8-8-2022

On Equivalence of Anomaly Detection Algorithms

Carlos Ivan Jerez

Jun Zhang

Marcia R. Silva

On Equivalence of Anomaly Detection Algorithms

CARLOS IVAN JEREZ, JUN ZHANG, and MARCIA R. SILVA, University of Wisconsin-Milwaukee, USA

In most domains anomaly detection is typically cast as an unsupervised learning problem because of the infeasability of labelling large datasets. In this setup, the evaluation and comparison of different anomaly detection algorithms is difficult. Although some work has been published in this field, they fail to account that different algorithms can detect different kinds of anomalies. More precisely, the literature on this topic has focused on defining criteria to determine which algorithm is better, while ignoring the fact that such criteria are meaningful only if the algorithms being compared are detecting the same kind of anomalies. Therefore, in this paper we propose an equivalence criterion for anomaly detection algorithms that measures to what degree two anomaly detection algorithms detect the same kind of anomalies. First, we lay out a set of desirable properties that such an equivalence criterion should have and why; second, we propose, Gaussian Equivalence Criterion (GEC) as equivalence criterion and show mathematically that it has the desirable properties previously mentioned. Finally, we empirically validate these properties using a simulated and a real-world dataset. For the real-world dataset, we show how GEC can provide insight about the anomaly detection algorithms as well as the dataset.

CCS Concepts: • **Computing methodologies** → **Anomaly detection**; • **Information systems**;

Additional Key Words and Phrases: Unsupervised learning, anomaly detection, comparison

1 INTRODUCTION

Anomaly detection (AD) is ubiquitous in many domains. Some of its applications are: intrusion detection, which refers to detecting malicious activity in a computer-related system; fraud detection such as credit card fraud, insurance fraud, tax fraud; medical anomaly detection, where the anomalies are detected in medical images or clinical electroencephalography (EEG) records to diagnose or prevent diseases; anomalies in social networks, which refers to irregular and often unlawful behavior patterns of individuals in social networks, some examples of these are scammers, sexual predators, online fraudsters; industrial AD, which refers to detecting anomalies in industrial processes that are carried out countless times; AD in autonomous vehicles to prevent attackers from taking over the vehicle, etc. [3, 27].

For real-world applications labelling can be expensive and time-consuming, for this reason AD is generally cast as an unsupervised learning problem. In this setup the learning is done directly from patterns naturally occurring in the data. Then, a follow-up question is: out of a set of algorithms, how to determine which AD algorithm is better and in what way without having labels? As it turns out, evaluating an algorithm's performance under the unsupervised learning setting is not a trivial task. Nonetheless, it is of paramount importance to be able

Authors' address: Carlos Ivan Jerez, jerez@uwm.edu; Jun Zhang, junzhang@uwm.edu; Marcia R. Silva, msilva@uwm.edu, University of Wisconsin-Milwaukee, 3200 N Cramer Street, Milwaukee, Wisconsin, USA, 53211-3029.

1556-4681/2022/5-ART $15.00
https://doi.org/10.1145/3536428

to compare AD algorithms beyond the supervised setting in a toy dataset (i.e., simulated datasets that include labels), and labelling real-world datasets is not a practical or scalable solution for most domains.

An anomaly is defined as a rare observation or an observation that deviates from the norm [3]. Mathematically, it can be described as a low-likelihood data point with respect to some unknown underlying likelihood function $f : \mathcal{X} \to \mathbb{R}^+$, where $\mathcal{X}$ is the spaces where the data lives. Furthermore, the set of anomalies is given by $a = \{x : f(x) < \epsilon\}$ for some small ϵ. The goal of anomaly detection is to approximate f or some form of it, especially in the regions of low likelihood. Note that if $g = T \circ f$ with T being a increasing transformation, then $a = \{x : g(x) < \delta\}$ using an appropriate δ yields exactly the same a as when using f, which is why approximating a "form" of f suffices.

An AD algorithm (sometimes called scoring function) is a mapping $r : \mathcal{X} \to \mathbb{R}^+$. Suppose r and s are two different AD algorithms that can detect anomalies $a_r = \{x : r(x) < \delta_r\}$ and $a_s = \{x : r(x) < \delta_s\}$, then the challenge is to *meaningfully* compare r and s in the context of AD. This meaningful comparison entails special considerations that will be discussed later on. It is important to recall that even though AD can be described mathematically, its motivation is practical, and usually there is interest in detecting only a particular subset of a (the subset relevant to the application in question); Foorthius et al. [13] provide a qualitative description of different types of anomalies of interest. As a consequence, whereas mathematically one can compute a sensible distance between f and r, and f and s to determine which one is closer to f, this is only truly meaningful if there is a high overlap between a_r and a_s. This reasoning is further demonstrated in Figure 1.

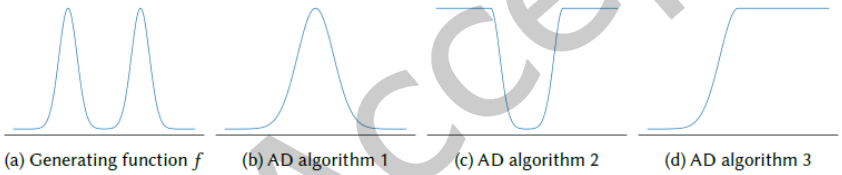


Fig. 1. Algorithm 1 correctly detects the outwards anomalies but fails to detect the inwards anomalies, algorithm 2 is the opposite. If the objective is to capture all anomalies, then one cannot prescind of algorithm 1 or 2 (even though one can find which one is closer to f). On the other hand, algorithms 1 and 3 detect some anomalies in common, hence they can be compared and one realizes algorithm 1 is better (also closer to f).

We have distilled the problem of comparing AD algorithms in the unsupervised setting into two tasks: 1) determining to what extent the algorithms are detecting the same anomalies, if they are, 2) choosing a criterion to determine which one is "better"; we use "better" to emphasize that there is no intrinsic better, instead one chooses a criterion that has suitable properties and that criterion is used to evaluate the different AD algorithms. There has been pivotal work on 2) such as the work by Goix et al. [9, 15, 16], who proposed criteria to evaluate the quality of unsupervised AD algorithms based on the MV (Mass-Volume) and EM (Excess-Mass) curves. In a similar fashion, Marques et al. [22, 23] proposed a different criterion that considers the degree of separability of each point with respect to the rest weighted by the anomaly score. We will discuss these approaches and others in more depth in the Related Work section.

Our work focuses on task 1), which consists of defining a notion of equivalence between r and p (two AD algorithms) that is meaningful in the context of AD. The fact we are in the context of AD is crucial because it requires the notion of proximity to have certain properties. If this were not the case, a viable notion of proximity would be the Kullback-Leibler (KL) divergence assuming that r and s are converted into probability distributions.

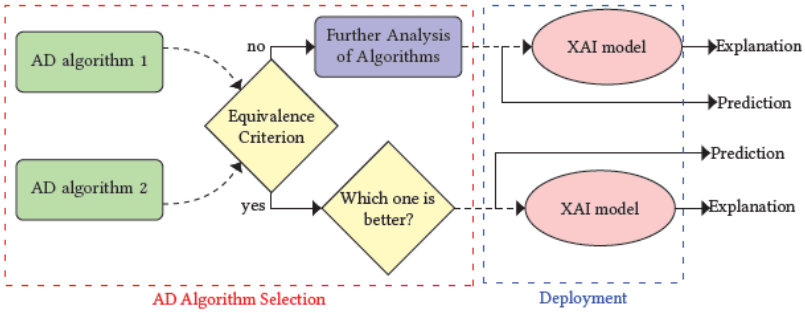


Fig. 2. Potential workflow for model selection and deployment of AD algorithm

Therefore, task 1) comprises of designing a notion of distance or equivalence that exhibits certain properties. One of them is that it is more important that two AD algorithms agree over sets of low-likelihood than sets of high-likelihood (which in the case of KL divergence is the opposite because sets of high-likelihood have higher mass, hence higher weight in the calculation).

Our work is complementary to that by Goix et al. and Marques et al. [15, 22] in the quest to evaluate unsupervised AD algorithms. More specifically, any two algorithms could pass first through our algorithm to determine whether they detect the same of kind of anomalies, if so, then [15] or [22] provide criteria that can be used to indicate which one is "better", if not, then further comparisons to determine which one is "better" are not meaningful; this procedure is illustrated in Figure 2, where we also included a deployment block that uses an additional model to produce of an explanation of that prediction. Moreover, this line of work as standalone is worthy of pursuit as it helps elucidate what each of the AD algorithms is doing, which is particularly useful since modern AD algorithms are often constructed as black boxes [26].

The rest of the paper is organized as follows: Section 2 presents a literature review, and also lists our specific contributions, in comparison with those contributions by other works; Section 3 provides the background and the technical motivation for our algorithm; Section 4 describes the algorithm in detail; Section 5 shows and discusses the properties our algorithm has; Section 6 shows empirical evidence of the results our equivalence criterion can achieve on simulated data and on a real-world dataset of the daily energy consumption registered by electrical meters; finally Section 7 provides the conclusions.

2 RELATED WORK

The literature on AD is vast, as there are countless AD algorithms [11]. In contrast, the literature on how to compare different AD algorithms is scant. Traditionally, there have been three approaches to compare anomaly detection algorithms: the first one is supervised learning on publicly available datasets that are labelled. Among these there is the work by Hasain et al. [17], where they deal with detecting anomalies in data that is being streamed. Another example is [28], where the application is cybersecurity and network intrusion in emerging technologies such IoT(Internet of Things) or the work in [10] that deals on a similar problem. In this approach, comparing AD algorithms is trivial because the receiving operating characteristic (ROC) curve can be calculated, as well as the area under the ROC curve (AUC), which also serves as comparison criterion. The drawback, though, is that it can only be utilized for applications that have publicly available datasets that have been labelled. Even in those cases it is necessary to assess whether the data in question has a similar distribution to that of the public

dataset, e.g., fraudsters are adaptive [7], then an outdated dataset becomes obsolete for training algorithms for fraud detection.

The second approach to comparing two AD algorithms is supervised learning on simulated data, where labels can be generated. Some of the works using this approach are: Anton et al. [2], where they deal with cybersecurity and intrusion detection; Flach et al. [12], where they want to detect anomalies on Earth observations; or Meire et al. [24], where they want to detect anomalies on acoustic sounds. Comparing algorithms is trivial for the same reason as in the first approach. The drawback with this approach is that it requires an accurate model to simulate non-anomalous and anomalous points, and for many real-world applications even state-of-the-art simulators still fail to close the gap between real-world data and simulated data [31].

Finally, the third approach is unsupervised learning, which is the case when there are no labels. This is the most useful case because it does not require any human labelling and it can be used on any application; the challenge is that it is not trivial to compare different AD algorithms. In fact, despite its usefulness this approach has been vastly understudied, as pointed out by Ma et al. [21]. More specifically, Ma et al. found only three techniques (that we also found independently) that address the problem of algorithm comparison and evaluation in the context of anomaly detection in the unsupervised setting. Those three approaches are [15, 22], which we mentioned previously, and [25].

The criterion provided by Goix et al. [15], namely the MV curve, is defined by the following equation $MV_s(\alpha) = \{\inf_{u \geq 0} \lambda(s \geq u) \,|\, \mathbb{P}(s(X) \geq u) \geq \alpha\}$, where λ is the Lebesgue measure, s is the AD algorithm, and $\alpha \in [0, 1]$. For a given α, $MV_s(\alpha)$ consists of finding the infimum over u of the set $\lambda(s \geq u)$ (which amounts to making u as big as possible), such that the probability of $s(X)$ being greater or equal than u is greater or equal than α, then the $MV_s(\alpha)$ corresponds to $\lambda(s \geq u)$. If $MV_r(\alpha) \leq MV_s(\alpha)$, then AD algorithm r is better than s. The EM curve has a similar construction. Intuitively, the idea is that the level sets have minimum volume, hence minimum MV curve, when the generating function, f, is used to generate the MV curve, therefore as an anomaly scoring function r gets closer to f, its MV curve will decrease (we recognize this discussion of that work is rather opaque, but an in-depth discussion is out of the scope. Interested readers may check [9]). We borrow an idea from this work indirectly, which is that you do not have to compare directly the generating function f with the scoring function r, instead you can compare their level sets or even their rankings.

