better estimates for the anomaly scores. However, no work looks at transferring the contamination factor between domains in the anomaly detection setting, allowing us to set a prediction threshold on these anomaly scores.

A second related research line revolves around converting anomaly scores into calibrated probabilities (Gao and Tan 2006). Although calibration usually requires either labeled examples or a *known* contamination factor, Kriegel et al. (2011) introduce UNIFY, a method to obtain calibrated probabilities from anomaly scores without such requirements. In absence of labeled data, Marques et al. (2020) develop an internal measure to evaluate the quality of an anomaly detector, while Schubert et al. (2012) and Perini et al. (2020) develop rank similarity measures to compare the anomaly rankings of different detectors. However, none of these works propose a method to find an appropriate decision threshold

for the anomaly scores in an (unlabeled) dataset. On the other hand, TRADE proposes a concrete algorithm for deriving the contamination factor of an unlabeled target dataset given the relevant source information.

## Preliminaries

Let  $(\Omega, \mathfrak{F}, \mathbb{P})$  be a probability space. Let  $X^S, X^T : \Omega \rightarrow \mathbb{R}^d$  be two multivariate real random variables with values in the feature space  $\mathbb{R}^d$ , and  $Y^S, Y^T : \Omega \rightarrow \{0, 1\}$  be the related class label (i.e., normal or anomalous). Assume that  $D^S$  and  $D_m^T$  are respectively the source and target dataset.  $D^S$  can be seen as an i.i.d. sample drawn from the joint distribution  $(X^S, Y^S)$ , while  $D_m^T$  ( $|D_m^T| = m$ ) is a small (and therefore potentially biased) sample drawn from  $(X^T, Y^T)$ . From now on, every target domain variable is indicated with the index  $m$ , referring to the number of target examples. An anomaly detection problem is the setting where there exists a measurable function  $h : \mathbb{R}^d \rightarrow \mathbb{R}$  that maps the examples in a dataset to a real-valued anomaly score. We focus on the *anomaly score random variables*  $S, T$  and  $T_m$  referring to the ground-truth anomaly scores of, respectively, the source domain, the target domain, and the sampled target dataset. We indicate their distributions with  $s, t$  and  $t_m$ . We linearly normalize the distributions  $s, t$  and  $t_m$  to have support in  $[0, 1]$ . Formally, we define their contamination factors as  $\gamma^S = \mathbb{P}(Y^S = 1)$ ,  $\gamma^T = \mathbb{P}(Y^T = 1)$ ,  $\gamma_m^T = \mathbb{P}(Y_m^T = 1)$ , where 1 is the anomaly class.

## Transferring the Contamination Factor by Shape Similarity

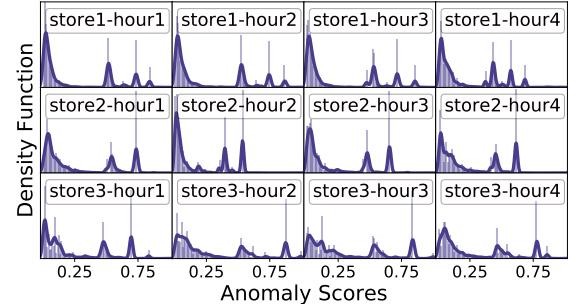


Figure 1: Illustration of how the distribution anomaly scores produced by the same anomaly detection algorithm  $h$  on related real-world water exhibit a similar shape.

This paper tackles the following problem:

**Given:** an unlabeled source dataset  $D^S$  with a known contamination factor  $\gamma^S$ , an unlabeled target dataset  $D_m^T$ , and an anomaly detection algorithm  $h$ ;

**Estimate:** the contamination factor  $\gamma_m^T$  of the target domain.

Our method TRADE estimates  $\gamma_m^T$  and works as follows. First, TRADE trains two separate anomaly detectors. It trains one on the source data and uses it to assign an anomaly score to each example in  $D^S$ . It trains the other one on the target data and uses it to assign an anomaly score to each

example in  $D^T$ . Because the domains are related and normal behaviors are similar, the key insight is that the source and target distributions of the normal examples' anomaly scores will be **similar** (but not necessarily equal). That is, there may be scales, offsets or shifts but not fundamental changes in the underlying distribution. Figure 1 motivates this assumption, showing that anomaly scores produced by algorithm  $h$  on multiple related domains follow a similar distribution when looking only at the low scores, which by construction correspond to the normal examples. However, because both datasets are unlabeled, we do not know the distribution of the normal examples' anomaly scores. Second, TRADE uses the known source contamination factor  $\gamma^S$  to set a threshold  $\lambda^S$  on the source anomaly scores. Examples with an anomaly score lower than  $\lambda^S$  are considered normal, yielding the distribution over their anomaly scores, which we call the  $\lambda$ *cut* distribution. Third, TRADE derives the target threshold  $\lambda_m^T$  by solving an optimization problem:  $\lambda_m^T$  is chosen such that shapes of the resulting  $\lambda_m^T$ *cut* distribution and the  $\lambda^S$ *cut* distribution are as **similar** as possible. This leverages our earlier insight. Finally, TRADE predicts the target contamination factor  $\gamma_m^T$  as the proportion of target examples with an anomaly score above the value of  $\lambda_m^T$ . The following subsections describe each of these steps in details. Next, we explore the theoretical properties of TRADE.

