mathematics

Article

GAN-Based Anomaly Detection Tailored for Classifiers

Ľubomír Králik, Martin Kontšek *, Ondrej Škvarek and Martin Klimo

Department of Communication Networks, Faculty of Management Science and Informatics, University of Žilina, 010 26 Žilina, Slovakia; lubomir.kralik@fri.uniza.sk (Ľ.K.); ondrej.skvarek@fri.uniza.sk (O.Š.); martin.klimo@fri.uniza.sk (M.K.)

* Correspondence: martin.kontsek@fri.uniza.sk

Abstract: Pattern recognition systems always misclassify anomalies, which can be dangerous for uninformed users. Therefore, anomalies must be filtered out from each classification. The main challenge for the anomaly filter design is the huge number of possible anomaly samples compared with the number of samples in the training set. Tailoring the filter for the given classifier is just the first step in this reduction. Paper tests the hypothesis that the filter trained in avoiding "near" anomalies will also refuse the "far" anomalies, and the anomaly detector is then just a classifier distinguishing between "far real" and "near anomaly" samples. As a "far real" samples generator was used, a Generative Adversarial Network (GAN) fake generator that transforms normally distributed random seeds into fakes similar to the training samples. The paper proves the assumption that seeds unused in fake training will generate anomalies. These seeds are distinguished according to their Chebyshev norms. While the fakes have seeds within the hypersphere with a given radius, the near anomalies have seeds within the sphere near cover. Experiments with various anomaly test sets have shown that GAN-based anomaly detectors create a reliable anti-anomaly shield using the abovementioned assumptions. The proposed anomaly detector is tailored to the given classifier, but its limitation is due to the need for the availability of the database on which the classifier was trained.

Keywords: anomaly detection; generative models; deep neural networks; machine learning

MSC: 68T07

Citation: Králik, Ľ.; Kontšek, M.; Škvarek, O.; Klimo, M. GAN-Based Anomaly Detection Tailored for Classifiers. *Mathematics* **2024**, *12*, 1439. https://doi.org/10.3390/math12101439

Academic Editor: Radu Tudor Ionescu

Received: 17 April 2024
Revised: 30 April 2024
Accepted: 4 May 2024
Published: 7 May 2024

1. Introduction

Today, an important task in analyzing information is the recognition of unusual data patterns (points, events, observations). Such unusual data are referred to as outliers or anomalies [1]. According to Hawkins [2], 'An outlier is an observation which deviates so much from other observations as to arouse suspicions that different mechanism generated it'. Ref. [1] explains that one or more generating processes usually create the observed data. When the generating process behaves unusually, outliers are observed. The recognition of outliers can provide valuable information on observed applications or the systems inside.

We include the process of outlier recognition into anomaly detection. Anomaly detection should be viewed with an extra notion, as their presence indicates an unusual event or significant situation that may cause harm when not dealt with properly. Moreover, mishandling anomalies can result in incorrect decision-making, causing invalid predictions or even wrong explanations.

Anomaly detection is crucial in many fields and applications, including in healthcare [3–6], cybersecurity and computer networks with Fast Reroute mechanisms [7,8], financial industry [9–12], robotics [13–15], and video surveillance [16–18], among many others [19–22].

Anomalies are usually detected in large amounts of data, so they must be processed automatically. This is typically done using machine learning (ML) methods. Supervised, semi-supervised, or unsupervised anomaly-detection approaches are known. We use the semi-supervised anomaly-detection approach in our work.

Machine learning methods for anomaly detection can be divided into two main categories: classical ML methods (e.g., isolation forests [23,24], one-class support vector machine [25,26]); and new deep learning-based methods like autoencoders (AE) [27–29], generative adversarial networks (GAN), long short-term memory models (LSTM) [30,31], temporal convolutional networks (TCN) [32,33], or transformers [34]. Deep learning methods extract significant features from data; thus, they can use significant properties unseen by users. On the other hand, these methods' limitations include using only the training set as a normal data reference and, optionally, specific databases as anomaly references. This raises the question of whether other anomalies that are not covered can be detected and to what extent.

Our article focuses on the GAN-based model. GAN model principles and implementation cases can be found in [35,36]. A review analysis of publications of GAN models and their applications, based on libraries such as Embase (Scopus), WoS, and PubMed, can be found in [37]. A systematic literature review on the relation between anomaly-detection techniques and types of GAN is presented in [38]. Article [39] summarizes the evolution of GAN-based anomaly-detection methods and discusses the theory and their application in different areas. Ref. [40] investigates deep anomaly detection with comprehensive taxonomy categories and methods. Authors in [41] survey the principal GAN-based anomaly-detection methods and present empirical validations of the GAN models for anomaly detection on different datasets (MNIST, CIFAR-10, etc.) using an open-source toolbox. They also supply references to the literature that originally proposed concepts of GAN-based models.