Marques et al. [22] criterion to compare AD algorithms is named IREOS (Internal, Relative Evaluation of Outlier Solutions). It estimates how separable is each point from all the other points by using a maximum-margin classifier, which in their case it is a nonlinear SVM, then this is weighted by the anomaly score of the corresponding point. If the points with largest margins have the highest anomaly scores, then IREOS will be the highest possible. The higher the IREOS, the better the scoring function. Nguyen et al. [25] propose different indexes whose underlying idea is that normal data should form one big cluster whereas anomalies will form a smaller cluster. They focus on two concepts: compactness and separability. Therefore, according to them, a good AD algorithm should separate the data in compact clusters that are far from each other. One of their indexes, root-mean-square standard deviation (SDT) is given by the formula $RMSSTD = (\sum_{i=1}^{NC} \sum_{x \in C_i} ||x - c_i||^2)/(P \sum_{i=1}^{NC}(n_i - 1))$, where NC is the number of clusters, C_i is the ith cluster, c_i is the center of C_i, n_i is the number of objects in C_i, P is the number of attributes in the dataset, and x is the datapoint. The numerator is measuring how compact each cluster is by calculating $||x - c_i||$ and summing over clusters; the denominator is a normalization constant. Of those three approaches, Ma et al. [21] suggests that none of them are useful in practice, as they select models only comparable to a state-of-the-art detector with random hyperparameters. While we find that claim arguable, because the AD models were not tested as to whether they detect the same anomalies or not, it is clear that significant improvement is needed in this area.

A research area, relevant to anomaly detection, that has received significant attention in the last few years is Explainable Artificial Intelligence (XAI) [6]. XAI seeks to equip opaque models with explanations or interpretations that are understood and accepted by a human audience [6, 8]; this process is known as post-hoc explainability [6]. The definition given by Arrieta et al. [6] recites as follows: "post-hoc explainability techniques aim at communicating understandable information about how an already developed model produces its predictions for any given input". XAI is important for AD because many times flagging a data point as anomaly is not enough; an explanation of why the point was flagged is also needed.

An example of this is the work by Kim et al. [20]. Their work is on anomaly detection on maritime engines. For this application knowing that there is an anomaly is not sufficient, instead one needs to know what sensor is triggering the anomaly. In their work they use traditional anomaly detection algorithms and equip them with XAI models that generate an explanation. Another example is the work by Gnoss et al. [14], which is on detection of erroneous or fraudulent business transactions and corresponding journal entries; and in this case too, when an anomaly detected, an explanation is needed. Furthermore, there is work such as the one by Barbado et al. [4], whose application is AD for fuel consumption in fleet vehicles. Their work, not only generates an explanation why a point was labelled an anomaly, but it also provides a counterfactual recommendation, that is, it provides what could have been done differently about that vehicle to turn it into an inlier. All the works presented in this paragraph employ post-hoc explainability techniques. There are many more examples of AD applications that have incorporated XAI; a more comprehensive list can be found on the review by Yepmo et al. [30].

Another approach to having AD algorithms with explanations is to make the AD algorithms transparent, that is, explainable by themselves ; a typical example of a transparent algorithm is linear regression [6]. In this category, Alvarez-Melis et al. [1] proposes self-explaining neural networks. A self-explaining neural network is a neural network that behaves smoothly, as a result it can be approximated by a linear function at any point. Alvarez-Melis et al. enforce the smoothness by adding the term $\mathcal{L}_\theta = ||\nabla_x f(x) - \theta(x)^T J_x^h(x)||$ to the loss function, where f is the neural network function, x is the datapoint, h is a function that maps from input space to feature space, $J_x^h(x)$ is the Jacobian of h with respect to x, and $\theta(x)$ refers to the parameters of the linear approximation at point x. If $\mathcal{L}_\theta = 0$, then the model can be locally approximated by a linear function, thus making the model explainable. A different approach with the same underlying idea was taken by Barbado et al. [5]. Their approach consists of taking a trained one-class SVM and extracting rules from it. The rules are obtained by partitioning the input space in hypercubes that separate anomalous points from non-anomalous points, in other words, the rules are of the form: if x (the datapoint) is inside a hypercube that contains inliers, then x is an inlier. Then, these rules are used for prediction and also serve as explanations.

Above we presented some examples of AD algorithms equipped with XAI post-hoc techniques, and transparent AD algorithms. Our work and post-hoc techniques operate in a complementary fashion. To see this, we consider the definition of post-hoc technique presented above. From that definition, we can notice there are already two differences between our work and post-hoc techniques. First, our work does not say anything about *how* the models produce predictions, rather our work deals with *what* predictions one model produces in contrast to another model; second, our work does not require a human audience to interpret explanations. Note that the first and second differences come together because if there is no explanation, then there is no need for an explainee. Another advantage of not using explanations to analyze a model is that one does not have to compare different explanations for the same model to determine which is explanation is better. In fact, this is one weakness of XAI, namely, there is no consensus or a rigorous mathematical definition of what constitutes a good explanation [6, 8]. Although there has been progress, such as the work by Hoffman et al. [18] that proposes metrics to quantify the quality of explanations, Arrieta et al. still concludes that more quantifiable, general XAI metrics are needed to support the existing measurement procedures and tools proposed by the community [6].

To sum up, our work and XAI try to achieve the same general goal, namely, to obtain insights from black-box models; but do so in a different way. XAI does so by generating explanations that have to interpreted by a human, whereas our work directly answers the question whether two AD algorithms are equivalent without. XAI and our work can work jointly, as illustrated in Figure 2. In fact, if two algorithms are equivalent as given by our work, then those two algorithms should have the same explanations; the converse should also hold, that is, if two algorithms, for every data point, have the same explanation, then they should be equivalent.

In this work we propose, to best of the authors' knowledge, the first equivalence criterion for AD algorithms. More concretely, our work aims to determine to what degree two AD algorithms (scoring functions) are detecting the same kind of anomalies, which is different from designing a criterion to determine which AD algorithm is better. An equivalence measure is crucial because it only makes sense to determine "better" algorithms from a set of AD algorithms that detect the same kind of anomalies, otherwise they are just different and the search for "better" AD algorithms becomes superfluous.

3 BACKGROUND

3.1 Problem Setup

The goal of this work is to develop a criterion of equivalence between two anomaly scores that matches intuition. While it is not clear what this criterion should be, a set of desirable properties can be established and then be used as a heuristic on the design of it.

First, let us introduce some notation: $C(\boldsymbol{r}, \boldsymbol{s})$ is short hand notation for $C(r(X), s(X))$, where X is a dataset with n data points, and C is the equivalence criterion; in our notation r refers to the scoring function itself and $\boldsymbol{r}$ (in boldface) refers to the scoring function evaluated at a dataset, thus resulting in a set with n elements that we will call *anomaly score*. We are interested in measuring a distance between $\boldsymbol{r}$ and $\boldsymbol{s}$, that is the anomaly scores, rather than r and s, that is the scoring functions. Consequently, $C(\boldsymbol{r}, \boldsymbol{s})$ is only meaningful if the dataset contains enough points, some of which have to be anomalies.

In our case we consider an equivalence criterion that can be interpreted as a correlation. Specifically, it is defined as follows: let $C : \mathcal{A} \times \mathcal{A} \to [-1, 1]$, where $\mathcal{A}$ is the space of anomaly scores and C is the equivalence criterion. By definition if $C(\boldsymbol{r}, \boldsymbol{s}) = 1$, then $\boldsymbol{r}$ and $\boldsymbol{s}$ are equivalent; if $C(\boldsymbol{r}, \boldsymbol{s}) = 0$, then $\boldsymbol{r}$ and $\boldsymbol{s}$ are uncorrelated, i.e., $\boldsymbol{r}$ looks completely random with respect to $\boldsymbol{s}$; if $C(\boldsymbol{r}, \boldsymbol{s}) = -1$, then $\boldsymbol{r}$ and $\boldsymbol{s}$ are inversely correlated, e.g., $\boldsymbol{r}(x) = x$ and $\boldsymbol{s}(x) = -x$. We will refer to $C(\boldsymbol{r}, \boldsymbol{s})$ as the equivalence criterion value, which is the numerical value obtained by evaluating C at $\boldsymbol{r}$ and $\boldsymbol{s}$.

Some of the fundamental properties any equivalence criterion should have are: a) $C(\boldsymbol{r}, \boldsymbol{r}) = 1$; b) $C(\boldsymbol{r}, \boldsymbol{s}) = C(\boldsymbol{s}, \boldsymbol{r})$; c) $C(\boldsymbol{r}, -\boldsymbol{r}) = -1$. We will discuss more properties in section 3.4.

3.2 Simple Equivalence Criterion (SEC)

Note that if we let $g = q - f$ for some constant q to guarantee g is nonnegative, then the set of anomalies is identical to before, but it is now given by $a = \{x : g(x) > \delta\}$. From now on, we assume the higher the score, the more anomalous the point is, and we consider sets of the form $\{x : g(x) > \delta\}$, also known as super-level sets. From this definition, a natural starting point for an equivalence criterion between $\boldsymbol{r}$ and $\boldsymbol{s}$ is:

$$\hat{\sigma}(\boldsymbol{r}, \boldsymbol{s}) = \sum_i \sum_j |r_i \cap s_j| \qquad r_i = \{x : r(x) \geq \delta_i\} \tag{1}$$

where $\delta_i > \delta_j$ if $i > j$ and $r_i \neq r_j$ for $i \neq j$, and $|\cdot|$ represents the order of set, which is the number of elements in the set.

Intuitively, $\hat{\sigma}$ compares how similar are all super-level sets of $\boldsymbol{r}$ with all super-level sets of $\boldsymbol{s}$, which really is a comparison between anomalies detected by $\boldsymbol{r}$ and $\boldsymbol{s}$. Comparing all super-level sets, as opposed to some, is advantageous because it produces a robust equivalence criterion because even if $\boldsymbol{r}$ and $\boldsymbol{s}$ differ slightly, there will be super-level sets that account for that difference. If $\boldsymbol{r} = \boldsymbol{s}$, then $\hat{\sigma}(\boldsymbol{r}, \boldsymbol{s})$ will be maximum; in contrast, if $\boldsymbol{r}$=$-\boldsymbol{s}$, then $\hat{\sigma}(\boldsymbol{r}, \boldsymbol{s})$ will be minimum, although not zero.

If a data point has rank k in $\boldsymbol{r}$ and l in $\boldsymbol{s}$, then the element will be contribute $k \cdot l$ times to the sum, which are $\delta_1^{(r)}, \delta_2^{(r)}, ..., \delta_k^{(r)}$ crossed with $\delta_1^{(s)}, \delta_2^{(s)}, ..., \delta_l^{(s)}$. Moreover, since we want a correlation-like equivalence criterion we need to include a normalization factor to set the maximum equal to 1, which all together yields equation (2).

$$\sigma_n(\boldsymbol{r}, \boldsymbol{s}) = \frac{\sum_i^n \hat{r}_i \hat{s}_i}{\sum_{i=1}^n i^2} \tag{2}$$

where $\hat{\boldsymbol{r}}$ is the ranking of data points given by $\boldsymbol{r}$. Recall that the data point ranked one has the lowest score, hence it is the least anomalous (according to $\boldsymbol{r}$); conversely the data point ranked last has the highest score and, hence it is the most anomalous (again, according to $\boldsymbol{r}$). If we consider σ_n as a random variable with the rankings, $\hat{\boldsymbol{r}}$ and $\hat{\boldsymbol{s}}$, being uniformly distributed across all permutations, then $\mathbb{E}[\sigma_n] = 3/4$ (The proof is in the supplementary material A.1).

Then, to make σ_n have range $[-1, 1]$, since its range is $[0.5, 1]$ we simply apply $\sigma = 4\sigma_n - 3$. We will analyze this equivalence criterion further and compare it to our proposed measure in the Theoretical Analysis section. We also present an example on the calculation of SEC in the supplementary material part D.

3.3 Toolbox for Proposed Criterion

Let $\boldsymbol{r}$ and $\boldsymbol{s}$ be the anomaly scores of two AD algorithms on a dataset $\mathcal{D}$. Let $\boldsymbol{m} = \mathcal{R}(\boldsymbol{r})$, $\boldsymbol{n} = \mathcal{R}(\boldsymbol{s})$ be the rankings generated from $\boldsymbol{r}$ and $\boldsymbol{s}$. Then, $\boldsymbol{n}$ can be seen as a permutation of $\boldsymbol{m}$ (or vice versa). One can calculate a distance between $\boldsymbol{m}$ and $\boldsymbol{n}$; in our case we use as distance the minimum number of moves that it would take to convert $\boldsymbol{n}$ into $\boldsymbol{m}$ divided by a normalization factor, which is known as the Kendall Tau correlation coefficient [19] and it is calculated as follows:

$$\tau(\boldsymbol{m}, \boldsymbol{n}) = \frac{\langle \boldsymbol{m}, \boldsymbol{n} \rangle}{||\boldsymbol{m}|| \cdot ||\boldsymbol{n}||} \tag{3}$$

where $\langle \boldsymbol{m}, \boldsymbol{n} \rangle = \sum_{j<i} \text{sgn}(m_i - m_j)\text{sgn}(n_i - n_j)$, and $||\boldsymbol{m}|| = \sqrt{\langle \boldsymbol{m}, \boldsymbol{m} \rangle}$

A different perspective to Kendall Tau coefficient is:

$$\tau(\boldsymbol{m}, \boldsymbol{n}) = \frac{(\text{number of concordant pairs}) - (\text{number of discordant pairs})}{\binom{l}{2}}$$

where concordant pairs refers to set of pairs about which $\boldsymbol{m}$ and $\boldsymbol{n}$ agree as to which element of the pair has a higher ranking (is greater), and discordant pairs is the set of pairs with disagreement between $\boldsymbol{m}$ and $\boldsymbol{n}$. An example of the calculation of the Kendall Tau correlation coefficient is included in the appendix section E.