### Modeling the distribution of the anomaly scores of the normal examples in $D^S$

Modeling the distribution of anomaly scores assigned to the source normal examples is challenging because we lack labels. Instead, we exploit the fact that the source domain's contamination factor  $\gamma^S$  is known. First, we set a threshold  $\lambda^S$  on the source anomaly scores such that the proportion of examples with score  $> \lambda^S$  is equal to  $\gamma^S$ . Then, we model the distribution of normal scores as the distribution of scores  $\leq \lambda^S$  by performing a normalization such that the support of the new distribution is again  $[0, 1]$  and its area is equal to 1. More generally, for an arbitrary threshold value  $\lambda$ , we call this derived distribution the  $\lambda$ *cut distribution* and define it as follows:

**Definition 1.** Let  $X$  be a random variable on the probability space  $(\Omega, \mathfrak{F}, \mathbb{P})$  with distribution  $p(x)$  and support in  $[0, 1]$ . Then, for any  $\lambda \in [0, 1]$ , we define the  $\lambda$ *cut* distribution as:

$$p^\lambda(x) := p(\lambda x) \cdot \frac{\lambda}{\int_0^\lambda p(y) dy}.$$

**Proposition 1.** For any  $\lambda \in [0, 1]$ ,  $p^\lambda(x)$  is a distribution.

*Proof.* See the online Appendix<sup>1</sup> for the formal proof.  $\square$

This step assumes that the anomaly detection algorithm yields a reasonably ranking of the examples from least to most anomalous. However, even if the ranking is not perfect, the subsequent transfer step can still be accurate because the same algorithm is used to derive both the source and target  $\lambda$ *cut* distributions. Thus, incorrect predictions are likely similarly distributed in both domains.

<sup>1</sup><https://github.com/Lorenzo-Perini/TransferContamination>

### Finding the target threshold $\lambda_m^T$ via transfer

If we knew the threshold  $\lambda_m^T$  on the target anomaly scores that separates the normal examples from the anomalies, we could trivially estimate the target contamination factor. Therefore, we attempt to derive  $\lambda_m^T$  by exploiting our assumption that the source and target distributions of the normal examples' anomaly scores are similar (given they are derived using the same anomaly detector). This can be solved by attempting to find a value  $\lambda_m^T$  that yields a  $\lambda_m^T$ *cut* distribution in the target domain that is similar to the source's  $\lambda^S$ *cut* distribution. We can measure the similarity between two distributions  $S$  and  $T$  using the Kullback-Leibler (KL) divergence:

$$KL(S \parallel T) = \int_0^1 s(x) \log \left( \frac{s(x)}{t(x)} \right) dx,$$

where  $s$  and  $t$  are continuous distributions. Intuitively, the KL divergence quantifies the amount of information lost when approximating  $S$  with  $T$  with small KL divergence scores corresponding to little lost information, and hence similar shapes. We selected the KL divergence for three reasons. First, its theoretical properties enables a convergence study (Garrido 2009). Second, it is a widely used measure in the literature (Belov and Armstrong 2011). Third, it is stronger than several other similarity measures (e.g., maximum gap) as they are upper boundaries of KL (Gibbs and Su 2002).

We formulate our task as finding the threshold  $\lambda_m^T$  such that the KL divergence between the corresponding target  $\lambda_m^T$ *cut* distribution and the source's  $\lambda^S$ *cut* distribution is minimal:

$$\lambda_m^T = \arg \min_{\lambda \in [\delta, 1]} \left\{ KL \left( S^{\lambda^S} \parallel T_m^\lambda \right) \right\}, \quad (1)$$

where  $S^{\lambda^S}$  and  $T_m^\lambda$  are the random variables that follow, respectively, the  $\lambda^S$ *cut* and  $\lambda$ *cut* distributions. The  $\delta > 0$  is a small value which depends on the detector  $h$  and on the datasets, and represents the lower boundary for the choice of  $\lambda_m^T$ . The contamination factor is usually small such that  $\lambda_m^T > 0$ . If  $\lambda_m^T = 0$ , all the examples would be anomalous.

Theoretically, there may be more than one solution to Equation 1 because the objective might not be smooth such that  $\arg \min$  returns a set of solutions. However, in practice this is unlikely to occur and it did not happen in our experiments.

### Deriving the target contamination factor

Mirroring the reasoning for setting the source threshold  $\lambda^S$ , a reasonable estimate of the target domain's contamination factor can be derived by looking at the proportion of examples in the target domain with an anomaly score greater than  $\lambda_m^T$ . Theoretically, given the target threshold  $\lambda_m^T \in [\delta, 1]$  we should estimate the contamination factor through the continuous score variable  $T_m$  as  $\mathbb{P}(h(X^T) \geq \lambda_m^T) = \mathbb{P}(T_m \geq \lambda_m^T)$ . However, because in practice we can only use a finite number of examples, we estimate the contamination factor as the discrete proportion of examples with anomaly scores greater than  $\lambda_m^T$ :

$$\hat{\gamma}_m^T := \frac{|\{h(x) \geq \lambda_m^T \mid x \in D_m^T\}|}{m} = \frac{\sum_{i=1}^m \mathbb{1}_{\{h(x_i) \geq \lambda_m^T\}}(x_i)}{m} \quad (2)$$

where  $|\cdot|$  indicates the cardinality of a set,  $\mathbb{1}$  is the indicator function,  $h(x)$  is the anomaly score of the example  $x \in D_m^T$  and  $\lambda_m^T$  is the transferred target predictive threshold. In the following proposition, we prove that if the target threshold  $\lambda_m^T$  is correct, our estimator  $\hat{\gamma}_m^T$  is unbiased, meaning that it recovers the target domain's true contamination factor  $\gamma_m^T$ .

**Proposition 2.** *Given the target threshold  $\lambda_m^T \in [\delta, 1]$  such that  $\mathbb{P}(Y_m^T = 1) = \mathbb{P}(T_m \geq \lambda_m^T)$ , the contamination factor's estimator defined in Eq. 2 is unbiased.*

*Proof.* See the online Appendix for the formal proof.  $\square$

### Choice of anomaly detection algorithm $h$

In theory, TRADE can use any anomaly detection algorithm  $h$  to estimate  $\hat{\gamma}_m^T$ . In practice, we find that using an ensemble of anomaly detectors yields better results.<sup>2</sup> First, each detector  $i$  in the ensemble produces an estimate of the target contamination factor as described above. Then, TRADE computes the final estimate  $\hat{\gamma}_m^T$  as a weighted average of each ensemble member's estimate. The weight of each member  $w_i$  is inversely proportional to its obtained KL divergence  $KL_i$ :

$$w_i = \frac{1}{|E|-1} \times \left( 1 - \frac{KL_i(S, T_m)}{\sum_{j=1}^{|E|} KL_j(S, T_m)} \right),$$

where  $|E|$  is the number of detectors in the ensemble. This weighting scheme awards ensemble members that produce similar score distributions for the source and target domain.

