Adversarial Regularized Reconstruction for Anomaly Detection and Generation

Anonymous

***Abstract*—We propose ARN, a semisupervised anomaly detection and generation method based on adversarial reconstruction. ARN exploits a regularized autoencoder to optimize the reconstruction of variants of normal examples with minimal differences, that are recognized as outliers. The combination of regularization and adversarial reconstruction helps to stabilize the learning process, which results in both realistic outlier generation and substantial detection capability. Experiments on several benchmark datasets show that our model improves the current state-of-the-art by valuable margins because of its ability to model the true boundaries of the data manifold.**

***Index Terms*—Anomaly Detection, Outlier Detection, Anomaly Generation, Outlier Generation, Generative Adversarial Networks, Variational Autoencoders**

I. INTRODUCTION

Anomaly detection is a prominent research topic in data mining and machine learning that aims at discovering unexpected elements in data populations, with relevant applications in several fields such as smart manufacturing, healthcare, security and finance. Historically, this research task has been extensively investigated and several methods have been proposed that find outliers based on either statistical modeling or spatial proximity [1]. In general, approaches to outlier detection can be classified as supervised, semi-supervised, and unsupervised.

Supervised methods exploit the availability of a labeled data set, containing observations already labeled as normal and abnormal, in order to build a model for the normal class. Since usually normal observations are the great majority, these data sets are unbalanced and specific classification techniques must be designed to deal with the presence of rare classes. Extreme class imbalance can hamper the discovery of local patterns characterizing rare classes, thus impeding the learning of effective models.

Non-supervised methods overcome these limitations as they do not require prior information concerning the anomalous examples. Semi-supervised methods typically assume that only normal examples are given. The goal is hence to find a partitioning of the domain space into dense accepting regions, containing the normal objects, and sparse rejecting regions, containing all the other objects significantly deviating from normality [2]. By contrast, unsupervised methods make no assumption on the class distribution and search for outliers in an unlabelled data set by assigning to each example a score which reflects its degree of abnormality.

In this paper we focus on semi-supervised anomaly detection with a slightly different objective: By exploiting only the examples labeled as normal, can we build a model that accurately characterizes both normal and abnormal behaviors? In other words, for a given example, what is the core set of relevant features characterizing the decision boundaries between normality and outlierness? This formulation resembles the zero-shot learning approach that is gaining popularity in computer vision and natural language processing [3]. The focus is hence on methods that can explain the data by generating feature representations of both seen and unseen classes, which can hence be exploited to build a classifier capable of discriminating among them. Probabilistic generative models are the basic tool for achieving this.

Generative models for anomaly detection, based on latent representations, are gaining substantial attention in the current literature [4]–[8], due to their capabilities in modeling the hidden causal relationships that ultimately characterize data. The expressive representation schemes offered by deep networks [9], combined with sophisticated yet effective learning mechanisms based on stochastic backpropagation, approximate Bayesian inference [10] and adversarial learning [11], make these models extremely flexible and accurate in describing the properties of the data.

Despite their flexibility, the approaches proposed in the current literature are still unable to supply realistic outlying properties that can support the detection process. Typically, generative methods tend to model the domain space via biased probability distributions; here, anomalies can be regarded as samples lying in low-density regions within such a probability space. However, within complex manifolds, overgeneralization can occur [12], thus hindering the capability to detect realistic outliers: in such cases, only anomalies with dramatically altered data properties turn out to be easily identifiable. Generative adversarial learning methods mitigate this issue with their capability to accurately reconstruct portions of the true distribution. Still, they exhibit two potential shortcomings. On one side, mode collapse and dropping can prevent a faithful reconstruction [13], with the result that normal samples can be deceived as outliers. On the other side, the generated adversarial samples tend to overlap the true distribution [14], with the consequence that the resulting discriminator exhibits limited detection abilities besides trivial outliers. The problem hence becomes: how to generate reliable outliers that can support the discriminator in devising the actual boundaries of a complex data manifold?

We claim that an effective anomaly detection and generation strategy can be obtained by (i) providing an efficient exploration of the data manifold, and (ii) generating, for each available normal sample, its abnormal counterpart. In practice, we aim at mapping each sample in a suitable latent feature representation space, from which an alternative reconstruction can be obtained with minimal but substantial differences from the original sample. The mapping within the latent space can be obtained in a controlled way by exploiting simple yet effective regularization schemes [10], [15]. At the same time, the adversarial reconstruction can enable the generation of data that is still consistent with the underlying manifold, while at the same time relating the latent representation to abnormal behavior.

Our contributions can hence be summarized as follows:

- We propose a framework based on the combination of variational autoencoders and adversarial learning with the objective of generating realistic outliers supporting the learning of an outlier detector. The framework relies essentially on normal data but can be easily extended to also take into account a limited amount of supervision. We discuss different modeling alternatives for tackling this task through the adoption of a neural network architecture that models latent dependencies at different abstraction levels.
- We evaluate the proposed framework on several benchmark datasets, by showing that: (a) the generated outliers are realistic, as they resemble the original data and still exhibit some specific features that the discriminator can recognize; (b) the resulting anomaly detector is competitive with the state of the art, robust to noise and capable of taking advantage of the efficient exploration of the entire data manifold in the learning process.

The rest of the paper is organized as follows. Section II discusses the recent contributions in the current literature and provides a systematic review of the approaches related to our task of interest. Section III discusses the mathematical details of our proposal. The effectiveness of the proposed model is illustrated in section IV, and pointers to future developments are discussed in section V.

II. Related Work

We structure the analysis of the literature by considering the tasks of outlier detection and generation. The former has been extensively studied in the literature [16] and can be characterized as a prediction problem: given an instance from a specific domain space, the objective is to score its anomaly likelihood. The latter is relatively new and has gained attention with the recent spread of deep generative models [10], [11], [17].

A. Outlier Detection

Anomaly detection is a challenging task due to the underlying class imbalance. To overcome this problem, most of the proposed solutions are based on unsupervised or semi-supervised approaches. Traditional non-supervised approaches rely on one-class classification (e.g., One-class Support Vector Machines [2]), distance-metrics (e.g. Isolation Forest [18]) or nearest neighbor algorithms [19]. Recently, deep anomaly detection has emerged as a critical direction [9]. In particular, autoencoders [20] have been extensively used for unsupervised anomaly detection based on deep learning [21]–[25].

An autoencoder is a neural network that learns low dimensional representations of the input data and, at the same time, its reconstruction from such a reduced encoding that is as close as possible to its original input. The core feature is the capability of devising encodings that ignore the "signal noise". As a consequence, they can be naturally employed for anomaly detection: Normal examples tend to map back to themselves, while anomalous tuples tend to produce divergent reconstructions. In practice, for a given example x, the outlierness score is given by the reconstruction error $\|x - D(E(x))\|$, where E and D represents the encoder and decoder components of the autoencoder architecture.

Within an autoencoding framework, outlierness is typically established on the basis of a pre-defined threshold T: If the reconstruction error is higher than T the sample is labeled as outlier. Tian, Zhou, Fan, *et al.* [26] propose a (weakly) supervised anomaly detection approach that allows to identify anomalies without the need for T. In particular, their model is built upon an autoencoder which use two decoders, namely *inlier decoder* (D_{in}) and *outlier decoder* (D_{out}). The former performs the reconstruction for inlier samples, while the latter focuses on outlier samples. Both decoders work in a competitive way as they both associate a reconstruction score for the input samples. Unlabelled data are fed in both decoders and they are labeled as either outliers or inliers, based on the lowest reconstruction error.