In subsequent work, Vigna et al. [29] showed that adding weights to the pairs still preserves many of properties of the original Kendall Tau coefficient (e.g., $\langle \boldsymbol{m}, \boldsymbol{n} \rangle$ forms an inner product, and hence one gets Cauchy-Schwarzt-like

inequalities such as $|\langle \boldsymbol{m}, \boldsymbol{n} \rangle| \leq ||\boldsymbol{m}||||\boldsymbol{n}||$), while still adding new ones. Their formulation of the weighted Kendall Tau coefficient uses the following calculation:

$$\langle \boldsymbol{m}, \boldsymbol{n} \rangle_\omega = \sum_{j<i} \text{sgn}(m_i - m_j)\text{sgn}(n_i - n_j)\omega(i,j) \tag{4}$$

The weighted Kendall Tau coefficient becomes $\tau(\boldsymbol{m}, \boldsymbol{n})_\omega = \dfrac{\langle \boldsymbol{m}, \boldsymbol{n} \rangle_\omega}{||\boldsymbol{m}||_\omega \cdot ||\boldsymbol{n}||_\omega}$

This weighted formulation gives leeway to enforce additional desirable properties for our proposed equivalence criterion.

3.4 Desirable Properties of an Equivalence Criterion

These properties are considered as an addition to the properties mentioned in section 3.1. With the properties mentioned here and section 3.1 we formalize our intuition with respect to the equivalence criterion between anomaly scores. They, however, are by no means exhaustive and only reflect what we consider is important when formulating an equivalence criterion.

The properties are described in terms of the fact that one anomaly score be seen as the permutation of another anomaly score. More specifically, a permutation move refers to the number of moves that it would take to move data point j from the ranking s_j so that it has the same ranking given by r_j, e.g., if $\boldsymbol{s}$ ranks a point uth, and $\boldsymbol{r}$ ranks the same point vth, then the permutation move for that point would be $|u - v|$ (the sign is not relevant, as one can always fix $\boldsymbol{s}$ and see $\boldsymbol{r}$ as a permutation or vice versa to get positive permutation moves). We will consider two types of permutations moves: large permutation moves which refers to when the difference in rankings of $\boldsymbol{s}$ and $\boldsymbol{r}$ in absolute value is large (this corresponds to element a in Figure 3); and small or local permutation moves which refers to when the difference in rankings of $\boldsymbol{s}$ and $\boldsymbol{r}$ in absolute value is small (this corresponds to element b in Figure 3). Then, the question is how to assign different weights to different permutations moves; this is going be addressed by our properties, which are:

(1) A k-move permutation move should reduce the equivalence criterion value more than k 1-move permutation moves.

(2) Locally moving points that have high anomaly scores should reduce the equivalence criterion value more than locally moving points that have low anomaly scores.

The rationale behind these two properties is as follows: with respect to property one, firstly, we have an immediate corollary, which is l k-move permutations should reduce the equivalence criterion value more than k l-move permutation as long as $l < k$. We will refer to l-move permutations as small disagreements, and k-move permutations as major disagreements. Intuitively, property 1 is desirable because small disagreements indicate the anomaly scores differ locally, hence those local differences should not reduce the equivalence criterion value significantly; on the contrary, major disagreements are signs that the anomaly scores are fundamentally different, hence they should reduce the equivalence criterion value more dramatically. An example of what property 1 implies is: if $\boldsymbol{r}$ is an anomaly score, and $\boldsymbol{s} = \{r_i + \epsilon_i\}_{i=1}^n$ for small ϵ_i, then $C(\boldsymbol{r}, \boldsymbol{s})$ should be close to 1. With respect to the second property: high density regions on the anomaly score are more susceptible to having discordant pairs by pure randomness as points are close together; these high density regions correspond precisely to the data points that have low anomaly scores because that is where all non-anomalous data points reside (recall the majority of the points are not anomalous). Therefore, discordant pairs in these high density regions should have

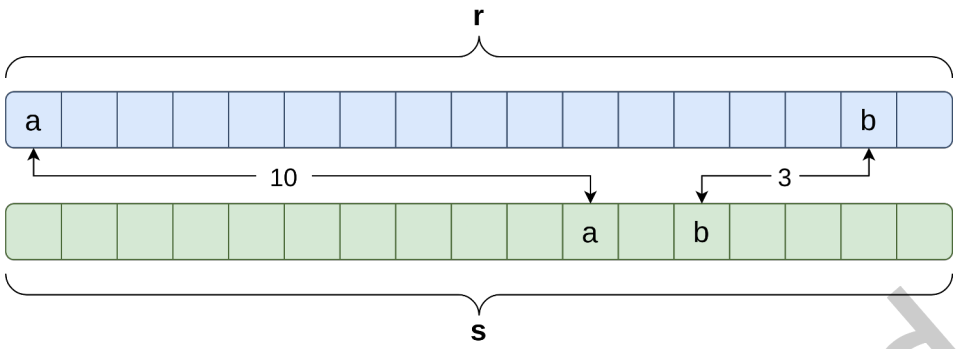


Fig. 3. 10-move permutation and 3-move permutation between anomaly scores $\boldsymbol{r}$ and $\boldsymbol{s}$. Left to right is increasing score. $\boldsymbol{r}$ is the reference. It takes a 10-move permutation for element a in $\boldsymbol{s}$, so that it is ranked the same as in r; for b, it only takes a 3-move permutation.

lower weights, which is equivalent to saying that non-high density regions should have larger weights; in other words, regions where the anomalous points reside should have higher weights.

In the Theoretical Analysis section we mathematically define these two properties and show that our proposed equivalence criterion satisfies both.

4 GAUSSIAN EQUIVALENCE CRITERION (GEC)

The analysis so far uses rankings, instead of anomaly scores, because the work done in [19, 29] was formulated in terms of rankings. However, equation (3) and its weighted version can be calculated with anomaly scores because the ranking map $\mathcal{R} : \mathbb{R} \to \{1, 2, ..., n\} \in \mathbb{N}$ is order preserving. Moreover, we have that converting $\boldsymbol{s}$ into $\boldsymbol{r}$ takes a permutation; and a re-scaling map g, that matches the values of $\boldsymbol{r}$ with those values of the permuted version of $\boldsymbol{s}$. While our equivalence criterion does not concern itself with g, it explicitly uses the anomaly scores, $\boldsymbol{r}$ and $\boldsymbol{s}$, in the weight calculation.

4.1 Algorithm for GEC

Sort and then divide $\boldsymbol{r}$ and $\boldsymbol{s}$ into k subsets that correspond to different classes within the anomaly scores, e.g., normal (non-anomalous) set can be from 0th to 90th percentile; 90th to 98th percentile can be grey area set; 98th to 100th percentile can be anomalous set, which would give $k = 3$. Then, from the points corresponding to each subset calculate the mean $\mu = \{\mu_1, \mu_2, ..., \mu_k\}$ and the standard deviation $\sigma = \{\sigma_1, \sigma_2, ..., \sigma_k\}$.

Then, GEC of $\boldsymbol{r}$ and $\boldsymbol{s}$, $\phi(\boldsymbol{r}, \boldsymbol{s})$, is calculated using the following equations:

$$\phi(\boldsymbol{r}, \boldsymbol{s}) = \frac{\langle \boldsymbol{r}, \boldsymbol{s} \rangle_\omega}{||\boldsymbol{r}||_\omega \cdot ||\boldsymbol{s}||_\omega} \tag{5}$$

$$\langle \boldsymbol{r}, \boldsymbol{s} \rangle_\omega = \sum_{j<i} \text{sgn}(r_i - r_j)\text{sgn}(s_i - s_j)\omega_r(i,j)\omega_s(i,j) \qquad ||\boldsymbol{r}||_\omega = \sqrt{\langle \boldsymbol{r}, \boldsymbol{r} \rangle_\omega} \tag{6}$$

$$\omega_r(i,j) = \sum_{d=1}^{k} F_{\mu_d, \sigma_d}(r_i, r_j) \tag{7}$$

$$h_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad F_{\mu,\sigma}(x_1, x_2) = \int_{x_1}^{x_2} h_{\mu,\sigma}(x)dx \tag{8}$$

We propose that two anomaly scores are equivalent if their GEC is greater than 0.8; this, however, is an empirical observation whose analysis is discussed in the Results section. Depending on the application, one might use a threshold different than 0.8. GEC is clearly symmetrical, i.e. $\phi(\boldsymbol{r}, \boldsymbol{s}) = \phi(\boldsymbol{s}, \boldsymbol{r})$, and exhibits a resemblance of the transitive property, that is, if $\boldsymbol{r}$ is equivalent to $\boldsymbol{s}$ and $\boldsymbol{s}$ is equivalent to $\boldsymbol{p}$, then $\boldsymbol{r}$ is equivalent to $\boldsymbol{p}$. Therefore, if there are n different anomaly scores, there is no need to run the equivalence algorithm $O(n^2)$ times; instead, $O(n \log n)$ times suffices to determine the equivalence relations among all of them. The pseudocode for the GEC is provided in the supplementary material section E.

4.2 Intuition about GEC

In this section we will discuss the intuition and motivation of each of the steps of GEC.

Anomaly scoring functions associate levels of anomaly with datapoints, i.e., the higher the score, the more anomalous. Dividing anomaly scores in subsets the way we propose allows to have a distinction between subsets of points that are highly anomalous and subsets of points that are not, which is beneficial as different subsets can then be treated differently. This distinction, while helpful, creates an issue, namely, some points might be arbitrarily close, but they might be assigned to different subsets. The way weights are defined resolves this issue as points that are arbitrarily close will have the same F_μ, σ value and hence the same weight, i.e., $F_{\mu,\sigma}(x_1, x_2) \approx F_{\mu,\sigma}(x_1, x_2 + \epsilon)$ for some small ϵ; and more generally, $\omega_r(i, j)$ will be approximately equal to $\omega_r(i, j+k)$ for a small k, that is, the weight function is smooth. In addition, this weighting scheme has the property that all weights will be bounded by k.

To sum up, the partition of the anomaly scores into subsets introduces the notion of classes (i.e., anomalous, normal, grey area, etc), and the weighting scheme gives a sensible notion of distance between anomaly score points that is smooth and bounded, thus resulting in an equivalence criterion that allows flexibility while being also robust.

5 THEORETICAL ANALYSIS

The proofs of the mathematical statements presented here can be found in the Supplementary Material section at the end of the document.

5.1 Properties of GEC

GEC inherits all the properties that applied to the Kendall Tau Correlation coefficient. Here we discuss additional properties, in particular property 1 and 2 defined in section 3.4.

5.1.1 Property 1. Note that 1 k-move permutation will yield the same number of discordances as k 1-move permutations, namely k. Additionally, the weights for the 1 k-move permutation case are going to be of the form $\omega(i, i + j)$ for $j \in [1, ..., k]$, whereas the weights for the k 1-move permutation case $\omega(i, i + 1)$ for $i \in [j, ..., j + k - 1]$. Due to the weighting scheme we have that $\omega(i, i+j) > \omega(i, i+1)$ if $j > 1$, therefore the discordances in the 1 k-move permutation case, although the same in number, will have a higher weight than the k 1-move permutations case. As discordances subtract from the GEC value, a higher quantity is subtracted in the 1 k-move permutation case than the k 1-move permutations, thus showing property 1.

Theorem 5.1 provides a general and formal result that reflects the reasoning presented above; it also provides a relation between GEC and the Kendall Tau correlation coefficient. Its proof along with other necessary Lemmas and their proof are provided in the Supplementary Material section A.1.

