Explaining Anomalies through Semi-supervised Autoencoders

Fabrizio Angiulli[a], Fabio Fassetti[a], Luca Ferragina[a], Simona Nisticò[a,*]

[a]*DIMES, University of Calabria, Via Pietro Bucci, Rende, 87036, Italy*

Abstract

This work deals with the problem of providing intelligible *explanations* to abnormal behaviours in input data observations. In particular, we adopt *heatmaps* as explanations, where a heatmap can be regarded as a collection of per-feature scores. In order to explain anomalies, our approach, called AE–XAD[1] (for AutoEncoder-based eXplainable Anomaly Detection), extends a recently introduced semi-supervised variant of the Autoencoder architecture. The main idea of AE–XAD is to exploit a reconstruction error strategy for detecting deviating features. Unlike standard Autoencoders, it leverages a semi-supervised loss designed to maximise the distance between the reconstruction and the original value assumed by anomalous features. By means of this strategy, our approach learns to isolate anomalous portions of the input observations using only a few anomalous examples during training. Experimental results highlight that AE–XAD delivers high-level performance in explaining anomalies in different scenarios while maintaining a minimal CO_2 footprint, showcasing a design that is not only highly effective but also environmentally conscious.

Keywords: Explainable Anomaly Detection, Green-aware AI, Explainability by design

[*] Corresponding author(s)

Email addresses: fabrizio.angiulli@unical.it (Fabrizio Angiulli), fabio.fassetti@unical.it (Fabio Fassetti), luca.ferragina@unical.it (Luca Ferragina), simona.nistico@unical.it (Simona Nisticò)

[1] The code of AE–XAD will be available upon acceptance.

1. Introduction

Explainable Artificial Intelligence (xAI) [1, 2] refers to the development of all those techniques pursuing the objective of making Machine Learning models less opaque to humans. This includes not only the ability to interpret the model's predictions, but also to understand the underlying reasoning. Explainable models can help increase trust and awareness in the model's decisions, as well as enable an easier identification and correction of the model's errors and weaknesses. Although the xAI field is receiving considerable attention, most of the work related to model explanation is tailored to classification or regression tasks. Differently, less attention is dedicated to models addressing the anomaly detection task, which, however, due to its peculiarities, requires ad-hoc methods. The explanation problem in the Anomaly Detection field translates into the search for reasons justifying why a model judges data points as anomalous.

This work deals with the *Explainable Anomaly Detection* problem (xAD). Explainable Anomaly Detection is defined in [3] as "the extraction of relevant knowledge from an anomaly detection model concerning relationships either contained in data or learned by the model, where the knowledge is considered relevant if it can provide insights into the anomaly detection problem investigated by the end-user", so it includes both model explainability (problem addresses in this paper) and data explainability [4].

The goal here is not only to signal anomalous test observations as in classical anomaly detection, but more specifically to be able to single out an intelligible *explanation* for the provided anomaly score. In this work, we adopt *heatmaps* as explanations, where a heatmap can be regarded as a collection of per-feature scores.

The proposed approach, called AE-XAD (for AutoEncoder-based eXplainable Anomaly Detection), extends a recently introduced semi-supervised variant of the Autoencoder architecture to inject explainability inside the AD architecture [5].

Specifically, AE-XAD exploits a reconstruction error strategy for isolating deviating examples. Differently from standard Autoencoders, it takes advantage of a semi-supervised loss aiming at maximizing the distance between the reconstruction and the original value assumed by anomalous features, and dissimilarly from the approach in [5], our approach is capable to learn to isolate anomalous portions of the input observations by leveraging only a few anomalous examples during training. This explanation, derived from the model's internal state, highlights the features that contribute most to the anomaly score, which is used to single out outliers, thereby providing more informative predictions.

In this work, we will consider also sustainability concerns by adopting a two-faced green-aware perspective. On one side, It will indeed pursue the goal of optimizing explanations to provide the user with more informative and precise explanations enabling for a more detailed analysis, which is helpful in more precisely detect defects or other similar issues when employed in settings involving, for example, industrial application as well as predictive maintenance and thus reduce wastes thanks to more targetted human intervention. Furthermore, even more importantly, we will consider the system carbon footprint to search for a trade-off between energy consumption and model performances.

Experimental results highlight the algorithm's effectiveness in explaining anomalies across different scenarios, as well as its ability to well separate anomalous portions, resulting in more interpretable explanations that enhance user understanding.

The main contribution of the paper can be summarized as follows.

- We introduce AE–XAD, a novel AE-based methodology for eXplainable Anomaly Detection based on the introduction of a novel loss that enhances the reconstruction error on anomalous portions of the images.
- We demonstrate that AE–XAD outperforms the competitors across various datasets and experimental scenarios.
- We measure the CO_2 emissions of AE–XAD, demonstrating a design that is both effective and environmentally responsible.

The rest of the paper is organised as follows. In Section 2, we discuss the related works. In Section 3, we formalise the specific setting of the problem considered in this paper. In Section 4, we provide a description of the model as well as a detailed example that clarifies how AE–XAD works. Section 5 is devoted to experimental tests and, finally, Section 6 concludes the paper.

2. Related Works

2.1. Explainable Artificial Intelligence.

The methods described above have the drawback of giving final users no insight into which determinants led to the model's outcome. Such knowledge is required in various scenarios for debugging or informative purposes and developing the awareness of experts and final users. To fill this gap, the eXplainable Artificial Intelligence (xAI) field proposes solutions to explain black-box models either after their development, a setting referred to as post-hoc explainability, or yet in the development process, a task known as explainability by design.

Since they work with yet designed and ready-to-use models, methods falling in the first setting are broadly applicable; for this reason, this scenario has attracted much research attention. As a matter of fact, several methods have been proposed to design explanations for classification and regression [6, 7, 8] tasks, as well as for recommendation systems [9, 10, 11, 12] and so forth. Additionally, many data types have been considered, resulting in having post-hoc methods tailored to, for example, tabular data [13, 14, 15, 16], images [17, 18], time series [19, 20] and textual data [21, 22]. Apart from fidelity concerns arising over those methods [23], and despite their applicability to several situations, they perform poorly in the specific context of anomaly detection. Anomalies are, by definition, rare; thus, methods aiming at explaining anomaly detectors cannot rely on large data availability or on a straightforward data generation process. Some attempts have been made to design post-hoc explainability techniques for anomaly detection models [24, 25, 26, 27, 28, 29]. Alternatively, outlier explanation methods are used by substituting expert-generated anomaly annotations with model labels [30, 31, 32, 33]. In any case, neither strategy exploits samples beyond those contained in the available set, as is typically in "standard" post-hoc explanation settings.

As for the latter category of methodologies, in addiction to models that can be explained by virtue of their simplicity, methods leverages the attention mechanism that they use to develop self-explainability [34], or binds the latent space of their networks to be aligned with concept known from the training data [11, 35] or construe interpretability in terms of concept learning [36, 37] has been proposed in literature. Another path explored is to progressively generalise linear models (which are explainable by default) to complex but architecturally explicit models [38]. Recently, some efforts have been made to build frameworks in which a classifier and an explainer learns jointly [39, 40].

None of the previously described self-explainable methods considers the anomaly detection task, which has peculiarities compared to common classification. This class of methods is described below.

2.2. Deep Explainable Anomaly Detection.

Because of the great performances they reached, deep learning methods for anomaly detection [41, 42] have become very widespread. Typically, these methods can be divided into three categories: *reconstruction error*-based approaches employing Autoencoders (AEs), models relying on Generative Adversarial Networks (GANs), and *Deep–SVDD*-based methods that combine the approach of SVM with deep neural networks.

An Autoencoder [43, 44] is a special type of neural network composed by an *encoder* compressing data into a low-dimensional *latent space* and a decoder *decoder* mapping them back into the original space and aiming at obtaining a reconstruction of the data. Since anomalies usually form a rare class in the dataset, they will be reconstructed worst than the normal items, thus the *reconstruction error*, namely the mean squared error between a point and its reconstruction obtained with the AE, can be used as anomaly score [45, 46, 47, 48, 5].

As for methods based on Generative Adversarial Networks [49, 50, 51], basically they rely on a training mechanism where a generative network creates artificial anomalies gradually more realistic, and a discriminative network that performs the anomaly detection task by assigning an anomaly score to each item.

Deep–SVDD-based approaches address the task of anomaly detection by mapping normal data close to a fixed center, by means of a deep neural network, into a low-dimensional feature space [52, 53, 54] or directly providing, in the last layer of the network, the value of the anomaly score, forcing it to be, for anomalous data, as far as possible to the mean of the ones of randomly sampled normal data [55].

The neural networks-based anomaly detection models described above all produce an anomaly score, providing no justification about it. The only exception is represented by Autoencoders, indeed these architectures have an intrinsic concept of explanation, namely an *heatmap* highlighting most anomalous features, represented by the feature-wise squared difference between a point and its reconstruction. The sum of the entries of the heatmap is equal to the reconstruction error of the item; thus, the value of each feature represents the contribution of this feature to its anomaly score. However, the weak point of Autoencoders is that they usually become so able to generalize to well reconstruct also anomalies. This fact implies a deterioration of the performance in both the anomaly detection and explanation tasks.

As for the other deep learning models for Anomaly Detection, some of them have been expanded to include an explanation module.

FCDD [56], for example, arises as a variant of the algorithms in [52, 53] designed to work on image data and aiming at obtaining a heatmap highlighting anomalous areas of images.

In more detail, the idea behind FCDD is to train a Convolutional Neural Network that maps images from their native space to a lower-dimensional space that produces a feature map $A(\mathbf{x})$. The goal of loss of FCDD is to minimize the entries of the feature map $A(\mathbf{x})$ of normal examples and maximize the ones of anomalies in the training set, relying on the idea that the network will map the portions of test images that it considers normal on areas of $A(\mathbf{x})$ with low values, and portions that it considers anomalous on areas of $A(\mathbf{x})$ with high values. Thus, the feature map $A(\mathbf{x})$, resized as the input data, represents the heatmap, given in output by the model, and its norm $||A(\mathbf{x})||_1$ is the anomaly score.

Differently, BGAD [57] proposes a two-phase training strategy built upon a Conditional Normalizing Flow, which models the log-likelihood distribution of normal features. In the first phase, the model learns a normalised distribution of normal data and derives a compact and explicit anomaly-independent separating boundary based only on the log-likelihoods of normal samples. In the second phase, tailored to boundary refinement, the model simultaneously pulls toward the distribution centre normal samples whose log-likelihoods are below the boundary, while simultaneously pushing away from the boundary above-threshold anomalous samples.