Besides reconstruction error, an alternative emerging research direction is the adoption of Generative Adversarial Networks (GANs) [4] to directly embed the detection process within a generative framework. GANs [11] estimate generative models via an adversarial process, in which two models are trained simultaneously. A generator G aims at capturing the data distribution, while a discriminator D aims at estimating the probability that an example came from the training data (i.e. real data) rather than from the generated data (i.e. fake data). To learn the generative distribution p_G over the data, a prior noise distribution is defined and then a mapping from the prior noise distribution to the data space is learned. The discriminator output is a single scalar value and it can be interpreted as probability that the input sample came from the real data rather than p_G.

To the best of our knowlege, the first approach that combines generative adversarial networks and outlier detection is AnoGAN [27]. The model aims at learning a mapping from the latent space to realistic (normal) samples. The mapping procedure is defined as an iterative process: The goal is to find a point z in the latent space corresponding to the generated sample that is most similar to the input sample. The similarity is computed as a combination of residual and feature matching loss [28] and represents the anomaly score. The mapping is devised as a sequence of backpropagation steps that make the overall detection procedure inefficient. To overcome this limitation, some variants [7], [29], [30] are proposed. For example, ALAD [7] represents an improvement that, in addition to generator and discriminator, exploits an encoder to refine the detection capabilities by discriminating the pair $(x, E(x))$ versus $(G(z), z)$. The underlying architecture also tries to overcome the cycle-consistency problem [31], suffered by architectures based on BiGAN [32], like [30].

GANomaly [5] combines adversarial learning and latent representation through an encoding-decoding framework. The model is composed by a generator and a discriminator, but the generator consists of three components: an encoder-decoder that maps the input samples into a latent space and back, and a further encoder that is used to map the reconstruction of the decoder in the latent space. The anomaly score is defined as the distance between the latent representation of the input sample and that of the reconstruction. Akçay, Atapour-Abarghouei, and Breckon [33] propose an extension of GANomaly particularly suited for input data represented by images. In such case in fact, the adoption of skip connections can substantially improve the reconstruction and consequently boost the detection abilities.

ADAE [34] models both the generator and the discriminator as autoencoders. The reconstruction error of the discriminating autoencoder represents the anomaly score. The claimed advantage is that this choice allows the better split between normal and anomalous scores, leading to superior performance.

Finally, a recent line of research is exploring the adoption of ensemble architectures based either on autoencoders [35] or GANs [36]. In general, ensembles can efficiently combine baseline models and thus better model the distribution of normal data. Ensembles are particularly effective with GANs, where a group of generators and a group of discriminators can be trained together, so that every generator gets feedback from multiple discriminators, and vice versa.

B. Outlier Generation

The generation of artificial outliers can serve the double purpose of testing outlier detection algorithms and aiding the training phase. For example, the class imbalance can be solved by generating artificial outlier that can be exploited to refine the detection phase for real outliers.

In principle, probabilistic generative models can be easily adapted to produce outliers, by sampling from low density regions. For example, Laptev [8] proposes an approach based on variational autoencoders to generate synthetic time series with anomalies. The idea is to learn the latent space representation of real data and then to generate anomalies by sampling from the outlier region of the latent space. The problem with such a naive approach is that, when the original data is characterized by complex manifolds, overgeneralization is likely to weaken the generation, as discussed in the introduction.

Rizzo, Pang, Chen, *et al.* [6] propose a (weakly) supervised framework based on the detection and generation of outliers, called WALDO (Wasserstein Autoencoder for Learning the Distribution of Outliers) that combines the approach outlined in [26] (discussed above) with Wasserstein autoencoders [17]. The basic assumption is that data is generated from an unknown and unlabeled mixture of inlier and outlier distributions $(P_X^u = (1-v)P_X^i + vP_X^o)$ and the goal is to learn generating distributions P_G^i and P_G^o which minimize $W_p(P_X^i, P_G^i)$ and $W_p(P_X^o, P_G^o)$. The approach requires that both examples from P_X^u and P_X^i are provided in the training phase.

FenceGAN [14] extends the basic framework of GANs by observing that the generated adversarial samples tend to overlap the true distribution. As a consequence, they propose a modification of the underlying adversarial framework to devise a generator capable of generating samples lying on the boundaries of the data distribution. The resulting learning process yields a discriminator specifically tailored for "difficult" outliers. The problem with such an approach is that the boundary is parameterized by a threshold representing the discrimination uncertainty. The latter can be domain-dependent and as a consequence tuning the relative threshold can be difficult.

III. Adversarial Reconstruction Networks

We structure our approach within a probabilistic framework where, given x, y with $x \in \mathcal{D}$ and $y \in \{0, 1\}$, we would like to devise a probability measure $p(y|x)$ quantifying whether x qualifies as anomalous $(y = 1)$. Within a semi-supervised setting, we assume that observable samples come from a distribution $\mathbb{P}_\mathcal{D}$ such that $x \sim \mathbb{P}_\mathcal{D}$ is associated with $y = 0$. Our approach relies on three components. First, we devise a probabilistic classifier $p_\theta(y|x)$, which models the outlierness degree and is parameterized by θ. This is the main expected outcome of our framework, which is modeled as a deep neural network classifier. However, our approach relies on learning $p_\theta(y|x)$ by only looking at samples from $\mathbb{P}_\mathcal{D}$.

The learning process is based on an outlier generator $g_\phi(\cdot)$, which starting from random noise z produces an outlier $\tilde{x}$ upon which to train the classifier. In principle, within a standard generative setting, it would suffice to jointly optimize ϕ and θ to maximize the likelihood

$$\mathbb{E}_{x\sim\mathbb{P}_\mathcal{D}}\left[\log p_\theta(0|x)\right] + \mathbb{E}_{\substack{z\sim\mathcal{N}(0,I)\\ \tilde{x}\sim g_\phi(z)}}\left[\log p_\theta(1|\tilde{x})\right],$$

stating that the distribution $\mathbb{P}_\phi$ (the distribution related to g_ϕ) differs substantially from the distribution $\mathbb{P}_\mathcal{D}$ of real data. This simple approach has the shortcoming that a simple solution would be a trivial generator g_ϕ producing extreme values without any informative value. By contrast, real life anomalies can represent borderline situations, such as anomalous combinations of eligible values within an observation. Essentially, a generator g_ϕ is informative when it is capable of generating realistic anomalies that force p_θ to detect relevant features from $\mathbb{P}_\mathcal{D}$ to be exploited for classification purposes.