Theorem 5.1. *Let $\boldsymbol{r}$ be a sorted set of n anomaly detection scores, let $\boldsymbol{s}$ be a permutation of $\boldsymbol{r}$ that only allows permutation moves with at most $\omega < \frac{2}{n(n-1)} \sum \omega_r(i,j)\omega_s(i,j) = \mu$ weight, and more concordant pairs, c, than discordant pairs, d, with respect to r, then $0 < \tau(\boldsymbol{r}, \boldsymbol{s}) < \phi(\boldsymbol{r}, \boldsymbol{s})$.*

Corollary 5.1.1. *Let $\boldsymbol{r}$ be a sorted set of n anomaly detection scores, let $\boldsymbol{s}$ be a permutation of $\boldsymbol{r}$ that only allows permutation moves with at most $\omega > \frac{2}{n(n-1)} \sum \omega_r(i,j)\omega_s(i,j) = \mu$ weight, and more concordant pairs, c, than discordant pairs, d, with respect to r, then $0 < \phi(\boldsymbol{r}, \boldsymbol{s}) < \tau(\boldsymbol{r}, \boldsymbol{s})$.*

Kendal Tau coefficient weighs all permutation moves equally (a k-move is equal to k 1-moves). Therefore, the fact that $\tau(\boldsymbol{r}, \boldsymbol{s}) < \phi(\boldsymbol{r}, \boldsymbol{s})$ for local permutations shows that ϕ assigns less priority to local permutations. Conversely, corollary 5.1.1 shows that ϕ assigns more priority to global permutations.

5.1.2 Property 2. To intuitively see why property 2 holds for GEC we will consider an extreme case. Suppose there are $n + 2$ points and only two regions are defined: region A between the two most anomalous points, and region B among the remaining n points. Furthermore, suppose region A and B are far enough apart that region A has no contribution over region B and vice versa. In this setting the highest weight is going to be between the end points of a region. For region A, that corresponds to the only two points in that region, whereas for region B, that corresponds to point 1 and n; this suggests that any 1-move permutation in region B will have a much smaller weight than the only 1-move permutation in region A, hence showing property 2.

Theorem 5.2 provides a general and formal result that reflects the reasoning presented above. Its proof is provided Supplementary Material section A.2

Theorem 5.2. *Let $\boldsymbol{r}$ be an anomaly score, let R_m and R_n be two different regions such that $R_m = \{1, 2, ..., u\}, R_n = \{1, 2, ..., v\}$ with $u \gg v$ (meaning that R_m has many more data points than R_n), and u and v are large enough for the law of large numbers to be applicable. Consider all k-move permutations within a class with $v \gg k$, then $\frac{1}{u-k} \sum_{i \in R_n} \omega_r(i, i+k) < \frac{1}{v-k} \sum_{i \in R_m} \omega_r(i, i+k)$, namely, the average weight for k-move permutations is higher in region R_n.*

In the context of anomaly detection, the non-anomalous (normal) region will have many more data points than the anomalous region (in our case we set it to 98% to 2%). By Theorem 5.2, we have that permutations of non-anomalous data points will have less weight than permutations of anomalous data points, therefore showing that GEC satisfies property 2.

An additional property to GEC is that it is invariant to linear scaling of the anomaly scores, mathematically that is $C(\boldsymbol{r}, \boldsymbol{s}) = C(\alpha \boldsymbol{r}, \boldsymbol{s})$ for $\alpha > 0$. This is formally stated in Lemma 5.2.1 and the proof is presented in the supplementary material A.2 along with the proof of Theorem 5.2.

Lemma 5.2.1. *Let $S = \{s_1, s_2, ..., s_n\}$ be 0 mean and have standard deviation σ, and $s_i < s_j$ for $i < j$, additionally let $P = \alpha S = \{p_1, p_2, ..., p_n\}$ with α positive, then the standard deviation of P is $\sigma_p = \alpha\sigma$ and $F_\sigma(s_i, s_j) = F_{\sigma_p}(p_i, p_j)$*

The constraint for the scores to be 0 mean makes the proof easier, but it is not restrictive because the mean can be added later and the weights remain unchanged. This fact in combination with Lemma 5.2.1 show that the mean of the anomaly score does not matter, instead what matters is how the points are spread. This property is useful because it implies there is no need for any preprocessing of the anomaly scores, they can be used as they come.

5.2 Properties of SEC

To study the properties of SEC we will consider the following: first, if two anomaly scores $\boldsymbol{r}$ and $\boldsymbol{s}$ have the same rankings, then $\sigma(\boldsymbol{r}, \boldsymbol{s}) = 1$; if there is a permutation in $\boldsymbol{s}$, then $\sigma(\boldsymbol{r}, \boldsymbol{s})$ can be calculated again by subtracting what was discounted by that permutation. Second, in equation (2), the only terms that depends on the rankings are the terms in the numerator; everything else is normalization constants, so, we will focus on the numerator terms.

5.2.1 Property 1. Let $\boldsymbol{r}$ and $\boldsymbol{s}$ be the same ranking, except in $\boldsymbol{s}$ element n and $n+k$ are swapped, then for the new index we have that $nn + (n+k)(n+k) - (n(n+k) + (n+k)n) = 2n^2 + 2nk + k^2 - (2n^2 + 2kn) = k^2$. This means that 1 k-swap reduces the score by k^2 (normalization factors need to be included), which is higher than k 1-swaps, which would reduce the score only by k, thus showing property 1.

5.2.2 Property 2. Let $\boldsymbol{r}$ be a ranking, then let $\boldsymbol{p} = M\boldsymbol{r}$ where M is a permutation matrix that only permutes k consecutive elements arbitrarily, moreover, it could be any k consecutive elements, i.e., the first k, or the last k, etc.

Lemma 5.2.2. $\sum_{i=n}^{n+k} i^2 - \sum_{i=n}^{n+k} i \cdot p_i$ *does not depend on* n

Lemma 5.2.2 shows that property 2 does not hold for SEC by showing that it is irrelevant whether the permutation is made within low anomaly scores or high anomaly scores since they both reduce the score equally. The proof of lemma 5.2.2 is in the supplementary material

5.3 Discussion

Table 1 contains a summary and a comparison for the properties of GEC and SEC. These properties are considered in addition to the fundamental properties mentioned in section 3.1.

Table 1. Properties summary for proposed algorithm and naive approach

Property	GEC		SEC	
	Check	Discussion	Check	Discussion
Priority to high anomalous points	✓	Property 1	✓	Property 1
Swaps of high score points matter more	✓	Property 2	✗	It does not have property 2
Invariant to linear scaling	✓	Shown by virtue of Lemma 5.2.1	✓	Rankings are invariant to linear scaling
Weights are customizable	✓	Weights depend on the classes	✗	Weights are predefined by the rankings

More specifically, the property of weights being customizable for GEC comes from the fact that the number of classes and their "borders" are defined arbitrarily, which allows to assign importance to anomalies in the calculation of the equivalence criterion. This option to assign importance to anomalies is crucial because different applications assign different importance to anomalies; for example in medical applications detecting anomalies is

generally more important than in commercial applications. SEC does not provide that option because the weights are fixed by the rankings.

The comparison for the properties presented in Table 1 will be further discussed in the Results section, where we present examples that allow quantitative analysis.

6 RESULTS

The results section is divided into two parts: the first part uses simulated data and serves as benchmark for comparing GEC with SEC, as well as other future algorithms dealing with the same problem; in this part we also showcase the flexibility of GEC as regions can be defined to assign different levels of importance to high anomalous point. In this part we also validate how GEC matches intuition using a dataset with a Gaussian distribution. The second part uses real data from residential electric meters, where we test GEC in a real-world application. In this part, not only GEC provides insight about the AD algorithms, but about the dataset itself.

6.1 Simulated Data

6.1.1 Comparison of GEC and SEC. We directly simulate the anomaly scores, that is $\boldsymbol{r}$ and $\boldsymbol{s}$, using the following procedure: sample points from a uniform distribution between 0 and 1, and then take the seventh power of those sampled values; the resulting values are the anomaly scores (Note that we do not claim the data points are from the uniform distribution, and the scoring function is $r(x) = x^7$; instead we directly provide $\boldsymbol{r}$). The reason to elevate to a high power is to get a distribution over the anomaly scores that resembles that of real data, namely, many points with low scores (non-anomalous points) and a few with high scores (anomalous). 7 was chosen arbitrarily. Following this procedure generates the first anomaly score $\boldsymbol{r}$; to generate the second anomaly score, we take $\boldsymbol{r}$ and randomly permute elements between two percentiles, specifically we say that $\boldsymbol{s}_{(x,y)} = \boldsymbol{r}((x, y))$ which means $\boldsymbol{s}_{(x,y)}$ is identical to $\boldsymbol{r}$ except between percentile x and y where the elements are randomly shuffled. By definition $\boldsymbol{s} = \{\boldsymbol{s}_{(x,y)}\}_{x<y}$ for $x \in \{0, 1, ..., 99\}$ and $y \in \{1, 2, ..., 100\}$. The result obtained by following that process is illustrated in Figure 4, where different cases are illustrated.

Note that in principle, it is always possible to plot $\boldsymbol{r}$ with respect to $\boldsymbol{s}$ since they are both parameterized by the same dataset; if the relationship is monotone, then they are equivalent; however the relationship is usually more like the data points inside the square, where it is not obvious to what extent the relationship is monotone or not. In that case, an equivalence criterion is a more precise and consistent way to determine to what degree the AD algorithms are equivalent.

In our simulated data $\boldsymbol{r}$ is fixed and $\boldsymbol{s}$ varies. We consider $C(\boldsymbol{r}, \boldsymbol{s})$, where C is an equivalence criterion. The results for different equivalence criterion are shown in Figure 5. The upper left corner is the equivalence criterion $C(\boldsymbol{r}, \boldsymbol{s}_{(0,1)})$; the lower right corner is the criterion $C(\boldsymbol{r}, \boldsymbol{s}_{(99,100)})$; and the upper right corner $C(\boldsymbol{r}, \boldsymbol{s}_{(0,100)})$, which is the largest permutation there can be, where $\boldsymbol{s}_{(0,100)}$ is random with respect to $\boldsymbol{r}$. This explains why in the three cases in Figure 5 that upper right corner has the lowest equivalence criterion value.

SEC is invariant along the direction of the main diagonal, namely, points of the form $(i + k, j + k)$ for a fixed i and j, and varying k, which is due to the absence of property 2. We can observe that this is not the case for GEC in both cases. Moreover, GEC significantly decreases when high score points are randomly permuted, while it is robust to random permutations done over low score points.

In Figure 5 we can observe the difference between setting different regions: the middle figure is more sensitive than the right one to disagreements over high anomaly scores, which is reflected by the fact that the blue fringe

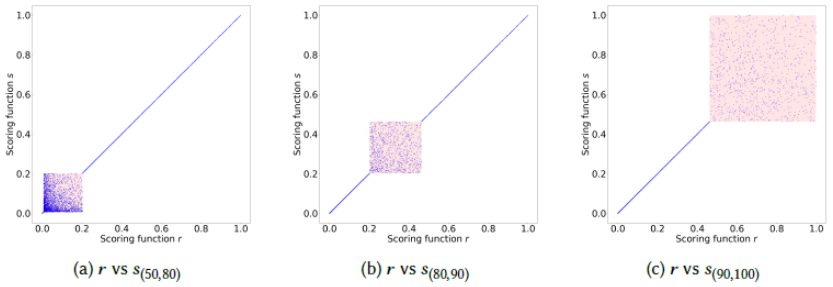


Fig. 4. Anomaly score r vs s. The percentiles indicate the points within which random permutations where made; this is further illustrated by the squares. This figure also hints about the distribution of scores.

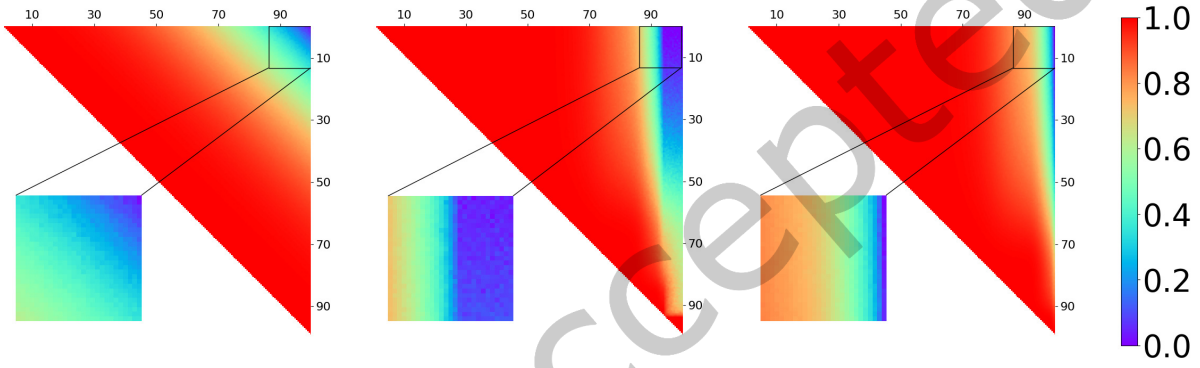


Fig. 5. Equivalence criterion for different algorithms Left: SEC; middle: GEC with classes at 80th, 90th, 94th, and 98th percentile; right: GEC with classes at 90th and 98th percentile.

