UNSUPERVISED THRESHOLD LEARNING WITH "L"-TREND PRIOR FOR VISUAL ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper considers unsupervised threshold learning, a practical yet under-researched module of anomaly detection (AD) for image data. AD comprises two separate modules: score generation and threshold learning. Most existing studies are more curious about the first part. It is often assumed that if the scoring module is good, estimating an accurate threshold is within easy reach. However, we argue that in the context of computer vision, some challenges in high-dimensional space lead threshold estimation be a non-trivial problem. In this paper, we leverage the inherent difference between normal instances and anomalies by ranking their anomaly score, which shows a phenomenon that involves two distinct trends. We term it as the "L"-trend prior. With that finding, we utilize an adaptive polynomial regression model to determine the threshold. Unlike the classic threshold learners which rely on enough training samples or statistical assumptions, this method is plug-and-play that can be implemented into different anomaly score function among various datasets. Also, the evaluation results demonstrate an obvious improvement.

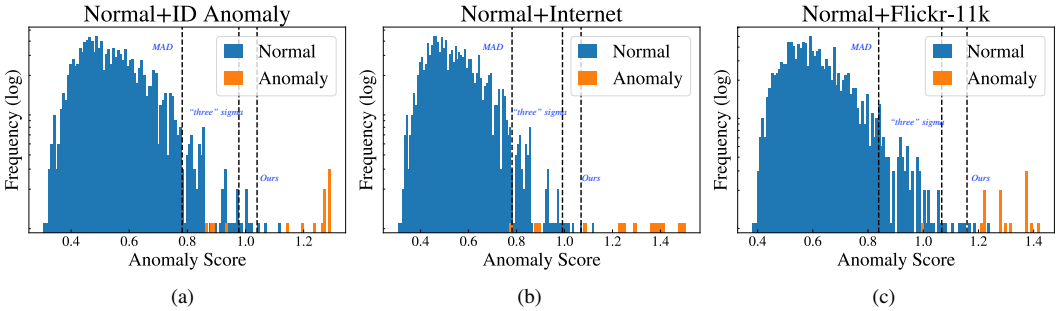


Figure 1: **Comparison with classic statistical methods: MAD and "three"-sigma rule.** For STL-10, (a) refers to normal+ID-anomaly and (b), (c) refers to combining normal and two OOD anomalies respectively. Obviously, the statistical methods tend to predict a loose threshold with lower *Precision* while our predicted threshold can maintain the trade-off between *Precision* and *Recall*.

1 INTRODUCTION

Visual anomaly detection may be the key to numerous vision problems like defect detection (Li et al., 2021; Bergmann et al., 2019), disease diagnosis (Ukil et al., 2016; Lenning et al., 2017), and automatic driving (Luo et al., 2021). Interestingly, while our ability to quantify anomalousness has improved significantly, (Lai et al., 2019; Li et al., 2022; Lin et al., 2022), there are few examples of anomaly detectors being deployed on the real-world problems. Our paper argues that it is insufficient to merely quantify anomalousness. To be effective, the anomaly detector must also determine a threshold between normal and anomalous instances. This is trivial in many applications but surprisingly difficult in visual anomaly detection.

Threshold learning performs like a classifier on a series of continuous or discrete variables. It is a crucial module in segmentation (Bergmann et al., 2019); Re-ID (Wang et al., 2020); face verification

(Wang et al., 2022; Gera et al., 2022). To our best knowledge, little attention has been paid to visual anomaly detection. We believe the thresholding problem arises because within-class variations are especially large in computer vision problems, making anomalies only slightly more different from the normal instance, than normal instances are from each other. For example, the variation among cats is so large that the difference between a dog and a cat, is only slightly greater than the difference between two breeds of cats, making it hard to define a threshold for what constitutes "too anomalous".

Within anomaly detection, thresholding can be considered as a subsequent technique after score generation. With the rapid progress of deep neural networks, which gives us more discriminative feature embedding, recent anomaly detectors have reached faithful score functions. However, it is still challenging to exploit the threshold since the boundary between normal instances and anomalies is ambiguous. In many cases, thresholds are so badly estimated that *Precision* or *Recall* is almost zero. As the anomaly ratio is usually less than 0.01 (Berg et al., 2019), there is an implication that the learned threshold may mistakenly involve many normal instances.

We address this problem by understanding the inherent distinctiveness of anomalous instances. It is troublesome that cover the given score function $\mathcal{Z}(\cdot)$ into mathematics. For example, the "three-sigma" rule (Leys et al., 2013; Bakar et al., 2006) assume the normality is Standard Gaussian Distribution, usually fails in realistic scenarios (Li et al., 2022), illustrated in Figure 1. However, children can easily recognize anomalies from the contaminated dataset, even if they are not taught before. We wonder this because human cognition can detect something distinct in a complex scene, which provides us with a novel perspective that an anomalous instance is not only different from normal observations but varies from other anomalies. Most previous works (Schölkopf et al., 2001; Liu et al., 2008; Lin et al., 2021) only focus on the first cue while we will concentrate on both of them. Thus, threshold learning can be quite easy.

How to view the score function is crucial. We first propose a sort transformation, which projects the anomaly score into a two-dimensional (x-y) space. The y-component is the instance's anomaly score while the x-component refers to its ranking index. Given an accurate scoring module $\mathcal{Z}(\cdot)$, the anomalies $\mathcal{A}$ will have higher value: $\mathcal{Z}(\mathcal{A}) > Z(\mathcal{N})$. Besides, anomalies have higher variation: $var(\mathcal{Z}(\mathcal{A})) > var(\mathcal{Z}(\mathcal{N}))$ and much lower density: $|\mathcal{A}| \ll |\mathcal{N}|$. By leveraging two natural properties, we observe: ***sorted*** $\mathcal{Z}(\mathcal{N})$ ***varies slightly on*** y***-component and varies exceedingly on*** x***-component while sorted*** $\mathcal{Z}(\mathcal{A})$ ***performs on the contrary.*** If this empirical hypothesis is correct, the sort transform will cause a "L"-trend. As the normal data $\mathcal{N}$ form a long quasi-horizontal line, which is the base of the "L". And $\mathcal{A}$ will create a vertical branch of the "L".