Another approach to Explainable Deep Anomaly Detection that has been recently proposed is described in [58] and consists in a modification of DevNet, the network introduced in [55], aimed at producing a saliency map for interpreting anomalous data by applying a gradient-based methodology. Although similar to ours, this approach provides an explanation a posteriori with respect to the anomaly detection process.

Similarly, in [59] DRAEM is introduced, a discriminatively trained surface anomaly detection method that jointly learns a reconstruction and anomaly embedding. It employs a reconstructive sub-network to inpaint anomaly-free content and a discriminative sub-network to segment anomalies from the joint original-reconstructed input. Unlike our approach, DRAEM operates in a fully unsupervised setting, relying on synthetically generated anomalies to enable accurate per-pixel localization without requiring real anomaly samples.

3. Problem statement

We consider a *training set* $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ of n examples $\mathbf{x}_i \in [0,1]^D$ (w.l.o.g. we assume that each feature of the data belongs to the interval $[0,1]$) and a *ground-truth* $Y = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$ consisting of n *heatmaps* $\mathbf{y}_i \in \{0,1\}^D$. For each $\mathbf{x}_i \in X$, the relative ground-truth heatmap $\mathbf{y}_i \in Y$ expresses for each feature whether it explains the potential abnormality of $\mathbf{x}_i$ or not; in particular, we have that $\mathbf{y}_{i,j} = 0$ if the feature $\mathbf{x}_{i,j}$ is normal and $\mathbf{y}_{i,j} = 1$ if it is anomalous ($1 \leq j \leq D$). Clearly if $\mathbf{x}_i$ is an inlier, its heatmap $\mathbf{y}_i$ will be identically equal to 0. Let $\|\mathbf{y}_i\|_1$ denote the sum of the elements of the heatmap $\mathbf{y}_i$. We say that $\mathbf{x}_i$ is an anomaly (or an outlier) if $\|\mathbf{y}_i\|_1 > 0$ and that it is normal (or an inlier) otherwise. Let I (O, resp.) denote the subset $\{i \in \{1, \dots, n\} \mid \mathbf{x}_i \text{ is an inlier}\}$ ($\{i \in \{1, \dots, n\} \mid \mathbf{x}_i \text{ is an outlier}\}$, resp.) of the element indexes associated with inliers (outliers, resp.) in the training set X.

Given a test set $T = \{\mathbf{t}_1, \dots, \mathbf{t}_m\}$, the goal is to to provide an *heatmap* $\mathbf{h}_i$ that explains which are the feature that contribute most to the anomaly score value. The entries of the heatmap $\mathbf{h}_i$ can be binary, meaning that each feature value in $\mathbf{t}_i$ is considered normal if the corresponding element in $\mathbf{h}_i$ evaluates to 0 and anomalous if evaluates to 1, or continuous (typically between 0 and 1) containing the outlierness degree of each feature. As a side effect, an anomaly score $\mathcal{S}(\mathbf{t}_i) = \|\mathbf{h}_i\|$ representing the outlierness of the whole item $\mathbf{t}_i$ is assigned to each point $\mathbf{t}_i \in T$.

Theoretically, this scenario, such as the method described in the following, can be applied to any type of data. However, in this work we focus on data on the domain of the images, where an explanation of the outlierness of an item is represented by a *heatmap* encoded as a one-channel image of the same dimension of the input image that highlights anomalous portions of this item. The higher the values of the pixels in a certain area of a heatmap, the more the corresponding area of the input image explains its outlierness and contributes to its anomaly score. Thus, the dimension D of the data in the following will be equal to $H \times W \times C$ where $H \times W$ are the dimensions of the images and C is the number of channels.

4. Method

4.1. Loss Function

The idea at the basis of AE–XAD is to train the Autoencoder to perform a reconstruction $\tilde{\mathbf{x}}$ of the point $\mathbf{x}$ that is similar to the original item in the

Algorithm 1: AE–XAD

```
Input: Dataset $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, ground-truth heatmaps
       $Y = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$
Output: Heatmaps for the test set $T = \{\mathbf{t}_1, \dots, \mathbf{t}_m\}$
Perform data augmentation on the anomalies in $X$;
Train the Autoencoder on $X$ and $Y$ with the loss (2), oversampling
  anomalies in each batch;
foreach item $\mathbf{t} \in T$ do
    Compute the raw reconstruction vector $\mathbf{e} = (\mathbf{t} - \tilde{\mathbf{t}})^2$;
    Select the value $\hat{k}$ for the size of the filter;
    Compute the score $\mathcal{S}(\mathbf{t})$ using the filter $\mathcal{F}_{\hat{k}}$ in equation (4);
    Normalize $\mathbf{e}$ with equation (3), obtaining $\tilde{\mathbf{e}}$;
    Apply the Gaussian Filter $\mathcal{F}_{\hat{k}}$ and compute the heatmap by applying
      the filter $\mathbf{h} = \mathcal{F}_{\hat{k}}(\tilde{\mathbf{e}})$;
    Compute the mean $\mu_{\mathbf{h}}$ and the standard deviation $\sigma_{\mathbf{h}}$ in $\mathbf{h}$;
    Obtain the binary heatmap by applying the threshold $\mu_{\mathbf{h}} + \sigma_{\mathbf{h}}$ to $\mathbf{h}$;
```

normal features and as different from it as possible in the anomalous features. This is done by considering the following loss function applied to an object $\mathbf{x}$ with heatmap $\mathbf{y}$:

$$\ell(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{D} \left[(1 - \mathbf{y}_j) \frac{(\mathbf{x}_j - \tilde{\mathbf{x}}_j)^2}{(F(\mathbf{x}_j) - \mathbf{x}_j)^2} + \lambda_{\mathbf{y}} \mathbf{y}_j \frac{(F(\mathbf{x}_j) - \tilde{\mathbf{x}}_j)^2}{(F(\mathbf{x}_j) - \mathbf{x}_j)^2} \right] \tag{1}$$

where $F : [0, 1] \to \mathbb{R}$, and $\lambda_{\mathbf{y}} = \frac{D}{\|\mathbf{y}\|_1}$ if $\|\mathbf{y}\|_1 > 0$ and $\lambda_{\mathbf{y}} = 1$ otherwise. The whole training set loss is then computed as

$$\mathcal{L}(X, Y) = \frac{1}{|X|} \sum_{(\mathbf{x}, \mathbf{y}) \in X \times Y} \ell(\mathbf{x}, \mathbf{y}) \tag{2}$$

In Equation (1), when $\mathbf{y}_j = 0$ the contribution to the loss is given by $(\mathbf{x}_j - \tilde{\mathbf{x}}_j)^2$, while when $\mathbf{y}_j = 1$ the contribution is $(F(\mathbf{x}_j) - \tilde{\mathbf{x}}_j)^2$; this means that in the former case the AE is forced to reconstruct the feature $\mathbf{x}_j$ as is, while in the latter case it is forced to reconstruct it as $F(\mathbf{x}_j)$.

The function F is a hyperparameter of the network whose aim is to map the input into a point that is as distant as possible from it in order to enlarge the error. For example, a possible choice is $F_{-}(\mathbf{x}) = 1 - \mathbf{x}$, which maps $\mathbf{x}$ into its symmetric with respect to the central point $\frac{1}{2}$ and which represents, in the domain of the grayscale images, the negative of a pixel. An alternative to $F_{-}(\mathbf{x})$ is represented by the function $F_v(\mathbf{x}) = v$ with $v \in \mathbb{R} \setminus [0,1]$, which assigns to all the anomalous pixels a value v lying outside the pixel's values domain with the aim of enhancing the reconstruction error of anomalous pixels.

4.2. Network training

Anomalous samples are often under-represented in the training set X and, moreover, anomalies are heterogeneous, being quite dissimilar from one another. This fact may lead to a poor generalization; thus, to overcome this issue, we adopt a training strategy, similar to the one used in [57], based on the combination of *data augmentation* and *oversampling*. Specifically, each anomalous item $\mathbf{x}$ in the training set X is replicated 5 times as it is, and, additionally, 10 times more, the anomalous portion of the image is cut and pasted into a copy of a normal image after some standard geometric manipulations (zoom in/out, rotations, translations, etc.). Once the training set is augmented, for each batch B, the items in it are oversampled to ensure that we always have $\frac{1}{3}|B|$ anomalies and $\frac{2}{3}|B|$ inliers in it.

4.3. Inference and computation of the heatmaps

After the training, hopefully, the AE provides a reconstruction $\tilde{\mathbf{t}}$ of a point in the test set $\mathbf{t}$ that is similar to it in the normal features and different in the anomalous ones. Thus, if we consider the *reconstruction error vector* $\mathbf{e} = (\mathbf{t} - \tilde{\mathbf{t}})^2$, which is obtained as the squared difference between a point and its reconstruction (the exponential must be intended element-wise), this will be close to 0 in the normal features and far from 0 in the anomalous ones.

To make this error vector more comparable across different features, we consider a normalized reconstruction error vector $\tilde{\mathbf{e}}$, where each component is divided by the corresponding maximum possible reconstruction error

$$\tilde{\mathbf{e}} = \frac{\|\mathbf{t} - \tilde{\mathbf{t}}\|_2^2}{\|F(\mathbf{t}) - \mathbf{t}\|_2^2} \tag{3}$$

with the division performed element-wise. This normalization ensures that all error components lie in the range $[0,1]$, highlighting the relative significance of the deviations in each feature. As a result, $\tilde{\mathbf{e}}$ provides a clearer, scale-independent signal: values close to 0 indicate well-reconstructed (likely normal) features, while values approaching 1 flag poorly reconstructed (potentially anomalous) ones.

In practice, the normalized reconstruction error maps $\tilde{\mathbf{e}}$ can contain noise, which may obscure the true anomalous regions. To reduce this noise and enhance the visual clarity of the resulting heatmaps, we smooth $\tilde{\mathbf{e}}$ using a Gaussian filter $\mathcal{F}_k$ with size $(2k+1)\times(2k+1)$. This filter effectively blurs small fluctuations while preserving broader structures. However, the choice of the filter size parameter k is crucial: if k is too large, the filter may overly smooth the map, potentially eliminating small but important anomalous areas; if k is too small, the filter may fail to suppress spurious noisy peaks, resulting in a cluttered and misleading heatmap.