### Theoretical Convergence Analysis

Our main theoretical result is Theorem 3, which states that our approach for estimating the contamination factor will converge to the theoretical target value in the limit. This theorem rests on making the following two theoretical assumptions.

**Assumption 1.** We assume that the sample of scores from the source domain is an i.i.d. sample drawn from the real distribution  $S$ . This is coherent with a practical setting, where the source sample is large enough to represent the ground truth distribution. On the other hand, we assume that there may be some bias in the distribution of scores  $T_m$  with respect to  $T$ , and that the bias gradually fades out when adding examples. Formally, we require that, for  $m \rightarrow +\infty$ ,  $t_m \rightarrow t$  uniformly in  $[0, 1]$ , which means that, for every  $\varepsilon > 0$ , there exists  $M \in \mathbb{N}$  such that, for all  $m \geq M$  and  $x \in [0, 1]$ , the inequality  $|t(x) - t_m(x)| < \varepsilon$  holds. We also indicate this assumption by  $T_m \rightarrow T$ .

**Assumption 2.** We assume that the normal scores distribution of the theoretical target distribution  $T$  shares exactly the same shape with the normal scores distribution of the source domain. Formally, we require that  $KL(S^{\lambda^S} \parallel T^{\lambda^T}) = 0$ , where  $S^{\lambda^S}$  and  $T^{\lambda^T}$  represents the distribution of normal scores of the two domains. This assumption is a theoretical generalization of what Figure 1 shows.

Formally, our **main theoretical result** is stated as:

**Theorem 3.** *Let  $S$  and  $T_m$  be two continuous random variables representing the anomaly scores produced by an anomaly detector  $h$  on, respectively, the source ( $D^S$ ) and the target ( $D_m^T$ ) domains. Assume that  $T$  is the random variable with the ground-truth distribution of the target domain scores. Let  $\gamma^S$  be the contamination factor of the source domain. Let us fix  $\delta > 0$  small enough and let  $\lambda^S$  and  $\lambda^T$  be the real predictive thresholds of  $S$  and  $T$ . Let's assume that  $s$ ,  $t$  and  $t_m$  are the positive distributions of  $S$ ,  $T$  and  $T_m$  such that  $t_m \rightarrow t$  uniformly in  $[0, 1]$  (Assumption 1) for  $m \rightarrow +\infty$  and that  $KL(S^{\lambda^S} \parallel T^{\lambda^T}) = 0$  (Assumption 2). Also, let  $\lambda_m^T \in [\delta, 1]$  be the estimate of the target predictive threshold through Eq. 1. Then,*

$$\lim_{m \rightarrow +\infty} \lambda_m^T = \lambda^T.$$

*Furthermore, let  $\hat{\gamma}_m^T$  be the estimate of the target contamination factor by the estimator defined in Eq. 2. Then,*

$$\mathbb{E}[\hat{\gamma}_m^T] \rightarrow \gamma^T \quad \text{for } m \rightarrow +\infty.$$

*Proof.* We now sketch the proof for this theorem. The detailed proofs are in the online Appendix along with the supporting theorems used in the sketch.<sup>3</sup> In order to prove the first part, we need to motivate the transition of the limit symbol through the functions, following these steps:

$$\begin{aligned} \lambda^T &\stackrel{(i)}{=} \arg \min_{\lambda \in [\delta, 1]} \left\{ KL \left( S^{\lambda^S} \parallel T^\lambda \right) \right\} \\ &\stackrel{(ii)}{=} \arg \min_{\lambda \in [\delta, 1]} \left\{ KL \left( S^{\lambda^S} \parallel \lim_{m \rightarrow +\infty} T_m^\lambda \right) \right\} \\ &\stackrel{(iii)}{=} \arg \min_{\lambda \in [\delta, 1]} \left\{ \lim_{m \rightarrow +\infty} KL \left( S^{\lambda^S} \parallel T_m^\lambda \right) \right\} \\ &\stackrel{(iv)}{=} \limsup_{m \rightarrow +\infty} \arg \min_{\lambda \in [\delta, 1]} \left\{ KL \left( S^{\lambda^S} \parallel T_m^\lambda \right) \right\} \stackrel{(v)}{=} \lim_{m \rightarrow +\infty} \lambda_m^T. \end{aligned} \tag{3}$$

The first (i) and the last (v) equalities come from the uniqueness of the solution shown in Theorems 4 and 5; the second step (ii) is motivated by the convergence of  $\lambda$ -cut distributions proved in Theorem 6; the third equality (iii) holds by Theorem 7; the fourth result (iv) is guaranteed by Theorems 8 and 9. Note that the equal in (iv) is not an inclusion because of the uniqueness of the solution  $\lambda^T$  (shown in Theorem 5).

Once we proved that the threshold converges as expected, the second part of this theorem focuses on the contamination factor's convergence, which comes directly as follows:

$$\begin{aligned} \lim_{m \rightarrow +\infty} \mathbb{E}[\hat{\gamma}_m^T] &\stackrel{(i)}{=} \lim_{m \rightarrow +\infty} \mathbb{E} \left[ \frac{\sum_{i=1}^m \mathbb{1}_{\{h(x) \geq \lambda_m^T\}}(x_i)}{m} \right] \\ &\stackrel{(ii)}{=} \lim_{m \rightarrow +\infty} \frac{\sum_{i=1}^m \mathbb{E}[\mathbb{1}_{\{h(x) \geq \lambda_m^T\}}(x_i)]}{m} \\ &\stackrel{(iii)}{=} \lim_{m \rightarrow +\infty} \frac{\sum_{i=1}^m \mathbb{E}[\mathbb{1}_{\{T_m \geq \lambda_m^T\}}]}{m} \\ &\stackrel{(iv)}{=} \mathbb{E} \left[ \lim_{m \rightarrow +\infty} \mathbb{1}_{\{T_m \geq \lambda_m^T\}} \right] \stackrel{(v)}{=} \mathbb{E}[\mathbb{1}_{\{T \geq \lambda^T\}}] \\ &\stackrel{(vi)}{=} \mathbb{P}(T \geq \lambda^T) = \gamma^T. \end{aligned}$$

<sup>2</sup>We provide empirical evidence in the experimental section.