(low GEC) is much wider. This happens because the middle figure contains four regions defined from the 80th to the 100th percentile, whereas the right figure only has two regions from 90th to 100th percentile. In general, the more regions that are defined over a certain interval of the anomaly score, the more weight points over that region have, therefore, for applications where anomalies are critical one may want to define multiple regions over highly anomalous points.

To sum up, this empirical analysis, while not exhaustive, illustrates the flexibility GEC provides through the definition of different classes as well as its behavior with respect to different permutations. As aforementioned, this flexibility is desirable because different applications will assign different levels of importance to anomalies. Additionally, we can observe that GEC prioritizes anomalous points over non-anomalous points because local permutations of high anomalous points, shown in the lower right corner of the subfigures of Figure 5 (coordinates (90,90)), reduce the GEC value significantly, whereas local permutations of non-high anomalous points, shown in the upper left corner of Figure 5, barely affect the GEC value. In contrast, SEC does not provide flexibility and treats local permutations equally regardless of where they occur.

6.1.2 Further Experiments with SEC. Let the data, $X \in \mathbb{R}^3$, be drawn from $\mathcal{N}(0, I)$, that is, a Gaussian distribution with 0 mean and covariance the identity matrix of size 3. The bigger the L_2 distance from the mean to a sample,

Table 2. GEC for Gaussian data

Alg 1 \ Alg 2	$\|\|x\|\|_2$	$\|\|x\|\|_1$	$\|\|x\|\|_3$	$\|\|(u,v)\|\|_1$	$\|\|(v,w)\|\|_3$	$\|\|u\|\|_1$	$\|\|w\|\|_1$	AUC
$\|\|x\|\|_2$		0.9443	0.9875	0.7687	0.7858	0.4383	0.4330	1
$\|\|x\|\|_1$			0.8890	0.7850	0.7385	0.4308	0.4458	0.9841
$\|\|x\|\|_3$				0.7365	0.7861	0.4303	0.4165	0.9947
$\|\|(u,v)\|\|_1$					0.4401	0.5008	0.1568	0.8610
$\|\|(v,w)\|\|_3$						0.1001	0.4944	0.8638
$\|\|u\|\|_1$							0.1099	0.5490
$\|\|w\|\|_1$								0.4572

the less likely that sample is, therefore, anomalous points will have large L_2 distance. Hence, the optimal AD algorithm calculates the L_2 distance from the mean, in this case the origin, to the sample. We will label as anomalies 5% of the samples with largest distance. Then, the dataset is given by $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$, where $x_i \in X$ and $y_i \in \{0, 1\}$. Moreover, we say that u_i, v_i, and w_i are the first, second, and third component of x_i, respectively.

Table 2 illustrates the GEC values after comparing different AD algorithms using $\mathcal{D}$, as previously described, and with regions defined in the 0 to 85th, 85th to 95th, and 95th to 100th percentiles. The AUC is also included. $||\cdot||_p$ denotes the L_p norm, and $||(u,v)||_p$ denotes the L_p norm only considering components u and v of x. Notably, since the L_2 norm was used to generate labels, the AUC of $||x||_2$ is 1. Moreover, $||x||_2, ||x||_1, ||x||_3$ all have a high AUC, and their pairwise GEC value is greater than 0.85 for all combinations. In contrast, if we consider $||x||_1$ with AUC of 0.9841 and $||u||_1$ with AUC of 0.549 (barely better than random guessing), we see they exhibit a low GEV value of 0.4383. In general, if algorithm A has a high AUC, then GEC(A, B) will only be high if algorithm B has also a high AUC. The opposite does not hold, namely, if algorithm A has a low AUC, then GEC(A, B) may or may not be high depending on B. An example of this is $||u||_1$ and $||w||_1$ as $||u||_1$ has low AUC, similar to $||w||_1$, but their GEC value is low.

We can observe a resemblance of the transitive property, in that GEC($||x||_2, ||x||_1$) = 0.9443 and GEC($||x||_2, ||x||_3$) = 0.9875 are high, which implies GEC($||x||_3, ||x||_1$) should be high, as it is the case. Similarly, GEC($||x||_2, ||u||_1$) = 0.4338 and GEC($||x||_2, ||w||_1$) = 0.4330 are both low, hence GEC($||u||_1, ||w||_1$) should be low, as it is the case. Note, however, that GEC($||x||_3, ||(u,v)||_1$) = 0.7365 and GEC($||x||_1, ||(u,v)||_1$) = 0.7850 are not equivalent by the 0.8 GEC value criterion, but GEC($||x||_1, ||x||_3$) is, with a GEC value of 0.8890. This is the reason why say a resemblance of the transitive property and 0.8 only serves as a guideline value.

To end this subsection, we will discuss why we propose 0.8 as the threshold for determining whether two anomaly scores are equivalent or not. The GEC values between $||x||_1, ||x||_2, ||x||_3$ are all above 0.8, with $GEC(||x||_1, ||x||_3$= 0.8890 being the lowest. This is to be expected, as the only difference is the L_p distance used for each of them. On the other hand, if we consider $GEC(||x||_1, ||(u,v)||_1)$ = 0.7850, we see their GEC value is below the proposed

0.8 threshold. This time, while for both cases the L_1 distance is used, one case, x, uses all components (features), and the other cases uses only the u and v components (features). Intuitively, one would expect that using different features should lead to AD algorithms that are not equivalent, which is what leads us to propose the 0.8 threshold. Moreover, if we consider pairs of AD algorithms that do not use the same features, we see the highest is $GEC(||x||_3, ||(u, w)||_3) = 0.7861$, which is still below the proposed threshold. We, nonetheless, emphasize that this is just an empirical observation. For example, if L_∞ were used, then it is possible that $GEC(||x||_\infty, ||(u, v)||_\infty) = 1$ because L_∞ outputs the highest feature while dismissing the rest.

6.2 Real-world Data

The dataset we use was provided by WE Energies and consists of the daily energy consumption at user level recorded by electrical meters. It contains over 400,000 meters with data recorded during 2018 and 2019. In this application the goal is to find faulty meters by detecting two kinds of anomalies: general anomalies, which entails any unusual consumption patterns; and slowing down meters, which entails meters whose consumption is gradually decreasing without apparent reason. For this purpose three different AD algorithms were developed: General Anomaly Detection (GAD), which uses a neural network as backbone and establishes whether a point is an anomaly or not based on the prediction error. We will denote by GAD $\mathcal{NN}_k$ the GAD AD algorithm whose neural network has k hidden layers with a predefined architecture. Secondly, Anomaly Detection by Fourier Transform (ADFT), which detects anomalies by using frequency decomposition, and then ranks data points based on how large the high frequency component is. Finally, Slowing-down Meters Detection (SD), which computes the difference between initial and final values over a time interval divided by the total variation of the signal along that interval. The algorithms are described in detail in the supplementary material section C.

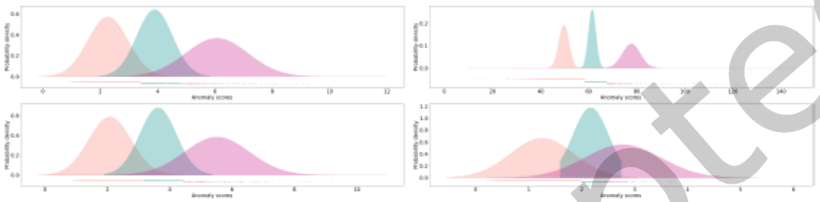
Table 3 shows the GECs for the experiments carried out. In addition, Figure 6 illustrates the anomaly score points for the different AD algorithms with the corresponding regions, which are "normal" from 0th to 90th percentile, "grey area" from 90th to 98th pecentile, and "abnormal" from 98th to 100th percentile (normal, grey area and abnormal are just names that we are assigning to the region). Figure 7(a) illustrates all the terms of the form $\text{sgn}(r_i - r_j)\omega_r(i, j)$ for the AD algorithms GAD $\mathcal{NN}_3$ and GAD $\mathcal{NN}_5$. The subfigure on the left of Figure 7(a) does not have discordances (hence all points are colored orange) because the points are sorted; in contrast, the right subfigure will have discordances (blue points) unless the two AD algorithms agree over all points. Recall that GEC is calculated by multiplying the right and left subfigures element-wise, then adding up, and finally dividing by the normalization terms.

The subsequent sections will discuss the details of each of the experiments, as well as how they are relevant to empirically show that our equivalence criterion has the properties that were mentioned previously and what insights can provide about the data.

6.2.1 GAD $\mathcal{NN}_3$ and GAD $\mathcal{NN}_5$. GAD $\mathcal{NN}_3$ and GAD $\mathcal{NN}_5$ are the same algorithm, namely GAD, but the neural network used for prediction is different. As such, one would expect they should be equivalent; Table 3 confirms that. On one hand, Figure 6(a) visually shows the similarity between the two sets of scores, not only with the score themselves, but their means and standard deviations depicted as Gaussian curves; on the other hand, we have the weights in Figure 7(a), where orange means positive (concordance) and blue means negative (discordance). We can observe from Figure 7(a) that the two algorithms did not have major disagreements; it was only minor disagreements that we can be seen as blue pixels near the diagonal.

Table 3. GEC for different AD algorithms

Alg 1 \ Alg 2	GAD $\mathcal{NN}_5$	SD	ADFT
GAD $\mathcal{NN}_3$	0.9592	0.3339	-0.202
GAD $\mathcal{NN}_5$		0.314	-0.218
SD			-0.222



(a) Top: GAD $\mathcal{NN}_3$; bottom: GAD $\mathcal{NN}_5$ (b) Top: SD; bottom: ADFT

Fig. 6. Regions for all the AD algorithms. The points belonging to the three classes are illustrated beneath the Gaussian curves with each color corresponding to each class. The shaded curves depict the Gaussian curves used for the weight calculation in equation (7).

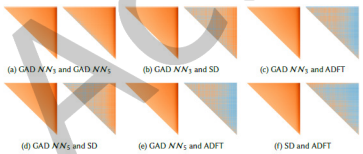


Fig. 7. Weights for GEC when comparing different AD algorithms. Both subfigures are indexed by the same index, which is the first anomaly score sorted. Orange means the weight is positive (concordance), and blue means the weight is negative (discordance). Since the indexing is the first anomaly score the left subfigure does not have discordances, in contrast, the right subfigure will always have discordances unless the second anomaly score yields the same ordering of the first anomaly score.

6.2.2 GAD $\mathcal{NN}_3$ and SD. This is a more interesting example, in that, whereas GAD and SD are both anomaly detection algorithms, they are intended to detect different patterns. Having an equivalence score of 0.3339 confirms that. We can observe in Figure 7(b) that SD has some local and nonlocal permutations with respect to GAD, hence the low score.

The analysis between GAD $\mathcal{NN}_5$ and SD or ADFT is the same as GAD $\mathcal{NN}_3$, so we will omit it.

6.2.3 A Case Study of How GEC Can Generate Insight on Data; GAD $\mathcal{NN}_3$ and ADFT. At first glance, it is unclear to what degree GAD and ADFT are equivalent. A score of -0.202 not only says that they are uncorrelated, but that they are inverse to a small degree. From Figure 7(c) we can observe that ADFT has many discordances

over points (points refers to the energy consumption over time signal) that are ranked highly anomalous by GAD; GAD ranks a point highly anomalous if such point has high prediction error. Then, this implies that points that have high prediction error do not have a large high frequency component, otherwise ADFT would rank them higher and there would not be so many discordances. Conversely, this suggests that points that have large high frequency component might have small prediction error. If we assume that neural network learns the most common patterns over the data distribution, then the implication is that high frequency points are prevalent over the dataset and hence not anomalous.

The analysis above illustrates a way in which insights can be gained about the data through GEC by comparing learning algorithms (GAD), whose behavior is driven mostly by data, with handcrafted algorithms (ADFT), whose behavior is driven by their designer.

6.2.4 SD and ADFT. SD is intended to capture when meters slow down in consumption. ADFT will capture meters whose consumption fluctuates significantly. That explains their low GEC, which like the previous case, it also indicates they are somewhat inversely correlated. In this case, unlike the previous case we cannot draw any conclusions about the data, since SD and ADFT are both handcrafted algorithms.

