Knowledge and Information Systems
https://doi.org/10.1007/s10115-024-02172-w

REGULAR PAPER

Robust anomaly detection via adversarial counterfactual generation

Angelica Liguori[1] · Ettore Ritacco[2] · Francesco Sergio Pisani[1] · Giuseppe Manco[1]

Received: 30 November 2022 / Revised: 8 May 2024 / Accepted: 25 June 2024

Abstract
The capability to devise robust outlier and anomaly detection tools is an important research topic in machine learning and data mining. Recent techniques have been focusing on reinforcing detection with sophisticated data generation tools that successfully refine the learning process by generating variants of the data that expand the recognition capabilities of the outlier detector. In this paper, we propose ARN, a semi-supervised anomaly detection and generation method based on adversarial counterfactual 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 counterfactual reconstruction helps to stabilize the learning process, which results in both realistic outlier generation and substantially extended detection capability. In fact, the counterfactual generation enables a smart exploration of the search space by successfully relating small changes in all the actual samples from the true distribution to high anomaly scores. 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.

Keywords Anomaly detection · Outlier detection · Anomaly generation · Outlier generation · Generative adversarial networks (GANs) · Variational autoencoders (VAEs)

✉ Angelica Liguori
angelica.liguori@icar.cnr.it

Ettore Ritacco
ettore.ritacco@uniud.it

Francesco Sergio Pisani
francescosergio.pisani@icar.cnr.it

Giuseppe Manco
giuseppe.manco@icar.cnr.it

[1] Institute for High Performance Computing and Networking, Italian National Research Council, Via P. Bucci, 87036 Rende, Cosenza, Italy

[2] Department of Mathematics, Computer Science and, Physics, University of Udine, Via Palladio, 33100 Udine, Italy

Published online: 17 July 2024

1 Introduction

Anomaly detection [1] 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 [2], healthcare [3], security [4] and finance [5]. 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 [6]. In general, approaches to outlier detection can be classified as supervised, semi-supervised, and unsupervised [7].

Supervised methods exploit the availability of a labeled data set containing observations already labeled as normal and abnormal 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 [8]. By contrast, unsupervised methods make no assumption on the class distribution and search for outliers in an unlabeled data set by assigning to each example a score that 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 [9]. 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 [10–14], due to their capabilities in modeling the hidden causal relationships that ultimately characterize data. The expressive representation schemes offered by deep networks [1], combined with sophisticated yet effective learning mechanisms based on stochastic backpropagation, approximate Bayesian inference [15] and adversarial learning [16], 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, over-generalization can occur [17], 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 reconstruct portions of the true distribution accurately. Still, they exhibit two potential short-

comings. On one side, mode collapse and dropping can prevent a faithful reconstruction [18], 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 [19], 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 to map each sample in a suitable latent feature representation space, from which an alternative counterfactual 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 [15, 20]. 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 2 discusses the recent contributions in the current literature and provides a systematic review of the approaches related to our task of interest. Section 3 discusses the mathematical details of our proposal. The effectiveness of the proposed model is illustrated in Sect. 4, and pointers to future developments are discussed in Sect. 5.

2 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 [7] 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 [15, 16, 21].

2.1 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 [8]), distance metrics (e.g., Isolation Forest [22]) or nearest

neighbor algorithms [23]. Recently, deep anomaly detection has emerged as a critical direction [1]. In particular, autoencoders [24] have been extensively used for unsupervised anomaly detection based on deep learning [25–30].

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." Consequently, 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 represent 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. [31] proposes 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 that uses 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. Unlabeled data are fed in both decoders and labeled as either outliers or inliers, based on the lowest reconstruction error. Also, [32] employs clustering-based approaches to improve the performance of autoencoders in detecting outliers without the need of a threshold.

Besides reconstruction error, an alternative emerging research direction is the adoption of generative adversarial networks (GANs) [10] to embed the detection process within a generative framework directly.

GANs [16] 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 the probability that the input sample came from the real data rather than p_{G}.

To the best of our knowledge, the first approach that combines generative adversarial networks and outlier detection is AnoGAN [33]. 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 [34] 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 [13, 35, 36] are proposed.

For example, ALAD [13] 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 [37], suffered by architectures based on BiGAN [38], like [36].

GANomaly [11] combines adversarial learning and latent representation through an encoding–decoding framework. The model is composed of a generator and a discrimina-

tor, 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. [39] proposes an extension of GANomaly particularly suited for input data represented by images. In such cases, in fact, the adoption of skip connections can substantially improve the reconstruction and consequently boost the detection abilities. In [40], a self-training approach is proposed exploiting a similar generator–discriminator structure. The generator tries to learn the normal data distribution, whereas the discriminator, together with a classifier, amplifies the reconstruction error of anomalous data for better identification of anomalies.

ADAE [41] 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 a better split between normal and anomalous scores, leading to superior performance. Also, [42] exploits two autoencoder subnetworks acting as generator and discriminator. In addition, the authors introduce the mutual adversarial training approach that allows the swap of the two subnetworks' roles in training, boosting the reconstruction capability. Moreover, an anomaly determination mechanism is proposed to circumvent the 'high anomaly' issue caused by the noisy records. Finally, a recent line of research is exploring the adoption of ensemble architectures based either on autoencoders [43, 44] or GANs [45]. 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.

2.2 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 outliers 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, [14] 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 are characterized by complex manifolds, over-generalization is likely to weaken the generation, as discussed in the introduction.

A (weakly) supervised framework based on the detection and generation of outliers is proposed in [12]. The framework, called WALDO (Wasserstein Autoencoder for Learning the Distribution of Outliers), combines the approach outlined in [31] (discussed above) with Wasserstein autoencoders [21].

The basic assumption is that data are generated from an unknown and unlabeled mixture of inlier and outlier distributions ($P_X^u = (1-v)P_X^{\mathrm{i}} + vP_X^{\mathrm{o}}$) and the goal is to learn generating distributions (inlier and outlier distributions) $P_{\mathrm{G}}^{\mathrm{i}}$ and P_G^{o} which minimize the Wasserstein Distances $W_p(P_X^{\mathrm{i}}, P_{\mathrm{G}}^{\mathrm{i}})$ and $W_p(P_X^{\mathrm{o}}, P_{\mathrm{G}}^{\mathrm{o}})$.

The approach requires that both examples from unlabeled and inlier distributions are provided in the training phase.

FenceGAN [19] 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.

3 Adversarial reconstruction networks

3.1 Preliminaries

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 $y = 0$ (i.e., x is a normal sample). 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

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

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. As an example in the cybersecurity domain [46, 47], consider the case of stealthy attacks. The latter in fact are designed to slightly alter legit metadata values within certain samples of legitimate traffic, in order to exfiltrate secret information. Consequently, while these values persist as legitimate within the communication, these subtle alterations allow to perform an attack. 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.

3.2 Methodology overview

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 that can summarize the relevant features of x, to be exploited afterward 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:

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

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. Therefore, the intuition behind the proposed approach is that although $\tilde{x}$ is as close as possible to x (i.e., $\tilde{x}$ is a variant of x, as deemed by the third term of the equation), the minimal changes produced by the generation cause a reversed decision by the classifier (expressed second term in the equation), whereas the original example x is still deemed as normal (i.e., associated with label $y = 0$ in the first term in the equation). According to these properties, we call $\tilde{x}$ a realistic anomalous counterpart of x. The fourth term refers to some regularization on the model parameters. A formal justification for the above loss can be seen in a variational setting [48]. 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:

$$\log p(x,y) = \log \int p(x,y,\tilde{x},\tilde{y}=1)\,\mathrm{d}\tilde{x} = \log \int p(x,y,\tilde{x},\tilde{y}=1,z)\,d\tilde{x}\mathrm{d}z\,. \tag{3}$$

Consider now the decomposition $p(x,y,\tilde{x},\tilde{y}=1,z) \approx p(y|x)\,p(\tilde{y}{=}1|\tilde{x})\,p(x|\tilde{x})\,p(\tilde{x}|z)\,p(z)$ and a proposal variational distribution $q(z|x)$. By way of the Jensen inequality [49], we have:

$$\begin{aligned}
\log p(x,y) &\geq \int q(z|x)p(\tilde{x}|z)\log p(y|x)\,\mathrm{d}\tilde{x}\,\mathrm{d}z \\
&\quad + \int q(z|x)p(\tilde{x}|z)\log\{p(\tilde{y}=1|\tilde{x})p(x|\tilde{x})\}\,\mathrm{d}\tilde{x}\,\mathrm{d}z \\
&\quad - \int q(z|x)\log\frac{q(z|x)}{p(z)}\,\mathrm{d}z \\
&= \log p(y|x) + \mathbb{E}_{\substack{z\sim q(\cdot|x)\\ \tilde{x}\sim p(\cdot|z)}}\Big[\log p(\tilde{y}=1|\tilde{x})\Big] \\
&\quad + \mathbb{E}_{\substack{z\sim q(\cdot|x)\\ \tilde{x}\sim p(\cdot|z)}}\Big[\log p(x|\tilde{x})\Big] - \mathbb{KL}\left[q(z|x)\|p(z)\right]\,.
\end{aligned} \tag{4}$$

We finally obtain Eq. 2 by averaging over all possible $x \sim \mathbb{P}_\mathcal{D}$. $\mathbb{KL}$ stands for Kullback–Leibler divergence [50]. 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.