"L"-trend prior describes there exist two disparate patterns on the contaminated data, which inspires us to fit a naive polynomial regressor to the normal pattern. However, the polynomial degree is uncertain. To automatically learn it without tuning any hyperparameter, we propose a constrained optimizer. The objective is predicting the anomalous set contains a smaller number of instances but with higher variation. Our proposed method achieves state-of-the-art performance, moreover, it is a novel perspective for analyzing unsupervised data distribution. We hope it can contribute to further research domains like meta-learning. The main contributions of this paper are summarised as follows:

- We first suggest threshold learning be an important module for unsupervised visual AD.
- We introduce a new perspective: "L"-trend prior, for a better understanding of the anomalous procedure, making the further methods close to real application domains.
- We propose a data-adaptive regressor to fit any anomaly score function, then learn the threshold. It shows an obvious increase compared with previous threshold learners.

2 Related Work

2.1 Unsupervised Visual Anomaly Detector

Unsupervised visual anomaly detectors can be categorized into two approaches: classical models and deep learners. Classical anomaly detection models are mostly manifold-based approaches that attempt to describe the underlying normal pattern within the data through some form of statistical technique. OC-SVM (Schölkopf et al., 2001) which estimates a hypersphere containing most of the normal samples in the input space with kernel tricks and Local Outlier Factor (LOF) (Breunig et al., 2000) which determines anomalies via a distance to a local neighborhood of instances are popular

example of a classical model approach. The distance-based approach (Lin et al., 2022) leverages the fact that normal instances tend to be close together, while anomalies are far apart even from their closest neighbors with respect to some distance metric. With the rapid progress of deep neural networks, a series of representation-based methods have also been proposed. They tend to learn the discriminative features in the hidden space with auto-encoder such as DRAE (Xia et al., 2015); RDAE (Zhou & Paffenroth, 2017) and RSRAE (Lai et al., 2019).

2.2 Threshold Learner

Threshold Learning is an important module of various computer vision fields such as segmentation (Bergmann et al., 2019), Re-ID (Wang et al., 2020) and face verification (Wang et al., 2022; Gera et al., 2022). The above-mentioned utilizes the normal samples to decide the threshold (boundary). While it is common to approach this task in a supervised manner, the reality is for most practical use cases, the normal instances are not available which makes unsupervised anomaly detection challenging. As such to tackle this task in an unsupervised manner, we split the previous threshold learning methods of unsupervised anomaly detection into two sections: detection-based and detection-free.

Detection-Based. Some classical anomaly detectors such as OC-SVM (Schölkopf et al., 2001), IF (Liu et al., 2008), Shell (Lin et al., 2021) have their built-in threshold learning models, so they can predict both the anomaly score and label at the same time. Alternatively, the outlier removal model DRAE (Xia et al., 2015) produces the discriminative anomaly score function and then implements clustering techniques to decide the threshold.

Detection-Free. The detection-free threshold learner most is based on statistical analysis that can be utilized for any given anomaly score. The representative works are "three-sigma" rule (Leys et al., 2013; Bakar et al., 2006), MAD (Iglewicz & Hoaglin, 1993) and RANSAC (Lai et al., 2019). MAD is the simplest threshold that measures how to spread out a set of data, while RANSAC is the robust linear fitting method, which also implements the MAD as the threshold of fitted value.

3 Problem Statement

Traditionally, the objective of anomaly detection is to learn a score function $\mathcal{Z}(\cdot)$, which maps each instance $x_i \in \mathbb{X}$ to an anomaly score $\mathcal{Z}(x_i)$ that refers to the likelihood of x_i being anomalous. For example, if we choose the euclidean distance (Squared L_2-norm) as the score generator, the function is: $\mathcal{Z}(x_i) = \|x_i - \mu_{\mathbb{X}}\|^2$. After that, the threshold is decided by the following rule:

$$g(x_i) = \begin{cases} \text{normal}, & \mathcal{Z}(x_i) < \tau \\ \text{anomalous}, & \mathcal{Z}(x_i) \geq \tau \end{cases} \tag{1}$$

$g(\cdot)$ is the indicator, τ is the threshold, which helps us to predict the labels $[g_1, g_2, \cdots]$, where $g_i =$ normal *or* anomalous. Thus, we suggest the general anomaly detection pipeline involves two separated modules: *score generation* and *threshold learning*.

$$[x_1, x_2, \cdots] \xrightarrow[score\ function: \mathcal{Z}(\cdot)]{score\ generation} [\mathcal{Z}(x_1), \mathcal{Z}(x_2), \cdots] \xrightarrow[threshold: \tau]{thresholding} [g_1, g_2, \cdots] \tag{2}$$

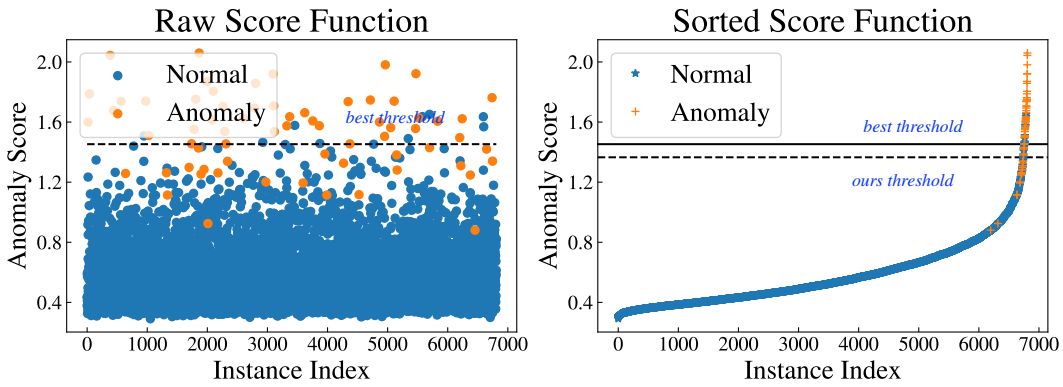
In this paper, we will concentrate on threshold learning given any kind of anomaly score function. We believe an accurate threshold can split the anomalies well from the normal set, achieving a trade-off between *Precision* and *Recall*, approaching the maximum possible *F1_score*. Moreover, it needs the generalization ability with different anomaly score functions among various datasets.