6.2.5 How robust is the algorithm to permutations? We perform an additional experiment to empirically test property 1. In this experiment we leave the output of GAD $\mathcal{NN}_3$ untouched and shuffle the output of GAD $\mathcal{NN}_5$, i.e., swap scores between points. The local shuffling consists of randomly shuffling scores from the 0 to 10th percentile, 10th to 20th, etc. The weights are shown in Figure 8(a), where we can see blocks near the diagonal that correspond to random shuffling. The GEC value for local random shuffling was 0.8671 which indicates the two algorithms are equivalent. The nonlocal (global) shuffling consists of swapping the first 2.5% scores with the last 2.5%, which is why the weights in Figure 8(b) have those blue regions. Even though most of the weights of GEC were unchanged, its value drops from 0.9592 to 0.3802.

Fig. 8. Robustness to permutations

7 CONCLUSIONS

We argue for the importance of the overlooked problem of having a criterion to measure equivalence between AD algorithms in the context of unsupervised learning. We devise a set of properties that are desirable for such equivalence criterion as they ensure highly anomalous points will have priority on the calculation of it. We propose, GEC, an equivalence criterion that satisfies all devised properties; in addition, GEC is flexible in that it allows the definition of classes, which serve to assign importance to anomalies; this is useful because different applications assign different values to anomalies (e.g., anomalies in medical applications are more critical than in

commercial applications). We mathematically show, and empirically validate that GEC satisfies the properties in simulated data as well as in real-world data.

REFERENCES

[1] David Alvarez-Melis and Tommi S. Jaakkola. 2018. Towards Robust Interpretability with Self-Explaining Neural Networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Montréal, Canada) *(NIPS'18)*. Curran Associates Inc., Red Hook, NY, USA, 7786–7795.

[2] Simon Duque Anton, Suneetha Kanoor, Daniel Fraunholz, and Hans Dieter Schotten. 2018. Evaluation of Machine Learning-Based Anomaly Detection Algorithms on an Industrial Modbus/TCP Data Set. In *Proceedings of the 13th International Conference on Availability, Reliability and Security* (Hamburg, Germany) *(ARES 2018)*. Association for Computing Machinery, New York, NY, USA, Article 41, 9 pages. https://doi.org/10.1145/3230833.3232818

[3] Riyaz Ahamed Ariyaluran Habeeb, Fariza Nasaruddin, Abdullah Gani, Ibrahim Abaker Targio Hashem, Ejaz Ahmed, and Muhammad Imran. 2019. Real-time big data processing for anomaly detection: A Survey. *International Journal of Information Management* 45 (2019), 289–307. https://doi.org/10.1016/j.ijinfomgt.2018.08.006

[4] Alberto Barbado. 2020. Anomaly detection in average fuel consumption with XAI techniques for dynamic generation of explanations. *ArXiv* abs/2010.16051 (2020).

[5] Alberto Barbado, Óscar Corcho, and Richard Benjamins. 2022. Rule extraction in unsupervised anomaly detection for model explainability: Application to OneClass SVM. *Expert Systems with Applications* 189 (2022), 116100. https://doi.org/10.1016/j.eswa.2021.116100

[6] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion* 58 (2020), 82–115. https://doi.org/10.1016/j.inffus.2019.12.012

[7] Richard J. Bolton and David J. Hand. 2002. Statistical Fraud Detection: A Review. *Statist. Sci.* 17, 3 (2002), 235 – 255. https://doi.org/10.1214/ss/1042727940

[8] Diogo V. Carvalho, Eduardo M. Pereira, and Jaime S. Cardoso. 2019. Machine Learning Interpretability: A Survey on Methods and Metrics. *Electronics* 8, 8 (2019). https://doi.org/10.3390/electronics8080832

[9] Stéphan Clémençon and Jérémie Jakubowicz. 2013. Scoring anomalies: a M-estimation formulation. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 31)*, Carlos M. Carvalho and Pradeep Ravikumar (Eds.). PMLR, Scottsdale, Arizona, USA, 659–667. https://proceedings.mlr.press/v31/clemencon13a.html

[10] Filipe Falcão, Tommaso Zoppi, Caio Barbosa Viera Silva, Anderson Santos, Baldoino Fonseca, Andrea Ceccarelli, and Andrea Bondavalli. 2019. Quantitative Comparison of Unsupervised Anomaly Detection Algorithms for Intrusion Detection. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing* (Limassol, Cyprus) *(SAC '19)*. Association for Computing Machinery, New York, NY, USA, 318–327. https://doi.org/10.1145/3297280.3297314

[11] Tharindu Fernando, Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. 2021. Deep Learning for Medical Anomaly Detection – A Survey. *ACM Comput. Surv.* 54, 7, Article 141 (July 2021), 37 pages. https://doi.org/10.1145/3464423

[12] Milan. Flach, Fabian. Gans, Alexander. Brenning, Joachim. Denzler, Markus. Reichstein, Erik. Rodner, Sebastian. Bathiany, Paul. Bodesheim, Yanira. Guanche, Sebastian. Sippel, and Miguel. D. Mahecha. 2017. Multivariate anomaly detection for Earth observations: a comparison of algorithms and feature extraction techniques. *Earth System Dynamics* 8, 3 (2017), 677–696. https://doi.org/10.5194/esd-8-677-2017

[13] Ralph Foorthuis. 2021. On the nature and types of anomalies: a review of deviations in data. *International Journal of Data Science and Analytics* 12 (10 2021), 1–35. https://doi.org/10.1007/s41060-021-00265-1

[14] Martin; Gnoss, Nico; Schultz and Marina Tropmann-Frick. 2022. XAI in the Audit Domain- Explaining an Autoencoder Model for Anomaly Detection. *Wirtschaftsinformatik 2022 Proceedings 1* (2022). https://aisel.aisnet.org/wi2022/business_analytics/business_analytics/1

[15] Nicolas Goix. 2016. How to Evaluate the Quality of Unsupervised Anomaly Detection Algorithms? arXiv:1607.01152 [stat.ML]

[16] Nicolas Goix, Anne Sabourin, and Stéphan Clémençon. 2015. On Anomaly Ranking and Excess-Mass Curves. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 38)*, Guy Lebanon and S. V. N. Vishwanathan (Eds.). PMLR, San Diego, California, USA, 287–295. https://proceedings.mlr.press/v38/goix15.html

[17] Zirije Hasani. 2017. Robust anomaly detection algorithms for real-time big data: Comparison of algorithms. In *2017 6th Mediterranean Conference on Embedded Computing (MECO)*. 1–6. https://doi.org/10.1109/MECO.2017.7977130

[18] Robert R. Hoffman, Shane T. Mueller, Gary Klein, and Jordan Litman. 2018. Metrics for Explainable AI: Challenges and Prospects. *CoRR* abs/1812.04608 (2018). arXiv:1812.04608 http://arxiv.org/abs/1812.04608

[19] Maurice. G. Kendall. 1938. A New Measure of Rank Correlation. *Biometrika* 30, 1/2 (1938), 81–93. http://www.jstor.org/stable/2332226

[20] Donghyun Kim, Gian Antariksa, Melia Putri Handayani, Sangbong Lee, and Jihwan Lee. 2021. Explainable Anomaly Detection Framework for Maritime Main Engine Sensor Data. *Sensors* 21, 15 (2021). https://doi.org/10.3390/s21155200

[21] Martin Q. Ma, Yue Zhao, Xiaorong Zhang, and L. Akoglu. 2021. A Large-scale Study on Unsupervised Outlier Model Selection: Do Internal Strategies Suffice? *ArXiv* abs/2104.01422 (2021).
[22] Henrique O. Marques, Ricardo J. G. B. Campello, Jörg Sander, and Arthur Zimek. 2020. Internal Evaluation of Unsupervised Outlier Detection. *ACM Trans. Knowl. Discov. Data* 14, 4, Article 47 (June 2020), 42 pages. https://doi.org/10.1145/3394053
[23] Henrique O. Marques, Ricardo J. G. B. Campello, Arthur Zimek, and Jörg Sander. 2015. On the internal evaluation of unsupervised outlier detection. In *Proceedings of the 27th International Conference on Scientific and Statistical Database Management, SSDBM '15, La Jolla, CA, USA, June 29 - July 1, 2015*, Amarnath Gupta and Susan L. Rathbun (Eds.). ACM, 7:1–7:12. https://doi.org/10.1145/2791347.2791352
[24] Maarten Meire and Peter Karsmakers. 2019. Comparison of Deep Autoencoder Architectures for Real-time Acoustic Based Anomaly Detection in Assets. In *2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Vol. 2. 786–790. https://doi.org/10.1109/IDAACS.2019.8924301
[25] Thanh. Nguyen, Van. Nguyen and Uy Nguyen. 2017. An evaluation method for unsupervised anomaly detection algorithms. *Journal of Computer Science and Cybernetics* (2017). https://vjs.ac.vn/index.php/jcc/article/view/8455
[26] Guansong Pang and Charu Aggarwal. 2021. Toward Explainable Deep Anomaly Detection. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Virtual Event, Singapore) *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 4056–4057. https://doi.org/10.1145/3447548.3470794
[27] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. 2021. Deep Learning for Anomaly Detection: A Review. *ACM Comput. Surv.* 54, 2, Article 38 (March 2021), 38 pages. https://doi.org/10.1145/3439950
[28] Mahdi Rabbani, Yongli Wang, Reza Khoshkangini, Hamed Jelodar, Ruxin Zhao, Sajjad Bagheri Baba Ahmadi, and Seyedvalyallah Ayobi. 2021. A Review on Machine Learning Approaches for Network Malicious Behavior Detection in Emerging Technologies. *Entropy* (04 2021). https://doi.org/10.3390/e23050529
[29] Sebastiano Vigna. 2015. A Weighted Correlation Index for Rankings with Ties. In *Proceedings of the 24th International Conference on World Wide Web* (Florence, Italy) *(WWW '15)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1166–1176. https://doi.org/10.1145/2736277.2741088
[30] Véronne Yepmo, Grégory Smits, and Olivier Pivert. 2022. Anomaly explanation: A review. *Data & Knowledge Engineering* 137 (2022), 101946. https://doi.org/10.1016/j.datak.2021.101946
[31] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. 2020. Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. 737–744. https://doi.org/10.1109/SSCI47803.2020.9308468

A PROPERTIES OF GEC

A.1 Property 1

Lemma A.0.1. *For a given $0 < \omega \leq \omega_{\max}$, there is a maximum k, which is the largest permutation move whose weight is smaller than ω.*

Proof. We will provide a constructive proof. 1) Pick an arbitrary k, then calculate $\omega(i, i+k)$ for $1 \leq i \leq n-k$ and pick the largest one, let us call it $\omega_{k_{\max}}$. If $\omega_{k_{\max}} \leq \omega$, then 2.a) repeat step 1) for $k+1$ and check for terminating condition. Else if $\omega_{k_{\max}} \geq \omega$, then 2.b) repeat step 1) for $k-1$ and check for terminating condition. The terminating condition occurs when $\omega_{l_{\max}} \leq \omega \leq \omega_{(l+1)_{\max}}$ for $l \in \{1, 2, ..., n-1\}$. By properties of the Gaussian distance $\omega(h, j) < \omega(h', j')$ for $h' < h < j < j'$ (Recall h, j, h', j' are indexes of an anomaly score), therefore $\omega_{k_{\max}} < \omega_{(k+1)_{\max}}$. Hence, if $\omega_{k_{\max}} \leq \omega \leq \omega_{(k+1)_{\max}}$ implies that $k+1$-move (or higher) permutations have a weight higher than ω, and any $k-1$-move (or smaller) permutation weight will be smaller than $\omega_{k_{\max}}$, therefore l is the largest permutation move whose weight is smaller than ω. This program is guaranteed to terminate because the number of possible permutation moves is finite. □

Corollary A.0.1. *Any permutation move smaller than k moves will have a weight smaller than $\omega_{k_{\max}}$*

Corollary A.0.2. *Only allowing permutations with weights smaller than ω is equivalent to only allowing permutations with at most k moves.*

Theorem 5.1. *Let $\boldsymbol{r}$ be a sorted set of n anomaly detection scores, let $\boldsymbol{s}$ be a permutation of $\boldsymbol{r}$ that only allows permutation moves with at most $\omega < \frac{2}{n(n-1)} \sum \omega_r(i,j)\omega_s(i,j) = \mu$ weight, and more concordant pairs, c, than discordant pairs, d, with respect to r, then $0 < \tau(\boldsymbol{r}, \boldsymbol{s}) < \phi(\boldsymbol{r}, \boldsymbol{s})$.*

