A Robust Likelihood Model for Novelty Detection

Ranya Almohsen* Shivang Patel* Donald A. Adjeroh Gianfranco Doretto
West Virginia University
Morgantown, WV 26506
{ralmohse, sap00008, daadjeroh, gidoretto}@mix.wvu.edu

Abstract

Current approaches to novelty or anomaly detection are based on deep neural networks. Despite their effectiveness, neural networks are also vulnerable to imperceptible deformations of the input data. This is a serious issue in critical applications, or when data alterations are generated by an adversarial attack. While this is a known problem that has been studied in recent years for the case of supervised learning, the case of novelty detection has received very limited attention. Indeed, in this latter setting the learning is typically unsupervised because outlier data is not available during training, and new approaches for this case need to be investigated. We propose a new prior that aims at learning a robust likelihood for the novelty test, as a defense against attacks. We also integrate the same prior with a state-of-the-art novelty detection approach. Because of the geometric properties of that approach, the resulting robust training is computationally very efficient. An initial evaluation of the method indicates that it is effective at improving performance with respect to the standard models in the absence and presence of attacks.

1. Introduction

Recognizing inliers or outliers with respect to a probability distribution is a task known as *novelty* or anomaly *detection* [39]. It is a fundamental problem in many applications, and when computer vision supports agents exploring the world "in-the-wild" it is expected that sensed data might not belong to the distribution with respect to which models were previously trained. Such data has to be detected as being *out of distribution* and subsequent appropriate action should be taken for its processing (e.g., open-set recognition [46]). This novelty detection task is inherently challenging because out of distribution data is normally not available, or even dangerous to obtain in certain applications; thus, using fully unsupervised approaches is often mandatory.

*Denotes equal contribution.

Among the most succesful methods for novelty detection there are those based on deep neural networks [36, 37, 40]. Although very effective, neural networks are vulnerable to even small perturbations of the input [51]. This means that inlier or outlier samples could be easily misclassified, despite a seemingly unnoticeable change, and it might even be possible that such changes could be operated by an adversarial entity, effectively instantiating an adversarial attack. For the case of supervised learning the problem has received considerable attention, and several approaches have been designed to deploy different types of defenses, starting from one of the most popular and effective, which is adversarial training [15]. Surprisingly, very little attention has been devoted to the development of defenses for the unsupervised case of novelty detection.

In this work, we propose a new learning prior as a defense mechanism against input data distortions or attacks that might affect a novelty detection test. We specifically focus on the case where the test statistic is the likelihood of the input data sample, which is also a very general and principled approach to novelty detection. We take a robust optimization approach, where instead of optimizing a robust supervised loss [27], we aim at learning a robust likelihood statistic. We then integrate this principle with a recent probabilistic novelty detection approach [4]. We show that because of the geometric properties of that method, the implementation of the proposed defense prior is particularly advantageous, since the synthetic generation of outliers and inliers turns out to be very efficient. We present an initial study of the proposed approach, and show that the performance metrics improve when the robust model is compared against the starting model under standard benchmarks, as well as when the benchmarks are under attacks.

2. Related Work

Novelty and anomaly detection have been studied in many domains. The central idea is to learn a model of normality from *in distribution* data in an unsupervised manner such that during training no prior knowledge is available about abnormal, *out of distribution* samples. Under this formu-

lation *novelty* and *anomaly* detection are used interchangeably [39, 44]. Here we review several traditional and deep learning based approaches.

Traditional Approaches. Most traditional novelty detection approaches are based on either density estimation [7, 12, 19] or reconstruction [2]. One-Class Support Vector Machines (OCSVM) [48] and its extension Support Vector Data Description (SVDD) [52] are unsupervised methods, where the former learns a boundary around samples of a normal class, and the latter uses an hypersphere containing all normal samples with a minimum radius. The performance of these approaches is reported to degrade on complex high dimensional datasets [10]. Other unsupervised approaches include Robust Principal Component Analysis (RPCA) [9], and Isolation Forest (IF) [24]. RPCA learns a linear subspace and it identifies the anomalies in the training data, thereby removing them and retraining at each iteration. IF tries to isolate anomalies from normal samples via successive random partitions of the feature space. Compared to those approaches, our method learns how to compute the likelihood of data samples and does so by making the likelihood training robust.

Deep Learning Approaches. Traditional approaches have been extended with deep learning. One-class Neural Network (OC-NN) [10] is the first approach that integrates the OC-SVM loss in the network training. Deep SVDD [40] instead works by jointly training a deep neural network while optimizing a data-enclosing hypersphere in the output space. Autoencoders have been used effectively for learning representations of the normal distribution [42, 63]. They learn common features in normal data and abnormal samples cannot be reconstructed accurately because they usually contain also other features, although it has been reported that different types of out of distribution samples can sometimes be reconstructed reasonably well [54]. Some variants of the autoencoders proposed for anomaly detection include: denoising autoencoders [57], sparse autoencoders [28], variational autoencoders (VAEs) [5], and deep convolutional autoencoders (DCAEs) [29, 31]. We also use an autoencoder, but the likelihood model that we build on, does not rely only on reconstruction error, and therefore it is less affected by the potential issues related to the reconstruction of outliers.

Other approaches combine autoencoders with Generative Adversarial Networks GANs [14]. AnoGAN [47] trains a GAN to generate samples according to the normal training data. At inference time given a test sample AnoGAN finds the latent representation that best reconstructs the sample. The anomaly score is based on the reconstruction error. AnoGAN is effective but not computationally efficient. Efficient GAN Based Anomaly Detection (EGBAD) addresses the performance issues of AnoGAN by adopting a Bidirectional GAN architecture [11]. In [1] they proposed to model a latent distribution obtained from a deep autoencoder using an auto-regressive network. [56] leverages GANs to learn the latent distribution of normal data and uses a perceptual loss for the detection of image abnormality. Our approach also builds on an architecture that combines autoencoders with GANs under the form of adversarial autoencoders as in Generative Probabilistic Novelty Detection (GPND) [37], but we build on the additional geometric properties of that architecture that were introduced in [4], and design an efficient procedure to make the likelihood training robust.