Based on the above intuition, we would like to devise a generator that, starting from an $x \sim \mathbb{P}_\mathcal{D}$, generates a variant $\tilde{x}$ which resembles x although representing an outlier. This can be done by resorting to an encoder which can summarize the relevant features of x, to be exploited afterwards for reconstructing a suitable variant. We adopt a probabilistic encoder $q_\psi(z|x)$, which can be easily regularized to ensure stability to the learning process. The objective function to maximize can hence be rewritten as:

$$\begin{aligned}\mathcal{L}(\theta,\phi,\psi) = \mathbb{E}_{x\sim\mathbb{P}_\mathcal{D}}\Big[\log p_\theta(0|x)\Big] \\ + \mathbb{E}_{\substack{x\sim\mathbb{P}_\mathcal{D}\\ z\sim q_\psi(\cdot|x)\\ \tilde{x}\sim g_\phi(z)}}\Big[\log p_\theta(1|\tilde{x})\Big] \\ + \mathbb{E}_{\substack{x\sim\mathbb{P}_\mathcal{D}\\ z\sim q_\psi(\cdot|x)\\ \tilde{x}\sim g_\phi(z)}}\Big[\log p(x|\tilde{x})\Big] + \mathit{Reg}(q_\psi)\,.\end{aligned} \tag{1}$$

The third term in the equation models the fact that it is possible to easily reconstruct x from $\tilde{x}$: In practice, this means that both x and $\tilde{x}$ are equally probable from the latent z. A justification for the above loss can be seen in a variational setting. Consider an observation x, y and consider all possible variants $\tilde{x}$ of x for which the response is anomalous (i.e., $\tilde{y} = 1$), then:

$$\begin{aligned}\log p(x,y) &= \log\int p(x,y,\tilde{x},\tilde{y})\,\mathrm{d}\tilde{x} \\ &= \log\int p(x,y,\tilde{x},\tilde{y},z)\,\mathrm{d}\tilde{x}\,\mathrm{d}z\,.\end{aligned}$$

Consider now the decomposition $p(x,y,\tilde{x},\tilde{y},z) \approx p(y|x)\,p(\tilde{y}|\tilde{x})\,p(x|\tilde{x})\,p(\tilde{x}|z)\,p(z)$ and a proposal variational distribution $q(z|x)$. By way of the Jensen inequality, we have:

$$\begin{aligned}\log p(x,y) \geq &\int q(z|x)p(\tilde{x}|z)\log p(y|x)\,\mathrm{d}\tilde{x}\,\mathrm{d}z \\ &+ \int q(z|x)p(\tilde{x}|z)\log\{p(\tilde{y}|\tilde{x})p(x|z)\}\,\mathrm{d}\tilde{x}\,\mathrm{d}z \\ &- \int q(z|x)\log\frac{q(z|x)}{p(z)}\,\mathrm{d}z \\ = &\log p(y|x) + \mathbb{E}_{\substack{z\sim q(\cdot|x)\\ \tilde{x}\sim p(\cdot|z)}}\Big[\log p(\tilde{y}|\tilde{x})\Big] \\ &+ \mathbb{E}_{\substack{z\sim q(\cdot|x)\\ \tilde{x}\sim p(\cdot|z)}}\Big[\log p(x|\tilde{x})\Big] - \mathbb{KL}\left[q(z|x)\|p(z)\right]\,.\end{aligned}$$

We finally obtain eq. 1 by averaging over all possible $x \sim \mathbb{P}_\mathcal{D}$. Figure 1a summarizes the components of the model and the main flow in the learning process: Given an observation $x \sim \mathbb{P}_\mathcal{D}$, we encode it in a latent code z upon which a reconstruction $\tilde{x}$ can be obtained that the classifier should in principle classify as negative, while still resembling the original x as much as possible.

A. Adversarial Learning

A problem with eq. 1 is the presence of two apparently contrasting objectives within the same function. By simultaneously optimizing θ, ϕ and ψ, the learning process has to face the problem of generating an observation $\tilde{x}$ which should be classified as positive, while at the same time penalizing any departure from the original observation x. Again, this can result in an unstable learning process, which can be solved by alternating optimization with competing objectives: The generator and encoder aiming at obtaining the best possible reconstruction, and by contrast the classifier aiming at spotting all possible differences. In practice, the learning process can be restructured into an adversarial game with associated discriminator loss

$$\begin{aligned}\mathcal{L}_D(\theta|\phi,\psi) = \mathbb{E}_{x\sim\mathbb{P}_\mathcal{D}}\Big[\log p_\theta(0|x)\Big] \\ + \mathbb{E}_{\substack{x\sim\mathbb{P}_\mathcal{D}\\ z\sim q_\psi(\cdot|x)\\ \tilde{x}\sim g_\phi(z)}}\Big[\log p_\theta(1|\tilde{x})\Big]\end{aligned} \tag{2}$$

and generator loss

$$\begin{aligned}\mathcal{L}_G(\phi,\psi|\theta) = \mathbb{E}_{\substack{x\sim\mathbb{P}_\mathcal{D}\\ z\sim q_\psi(\cdot|x)\\ \tilde{x}\sim g_\phi(z)}}\Big[\log p_\theta(0|\tilde{x})\Big] \\ + \mathbb{E}_{\substack{x\sim\mathbb{P}_\mathcal{D}\\ z\sim q_\psi(\cdot|x)\\ \tilde{x}\sim g_\phi(z)}}\Big[\log p(x|\tilde{x})\Big] \\ - \mathbb{KL}\left[q_\psi(z|x)\|p(z)\right]\,.\end{aligned} \tag{3}$$

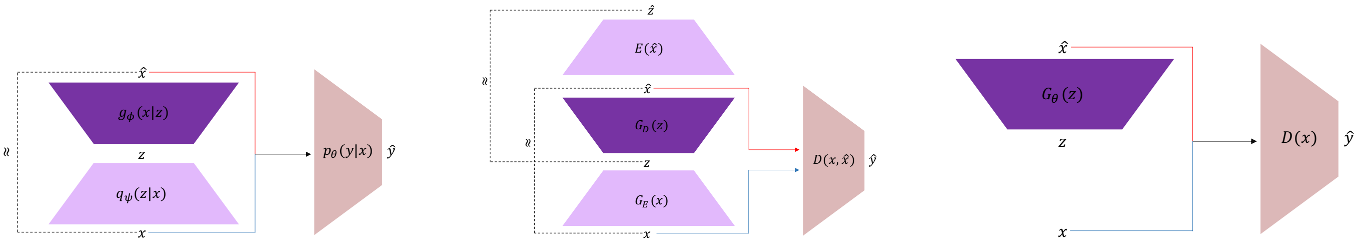
Figure 1 shows the differences between the proposed ARN and two similar approaches from the literature. FenceGAN ([14], fig. 1c) uses the typical GAN architecture with modified loss functions to generate samples that lie at the boundary of the real data. The discriminator score is directly used as an anomaly score. Like FenceGAN, ARN builds realistic outliers by training the discriminator to recognize even minimal discrepancies from normality. However, within ARN the generation phase relies on a faithful reconstruction, thus correlating outliers to negligible noise relative to the true distribution. In other words, the contrasting objective that $\tilde{x}$ must be scored as an outlier, despite the fact that it resembles the true x as faithfully as possible, forces the discriminator to capture the true essence of the original data manifold.

GANomaly ([5], fig. 1b) is a GAN-based model that also exploits faithful reconstructions, but the anomaly score relies on the difference in the reconstruction. The last encoder is used to map the reconstruction of the decoder in the latent space in order to have the best input representation. The anomaly score is defined as the difference between $G_E(x)$ and $E(G_D(G_E(x)))$. While this is not an issue on outliers coming from substantially different distributions, minimal differences would map in the same latent space and hence they would be difficult to identify.