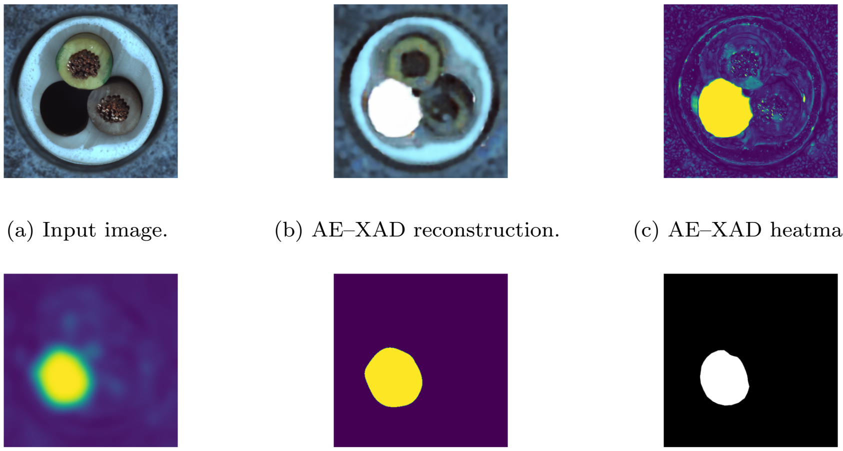
To avoid the need for manually tuning k in advance, we introduce a simple, automatic rule-of-thumb for selecting a suitable value. The procedure is as follows.

- Compute the mean $\mu_{\tilde{\mathbf{e}}}$ and standard deviation $\sigma_{\tilde{\mathbf{e}}}$ of the values in the normalized reconstruction error map.
- Binarize the map by setting all pixels above the threshold $\mu_{\tilde{\mathbf{e}}} + \sigma_{\tilde{\mathbf{e}}}$ to 1 (indicating potentially anomalous regions), and all others to 0.
- In the binarized map, determine the average lengths of the connected anomalous segments in both horizontal and vertical directions.
- Define the optimal filter size $\hat{k}$ as half the maximum of these two average lengths.

This adaptive value $\hat{k}$ can be interpreted as a generalized notion of radius that adapts not only to circular but also to irregularly shaped anomalous regions, enabling more effective smoothing tailored to the actual structure of the anomalies.

Based on this procedure, the final heatmap produced by AE–XAD is denoted as $\mathbf{h} = \mathcal{F}_{\hat{k}}(\tilde{\mathbf{e}})$, where $\mathcal{F}_{\hat{k}}$ represents the Gaussian filter applied using the parameter $\hat{k}$. This filtered heatmap $\mathbf{h}$ highlights regions with significant reconstruction error while suppressing noise, thus offering a clearer visual representation of potential anomalies.

If a binary segmentation of the anomalous regions is desired, $\mathbf{h}$ can be thresholded using a strategy similar to the one of the filter size selection.



(a) Input image.

(b) AE–XAD reconstruction.

(c) AE–XAD heatmap.

(d) Filter application.

(e) Binarized heatmap.

(f) Ground truth heatmap

Figure 1: Every step of AE–XAD algorithm.

Specifically, first, compute the mean $\mu_{\mathbf{h}}$ and standard deviation $\sigma_{\mathbf{h}}$ of the values in $\mathbf{h}$; then, binarize the map by setting all pixels above the threshold $\mu_{\mathbf{h}} + \sigma_{\mathbf{h}}$ to 1 (indicating potentially anomalous regions), and all others to 0.

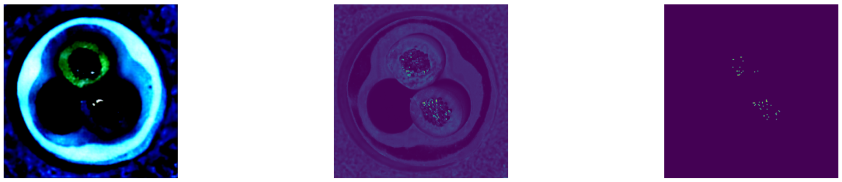
The reconstruction error vector $\mathbf{e}$ generated by AE–XAD during the heatmap computation of an image $\mathbf{t}$ encodes valuable information about how much this image deviates from the expected distribution of the test set T. A straightforward approach is to use its norm $\|\mathbf{e}\|$ as an anomaly score, where larger values indicate a higher likelihood that the image contains anomalous regions.

However, this baseline score is heavily influenced by the size of the anomalous area in the image. As a result, images with large anomalies are more easily detected, while images with small but meaningful anomalies may receive a lower score than normal images affected only by minor reconstruction noise.

To overcome this limitation, we define a new anomaly score:

$$\mathcal{S}(\mathbf{t}) = \|\mathbf{e} \cdot \mathcal{F}_k(\mathbf{e})\| \tag{4}$$

where $\mathcal{F}_k(\mathbf{e})$ denotes the same Gaussian-filtered version of $\mathbf{e}$ used for the heatmap $\mathbf{h}$. This formulation increases the relevance of pixels that are surrounded by other anomalous pixels, while reducing the impact of isolated



(a) AE reconstruction.

(b) AE heatmap.

(c) Binarized heatmap.

Figure 2: Every step of the standard AE application.

errors likely due to noise. In doing so, $\mathcal{S}(\mathbf{t})$ better captures the spatial consistency of anomalies, making it more robust to noise and more sensitive to small, structured anomalous regions that might otherwise be overlooked.

All the steps of AE–XAD are reported in Algorithm 1.

4.4. Motivating example.

In Figure 1 we can observe every step of the AE–XAD algorithm. The images are relative to the class "cable" of the MvTec dataset.

Specifically, Figure 1a reports an anomalous example $\mathbf{t}$ as it appears in the test set.

Figure 1b depicts the reconstruction $\tilde{\mathbf{t}}$ obtained by AE–XAD using the function $F(\mathbf{x}) = 1 - \mathbf{x}$. Here, we can observe that the pixels relative to the normal part of the image are well reconstructed, while the anomalous portion is reconstructed, according to F, with white pixels.

In Figure 1c is showed the heatmap obtained as the normalized reconstruction error image $\tilde{\mathbf{e}}$, which is almost everywhere equal to 0 except for the anomalous area, where the error is maximized, and a smaller areas all across the rest of the image where AE–XAD identifies what can be considered as anomalies of less relevance for the image.

With the application of the filter, Figure 1d, we obtain a heatmap that focuses on the main anomalous area of the image, giving less relevance to the details. In real applications, this representation may be more useful to a user since it better highlights the location of potential damages or disruptions.

Finally, Figure 1e shows the effect of the binarization. We can observe as this final output is really close to the ground-truth value (Figure 1f).

Figure 2 shows the reconstruction and the heatmaps obtained with a standard Autoencoder on the same input image. As we can see, while our method succeeds in identifying the anomalous portion of the image, the standard Autoencoder only detects a few pixels in that area, and the other considered anomalous pixels seem to be located around the contour of the object. This happens because for the standard Autoencoder, the reconstruction of the anomalous area is not much more difficult than the reconstruction of the rest of the image. Instead, with our method, the model becomes able to identify the presence of an anomalous area in the image, locate it, and subsequently reconstruct it worse than the normal area. The result is that, in this case, the reconstruction error is higher in those pixels close to the anomalous area than in the pixels that are far away from it.

5. Experimental results

This section investigates AE–XAD effectiveness by considering a twofold perspective, which analyses this paper's proposal under both quality and environmental impact lenses. To evaluate the contribution given by AE–XAD's loss, we take the standard autoencoder as the baseline used to carry out the ablation study. Additionally, we compare our proposal with two state-of-the-art methods, namely FCDD and BGAD, tackling the explainable anomaly detection task. The rest of this section is organised as follows. Section 5.1 describes the setup of the experiments. In Section 5.2, we perform a sensitivity analysis of its main hyperparameters and an ablation study investigating the impact of AE–XAD's component. Section 5.3 compares AE–XAD with the other methods involved in these analyses on real-world datasets. Section 5.4 provides an analysis of the environmental impact of AE–XAD and its competitors. Finally, in Section 5.5, the behaviour on the anomaly detection task is described.

5.1. Experimental setup

Datasets. Throughout the entire experimental campaign, we consider the following datasets.

- **MvTec**[2] is a dataset composed of 5354 high-resolution color images of 15 different object and texture classes. It contains both normal and anomalous images of examples with defects of different categories (on average, there are 5 anomalous categories for each class). Each anomaly

[2] https://www.mvtec.com/company/research/datasets/mvtec-ad

is annotated by means of a binary heatmap highlighting the area of the image where the defect of the sample is located.

- **WFDD**[3] (Woven Fabric Defect Detection) comprises 4,101 woven fabric images divided into four categories: grey cloth, grid cloth, yellow cloth, and pink flower. Each category features block-shaped, point-like, and line-type defects, all provided with pixel-level annotations.

- **BTAD**[4] (BeanTech Anomaly Detection) comprises 2,540 high-resolution RGB images of industrial products divided into three categories. Images vary in size and include both normal and defective samples. Defects cover a variety of surface and structural anomalies such as scratches, deformations, and material irregularities. Each defective image is provided with pixel-level ground truth masks.

- **Wood** data collection comes from *The Large Scale Image Dataset of Wood Surface Defects* [5], gathering large-scale images of wood surfaces containing many kinds of defects. The set of samples considered consists of 76 anomalies and 60 normal samples.

- **Hazelnut**[6] collects synthetic images created leveraging generative AI algorithms to generate defects on hazelnut stomium surfaces. Each image mimics real-world conditions and is equipped with metadata describing relevant generation-related data. Overall, the dataset contains 25 anomalous and 75 normal images.

- **Road Inspection** contains data that has been extracted from the crack class of the Road Defect Images[7] dataset, which is tailored to automate road crack detection. This collection comprises 17 anomalous images and 35 normal images.

MvTec, WFDD, and BTAD are all divided by design into a training set containing only normal items and a test set containing both normal and anomalous ones. To make them suitable for our setting, we include three

[3] https://www.kaggle.com/datasets/hodinhtrieu/the-woven-fabric-defect-detection-wfdd

[4] https://www.kaggle.com/datasets/thtuan/btad-beantech-anomaly-detection

[5] https://www.kaggle.com/datasets/nomihsa965/large-scale-image-dataset-of-wood-surface-defects

[6] https://www.kaggle.com/datasets/neurobotdata/hazelnut-cracking-synthetic-dataset

[7] https://www.kaggle.com/datasets/patelmihir/road-defects-nonaugmented

items of each anomaly category with their ground-truth heatmaps in the training set.