Despite their success, GAN-based approaches for anomaly detection suffer from several training issues such as mode collapse [53], non-convergence and instability that leads to oscillations during training, instead of a fixed-point convergence [45]. On the other hand, autoencoders based architectures are more stable and more convenient to train, but can overfit to a pass-through identity (null) function, and potentially reconstruct outliers when they share common features with the normal class [13]. To prevent this, regularization in the form of adding deliberate perturbation to the input data often takes place. [43] proposed the Adversarially Robust Autoencoder (ARAE), which works by forcing perceptually similar samples to be mapped closer in their latent representations. This is achieved by crafting adversarial examples that are perceptually similar to the input, but also have distant latent encoding from it. [3] trains the autoencoder to directly output the desired per-pixel measure of abnormality without first having to perform reconstruction. This is achieved by corrupting training samples with noise and then predicting how pixels need to be shifted to remove the noise. [18] proposed the One-Class Learned Encoder-Decoder (OLED) an adversarial framework for novelty detection in both images and videos. Rather than noise perturbations a Mask Module based on a convolutional autoencoder learns to cover the most important parts of images, and the a Reconstructor is another encoder-decoder that reconstructs the masked images. [6] introduced Adversarially Learned Continuous Noise (ALCN), which is an approach to maximally globally corrupt the input prior to denoising and verified its benefits for novelty detection.

The use of perturbed data has been studied in the area of adversarial attacks. [27] provides evidence that deep neural networks for supervised learning can be made resistant to adversarial attacks. They study the adversarial robustness of neural networks in terms of robust optimization. Other methods instead are based on adding priors for regularizing the supervised loss [21, 62]. Related to novelty detection, instead, [16] focusses on examining the adversarial impact to deep autoencoders, and introduces a defense strategy. Similarly, [26] introduces Principal Latent Space, a defense strategy that is applicable to autoencoders based novelty detection approaches, and that resembles PCA-based denoising done in the latent space.

Our approach also leverages perturbed data, but with the

goal of learning for the first time a likelihood function that is robust, as opposed to "robustifying" a supervised loss function, or proposing defense techniques that are transferrable. This should make the novelty test statistic less prone to errors in presence of perturbations within a certain set, but the approach would also help address the overfitting problem of the underlying autoencoder architecture.

3. A Prior for Robust Novelty Detection

We assume that x represents a quantity of interest, e.g., an image, which can be seen as a realization of a random variable X, distributed according to $p_X(x)$. Due to external factors, we assume we have access only to a modified version $\bar{x} = x + \delta$. Here δ could model noise, or an alteration due to an adversarial attack, or the sensing of unexpected or unknown data, as it normally happens in settings in-the-wild, when data is identified as not being *in distribution* (i.e., the distribution used for training the system), but in fact, it is *out of distribution*, and a subsequent appropriate action needs to be taken. Therefore, a central question to answer is whether or not the sample $\bar{x}$ was drawn from p_X. In general terms, the problem can be approached by performing the test

$$p_X(\bar{x}) = \begin{cases} \geq \gamma & \Longrightarrow \quad \text{Inlier} \\ < \gamma & \Longrightarrow \quad \text{Outlier} \end{cases} \tag{1}$$

where γ is a suitable application dependent threshold.

Methods that perform test (1) almost interchangeably use names like *novelty* [36, 37], *anomaly* [12, 34], *outlier*, or *out of distribution* [20, 49] detection, although subtle differences are often drawn [39]. Also, in this context, p_X does not really have the meaning of probability, but rather of *likelihood*, which means that it depends on a particular model that was learned from training data. Out of all the possible models, we propose to seek for one that exhibits *robustness* against a set $\mathcal{S}$ of predefined perturbations $\delta \in \mathcal{S}$. This means that if x belongs to the set of inliers $\mathcal{X}$, then it should be that $p_X(x + \delta | x \in \mathcal{X}) \geq \gamma$, and if x is an outlier, then it should be that $p_X(x + \delta | x \in \mathcal{X}^{\complement}) < \gamma$. Seeking for a robust model would allow the novelty detector to offer improved guarantees that it will be less affected by the attacks defined by the set of perturbations, which could be either intentional, or simply due to natural environmental causes.

From this discussion, we suggest that models that aim at performing test (1) could be made *robust* by including in their training a mechanism for *maximizing* the quantity

$$\begin{aligned} &E[\min_{\delta \in \mathcal{S}} p_X(x + \delta | x \in \mathcal{X}) - \gamma] + \\ &\quad E[\gamma - \max_{\delta \in \mathcal{S}} p_X(x + \delta | x \in \mathcal{X}^{\complement})] \ , \end{aligned} \tag{2}$$

where $E[\cdot]$ denotes expectation. The approach is based on robust optimization, which has also inspired the recent adversarial training methods for supervised learning [27]. However, rather than "robustifying" a classification loss function, (2) tries to make the test statistic p_X robust. Maximizing the first term of the *robust prior* (2) aims at ensuring that the worst attacks do not turn inliers into outliers, while the second term aims at ensuring that the worst attacks do not turn outliers into inliers.

Note that in (2) the threshold γ cancels out and it reduces to $E[\min_{\delta \in \mathcal{S}} p_X(x+\delta | x \in \mathcal{X})] - E[\max_{\delta \in \mathcal{S}} p_X(x+\delta | x \in \mathcal{X}^{\complement})]$. This suggests that maximizing (2) could be achieved by turning the two terms into a single fractional term like

$$\frac{E[\max_{\delta \in \mathcal{S}} p_X(x + \delta | x \in \mathcal{X}^{\complement})]}{E[\min_{\delta \in \mathcal{S}} p_X(x + \delta | x \in \mathcal{X})]} \ . \tag{3}$$

This new *fractional robust prior* (3) aims for the same goals as (2) when it is *minimized*, and it could be added to training losses as a regularizer. We note that for supervised learning, adversarial training by regularization is not new [21, 62], but to the best of our knowledge, it is new for unsupervised novelty or anomaly detection.

To solve the optimizations in the argument of the expectations in (2) and (3), one can take a projected gradient descent (PGD) approach by implementing the iteration

$$x^{t+1} = \Pi_{x+\mathcal{S}}(x^t + \alpha \nabla_x p_X(x)) \ , \tag{4}$$

where $\Pi_{x+\mathcal{S}}$ is a projection operator. On the other hand, in §5 we show that by leveraging the properties of our network architecture it turns out that such optimizations can be solved in closed form. To keep the paper self-contained and to introduce notation, in §4 we summarize the novelty detection approach that we build on top of, [4], while in §5 we describe how we make it robust by defining the set of perturbations $\mathcal{S}$ and enabling the training based on the prior (3).