Another substantial difference between ARN and the other adversarial approaches lies in the fact that, for each $x \sim \mathbb{P}_\mathcal{D}$, the generator produces an outlying variant $\tilde{x} \sim g_\phi(z)$, given $z \sim q_\psi(z|x)$. Since $q_\psi(z|x)$ approximates the true posterior $p(z|x)$, the entire manifold is efficiently explored, thus avoiding the mode collapse that typically affects adversarial approaches.

B. Unbalanced Learning

The above framework can be easily adapted to cope with partial supervision provided by a limited number of samples from the minority (outlier) class. When y is known for both normal and anomalous samples (and consequently $\mathbb{P}_\mathcal{D}$ represents the entire domain $\mathcal{D}$, including x related to $y = 1$), ARN can be trained with the objective



(a) ARN (b) GANomaly (c) FenceGAN

Fig. 1: ARN compared to other generative adversarial approaches.

$$
\begin{aligned}
\max_{\theta} \min_{\phi,\psi} \quad & \mathbb{E}_{x,y \sim \mathbb{P}_\mathcal{D}} \Big[\log p_\theta(y|x) \Big] \\
& + \mathbb{E}_{\substack{x,y \sim \mathbb{P}_\mathcal{D} \\ z \sim q_\psi(\cdot|x) \\ \tilde{x} \sim g_\phi(z)}} \Big[\log(1 - p_\theta(y|\tilde{x})) \Big] \\
& + \mathbb{E}_{\substack{x \sim \mathbb{P}_\mathcal{D} \\ z \sim q_\psi(\cdot|x) \\ \tilde{x} \sim g_\phi(z)}} \Big[\log p(x|\tilde{x}) \Big] \\
& + \mathbb{KL} \left[q_\psi(z|x) \| p(z) \right] ,
\end{aligned}
\tag{4}
$$

stating that the generator should be aimed at opposing the true class, while at the same time maintaining the best possible reconstruction (that is to say, by introducing the minimal changes that cause a reversal in the classification). In practice, the adversarial game introduces a synthetic resampling mechanism that allows to build a robust classification.

IV. Experimental Assessment

We conduct an extensive empirical evaluation of the proposed model on real-world datasets. Our goal is to answer to the following research questions:

- **RQ1**. Does the outlier generator produce realistic outliers? In other words, does the output of the data generator produces data with meaningful, realistic and non-trivial anomalies? How does it affect the predictive power? (Section IV-A)
- **RQ2**. In real-world scenarios, can the classifier component be used to predict unobserved anomalies? How does its predictive power compare to other state-of-the-art approaches? (Section IV-B)
- **RQ3**. How is the accuracy affected by contamination in the learning process? In what degree a limited amount of supervision helps the learning process? (Section IV-C)
- **RQ4**. Which components of the model contribute to the overall quality? How do the architectural choices affect the accuracy of the resulting predictions? (Section IV-D)
- **RQ5**. How efficient is the learning process of ARN compared to the other state of the art approaches? (Section IV-E)

We implemented ARN using the PyTorch framework [37]. In order to foster reproducibility, we publicly release all the data and code necessary to replicate our experiments[1]. Among the specific implementation details, it is worth noticing that, during model learning, the sampling $\tilde{x} \sim g_\theta(z)$ has to be properly arranged in a way that avoids breaking the back-propagation of the gradient. Numerical attributes are modeled by Gaussian distributions, unless otherwise specified. The sampling can hence follow the usual reparametrization trick. For binary/discrete attributes we explore two choices. The first one is to model such attributes as numeric and resort to the standard sampling for numerical attributes. Alternatively, we can adapt the framework described in [38], by exploiting the Gumbel Distribution and devising a Straight-Through (ST) Gumbel Estimator with the further trick of annihilating the temperature during the training process. We call the two alternative instantiations respectively ARN^N and ARN^G. In both cases, the component $p(x|\tilde{x})$ within the loss is modeled as a Gaussian reconstruction probability, i.e. $\log p(x|\tilde{x}) \approx -\gamma\|x - \tilde{x}\|^2$, with γ representing a weighting constant.

A. Outlier Generation

In a first set of experiments, we answer to **RQ1** and specifically we evaluate the capability of the generator to produce data with meaningful, realistic and non-trivial anomalies. The evaluation is performed on MNIST[2], a dataset of handwritten digits. Each instance consists of a 28 × 28 gray-scale image representing a digit in the interval $\{0, \ldots, 9\}$. The objective of the analysis is twofold: On one side, we would like to get a visual perception of the changes that the generator produces on the input data; On the other side, we want to show how these changes affect the resulting prediction of the discriminator. To do so, we binarize the input data and then train ARN on the whole set of images. The graphs in fig 2 describe a stable learning process, with both $\mathcal{L}_D$ and $\mathcal{L}_G$ achieving convergence. Concerning $\mathcal{L}_G$ the term representing the loss in reconstruction consistently converges, while the loss on the adversarial component increases: A clear sign that, despite the efforts by the generator, the classifier progressively refines its capability of discriminating between normal and generated samples.

The results of the generation are illustrated in fig. 3. The first row represents the original (greyscale) image, from which

[1] https://github.com/arnwg/arn
[2] http://yann.lecun.com/exdb/mnist

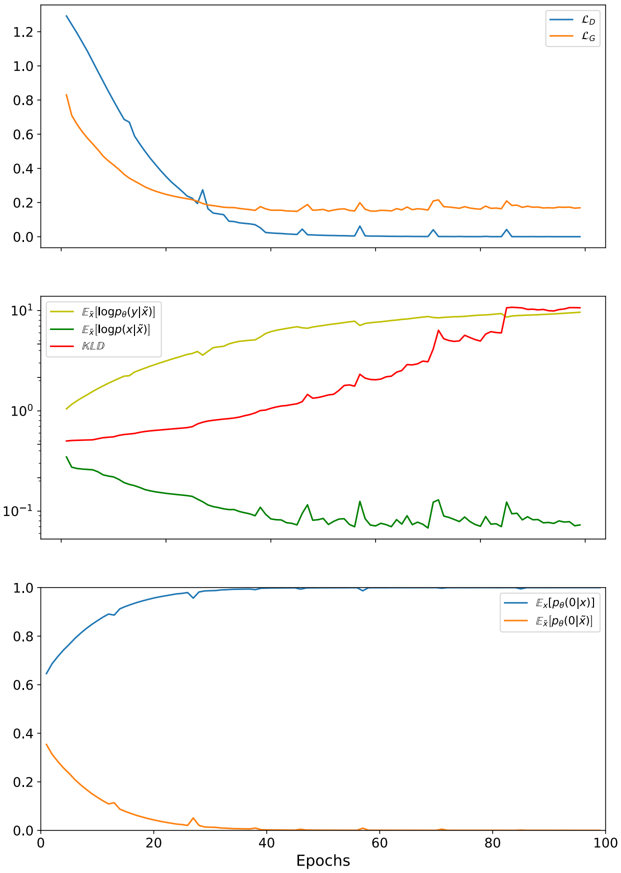


Fig. 2: Stability of the learning process.

the binary representation (on the third row) is sampled. The images in the second row represent, for each image, the parameters of the reconstructed Gumbel distribution, from which $\tilde{x}$ is sampled (fourth row). We can notice that, despite the strong similarity between the first two rows (and the last two as well), the last row exhibits some artifacts. Such artifacts severely affect the response of the classifier, which recognizes x as normal and $\tilde{x}$ as abnormal. In practice, the small variations do not affect the semantic of the image (a human eye can easily still recognize the number represented in $\tilde{x}$) but the few artifacts (either missing or redundant white pixels) are recognized by the classifier as anomalous.