Moreover, for each dataset D, we also consider an additional extremely challenging scenario, in which no normal items are available in the training set, and it is composed only of anomalies. Throughout this section, we denote the results for this scenario as D^*.

Architecture and Training Details. AE–XAD is a method independent of the underlying architecture, meaning that the specific design of the employed Autoencoder can be adapted depending on the type of input data, available computational resources, training set size, and other practical considerations. In this paper, we adopt a particular Autoencoder architecture based on an asymmetric composition of Encoder and Decoder.

The Encoder is built by taking the first three residuals blocks of a ResNet [60] network pre-trained on ImageNet [61], providing a powerful and general-purpose feature extractor. To tailor its output for the downstream decoding step, we add an extra convolutional layer that reduces the number of channels from the ResNet's output. As a whole, this Encoder transforms RGB images of size $224 \times 224 \times 3$ into compact latent representations of size $28 \times 28 \times 64$.

Importantly, the ResNet module is kept frozen during training (i.e., its weights are not updated) to leverage robust features learned from large-scale datasets, and to reduce both memory and computational overhead, allowing the training to focus on the downstream convolutional layers that are specifically adapted to the anomaly detection task.

The Decoder consists of two parallel branches, each playing a distinct role in reconstructing the input.

- Branch 1 (non-trainable): This branch performs a simple upsampling of the latent representation from $28 \times 28 \times 64$ to $224 \times 224 \times 64$. Then, a *tanh* activation is applied, and the number of channels is reduced from 64 to 8 by summing groups of 8 channels (i.e., summing across each group of 8 out of 64 channels), resulting in a tensor b_1 of shape $224 \times 224 \times 8$.

- Branch 2 (trainable): This branch applies three convolutional blocks, each composed of a convolutional layer followed by a transposed convolutional layer (both using SeLU activation). These layers progressively reconstruct spatial structure and features, producing an output b_2 of shape $224 \times 224 \times 8$.

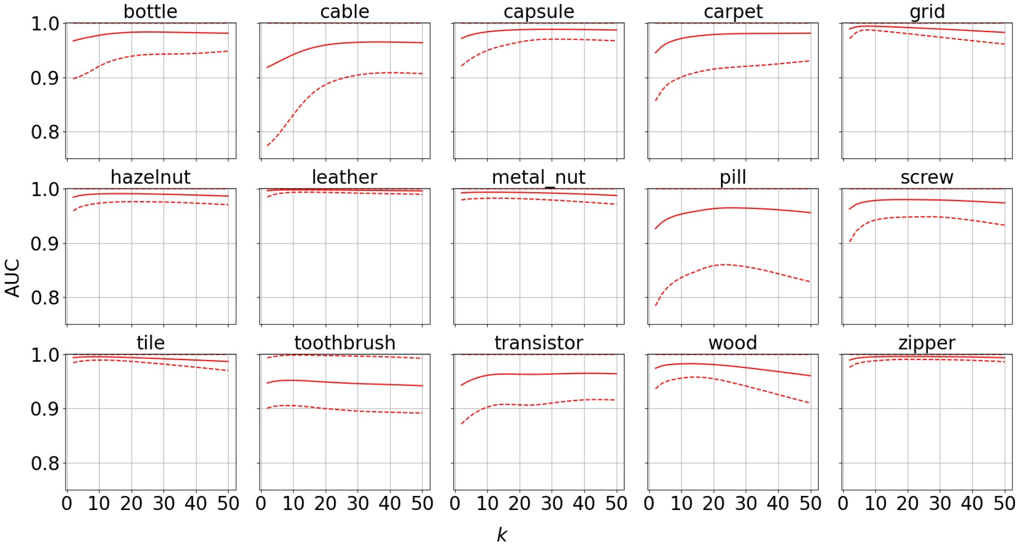


Figure 3: AUC of AE–XAD on the MvTec dataset as k increases.

The two branches are then combined via a pixel-wise operation $b_1 \cdot b_2 + b_2$ where b_1 acts as a soft mask that selectively emphasizes particular spatial regions of b_2, enhancing the reconstruction of important features. Finally, two additional convolutional layers refine and reshape the combined output to match the original input dimensions $224 \times 224 \times 3$.

In all our experiments AE–XAD, such as our competitor FCDD [56], is trained for 200 epochs. We set $F(\mathbf{x}) = v$ with $v = 2$ as transformation function. As for the optimisation function, we use Adam [62] with a learning rate set to 0.001 and a weight decay equal to 0.0001. As for FCDD [56] and BGAD [57], they counter many hyperparameters, whose values have been set accordingly to those suggested in the respective papers. All experiments have been performed on a Linux machine equipped with a 2.9 GHz Intel Core™ i7-10700, 32 GB of main memory, and a NVIDIA GeForce RTX 2070 Super having 8 GB of dedicated memory.

5.2. Ablation and sensitivity analysis

In the experiments illustrated in this section, we delve into the study of the components characterising AE–XAD's impact on its performances. We employ the MvTec dataset [63] in this evaluation, which is commonly used as a benchmark for the task at hand.

The first step of this investigation analyses how the adoption of different dimensions of the Gaussian filter affects AE–XAD's performance. To understand how different filter widths influence the behaviour of our method, we consider Gaussian filters of dimension $(2k + 1) \times (2k + 1)$ and standard deviation $\sigma = k/3$ (corresponding to the value correctly represent the normal distribution for a fixed k) and we measure the average AUC for the explanations of all the anomalies in the test set.

Figure 3 reports the trend for all the object types composing the MvTec dataset with $k \in [2, 30]$. A glance at the reported results reveals that the AUC trend resulting from applying Gaussian filters of different sizes varies across the MvTec object types. Indeed, when considering certain object categories (some examples are "grid", "hazelnut" and "leather"), smaller filters are preferable to use; differently, for other ones, applying wider filters results in improved AUCs. This non-uniformity in results may be somehow connected to the diversity in the structure and size of the anomalies potentially appearing in the considered objects. Our intuition is that small filters are therefore less effective in smoothing spurious spiky heatmap's pixels that may appear after the reconstruction and do not relate to an actual anomalous area. On the other hand, when dealing with small anomalies, too big filters spread the value of the anomalous spot detected all over the image, lowering the heatmap's delivery. Thus, to be effective even in this heterogeneous scenario, the Gaussian filter dimension should be chosen accordingly to the anomaly estimated size. This observation led us to introduce the procedure described in Section 4 for the automatic estimation of the suggested filter size based on the geometric properties of the detected anomalous regions.

5.2.1. Ablation study

We now move towards the assessment of the impact of each contribution introduced by this work. To evaluate the improvement due to AE–XAD's loss, we take as a baseline a "standard" AE, i.e. the AE trained using the mean squared error loss function to well-reconstruct normal samples. Additionally, to evaluate the contribution of the Gaussian-filtering based post-processing, we consider AUCs resulting from: (i) its application to the AE and (ii) AE–XAD raw heatmaps. We will refer to these two intermediate version as AE$\cup\mathcal{F}$ and AE–XAD$\setminus\mathcal{F}$, respectively. As for the filter application to AE outputs, the heatmaps values have been truncated to 1 to allow the application of the automatic radius estimation procedure.

Table 1, reporting the results collected in the four considered scenar-

Dataset	AE	AE $\cup \mathcal{F}$	AE–XAD $\setminus \mathcal{F}$	AE–XAD
bottle	0.340	0.267	*0.964*	**0.981**
cable	0.274	0.329	*0.913*	**0.934**
capsule	0.223	0.170	*0.963*	**0.985**
carpet	0.436	0.290	*0.932*	**0.971**
grid	0.469	0.468	*0.984*	**0.994**
hazelnut	0.840	0.921	*0.978*	**0.990**
leather	0.413	0.494	*0.995*	**0.998**
metal nut	0.748	0.741	*0.991*	**0.991**
pill	0.793	0.798	*0.913*	**0.948**
screw	0.140	0.126	*0.939*	**0.975**
tile	0.446	0.497	*0.992*	**0.995**
toothbrush	0.831	0.852	*0.943*	**0.953**
transistor	0.390	0.413	*0.935*	**0.965**
wood	0.516	0.571	*0.965*	**0.984**
zipper	0.320	0.197	*0.982*	**0.996**

Table 1: Mean AUC scored over all the object types belonging to the MvTec dataset. Best results are highlighted in bold, while runner-up is indicated by the italic font.

ios, shows how the filtering used to post-process explanations always improves results scored by the AE–XAD's raw heatmaps. The filter-based post-processing is not equally effective when applied to AE's heatmaps, it indeed does not bring improvements in many cases even worsening performances. AUCs scored by both the AE and AE $\cup \mathcal{F}$'s resulting explanations are furthermore considerably worse compared to those given by AE–XAD, witnessing the contribution of the AE–XAD's loss.

5.3. Comparison with competitors

Table 2 presents the results achieved by AE–XAD and its competitors in the standard scenario across all the considered datasets. These results are computed by averaging over all anomalous images in the test set of each dataset class.

As shown, our method attains the highest AUC value in more than half of the classes and ranks as the runner-up in nearly all the remaining ones. For the other two metrics, F1 and IoU, the trend differs: AE–XAD clearly emerges as the best-performing method, underscoring the exceptional quality of the binary heatmaps it produces.