3.3 Adversarial learning

A problem with Eq. 2 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

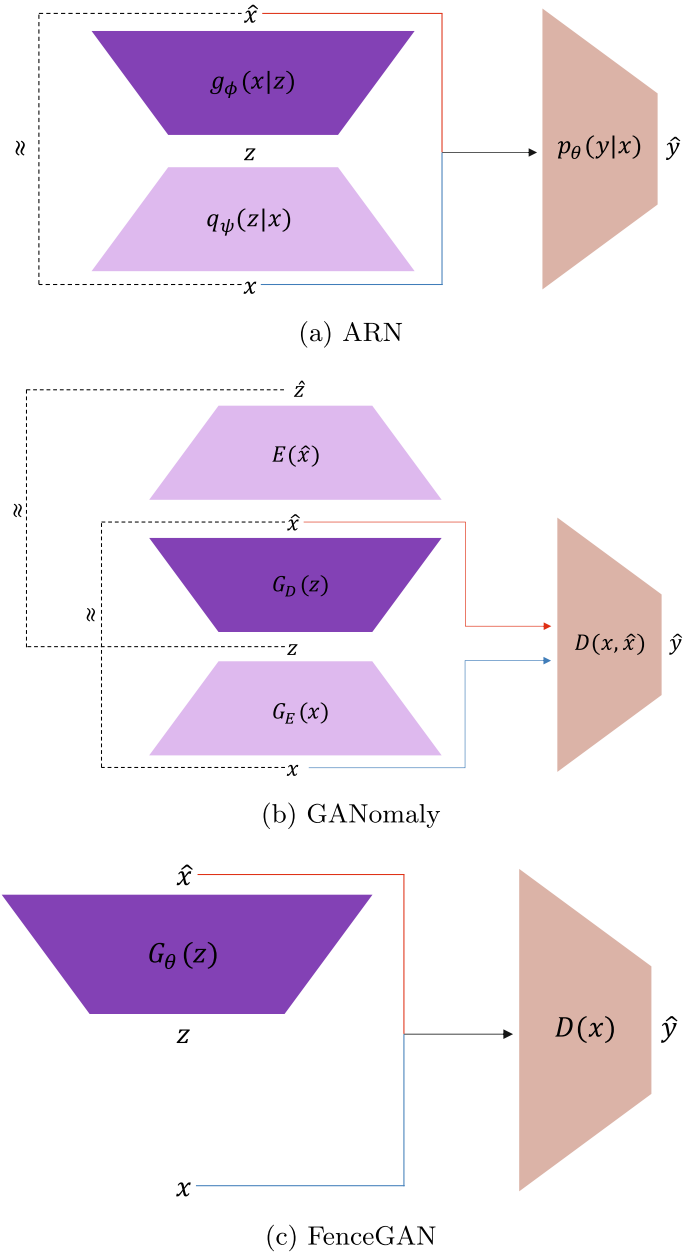

Fig. 1 ARN compared to other generative adversarial approaches

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

$$\mathcal{L}_D(\theta|\phi,\psi) = \mathbb{E}_{x\sim\mathbb{P}_\mathcal{D}}\Big[\log p_\theta(0|x)\Big] + \mathbb{E}_{\substack{x\sim\mathbb{P}_\mathcal{D}\\ z\sim q_\psi(\cdot|x)\\ \tilde{x}\sim g_\phi(z)}}\Big[\log p_\theta(1|\tilde{x})\Big] \tag{5}$$

and generator loss

$$\begin{aligned}\mathcal{L}_G(\phi,\psi|\theta) = \mathbb{E}_{\substack{x\sim\mathbb{P}_\mathcal{D}\\ z\sim q_\psi(\cdot|x)\\ \tilde{x}\sim g_\phi(z)}}\Big[\log p_\theta(0|\tilde{x})\Big] + \mathbb{E}_{\substack{x\sim\mathbb{P}_\mathcal{D}\\ z\sim q_\psi(\cdot|x)\\ \tilde{x}\sim g_\phi(z)}}\Big[\log p(x|\tilde{x})\Big] \\ - \mathbb{KL}\left[q_\psi(z|x)\|\,p(z)\right].\end{aligned} \tag{6}$$

By using Eqs. 5 and 6, we generate for each normal sample x, a counterpart $\tilde{x}$, i.e., a sample that resembles x (see the first two terms in Eq. 6), and at the same time, $\tilde{x}$ is classified as anomalous (second term in Eq. 5), while the original sample is recognized as normal (first term in Eq. 5). Thus, the entire process allows the generation of synthetic but realistic anomalies that enable the training of the outlier detector (the discriminator). Indeed, the aim is to improve the abilities of the discriminator, which can devise a boundary between normality and abnormality, only exploiting the normal samples. In this way, the detector can identify existing anomalies without needing explicit information about them.

Figure 1 shows the differences between the proposed ARN and two similar approaches from the literature. FenceGAN ([19], 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 ([11], 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.

3.4 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

$$\begin{aligned}\max_{\theta}\min_{\phi,\psi} \quad & \mathbb{E}_{x,y\sim\mathbb{P}_{\mathcal{D}}}\Big[\log p_\theta(y|x)\Big] + \mathbb{E}_{\substack{x,y\sim\mathbb{P}_{\mathcal{D}}\\ z\sim q_\psi(\cdot|x)\\ \tilde{x}\sim g_\phi(z)}}\Big[\log(1-p_\theta(y|\tilde{x}))\Big] \\ & + \mathbb{E}_{\substack{x\sim\mathbb{P}_{\mathcal{D}}\\ z\sim q_\psi(\cdot|x)\\ \tilde{x}\sim g_\phi(z)}}\Big[\log p(x|\tilde{x})\Big] + \mathbb{KL}\big[q_\psi(z|x)\|p(z)\big]\,,\end{aligned} \tag{7}$$

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.

4 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? (Sect. 4.1)
- *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? (Sect. 4.2)
- *RQ3*. How is the accuracy affected by contamination in the learning process? In what degree a limited amount of supervision helps the learning process? (Sect. 4.3)
- *RQ4*. Which components of the model contribute to the overall quality? How do the architectural choices affect the accuracy of the resulting predictions? (Sect. 4.4)
- *RQ5*. How efficient is the learning process of ARN compared to the other state-of-the-art approaches? (Sect. 4.5)

We implemented ARN using the PyTorch framework [51]. All the technical details about ARN, its variants and competitors, are described in the Supplementary material. 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 backpropagation 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 [52], 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.

[1] https://github.com/arnwg/arn.

4.1 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. For this, we use both real and synthetic data.

4.1.1 Experiments on real data

A first evaluation is performed on MNIST,[2] a dataset of handwritten digits. Each instance consists of a 28×28 grayscale 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 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 allow for minimal modifications that lead ARN to classify the instance as anomalous. Indeed, the generated point $\tilde{x}$ represents a sample that is similar to the normal one x, while lying outside the original (normal) data distribution. Consequently, the classifier 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) [53]. 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. Let f and $\tilde{f}$ be the latent representation of the original images and the generated variants, respectively, the Fréchet inception distance is defined as follows:

$$\text{FID} = \|\mu_f - \mu_{\tilde{f}}\|^2 - Tr(\Sigma_f + \Sigma_{\tilde{f}} - 2\Sigma_f \Sigma_{\tilde{f}}) \tag{8}$$

where $\mu_f, \mu_{\tilde{f}}$, Σ_f and $\Sigma_{\tilde{f}}$ are the means and the covariance matrices of the vectors f and $\tilde{f}$, respectively. 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

[2] http://yann.lecun.com/exdb/mnist.