4 "L"-trend Prior

4.1 Sort Transformation

In Figure 2 (a), it is difficult to explore the normality in raw anomaly score function. To address this problem, we propose a sort-transformation $\widetilde{\mathcal{Z}} = sort(\mathcal{Z})$. It projects $\mathcal{Z}$ into a two-dimensional (x-y) space. The sort-transform firstly ranks instances by their anomaly scores. It then represents each instance as a two-dimensional data point, the y-component being the instance's anomaly score

and the x-component refers to its ranking index. In this way, the sorted anomaly score function [1]is monotonically increasing. Obviously, it helps better visualize normal behavior, shown in Figure 2 (b).



(a) (b)

Figure 2: **Visualization of different types of the anomaly score function.** (a) is the raw anomaly score function, it is quite hard to define the boundary. (b) means the determination of the threshold between normal data and anomalies is possible if we sort the distance. We can apply a polynomial curve to fit the normal pattern which allows the abnormal group to be detected.

4.2 Understanding the Distribution

Ideally, there exist a distinctive-shell that almost-surely encapsulates all normal instances $\mathcal{N} = \{x_i|\mathcal{Z}(x_i) = const\}$ and excludes almost-all anomalies $\mathcal{A} = \{x_i|\mathcal{Z}(x_i) > const\}$, which indicates the normal data share the same function value (Lin et al., 2021). In practice, the assumption is not realistic. Assuming the major class is "airplane" if there are no anomalies from other classes, the "wing", and "cockpit" will be relatively more anomalous-like than other airplane images. Therefore, normality has sub-clusters that cannot be covered with a single mechanism. In other word, it is *non-linear*. Within normal set $\mathcal{N}$, we denote these anomalous-like instances as $\mathcal{V}$ and the other as $\mathcal{U}$, so $\mathcal{N} = \mathcal{U} \cup \mathcal{V}$. $\mathcal{V}$ share the congruous semantic attributes with the real normal images but heterogeneous feature embedding, resulting in the variation of normal anomaly score set $\mathcal{Z}(\mathcal{N}) = \mathcal{Z}(\mathcal{U}) \cup \mathcal{Z}(\mathcal{V})$.

Most previous anomaly detectors tend to explore the normality. Drawn a lesson from the above analysis, we suggest the anomalous pattern also should be considered. We believe that anomalies are not only different from normal instances but also vary from other anomalous data. For example, a diagnosed medical disease can have multiple etiologies. The distinction can be reflected in the score function, where $\mathcal{Z}(\mathcal{A})$ demonstrate higher variance and lower density than $\mathcal{Z}(\mathcal{N})$.

4.3 "L"-trend

Sort transform will cause contaminated data to form a "L"-trend. As the majority normal samples are similar to each other and amount accounts for the majority of the normal set, they will be concentrated at a specific score and thus form a long quasi-horizontal line, which is the base of the "L". While noisy instances are different from each other and rare, they will create a vertical branch of the "L"-trend.

"L"-trend prior explains there are two natural different patterns in the contaminated data. Thus, it's trivial to generalize the prior from $\mathcal{Z}(\mathcal{N})$ to $\mathcal{Z}(\mathcal{N} \cup \mathcal{A})$. As $var(\mathcal{Z}(\mathcal{A})) > var(\mathcal{Z}(\mathcal{N}))$ and $|\mathcal{A}| \ll |\mathcal{N}|$, thus $\frac{var(\mathcal{Z}(\mathcal{A}))}{|\mathcal{A}|} \gg \frac{var(\mathcal{Z}(\mathcal{N}))}{|\mathcal{N}|}$. Within the normality $\mathcal{Z}(\mathcal{N}) = \mathcal{Z}(\mathcal{U}) \cup \mathcal{Z}(\mathcal{V})$, $\mathcal{Z}(\mathcal{V})$ shows a sharp ascending trend compared with $\mathcal{Z}(\mathcal{U})$ while in the whole target dataset, $\mathcal{Z}(\mathcal{A})$ shows a sharper increasing trend compared with $\mathcal{Z}(\mathcal{N})$. This gives us a blessing that fit the normality with a constrained optimization problem.

[1]Unless otherwise stated, the term anomaly score function of visualization is used to refer to Squared L_2-norm, which is naive and intuitive.

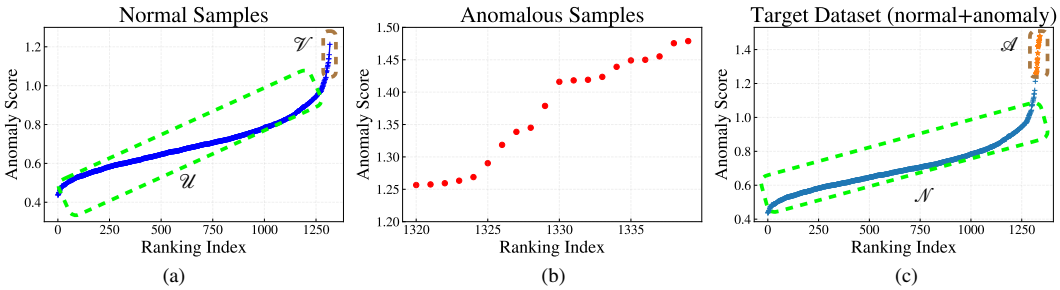


Figure 3: **A new prospective of unsupervised threshold learning: "L"-trend prior.** (a) shows the "L"-trend prior of normality. The majority $\mathcal{U}$ are gathering together, form a long quasi-horizontal line, which is the base of the "L". As noisy normal instances $\mathcal{A}$ are different from each other, they will have a variety of different scores, and thus create a vertical branch. (b) demonstrate the natural properties of anomalies: sparse and higher anomaly likelihood. (c) is the visualization of the whole target dataset, which proves the prior knowledge can be generalized and robust.