Importantly, this observation holds regardless of the binarization threshold. To support this claim, Figures 4 and 5 present the F1 score and IoU values, respectively, across various thresholds of the form $\mu_{\mathbf{h}} + a \cdot \sigma_{\mathbf{h}}$, with a ranging from 1 to 3.5 in increments of 0.5. These figures demonstrate that AE–XAD consistently outperforms the competing methods at all eval-

		AE–XAD			FCDD			BGAD		
		AUC	F1	IoU	AUC	F1	IoU	AUC	F1	IoU
MvTec	bottle	*0.981*	**0.717**	**0.580**	0.966	*0.488*	*0.349*	**0.986**	0.000	0.000
	cable	0.934	**0.539**	**0.401**	*0.955*	0.272	0.197	**0.972**	*0.359*	*0.236*
	capsule	**0.985**	**0.362**	**0.234**	0.978	0.144	0.090	*0.984*	0.088	0.050
	carpet	0.971	**0.541**	**0.401**	*0.983*	*0.332*	*0.226*	**0.993**	0.289	0.184
	grid	**0.994**	**0.393**	**0.253**	0.970	0.141	0.080	*0.990*	0.147	0.082
	hazelnut	*0.990*	**0.597**	**0.436**	0.977	0.301	0.205	**0.993**	0.190	0.119
	leather	**0.998**	**0.388**	**0.252**	*0.995*	*0.192*	*0.119*	**0.998**	0.120	0.067
	metal nut	**0.991**	**0.766**	**0.637**	0.975	0.521	0.402	*0.980*	*0.707*	*0.586*
	pill	0.948	**0.483**	**0.359**	*0.979*	*0.260*	*0.182*	**0.991**	0.189	0.124
	screw	*0.975*	**0.185**	**0.111**	0.937	0.075	0.042	**0.980**	0.034	0.018
	tile	**0.995**	**0.768**	**0.647**	0.965	0.625	0.519	*0.977*	*0.694*	*0.572*
	toothbrush	**0.953**	**0.369**	**0.254**	0.946	0.181	0.115	0.928	0.117	0.067
	transistor	**0.965**	**0.601**	**0.448**	0.932	0.300	0.203	*0.957*	*0.360*	*0.264*
	wood	**0.984**	**0.620**	**0.469**	0.923	0.310	0.215	*0.984*	*0.454*	*0.315*
	zipper	**0.996**	**0.628**	**0.470**	0.983	*0.415*	*0.284*	*0.991*	0.002	0.001
WFDD	grey cloth	*0.982*	**0.300**	**0.191**	0.954	0.167	0.110	**0.985**	0.197	0.124
	grid cloth	*0.896*	*0.164*	*0.098*	0.790	0.049	0.028	**0.977**	**0.241**	**0.165**
	pink flower	**0.981**	**0.241**	**0.153**	0.962	*0.150*	*0.095*	*0.972*	0.128	0.082
	yellow cloth	*0.954*	*0.133*	*0.083*	0.938	0.100	0.064	**0.980**	**0.152**	**0.093**
BTAD	btad 01	**0.963**	**0.464**	**0.324**	0.740	0.214	*0.136*	*0.940*	0.000	0.000
	btad 02	0.798	**0.327**	**0.247**	*0.848*	0.262	*0.203*	**0.865**	*0.290*	0.193
	btad 03	**0.961**	**0.553**	**0.403**	*0.943*	*0.213*	*0.129*	0.939	0.000	0.000
	wood	0.927	**0.428**	**0.298**	*0.949*	*0.261*	*0.163*	**0.966**	0.239	0.147
	hazelnut	*0.985*	**0.500**	**0.339**	0.967	*0.270*	*0.163*	**0.986**	0.159	0.088
	road inspection	**0.978**	**0.348**	**0.211**	0.928	0.167	0.092	*0.960*	*0.306*	*0.182*

Table 2: Mean pixel-wise AUCs, F1 score, and Intersection over Union for the considered datasets in the standard setting. Best results are highlighted in bold, while runner-up is indicated by the italic font.

uated thresholds, as its curve is almost always positioned above those of the competitors

Table 2 presents the results for the three metrics aggregated across all images within each class. For a more fine-grained comparison at the individual image level, Figure 3 reports, for each dataset, a pairwise comparison between methods, showing the percentage of test images where one method outperforms the other. As observed, AE–XAD dominates in all cases except for the AUC comparison with BGAD on the WFDD dataset. This indicates that, for the vast majority of images, AE–XAD produces superior heatmaps. This is particularly evident for the binary heatmaps, as it outperforms both competitors in terms of F1 and IoU.

We also tested our algorithm in the particularly challenging scenario in which the training set is composed of (a few) anomalies. The results are reported in Table 4. The observations made in the standard setting are confirmed also in this case, and AE–XAD achieves the best AUC in the

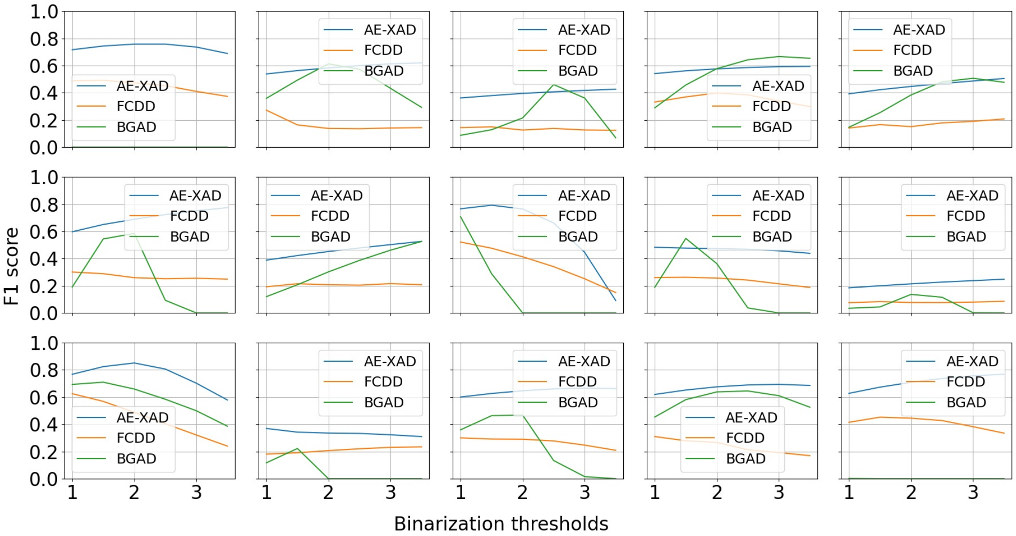


Figure 4: F1 score evolution over different binarization thresholds for all the MvTec object classes.

majority of the classes, and the best F1 and IoU in almost all the classes. The main difference, in this case, is that the method that behaves better, besides AE–XAD, is FCCD, while in the standard setting is BGAD, which means that BGAD is more able than FCDD to exploit information deriving from the distribution of normal data, but it cannot do without normal items in the training set. On the other hand, AE–XAD is able to perform well in both scenarios, demonstrating great adaptability.

5.4. Measurement of the environmental impact

Deep Learning models generally require extensive training, which results in significant energy consumption. It is therefore essential to track the energy usage of our proposed model, as well as that of the competing methods, to evaluate their environmental sustainability.

The CO_2 emissions are estimated using the Python library *CodeCarbon* [64], which computes its measurements based on power consumption and the geographic location where the code is executed.

The overall CO_2 emissions are reported in Table 5 for the standard scenario and in Table 5 for the scenario where only anomalies are included in the training set. As expected, emissions vary considerably across datasets (and

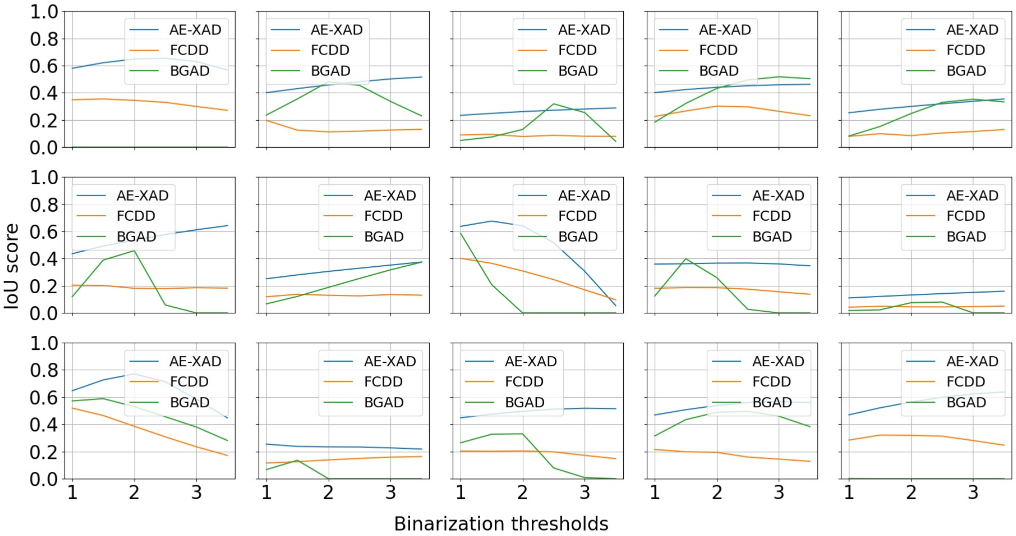


Figure 5: Intersection over Union evolution over different binarization thresholds for all the MvTec object classes.

are markedly lower in the only-anomalies scenario), primarily due to differences in dataset size. Among the methods, BGAD stands out as the largest contributor to CO_2 emissions, while FCDD exhibits the lowest environmental impact. AE–XAD shows an intermediate profile, producing approximately 50% more CO_2 than FCDD. This outcome is consistent with the models' architectures: whereas FCDD adopts a decoder-only design, our approach includes both an encoder and a decoder, resulting in roughly twice as many parameters as FCDD.

To better assess the trade-off between model accuracy and CO_2 emissions, we introduce a dedicated *ad-hoc* score. Given a specific performance metric m, the corresponding trade-off score is defined as:

$$\mathcal{T}(m) = \frac{\mathcal{E}}{\frac{m}{1-m}}, \tag{5}$$

where $\mathcal{E}$ denotes the amount of CO_2 emitted during training and evaluation. The denominator $\frac{m}{1-m}$ expresses the ratio of the achieved performance m relative to the remaining margin for improvement $(1 - m)$. For low values of m, the ratio $\frac{m}{1-m}$ is small, resulting in a larger trade-off score $\mathcal{T}(m)$, meaning more CO_2 is "spent" per unit of progress. As m approaches 1, the denominator grows rapidly, reflecting the fact that improvements near

DS	Method	Metric								
		AUC			F1			IoU		
		AE–XAD	BGAD	FCDD	AE–XAD	BGAD	FCDD	AE–XAD	BGAD	FCDD
MvTec	AE–XAD	–	**0.803**	**0.663**	–	**0.864**	**0.843**	–	**0.864**	**0.843**
	BGAD	0.197	–	0.362	0.090	–	**0.543**	0.090	–	**0.543**
	FCDD	0.337	**0.638**	–	0.157	0.442	–	0.157	0.442	–
WFDD	AE–XAD	–	**0.789**	0.455	–	**0.677**	**0.582**	–	**0.672**	**0.575**
	BGAD	0.211	–	0.221	0.210	–	0.362	0.215	–	0.367
	FCDD	**0.545**	**0.779**	–	0.408	**0.623**	–	0.415	**0.619**	–
BTAD	AE–XAD	–	**0.736**	**0.665**	–	**0.730**	**0.761**	–	**0.730**	**0.761**
	BGAD	0.264	–	0.416	0.063	–	**0.578**	0.063	–	**0.578**
	FCDD	0.335	**0.584**	–	0.099	0.143	–	0.099	0.143	–
wood	AE–XAD	–	**0.560**	**0.600**	–	**0.840**	**0.840**	–	**0.840**	**0.840**
	BGAD	0.440	–	**0.560**	0.160	–	**0.640**	0.160	–	**0.640**
	FCDD	0.400	0.440	–	0.160	0.360	–	0.160	0.360	–
hazelnut	AE–XAD	–	**1.000**	**0.533**	–	**1.000**	**1.000**	–	**1.000**	**1.000**
	BGAD	0.000	–	0.067	0.000	–	**0.867**	0.000	–	**0.867**
	FCDD	0.467	**0.933**	–	0.000	0.133	–	0.000	0.133	–
road	AE–XAD	–	**0.857**	**1.000**	–	**1.000**	**0.571**	–	**1.000**	**0.571**
	BGAD	0.143	–	0.143	0.000	–	0.000	0.000	–	0.000
	FCDD	0.000	**0.857**	–	0.429	**1.000**	–	0.429	**1.000**	–

Table 3: Probability of winning for each pair of methods. Each value represents the percentage of images on which the row method outperforms the column method. In bold are reported the values greater than 0.5.

perfect accuracy are harder to achieve and should be weighted more heavily.