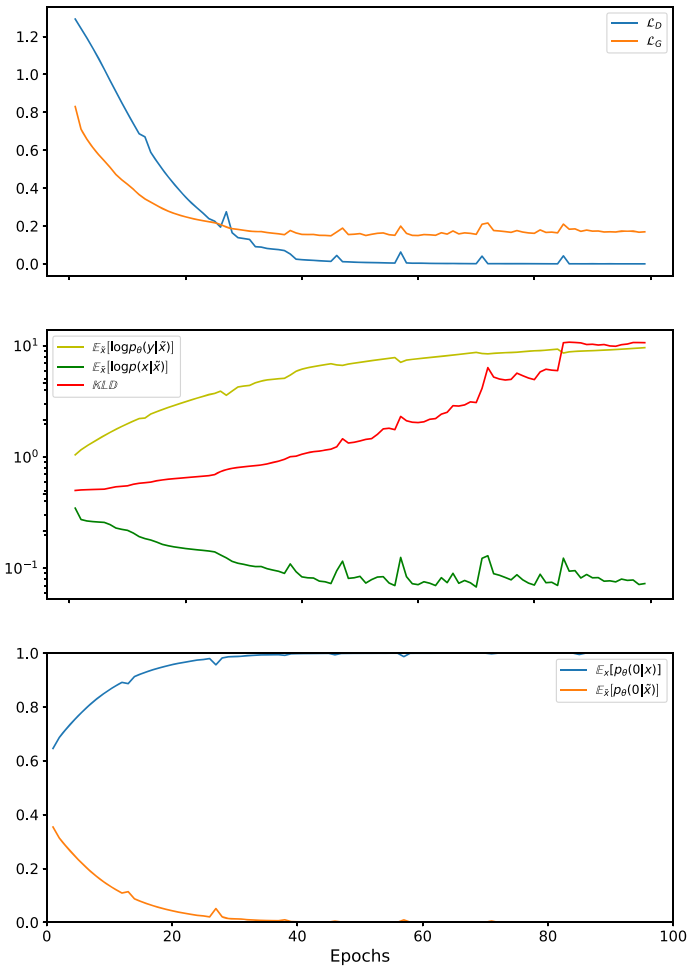

Fig. 2 Stability of the learning process

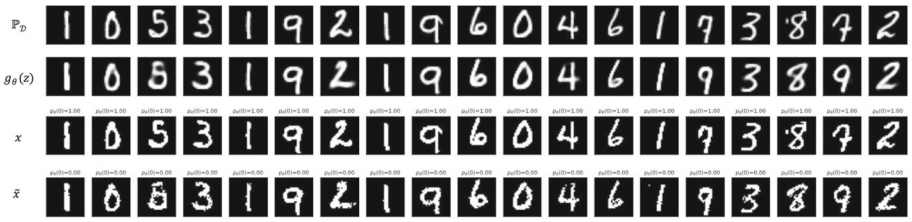

Fig. 3 Anecdotal evidence of Outlier generation on MNIST

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

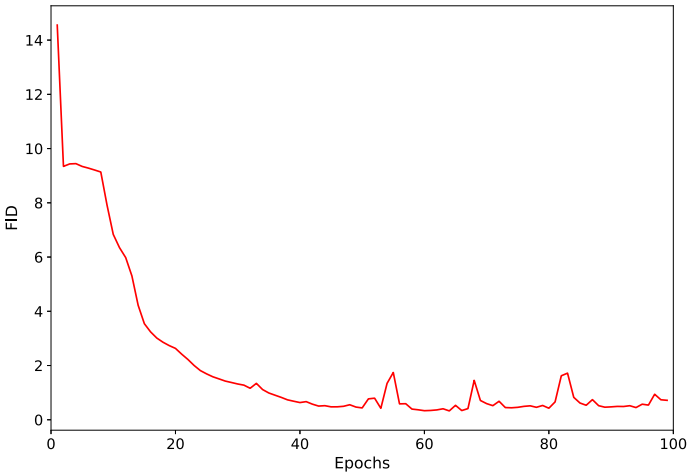

Fig. 4 FID between x and $\tilde{x}$ along the training process

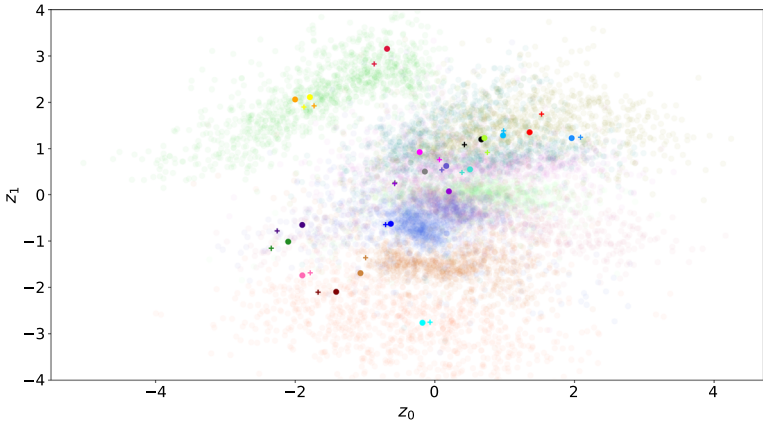

Fig. 5 Representation of real and generated images on a two-dimensional latent space

from them and sometimes they even cross the boundaries of the corresponding regions within the manifold.

4.1.2 Experiment on synthetic data

In order to further strengthen the findings in the previous section, we study the behavior of ARN in a more controlled environment where the data are synthetically generated and the underlying generation patterns can be observed. In particular, we generate a dataset with $m = 32$ features and $n = 12000$ samples, where each point is randomly sampled by choosing from two separate multivariate Gaussian distributions. The whole dataset is split into $X1$, $S1$ (sampled from the first Gaussian) and $X2$, $S2$ (sampled from the second one). Specifically, the cardinality of $X1$ and $X2$ is 6000 and 4000, respectively, whereas those of $S1$ and $S2$ are 1000

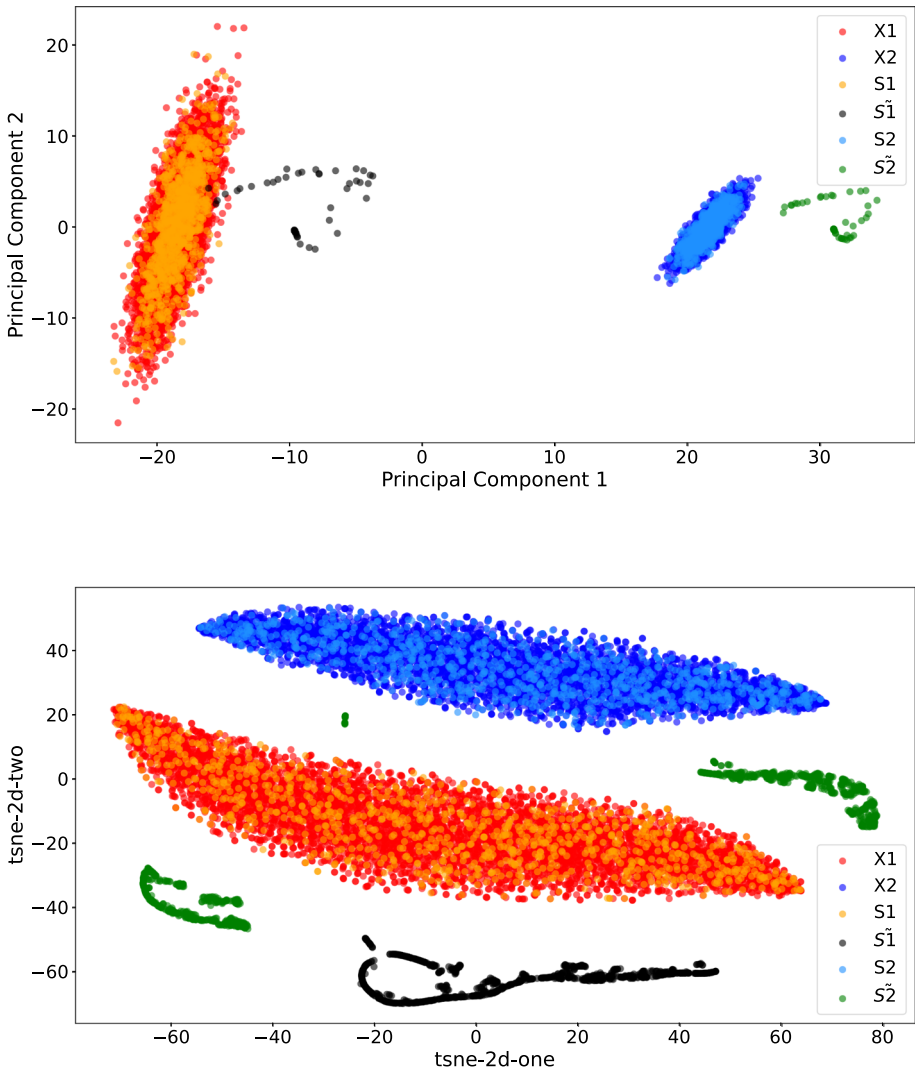

Fig. 6 Experiments on synthetic data. The generated adversarial examples ($\tilde{S1}$ and $\tilde{S2}$ in the figure) are clearly separated from the reference data points that generated them

and 1000, respectively. Figure 6 represents these points on two latent spaces, respectively, generated by using PCA and t-SNE transformations. We use $X1$ and $X2$ to train ARNand use $S1$ and $S2$ to generate adversarial samples (denoted as $\tilde{S1}$ and $\tilde{S2}$ in the figure). We can observe that the adversarial examples are clearly out of the main distributions, represented by the clusters of inlier points generated from the Gaussian priors.