4. Generative Probabilistic Novelty Detection

We summarize the formulation, properties, and training objective function of the novelty/anomaly test initially introduced in [4, 37]. Specifically, we assume that training data points $\mathcal{D} = \{x_1, \ldots, x_N\}$, where $x_i \in \mathbb{R}^m$, are sampled, possibly with noise ξ_i, from the model

$$x_i = f(z_i) + \xi_i \qquad i = 1, \cdots, N \ , \tag{5}$$

where z_i is defined in a *latent* space $\Omega \subset \mathbb{R}^n$. The mapping $f : \Omega \to \mathbb{R}^m$ defines $\mathcal{M} \equiv f(\Omega)$, which is a parameterized manifold of dimension n, with $n < m$. It is also assumed that the Jacobi matrix of f is full rank at every point of the manifold.

Given a new data point $\bar{x} \in \mathbb{R}^m$, the novelty test to assert whether $\bar{x}$ was sampled from model (5), is derived under a number of assumptions. Specifically, f is imposed to be an isometry, and in order to compute the test it is also necessary to estimate the latent representation $\bar{z}$ of $\bar{x}$. This is done by first applying to $\bar{x}$ an orthogonal projection $P_{\mathcal{M}}$ from

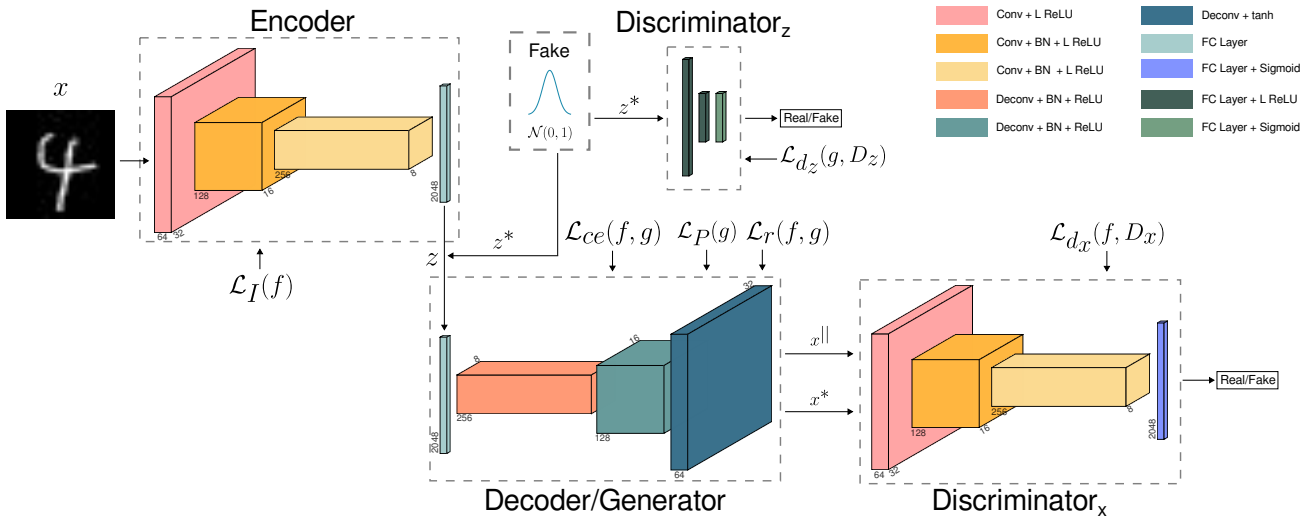

Figure 1. **Autoencoder architecture.** Overview of the architecture and losses of the Adversarial Autoenconder (AAE) [30] used for learning the maps f and g. While the backbone architecture is the same as in [4], here the major addition is the use of the robust prior (23). Some details of the architecture layers of the discriminators D_x and D_z are specified on the right. During training, the fake samples are generated from an n-dimensional normal distribution $\mathcal{N}(0,1)$. x^* represents a mapping of z^* onto the learned manifold $\mathcal{M}$, and $x^{||}$ is $f(z)$.

the ambient space onto $\mathcal{M}$, and then map the projection to the representation space via f^{-1}. This means that besides the manifold representation f, it is also necessary to learn a function g, defined as $g(x) \doteq f^{-1} \circ P_{\mathcal{M}}(x)$.

Mainly with the assumptions described above, given $\bar{x}$, in [4, 37] they compute the test statistic p_X in (1) as

$$p_X(\bar{x}) = p_Z(\bar{z}) p_{X^\perp}(\bar{x}^\perp) \,, \tag{6}$$

where $p_Z(z)$ is the probability distribution of the random variable Z, representing the latent space, and $\bar{x}^\perp = \bar{x} - f(\bar{z})$, represents the component of $\bar{x}$ that is orthogonal to the tangent space $\mathcal{T}$ of the manifold $\mathcal{M}$. Such space is defined as $\mathcal{T} = \text{span}(J_f(\bar{z}))$, with $J_f(\bar{z})$ being the Jacobi matrix computed at $\bar{z}$.

The distribution $p_Z(z)$ is learned from training data by fitting a parametric generalized Gaussian distribution. Instead, the distribution $p_{X^\perp}(x^\perp)$, is given by

$$p_{X^\perp}(\bar{x}^\perp) = \frac{\Gamma\left(\frac{m-n}{2}\right)}{2\pi^{\frac{m-n}{2}} \|\bar{x}^\perp\|^{m-n-1}} p_{\|X^\perp\|}(\|\bar{x}^\perp\|) \,, \tag{7}$$

where $\Gamma(\cdot)$ is the gamma function, and $\|\cdot\|$ denotes ℓ_2-norm. The distribution $p_{\|X^\perp\|}(\|x^\perp\|)$ is learned by computing the orthogonal projections of the training data, and histogramming the norms between data and projectons.

4.1. Manifold Learning

A major training task is the learning of the maps f, and g. They are modeled by an adversarial autoencoder. To satisfy the requirements, the Jacobian $J_f(z)$ will need to have orthonormal columns. This means that

$$J_f(z)^\top J_f(z) = I \,, \tag{8}$$

where I is the identity matrix. Moreover, if f is an isometry, then g should be such that

$$J_g(f(z)) J_g(f(z))^\top = I \,, \tag{9}$$

$$J_g(f(z)) = J_f(z)^\top \,. \tag{10}$$

where $J_g(x)$ denotes the Jacobi matrix of g.