Authors in [42] deeply analyzed the GAN-based methods and propose categorizing them into three main categories: AnoGAN [43], EGBAD [44], and GANomaly [45]. They state that almost all other GAN-based approaches for anomaly detection are based on one of these three methods. AnoGAN-based methods are built on a classic GAN structure where the generator and discriminator are implemented by a convolutional neural network. Since AnoGAN-based methods are relatively simple, they often perform poorly and are not suited for anomaly-detection tasks. This is primarily due to the missing inverse mapping. The anomaly score relies solely on comparing samples rather than comparing the latent spaces. EGBAD-based methods extend the idea of AnoGAN by introducing a bidirectional GAN network instead of DCGAN with an extra encoder that maps the real samples back to the latent space. This also allows the discriminator to consider the latent space of the tested sample. On the other hand, there is no guarantee that reconstructed samples are accurately remapped to the same latent space as the tested samples. Therefore, even if two samples appear close in the sample space, they may end up far apart from each other once mapped onto the latent space. GANomaly-based methods extend the idea of EGBAD and use the second encoder to encode the data after the first reconstruction. As a result, the encoded latent variable can be compared with the re-encoded latent variable. Additionally, the accuracy may be further enhanced by combining the measurements of the original sample and its reconstruction. On the contrary, the GANomaly-based methods may be much more computationally demanding due to the inner architecture. Similar to EGBAD, the distances in the sample and latent spaces may differ.

We use a GAN-based model to detect anomalies. The GAN network comprises two adversarial networks, a generator, and a discriminator, which are trained in tandem [46]. The role of the generator is to produce data (e.g., images) that resemble real samples. The discriminator must classify these generated data as fakes. Thus, the trained discriminator can be used to distinguish anomalies from real samples. Moreover, the generator can create new data samples (fakes) like the original data. Therefore, additional "realistic" data samples can be generated.

Ref. [46] proposes using generated samples to improve discriminator anomaly-detection capabilities further. Our article extends this idea. An additional anomaly discriminator is trained using fakes and 'near' anomalies generated by the GAN's generator. The contributions of this study are as follows:

- The proposed approach is based on GAN-generated near anomalies. They are not specific like database-created counterexamples, but they are general worst-case anomalies (from the detection point of view) given by the classifier application;
- The obtained results confirm that specific databases applied as anomaly samples are mostly detected with higher probability than GAN-generated near anomalies. An exception is a database with samples similar to the training samples (e.g., MNIST vs. SVHN). However, if the user wants to treat these as anomalies as well, they can include them in the training of the anomaly detector;
- We optimized the additional anomaly detector for the best fake–near anomaly ratio. This approach gives better results compared to published anomaly-detection methods.

The remainder of this paper is organized as follows: Section 2 serves as a background for understanding the proposed approach. Section 3 discusses the training and testing databases, the neural network architectures, and the methods applied to train them. Section 4 presents the experimental results, and Section 5 discusses them. Finally, Section 6 concludes the paper.

2. Background

Pattern recognizers implemented by deep neural networks represent a nonlinear transform that, for any input (picture), gives some output (class). For an uninformed user, this may cause incorrect use of the recognition result. Therefore, the prior anomalous pattern detector must solve the problem of whether the input sample matches samples from the set on which the recognizer has been trained.

The supervised neural networks were gained from the occurrence samples of all recognized classes in the training set. However, a supplement to the given training set gives so many anomalies that a curse of dimensionality occurs, and the selection of anomalies raises questions even when testing anomaly detectors. The idea that anomalies are samples formed by a different mechanism than the training samples [2] leads us to use generative models. Generative adversarial networks (GAN) generate samples (fakes) similar to the training samples and may be used for the training set augmentation. Fake generation is based on the nonlinear transformation of noise into a picture. A standard Gaussian noise script is usually used: d a standard Gaussian noise $\mathcal{N}(0,\ 1)$. We use the Chebyshev vector norm for the generator seed $\boldsymbol{n} = (n_1, n_2 \ldots, n_N,)$ characterization that gives

$$\|\boldsymbol{n}\| = \max_{i=1,\ldots,\mathrm{N}} \{|n_i|\}, n_i \in \mathcal{N}(0,1),$$

and the following probability density distribution:

$$f(x) = \frac{d}{dx}\prod_{i=1}^{N} P(n_i < x) = \frac{N}{\sqrt{2\pi}} e^{-x^2/2} \left(\frac{1}{2} + \frac{1}{2} erf\left(\frac{x}{\sqrt{2}}\right)\right)^{N-1} \tag{1}$$