4.2 Outlier detection

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

4.2.1 Datasets

The experiments focus on seven 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 toward the anomalous class. KDDCUP99$_{Rev}$ is 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%) toward 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 [54] 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 and stats on packet length and packet time. We only focus on DoH traffic and use the "benign" and "malicious" classes. Since DoH is unbalanced toward the "malicious" class label, we also consider its reverted variant that we call DoH$_{Inv}$ in the following.
- *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] [55] 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.

[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.

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

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

- *Bank*[8] is a dataset relative to direct marketing campaigns of a banking institution [56]. 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.
- Finally, **MNIST** is already described in Sect. 4.1. This dataset was used by some competitors, and hence, it is useful to consider.
- *Pump-Sensor*[9] is a sequential dataset related to failure in a water pump in which the period of observation is of 5 months, sampled each 1 min. The dataset consists of raw (numerical) values collected from 52 sensors. The normal operation system is indicated with the class "normal"; when a failure occurs, the status "broken" is reported, after that the dataset presents the status "recovering." The status "broken" and "recovering" are considered as anomalous class.

For each of the above datasets, we identify a normal and an anomalous class. For MNIST, we consider multiple instantiations where each class (digit) is considered anomalous and the others are instead considered normal. 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 1 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.

4.2.2 Evaluation protocol

In our experiments, from each dataset we sample train D_{tr}, test D_{te} and validation D_v subsets. Training data only contain normal samples, whereas validation and test contain both normal and anomalous samples. We assume that training examples are normal as in real-world scenarios, most of the available data resembles normal behaviors; anomalies are rare events and mostly we are not aware of the nature of the data. Moreover, the intuition of our work is to build a detector that can accurately separate normal and abnormal behavior without having access to the anomalous data. Therefore, the assumption is that only normal data are available and in an ideal situation, training examples are normal. Beyond the ideal situation, we also consider a setting where normal data could be contaminated, i.e., a small amount of anomalous data is contained in the dataset, but it is still considered as normal. Details about this experiment are reported in Sect. 4.3.

The sampling process is structured to guarantee that *(i)* the normal/anomalous proportions are kept consistent in the test and validation sets and *(ii)* all samples in the minority class are exploited. 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.

First of all, anomalous samples are split into two subsets (to enable the contamination process, as we will discuss in Sect. 4.3). Next, we consider the two cases that can occur with the datasets described in Table 1: Either the normal samples are less than the anomalous samples or the normal samples are greater than the anomalous ones. In the first case, the normal samples maintain the $80/15/5$ proportions on train/test/validation, and the anomalous samples $15/5$, respectively. In the second configuration, the anomalous samples maintain the $75/25$ proportions on test/validation and the normal samples $15/5$ on test/validation with respect to the entire sample (i.e., the total number of normal and anomalous samples). The rest of the normal samples are included in the train set.

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

[9] https://www.kaggle.com/datasets/nphantawee/pump-sensor-data.

Table 1 Dataset description

Dataset	Normal		Anomalous		Features				
	type	size	type	size	categorical	numeric	D_{tr}	$D_{\text{te}}\,(N/A)$	$D_{\text{v}}\,(N/A)$
KDDCUP99	Normal	97,278	DoS, probe, R2L, U2R	396,743	7	34	77,822	14,592/29,756	4,864/9,919
KDDCUP99$_{\text{Rev}}$	Normal	97,278	DoS\{smurf, neptune}, probe, R2L, U2R	8,752			81,323	11,966/3,282	3,989/1,094
KDDCUP99$_{\text{Inv}}$	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$_{\text{Inv}}$	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
MNIST	All classes but one	∼63,000	Each single class (separately)	∼7,000	-	784	∼53,000	∼2,500/∼900	∼7,000/∼2,500

To evaluate the performance of our model and to compare with the baseline methods, we compute the area under the curve (AUC) [57] as well as the area under the precision–recall Curve (AUPRC) [58]. All the experiments are performed on 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.

4.2.3 Baselines

We choose the state-of-the-art anomaly detection methods shown in Fig. 1: **FenceGAN** [19], 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** [11], a GAN-based model structured into an encoder–decoder–encoder network. In addition, we compare with **OC-SVM** [8], 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 Sect. 2. Specifically, we compute the probability that a sample is abnormal as $p(x) = \exp^{-MSE(x)}$. 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.

4.2.4 Results

Table 2 reports the results of the evaluation on the first six datasets. 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. This can also be appreciated on the MNIST dataset in Fig. 7, where we compare ARN with FenceGAN and GANomaly on each class (digit) (as opposed to all the other classes which are hence deemed normal). We can see that ARN clearly outperforms the competitor by consistently achieving higher AUC and AUPRC. Additionally, we also performed two sets of experiments in which we considered two classes that are easily interchangeable with few pixel changes. Specifically, in the first set of experiments, we considered classes 1 and 7, and in the second one, the classes 8 and 9. Each set is composed by two rounds of experiments, where at turn each class is considered as anomalous. The results are, respectively, reported in Figs. 8 and 9: ARN shows a better performance in terms of AUC and AUPRC compared to its competitors in almost all configurations, except for the one where the number 9 serves as the anomalous counterpart to the number 8.

Figure 10 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, since only few tens of features differ over 120.

It is worth mentioning that 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 are comparable since the confidence intervals of the two strategies overlap.

Table 2 Comparative analysis

Dataset	Model	AUC	AUPRC
KDDCUP99	ARN^G	*0.99 ±0.00*	*0.99 ± 0.00*
	ARN^N	1.00 ±.00	*0.99 ± 0.00*
	FenceGAN	*0.99 ± 0.00*	*0.99 ± 0.00*
	GANomaly	1.00 ±.00	1.00 ±0.00
	OC-SVM	*0.96 ± 0.00*	*0.97 ± 0.00*
	Baseline	1.00 ±0.00	1.00 ±0.00
KDDCUP99$_{Rev}$	ARN^G	*0.97 ±0.01*	**0.95 ± 0.02**
	ARN^N	**0.99 ± 0.00**	**0.95 ± 0.02**
	FenceGAN	*0.84 ± 0.01*	*0.77 ± 0.01*
	GANomaly	*0.92 ± 0.01*	*0.86 ± 0.01*
	OC-SVM	*0.81 ± 0.00*	*0.71 ± 0.00*
	Baseline	*0.91 ± 0.01*	*0.87 ±0.01*
KDDCUP99$_{Inv}$	ARN^G	1.00 ±0.00	1.00 ±0.00
	ARN^N	1.00 ±0.00	1.00 ±0.00
	FenceGAN	*0.92 ±0.03*	*0.72 ± 0.08*
	GANomaly	*0.91 ± 0.04*	*0.90 ±0.03*
	OC-SVM	*0.95 ± 0.00*	*0.82 ± 0.00*
	Baseline	1.00 ±0.00	1.00 ±0.00
NSL-KDD	ARN^G	0.99 ±0.00	0.99 ±0.01
	ARN^N	0.99 ±0.00	0.98 ±0.01
	FenceGAN	*0.96 ± 0.00*	*0.97 ± 0.00*
	GANomaly	*0.97 ± 0.01*	*0.97 ± 0.01*
	OC-SVM	*0.96 ± 0.00*	*0.97 ± 0.00*
	Baseline	0.99 ±0.00	0.98 ±0.00
DoH	ARN^G	0.99 ±.01	1.00 ±0.00
	ARN^N	0.99 ±0.01	1.00 ±0.00
	FenceGAN	*0.88 ± 0.02*	*0.97 ± 0.00*
	GANomaly	0.99 ±0.00	1.00 ±0.00
	OC-SVM	*0.88 ± 0.00*	*0.97 ± 0.00*
	Baseline	*0.96 ± 0.00*	*0.99 ± 0.00*
DoH$_{Inv}$	ARN^G	*0.98 ± 0.01*	*0.97 ± 0.02*
	ARN^N	1.00 ±0.00	**1.00 ± 0.00**
	FenceGAN	*0.89 ± 0.02*	*0.44 ± 0.05*
	GANomaly	1.00 ±.00	*0.98 ± 0.01*
	OC-SVM	*0.90 ± 0.00*	*0.49 ± 0.01*
	Baseline	*0.99 ± 0.00*	*0.91 ± 0.04*
CoverType	ARN^G	**0.94 ± 0.01**	**0.95 ± 0.01**
	ARN^N	**0.92 ± 0.04**	**0.93 ±0.03**
	FenceGAN	*0.70 ± 0.03*	*0.41 ± 0.02*
	GANomaly	*0.56 ± 0.05*	*0.30 ± 0.04*