In other words, the trade-off score $\mathcal{T}(m)$ quantifies the amount of CO_2 required to obtain a single percentage point of the metric m, while prioritizing improvements in the high-performance regime (i.e., when m is close to 1). A lower value of $\mathcal{T}(m)$ indicates a more environmentally efficient model for the given level of accuracy.

The trade-off results (considering all three evaluation metrics) are reported in Table 7 for the standard scenario and in Table 8 for the anomaly-only scenario.

As shown, AE–XAD consistently achieves the best results across almost all datasets in both scenarios. This highlights not only its strong detection performance but also its *green-awareness*, as it provides the most favorable compromise between accuracy and environmental impact. In other words, AE–XAD is able to reach top-level performance while minimizing its CO_2 footprint, demonstrating a design that is both effective and environmentally responsible.

		AE–XAD			FCDD			BGAD		
		AUC	F1	IoU	AUC	F1	IoU	AUC	F1	IoU
MvTec*	bottle	*0.777*	*0.208*	*0.122*	**0.939**	**0.411**	**0.276**	0.703	0.179	0.101
	cable	*0.936*	**0.537**	**0.407**	**0.938**	*0.382*	*0.278*	0.796	0.201	0.116
	capsule	0.914	**0.216**	**0.136**	**0.967**	*0.119*	*0.073*	*0.942*	0.082	0.047
	carpet	**0.985**	**0.508**	**0.375**	*0.985*	*0.371*	*0.255*	0.929	0.186	0.109
	grid	**0.987**	**0.381**	**0.248**	*0.975*	*0.165*	*0.098*	0.745	0.089	0.049
	hazelnut	**0.992**	**0.571**	**0.412**	*0.983*	*0.347*	*0.233*	0.977	0.192	0.117
	leather	**0.997**	**0.365**	**0.235**	*0.995*	*0.220*	*0.137*	0.989	0.108	0.060
	metal nut	**0.948**	**0.487**	**0.391**	*0.944*	*0.387*	*0.257*	0.813	0.177	0.102
	pill	**0.989**	**0.490**	**0.353**	0.894	*0.138*	*0.090*	*0.932*	0.132	0.081
	screw	0.884	**0.072**	**0.039**	**0.958**	*0.048*	*0.025*	*0.950*	0.037	0.019
	tile	**0.991**	**0.774**	**0.660**	*0.976*	*0.655*	*0.541*	0.755	0.293	0.181
	toothbrush	*0.910*	**0.211**	**0.133**	**0.948**	*0.203*	*0.130*	0.839	0.091	0.051
	transistor	**0.952**	**0.559**	**0.413**	*0.858*	*0.240*	*0.159*	0.728	0.174	0.102
	wood	**0.962**	**0.544**	**0.404**	*0.933*	*0.328*	*0.219*	0.920	0.326	0.205
	zipper	**0.994**	**0.632**	**0.477**	*0.989*	*0.436*	*0.293*	0.901	0.239	0.140
WFDD*	grey cloth	0.934	**0.298**	**0.199**	**0.966**	*0.177*	*0.116*	*0.939*	0.148	0.092
	grid cloth	**0.698**	**0.064**	**0.036**	0.622	0.016	0.009	*0.629*	*0.039*	*0.021*
	pink flower	*0.921*	0.124	0.084	**0.980**	**0.170**	**0.117**	0.871	0.079	0.046
	yellow cloth	0.823	*0.096*	**0.065**	**0.904**	0.071	0.048	*0.893*	**0.096**	*0.054*
BTAD*	btad 01	**0.892**	**0.278**	**0.178**	*0.755*	*0.206*	*0.131*	0.649	0.054	0.028
	btad 02	**0.844**	**0.319**	**0.246**	0.587	0.163	0.110	*0.756*	*0.199*	*0.132*
	btad 03	*0.780*	*0.173*	*0.106*	**0.912**	**0.235**	**0.144**	0.686	0.067	0.035
	wood*	*0.939*	**0.454**	**0.330**	**0.949**	*0.297*	*0.192*	0.901	0.218	0.131
	hazelnut*	**0.979**	**0.561**	**0.401**	*0.967*	*0.316*	*0.198*	0.942	0.164	0.091
	road inspection*	**0.987**	**0.445**	**0.290**	0.871	0.127	0.070	*0.908*	*0.218*	*0.124*

Table 4: Mean pixel-wise AUCs, F1 score, and Intersection over Union for the considered datasets in the outliers only setting. Best results are highlighted in bold, while runner-up is indicated by the italic font.

5.5. Detecting anomalous images

Finally, we report the results of the anomaly detection task, which in our context can be regarded as a secondary objective. In this setting, the goal is to identify test images containing anomalies. As shown in Table 9, AE–XAD demonstrates strong capabilities even in this accessorial task, despite it not being its primary focus. Remarkably, it achieves the highest AUC in more than half of the evaluated datasets, further confirming its versatility and robustness.

6. Conclusion

In this work, we have introduced AE–XAD, a reconstruction error-based approach designed for the challenging task of anomaly explanation. Our method is built upon a novel loss function that leverages the knowledge of the heatmaps of anomalous examples available in the training set. By

		AE–XAD	FCDD	BGAD
MvTec	bottle	*5.394*	**3.391**	10.363
	cable	*6.335*	**4.129**	12.245
	capsule	*6.074*	**4.088**	11.460
	carpet	*8.027*	**4.995**	14.467
	grid	*6.390*	**4.116**	13.463
	hazelnut	*11.066*	**6.606**	19.421
	leather	**0.050**	*4.247*	12.744
	metal nut	*6.033*	**3.961**	11.184
	pill	*7.237*	**4.811**	14.443
	screw	*7.305*	**4.604**	15.684
	tile	*6.595*	**4.088**	12.103
	toothbrush	*1.513*	**1.030**	2.663
	transistor	*5.928*	**3.679**	11.099
	wood	*6.818*	**4.281**	12.734
	zipper	**0.042**	*3.802*	13.019
WFDD	grey cloth	*7.722*	**5.180**	14.041
	grid cloth	*72.991*	**34.752**	101.350
	pink flower	*8.123*	**4.707**	13.972
	yellow cloth	*25.474*	**15.461**	44.937
BTAD	btad 01	*10.990*	**6.880**	19.012
	btad 02	*11.408*	**6.256**	19.286
	btad 03	65.806	**15.426**	*48.362*
	wood	*2.272*	**1.528**	4.399
	hazelnut	*1.386*	**0.883**	2.668
	road inspection	*0.756*	**0.422**	1.058

Table 5: Emissions (in grams of CO_2) on standard scenario. Best results are highlighted in bold, while runner-up is indicated by the italic font.

explicitly exploiting this information, AE–XAD is able to accurately highlight the regions of the input data that most strongly contribute to their anomalous nature, thereby offering more interpretable and reliable explanations.

A thorough experimental campaign, conducted across a wide range of datasets and scenarios, has demonstrated the robustness and versatility of AE–XAD. The proposed method consistently achieves high performance in terms of multiple evaluation metrics, outperforming or closely matching competitive baselines in most settings. Furthermore, AE–XAD distinguishes itself for its environmental sustainability: it maintains low CO_2 emissions when compared to competing methods, striking a favorable balance between effectiveness and computational efficiency.

Overall, AE–XAD provides a principled and environmentally conscious solution for anomaly explanation, capable of combining accurate detection with interpretable heatmaps. Future work will explore its extension to more complex data modalities and real-time scenarios, further broadening its applicability in sustainable AI-driven anomaly analysis.

		AE–XAD	FCDD	BGAD
MvTec*	bottle	*0.214*	**0.182**	0.677
	cable	*0.547*	**0.435**	1.345
	capsule	*0.351*	**0.264**	0.982
	carpet	*0.357*	**0.265**	1.000
	grid	*0.299*	**0.213**	0.961
	hazelnut	*0.286*	**0.213**	0.835
	leather	*0.364*	**0.252**	0.984
	metal nut	*0.285*	**0.221**	0.850
	pill	*0.503*	**0.374**	1.273
	screw	*0.299*	**0.226**	1.030
	tile	*0.350*	**0.264**	1.000
	toothbrush	*0.084*	**0.075**	0.374
	transistor	*0.273*	**0.207**	0.842
	wood	*0.356*	**0.247**	0.990
	zipper	*0.417*	**0.360**	1.267
WFDD*	grey cloth	*0.291*	**0.212**	0.810
	grid cloth	*0.089*	**0.075**	0.397
	pink flower	*0.086*	**0.072**	0.387
	yellow cloth	*0.091*	**0.087**	0.420
BTAD*	btad 01	**0.101**	*0.127*	0.414
	btad 02	*0.101*	**0.096**	0.547
	btad 03	**0.092**	*0.096*	0.458
wood*		*1.192*	**0.903**	2.386
hazelnut*		*0.226*	**0.197**	0.602
road inspection*		*0.246*	**0.205**	0.655

Table 6: Emissions (in grams of CO_2) on anomalies only scenario. Best results are highlighted in bold, while runner-up is indicated by the italic font.