5 METHODOLOGY

5.1 POLYNOMIAL CURVE FITTING

Inspired by the "L"-trend prior, the normality is non-linear but varies smoothly compared to their anomaly counterparts. So we propose a robust polynomial regression with an upper bound to identify the nonlinear normality distribution. We construct a polynomial curve $y_d = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_d x^d + \varepsilon$ to fit the objective: $\widetilde{\mathcal{Z}}$. The parameters $\beta_0, \cdots, \beta_d, \varepsilon$ are automatically learned via least squares method. With the degree list $\{d_1, d_2, \cdots, d_{max}\}$, we have a series of candidate polynomial curves $\{y_1, y_2, \cdots\}$. For stability, the maximum degree is capped at 5. For each fitted curve, the threshold is determined as:

$$\tau_d = \max(y_d) \tag{3}$$

5.2 AVERAGE SEPARABILITY

As there is no supervision, we need to evaluate how well the polynomial regressor fits normality. After determining the threshold τ, we have the predicted normal data $\mathcal{N} = \{x_i | \mathcal{Z}(x_i) < \tau\}$ and the anomalies $\mathcal{A} = \{\mathcal{Z}(x_i) | \mathcal{Z}(x_i) \geq \tau\}$. We define the separability s of each set as:

$$s(\mathcal{N}) = var(\mathcal{Z}(\mathcal{N})), s(\mathcal{A}) = var(\mathcal{Z}(\mathcal{A})) \tag{4}$$

Thus, the average separability $\bar{s}$ is:

$$\bar{s}(\mathcal{N}) = \frac{s(\mathcal{N})}{|\mathcal{N}|} = \frac{var(\mathcal{Z}(\mathcal{N}))}{|\mathcal{N}|}, \bar{s}(\mathcal{A}) = \frac{s(\mathcal{A})}{|\mathcal{A}|} = \frac{var(\mathcal{Z}(\mathcal{A}))}{|\mathcal{A}|} \tag{5}$$

We wish the predicted anomalous set contains fewer instances while higher variation. So the accuracy of normality fitting Acc is proportional to the compactness $Comp$ of separability.

$$Acc \propto \frac{var(\mathcal{Z}(\mathcal{A}))}{var(\mathcal{Z}(\mathcal{N}))} * \frac{|\mathcal{N}|}{|\mathcal{A}|} = \frac{\frac{s(\mathcal{A})}{|\mathcal{A}|}}{\frac{s(\mathcal{N})}{|\mathcal{N}|}} = \frac{\bar{s}(\mathcal{A})}{\bar{s}(\mathcal{N})} = Comp \tag{6}$$

5.3 DEGREE OPTIMIZATION

For each polynomial regressor with each degree d, the predicted normal and anomalous set is $\mathcal{N}_d = \{x_i | \mathcal{Z}(x_i) < \tau_d\}$, $\mathcal{A}_d = \{x_i | \mathcal{Z}(x_i) \geq \tau_d\}$. So we choose the optimal polynomial degree with the following rule. The only parameter in our method is the degree of the polynomial curve. To find the optimal degree d^*, we utilize the following rule:

$$d^* = \begin{cases} \arg\max\limits_{d} \ Comp_d \\ \text{s.t. } d \leq d_{max} \end{cases} \tag{7}$$

Based on Equations (6) and (7), the automatically learned threshold is:

$$\tau^* = \max(y_{d^*}) \tag{8}$$

Data-Type	Dataset-Name	Description	Feature	# Instance/class
Normal-Dataset	MNIST LeCun & Cortes, 2010	hand-wrcitten digit	Raw Pixel	$\sim$ 6,000
	Fashion-MNIST Xiao et al., 2017	fashion product	Raw Pixel	6,000
	STL-10 Coates et al., 2011	subset of ImageNet	ResNet-50	1,300
	CIFAR-10 Krizhevsky et al., 2009	from ImageNet	ResNet-50	6,000
	MIT-Places-Small Zhou et al., 2017	subset of MIT-Places	ResNet-50	$\sim$ 3,000
OOD Anomaly	Internet-STL10 Lin et al., 2021	web image	ResNet-50	$\sim$ 580
	Flickr-11k Van Miltenburg, 2016	sentence-based image	ResNet-50	$\sim$ 11,000

Table 1: Summary statistics of all benchmark datasets and OOD anomalies. The OOD anomaly means the abnormal data comes from the out-of-distribution dataset: Internet-STL10 and Flickr-11k.