Bold (resp. italic) values represent models with statistically higher (resp. lower) scores

Table 2 continued

Dataset	Model	AUC	AUPRC
	OC-SVM	*0.73 ± 0.02*	*0.43 ± 0.02*
	Baseline	*0.53 ± 0.02*	*0.28 ± 0.02*
CreditCard	ARN^G	–	–
	ARN^N	0.99 ±.01	*0.59 ± 0.06*
	FenceGAN	*0.90 ± 0.01*	*0.51 ± .03*
	GANomaly	*0.84 ± 0.02*	*0.36 ± 0.05*
	OC-SVM	*0.92 ± 0.01*	*0.57 ± 0.01*
	Baseline	0.99 ±0.00	**0.76 ± 0.01**
Bank	ARN^G	**0.77 ± 0.06**	**0.63 ± 0.09**
	ARN^N	**0.69 ± 0.07**	**0.50 ± 0.11**
	FenceGAN	*0.56 ± 0.01*	*0.23 ± 0.01*
	GANomaly	*0.53 ± 0.02*	*0.22 ± 0.02*
	OC-SVM	*0.60 ± 0.00*	*0.28 ± 0.00*
	Baseline	*0.65 ± 0.00*	*0.32 ± 0.01*

Bold (resp. italic) values represent models with statistically higher (resp. lower) scores

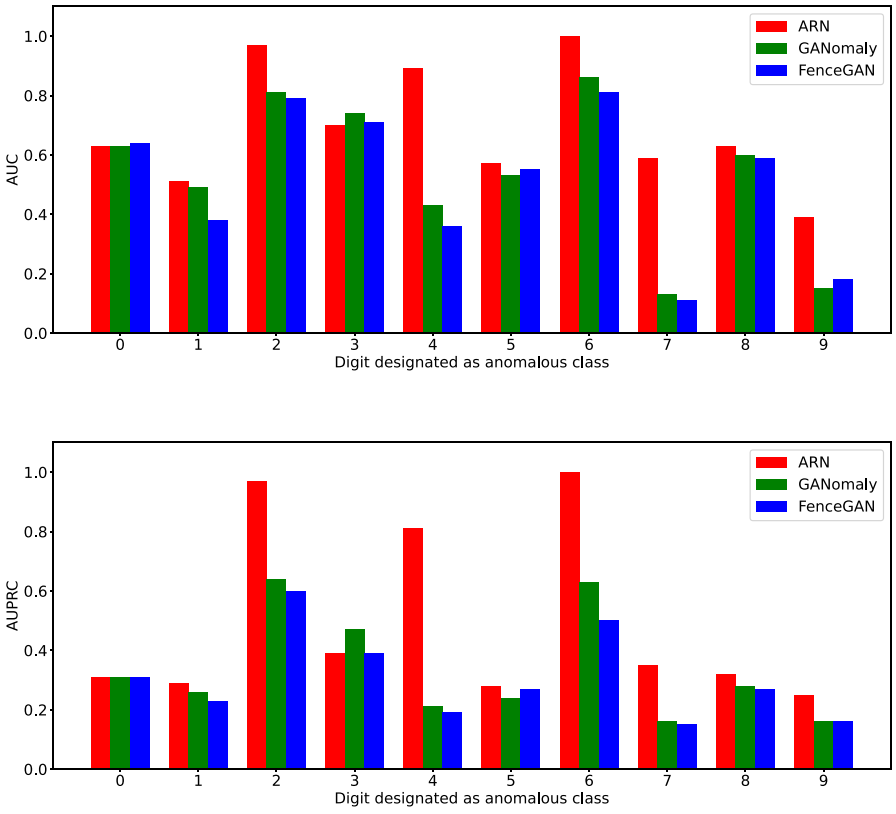

Fig. 7 Comparison between ARN , FenceGAN and GANomaly on MNIST

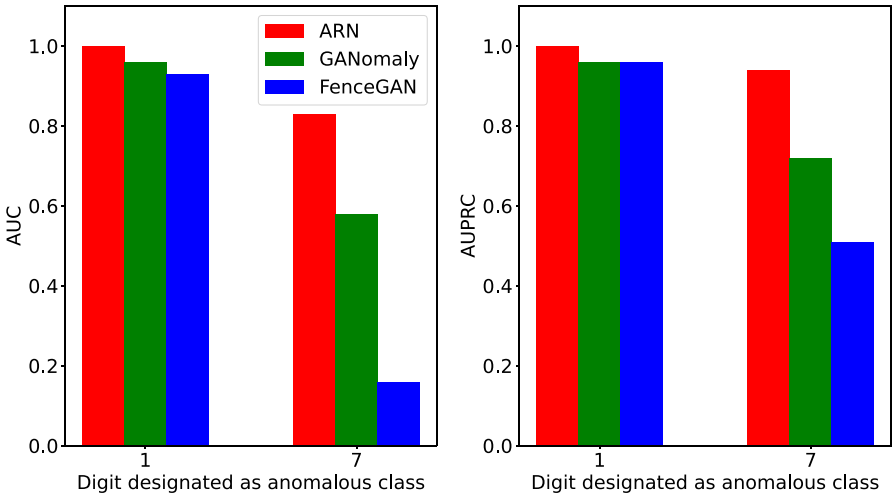

Fig. 8 Comparison between ARN, GANomaly and FenceGAN on MNIST by considering only the classes 1 and 7, which are extremely close in the pixel space

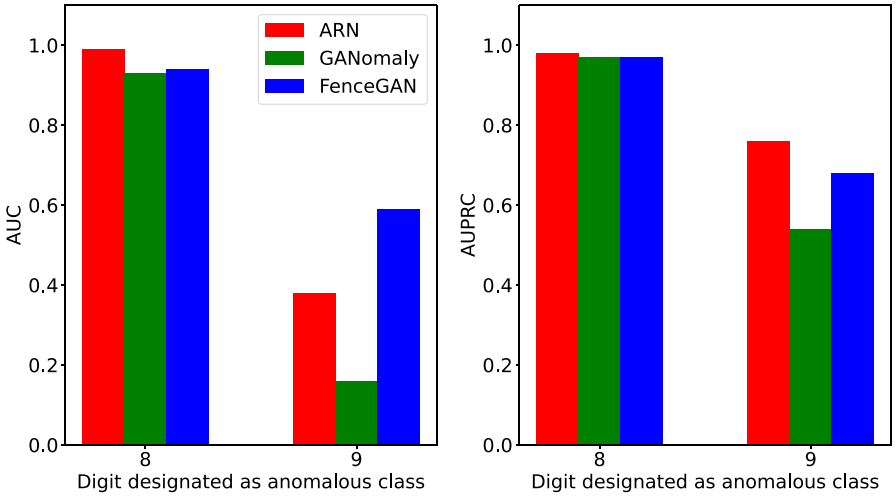

Fig. 9 Comparison between ARN, GANomaly and FenceGAN on MNIST by considering only the classes 8 and 9, which are extremely close in the pixel space

4.2.5 Experiments on sequential data

In order to assess experiments on sequential data, we provide a sequential version of our model, called **SARN**. SARN is composed by a generator and discriminator in which a recurrent neural network (RNN) [59] is adopted. In our implementation, we decide to instantiate the RNN as long short-term memory (LSTM) [60]. Since the competitors are not suited for sequential datasets, we also provide a sequential version of them (i.e., **SAE**, **SGANomaly** and **SFenceGAN** that are the sequential versions of the Baseline, GANomaly and FenceGAN, respectively). The sequential models work on sequences of temporally sorted events, so we