Acknowledgments.

We acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), Spoke 9 - Green-aware AI, under the NRRP MUR program funded by the NextGenerationEU.

References

[1] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. 51 (5) (2019) 93:1–93:42.

[2] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, K.-R. Müller, Explaining deep neural networks and beyond: A review of methods and applications, Proceedings of the IEEE 109 (3) (2021) 247–278.

[3] Z. Li, Y. Zhu, M. van Leeuwen, A survey on explainable anomaly detection, CoRR abs/2210.06959 (2022). arXiv:2210.06959, doi:10.48550/arXiv.2210.06959.

		AE–XAD			FCDD			BGAD		
		AUC	F1	IoU	AUC	F1	IoU	AUC	F1	IoU
MvTec	bottle	**0.1020**	**2.1313**	**3.9004**	*0.1194*	*3.5598*	*6.3251*	0.1421	inf	inf
	cable	0.4476	**5.4260**	**9.4727**	**0.1929**	*11.043*	*16.867*	*0.3532*	21.821	39.583
	capsule	*0.0933*	**10.694**	**19.839**	**0.0907**	24.281	41.331	0.1870	119.41	219.06
	carpet	0.2373	**6.8032**	**11.971**	**0.0845**	*10.048*	*17.104*	*0.1064*	35.512	64.015
	grid	**0.0401**	**9.8885**	**18.832**	*0.1267*	25.051	47.515	0.1325	78.384	150.37
	hazelnut	**0.1142**	**7.4604**	**14.341**	0.1566	15.345	25.696	*0.1426*	82.595	143.34
	leather	**0.0001**	**0.0782**	**0.1478**	0.0220	17.826	31.465	0.0216	93.544	177.82
	metal nut	**0.0573**	**1.8411**	**3.4383**	*0.1016*	*3.6366*	*5.8965*	0.2325	4.6255	7.9170
	pill	0.3974	**7.7467**	**12.904**	**0.1035**	*13.672*	*21.684*	*0.1264*	62.071	101.69
	screw	**0.1838**	**32.260**	**58.774**	*0.3084*	56.700	*103.79*	0.3203	440.50	876.86
	tile	**0.0338**	**1.9921**	**3.5956**	*0.1485*	*2.4479*	*3.7816*	0.2854	5.3473	9.0541
	toothbrush	0.0744	**2.5821**	**4.4347**	**0.0583**	4.6623	7.8913	0.2070	20.037	36.858
	transistor	**0.2137**	**3.9365**	**7.2974**	*0.2665*	*8.5688*	*14.452*	0.4964	19.753	30.901
	wood	**0.1115**	**4.1819**	**7.7207**	0.3586	*9.5212*	*15.639*	*0.2134*	15.326	27.690
	zipper	**0.0002**	**0.0248**	**0.0472**	0.0650	5.3597	9.5758	0.1154	7508.1	13887.
WFDD	grey cloth	**0.1409**	**18.057**	**32.633**	0.2373	*25.630*	*42.452*	*0.2123*	57.922	100.58
	grid cloth	*8.4842*	*371.73*	*673.28*	10.832	786.80	1375.9	**2.3156**	**320.22**	**524.06**
	pink flower	*0.1562*	*25.632*	*44.982*	**0.1421**	**24.779**	**40.925**	0.3984	95.916	157.455
	yellow cloth	1.2313	*166.26*	*281.74*	*1.1544*	**153.06**	**240.14**	**0.9526**	256.29	448.54
BTAD	btad 01	**0.4175**	**12.683**	**22.967**	2.4196	*25.300*	*43.776*	*1.2167*	inf	inf
	btad 02	*2.8945*	*23.435*	*34.773*	**1.1212**	**17.583**	**24.489**	3.0075	47.262	80.875
	btad 03	*2.6691*	**53.286**	**97.364**	**0.9384**	*57.103*	*104.58*	3.1576	inf	inf
	wood	0.1795	**3.0355**	**5.3637**	**0.0816**	*4.3182*	*7.8314*	*0.1555*	13.978	25.457
	hazelnut	**0.0215**	**1.3862**	**2.7075**	*0.0298*	*2.3856*	*4.5311*	0.0374	14.162	27.652
	road inspection	**0.0167**	**1.4179**	**2.8177**	*0.0327*	*2.1017*	*4.1783*	0.0441	2.4035	4.7577

Table 7: Trade-off score (Equation (5)) for standard scenario.

[4] F. Angiulli, F. Fassetti, S. Nisticò, L. Palopoli, Outlier explanation through masking models, in: Advances in Databases and Information Systems: European Conference, 2022, pp. 392–406.

[5] F. Angiulli, F. Fassetti, L. Ferragina, Reconstruction error-based anomaly detection with few outlying examples (2023). `arXiv:2305.10464`.
URL `https://arxiv.org/abs/2305.10464`

[6] M. T. Ribeiro, S. Singh, C. Guestrin, " why should i trust you?" explaining the predictions of any classifier, in: Int. Conf. ACM SIGKDD, 2016, pp. 1135–1144.

[7] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, Advances in neural information processing systems 30 (2017).

[8] F. Angiulli, F. Fassetti, S. Nisticò, Local interpretable classifier explanations with self-generated semantic features, in: Int. Conf. on Discovery Science, 2021, pp. 401–410.

		AE–XAD			FCDD			BGAD		
		AUC	F1	IoU	AUC	F1	IoU	AUC	F1	IoU
MvTec*	bottle	*0.0611*	*0.8139*	*1.5377*	**0.0119**	**0.2613**	**0.4774**	0.2865	3.1109	6.0090
	cable	0.0376	*0.4721*	*0.7969*	*0.0288*	0.7057	1.1305	0.3453	5.3503	10.274
	capsule	*0.0333*	**1.2742**	**2.2324**	**0.0089**	*1.9444*	*3.3401*	0.0610	10.927	20.091
	carpet	*0.0054*	**0.3458**	**0.5953**	**0.0041**	*0.4507*	*0.7752*	0.0767	4.3712	8.1977
	grid	**0.0039**	**0.4857**	**0.9096**	*0.0055*	*1.0776*	*1.9726*	0.3286	9.8207	18.472
	hazelnut	**0.0022**	**0.2149**	**0.4079**	*0.0038*	*0.4010*	*0.7029*	0.0199	3.5096	6.3206
	leather	**0.0009**	**0.6335**	**1.1839**	*0.0013*	*0.8939*	*1.5813*	0.0110	8.1615	15.552
	metal nut	*0.0156*	**0.3009**	**0.4448**	**0.0131**	*0.3495*	*0.6391*	0.1949	3.9512	7.4549
	pill	**0.0058**	**0.5229**	**0.9218**	*0.0442*	*2.3320*	*3.7746*	0.0931	8.3937	14.373
	screw	*0.0392*	**3.8694**	**7.3379**	**0.0099**	*4.4994*	*8.8909*	0.0546	26.8468	53.420
	tile	**0.0032**	**0.1022**	**0.1798**	*0.0066*	*0.1393*	*0.2238*	0.3248	2.4110	4.5292
	toothbrush	*0.0083*	*0.3126*	*0.5466*	**0.0041**	**0.2952**	**0.5031**	0.0715	3.7267	7.0242
	transistor	**0.0138**	**0.2152**	**0.3869**	*0.0341*	*0.6535*	*1.0898*	0.3151	4.0077	7.3782
	wood	**0.0141**	**0.2978**	*0.5244*	*0.0176*	*0.5061*	0.8800	0.0856	2.0431	3.8380
	zipper	**0.0027**	**0.2427**	**0.4570**	*0.0040*	*0.4654*	*0.8671*	0.1397	4.0325	7.7885
WFDD*	grey cloth	*0.0205*	**0.6860**	**1.1750**	**0.0074**	*0.9867*	*1.6162*	0.0529	4.6706	7.9600
	grid cloth	**0.0384**	**1.2993**	**2.3677**	*0.0453*	*4.5546*	*8.6614*	0.2348	9.8376	18.443
	pink flower	*0.0074*	*0.6098*	*0.9444*	**0.0014**	**0.3494**	**0.5395**	0.0572	4.5334	8.0701
	yellow cloth	*0.0196*	**0.8579**	**1.3155**	**0.0092**	*1.1396*	*1.7235*	0.0502	3.9421	7.3640
BTAD*	btad 01	**0.0122**	**0.2613**	**0.4643**	*0.0411*	*0.4882*	*0.8431*	0.2243	7.3072	14.361
	btad 02	**0.0187**	**0.2159**	**0.3107**	*0.0676*	*0.4932*	*0.7763*	0.1762	2.2036	3.5824
	btad 03	*0.0259*	*0.4400*	*0.7764*	**0.0092**	**0.3113**	**0.5676**	0.2093	6.4143	12.582
	wood*	*0.0755*	**1.4632**	**2.5348**	**0.0487**	*2.1395*	*3.8139*	0.2619	8.5810	15.814
	hazelnut*	**0.0042**	**0.1999**	**0.3831**	*0.0067*	*0.4276*	*0.8006*	0.0374	3.0720	5.9975
	road inspection*	**0.0023**	**0.3079**	**0.6074**	*0.0304*	*1.4149*	*2.7418*	0.0660	2.3437	4.6261

Table 8: Trade-off score (Equation (5)) for anomalies only scenario.

[9] S. Verma, A. Beniwal, N. Sadagopan, A. Seshadri, Recxplainer: Post-hoc attribute-based explanations for recommender systems, in: Progress and Challenges in Building Trustworthy Embodied AI.

[10] G. Peake, J. Wang, Explanation mining: Post hoc interpretability of latent factor models for recommendation systems, in: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, 2018, pp. 2060–2069.

[11] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, J. K. Su, This looks like that: deep learning for interpretable image recognition, Advances in neural information processing systems 32 (2019).

[12] C. Nóbrega, L. Marinho, Towards explaining recommendations through local surrogate models, in: Proceedings of the 34th ACM/SIGAPP symposium on applied computing, 2019, pp. 1671–1678.