Proof. If $k_1 = \omega_r(i,j)$ and $k_2 = \omega_s(i,j)$ for every i, j, then $\tau_{r,s} = \phi_{r,s}$. We also have that $\phi(\boldsymbol{r}, \boldsymbol{s})$ can be decomposed as $\phi(\boldsymbol{r}, \boldsymbol{s}) = \sum_{i,j \in c} \omega_r(i,j)\omega_s(i,j) - \sum_{i,j \in d} \omega_r(i,j)\omega_s(i,j) = \mu^+ c - \mu^- d$, where c and d refer to the concordant and discordant pairs, respectively. For τ we have that $\mu^+ = \mu^-$. Therefore, to show the inequality it is enough to show that $\mu^+ > \mu^-$ for $\phi_{r,s}$. We have that $\mu = \frac{\mu^+ + \mu^-}{2} \Rightarrow 2\mu = \mu^+ + \mu^-$. We have that μ^- is the average of all discordant pairs, which are, as required by the Theorem, all smaller than ω, hence $\mu^- < \mu$, which implies $\mu^+ > \mu > \mu^-$, thus proving the inequality. □

A.2 Property 2

Theorem 5.2. *Let $\boldsymbol{r}$ be an anomaly score, let R_m and R_n be two different regions such that $R_m = \{1, 2, ..., u\}, R_n = \{1, 2, ..., v\}$ with $u \gg v$ (meaning that R_m has many more data points than R_n), and u and v are large enough for the law of large numbers to be applicable. Consider all k-move permutations within a class with $v \gg k$, then $\frac{1}{u-k} \sum_{i \in R_n} \omega_r(i, i+k) < \frac{1}{v-k} \sum_{i \in R_m} \omega_r(i, i+k)$, namely, the average weight for k-move permutations is higher in region R_n.*

Proof. We will first consider the contribution of the point's own region, which is on average much higher than the contributions of other regions. We have that $\omega_r(-\infty, \infty) = 1$, and $\omega_r(a,b) + \omega_r(b,c) = \omega_r(a,c)$, moreover for a small enough and large enough a, b, respectively, we have that $\omega_r(a,b) \approx 1$. Let $P_i = \{j : j \mod i = i-1 \text{ for } j \in R_q\}$. We have that $\sum_{i \in R_n} \omega_r(i, i+k) = \sum_{j=1}^{k} \left(\sum_{p \in P_j} \omega_r(p, p+1) \right) = \sum_{j=1}^{k} \omega_r(\min(P_j), \max(P_j)) \approx \sum_{j=1}^{k} 1 = k$; this is true regardless of R_m or R_n. Since $u \gg v$, then $\frac{1}{u-k}k < \frac{1}{v-k}k$.

For the contributions of other regions, we use a similar argument to the one presented above, but this time we have that $\sum_{i \in R_n} \omega_r(i, i+k) = \epsilon$, where ϵ is a small number, and just like in the argument above, the term $\frac{1}{u-k}$ being much smaller than $\frac{1}{v-k}$ will dominate. □

Lemma 5.2.1. *Let $S = \{s_1, s_2, ..., s_n\}$ be 0 mean and have standard deviation σ, and $s_i < s_j$ for $i < j$, additionally let $P = \alpha S = \{p_1, p_2, ..., p_n\}$ with α positive, then the standard deviation of P is $\sigma_p = \alpha\sigma$ and $F_\sigma(s_i, s_j) = F_{\sigma_p}(p_i, p_j)$*

Proof. $\sigma = \sqrt{\frac{1}{n} \sum_i s_i^2}$, and $\sigma_p = \sqrt{\frac{1}{n} \sum_i p_i^2} = \sqrt{\frac{1}{n} \sum_i (\alpha s_i)^2} = \alpha \sqrt{\frac{1}{n} \sum_i s_i^2} = \alpha\sigma$, thus proving the first statement.

We have that $F_\sigma(s_i, s_j) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{s_j}{\sigma\sqrt{2}}\right)\right] - \frac{1}{2}\left[1 + \text{erf}\left(\frac{s_i}{\sigma\sqrt{2}}\right)\right] = \text{erf}\left(\frac{s_j}{\sigma\sqrt{2}}\right) - \text{erf}\left(\frac{s_i}{\sigma\sqrt{2}}\right)$, where erf is the Gauss error function; on the other hand $F_{\sigma_p}(p_i - p_j) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{p_j}{\sigma_p\sqrt{2}}\right)\right] - \frac{1}{2}\left[1 + \text{erf}\left(\frac{p_i}{\sigma_p\sqrt{2}}\right)\right] = \text{erf}\left(\frac{p_j}{\sigma_p\sqrt{2}}\right) - \text{erf}\left(\frac{p_i}{\sigma_p\sqrt{2}}\right) = \text{erf}\left(\frac{\alpha s_j}{\alpha\sigma\sqrt{2}}\right) - \text{erf}\left(\frac{\alpha s_i}{\alpha\sigma\sqrt{2}}\right) = F_\sigma(s_i, s_j)$ □

B FURTHER ANLYSIS OF SEC

B.1 Mean

We will consider the mean assuming the rankings are random with uniform distribution.

Let $Z = \sum_{i=1}^{n} x_i y_i$ and $k = \sum_{i=1}^{n} i^2$. Let y_i be the ith component of Y. We are interested in the mean and standard deviation of $W = \frac{Z}{k}$. Note that $0.5 \leq W \leq 1$.

As for the mean

$$\mathbb{E}[W] = \mathbb{E}\left[\frac{Z}{k}\right] = \frac{1}{k}\mathbb{E}\left[\sum_{i=1}^{n} x_i y_i\right] = \frac{1}{k}\sum_{i=1}^{n}\mathbb{E}[x_i y_i] = \frac{n}{k}\mathbb{E}[x]\mathbb{E}[y] = \frac{n}{k}\frac{n}{2}\frac{n}{2} \tag{9}$$

$k = \frac{n(n+1)(2n+1)}{6}$. Then $\mathbb{E}[W] = \frac{6n^3}{8n^3 + 12n^8 + n}$, for n large enough the cube terms dominate, so $\mathbb{E}[W] = 3/4$, which actually agrees with the experiments.

B.2 Property 2

Lemma 5.2.2. *$\sum_{i=n}^{n+k} i^2 - \sum_{i=n}^{n+k} i \cdot p_i$ does not depend on n*

$$\begin{aligned}
\sum_{i=n}^{n+k} i^2 - \sum_{i=n}^{n+k} i \cdot p_i &= n + (n+1)^2 + \ldots + (n+k)^2 - (n(n+p_n) + \ldots + (n+k)(n+p_{n+k})) \\
&= kn^2 + n(2 + 4 + \ldots + 2k) + k_c - (kn^2 + n(p_n + 1 + p_{n+1} + \ldots + k + p_{n+k}) + k_p) \\
&= kn^2 + n(2 + 4 + \ldots + 2k) + k_c - (kn^2 + n(1 + 1 + 2 + 2 + \ldots + k + k) + k_p) \\
&= k_c - k_p
\end{aligned} \tag{10}$$

C ANOMALY DETECTION ALGORITHMS

C.1 General Anomaly Detection (GAD)

This algorithm assumes that a neural network, $\mathcal{NN}$, has already been trained to predict the next time interval given the current interval. kWh sequence refers to the daily energy consumption registered by the meters.

Algorithm 1: General anomaly detection (GAD)

Input: kWh sequences, $\mathcal{X}$; neural network, $\mathcal{NN}$; smoothing kernel, $\mathcal{K}$
Output: Anomaly scores, $\mathcal{S}$
Obtain prediction $\overline{\mathcal{X}}_{:,t:T} = \mathcal{NN}(\mathcal{X}_{:,1:t})$
$e \leftarrow (\mathcal{X}_{:,t:T} - \overline{\mathcal{X}}_{:,t:T})^2$
for $i = t, T$ **do**
    $\delta_i \leftarrow |e_{:,i} - e_{:,i-k:i} \cdot \mathcal{K}|$
end
Divide δ by the median along time and meter axis: $\delta \leftarrow \delta/\overline{\mu}_{\delta_t}$, $\delta \leftarrow \delta/\overline{\mu}_{\delta_n}$
Compute max δ for each meter: $\mathcal{S} \leftarrow \max(\delta)$

Using neural network prediction error for detecting anomalies is a standard AD algorithm [11]. Figure 9 shows the four most anomalous points found by GAD; they are all characterized by an abrupt change in consumption, which explains why the neural network would have a high prediction error.

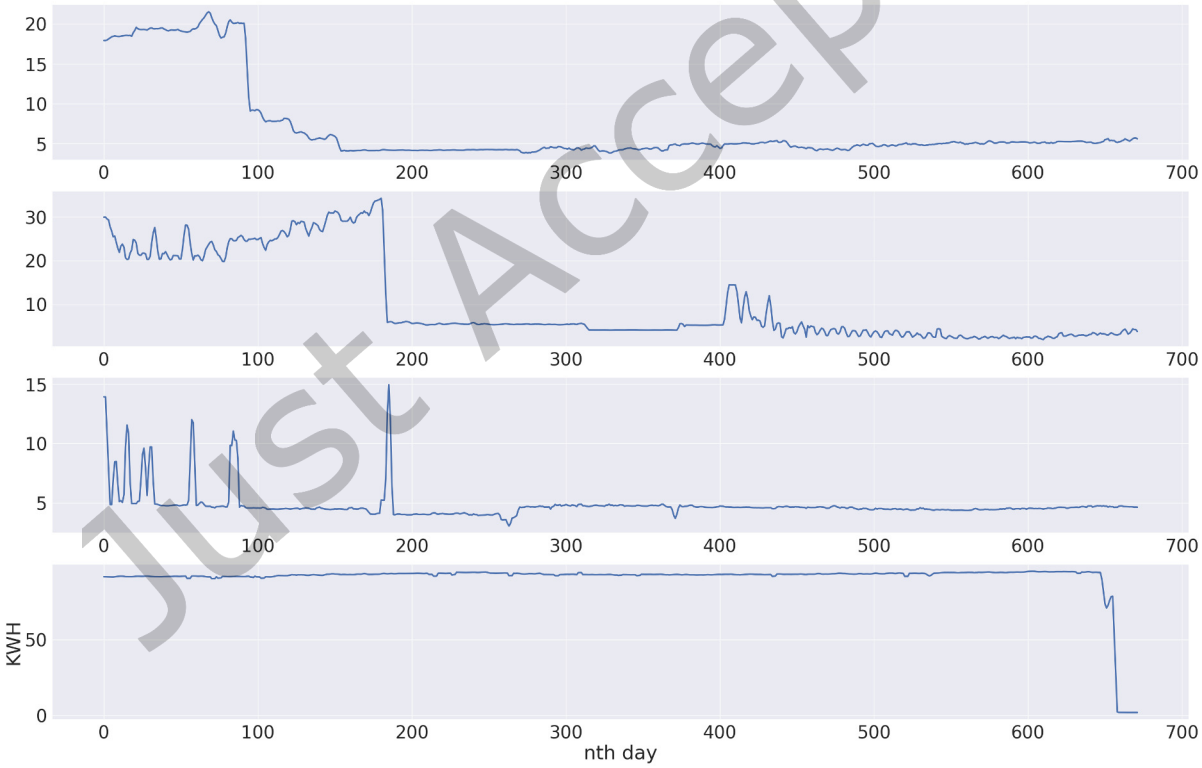


Fig. 9. Top 4 anomalous points found by GAD

C.2 Slowing-down Meters Detection Algorithm (SD)

This is a hand-crafted algorithm that specializes in finding meters whose consumption is slowing down (or speeding up). The algorithm is described by the following equation

$$SD = \frac{N(k(n) - k(n+N))}{\sum_{i=1}^{N-1} |k(i) - k(i+1)|} \tag{11}$$

where k is the kWh signal, and N is the data length to be considered.

Figure 10 shows the four most anomalous points found by SD. It can be observed that SD effectively finds meters whose consumption is decreasing whether abruptly as in the third image (top to bottom) or more steadily as in the first image (top to bottom). For this application, the first meter is not of much concern as gradual decrease in consumption is generally due to users decreasing consumption; on the other hand the third meter is of more concern as that sudden decrease in consumption could be a signal of a faulty meter.

Furthermore, the third meter was also flagged as anomalous by GAD. As mentioned previously, generally GAD detects meters that present an abrupt change in behavior; SD detects meters whose consumption is decreasing. In combination they would detect meters whose consumption is abruptly decreasing, which are more likely to be faulty meters. This reasoning is supported by their GEC of 0.3339, that is, it is not meaningful to determine which one is "better", instead they can be complimentary to each other, thus producing a more precise AD algorithm for the task at hand.

It is impossible to predict what insights GEC can provide about data and AD algorithms because it highly depends on the application and AD algorithms; however, GEC does provide a novel layer of analysis towards understanding different AD algorithms, and hence the dataset, in the unsupervised learning setting.