<sup>3</sup><https://github.com/Lorenzo-Perini/TransferContamination>

The first equality (i) holds by our definition of the estimator (Eq. 2); the second step (ii) exploits the properties of the expectation; the third equality (iii) follows from the fact that  $x_i$  is i.i.d.; the interchange between the expectation and the limit (iv) is allowed by the theorem of dominated convergence; the result of the limit (v) is motivated by both the assumptions of uniform convergence convergence ( $T_m \rightarrow T$ ) and the first part of this theorem ( $\lambda_m^T \rightarrow \lambda^T$ ); finally, the last step (vi) is a property of the characteristic function.  $\square$

## Experiments

We address the following four experimental questions:

- Q1.** Does TRADE accurately estimate the true target contamination factor?
- Q2.** Does a more accurate estimate of the target contamination factor improve the performance of the anomaly detector?
- Q3.** Does an ensemble of anomaly detectors produce a more accurate estimate of the target contamination factor than a single detector  $h$ ?
- Q4.** How does TRADE perform when varying the source contamination factor?

## Experimental Setup

**Methods.** We compare TRADE<sup>4</sup> against five baselines. SOURCE $_{\gamma}$  simply assumes the target contamination factor to be equal to the source contamination factor. SOURCE $_{\lambda}$  first uses an ensemble to estimate  $\lambda_m^T$  through a simple average of the ensemble members' estimates. Then, it estimates the target contamination factor as the proportion of target examples with anomaly score  $> \lambda_m^T$ . CORAL (Sun, Feng, and Saenko 2017) is an unsupervised domain adaptation technique that transforms the source distribution to be similar to the target distribution. After applying this transformation, it uses SOURCE $_{\lambda}$  approach to estimate the target contamination factor. Finally, UNIFY (Kriegel et al. 2011) and OTSU (Otsu 1979) are unsupervised approaches that can be applied to the target anomaly scores. The former transforms the anomaly scores into posterior probabilities and estimates the contamination factor as the proportion of target examples with posterior anomaly probability  $> 0.5$ . The latter selects the best-separating threshold by minimizing the intra-class variance and estimates the contamination factor as the proportion of scores above the threshold.

**Data.** Our experiments focus on how anomaly detection can impact real-world sustainability and security. Specifically, we look at preventing water loss, preventing blade icing in wind turbines, and detecting IoT traffic anomalies. For the first task, we use 12 proprietary water consumption datasets obtained in collaboration with a large retail company.<sup>5</sup> Each dataset contains the water consumption measured each day during a particular hour-long segment in one of three retail stores over the course of 4.5–5 years. The measurement interval is 5 minutes. The raw consumption data of each hour-long

segment are transformed into feature-vectors.<sup>6</sup> The goal is to detect hours of anomalous consumption (e.g., a leak). Accurate detection of the anomalies aids the company in preventing water losses, which can otherwise easily amount to 1000s of litres a year. For the second task, we use two public wind turbine datasets (Zhang et al. 2018). Various measurements (e.g., wind speed, power, etc.) are collected approximately every 7 seconds for either two months (turbine 15) or one month (turbine 21). We construct feature-vectors from the data as in the original paper, averaging over time segments of 1 hour. The goal is to detect ice formation on wind turbine blades, which could potentially damage the turbines and slow power production. To obtain the wind turbine data, see the original paper (Zhang et al. 2018). For the third task, we use 9 public<sup>7</sup> IoT datasets (Meidan et al. 2018; Mirsky et al. 2018). Each dataset contains real traffic data, collected from one commercial IoT device infected by authentic botnets in an isolated network. The features include statistics on the stream data (e.g., source IP, MAC, channel jitter, socket), time-frame (e.g., the decay factor), and statistics extracted from the packet stream (e.g., weight, mean, std, radius, magnitude) for a total of 115 attributes. For computational reasons, we use a random subsample of 2000 examples for each dataset. The online Appendix contains additional details.

**Setup.** Each experiment goes as follows: (i) pick a source-target dataset pair from the benchmark; (ii) train a separate anomaly detector on both the source and target domains and use them to compute the anomaly scores; (iii) estimate the target contamination factor and use it to make the target anomaly predictions; (iv) evaluate the estimated contamination factor using the *mean absolute error* (MAE) and the predictions using the *F<sub>1</sub> score*; and (v) derive the average relative improvements:

$$\text{MAE improvement} = \frac{\text{MAE}_{\text{BASELINE}} - \text{MAE}_{\text{TRADE}}}{\text{MAE}_{\text{BASELINE}}};$$

$$\text{F}_1 \text{ improvement} = \frac{\text{F}_1_{\text{TRADE}} - \text{F}_1_{\text{BASELINE}}}{\text{F}_1_{\text{BASELINE}}}.$$

In step (i) we do not mix the three types of datasets, as it would violate Assumption 2. For the water and wind turbines tasks, each dataset serves once as the target domain while the remaining ones serve as a source domain yielding  $12 \times 11 + 2 \times 1 = 134$  source-target pairs. For the IoT data, before taking a subsample we set the target contamination factor to 0.01 and vary the source contamination factor in  $[0.03, 0.05, 0.08, 0.10, 0.15, 0.20, 0.25]$ . This results in  $9 \times 8 = 72$  source-target pairs.