As we can see in Figure 1, seeds applied for the training of generating fakes have a limited size because the probability of size above a certain value (≈5) is negligible. We assume that these seeds will generate anomalies if not used for fake generation. Also, small seeds (under 1) are rarely used for fake generation, but they are like centroids; therefore, they do not have the character of anomalies. This is the fundamental hypothesis of the paper. Of course, this hypothesis does not solve the dimensionality curse problem because the space for anomaly generation is unlimited. Therefore, the second assumption used in the paper states that if the anomaly detector recognizes the anomalies generated by the seed with a given size (norm), it will be able to recognize anomalies generated by the seed with a higher norm. We use seeds with a norm below a certain value (≈5) for fake generating and seeds with a norm above this value for anomaly generation. While fakes can be interpreted as augmented samples in the training set, we call them "far real" samples.

Figures 2 and 3 support the hypotheses. Figure 2 indicates the anomaly convergence with increasing the seed norm even independently of the generated class. In contrast, Figure 3 shows a divergence in anomaly shape in the same class for different starting seeds if their norm increases. The anomalies are the most similar to fakes, generated by seed norms slightly above the border value (interval from 6 to 7), which we call "near anomalies", while anomalies with larger seed norms are called "far anomalies". Figures 2 and 3 demonstrate that the seed norm above 15 generates so far anomalies that they are not needed for testing. If the GAN is trained correctly, the probability that "far" anomalies will look like fakes is negligible.

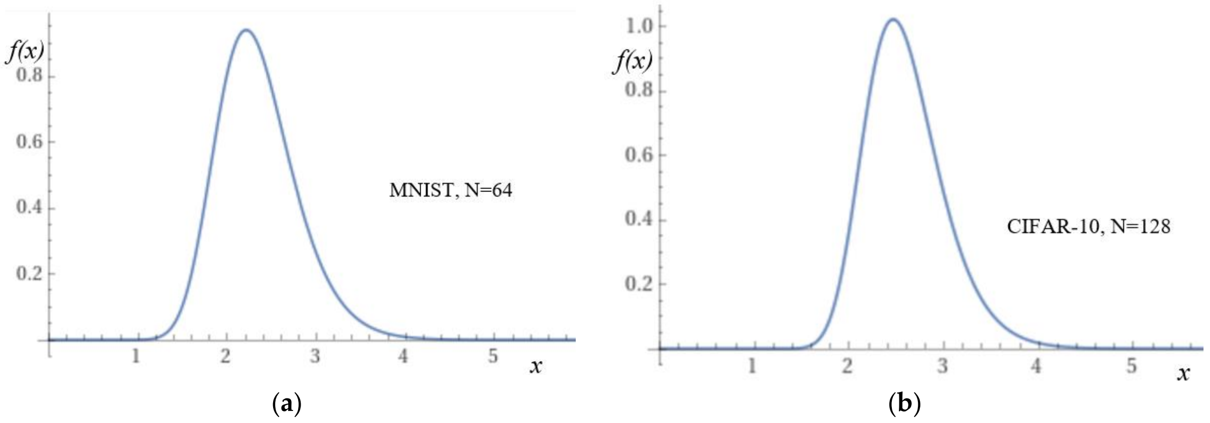


Figure 1. Chebyshev norm distribution for normally distributed generator random seeds (**a**) N = 64, (**b**) N = 128.