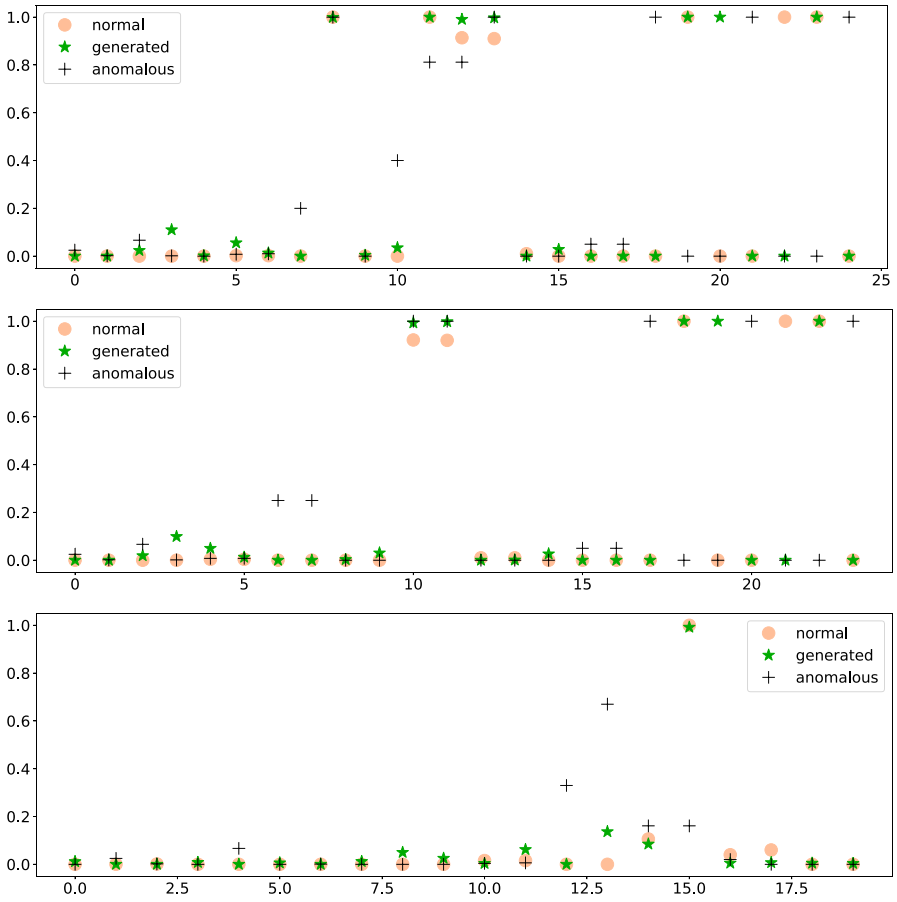

Fig. 10 Real and generated anomalies in KDDCUP99$_{Rev}$. The x-axis enumerates the data attributes that exhibit a deviation from normality higher than 10^{-3}, the y-axis normalizes the attribute values in the [0, 1] interval. Orange dots are attribute values for a random real normal sample; green stars are related to its generated anomalous counterpart; finally, black crosses represent a random real anomaly

applied a sliding window procedure to generate a set of fixed-size observation windows. The goal is to identify anomalous sequences, i.e., sequence containing events that are significantly different from the other events of the sequence.

Formally, we assume that the data are organized as a sequence of events $X = \{x_1, \ldots, x_N\}$ in which each event x_t is composed by features describing the t-*th* event in chronological order within X. For a given window $[t - w, t]$ where w is the window size, the window is anomalous if it contains an expected event, i.e., an event x_i significantly different from its neighbors $x_{t-w}, \ldots, x_{i-1}, x_{i+1}, \ldots, x_t$. The considered dataset, i.e., Pump-Sensor, is split in training, test and validation set. The training and validation set are composed by only normal windows and we have 19,962, 2218, respectively. The test set is composed by 5546 normal windows and 678 anomalous ones. The results on the sequential data are reported in Table 3, showing that our model is comparable with the competing methods.

Table 3 Comparative analysis

	SARN		SAE		SGANomaly		SFenceGAN	
Dataset	AUC	AUPRC	AUC	AUPRC	AUC	AUPRC	AUC	AUPRC
Pump sensor	0.89 ±0.04	0.64 ±0.07	0.93 ±0.01	0.54 ±0.09	*0.58 ± 0.08*	*0.20 ± 0.05*	0.81 ±0.08	0.55 ±0.09

Bold (resp. italic) values represent models with statistically higher (resp. lower) scores

4.3 Robustness

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

- *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 Sect. 3.4), 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. Table 4 reports the results in terms of AUC. 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 [12], an approach that combines Wasserstein autoencoders to detect and generates both inliers and outliers. This experiment is only illustrated on KDDCUP99_{Rev} and Bank in Table 5, presenting the results in terms of AUC. 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.

For completeness, we also examined the prediction accuracy of ARN across different sizes of normal samples. Specifically, we maintained a fixed percentage of anomalous samples, i.e., 3%, while varying the percentages for normal samples within the set $\{1, 2, 4, 6, 8, 10, 25, 50, 75, 100\}$. The results presented in Table 6 highlight the robustness of ARN, demonstrating its ability to effectively distinguish between normal and abnormal behaviors even under conditions of limited training data.

4.4 Ablation study

In a final set of experiments, we study the contribution of each component of the model to the accuracy. Table 7 presents the results in terms of AUC. 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 7, $\text{ARN}^{X-\mathbb{KLD}}$ represents model ARN^X trained without regularization, and ARN^{GE} represents ARN^G trained by optimizing the ELBO in Eq. 2. 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}.

Table 4 Robustness to contamination

	$KDDCUP99_{Rev}$			$KDDCUP99_{Inv}$			CoverType			Bank			DoH_{Inv}		
	ARN^N AUC	GANomaly AUC	FenceGAN AUC	ARN^G AUC	GANomaly AUC	FenceGAN AUC	ARN^G AUC	GANomaly AUC	FenceGAN AUC	ARN^G AUC	GANomaly AUC	FenceGAN AUC	ARN^N AUC	GANomaly AUC	FenceGAN AUC
No contamination	0.99 ±0.00	0.92 ±0.01	0.84 ±0.01	1.00 ±0.00	0.91 ±0.04	0.92 ±0.03	0.94 ±0.01	0.56 ±0.05	0.70 ±0.03	0.77 ±0.06	0.53 ±0.02	0.56 ±0.01	1.00 ±0.00	1.00 ±0.00	0.89 ±0.02
$p = 1\%$	0.99 ±0.00	0.88 ±0.02	0.83 ±0.01	1.00 ±0.00	.91 ±0.05	0.95 ±0.03	0.92 ±0.01	0.52 ±0.05	0.70 ±0.02	0.75 ±0.06	0.51 ±0.02	0.57 ±0.01	1.00 ±0.00	0.93 ±0.04	0.84 ±0.02
$p = 5\%$	0.97 ±0.01	0.81 ±0.01	0.82 ±0.01	0.99 ±0.01	0.84 ±0.03	0.89 ±0.07	0.92 ±0.01	0.51 ±0.06	0.68 ±0.02	0.74 ±0.05	0.50 ±0.02	0.55 ±0.01	0.99 ±0.00	0.76 ±0.03	0.75 ±0.05

	KDDCUP99			NSL-KDD			DoH		
	ARN^G AUC	GANomaly AUC	FenceGAN AUC	ARN^G AUC	GANomaly AUC	FenceGAN AUC	ARN^G AUC	GANomaly AUC	FenceGAN AUC
No contamination	0.99 ±0.00	1.00 ±0.00	0.99 ±0.00	0.99 ±0.00	0.97 ±0.01	0.96 ±0.00	0.99 ±0.01	0.99 ±0.00	0.88 ±0.02
$p = 1\%$	0.98 ±0.01	0.61 ±0.07	0.98 ±0.1	0.99 ±0.00	0.90 ±0.02	0.96 ±0.00	0.99 ±0.01	0.92 ±0.02	0.88 ±0.02
$p = 5\%$	0.97 ±0.01	0.49 ±0.09	0.94 ±0.1	0.98 ±0.01	0.73 ±0.05	0.96 ±0.00	0.98 ±0.01	0.73 ±0.04	0.85 ±0.02

	KDDCUP99 AUC	NSL-KDD AUC	DoH AUC
No contamination	0.99 ±0.00	0.99 ±0.00	0.99 ±0.01
$p = 1\%$	0.98 ±0.01	0.99 ±0.00	0.99 ±0.01
$p = 5\%$	0.97 ±0.01	0.98 ±0.01	0.98 ±0.01
$p = 10\%$	0.96 ±0.02	0.98 ±0.01	0.99 ±0.00
$p = 25\%$	0.86 ±0.07	0.93 ±0.04	0.96 ±0.02
$p = 50\%$	0.84 ±0.07	0.79 ±0.05	0.89 ±0.03

Table 5 Weak supervision: Comparison with WALDO

Datasets	Method	**0**% Anomalies AUC	1% Anomalies AUC	3% Anomalies AUC
KDDCUP99$_{\text{Rev}}$	ARN^N	0.99 ±0.00	0.99 ±0.00	0.99 ±0.00
	ARN^G	0.97 ±0.01	0.99 ±0.00	0.99 ±0.00
	WALDO	–	0.80 ±0.01	0.80 ±0.01
Bank	ARN^N	0.69 ±0.07	0.70 ±0.06	0.96 ±0.02
	ARN^G	0.77 ±0.06	0.83 ±0.06	0.90 ±0.04
	WALDO	–	0.56 ±0.01	0.56 ±0.01