Dataset: *Avg* ***(STL-10, CIFAR-10, MIT-Places-Small)***						
Feature	**Thresholding Algorithm**	**Anomaly Score**	**Precision**	**Recall**	**F1-score**	**F1-ratio**
Pretrained ResNet-50	Shell Lin et al., 2021	–	0.051	0.750	0.095	0.563
	Shell-re Lin et al., 2021	–	0.013	0.979	0.025	0.628
	OC-SVM Schölkopf et al., 2001	–	0.020	0.981	0.038	0.089
	IF Liu et al., 2008	–	0.277	0.119	0.089	0.174
	RANSAC Fischler & Bolles, 1981	Squared L_2-norm	0.220	0.683	0.173	0.484
		OC-SVM	0.138	0.737	0.226	0.495
		ECOD Li et al., 2022	0.095	0.631	0.162	0.508
		RSRAE Lai et al., 2019	0.134	0.831	0.229	0.514
		LVAD Lin et al., 2022	0.267	0.601	0.313	0.581
	Ours	Squared L_2-norm	0.484	0.423	0.451	0.843
		OC-SVM	0.496	0.456	**0.471**	**0.864**
		ECOD Li et al., 2022	0.322	0.291	0.301	0.738
		RSRAE Lai et al., 2019	0.392	0.394	0.389	0.667
		LVAD Lin et al., 2022	0.521	0.434	0.464	**0.864**
Dataset: *Avg* ***(MNIST, Fashion-MNIST)***						
Feature	**Thresholding Algorithm**	**Anomaly Score**	**Precision**	**Recall**	**F1-score**	**F1-ratio**
Raw-pixel	Shell Lin et al., 2021	–	0.030	0.489	0.057	0.647
	Shell-re Lin et al., 2021	–	0.011	0.959	0.023	0.834
	OC-SVM Schölkopf et al., 2001	–	0.019	0.949	0.036	0.124
	IF Liu et al., 2008	–	0.055	0.403	0.100	0.511
	DRAE Xia et al., 2015	–	0.301	0.429	0.213	0.574
	RANSAC Fischler & Bolles, 1981	Squared L_2-norm	0.071	0.706	0.126	0.438
		OC-SVM	0.083	0.713	0.147	0.437
		RSRAE Lai et al., 2019	0.046	0.656	0.085	0.376
		LVAD Lin et al., 2022	0.285	0.486	0.222	0.526
	Ours	Squared L_2-norm	0.317	0.287	0.299	0.783
		OC-SVM	0.365	0.353	**0.355**	0.816
		RSRAE Lai et al., 2019	0.272	0.251	0.261	0.732
		LVAD Lin et al. (2022)	0.440	0.335	**0.355**	**0.836**

Table 2: Average *Precision*, *Recall*, *F1_score* and *F1_ratio* per method on two data categories: ResNet-50 feature (STL-10, CIFAR-10 and MIT-Places-Small) and Raw pixel (MNIST and Fashion-MNIST). *F1_score* is important to justify whether a threshold learner is good or not, *F1_ratio* is compared with the optimal threshold (Ground Truth) given the score function. Our method holds the SOTA results both of them in various datasets. The best-performing method is in bold.

Dataset: STL-10 (Anomaly: Internet-STL10)					
Thresholding Algorithm	**Anomaly Score**	**Precision**	**Recall**	**F1-score**	**F1-ratio**
Shell Lin et al., 2021	–	0.060	0.869	0.112	0.483
Shell-re Lin et al., 2021	–	0.012	0.977	0.025	0.714
OC-SVM Schölkopf et al., 2001	–	0.020	1.000	0.039	0.044
IF Liu et al., 2008	–	0.478	0.154	0.219	0.356
RANSAC Fischler & Bolles, 1981	Squared L_2-norm	0.178	0.962	0.297	0.328
	OC-SVM	0.147	0.977	0.254	0.286
	ECOD Li et al., 2022	0.127	0.769	0.216	0.461
	RSRAE Lai et al., 2019	0.201	1.000	0.332	0.426
	LVAD Lin et al., 2022	0.408	0.792	0.466	0.566
Ours	Squared L_2-norm	0.837	0.831	0.819	0.929
	OC-SVM	0.870	0.854	**0.860**	0.959
	ECOD Li et al., 2022	0.471	0.431	0.448	0.883
	RSRAE Lai et al., 2019	0.749	0.769	0.755	**0.963**
	LVAD Lin et al., 2022	0.814	0.715	0.745	0.896
Dataset: STL-10 (Anomaly: Flickr-11k)					
Thresholding Algorithm	**Anomaly Score**	**Precision**	**Recall**	**F1-score**	**F1-ratio**
Shell Lin et al., 2021	–	0.064	0.938	0.120	0.330
Shell-Re Lin et al., 2021	–	0.013	0.992	0.025	0.635
OC-SVM Schölkopf et al., 2001	–	0.020	1.000	0.039	0.040
IF Liu et al., 2008	–	0.533	0.092	0.149	0.282
RANSAC Fischler & Bolles, 1981	Squared L_2-norm	0.184	1.000	0.308	0.317
	OC-SVM	0.155	1.000	0.265	0.273
	ECOD Li et al., 2022	0.159	0.962	0.270	0.404
	RSRAE Lai et al., 2019	0.201	0.985	0.332	0.362
	LVAD Lin et al., 2022	0.525	0.792	0.476	0.506
Ours	Squared L_2-norm	0.923	0.977	**0.948**	**0.978**
	OC-SVM	0.922	0.962	0.940	0.974
	ECOD Li et al., 2022	0.535	0.685	0.585	0.854
	RSRAE Lai et al., 2019	0.818	0.938	0.872	0.945
	LVAD Lin et al., 2022	0.863	0.808	0.815	0.868

Table 3: The performance of threshold learning with Out-of-Distribution anomalies (Internet-STL10 and Flickr-11k).

6 Evaluation

6.1 Data Preparation

For gray-scale datasets: MNIST and Fashion-MNIST, we utilize the rasterized pixels as image features and we choose ResNet-50 (He et al., 2016) as the feature extractor that is pre-trained on ImageNet (He et al., 2019) for other RGB datasets, The statistics details are shown in Table 1. There are two creation schemes for the target dataset: (I) Normal+In-Distribution (ID) anomaly: we designate every single class in the dataset in every round as normal data, while randomly choosing anomalies from the remaining, then combining them. (II) Normal+Out-Of-Distribution (OOD) anomaly: the anomalies are coming from Flickr-11k and Internet STL-10, and use the same protocol as (I). The above data mixing is applied to all datasets.