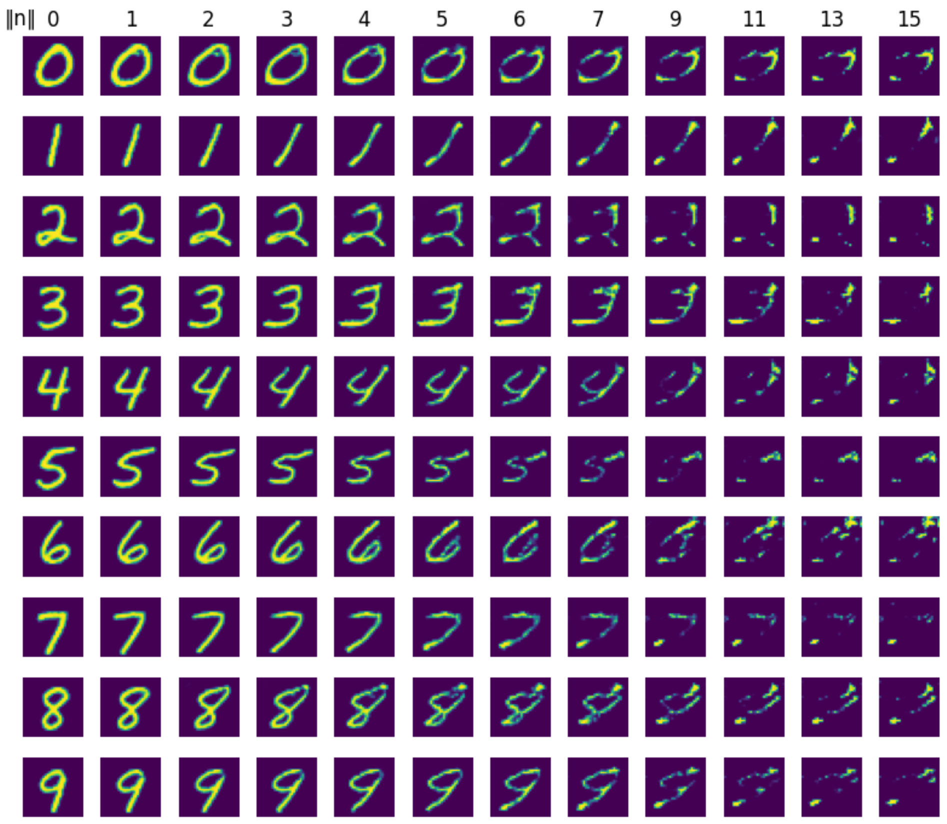


Figure 2. Samples from 10 classes were generated with the same random seed, modifying their norm.

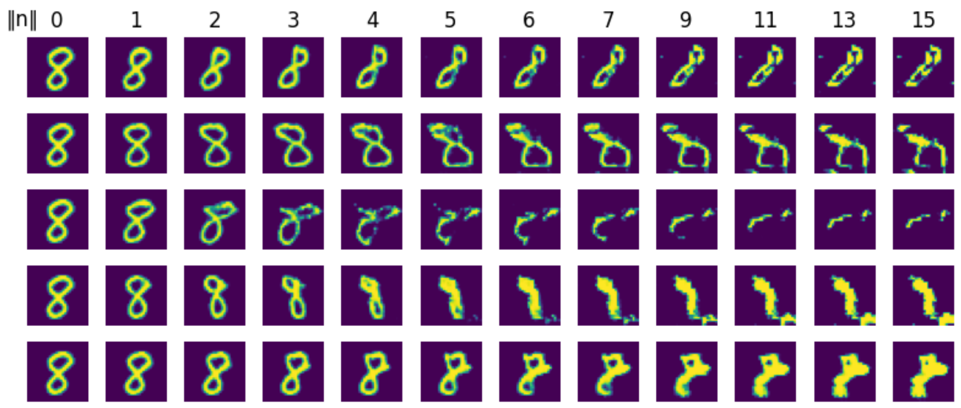


Figure 3. Class 8 samples were generated with different random seeds according to their norm.

The Chebyshev norm was chosen for the following reasons:

- Experiments with MNIST anomalies show that the Manhattan or Euclid norm (sum of all coordinate differences, in absolute values, or squares) gives a small anomality compared to the change in one coordinate (the same Chebyshev norm–maximum absolute difference over all coordinates);
- It is a well-known fact that, in the high dimensional space, a hypersphere with a Euclidean metric embedded in a hypercube (a hypersphere with a Chebyshev metric) fills only a negligible fraction of it, indicating that the hypercube represents a significantly larger subspace of the generated anomalies;
- the calculation of the Chebyshev metric in high dimensional space is faster than the calculation of the Euclid metric.

Figure 2 presents the output from the fake generator for 10 classes when the same random seed was applied to the generator input. The generator gives the class centroid samples for $\|n\| = 0$. We can see that samples slightly converge to the same anomaly sample with the increasing norm, regardless of the class. By contrast, Figure 3 shows this convergence according to the seed norm if the different seeds are applied for the same class. Figure 3 also demonstrates that the anomaly radius border depends on the seed. This phenomenon was studied in [47] but needs much human work. This paper simplifies this aspect, and we suppose that a seed generates anomalies with the norm $||n|| > r_A$, i.e., outside a hypersphere, not a hyperelipsoid.