In order to quantify the quality of the reconstruction, we compute a variant of the Fréchet Inception Distance (FID) [39]. In practice, we consider a traditional Variational Autoencoder trained on the original data. This autoencoder is then exploited to produce a latent representation of both the original images and the generated variants produced by ARN. The FID is then computed on these latent representations. Figure 4 shows how FID progressively decreases during the training process.

Figure 5 illustrates how both generated and real data are mapped in the latent space on an example training process, where the autoencoder is learned with a latent size $K = 2$. The shaded dots represent the manifold of the original (normal data), with each color representing a different digit. We also plot some sample images and their corresponding variant produced through the generator. Within the graph, original samples are represented with fully opaque circle markers, and the corresponding variants (exhibiting the same color) with the '+' marker. We can see that the variants lie in a neighborhood of the original images: Still, they diverge from them and sometimes they even cross the boundaries of the corresponding regions within the manifold.

B. *Outlier Detection*

In this section we answer to **RQ2** and specifically we study the predictive accuracy of ARN in comparison to other approaches in the literature.

1) Datasets: The experiments focus on six different standard benchmark datasets, described below.

- **KDDCUP99**[3] consists in a collection of network activity data, with each instance describing statistics relative to a connection (a sequence of TCP packets exchanged between two peers). Each connection is labeled as either normal or attack. There are 22 different types of attacks (with frequency ranging from popular to extremely rare), grouped in four macro-categories. We consider the reduced version of the dataset that contains 10% of the instances. In the experiments, we consider three variants. The first one (KDDCUP99) representing all possible attacks. This version is extremely unbalanced towards the anomalous class. KDDCUP99$_{Rev}$, a subset of KDDCUP99 where the majority classes (smurf and neptune) are removed. The rationale for the latter is that, without these attacks, the dataset exhibits a more realistic unbalance ($\sim 98\%$) towards normal connections. Finally, KDDCUP99$_{Inv}$ is a version where normal and anomaly classes are reverted. This interpretation of the dataset is adopted in several baselines and hence it is worth being considered in the comparisons.
- **NSL-KDD**[4] is a refined version of KDDCUP99. It is introduced in [40] to solve some of the inherent problems of KDDCUP99 dataset. In particular, it does not include redundant examples (which could bias both the learning process and the evaluation) and exhibits a balance between the classes.
- **DoH**[5] is a dataset describing packet flows representing benign and malicious DoH (DNS over HTTPS) traffic along with non-DoH traffic. The dataset is characterized by 28 features describing flow properties, such as number of bytes sent/received, stats on packet length and packet time, etc. We only focus on DoH traffic and use the "benign" and "malicious" classes. Since DoH is unbalanced towards the "malicious" class label, we also consider its reverted variant that we call DoH$_{Inv}$ in the following.

[3] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[4] https://www.unb.ca/cic/datasets/nsl.html
[5] https://www.unb.ca/cic/datasets/dohbrw-2020.html

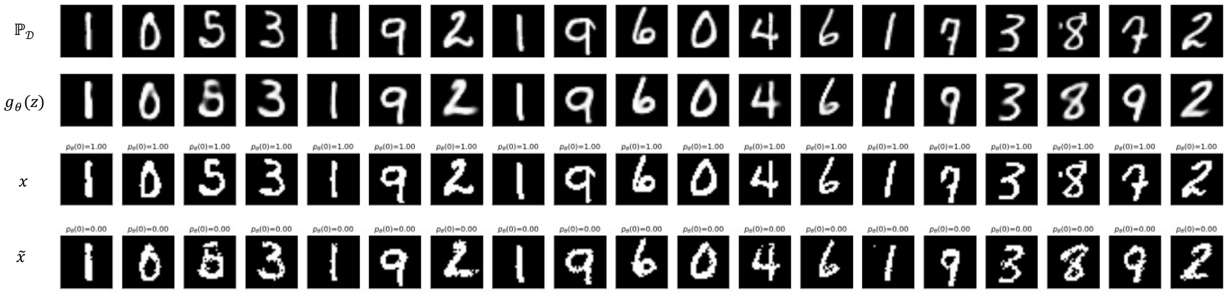


Fig. 3: Anecdotal evidence of Outlier generation on MNIST.

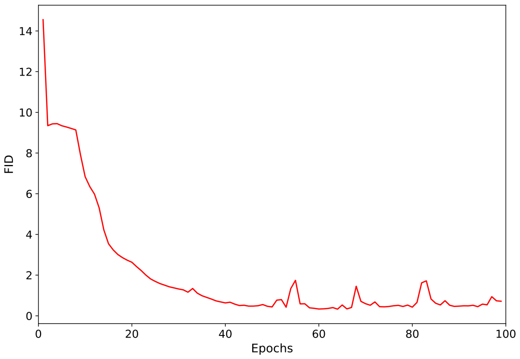


Fig. 4: FID between x and $\tilde{x}$ along the training process.

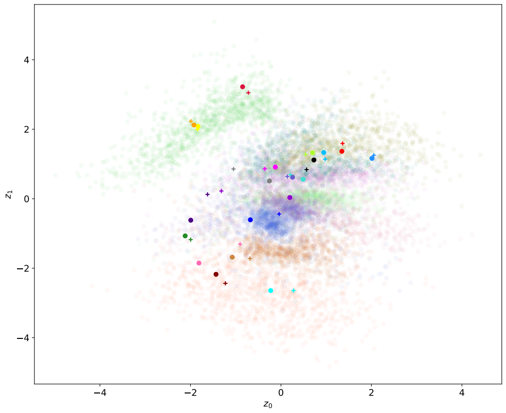


Fig. 5: Representation of real and generated images on a two-dimensional latent space.

- **CoverType**[6] is a multiclass classification dataset that is used in predicting forest cover type from cartographic variables only. Instances in this dataset represent areas within the Roosevelt National Forest of northern Colorado. Each area is associated with a type, relative to the underlying ecological processes. There are 7 classes. In our study, instances from classes 1,2 and 3 are considered as normal points and instances from the remaining classes, representing less typical types, are considered anomalies.
- **CreditCard**[7] [41] is a dataset representing online transactions occurred in a time span of two day, labeled either as legit or fraud. The dataset only contains numerical attributes and is highly unbalanced, since the transactions labeled as fraud represent only 0.17% of all transactions.
- **Bank**[8] is a dataset relative to direct marketing campaigns of a banking institution [42]. Each instance describes a prospective customer with a label describing whether the customer buys the product or not. Again, the dataset is unbalanced with the number of “yes” being a minority. In addition, all attributes are categorical.