6.2 Evaluation Metric

Adopting Area Under the Receiver Operating Characteristic curve (AUROC) (Fawcett, 2006) is common in score generation evaluation, but it cannot evaluate the threshold learning performance. Besides the current decision metrics: *Precision, Recall, F1_score*, we propose a new evaluation scheme called *F1_ratio* to achieve fair comparison as there are seldom perfect anomaly detectors.

$$F1\_ratio = \frac{F1\_score}{best\ F1\_score} = \frac{F1\_score}{max\{F1\_score_\tau, \forall \tau \in \mathcal{Z}\}} \tag{9}$$

6.3 Unsupervised Threshold Learner

- **OC-SVM:** proposed by Schölkopf et al., 2001. The one-class support vector machine aims to learn a hypersphere containing most of the normal samples in the input space with kernel tricks. Samples outside of the normal group are deemed anomalous.
- **IF:** proposed by Liu et al., 2008. Isolation forest is a classification algorithm based on a tree structure. It can learn the distribution of a sample and then separate this sample from other types of samples.
- **Shell, Shell-Re:** proposed by Lin et al., 2021. Shell theory suggests there exist naturally occurring class boundaries that are defined in terms of the mean-variance of each class. Those instances outside of the normal boundary are considered anomalous.
- **RANSAC:** proposed by Fischler & Bolles, 1981. RANdom SAmple Consensus is a resampling method that generates candidate solutions by choosing the minimum number of instances required to estimate parameters. We implement RANSAC with a sorted anomaly score as a robust linear regression model, the threshold is defined by the MAD (Median Absolute Deviation) strategy.
- **DRAE:** proposed by Xia et al., 2015. DRAE implements the clustering model after a discriminative autoencoder is learned, we can apply any clustering algorithm (such as K-Means) to partition reconstruction errors into two clusters. Samples in the cluster with a large average error are identified as outliers.

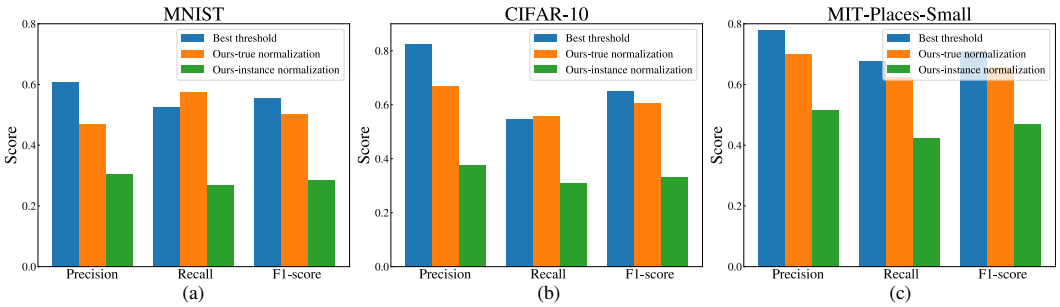


Figure 4: **Threshold learning with true normalization.** If normalized with the mean of the whole dataset (true normalization), the AUC will dramatically be increased, and the results of our method also improved, close to the best threshold manually computed. So with a more accurate anomaly score function, our threshold learner becomes more powerful.

6.4 Results Analysis

We show the performance of our method evaluated on two kinds of target dataset schemes: ID-anomaly and OOD-anomaly respectively with various anomaly score functions among a series of benchmarks. The current threshold learners often rely on statistical assumptions or fixed thresholds. From our view, these assumptions are not robust, especially in high-dimensional space. So, most of them predict a loose threshold, that is the predicted anomalous set involves many normal instances, resulting in a high recall score but low precision score, illustrated in the third and fourth columns in Table 2 and 3. Our proposed method outperforms all existing threshold learners, achieve the trade-off between *Precision* and *Recall*, thus demonstrate a dramatic increase on *F1-score* and *F1-ratio*. Besides the classical anomaly detection task with the in-dataset anomalies, we believe in practice, the anomalies will come from other out-of-dataset distributions. In Table 3, our method even close to perfection. More results with different anomaly score functions can be found in the Appendix.

A good anomaly score function is critical for threshold learning, we also test the performance of our method with the true normalized score generator. With a more accurate score function, our

performance also shows dramatic improvement in Figure 4. Moreover, the threshold learner should be available to recognize different categories of anomalous data. If there is no anomaly, our method can continuously identify the more anomalous-like instances until convergence (no anomaly detected), illustrated in Figure 5.

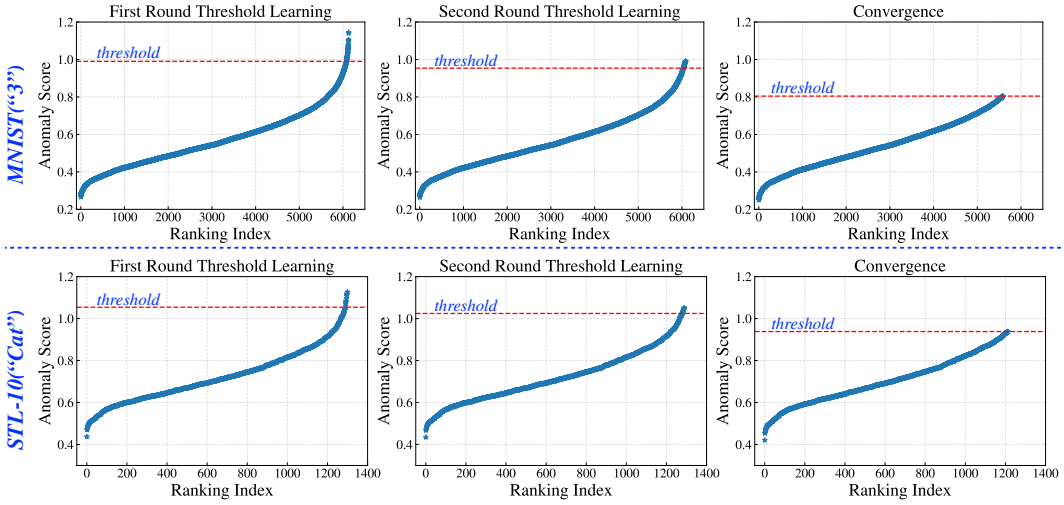


Figure 5: **Visualization of our method on no anomaly scene.** The empirical normal set involves some anomalous-like instances, shown on two datasets MNIST and STL-10. Our method can avoid overfitting to the normality. It can recursively defect the more anomalous-like data and clean the normal set until convergence.