3. Materials and Methods

3.1. Training Sets and Pattern Recognizers

Although the MNIST dataset [48] consists of grayscale images of handwritten digits and represents a "Hello world" in pattern recognition, it simply illustrates the ideas used in the article. CIFAR-10 [49] also showed results for slightly more complex datasets, as it consists of RGB images belonging to 10 individual classes. To evaluate the performance of anomaly detection, examples from the test sets of the following datasets were considered anomalous. Each dataset was balanced to consist of 10,000 images, either by default or by random sampling from the corresponding database.

For MNIST database:

(a) FashionMNIST [50] is a collection of grayscale images showcasing various fashion items, such as clothing, shoes, and accessories. The images are divided into 10 categories;

(b) Omniglot [51]. The Omniglot dataset consists of handwritten characters from various alphabets, including both modern and ancient scripts, such as Latin, Cyrillic, or Greek. Each character in the dataset is handwritten multiple times by different individuals to capture the natural variations in handwriting styles;

(c) notMNIST [52]. The notMNIST dataset consists of images of characters and letters from the English alphabet (from A to J) in various fonts and styles;
(d) CIFAR-10-bw. This dataset consists of grayscale CIFAR-10 images resized to 28 × 28 pixels;
(e) SVHN-bw. This dataset consists of grayscale SVHN [53] images resized to a resolution of 28 × 28 pixels;
(f) Uniform. The synthetic Uniform dataset comprises images where each pixel is independently and identically sampled from a uniform distribution [0, 1];
(g) Gaussian. The synthetic Gaussian dataset comprises images where each pixel is independently and identically sampled from a normal distribution with a mean of 0.5 and a standard deviation of 0.5 and then clipped to the range of [0, 1].

For the CIFAR-10 database:

(h) CIFAR-100 [49]. CIFAR-100 is like the CIFAR-10 dataset, except the images come from 100 different classes. There is no overlapping class in these datasets. Each dataset contains distinct sets of classes;
(i) MNIST-resized. We modified the original MNIST database to have a resolution of 32 × 32 pixels and duplicated the values to have three channels;
(j) LSUN [54]. The LSUN (Large-Scale Scene Understanding) dataset contains diverse images representing various indoor and outdoor scenes such as living rooms, bridges, streets, bedrooms, etc. We resized the original dataset to 32 × 32 pixels;
(k) SVHN. The SVHN (Street View House Numbers) dataset is a collection of real-world images of digits, such as house numbers visible on buildings and residences in Google Street View images. We created two versions of the SVHN dataset—for training and testing purposes;
(l) Uniform and Gaussian (identically to (f) and (g)).

We adapted residual network architectures for MNIST and CIFAR-10 classifiers. For both, we used pre-activation residual blocks (RBz) with two 3 × 3 convolutional layers, each with z filters and LeakyReLU with the 0.2 slope coefficient. In the case of MNIST, we used the following settings: RB32, AvgPool, RB32, AvgPool, RB32, RB32, AvgPool, RB32, RB32, GlobalAvgPool, Dense, and Softmax. For CIFAR-10, the WideResNet16-4 [55] architecture with GlobalAvgPool was used as the backbone for feature extraction, followed by a simple classification head with an additional 256 features. We achieved accuracies of 98.27% and 94.77% on MNIST and CIFAR-10, respectively.

3.2. GAN Network and Anomaly Detection

To filter anomalies for the specific pattern recognizer, we use a recognized class to support WGAN-GP [56] (see Figure 4), as it eliminates the problem of mode collapse and supports the diversity of generated samples by utilizing the Wasserstein distance. Unlike the standard GAN model, the output of the discriminator does not produce a probability of how real or fake the input sample is; rather, the output of WGAN-GP is a real value that correlates with how closely the input sample resembles the training images. This is achieved by removing the sigmoid normalization layer on the discriminator output. At the same time, gradient penalty (GP) is used, which provides a 1-Lipschitz function. WGAN-GP is based on WGAN [57]. With WGAN, the discriminator weights are clipped to a small value to prevent the outputs from reaching high values. WGAN-GP solves this via the gradient penalty. When training WGAN and WGAN-GP, we typically train the discriminator more times than the generator, typically at a ratio of 5:1 (in the original article); we trained at ratios of 3:1 and 5:1 in the cases of MNIST and CIFAR-10 databases, respectively. The generator G was trained according to the loss function:

$$G_{LOSS} = \alpha L_{ADV} + (1-\alpha)L_C, \tag{2}$$

$$L_{ADV} = -\mathrm{F}, \tag{3}$$

$$L_C = CrossEntropy(y, C(G_F)), \tag{4}$$

$$F = \frac{1}{N}\sum D\left(G_F, \hat{C}(G_F)\right), \tag{5}$$

$$G_F = G\left(n_F, \widetilde{y}\right), \tag{6}$$

where C is the fully trained pattern recognizer that we want to protect, $C(\cdot)$ is the softmax output for the given sample and $\hat{C}(\cdot)$ is the corresponding one-hot representation of the winning class. The hyperparameter α influences the importance of adversarial loss and classification loss. Lower values of α force the generator to generate images correctly classified by the given classifier C, even at the cost of artifact-like generated images. Conversely, with higher values of α, we force the generator to generate images that resemble the images in the training set. Thus, we set the value of α to 0.5 in all experiments. The discriminator was trained according to the loss function:

$$D_{LOSS} = F - R + \lambda GP, \tag{7}$$

$$R = \frac{1}{N}\sum D(x, y), \tag{8}$$

$$F = \frac{1}{N}\sum D(G_F, y), \tag{9}$$

$$G_F = G\left(n_F, \widetilde{y}\right), \tag{10}$$

where R is the average discriminator score for real samples x associated with the true label y, and F is the average discriminator score for fake samples (generated by the generator with the fake seed norm $n_F = \|\boldsymbol{n}\| < r_A = 5$ and randomly generated class label $\widetilde{y}$). Note that we conditioned both the discriminator and generator. For the final WGAN-GP training, we followed the general Algorithm 1 proposed in [56]. Details about the hyperparameter settings are provided in Sections 4.2 and 4.3.

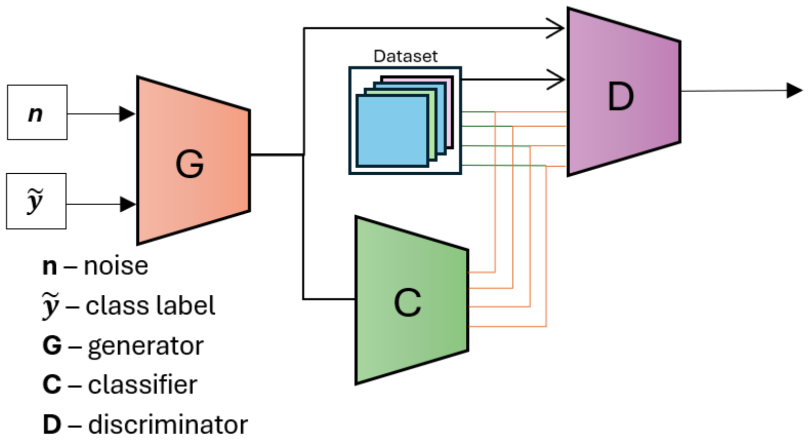
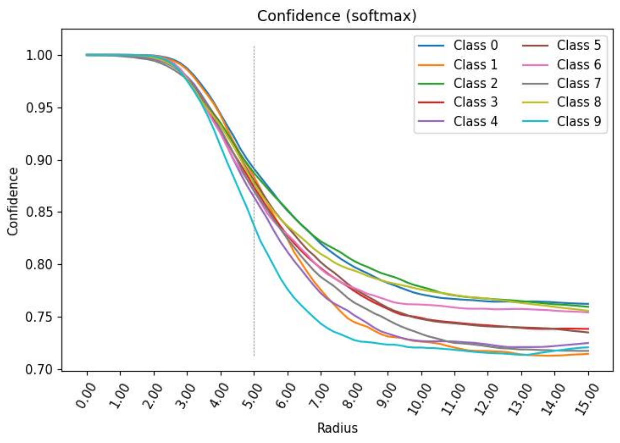


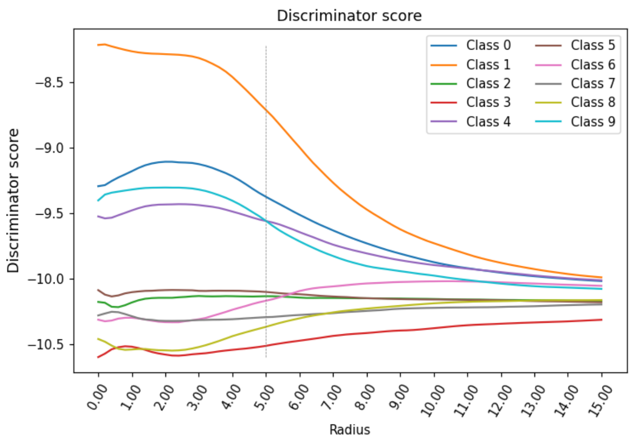
Figure 4. WGAN-GP architecture supported by pattern classifier.