For each of the above datasets, we identify a normal and an anomalous class. The objective is to train ARN on the normal samples in order to obtain a classifier that is able to correctly discriminate between normal and anomalous. Table I summarizes the description of each dataset. For each dataset, data preprocessing includes one-hot encoding for categorical attributes and min-max scaling on all numerical attributes.

2) Evaluation Protocol: In our experiments, from each dataset we sample train D_{tr}, test D_{te} and validation D_v subsets. Training data only contains normal samples, whereas validation and test contain both normal and anomalous samples. The sampling process is structured to guarantee that *(i)* the normal/anomalous proportions are kept consistent in the test and validation sets, and *(i)* all samples in the minority class are exploited. In particular, the normal samples maintain the $80/15/5$ proportions on train/test/validation, and the anomalous samples $75/25$ respectively, but the ratio N/A (where N and A represent the total number of normal and anomalous samples, respectively) is maintained in both test and validation, when possible.

To evaluate the performance of our model and to compare with the baseline methods we compute the Area Under the Curve (AUC) [43] and the Area under the Precision-Recall Curve (AUPRC) [44]. All the experiments are performed on

[6] https://archive.ics.uci.edu/ml/datasets/covertype

[7] http://www.ulb.ac.be/di/map/adalpozz/data/creditcard.Rdata

[8] https://github.com/GuansongPang/anomaly-detection-datasets

Dataset	Normal		Anomalous		Features				
	Type	Size	Type	Size	Categorical	Numeric	D_{tr}	$D_{te}(N/A)$	$D_v(N/A)$
KDDCUP99	normal	97,278	DoS, probe, R2L, U2R	396,743			77,822	14,592/29,756	4,864/9,919
KDDCUP99$_{Rev}$	normal	97,278	DoS$\setminus\{$smurf, neptune$\}$, probe, R2L, U2R	8,752	7	34	81,323	11,966/3,282	3,989/1,094
KDDCUP99$_{Inv}$	DoS, probe, R2L, U2R	396,743	normal	97,278			274,006	91,197/36,479	31,540/12,161
NSL-KDD	normal	77,054	DoS, probe, R2L, U2R	71,463			68,501	6.396/6,396	2,157/2,157
DoH	benign	19,807	malicious	249,836	2	28	15,846	2,971/18,738	990/6,246
DoH$_{Inv}$	malicious	249,836	benign	19,807			184,444	46,794/7,427	15,598/2,475
CoverType	1,2,3	530,895	4,5,6,7	50,117	44	10	444,764	64,599/18,794	21,532/6,265
CreditCard	normal	284,315	fraud	492	-	30	227,846	42,352/369	14,117/123
Bank	no	36,548	yes	4,640	10	-	26,383	7,614/1,740	2,551/580

TABLE I: Dataset Description

20 runs and the average values are reported, with statistical significance computed at 95% confidence. To guarantee sample variability in each run, the above described sampling process only considers half of the abnormal examples.

3) Baselines: We choose the state-of-the-art anomaly detection methods discussed in fig. 1: **FenceGAN** [14], which uses the typical GAN architecture with modified loss functions to generate samples that lie at the boundary of the real data manifold; and **GANomaly** [5], a GAN-based model structured into an encoder-decoder-encoder network. In addition, we compare with **OC-SVM** [2], for its flexibility and capability to identify a wide range of nonlinear boundaries separating classes of data in both a supervised and unsupervised way.

We also introduce a simple baseline that exploits an Autoencoder trained on normal data to achieve low MSE reconstruction error. The error can be used to identify whether an example is abnormal or not, as discussed in section II. Although autoencoding-based anomaly detection is not suitable for generating outliers, it is worth investigating how ARN and other baselines compare to this simple approach in detection ability.

4) Results: Table II reports the results of the evaluation. We include both the ARN^G and ARN^N variants, which consistently exhibits suitable values of AUC and AUPRC on all datasets. On datasets where the anomalous class balances with the normal class, the results are comparable with those of the competing methods. However, with class imbalance, ARN tends to have a better response in terms of precision, and in general there is a consistent gain in performance. Figure 6 shows how the generation process provides realistic samples with minimal but substantial deviations from their original that characterize the generated samples as anomalous. In fact, generated samples tend to share similarities with anomalous samples, in terms of discrepancy from the normal samples.

The modeling choices regarding the categorical attributes do not seem to indicate a clear winning strategy. By looking at the results for CoverType and Bank, we see that the responses, albeit comparable, are inverted with a predominance of the Gumbel softmax in CoverType and of the continuous relaxation for Bank.

C. Robustness

Besides the ideal situation where all training examples are actually normal, we consider other two specific experiments.

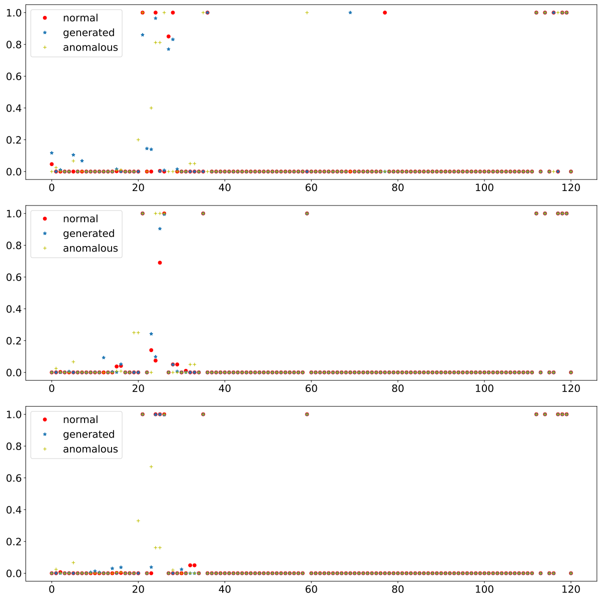


Fig. 6: Real and generated anomalies in KDDCUP99$_{Rev}$. The x axis enumerates the data attributes, the y axis normalizes the attribute values in the $[0, 1]$ interval. Each sample is represented as a set of dots according to its value for each attribute. Red dots are attribute values for real normal samples; blue stars are related to generated anomalies; finally, yellow crosses represent real anomalies.

- *Contamination*, in which we contaminate the training samples by adding a percentage of anomalous data that are still considered normal. The idea is that, in a truly unsupervised situation, little is known about the true distribution of the data. A model is deemed robust when the accuracy is tolerant to a moderate amount of abnormality.
- *Weak supervision*, where a limited amount of examples are known to belong to the minority class. Since in principle our approach is also capable of exploiting such supervision (as explained in section III-B), it is important to see whether the generation process is aided by a limited amount of tuples actually labeled as anomalous.

In order to measure robustness, we contaminate the training data with a percentage of anomalous tuples expressed as $p = A/N$, where N and A are the amounts of normal and anomalous tuples in the training set, respectively.