To satisfy (8), (9), and (10), two priors are introduced. The first is the isometry loss $\mathcal{L}_I(f)$, which encourages (8), and is defined as

$$\mathcal{L}_I(f) = E\left[(\|J_f(z)u\| - 1)^2\right] \,, \tag{11}$$

where u is uniformly sampled from the unit-sphere of dimension $n-1$. The second prior is the pseudo-inverse loss $\mathcal{L}_P(g)$, which encourages (9), and is defined as

$$\mathcal{L}_P(g) = E\left[(\|u^\top J_g(x)\| - 1)^2\right] \,, \tag{12}$$

where, again, u is sampled from the same unit sphere. These priors are combined as $\mathcal{L}_{IAE}(f,g) = \mathcal{L}_I(f) + \mathcal{L}_P(g)$.

The adversarial autoencoder architecture is shown in Figure 1, which follows the design in [37]. One adversarial component encourages the distribution on the latent space, to be a normal distribution $\mathcal{N}(0,1)$. Another adversarial

component encourages the distribution of the output of the decoder to match the distribution of real data, i.e., the manifold $\mathcal{M}$. The adversarial losses are as follows

$$\mathcal{L}_{d_z}(g, D_z) = E[\log(D_z(\mathcal{N}(0,1)))] + E[\log(1 - D_z(g(x)))] \ , \tag{13}$$

$$\mathcal{L}_{d_x}(f, D_x) = E[\log(D_x(x))] + E[\log(1 - D_x(f(\mathcal{N}(0,1))))] \ , \tag{14}$$

To minimize the reconstruction error for an inlier input x it is used the cross-entropy loss $\mathcal{L}_{ce}(f,g) = -E_z[\log(p(f(g(x))|x))]$, where $\mathcal{L}_{ce}$ also encourages (10). See [17] for details, also on the implementation of the isometric priors above. We combine the losses that do not involve discriminators in $\mathcal{L}_a$

$$\mathcal{L}_a(f,g) = \lambda_I \mathcal{L}_{IAE}(f,g) + \mathcal{L}_{ce} \tag{15}$$

Where λ_I is a balancing hyperparameter. The final objective function becomes

$$\mathcal{L}(f,g,D_x,D_z) = \mathcal{L}_{d_x}(f,D_x) + \mathcal{L}_{d_z}(g,D_z) + \lambda_a \mathcal{L}_a(f,g) \ , \tag{16}$$

where λ_a sets the trade off between the losses with and without discriminators, and f and g are estimated as

$$\hat{f}, \hat{g} = \arg\min_{f,g} \max_{D_x, D_z} \mathcal{L}(f,g,D_x,D_z) \ . \tag{17}$$

5. Robust Likelihood Model

We now describe how we make the novelty detection method in §4 robust, based on the ideas described in §3. First, we assume that the set of admissible perturbations is an ϵ-ball, which means that $\mathcal{S} = \{\delta : \|\delta\| \leq \epsilon\}$. Next, we make the following simplifying assumptions

$$\min_{\|\delta\| \leq \epsilon} p_X(x+\delta) \approx p_X(x + \epsilon 1_x) \ , \tag{18}$$

$$\max_{\|\delta\| \leq \epsilon} p_X(x+\delta) \approx p_X(x - \epsilon 1_x) \ , \tag{19}$$

where $1_x = (x - f(z))/\|x - f(z)\|$ is a unit norm vector. Given the properties of the autoencoder defined by g and f, $f(z)$ is the orthogonal projection of x onto the manifold $\mathcal{M}$, and 1_x is perpendicular to the tangent plane $\mathcal{T}$. Therefore, (18) stems from the fact that it is reasonable to expect that the largest drop of the likelihood p_X will be due to a perturbation that moves x away from $\mathcal{M}$ the furthest possible, and this should happen along the direction orthogonal to $\mathcal{M}$. Similarly, (19) stems from the fact that we are expecting to observe the highest increase of p_X when x moves the closest towards $\mathcal{M}$. See Figure 2.

We stress that (18) and (19) are possible thanks to the properties of the autoencoder $g \circ f$. They provide a remarkable computational saving in that the two optimizations are solved in closed-form without requiring the use of PGD (4). Moreover, (18) and (19) also suggest a very efficient strategy for generating inliers and outliers data for training purposes, as explained below, which again does not require PGD.

Since data points are modeled according to (5), then we have that $x = f(z) + \nu 1_x$, where $\nu \doteq \|\xi\|$. Therefore, given (18), as a general recipe for *generating inliers* from x, we use the following expression

$$f(z) \pm (\nu + \epsilon) 1_x \ , \tag{20}$$

where the $\pm$ sign takes into account that inliers can be generated on both sides of the tangent plane $\mathcal{T}$. Similarly, given (19), the general recipe for *generating outliers* from x becomes

$$f(z) \pm (\nu - \epsilon) 1_x \ , \tag{21}$$

where the $\pm$ sign is introduced for the same reason as in (20). Figure 2 illustrates the generation process.

Note, however, that (20) should be used only if $f(z) \pm \nu 1_x \in \mathcal{X}$, which means it is an inlier. Similarly, (21) should be used only if $f(z) \pm \nu 1_x \in \mathcal{X}^{\complement}$, which means it is an outlier. Deciding whether $f(z) \pm \nu 1_x$ belongs to $\mathcal{X}$ or $\mathcal{X}^{\complement}$ is straightforward once we know the value ν_0 such that $p_X(f(z) \pm \nu_0 1_x) = \gamma$. From (6) and (7), ν_0 can be computed by solving numerically the equation

$$p_Z(z) \frac{\Gamma(\frac{m-n}{2})}{2\pi^{\frac{m-n}{2}} \nu_0^{m-n-1}} p_{\|X\|^{\perp}}(\nu_0) = \gamma \ . \tag{22}$$

It follows that $f(z) \pm \nu 1_x \in \mathcal{X}$ if $\nu \leq \nu_0$, and that $f(z) \pm \nu 1_x \in \mathcal{X}^{\complement}$ if $\nu > \nu_0$. See Figure 2.