**Hyperparameters.** TRADE, SOURCE $_{\lambda}$ , CORAL, UNIFY, and OTSU all use an ensemble of 9 unsupervised anomaly detectors from different families (proximity-based, isolation-based, density-based, and reconstruction-based): the *k-Nearest Neighbours Detector* (KNNO) (Ramaswamy, Rastogi,

<sup>6</sup>We use 9 statistical (average, standard deviation, max, min, median, sum, entropy, skewness, kurtosis) and 2 binary features (whether its Friday or Sunday), 11 in total.

<sup>7</sup>[https://archive.ics.uci.edu/ml/datasets/detection\\_of\\_IoT\\_botnet\\_attacks\\_N\\_BaIoT](https://archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT)#

<sup>4</sup><https://github.com/Lorenzo-Perini/TransferContamination>

<sup>5</sup>The data was provided under an NDA and cannot be shared.

and Shim 2000), the *Clustering Based Local Outlier Factor* (CBLOF) (He, Xu, and Deng 2003), the *Histogram-based Outlier Detection* (HBOS) (Goldstein and Dengel 2012) and the *Subspace Outlier Detection* (SOD) (Kriegel et al. 2009), the *Isolation Forest Outlier Detector* (iFOREST) (Liu, Ting, and Zhou 2008), the *Copula Based Outlier Detector* (COPOD) (Li et al. 2020), the *Lightweight On-line Detector of Anomalies* (LODA) (Pevný 2016), the *Locally Selective Combination of Parallel Outlier Ensembles* (LSCP) (Zhao et al. 2019) with three *Local Outlier Factor* (LOF) (Breunig et al. 2000) as density based local detectors, and the *Variational AutoEncoder* (VAE) (Burgess et al. 2018). Their hyperparameters are set to the default values (Soenen et al. 2021).<sup>8</sup>

TRADE uses *differential evolution* (Storn and Price 1997) (maxit. = 100, mut. = 0.4, rec. = 0.2) as the optimization solver. We restrict the solution to be in the interval (0, 0.25).

**Computational cost.** The most expensive step of TRADE is the optimization algorithm. For a single experiment, the CPU time is  $\sim 10000$  seconds. To run all experiments, we use an internal cluster of six 24- or 32-thread machines (128 GB of memory). The experiments finish in  $\sim 24$  hours.

## Experimental Results

**Q1. Estimating the target contamination factor  $\gamma_m^T$ .** Table 1 (left) summarizes the results of using TRADE and the baselines to estimate the target contamination factor in each of the 206 source-target pairs. TRADE obtains the lowest (best) average MAE rank (computed following (Demšar 2006)). On average, it achieves the lowest MAE of the target contamination factor’s estimate across all experiments. TRADE estimates  $\gamma_m^T$  with a lower/similar error than each baseline in at least  $\sim 73\%$  of the experiments.

Figure 2 (left) shows TRADE’s average improvement in MAE compared to the baselines aggregated for each of the 23 target domains. Positive values imply that TRADE achieves a lower, i.e., better, MAE. TRADE produces better average estimates of the target contamination factor on 17 target domains vs. UNIFY, 18 vs. SOURCE $_\gamma$ , 21 vs. CORAL, SOURCE $_\lambda$  and OTSU.

We perform the Friedman rank test to test the null-hypothesis that all compared methods perform similarly (Demšar 2006; Iman and Davenport 1980). The obtained Friedman corrected statistic of 59 and corresponding p-value of  $\approx 10^{-16}$  allow us to reject this null-hypothesis. Applying the Bonferroni-Dunn post-hoc test (Dunn 1961) with  $\alpha = 5$ , shows that TRADE’s performance is statistically significantly better than all the baselines.

**Q2. Impact of estimating the target contamination factor correctly on the performance of the anomaly detector.** We evaluate how TRADE’s target contamination factor estimate (and that of the baselines) affects the target detector’s anomaly detection performance through the following experiment: (i) pick one of the 206 source-target pairs; (ii) use TRADE or one of the baselines to estimate the target contamination factor; (iii) compute the target anomaly scores using an anomaly detector on the target domain; (iv) use the

estimated contamination factor to convert the anomaly scores to hard predictions and compute the  $F_1$  score. To avoid the results being dependent on one specific anomaly detector, we repeat the experiment for each of the 9 considered detectors resulting in  $206 \times 9 = 1854$  experiments. We compute the  $F_1$  score because it *strictly* depends on using the target contamination factor  $\gamma_m^T$  to make hard predictions. In contrast, the AUC metric commonly used in anomaly detection (Campos et al. 2016), only evaluates a detector’s capability to rank examples correctly and does not change when  $\gamma_m^T$  changes.

Table 1 (right) summarizes the results of the  $F_1$  score obtained using the target contamination factor estimated by TRADE and the baselines in each of the 1854 experiments. TRADE has the lowest (best) average  $F_1$  rank. On average, TRADE enables the anomaly detector to achieve higher/similar  $F_1$  scores in at least 65% of the experiments.

Figure 2 (right) shows TRADE’s average improvement in  $F_1$  score compared to the baselines aggregated for each of the 23 target domains. Positive values indicate that TRADE obtains higher  $F_1$  scores. TRADE results in higher average  $F_1$  scores on 17 target domains vs. OTSU, 18 vs. SOURCE $_\lambda$ , 20 vs. UNIFY, 21 vs. SOURCE $_\gamma$ , and 22 vs. CORAL.

**Q3. Ensemble versus single anomaly detectors.** Our method uses an ensemble of anomaly detectors to estimate the target contamination factor and set the threshold. To see the effect of this choice, we compare TRADE using the ensemble with variants of TRADE using only one of the nine detectors. For computational reasons, this experiment only considers the water and wind turbines data. Compared to using a single detector, the ensemble results in an equivalent or better estimate of the contamination factor on between 59% (vs. iFOREST variant) to 85% (vs. HBOS variant) of the experiments. Overall, the ensemble variant reduces the MAE from 12% (vs. iFOREST variant) to 50% (vs. KNNO variant).