Dataset	ARN^G		ARN^N		FenceGAN		GANomaly		OC-SVM		Baseline	
	AUC	AUPRC	AUC	AUPRC	AUC	AUPRC	AUC	AUPRC	AUC	AUPRC	AUC	AUPRC
KDDCUP99	.98 ± .01	.99 ± .01	.99 ± .01	.99 ± .01	.99 ± .00	.99 ± .00	1.00 ± .00	1.00 ± .00	.96 ± .00	.97 ± .00	**1.00 ± .00**	**1.00 ± .00**
KDDCUP99$_{Rev}$	.97 ± .01	**.95 ± .02**	**.99 ± .00**	.94 ± .02	.84 ± .01	.77 ± .01	.92 ± .01	.86 ± .01	.81 ± .00	.71 ± .00	.91 ± .01	.87 ± .01
KDDCUP99$_{Inv}$	.98 ± .02	.98 ± .01	.93 ± .05	.95 ± .04	.92 ± .03	.72 ± .08	.91 ± .04	.90 ± .03	.95 ± .00	.82 ± .00	1.00 ± .00	**1.00 ± .00**
NSL-KDD	.98 ± .01	**.98 ± .01**	.98 ± .00	.97 ± .01	.96 ± .00	.97 ± .00	.97 ± .01	.97 ± .01	.96 ± .00	.97 ± .00	**.99 ± .00**	.98 ± .00
DoH	.99 ±.01	1.00 ± .00	**1.00 ± .00**	**1.00 ± .00**	.88 ± .02	.97 ± .00	.99 ± .00	1.00 ± .00	.88 ± .00	.97 ± .00	.96 ± .00	.99 ± .00
DoH$_{Inv}$	.98 ± .01	.97 ± .01	.99 ± .00	.96 ± .01	.89 ± .02	.44 ± .05	**1.00 ± .00**	.98 ± .01	.90 ± .00	.49 ± .01	.99 ± .00	.91 ± .04
CoverType	**.96 ± .01**	**.97 ± .01**	.93 ± .03	.94 ± .02	.70 ± .03	.41 ± .02	.56 ± .05	.30 ± .04	.73 ± .02	.43 ± .02	.53 ± .02	.28 ± .02
CreditCard	-	-	**.99 ± .01**	.59 ± .06	.90 ± .01	.51 ± .03	.84 ± .02	.36 ± .05	.92 ± .01	.57 ± .01	.99 ± .00	**.76 ± .01**
Bank	.70 ± .05	.63 ± .07	**.76 ± .04**	**.69 ± .07**	.56 ± .01	.23 ± .01	.53 ± .02	.22 ± .02	.60 ± .00	.28 ± .00	.65 ± .00	.32 ± .01

TABLE II: Comparative analysis. Bold (resp. gray) values represent models with statistically higher (resp. lower) scores.

Table III reports the results. We split the analysis in two tables: The first one focuses on imbalanced datasets and compares the response to contamination with GANomaly, for the values $p = 1\%$ and $p = 5\%$. The second table reports the results concerning ARN^G for balanced datasets, with a higher contamination ranging from 0% to 50% . We can see that as we increase the contamination the accuracy tends to decrease steadily. This decrease is lower than GANomaly which by contrast seems to be more heavily affected by contamination.

For the weak supervision, we compare ARN with **WALDO** [6], an approach that combines Wasserstein autoencoders to detect and generates both inliers and outliers. This experiment is only illustrated on KDDCUP99$_{Rev}$ and Bank on table IV. We can see a substantial improvement on the baseline performance, even with a minimal amount of supervision. The advantage over WALDO is substantial, which by contrast does not seem able to efficiently exploit the small portions of anomalous data to improve the detection accuracy.

D. Ablation Study

In a final set of experiments, we study the contribution of each component of the model to the accuracy. We already discussed the effects of modeling discrete attributes with either a Gumbel distribution or a continuous relaxation. We next evaluate two more aspects: The importance of regularization and of the adversarial learning process over a standard ELBO optimization. Within table V, ARN$^{X-\mathbb{KLD}}$ represents model ARN^X trained without regularization, and ARN^{GE} represents ARN^G trained by optimizing the ELBO in eq. 1. The regularization seems to play a prominent role in guaranteeing robust reconstructions. Also, the adversarial learning provides a substantial advantage in the learning process, as we can see by comparing ARN^G and ARN^{GE}.

E. Architecture and learning

For the last research question concerning the efficiency of ARN, we discuss here the architectural details and their effects on the computational efficiency of the learning process. First, ARN is learned via an alternate maximization algorithm. In order to stabilize the learning process, we progressively inject noise on the discriminator labels as suggested in [45]. As a result, the learning phase is generally smooth (as also illustrated in fig. 2). All the models were trained using the Adam optimization. We also adopt different learning rates for the discriminator and the generator.

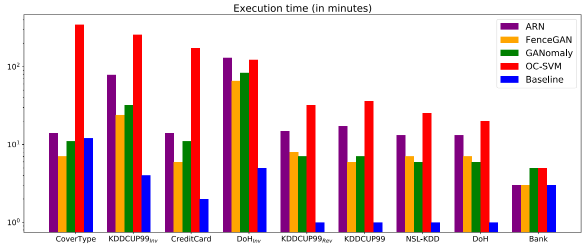


Fig. 7: Running times in minutes (log scale).

Throughout the experiments described in sections IV-B through IV-D, we devised simple architectures for both the generator and the discriminator, based on linear layers equipped with batch normalization. The architecture in IV-A also exploits convolutional layers in order to learn better representations for the MNIST images. All experiments were performed on an NVIDIA DGX equipped with 4 V100 GPU. Figure 7 summarizes the running times for each model on the datasets described in IV-B. Here, we can notice the ARN is the second last performer in almost all the datasets. However, the order of magnitude is comparable for all models except for OC-SVM that clearly requires much longer execution times w.r.t. all the other evaluated algorithms.

V. Conclusion

The Adversarial Reconstruction Network (ARN) is a twofold neural architecture aimed at generating and identifying anomalies in data sets. The learning scheme consists in an adversarial game between a generator and a discriminator, respectively designed as a variational encoder-decoder structure and a supervised network. Its peculiarity lies in its data generation approach: Anomalies are generated by observing non-anomalous samples based on two guiding principles. First, generated data are reconstructed from original input samples to lie in the same boundaries of the data manifold. Second, the reconstruction is guided to highlight elements that characterize the generated sample as anomalous. We showed that such an approach is effective in detecting outliers, robust to noise in the training data and easily adaptable to weak supervision. Furthermore, it is capable of generating realistic outliers.

The proposed approach requires prior knowledge of at least the samples labeled as normal. However, in principle the nature of the adversarial approach to reconstruction does not necessarily require knowledge concerning the prior class

	KDDCUP99$_{Rev}$		**KDDCUP99**$_{Inv}$		**CoverType**	
	ARN^N	**GANomaly**	**ARN**^{G}	**GANomaly**	**ARN**^{G}	**GANomaly**
No contamination	.99 ± .00	.92 ± .01	.98 ± .02	.91 ± .04	.96 ± .01	.56 ± .05
$p = 1\%$	.98 ± .00	.88 ± .02	.98 ± .02	.88 ± .07	.70 ± .06	.54 ± .11
$p = 5\%$	.93 ± .05	.81 ± .01	.95 ± .03	.76 ± .20	.72 ± .10	.52 ± .10

	KDDCUP99	**NSL-KDD**	**DoH**
No contamination	.98 ± .01	.98 ± .01	.99 ± .01
$p = 1\%$	.96 ± .03	.98 ± .01	1.00 ± .00
$p = 5\%$	.94 ± .03	.95 ± .04	.99 ± .01
$p = 10\%$	.83 ± .09	.93 ± .03	.99 ± .00
$p = 25\%$	.74 ± .14	.91 ± .04	.98 ± .01
$p = 50\%$	.65 ± .12	.74 ± .07	.78 ± .06

TABLE III: Robustness to contamination.