5.1. Robust Prior

From the previous discussion, we note that given a training dataset composed of only inliers, we can still generate synthetic outliers according to how we choose ν. In particular, we propose to randomly sample inliers by assuming that $\nu \sim \mathcal{U}([0, \nu_0))$, which means ν is uniformly distributed in the interval $[0, \nu_0)$. Therefore, inliers come from the region closer to $\mathcal{M}$. Outliers instead, are sampled by assuming that $\nu \sim \mathcal{E}(\lambda)$, which means that ν is exponentially distributed with rate parameter λ, and an offset ν_0 is also added.

In essence, we propose to add to the objective function (16) the following robust prior

$$\mathcal{L}_r(f,g) = \frac{E_{x\sim\mathcal{D}}[E_{\nu\sim\mathcal{E}}[p_X(f(z) \pm (\nu + \nu_0 - \epsilon)1_x)]]}{E_{x\sim\mathcal{D}}[E_{\nu\sim\mathcal{U}}[p_X(f(z) \pm (\nu + \epsilon)1_x)]]} \ . \tag{23}$$

The final procedure for the training of the *robust likelihood novelty detection (RLND)* model is summarized in Algorithm 1.

6. Experiments

We present the evaluation of the proposed robust likelihood novelty detection (RLND) method. We compare

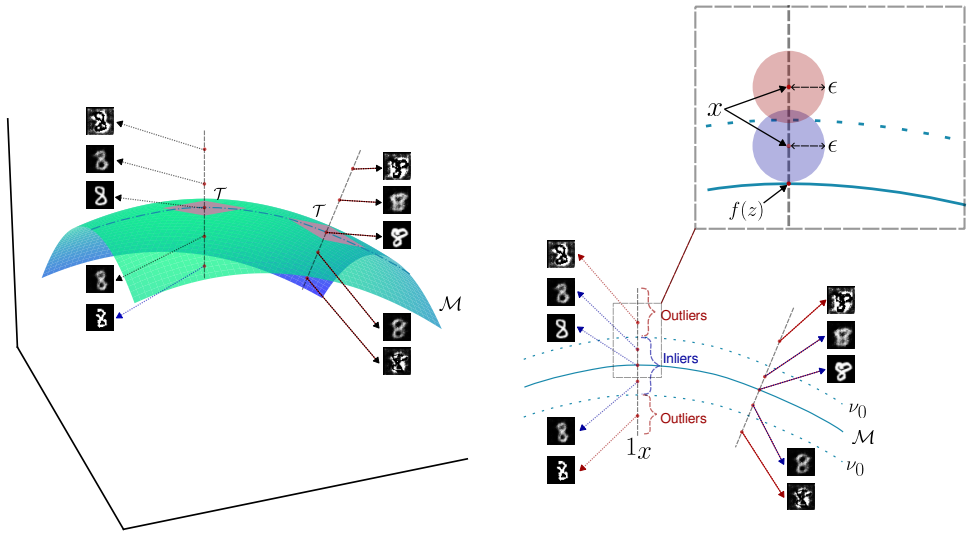

Figure 2. **Generation of inliers and outliers.** A point x is an *inlier* or an *outlier* depending on whether its distance from the manifold $\mathcal{M}$ is below or above a threshold ν_0. The orthogonal projection of x onto $\mathcal{M}$ is $f(z)$. x can be perturbed within a radius ϵ. The strongest perturbation for an inlier/outlier occurs in the outward/inward direction orthogonal to the tangent plane $\mathcal{T}$.

Algorithm 1 Robust Likelihood Novelty Detection

Input: Training dataset $\mathcal{D} = \{x_i\}_{i=1}^N$.
Parameters: Minibatch size M; radius ϵ; rate λ
Train f and g and obtain initial weights from (17)
Obtain γ from the validation dataset
repeat
  Sample a minibatch $\{x_{i_j}\}_{j=1}^M$
  for $j \in \{1, \dots, M\}$ **do**
    Compute ν_{0,i_j} by solving (22)
    $\nu \leftarrow$ draw from $\mathcal{U}([0, \nu_{0,i_j}))$
    ▷ *Inlier generation; randomly pick $+$ or $-$* ◁
    $x_j^I \leftarrow f(z_{i_j}) \pm (\nu + \epsilon) 1_{x_{i_j}}$
    $\nu \leftarrow$ draw from $\mathcal{E}(\lambda)$
    ▷ *Outlier generation; randomly pick $+$ or $-$* ◁
    $x_j^O \leftarrow f(z_{i_j}) \pm (\nu + \nu_{0,i_j} - \epsilon) 1_{x_{i_j}}$
  ▷ *Use the inlier batch $\{x_j^I\}$ for the following* ◁
  Maximize $\mathcal{L}_{d_x}$ (14) by updating weights of D_x
  Minimize $\mathcal{L}_{d_x}$ (14) by updating weights of f
  Maximize $\mathcal{L}_{d_z}$ (13) by updating weights of D_z
  ▷ *Use the inlier batch $\{x_j^I\}$ and outlier batch $\{x_j^O\}$ for the following* ◁
  Minimize $\mathcal{L}_a$ (15), $\mathcal{L}_{d_z}$ (14), and $\mathcal{L}_r$ (23), by updating weights of g and f.
  ▷ *Note that only $\mathcal{L}_r$ uses both inlier and outlier batches* ◁
until Convergence

RLND with state-of-the-art methods by using common benchmark datasets for the unsupervised novelty detection task, and we follow the same protocol as in [4] to maintain consistency across all experiments. We utilize two key metrics: the F_1 score and the area under the ROC (AUROC). These metrics provide a comprehensive assessment of the performance of our approach.

In each experiment, the datasets are partitioned into training, validation, and testing sets using a random split. Specifically, we allocate 60% of the data for training, where instances from each class are randomly sampled, and 20% for validation. The remaining 20% are reserved for testing.

6.1. Datasets

We utilize three benchmark datasets commonly used for novelty and anomaly detection, namely MNIST, Fashion-MNIST, and Coil-100.

MNIST [22] is composed of 70,000 28×28 handwritten single digits from 0 to 9.

Fashion-MNIST [59], similar to MNIST, contains 70,000 28×28 grayscale images of 10 fashion product categories.

Coil-100 [35] is a dataset of 7,200 color images with 100 object classes. For each of 100 objects, pictures were taken in different poses, 5 degrees apart from one another, resulting in 72 images for each object.