C.3 Anomaly Detection by Fourier Transform (ADFT)

This algorithm uses the Fourier transform to obtain the frequency decomposition and then the signals with high power in the high frequency range are labelled as anomalies. The detailed algorithm is shown in Algorithm 2.

Algorithm 2: Anomaly detection by Fourier transform (ADFT)

Input: kWh sequences, $\mathcal{X}$
Output: Anomaly scores, $\mathcal{S}$
Calculate Fourier transform $F = \mathcal{F}(\mathcal{X})$
Normalize with $F = F/\sum_i F_i$ to have sum of all components add up to 1
Consider only high frequency components F_H
Score is $\mathcal{S} \leftarrow \sum_i F_{H,i}$

Figure 10 shows the four most anomalous points found by ADFT. These points look substantially different than the ones found by GAD and SD, which explains why the GEC values were -0.202 and -0.218, respectively. Moreover, excluding the first one, those consumption profiles are not necessarily anomalous. This partially confirms the analysis done in section 6.2.3, where the claim is that ADFT might not be useful because the points with large high frequency component are not anomalous.

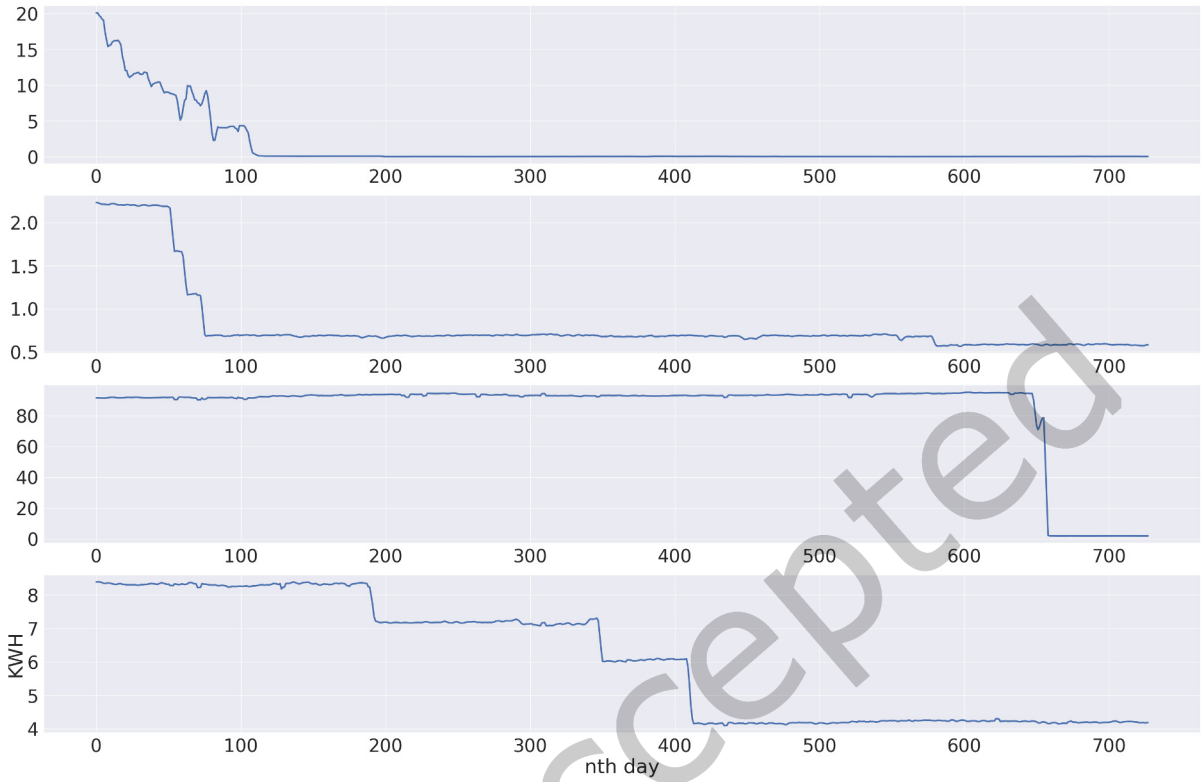


Fig. 10. Top 4 anomalous points found by SD

D KENDALL TAU COEFFICIENT AND SEC EXAMPLE

First, we will illustrate how the Kendall Tau coefficient is calculated with a simple example. Let $\boldsymbol{m}$ and $\boldsymbol{n}$ be rankings that rank the points $[1, 2, 3, 4, 5, 6]$ as follows: $\boldsymbol{m} = [4, 6, 3, 2, 1, 5]$ and $\boldsymbol{n} = [6, 4, 2, 3, 5, 1]$, namely element 4 is ranked one according to $\boldsymbol{m}$, but two according to $\boldsymbol{n}$. Note that the enumeration $[1, 2, 3, 4, 5, 6]$ is just an indexing; it could have been any other enumeration, e.g. $[a, b, c, d, e, f]$. In fact, we will re-index $\boldsymbol{m}$ such that $\boldsymbol{m} = [1, 2, 3, 4, 5, 6]$, which yields $\boldsymbol{n} = [2, 1, 4, 3, 6, 5]$, that is, we rename element 4 to 1, which is the rank given by $\boldsymbol{m}$, then 6 to 2, etc. This way, we can simply use the order of the naturals to find concordances and discordances, as shown in Figure 3. Note that $\langle \boldsymbol{m}, \boldsymbol{n} \rangle = \sum_{j<i} \operatorname{sgn}(m_i - m_j)\operatorname{sgn}(n_i - n_j)$ is the element-wise multiplication of the subfigures shown in Figure 11 and subsequent summation.

In this particular example $\langle \boldsymbol{m}, \boldsymbol{n} \rangle = 9$, which comes from 12 concordances, and 3 discordances.

For the SEC example, we will arbitrarily use the following two scores: $\boldsymbol{r} = [(a, 0.5), (b, 1.2), (c, 1.6)]$ and $\boldsymbol{s} = [(a, 1.3), (b, 0.2), (c, 0.7)]$, where for $\boldsymbol{r}$, the tuple $(a, 0.5)$ means the point labeled a got a score of 0.5. Recall that SEC, also denoted as σ, is given by $\sigma = 4\sigma_n - 3$, where $\sigma_n = \dfrac{\sum_i^n \hat{r}_i \hat{s}_i}{\sum_{i=1}^n i^2}$. The denominator is just a constant, whereas the numerator is calculated as follows: $\sum_{i=1}^3 \hat{r}_i \hat{s}_i = 1 \cdot 3 + 2 \cdot 1 + 3 \cdot 2 = 11$. Then, $\sigma = 4 \cdot \frac{11}{14} - 3 = 0.1428$, which is low because $\boldsymbol{r}$ and $\boldsymbol{s}$ order a, b, c completely differently.

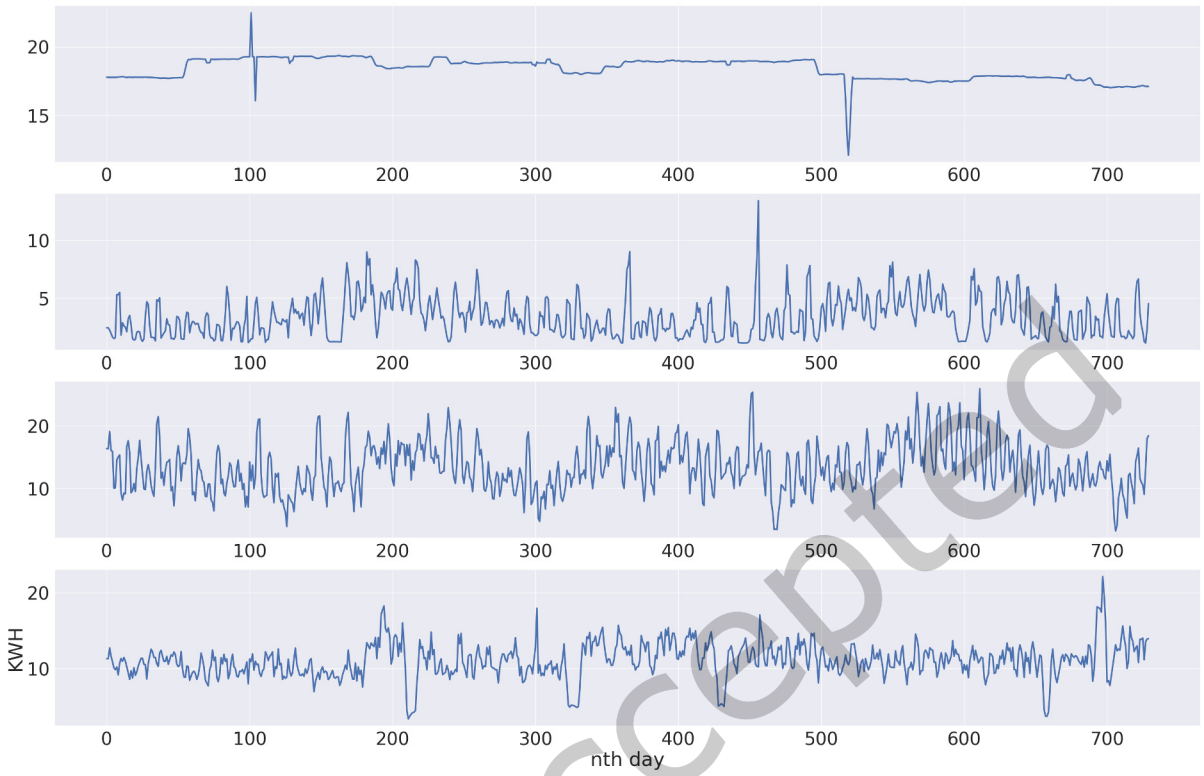


Fig. 11. Top 4 anomalous points found by ADFT

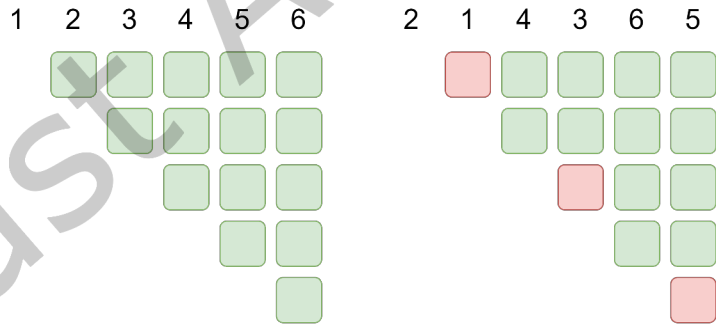


Fig. 12. Terms of the form $\text{sgn}(m_i - m_j)$ for $\boldsymbol{m}$ and $\boldsymbol{n}$. Green indicates accordance; red indicates discordance. For example, $\boldsymbol{m}$ assigns the order 1 then 2, whereas $\boldsymbol{n}$ 2 then 1, hence the discordance on the right side.

E PSEUDOCODE

The pseudocode will be divided into two blocks:

C_i, j refers to the jth region (subclass) of the anomaly score i (i takes values $\boldsymbol{r}$ or $\boldsymbol{s}$).

Algorithm 3: Distance measure

Input: $\boldsymbol{r}, \boldsymbol{s}$
Sort $\boldsymbol{r}$ and $\boldsymbol{s}$ according to $\boldsymbol{r}$ and assign corresponding indexes
Sort $\boldsymbol{s}$ according to $\boldsymbol{s}$
Define regions (subclasses) by setting the percentiles of each class for $\boldsymbol{r}$ and $\boldsymbol{s}$
Calculate mean and standard deviation for each region for $\boldsymbol{r}$ and $\boldsymbol{s}$
for $i = \boldsymbol{r}, \boldsymbol{s}$ **do**
    for j, k =*all combinations of regions* **do**
        Compute $\omega(j, k)$ for all points in $C_{i,j}$ and $C_{i,k}$
        Compute sign function, sgn, between $C_{i,j}$ and $C_{i,k}$
        Calculate coefficients as $\omega \cdot sgn$
        Input coefficients to the placeholder matrix in the corresponding coordinates
    end
end
Compute $\langle \boldsymbol{r}, \boldsymbol{s} \rangle_\omega$ coefficient from placeholder matrix

Algorithm 4: Gaussian Equivalence Criterion (GEC)

Input: r, p
Compute $||\boldsymbol{r}||^2$ with the Distance measure algorithm using as input (r, r)
Compute $||\boldsymbol{s}||^2$ with the Distance measure algorithm using as input $(\boldsymbol{s}, \boldsymbol{s})$
Compute $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$ with the Distance measure algorithm using as input $(\boldsymbol{r}, \boldsymbol{s})$
Calculate the GEC value, ϕ, as:

$$\phi = \frac{\langle \boldsymbol{r}, \boldsymbol{s} \rangle}{||\boldsymbol{r}||||\boldsymbol{s}||}$$