Table 6 Weak supervision: Assessment of the predictive capability of ARN across varied training data sizes, keeping fixed the percentage of anomalous samples to 3%

Datasets	Normal sizes (%)	Method ARN^N AUC	ARN^G AUC
KDDCUP99$_{Rev}$	1	1.00 ± 0.00	0.98 ± 0.02
	2	1.00 ± 0.00	1.00 ± 0.00
	4	1.00 ± 0.00	1.00 ± 0.00
	6	1.00 ± 0.00	1.00 ± 0.00
	8	1.00 ± 0.00	1.00 ± 0.00
	10	0.98 ± 0.01	0.98 ± 0.00
	25	0.99 ± 0.00	0.99 ± 0.00
	50	0.99 ± 0.00	1.00 ± 0.00
	75	0.99 ± 0.00	0.99 ± 0.00
	100	0.99 ± 0.00	0.99 ± 0.00
Bank	1	0.90 ± 0.03	0.71 ± 0.07
	2	0.96 ± 0.01	0.89 ± 0.04
	4	0.98 ± 0.01	0.93 ± 0.03
	6	0.98 ± 0.01	0.91 ± 0.05
	8	0.98 ± 0.00	0.91 ± 0.02
	10	0.77 ± 0.05	0.84 ± 0.06
	25	0.86 ± 0.02	0.88 ± 0.05
	50	0.91 ± 0.03	0.95 ± 0.02
	75	0.95 ± 0.03	0.92 ± 0.04
	100	0.96 ± 0.02	0.90 ± 0.04

4.5 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. To stabilize the learning process, we progressively inject noise on the discriminator labels as suggested in [61]. As a result, the learning phase is generally smooth (as also illustrated in Fig. 2). All the models were

Table 7 Ablation study

Dataset	ARN^{G} AUC	ARN^{N} AUC	$\text{ARN}^{G-\mathbb{KLD}}$ AUC	$\text{ARN}^{N-\mathbb{KLD}}$ AUC	ARN^{GE} AUC
KDDCUP99	0.99 ±0.00	1.00 ±0.00	0.98 ±0.02	0.98 ±0.02	0.74 ± 0.09
$\text{KDDCUP99}_{\text{Rev}}$	0.97 ±0.01	0.99 ±0.00	0.98 ±0.01	0.96 ±0.00	0.83 ±0.05
$\text{KDDCUP99}_{\text{Inv}}$	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	0.88±0.04
NSL-KDD	0.99 ±0.00	0.99 ±0.00	0.99 ±0.01	0.98 ±0.00	0.74 ±0.07
DoH	0.99 ±0.01	0.99 ±0.01	0.73 ±0.01	0.79 ±0.02	0.99 ±0.01
DoH_{Inv}	0.98 ±0.01	1.00 ±0.00	0.99 ±0.00	0.99 ±0.00	0.83 ±0.04
CoverType	0.94 ±0.01	0.92 ±0.04	0.94 ±0.01	0.94 ±0.01	0.77 ±0.08
CreditCard	–	0.99 ±0.01	–	0.96 ±0.05	0.75 ±0.06
Bank	0.77 ±0.06	0.69 ±0.07	0.74 ±0.07	0.63 ±.07	0.62 ±.04

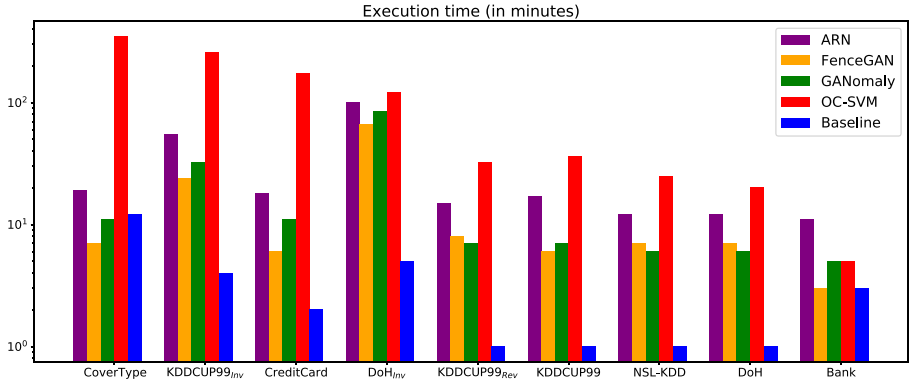

Fig. 11 Running times in minutes (log scale)

trained using the Adam optimization. We also adopt different learning rates for the discriminator and the generator. Throughout the experiments described in Sect. 4.2 through 4.4, we devised simple architectures for both the generator and the discriminator, based on linear layers equipped with batch normalization. The architecture in 4.1 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 GPUs. Figure 11 summarizes the running times for each model on the datasets described in 4.2. Here, we can notice that 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 which clearly requires much longer execution times with respect to all the other evaluated algorithms.

5 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 genera-

tion 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 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 toward 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.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10115-024-02172-w.

Acknowledgements This work has been partially supported by EU H2020 ICT48 project "Humane-AI-Net" and by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU.

Author Contributions Angelica Liguori is the principal investigator, and the other authors equally contribute to the work.

Funding Open access funding provided by Consiglio Nazionale Delle Ricerche (CNR) within the CRUI-CARE Agreement. This work has been partially supported by EU H2020 ICT48 project "Humane-AI-Net" and by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU

Data availability Analyzed datasets are listed in the GitHub repository, https://github.com/arnwg/arn, in which there is the reference for each datasets. All the datasets are free to use.

Declarations

Conflict of interest The author's declared that they have no conflict of interest.

Ethics approval Not applicable.

Code availability Yes at https://github.com/arnwg/arn.

References

1. Pang G, Shen C, Cao L, Hengel AVD (2021) Deep learning for anomaly detection: a review. ACM Comput Surv 54(1):1–38

2. Hsieh R-J, Chou J, Ho C-H (2019) Unsupervised online anomaly detection on multivariate sensing time series data for smart manufacturing. In: 2019 IEEE 12th conference on service-oriented computing and applications (SOCA), pp 90–97
3. Fernando T, Gammulle H, Denman S, Sridharan S, Fookes C (2021) Deep learning for medical anomaly detection: a survey. ACM Comput Surv. https://doi.org/10.1145/3464423
4. Mothukuri V, Khare P, Parizi RM, Pouriyeh S, Dehghantanha A, Srivastava G (2022) Federated-learning-based anomaly detection for IoT security attacks. IEEE Internet of Things J. https://doi.org/10.1109/JIOT.2021.3077803
5. Ahmed M, Mahmood AN, Islam MR (2016) A survey of anomaly detection techniques in financial domain. Fut Gener Comput Syst 55:278–288
6. Aggarwal CC (2016) Outlier analysis. Springer, Incorporated
7. Ruff L, Kauffmann JR, Vandermeulen RA, Montavon G, Samek W, Kloft M, Dietterich TG, Müller K-R (2021) A unifying review of deep and shallow anomaly detection. In: Proceedings of the IEEE
8. Schölkopf B, Williamson RC, Smola AJ, Shawe-Taylor J, Platt JC (1999) Support vector method for novelty detection. In: NIPS
9. Xian Y, Schiele B, Akata Z (2017) Zero-shot learning - the good, the bad and the ugly. In: CVPR
10. Mattia FD, Galeone P, Simoni MD, Ghelfi E (2019) A survey on gans for anomaly detection. CoRR arXiv:1906.11632
11. Akcay S, Atapour-Abarghouei A, Breckon TP (2018) Ganomaly: semi-supervised anomaly detection via adversarial training. In: ACCV
12. Rizzo SG, Pang L, Chen Y, Chawla S (2020) Probabilistic outlier detection and generation. CoRR arXiv:2012.12394
13. Zenati H, Romain M, Foo CS, Lecouat B, Chandrasekhar VR (2018) Adversarially learned anomaly detection. In: ICDM
14. Laptev N. AnoGen: deep anomaly generator. https://tinyurl.com/fbanogen
15. Kingma DP (2017) Welling, M.: Auto-encoding variational bayes. In: ICLR
16. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: NIPS, pp 2672–2680
17. Theis L, van den Oord A, Bethge M (2016) A note on the evaluation of generative models. In: ICLR
18. Hong Y, Hwang U, Yoo J, Yoon S (2019) How generative adversarial networks and their variants work: an overview. ACM Comput Surv 52(1):1–43
19. Ngo CP, Winarto AA, Li CKK, Park S, Akram F, Lee HK (2019) Fence gan: towards better anomaly detection. In: ICTAI
20. Makhzani A, Shlens J, Jaitly N, Goodfellow I (2016) Adversarial autoencoders. In: ICLR
21. Tolstikhin I, Bousquet O, Gelly S, Schoelkopf B (2019) Wasserstein auto-encoders. In: ICLR
22. Liu FT, Ting KM, Zhou Z-H (2008) Isolation forest. In: ICDM
23. Ramaswamy S, Rastogi R, Shim K (2000) Efficient algorithms for mining outliers from large data sets. In: SIGMOID
24. Bank D, Koenigstein N, Giryes R (2020) Autoencoders. CoRR arXiv:2003.05991
25. Chen Z, Yeo CK, Lee BS, Lau CT (2018) Autoencoder-based network anomaly detection. In: WTS
26. An J, Cho S (2015) Variational autoencoder based anomaly detection using reconstruction probability. Spec Lect IE 2(1):1–18
27. Alfeo AL, Cimino MGCA, Manco G, Ritacco E, Vaglini G (2020) Using an autoencoder in the design of an anomaly detector for smart manufacturing. Patt Recognit Lett 136:272–278
28. Hawkins S, He H, Williams GJ, Baxter RA (2002) Outlier detection using replicator neural networks. In: DaWaK
29. Zhou C, Paffenroth RC (2017) Anomaly detection with robust deep autoencoders. In: KDD
30. Cassavia N, Folino F, Guarascio M (2022) Detecting dos and ddos attacks through sparse u-net-like autoencoders. In: International conference on tools with artificial intelligence (ICTAI), pp 1342–1346
31. Tian K, Zhou S, Fan J, Guan J (2019) Learning competitive and discriminative reconstructions for anomaly detection. In: AAAI, vol. 33
32. Abhaya A, Patra BK (2023) An efficient method for autoencoder based outlier detection. Exp Syst Appl. https://doi.org/10.1016/j.eswa.2022.118904
33. Schlegl T, Seeböck P, Waldstein SM, Schmidt-Erfurth U, Langs G (2017) Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: IPMI
34. Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X (2016) Improved techniques for training gans. In: NIPS
35. Schlegl T, Seeböck P, Waldstein SM, Langs G, Schmidt-Erfurth U (2019) f-anogan: fast unsupervised anomaly detection with generative adversarial networks. Med Image Anal 54:30–44