6.2. Implementation Details

We implemented Algorithm 1, where the first step trains a GPNDI model. We refer to [4] for picking the parameters λ_I and λ_a. Instead, when also $\mathcal{L}_r$ is minimized, we weight

Table 1. F_1 scores on MNIST [22]. Inliers are taken to be images of one category, and outliers are randomly chosen from other categories. All results are averages from a 5-fold cross validation.

% of outliers	$\mathcal{D}(\mathcal{R}(X))$ [41]	$\mathcal{D}(X)$ [41]	LOF [8]	DRAE [58]	GPND [37]	GPNDI [4]	**RLND** (Ours)
10	0.97	0.93	0.92	0.95	0.983	0.984	**0.990**
20	0.92	0.90	0.83	0.91	0.971	0.976	**0.986**
30	0.92	0.87	0.72	0.88	0.961	0.968	**0.980**
40	0.91	0.84	0.65	0.82	0.950	0.960	**0.977**
50	0.88	0.82	0.55	0.73	0.939	0.953	**0.974**

Table 2. Results on Fashion-MNIST [59]. F_1 scores where inliers are taken to be images of one category, and outliers are randomly chosen from other categories.

% of outliers	10	20	30	40	50
GPND [37]	0.968	0.945	0.917	0.891	0.864
GPNDI [4]	0.974	0.953	0.930	0.904	0.873
RLND (Ours)	**0.986**	**0.977**	**0.970**	**0.961**	**0.954**

it by a hyperparameter λ_r, which is set to 0.001. In all the experiments the latent space size n, is set to 16, since it has been reported to yield the highest F_1 score on the validation sets [4, 37].

The initial GPNDI model is then further trained for 30 more epochs using an NVIDIA RTX A6000 GPU and the ADAM optimizer. For all datasets we use a batch size $M = 128$. Instead, to ensure optimal convergence, the learning rates are set to 0.0002 for both MNIST and Fashion-MNIST, while for COIL-100, the learning rate is set to 0.0003.

In Algorithm 1, we set the rate parameter λ to 5.0 in all the experiments, while the radius ϵ varies with the dataset. Specifically, the training ϵ is set to $\epsilon = 0.5 \times \nu_0$, where ν_0 here is intended as averaged over the inlier training samples. This choice ensures that a sample residing at the same distance from the manifold and the decision boundary will remain inside the inliers set $\mathcal{X}$ even after the largest admissible perturbation. For MNIST and Fashion-MNIST, we used ϵ values that were averaged over all choices of inlier manifolds, and they are 2.4 and 3.0, respectively. For COIL-100, ϵ was determined based on the random selection of inliers. This choice of ϵ tailors the model to the specific inlier manifold being learned.

6.3. Results without Attacks

MNIST dataset. For the MNIST dataset, we compose five random balaced data splits to evaluate the performance of our approach. We use three splits for training, reserving one split for validation and one for testing. The value of γ that yields the highest F_1 score on the validation set is then employed during the testing phase. We designate each digit as an inlier, while the remaining digit samples are selected to generate outlier percentages ranging from 10% to 50%. This allows us to assess the robustness of our approach across different levels of novelty in the dataset. We present

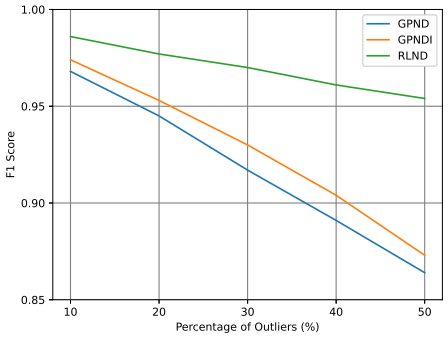

Figure 3. Results on FashionMNIST dataset.

these results in Table 1 and illustrate them graphically in Figure 4. The comparative evaluation against GPND, GPNDI and other methods, highlights that a consistent improvement is achieved. This suggests that the additional training based on the robust prior (23), which is the foundation of RLND, leads to better performance on this standard benchmark.

Fashion-MNIST dataset. We maintain consistency with the protocol followed for the MNIST dataset when conducting experiments on Fashion-MNIST. The results of these experiments are presented in Table 2, and visually depicted in Figure 3. It can be seen that the robust training of RLND, although designed for a specific class of perturbations, it can lead to a remarkable increase in performance even when data is not necessarily undergoing the same type of perturbations. This result further validates the effectiveness of RLND.

Coil-100 dataset. Similar to previous datasets, we adopt a 5-fold cross-validation approach for evaluating the performance on the Coil-100 dataset. However, in this case, we utilize four splits for training and reserve one split for testing. The optimal value of γ is determined based on the training set, ensuring the best possible performance during testing. For each experiment, we randomly select 1, 4, or 7 classes as inliers, while considering the remaining classes as outliers. The outliers are included at percentages of 50%, 25%, and 15%, respectively. The results are presented in Table 3. Our RLND approach consistently outperforms GPNDI in all cases, further confirming its superior performance, and the usefulness of a robust approach. Furthermore, we note that RLND is able to match or surpass the F_1 scores of R-graph [61]. This is significant because R-graph is based on a large pre-trained VGG network, whereas we are training from scratch a very small autoencoder architecture with a limited number of samples, which is around 70 per class.

6.4. Results with Attacks

In this experiment, our primary objective is to evaluate the robustness of our model against ϵ-attacks. ϵ-attacks involve perturbing the input data point x along the 1_x direction from the manifold's projection. We conduct this test on the

Table 3. Results on Coil-100. Inliers are taken to be images of one, four, or seven randomly chosen categories, and outliers are randomly chosen from other categories (at most one from each category).

	OutRank [32, 33]	CoP [38]	REAPER [23]	OutlierPursuit [60]	LRR [25]	DPCP [55]	ℓ_1 thresholding [50]	R-graph [61]	GPND [37]	GPNDI [4]	**RLND** (Ours)
Inliers: **one** category of images , Outliers: 50%											
AUROC	0.836	0.843	0.900	0.908	0.847	0.900	0.991	**0.997**	0.968	0.984	0.990
F_1	0.862	0.866	0.892	0.902	0.872	0.882	0.978	**0.990**	0.979	0.894	0.989
Inliers: **four** category of images , Outliers: 25%											
AUROC	0.613	0.628	0.877	0.837	0.687	0.859	0.992	**0.996**	0.945	0.960	0.980
F_1	0.491	0.500	0.703	0.686	0.541	0.684	0.941	**0.970**	0.960	0.953	**0.970**
Inliers: **seven** category of images , Outliers: 15%											
AUROC	0.570	0.580	0.824	0.822	0.628	0.804	0.991	**0.996**	0.919	0.950	0.985
F_1	0.342	0.346	0.541	0.528	0.366	0.511	0.897	0.955	0.941	0.964	**0.981**

Table 4. Precision, Recall, F_1 and AUROC measures for various ϵ-attacks on the MNIST test set.