6.5 Ablation Study

In this section, we first explore two different deep feature extractors: ResNet-50 and Swin-Transformer perform for anomaly detection. In Table 4, our threshold learning method can maintain the performance with two different features. In Table 5, we study the effect of the degree optimization of the polynomial curve. Compared with a non-degree optimization, our method shows about 0.5 and 0.2 *(F1_ratio)* improvement on STL-10 and CIFAR-10 respectively.

Feature Extractor	F1-ratio
ResNet-50 He et al., 2016	0.761
Swin-Transformer Liu et al., 2021	0.784

Table 4: Ablation Study 1: the impact of feature extraction on CIFAR-10.

Method	STL-10	CIFAR-10
Linear Regression	0.377	0.551
Ours	0.897	0.773

Table 5: Ablation Study 2: the impact of polynomial degree optimization.

7 Conclusion

This paper suggests the further anomaly detection domain should pay attention to threshold learning, and make it more useful in real applications. We introduce a new perspective for unsupervised anomaly detection called "L"- trend prior, which reflects the inherent variation. Based on the finding, we propose a data-adaptive and robust polynomial fitting algorithm to address the problem. The experimental results show that our method outperforms the traditional methods. It strengthens the robustness of the threshold and makes the selection process more tractable.

Moreover, we study the inherent coherence of normality and the contrast with anomalies, which demonstrate an unsupervised data distribution prior, we hope it can be further beneficial to other computer vision domains like few-shot learning.

REFERENCES

Zuriana Abu Bakar, Rosmayati Mohemad, Akbar Ahmad, and Mustafa Mat Deris. A comparative study for outlier detection techniques in data mining. In *Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems*, pp. 1–6, 2006.

Amanda Berg, Jörgen Ahlberg, and Michael Felsberg. Unsupervised learning of anomaly detection from contaminated image data using simultaneous encoder training. *arXiv preprint arXiv:1905.11034*, 2019.

Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9592–9600, 2019.

Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 93–104, 2000.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 215–223, 2011.

Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, pp. 861–874, 2006.

Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, pp. 381–395, 1981.

Darshan Gera, Naveen Siva Kumar Badveeti, Bobbili Veerendra Raj Kumar, and S Balasubramanian. Dynamic adaptive threshold based learning for noisy annotations robust facial expression recognition. *arXiv preprint arXiv:2208.10221*, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4918–4927, 2019.

Boris Iglewicz and David Caster Hoaglin. *How to detect and handle outliers*. 1993.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Chieh-Hsin Lai, Dongmian Zou, and Gilad Lerman. Robust subspace recovery layer for unsupervised anomaly detection. *arXiv preprint arXiv:1904.00152*, 2019.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

Michael Lenning, Joseph Fortunato, Tai Le, Isaac Clark, Ang Sherpa, Soyeon Yi, Peter Hofsteen, Geethapriya Thamilarasu, Jingchun Yang, Xiaolei Xu, et al. Real-time monitoring and analysis of zebrafish electrocardiogram with anomaly detection. *Sensors*, pp. 61, 2017.

Christophe Leys, Christophe Ley, Olivier Klein, Philippe Bernard, and Laurent Licata. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of experimental social psychology*, pp. 764–766, 2013.

Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9664–9674, 2021.

Zheng Li, Yue Zhao, Xiyang Hu, Nicola Botta, Cezar Ionescu, and George H Chen. Ecod: Unsupervised outlier detection using empirical cumulative distribution functions. *arXiv preprint arXiv:2201.00382*, 2022.

Daniel Lin, Siying Liu, Hongdong Li, Ngai-Man Cheung, Changhao Ren, and Yasuyuki Matsushita. Shell theory: A statistical model of reality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Wen-Yan Lin, Zhonghang Liu, and Siying Liu. Locally varying distance transform for unsupervised visual anomaly detection. 2022.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *Proceedings of International Conference on Data Mining*, pp. 413–422, 2008.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

Yuan Luo, Ya Xiao, Long Cheng, Guojun Peng, and Danfeng Yao. Deep learning-based anomaly detection in cyber-physical systems: Progress and opportunities. *ACM Computing Surveys (CSUR)*, pp. 1–36, 2021.

Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

Arijit Ukil, Soma Bandyoapdhyay, Chetanya Puri, and Arpan Pal. Iot healthcare analytics: The importance of anomaly detection. In *Proceedings of International Conference on Advanced Information Networking and Applications*, pp. 994–997, 2016.

Emiel Van Miltenburg. Stereotyping and bias in the flickr30k dataset. *arXiv preprint arXiv:1605.06083*, 2016.

Guan'an Wang, Shaogang Gong, Jian Cheng, and Zengguang Hou. Faster person re-identification. In *Proceedings of European Conference on Computer Vision*, pp. 275–292, 2020.

Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, Zhen Wu, and Jindong Wang. Freematch: Self-adaptive thresholding for semi-supervised learning. *arXiv preprint arXiv:2205.07246*, 2022.

Yan Xia, Xudong Cao, Fang Wen, Gang Hua, and Jian Sun. Learning discriminative reconstructions for unsupervised outlier removal. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1511–1519, 2015.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1452–1464, 2017.

Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 665–674, 2017.