Datasets	**Method**	0% **Anomalies**	1% **Anomalies**	3% **Anomalies**
KDDCUP99$_{Rev}$	**ARN**^{N}	.99 ± .00	.99 ± .00	.99 ± .00
	ARN^G	.97 ± .01	.99 ± .00	.99 ± .00
	WALDO	-	.80 ± .01	.80 ± .01
Bank	**ARN**^{N}	.76 ± .04	.79 ± .06	.89 ± .03
	ARN^G	.70 ± .05	.72 ± .04	.82 ± .05
	WALDO	-	.56 ± .01	.56 ± .01

TABLE IV: Weak supervision: Comparison with WALDO.

Dataset	**ARN**^{G}	**ARN**^{N}	**ARN**$^{G-\mathbb{KLD}}$	**ARN**$^{N-\mathbb{KLD}}$	**ARN**^{GE}
KDDCUP99	.98 ± .01	.99 ± .01	.93 ± .04	.99 ± .00	.74 ± 09
KDDCUP99$_{Rev}$	.97 ± .01	.99 ± .00	.96 ± .03	.97 ± .01	.83 ± .05
KDDCUP99$_{Inv}$	.98 ± .02	.93 ± .05	.97 ± .02	.93 ± .06	.88± .04
NSL-KDD	.98 ± .01	.98 ± .00	.98 ± .01	.95 ± .02	.74 ± .07
DoH	.99 ± .01	1.00 ± .00	.73 ± .01	.76 ± .02	.99 ± .01
DoH$_{Inv}$	.98 ± .01	.99 ± .00	.93 ± .05	.99 ± .00	.83 ± .04
CoverType	.96 ± .01	.93 ± .03	.72 ± .05	.93 ± .04	.77 ± .08
CreditCard	-	.99 ± .01	-	.86 ± .08	.75 ± .06

TABLE V: Ablation Study.

distribution. In fact, the latter could be directly inferred in the learning process (e.g. by exploiting bayesian inference), thus allowing a generalization of the proposed approach towards a fully unsupervised setting. We believe that this is an intriguing challenge and a direction worth further investigation that we plan to cope as future work.

References

[1] C. C. Aggarwal, *Outlier Analysis*. Springer, 2016.

[2] B. Schölkopf *et al.*, "Support vector method for novelty detection," in *NIPS*, 1999.

[3] Y. Xian, B. Schiele, and Z. Akata, "Zero-shot learning — the good, the bad and the ugly," in *CVPR*, 2017.

[4] F. D. Mattia *et al.*, "A survey on gans for anomaly detection," *CoRR*, 2019. arXiv: 1906.11632.

[5] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly: Semi-supervised anomaly detection via adversarial training," in *ACCV*, 2018.

[6] S. G. Rizzo *et al.*, "Probabilistic outlier detection and generation," *CoRR*, 2020. arXiv: 2012.12394.

[7] H. Zenati *et al.*, "Adversarially learned anomaly detection," in *ICDM*, 2018.

[8] N. Laptev, *Anogen: Deep anomaly generator*. [Online]. Available: https://tinyurl.com/fbanogen.

[9] G. Pang *et al.*, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, no. 1, pp. 1–38, 2021.

[10] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *ICLR*, 2014.

[11] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *NIPS*, 2014, pp. 2672–2680.

[12] L. Theis, A. van den Oord, and M. Bethge, "A note on the evaluation of generative models," in *ICLR*, 2016.

[13] Y. Hong *et al.*, "How generative adversarial networks and their variants work: An overview," *ACM Comput. Surv.*, vol. 52, no. 1, 2019.

[14] C. P. Ngo *et al.*, "Fence gan: Towards better anomaly detection," in *ICTAI*, 2019.

[15] A. Makhzani *et al.*, "Adversarial autoencoders," in *ICLR*, 2016.

[16] L. Ruff *et al.*, "A unifying review of deep and shallow anomaly detection," *Proceedings of the IEEE*, 2021.

[17] I. Tolstikhin *et al.*, "Wasserstein auto-encoders," in *ICLR*, 2019.

[18] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *ICDM*, 2008.

[19] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *SIGMOID*, 2000.

[20] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," *CoRR*, 2020. arXiv: 2003.05991.

[21] Z. Chen *et al.*, "Autoencoder-based network anomaly detection," in *WTS*, 2018.

[22] J. An and S. Cho, *Variational autoencoder based anomaly detection using reconstruction probability*, 2015. [Online]. Available: http://dm.snu.ac.kr/static/docs/TR/SNUDM-TR-2015-03.pdf.

[23] A. L. Alfeo *et al.*, "Using an autoencoder in the design of an anomaly detector for smart manufacturing," *Pattern Recognition Letters*, vol. 136, 2020.

[24] S. Hawkins *et al.*, "Outlier detection using replicator neural networks," in *DaWaK*, 2002.

[25] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *KDD*, 2017.

[26] K. Tian *et al.*, "Learning competitive and discriminative reconstructions for anomaly detection," in *AAAI*, vol. 33, 2019.

[27] T. Schlegl *et al.*, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *IPMI*, 2017.

[28] T. Salimans *et al.*, "Improved techniques for training gans," in *NIPS*, 2016.

[29] T. Schlegl *et al.*, "F-anogan: Fast unsupervised anomaly detection with generative adversarial networks," *Medical Image Analysis*, 2019.

[30] H. Zenati *et al.*, "Efficient gan-based anomaly detection," *CoRR*, 2019. arXiv: 1802.06222.

[31] Y. Kim and S. Choi, "Forward-backward generative adversarial networks for anomaly detection," in *ICML*, 2019.

[32] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," in *ICLR*, 2017.

[33] S. Akçay, A. Atapour-Abarghouei, and T. P. Breckon, "Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection," in *IJCNN*, 2019.

[34] H. S. Vu *et al.*, "Anomaly detection with adversarial dual autoencoders," *CoRR*, 2019. arXiv: 1902.06924.

[35] J. Chen *et al.*, "Outlier detection with autoencoder ensembles," in *SDM*, 2017.

[36] X. Han, X. Chen, and L.-P. Liu, "Gan ensemble for anomaly detection," in *AAAI*, 2020.

[37] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019.

[38] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *ICLR*, 2017.

[39] M. Heusel *et al.*, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *NIPS*, 2017.

[40] M. Tavallaee *et al.*, "A detailed analysis of the kdd cup 99 data set," in *CISDA*, 2009.

[41] A. D. Pozzolo *et al.*, "Calibrating probability with undersampling for unbalanced classification," in *SSCI*, 2015.

[42] S. Moro, P. Cortez, and P. Rita, "A data-driven approach to predict the success of bank telemarketing.," *Decis. Support Syst.*, vol. 62, 2014.

[43] F. Melo, "Area under the roc curve," in *Encyclopedia of Systems Biology*, W. Dubitzky *et al.*, Eds. 2013.

[44] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets," *PloS one*, vol. 10, no. 3, 2015.

[45] T. Salimans *et al.*, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems*, vol. 29, 2016.