	GPNDI					**RLND** (Ours)				
ϵ	0.0	0.5	1.0	2.0	3.0	0.0	0.5	1.0	2.0	3.0
Precision	0.971	0.946	0.885	0.704	0.576	0.971	0.954	0.924	0.745	0.605
Recall	0.951	0.925	0.902	0.905	0.950	0.977	0.946	0.915	0.908	0.950
F_1	0.961	0.935	0.893	0.781	0.706	0.974	0.950	0.919	0.806	0.726
AUROC	0.99	0.979	0.948	0.781	0.470	0.993	0.985	0.966	0.850	0.581

Table 5. Precision, Recall, F_1 and AUROC measures for various ϵ-attacks on the Fashion-MNIST test set.

	GPNDI					**RLND** (Ours)				
ϵ	0.0	0.5	1.0	2.0	3.0	0.0	0.5	1.0	2.0	3.0
Precision	0.939	0.905	0.856	0.741	0.608	0.954	0.932	0.912	0.840	0.744
Recall	0.931	0.931	0.924	0.909	0.958	0.956	0.947	0.934	0.915	0.917
F_1	0.937	0.921	0.886	0.811	0.737	0.954	0.939	0.922	0.881	0.814
AUROC	0.979	0.921	0.935	0.834	0.647	0.987	0.980	0.968	0.923	0.836

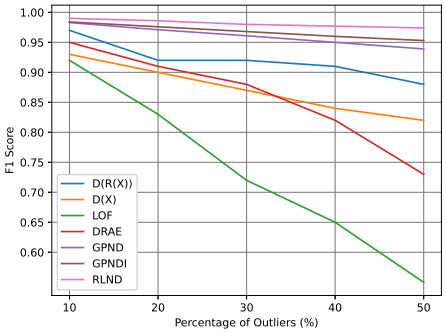

Figure 4. Results on MNIST [22] dataset.

MNIST and Fashion-MNIST datasets. The attack on a testing inlier sample x is generated by adding a perturbation $\epsilon 1_x$. The attack on a testing outlier sample x is generated by subtracting a perturbation $\epsilon 1_x$.

To assess the model's performance under attack, we measure precision, recall, F_1 score, and AUROC with varying values of ϵ. For both datasets, GPNDI and RLND were trained as described in §6.2 and §6.3. In particular, for a given dataset, the training ϵ value is the same for every ϵ-attack. The results are reported in Table 4 and Table 5. We note that RLND outperforms GPNDI according to all the metrics, conditions, in both datasets, and by a significant margin. This is a very encouraging result, which supports the major contribution of the proposed approach. We further note the quick and strong deterioration in performance of the baseline approach GPNDI, as ϵ grows, which is clearly due to the fact that GPNDI was not robustly trained to respond to these attacks. RLND instead, demonstrates a much smaller rate of deterioration.

7. Conclusion

In this work we introduced a new prior for learning a likelihood model for novelty or anomaly detection that is robust to a predefined set of perturbations. We then integrated this prior with GPNDI, an existing method for novelty detection, which is based on computing the likelihood of the input samples. The integration, referred to as Robust Likelihood Novelty Detection (RLND), is computationally efficient, and entails a training refinement of the initial model, by optimizing an updated loss with minibatches of sampled synthetically generated inliers and outliers. Our initial results reveal that integrating the robust prior leads to a clear performance improvement over the baseline method, when both are tested on the benchmark datasets MNIST, Fashion-MNIST, and COIL-100. This means that the prior is a beneficial regularizer when perturbations, or attacks are not present. Furthermore, when both the baseline model, GPNDI, and the robust model, RLND, are subject to ϵ-attacks, we observed that the proposed method can cope with the attacks significantly better than the baseline. While these are very encouraging results, in future work we plan to address other areas of investigation that we currently left out, such as testing our approach against other types of adversarial attacks, like those based on PGD.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grants No. 1920920, 2125872 and 2223793.

References

[1] Davide Abati, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Latent space autoregression for novelty detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 481–490, 2019. 2

[2] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010. 2

[3] Philip A Adey, Samet Akçay, Magnus JR Bordewich, and Toby P Breckon. Autoencoders without reconstruction for textural anomaly detection. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021. 2

[4] R. Almohsen, M. R. Keaton, D. A. Adjeroh, and G. Doretto. Generative probabilistic novelty detection with isometric adversarial autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 2002–2012. IEEE, June 2022. 1, 2, 3, 4, 6, 7, 8

[5] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special lecture on IE*, 2(1):1–18, 2015. 2

[6] Jack W Barker, Neelanjan Bhowmik, Yona Falinie A Gaus, and Toby P Breckon. Robust semi-supervised anomaly detection via adversarially learned continuous noise corruption. In *18th International Conference on Computer Vision Theory and Applications*, 2023. 2

[7] Arslan Basharat, Alexei Gritai, and Mubarak Shah. Learning object motion patterns for anomaly detection and improved object detection. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008. 2

[8] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *ACM sigmod record*, volume 29, pages 93–104. ACM, 2000. 7

[9] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):1–37, 2011. 2

[10] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360*, 2018. 2

[11] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016. 2

[12] Eleazar Eskin. Anomaly detection over noisy data using learned probability distributions. In *In Proceedings of the International Conference on Machine Learning*. Citeseer, 2000. 2, 3

[13] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1705–1714, 2019. 2

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2

[15] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015. 1

[16] Adam Goodge, Bryan Hooi, See Kiong Ng, and Wee Siong Ng. Robustness of autoencoders for anomaly detection under adversarial impact. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 1244–1250, 2021. 2

[17] Amos Gropp, Matan Atzmon, and Yaron Lipman. Isometric autoencoders. *arXiv preprint arXiv:2006.09289*, 2020. 5