**Q4. The effect of varying the source contamination factor  $\gamma^S$ .** In the IoT dataset, the target contamination is always 0.01. Therefore we explore the effect on performance of varying the source contamination factor. Figure 3 reports the TRADE’s average improvement in MAE over the baselines as a function of the source contamination factor. Because SOURCE $_\gamma$  and SOURCE $_\lambda$  depend on the source  $\gamma^S$ , TRADE achieves better results when  $\gamma^S$  increases. Compared to these methods, TRADE’s performance is not as adversely affected by increasing the difference between the source and the target contamination factors. Because UNIFY and OTSU are unsupervised methods using only the target domain, their estimate is constant as it does not depend on the source contamination factor. TRADE results in (large) gains over UNIFY and OTSU even for relatively large gaps between the source and target contamination factor (e.g., 0.01 for the target and 0.15 for the source). As the gap between the source and target contamination factor grows, TRADE wins in performance vs. UNIFY and OTSU shrinks, with the two baselines outperforming TRADE at the largest gaps.

## Discussion and Conclusion

We proposed a novel method TRADE for estimating the target domain contamination factor given a source dataset

<sup>8</sup>See the appendix for details.

Table 1: Comparison of TRADE with the baselines. The left-hand side of the table shows the average MAE of each method’s estimate of the target contamination factor, the average MAE rank  $\pm$  standard deviation (SD) of each method, and the number of times TRADE wins (lower MAE), draws, and loses (higher MAE) against each baseline (absolute differences  $\leq 0.001$  count as draw). The right-hand side of the table shows similar information for the  $F_1$  score, averaged over the 9 considered detectors.

Error on $\gamma$	MAE		Ranking		# times TrADE			$F_1$ score		Ranking		# times TrADE			
	Method	Avg. $\pm$ SD	Avg. $\pm$ SD	W	D	L	Avg. $\pm$ SD	Avg. $\pm$ SD	W	D	L	Avg. $\pm$ SD	W	D	
TRADE	<b>0.054 <math>\pm</math> 0.031</b>	<b>2.03 <math>\pm</math> 1.01</b>	-	-	-	0.35 $\pm$ 0.19	<b>2.58 <math>\pm</math> 1.44</b>	-	-	-	-	0.35 $\pm$ 0.19	149	654	
SOURCE $_{\gamma}$	0.074 $\pm$ 0.040	3.04 $\pm$ 1.56	<b>146</b>	5	55	0.31 $\pm$ 0.19	3.05 $\pm$ 1.51	<b>1051</b>	149	654	60	464	60	464	
SOURCE $_{\lambda}$	0.108 $\pm$ 0.083	3.82 $\pm$ 1.65	<b>151</b>	5	50	0.29 $\pm$ 0.18	3.85 $\pm$ 1.48	<b>1330</b>	92	439	92	439	92	439	
CORAL	0.114 $\pm$ 0.080	4.26 $\pm$ 1.55	<b>174</b>	1	31	0.28 $\pm$ 0.20	4.03 $\pm$ 1.59	<b>1323</b>	76	589	76	589	76	589	
UNIFY	0.077 $\pm$ 0.047	3.52 $\pm$ 1.49	<b>161</b>	4	41	0.29 $\pm$ 0.18	3.62 $\pm$ 1.71	<b>1189</b>	11	582	11	582	11	582	
OTSU	0.137 $\pm$ 0.078	4.33 $\pm$ 1.74	<b>177</b>	0	29	0.27 $\pm$ 0.14	3.88 $\pm$ 1.92	<b>1261</b>	-	-	-	-	-	-	-

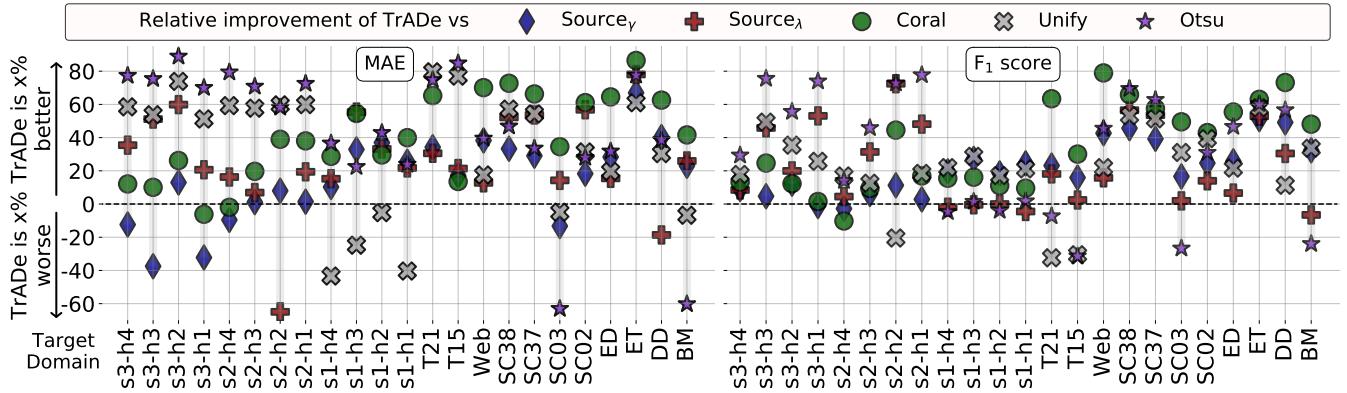


Figure 2: Average relative improvement in MAE (left) and the  $F_1$  (right) of TRADE versus each baseline, aggregated per target domain (x-axis). Positive values indicate that TRADE performs better than the baseline. For each target domain, TRADE’s relative improvement in MAE varies between 15% (vs SOURCE $_{\gamma}$ ) and 40% (vs CORAL), while the  $F_1$  score improves by at least 22% (vs SOURCE $_{\gamma}$ ) and up to 35% (vs CORAL).