36. Zenati H, Foo CS, Lecouat B, Manek G, Chandrasekhar VR (2019) Efficient gan-based anomaly detection. CoRR arXiv:1802.06222
37. Kim Y, Choi S (2019) Forward-backward generative adversarial networks for anomaly detection. In: ICML
38. Donahue J, Krähenbühl P, Darrell T (2017) Adversarial feature learning. In: ICLR
39. Akçay S, Atapour-Abarghouei A, Breckon TP (2019) Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection. In: IJCNN
40. Zhang Z, Li W, Ding W, Zhang L, Lu Q, Hu P, Gui T, Lu S (2023) Stad-gan: unsupervised anomaly detection on multivariate time series with self-training generative adversarial networks. ACM Trans Knowl Discov Data. https://doi.org/10.1145/3572780
41. Vu HS, Ueta D, Hashimoto K, Maeno K, Pranata S, Shen SM (2019) Anomaly detection with adversarial dual autoencoders. CoRR arXiv:1902.06924
42. Zhang L, Xie X, Xiao K, Bai W, Liu K, Dong P (2022) Manomaly: mutual adversarial networks for semi-supervised anomaly detection. Inf Sci 611:65–80. https://doi.org/10.1016/j.ins.2022.08.033
43. Chen J, Sathe S, Aggarwal C, Turaga D (2017) Outlier detection with autoencoder ensembles. In: SDM
44. An P, Wang Z, Zhang C (2022) Ensemble unsupervised autoencoders and gaussian mixture model for cyberattack detection. Inf Process Manag. https://doi.org/10.1016/j.ipm.2021.102844
45. Han X, Chen X, Liu L-P (2020) Gan ensemble for anomaly detection. In: AAAI
46. Cassavia N, Caviglione L, Guarascio M, Liguori A, Zuppelli M (2022) Ensembling sparse autoencoders for network covert channel detection in IoT ecosystems. In: ISMIS. Lecture notes in computer science, vol. 13515, pp 209–218
47. Guarascio M, Zuppelli M, Cassavia N, Manco G, Caviglione L (2022) Detection of network covert channels in IoT ecosystems using machine learning. In: ITASEC. CEUR workshop proceedings, vol. 3260, pp 102–113
48. Blei DM, Kucukelbir A, McAuliffe JD (2018) Variational inference: a review for statisticians. J Am Stat Assoc 112(518):859–877
49. Cover TM, Thomas JA (2006) Elements of information theory
50. Kullback S, Leibler RA (1951) On information and sufficiency. Ann Math Stat 22(1):79–86
51. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Kopf A, Yang E, DeVito Z, Raison, M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, Chintala S (2019) Pytorch: an imperative style, high-performance deep learning library. In: NeurIPS
52. Jang E, Gu S, Poole B (2017) Categorical reparameterization with gumbel-softmax. In: ICLR
53. Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S (2017) Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: NIPS
54. Tavallaee M, Bagheri E, Lu W, Ghorbani AA (2009) A detailed analysis of the kdd cup 99 data set. In: CISDA
55. Pozzolo AD, Caelen O, Johnson RA, Bontempi G (2015) Calibrating probability with undersampling for unbalanced classification. In: SSCI
56. Moro S, Cortez P, Rita P (2014) A data-driven approach to predict the success of bank telemarketing. Decis Support Syst 62:22–31
57. Melo F (2013) Area under the ROC curve
58. Saito T, Rehmsmeier M (2015) The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. PloS One 10(3):e0118432
59. Graves A (2012) Supervised sequence labelling with recurrent neural networks. Stud Comput Intell. https://doi.org/10.1007/978-3-642-24797-2
60. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
61. Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X, Chen X (2016) Improved techniques for training gans. In: Advances in neural information processing systems, vol. 29

Angelica Liguori received her Master's degree in Computer Engineering from the University of Calabria (UNICAL), Italy, in 2019. From the same institution, she received a Ph.D. degree in Information and Communication Technologies (ICT) in 2024. She is currently a researcher at the Institute for High Performance Computing and Networking - National Research Council of Italy (ICAR-CNR) and a contract professor at University of Calabria, Italy. Her research interests include machine and deep learning. Specifically, she is interested in developing solutions in the area of Anomaly Detection/Generation in data sets. She has served as a reviewer for several international conferences and journals.

Ettore Ritacco has served as a Researcher (RTDB) at the University of Udine, Department of Mathematics, Computer Science and Physics. Prior to this role, starting from February 2015, he held a position as a Researcher at the Institute of High Performance Computing and Networks (ICAR-CNR) of the National Research Council of Italy, located in Rende (CS) (Italy). His expertise spans across data science, data analytics, and the enabling technologies that drive them. His research focuses on a variety of areas including Data Generation, Generative AI, User Profiling and Behavioral Modeling, Anomaly Detection, Smart Maintenance, Social Network Analysis, Recommendation Systems, Information Propagation and Diffusion, Profiling for Cyber Security, as well as Latent Factor and Deep Learning models. Ettore is deeply engaged in exploring the latest frontiers of Computer Science and Technology, particularly those aimed at analyzing Complex Big Data.

Francesco Sergio Pisani holds a Ph.D. and an M.Sc. in Computer Engineering from the University of Calabria, Italy. Currently, he is a research fellow at the Institute for High Performance Computing and Networking (ICAR-CNR) of the Italian National Research Council. His research mainly focuses on machine and deep learning, recommender systems, ensemble learning, knowledge discovery and data mining for cybersecurity, computer vision, and anomaly detection. Currently, he is involved in national projects concerning machine learning and cybersecurity and on an industrial project to build an advanced platform for customer engagement satisfaction.

Giuseppe Manco graduated summa cum laude in computer science and received the PhD degree in computer science from the University of Pisa. He is currently Research Manager at the Institute of High Performance Computing and Networks (ICAR-CNR) of the National Research Council of Italy and a contract professor at University of Calabria, Italy. His current research interests include Artificial Intelligence, knowledge discovery and data mining, Cybersecurity, Recommender Systems, and Social Network Analysis. He has been the coordinator of several national and international research projects. He has been serving in the organization of several international/national conferences, including: IEEE ICDM, ECMLPKDD, SIAM SDM, PAKDD. He is currently serving as an associate editor for the Journal of Intelligent Information Systems, Knowledge and Information Systems and Machine Learning Journal.