The basic idea assumes that the discriminator in a standard-trained GAN will indicate the anomalies because they are unlike the training samples. This is not true, just as it is not true that classifier confidence indicates anomalies (see Figure 5).

We followed the proposed ResNet-like architectures for the generator and discriminator in the paper [56]. The detailed description of MNIST and CIFAR-10 WGAN-GP architectures is provided in Tables 1 and 2, respectively. We do not apply batch normalization within the residual block in discriminator models. All weights are initialized from a zero-centered normal distribution with a standard deviation 0.02. We chose LeakyReLU as the activation function and set the slope of the leak to 0.2. UpSampling refers to the interpolative resizing using the nearest interpolation approach.



(**a**)



(**b**)

Figure 5. Classifier (**a**); and untrained discriminator (**b**) average outputs if the input seed norm (radius) increases.

Table 1. Detailed description of MNIST WGAN-GP architecture.

Generator				**Discriminator**			
	Settings	**Resample**	**Output Shape**		**Settings**	**Resample**	**Output Shape**
Input $(n, \tilde{y})$	-	-	74	a: Input (x)	-	-	28 × 28 × 1
Dense	-	-	1568	b: Dense $(\tilde{y})$	-	-	784
Reshape	-	-	7 × 7 × 32	c: Reshape	b	-	28 × 28 × 1
Residual block	[3 × 3] × 1	UpSampling	14 × 14 × 32	Concatenate	[a, c]	-	28 × 28 × 2
Residual block	[3 × 3] × 1	UpSampling	28 × 28 × 32	Conv2D	3 × 3	-	28 × 28 × 32
Conv2D	3 × 3	-	28 × 28 × 1	Residual block	[3 × 3] × 1	AvgPool	14 × 14 × 32
Sigmoid	-	-	28 × 28 × 1	Residual block	[3 × 3] × 1	AvgPool	7 × 7 × 32
				Residual block	[3 × 3] × 2	AvgPool	3 × 3 × 32
				Residual block	[3 × 3] × 2	-	3 × 3 × 32
				Global pooling	Avg	-	32
				Dense	-	-	1

Table 2. Detailed description of CIFAR-10 WGAN-GP architectures.

Generator				**Discriminator**			
	Settings	**Resample**	**Output Shape**		**Settings**	**Resample**	**Output Shape**
Input $(n, \tilde{y})$	-	-	138	a: Input (x)	-	-	32 × 32 × 3
Dense	-	-	2560	b: Dense $(\tilde{y})$	-	-	1024
Reshape	-	-	4 × 4 × 160	c: Reshape	b	-	32 × 32 × 1
Residual block	[3 × 3] × 1	UpSampling	8 × 8 × 160	Concatenate	[a, c]	-	32 × 32 × 4
Residual block	[3 × 3] × 1	UpSampling	16 × 16 × 160	Conv2D	3 × 3	-	32 × 32 × 192
Residual block	[3 × 3] × 1	UpSampling	32 × 32 × 160	Residual block	[3 × 3] × 1	AvgPool	16 × 16 × 192
Conv2D	3 × 3	-	32 × 32 × 3	Residual block	[3 × 3] × 1	AvgPool	8 × 8 × 192
Sigmoid	-	-	32 × 32 × 3	Residual block	[3 × 3] × 1	AvgPool	4 × 4 × 192
				Residual block	[3 × 3] × 1	-	4 × 4 × 192
				Global pooling	Avg	-	192
				Dense	-	-	1

4. Results

4.1. Metrics

Anomaly detection can be formulated as a binary classification problem—deciding whether the test sample x is (negative class) or is not (positive class) an anomaly. In our work, given the classifier C and the trained discriminator D, the anomaly score is given by:

$$score(x) = D(x, C(x)) \tag{11}$$

and then

$$x = \begin{cases} not\ anomaly\ if\ score(x) > \tau \\ anomaly\ if\ score(x) \leq \tau, \end{cases}$$

where τ is a score threshold that needs to be specified. The choice of this value directly impacts the false positives and the false negatives cases. To measure the effectiveness of an anomaly detector in separating normal samples and anomalous samples, we adapted the AUROC and FPR95TPR metrics as those are mainly reported in the literature, e.g., [58–61]. Additionally, we measured the percentage of correctly classified real samples and anomalies for a selected threshold.