[18] John Taylor Jewell, Vahid Reza Khazaie, and Yalda Mohsenzadeh. One-class learned encoder-decoder network with adversarial context masking for novelty detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3591–3601, 2022. 2

[19] JooSeuk Kim and Clayton D Scott. Robust kernel density estimation. *Journal of Machine Learning Research*, 13(Sep):2529–2565, 2012. 2

[20] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Why normalizing flows fail to detect out-of-distribution data. June 2020. 3

[21] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *International Conference on Learning Representations*, 2017. 2, 3

[22] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998. 6, 7, 8

[23] Gilad Lerman, Michael B McCoy, Joel A Tropp, and Teng Zhang. Robust computation of linear models by convex relaxation. *Foundations of Computational Mathematics*, 15(2):363–410, 2015. 8

[24] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008. 2

[25] Guangcan Liu, Zhouchen Lin, and Yong Yu. Robust subspace segmentation by low-rank representation. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 663–670, 2010. 8

[26] Shao-Yuan Lo, Poojan Oza, and Vishal M Patel. Adversarially robust One-Class novelty detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(4):4167–4179, Apr. 2023. 2

[27] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 1, 2, 3

[28] Alireza Makhzani and Brendan Frey. K-sparse autoencoders. *arXiv preprint arXiv:1312.5663*, 2013. 2

[29] Alireza Makhzani and Brendan J Frey. Winner-take-all autoencoders. *Advances in neural information processing systems*, 28, 2015. 2

[30] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015. 4

[31] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *Artificial Neural Networks and Machine Learning–ICANN 2011: 21st International Conference on*

Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I 21, pages 52–59. Springer, 2011. 2

[32] HDK Moonesignhe and Pang-Ning Tan. Outlier detection using random walks. In *Tools with Artificial Intelligence, 2006. ICTAI'06. 18th IEEE International Conference on*, pages 532–539. IEEE, 2006. 8

[33] HDK Moonesinghe and Pang-Ning Tan. Outrank: a graph-based outlier detection framework using random walk. *International Journal on Artificial Intelligence Tools*, 17(01):19–36, 2008. 8

[34] Benjamin Nachman and David Shih. Anomaly detection with density estimation. *Phys. Rev. D*, 101(7):075042, Apr. 2020. 3

[35] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (coil-20). 1996. 6

[36] Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. Ocgan: One-class novelty detection using gans with constrained latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 3

[37] Stanislav Pidhorskyi, Ranya Almohsen, and Gianfranco Doretto. Generative probabilistic novelty detection with adversarial autoencoders. *Advances in neural information processing systems*, 31, 2018. 1, 2, 3, 4, 7, 8

[38] Mostafa Rahmani and George K Atia. Coherence pursuit: Fast, simple, and robust principal component analysis. *IEEE Transactions on Signal Processing*, 65(23):6260–6275, 2016. 8

[39] Lukas Ruff, Jacob R Kauffmann, Robert A Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G Dietterich, and Klaus-Robert Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 109(5):756–795, 2021. 1, 2, 3

[40] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR, 2018. 1, 2

[41] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. Adversarially learned one-class classifier for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3379–3388, 2018. 7

[42] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, page 4. ACM, 2014. 2

[43] Mohammadreza Salehi, Atrin Arya, Barbod Pajoum, Mohammad Otoofi, Amirreza Shaeiri, Mohammad Hossein Rohban, and Hamid R Rabiee. Arae: Adversarially robust training of autoencoders improves novelty detection. *Neural Networks*, 144:726–736, 2021. 2

[44] Mohammadreza Salehi, Hossein Mirzaei, Dan Hendrycks, Yixuan Li, Mohammad Hossein Rohban, and Mohammad Sabokrou. A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: Solutions and future challenges. *arXiv preprint arXiv:2110.14051*, 2021. 2

[45] Divya Saxena and Jiannong Cao. Generative adversarial networks (gans) challenges, solutions, and future directions. *ACM Computing Surveys (CSUR)*, 54(3):1–42, 2021. 2

[46] Walter J. Scheirer, Anderson Rocha, Archana Sapkota, and Terrance E. Boult. Towards open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 35, July 2013. 1

[47] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *Information Processing in Medical Imaging: 25th International Conference, IPMI 2017, Boone, NC, USA, June 25-30, 2017, Proceedings*, pages 146–157. Springer, 2017. 2

[48] Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for novelty detection. *Advances in neural information processing systems*, 12, 1999. 2

[49] Joan Serrà, David Álvarez, Vicenç Gómez, Olga Slizovskaia, José F Núñez, and Jordi Luque. Input complexity and out-of-distribution detection with likelihood-based generative models. In *International Conference on Learning Representations*, Feb. 2022. 3

[50] Mahdi Soltanolkotabi, Emmanuel J Candes, et al. A geometric analysis of subspace clustering with outliers. *The Annals of Statistics*, 40(4):2195–2238, 2012. 8

[51] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014. 1

[52] David MJ Tax and Robert PW Duin. Support vector data description. *Machine learning*, 54:45–66, 2004. 2

[53] Hoang Thanh-Tung and Truyen Tran. Catastrophic forgetting and mode collapse in gans. In *2020 international joint conference on neural networks (ijcnn)*, pages 1–10. IEEE, 2020. 2

[54] Alexander Tong, Guy Wolf, and Smita Krishnaswamyt. Fixing bias in reconstruction-based anomaly detection with lipschitz discriminators. In *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2020. 2

[55] Manolis C Tsakiris and René Vidal. Dual principal component pursuit. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 10–18, 2015. 8

[56] Nina Tuluptceva, Bart Bakker, Irina Fedulova, and Anton Konushin. Perceptual image anomaly detection. In *Pattern Recognition: 5th Asian Conference, ACPR 2019, Auckland, New Zealand, November 26–29, 2019, Revised Selected Papers, Part I*, pages 164–178. Springer, 2020. 2

[57] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008. 2

[58] 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*, pages 1511–1519, 2015. 7

[59] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 6, 7

[60] Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Robust pca via outlier pursuit. In *Advances in Neural Information Processing Systems*, pages 2496–2504, 2010. 8

[61] Chong You, Daniel P Robinson, and René Vidal. Provable self-representation based outlier detection in a union of subspaces. *arXiv preprint arXiv:1704.03925*, 2017. 7, 8

[62] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7472–7482. PMLR, 2019. 2, 3

[63] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 665–674, 2017. 2