A Appendix

Dataset	Algorithm	Anomaly Score	Precision	Recall	F1-score	F1-ratio
STL-10	Shell	–	0.054	0.785	0.100	0.571
	Shell-Re	–	0.013	0.992	0.025	0.683
	OC-SVM	–	0.020	1.000	0.039	0.058
	IF	–	0.362	0.131	0.188	0.314
	RANSAC	OC-SVM	0.136	0.869	0.233	0.350
		IF	0.101	0.615	0.171	0.512
		L_2-norm	0.083	0.667	0.147	0.493
		KDE	0.192	1.000	0.317	0.525
		LOF	0.134	1.000	0.232	0.430
		ECOD	0.110	0.669	0.187	0.490
		RSRAE	0.202	1.000	0.334	0.499
		LVAD	0.304	0.500	0.341	0.472
	Ours	OC-SVM	0.616	0.638	0.613	**0.897**
		IF	0.342	0.300	0.318	0.720
		L_2-norm	0.606	0.554	0.578	0.889
		KDE	0.575	0.562	0.564	0.870
		LOF	0.405	0.569	0.459	0.821
		ECOD	0.351	0.346	0.338	0.774
		RSRAE	0.600	0.685	**0.637**	0.840
		LVAD	0.619	0.515	0.545	0.831
CIFAR-10	Shell	–	0.046	0.698	0.087	0.576
	Shell-Re	–	0.013	0.952	0.025	0.701
	OC-SVM	–	0.019	0.950	0.037	0.129
	IF	–	0.269	0.025	0.045	0.089
	RANSAC	OC-SVM	0.113	0.638	0.190	0.599
		IF	0.089	0.395	0.141	0.661
		L_2-norm	0.083	0.667	0.147	0.493
		KDE	0.081	0.670	0.143	0.477
		LOF	0.058	0.572	0.104	0.583
		ECOD	0.105	0.585	0.174	0.569
		RSRAE	0.084	0.687	0.150	0.478
		LVAD	0.185	0.637	0.273	0.658
	Ours	OC-SVM	0.357	0.308	0.330	0.773
		IF	0.229	0.183	0.203	0.684
		L_2-norm	0.355	0.310	0.330	0.761
		KDE	0.358	0.312	0.333	0.770
		LOF	0.166	0.142	0.153	0.732
		ECOD	0.327	0.288	0.306	0.751
		RSRAE	0.309	0.277	0.291	0.661
		LVAD	0.422	0.343	**0.378**	**0.858**
MIT-Places-small	Shell	–	0.053	0.768	0.099	0.542
	Shell-Re	–	0.013	0.994	0.025	0.500
	OC-SVM	–	0.020	0.994	0.038	0.081
	IF	–	0.200	0.200	0.035	0.120
	RANSAC	OC-SVM	0.164	0.703	0.256	0.535
		IF	0.076	0.290	0.110	0.641
		L_2-norm	0.493	0.716	0.224	0.466
		KDE	0.148	0.697	0.234	0.482
		ECOD	0.069	0.639	0.125	0.464
		RSRAE	0.117	0.806	0.202	0.565
		LVAD	0.311	0.665	0.326	0.614
	Ours	OC-SVM	0.516	0.423	**0.470**	**0.922**
		IF	0.183	0.168	0.174	0.909
		L_2-norm	0.491	0.406	0.445	0.879
		KDE	0.477	0.394	0.431	0.840
		ECOD	0.287	0.239	0.260	0.690
		RSRAE	0.266	0.219	0.239	0.500
		LVAD	0.521	0.445	0.469	0.903

Table 6: Average Precision, Recall, F1_socre and F1_ratio per method on STL-10, CIFAR-10, MIT-Places and Fashion-MNIST.

Dataset	Algorithm	Anomaly Score	Precision	Recall	F1-score	F1-ratio
Fashion-MNIST	Shell	–	0.028	0.463	0.053	0.650
	Shell-Re	–	0.011	0.947	0.023	0.752
	OC-SVM	–	0.019	0.935	0.036	0.098
	IF	–	0.070	0.767	0.128	0.385
	RANSAC	OC-SVM	0.080	0.720	0.143	0.353
		IF	0.102	0.670	0.166	0.536
		L_2-norm	0.050	0.792	0.092	0.286
		KDE	0.014	0.233	0.027	0.077
		LOF	0.023	0.338	0.042	0.557
		ECOD	0.034	0.485	0.063	0.594
		RSRAE	0.046	0.747	0.086	0.247
		LVAD	0.453	0.312	0.248	0.517
	Ours	OC-SVM	0.424	0.437	**0.426**	0.840
		IF	0.293	0.293	0.280	0.757
		L_2-norm	0.375	0.350	0.359	0.824
		KDE	0.397	0.403	0.393	**0.868**
		LOF	0.057	0.052	0.054	0.605
		ECOD	0.260	0.253	0.256	0.657
		RSRAE	0.368	0.340	0.352	0.860
		LVAD	0.467	0.357	0.357	0.857
MNIST	Shell	–	0.032	0.514	0.061	0.644
	Shell-Re	–	0.012	0.971	0.023	0.915
	OC-SVM	–	0.019	0.963	0.037	0.150
	IF	–	0.039	0.039	0.071	0.637
	RANSAC	OC-SVM	0.085	0.706	0.151	0.521
		IF	0.035	0.530	0.065	0.612
		L_2-norm	0.092	0.619	0.160	0.589
		KDE	0.016	0.136	0.029	0.051
		LOF	0.063	0.952	0.119	0.273
		ECOD	0.027	0.398	0.049	0.609
		RSRAE	0.046	0.566	0.084	0.504
		LVAD	0.116	0.660	0.195	0.535
	Ours	OC-SVM	0.306	0.268	0.285	0.791
		IF	0.064	0.059	0.061	0.469
		L_2-norm	0.259	0.224	0.240	0.742
		KDE	0.271	0.234	0.250	0.749
		LOF	0.418	0.388	**0.401**	**0.899**
		ECOD	0.149	0.139	0.144	0.555
		RSRAE	0.176	0.162	0.169	0.604
		LVAD	0.412	0.312	0.353	0.815

Table 7: Average Precision, Recall, F1_socre and F1_ratio per method on STL-10, CIFAR-10, MIT-Places, and Fashion-MNIST.