[13] T. Chowdhury, R. Rahimi, J. Allan, Equi-explanation maps: concise and informative global summary explanations, in: Proceedings of the

		AE–XAD	FCDD	BGAD
MvTec	bottle	**1.000**	*0.994*	**1.000**
	cable	0.931	**0.976**	*0.964*
	capsule	**0.964**	*0.959*	0.926
	carpet	*0.978*	0.942	**1.000**
	grid	**0.994**	0.961	*0.971*
	hazelnut	**1.000**	**1.000**	*0.994*
	leather	**1.000**	**1.000**	**1.000**
	metal nut	**0.999**	0.983	*0.983*
	pill	0.955	**0.973**	*0.968*
	screw	**0.836**	*0.826*	0.805
	tile	**1.000**	**1.000**	*0.996*
	toothbrush	*0.855*	**0.954**	0.833
	transistor	**1.000**	*0.992*	0.987
	wood	*0.998*	0.994	**1.000**
	zipper	**1.000**	*0.999*	0.996
WFDD	grey cloth	*0.997*	0.958	**1.000**
	grid cloth	**0.991**	0.855	*0.967*
	pink flower	*0.886*	**0.992**	0.842
	yellow cloth	0.915	*0.996*	**0.998**
BTAD	btad 01	*0.972*	0.952	**0.987**
	btad 02	*0.816*	0.807	**0.858**
	btad 03	**1.000**	*0.993*	0.983
	wood	0.936	**0.960**	*0.956*
	hazelnut	**1.000**	*0.987*	**1.000**
	road inspection	*0.836*	**0.936**	0.364

Table 9: Detection AUC

2022 ACM Conference on Fairness, Accountability, and Transparency, 2022, pp. 464–472.

[14] Y. Ming, H. Qu, E. Bertini, Rulematrix: Visualizing and understanding classifiers with rules, IEEE transactions on visualization and computer graphics 25 (1) (2018) 342–352.

[15] D. Ley, S. Mishra, D. Magazzeni, Global counterfactual explanations: Investigations, implementations and improvements, in: ICLR 2022 Workshop on PAIR {\textasciicircum} 2Struct: Privacy, Accountability, Interpretability, Robustness, Reasoning on Structured Data.

[16] S. Tan, M. Soloviev, G. Hooker, M. T. Wells, Tree space prototypes: Another look at making tree ensembles interpretable, in: Proceedings of the 2020 ACM-IMS on foundations of data science conference, 2020, pp. 23–34.

[17] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra,

Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE ICCV, 2017, pp. 618–626.

[18] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PloS one 10 (7) (2015) e0130140.

[19] R. Guidotti, A. Monreale, F. Spinnato, D. Pedreschi, F. Giannotti, Explaining any time series classifier, in: 2020 IEEE second international conference on cognitive machine intelligence (CogMI), IEEE, 2020, pp. 167–176.

[20] A. Shrikumar, P. Greenside, A. Kundaje, Learning important features through propagating activation differences, in: International conference on machine learning, PMlR, 2017, pp. 3145–3153.

[21] F. Angiulli, F. De Luca, F. Fassetti, S. Nisticó, Large language models-based local explanations of text classifiers, in: International Conference on Discovery Science, Springer, 2024, pp. 19–35.

[22] J. Enouen, H. Nakhost, S. Ebrahimi, S. Arik, Y. Liu, T. Pfister, Textgenshap: Scalable post-hoc explanations in text generation with long documents, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Findings of the Association for Computational Linguistics: ACL 2024, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 13984–14011. `doi:10.18653/v1/2024.findings-acl.832`.
URL `https://aclanthology.org/2024.findings-acl.832/`

[23] B. Barr, N. Fatsi, L. Hancox-Li, P. Richter, D. Proano, C. Mok, The disagreement problem in faithfulness metrics, arXiv preprint arXiv:2311.07763 (2023).

[24] R. Li, Q. Li, Y. Zhang, D. Zhao, Y. Jiang, Y. Yang, Interpreting unsupervised anomaly detection in security via rule extraction, Advances in Neural Information Processing Systems 36 (2023) 62224–62243.

[25] L. Kong, A. Huet, D. Rossi, M. Sozio, Tree-based kendall's τ maximization for explainable unsupervised anomaly detection, in: ICDM, 2023, pp. 1073–1078.
URL `https://doi.org/10.1109/ICDM58522.2023.00126`

[26] S. Das, M. R. Islam, N. K. Jayakodi, J. R. Doppa, Active anomaly detection via ensembles, arXiv preprint arXiv:1809.06477 (2018).

[27] N. Liu, D. Shin, X. Hu, Contextual outlier interpretation, arXiv preprint arXiv:1711.10589 (2017).

[28] T. Pevnỳ, M. Kopp, Explaining anomalies with sapling random forests, in: Information Technologies-Applications and Theory Workshops, Posters, and Tutorials (ITAT 2014), 2014, p. 7.

[29] M. Kopp, T. Pevnỳ, M. Holena, Interpreting and clustering outliers with sapling random forests, Information Technologies—Applications and Theory (2014).

[30] M. Macha, L. Akoglu, Explaining anomalies in groups with characterizing subspace rules, Data Mining and Knowledge Discovery 32 (5) (2018) 1444–1480.

[31] F. Angiulli, F. Fassetti, S. Nisticò, L. Palopoli, Explaining outliers and anomalous groups via subspace density contrastive loss, Machine Learning 113 (10) (2024) 7565–7589.

[32] F. T. Liu, K. M. Ting, Z.-H. Zhou, Isolation forest, in: 2008 eighth ieee international conference on data mining, IEEE, 2008, pp. 413–422.

[33] D. Samariya, S. Aryal, K. M. Ting, J. Ma, A new effective and efficient measure for outlying aspect mining, in: International Conference on Web Information Systems Engineering, Springer, 2020, pp. 463–474.

[34] V. Mnih, N. Heess, A. Graves, et al., Recurrent models of visual attention, Advances in neural information processing systems 27 (2014).

[35] X. Li, X. Song, T. Wu, Aognets: Compositional grammatical architectures for deep learning, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6220–6230.

[36] Q. Zhang, Y. N. Wu, S.-C. Zhu, Interpretable convolutional neural networks, in: IEEE conference on computer vision and pattern recognition, 2018, pp. 8827–8836.

[37] Z. Chen, Y. Bei, C. Rudin, Concept whitening for interpretable image recognition, Nature Machine Intelligence 2 (12) (2020) 772–782.

[38] D. Alvarez-Melis, T. S. Jaakkola, Towards robust interpretability with self-explaining neural networks, in: Advances in Neural Information Processing Systems, 2018, pp. 7786–7795.

[39] I. Rio-Torto, K. Fernandes, L. F. Teixeira, Understanding the decisions of cnns: An in-model approach, Pattern Recognition Letters 133 (2020) 373–380.

[40] J. Parekh, P. Mozharovskyi, F. d'Alché Buc, A framework to learn with interpretation, Advances in Neural Information Processing Systems 34 (2021).

[41] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, K. Müller, A unifying review of deep and shallow anomaly detection, Proc. IEEE 109 (5) (2021) 756–795.

[42] G. Pang, C. Shen, L. Cao, A. van den Hengel, Deep learning for anomaly detection: A review, CoRR abs/2007.02500 (2020). `arXiv:2007.02500`. URL `https://arxiv.org/abs/2007.02500`

[43] M. A. Kramer, Nonlinear principal component analysis using autoassociative neural networks, AIChE Journal 37 (2) (1991) 233–243.

[44] R. Hecht-Nielsen, Replicator neural networks for universal optimal source coding, Science 269 (5232) (1995) 1860—1863.

[45] S. Hawkins, H. He, G. Williams, R. Baxter, Outlier detection using replicator neural networks, in: Int. Conf. (DAWAK), 2002, pp. 170–180.

[46] J. An, S. Cho, Variational autoencoder based anomaly detection using reconstruction probability, Tech. Rep. 3, SNU Data Mining Center (2015).

[47] F. Angiulli, F. Fassetti, L. Ferragina, Improving deep unsupervised anomaly detection by exploiting VAE latent space distribution, in: Int. Conf. on Discovery Science, 2020, pp. 596–611.

[48] F. Angiulli, F. Fassetti, L. Ferragina, Latent*Out*: an unsupervised deep anomaly detection approach exploiting latent space distribution, Mac. Lear. (2022) 1–27.

[49] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, G. Langs, Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, in: Int. Conf. IPMI, Vol. 10265, 2017, pp. 146–157.

[50] S. Akcay, A. Atapour-Abarghouei, T. P. Breckon, Ganomaly: Semi-supervised anomaly detection via adversarial training (2018). arXiv:1805.06725.

[51] Y. Liu, Z. Li, C. Zhou, Y. Jiang, J. Sun, M. Wang, X. He, Generative adversarial active learning for unsupervised outlier detection, IEEE Trans. Knowl. Data Eng. 32 (8) (2020) 1517–1528.

[52] L. Ruff, N. Görnitz, L. Deecke, S. A. Siddiqui, R. A. Vandermeulen, A. Binder, E. Müller, M. Kloft, Deep one-class classification, in: Int. Conf. on Machine Learning, ICML, Vol. 80, 2018, pp. 4390–4399.

[53] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, M. Kloft, Deep semi-supervised anomaly detection, in: Int. Conf. on Learn. Repr., 2020.

[54] F. Angiulli, F. Fassetti, L. Ferragina, R. Spada, Cooperative deep unsupervised anomaly detection, in: Int. Conf. on Discovery Science, 2022, pp. 318–328.

[55] G. Pang, C. Shen, A. van den Hengel, Deep anomaly detection with deviation networks, in: Int. Conf. ACM SIGKDD, ACM, 2019, pp. 353–362.

[56] P. Liznerski, L. Ruff, R. A. Vandermeulen, B. J. Franks, M. Kloft, K. Müller, Explainable deep one-class classification, in: Int. Conf. on Learn. Repr., 2021.

[57] X. Yao, R. Li, J. Zhang, J. Sun, C. Zhang, Explicit boundary guided semi-push-pull contrastive learning for supervised anomaly detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 24490–24499.

[58] G. Pang, C. Ding, C. Shen, A. v. d. Hengel, Explainable deep few-shot anomaly detection with deviation networks, arXiv preprint arXiv:2108.00462 (2021).

[59] V. Zavrtanik, M. Kristan, D. Skočaj, Draem-a discriminatively trained reconstruction embedding for surface anomaly detection, in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 8330–8339.

[60] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[61] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.

[62] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[63] P. Bergmann, M. Fauser, D. Sattlegger, C. Steger, Mvtec ad — a comprehensive real-world dataset for unsupervised anomaly detection, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9584–9592.

[64] B. C. et al., mlco2/codecarbon: v2.4.1 (May 2024). doi:10.5281/zenodo.11171501.
URL https://doi.org/10.5281/zenodo.11171501