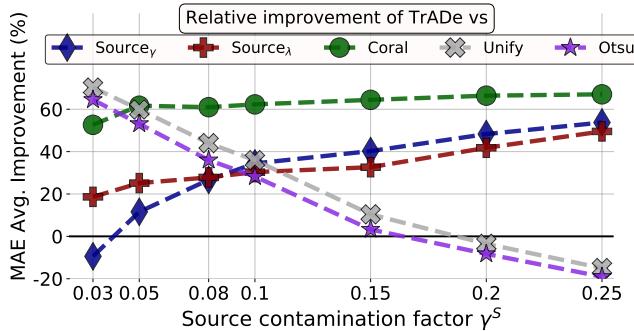


Figure 3: TRADE’s relative improvement in MAE versus each baseline as a function of the source contamination factor on the IoT datasets. As the gap between the source and target contamination factors increases, TRADE performance gains versus SOURCE $_{\gamma}$ , SOURCE $_{\lambda}$ , and CORAL grow. Moreover, when the gap is lower than 0.2, TRADE improves the MAE by up to 70% when compared to UNIFY.

with a known contamination factor. The key insight enabling our approach is that the distribution of the normal examples’ anomaly scores in both domains will be *similar* if they are derived using the same anomaly detection algorithm.

Theoretically, we proved that TRADE’s estimate of the contamination factor converges to its actual value when the size of the target dataset increases. Empirically, we demonstrated that TRADE can more accurately estimate the contamination factor than several baselines. More importantly, more accurate estimates lead to improved anomaly detection performance as shown by higher  $F_1$  scores.

**Benefits and limitations.** In the experiments we focused on anomaly detection in a sustainability context (preventing water losses in retail stores and blade icing in wind turbines) and security. The potential societal benefits, due to the more accurate detection models, are manifested in the avoidance of potentially costly anomalies (e.g., large water leaks). A potential downside would arise from missed detections and false alarms, which both result in real-world costs. Moreover, one could our approach to disadvantage or discriminate against marginalized groups, indicating them as anomalies.

## Acknowledgements

This work is supported by KU Leuven Research Fund C14/17/070 (JD), the Flemish government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” programme (JD, LP), by the “Agentschap Innoveren & Ondernemen” (VLAIO) as part of the innovation mandate HBC.2020.2297 (VV), and FWO-Vlaanderen aspirant grant 1166222N (LP).

## References

- Bandaragoda, T. R.; Ting, K. M.; Albrecht, D.; Liu, F. T.; Zhu, Y.; and Wells, J. R. 2018. Isolation-based anomaly detection using nearest-neighbor ensembles. *Computational Intelligence*, 34(4): 968–998.
- Bansod, S.; and Nandedkar, A. 2019. Transfer learning for video anomaly detection. *Journal of Intelligent & Fuzzy Systems*, 36(3): 1967–1975.
- Belov, D. I.; and Armstrong, R. D. 2011. Distributions of the Kullback–Leibler divergence with applications. *British Journal of Mathematical and Statistical Psychology*, 64(2): 291–309.
- Bergman, L.; and Hoshen, Y. 2020. Classification-based anomaly detection for general data. *arXiv preprint arXiv:2005.02359*.
- Breunig, M. M.; Kriegel, H.-P.; Ng, R. T.; and Sander, J. 2000. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 93–104.
- Burgess, C. P.; Higgins, I.; Pal, A.; Matthey, L.; Watters, N.; Desjardins, G.; and Lerchner, A. 2018. Understanding disentangling in  $\beta$ -VAE. *arXiv preprint arXiv:1804.03599*.
- Campos, G. O.; Zimek, A.; Sander, J.; Campello, R. J.; Mincenková, B.; Schubert, E.; Assent, I.; and Houle, M. E. 2016. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data mining and knowledge discovery*, 30(4): 891–927.
- Chandola, V.; Banerjee, A.; and Kumar, V. 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3): 1–58.
- Demšar, J. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan): 1–30.
- Du, B.; Zhang, L.; Tao, D.; and Zhang, D. 2013. Unsupervised transfer learning for target detection from hyperspectral images. *Neurocomputing*, 120: 72–82.
- Dunn, O. J. 1961. Multiple comparisons among means. *Journal of the American statistical association*, 56(293): 52–64.
- Gao, J.; and Tan, P.-N. 2006. Converting output scores from outlier detection algorithms into probability estimates. In *Sixth International Conference on Data Mining (ICDM’06)*, 212–221. IEEE.
- Garrido, A. 2009. About some properties of the Kullback–Leibler divergence. *Advanced Modeling and Optimization*, 11(4).
- Gibbs, A. L.; and Su, F. E. 2002. On choosing and bounding probability metrics. *International statistical review*, 70(3): 419–435.
- Goldstein, M.; and Dengel, A. 2012. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: Poster and Demo Track*, 59–63.
- He, Z.; Xu, X.; and Deng, S. 2003. Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9-10): 1641–1650.
- Idé, T.; Phan, D. T.; and Kalagnanam, J. 2017. Multi-task multi-modal models for collective anomaly detection. In *2017 IEEE International Conference on Data Mining (ICDM)*, 177–186. IEEE.
- Iman, R. L.; and Davenport, J. M. 1980. Approximations of the critical region of the friedman statistic. *Communications in Statistics-Theory and Methods*, 9(6): 571–595.
- Khan, S.; Liew, C. F.; Yairi, T.; and McWilliam, R. 2019. Unsupervised anomaly detection in unmanned aerial vehicles. *Applied Soft Computing*, 83: 105650.
- Kriegel, H.-P.; Kröger, P.; Schubert, E.; and Zimek, A. 2009. Outlier detection in axis-parallel subspaces of high dimensional data. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 831–838. Springer.
- Kriegel, H.-P.; Kroger, P.; Schubert, E.; and Zimek, A. 2011. Interpreting and unifying outlier scores. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, 13–24. SIAM.
- Kumagai, A.; Iwata, T.; and Fujiwara, Y. 2019. Transfer anomaly detection by inferring latent domain representations. In *Advances in Neural Information Processing Systems*, 2471–2481.
- Li, W.; Wu, G.; and Du, Q. 2017. Transferred deep learning for anomaly detection in hyperspectral imagery. *IEEE Geoscience and Remote Sensing Letters*, 14(5): 597–601.
- Li, Z.; Zhao, Y.; Botta, N.; Ionescu, C.; and Hu, X. 2020. CO-POD: copula-based outlier detection. In *IEEE International Conference on Data Mining (ICDM 2020)*. IEEE.
- Liu, F. T.; Ting, K. M.; and Zhou, Z.-H. 2008. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, 413–422. IEEE.
- Liu, K.; Zhu, M.; Fu, H.; Ma, H.; and Chua, T.-S. 2020. Enhancing Anomaly Detection in Surveillance Videos with Transfer Learning from Action Recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, 4664–4668.
- Marques, H. O.; Campello, R. J.; Sander, J.; and Zimek, A. 2020. Internal evaluation of unsupervised outlier detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 14(4): 1–42.
- Meidan, Y.; Bohadana, M.; Mathov, Y.; Mirsky, Y.; Shabtai, A.; Breitenbacher, D.; and Elovici, Y. 2018. N-baiot—network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, 17(3): 12–22.

- Mirsky, Y.; Doitshman, T.; Elovici, Y.; and Shabtai, A. 2018. Kitsune: an ensemble of autoencoders for online network intrusion detection. *arXiv preprint arXiv:1802.09089*.
- Otsu, N. 1979. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1): 62–66.
- Perini, L.; Galvin, C.; and Vercruyssen, V. 2020. A Ranking Stability Measure for Quantifying the Robustness of Anomaly Detection Methods. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 397–408. Springer.
- Perini, L.; Vercruyssen, V.; and Davis, J. 2020a. 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)*.
- Perini, L.; Vercruyssen, V.; and Davis, J. 2020b. Quantifying the confidence of anomaly detectors in their example-wise predictions. In *The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. Springer Verlag.
- Pevný, T. 2016. Loda: Lightweight on-line detector of anomalies. *Machine Learning*, 102(2): 275–304.
- Ramaswamy, S.; Rastogi, R.; and Shim, K. 2000. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 427–438.
- Randall, R. B. 2011. *Vibration-based condition monitoring: industrial, aerospace and automotive applications*. John Wiley & Sons.
- Robberechts, P.; Bosteels, M.; Davis, J.; and Meert, W. 2018. Query log analysis: Detecting anomalies in DNS traffic at a TLD resolver. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 55–67. Springer.
- Schubert, E.; Wojdanowski, R.; Zimek, A.; and Kriegel, H.-P. 2012. On evaluation of outlier rankings and outlier scores. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, 1047–1058. SIAM.
- Soenen, J.; Van Wolputte, E.; Perini, L.; Vercruyssen, V.; Meert, W.; Davis, J.; and Blockeel, H. 2021. The Effect of Hyperparameter Tuning on the Comparative Evaluation of Unsupervised Anomaly Detection Methods. In *Proceedings of the KDD’21 Workshop on Outlier Detection and Description*, 1–9. Outlier Detection and Description Organising Committee.
- Storn, R.; and Price, K. 1997. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4): 341–359.
- Sun, B.; Feng, J.; and Saenko, K. 2017. Correlation alignment for unsupervised domain adaptation. In *Domain Adaptation in Computer Vision Applications*, 153–171. Springer.
- Vercruyssen, V.; Meert, W.; and Davis, J. 2017. Transfer learning for time series anomaly detection. In *CEUR Workshop Proceedings*, volume 1924, 27–37.
- Vercruyssen, V.; Meert, W.; and Davis, J. 2020. Transfer learning for anomaly detection through localized and unsupervised instance selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 6054–6061.
- Vercruyssen, V.; Wannes, M.; Gust, V.; Koen, M.; Ruben, B.; and Jesse, D. 2018. Semi-supervised anomaly detection with an application to water analytics. In *Proceedings of 18th IEEE International Conference on Data Mining*, 527–536. IEEE.
- Vikram, A.; et al. 2020. Anomaly detection in Network Traffic Using Unsupervised Machine learning Approach. In *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, 476–479. IEEE.
- Wang, W.; Chen, Q.; He, X.; and Tang, L. 2019. Cooperative Anomaly Detection With Transfer Learning-Based Hidden Markov Model in Virtualized Network Slicing. *IEEE Communications Letters*, 23(9): 1534–1537.
- Wen, T.; and Keyes, R. 2019. Time series anomaly detection using convolutional neural networks and transfer learning. *arXiv preprint arXiv:1905.13628*.
- Xiong, P.; Zhu, Y.; Sun, Z.; Cao, Z.; Wang, M.; Zheng, Y.; Hou, J.; Huang, T.; and Que, Z. 2018. Application of transfer learning in continuous time series for anomaly detection in commercial aircraft flight data. In *2018 IEEE International Conference on Smart Cloud (SmartCloud)*, 13–18. IEEE.
- Yamaguchi, M.; Koizumi, Y.; and Harada, N. 2019. AdaFlow: Domain-adaptive density estimator with application to anomaly detection and unpaired cross-domain translation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3647–3651. IEEE.
- Zhang, L.; Liu, K.; Wang, Y.; and Omariba, Z. B. 2018. Ice detection model of wind turbine blades based on random forest classifier. *Energies*, 11(10): 2548.
- Zhao, H.; Liu, H.; Hu, W.; and Yan, X. 2018. Anomaly detection and fault analysis of wind turbine components based on deep learning network. *Renewable energy*, 127: 825–834.
- Zhao, Y.; Nasrullah, Z.; Hryniwicki, M. K.; and Li, Z. 2019. LSCP: Locally selective combination in parallel outlier ensembles. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, 585–593. SIAM.