4.1.1. AUROC

To evaluate the performance of the anomaly detector, we employed area under the receiver operating curve (AUROC) since it is a threshold-independent evaluation metric. The ROC curve depicts the relationship between the true positive rate ($TPR = TP/(TP + FN)$) and false positive rate ($FPR = FP/(FP + TN)$). Setting a threshold for given FP explains Figure 6. Moreover, the AUROC holds an important statistical property [62] in that it can be interpreted as the probability that a randomly chosen positive sample is assigned a higher score than a randomly chosen negative sample. Thus, the higher the value, the better.

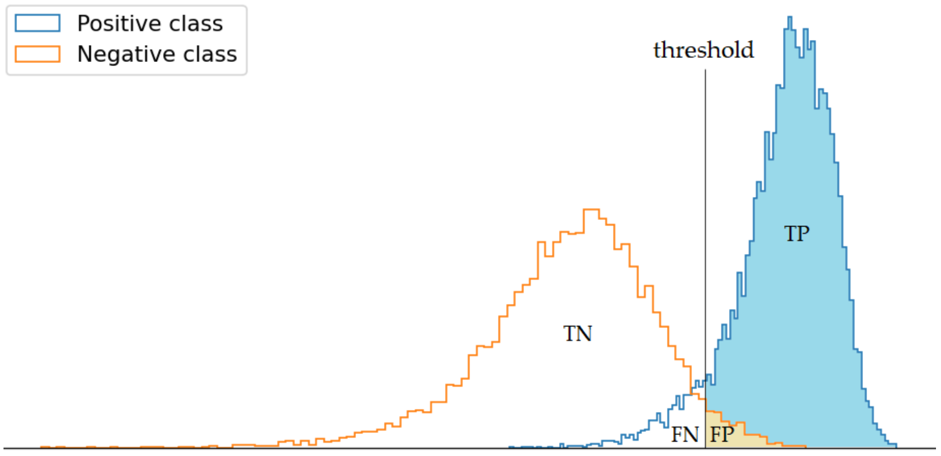


Figure 6. Effect of threshold selection on TPR and FPR.

4.1.2. FPR95TPR Metrics

The metric FPR@TPR can be interpreted as a probability that a negative sample is misclassified as positive when the score threshold is set so that the TPR is as high as @%. Thus, the lower the value, the better.

4.1.3. Accuracy

Accuracy (percentage of correctly classified real samples and anomalies) is obtained for a selected threshold. The threshold was calculated from a thousand randomly sampled real images from the training set and a thousand anomaly samples generated by the generator with random seeds with a norm $\|\boldsymbol{n}\| \in [6, 7]$. The threshold value was chosen so that 95%

and 99% (0.95 and 0.99 in the table heading) of these thousand randomly sampled real images were classified correctly.

4.2. WGAN-GP MNIST Network with a Discriminator Untrained for Anomalies

The WGAN-GP network, according to Figure 4, was trained using Adam optimizer with a learning rate of 2×10^{-4}, decayed linearly to 1×10^{-9} over 1500 epochs with default values for $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The batch size was set to 100, the generator's random seed length was N = 64, and the hyperparameter was $\lambda = 10$. For evaluation, we used the weights from the last epoch. The overall accuracy of the different metrics is shown in Table 3.

Table 3. Anomaly-detection results [%] for the WGAN-GP MNIST with a discriminator untrained for anomalies.

Dataset	AUROC	FPR95TPR	Accuracy (0.95)	Accuracy (0.99)
MNIST	-	-	91.71	97.79
FashionMNIST	63.99	80.14	23.82	15.41
Omniglot	28.89	98.97	1.42	0.58
notMNIST	61.62	70.92	31.68	25.68
Uniform	42.65	100.00	0.03	0.00
Gaussian	32.27	100.00	0.00	0.00
CIFAR-10-bw	47.17	91.71	9.89	6.18
SVHN-bw	32.94	97.67	2.93	1.63
Fakes $\|\boldsymbol{n}\| \in [0,5]$	63.35	78.47	27.75	13.56
Anomalies $\|\boldsymbol{n}\| \in [6,7]$	57.06	85.23	18.56	10.11

Table 3 shows that the WGAN-GP discriminator's anomaly-detection performance is insufficient if it is not trained for anomalies. Figure 5 gives the average class values depending on a seed norm. Figure 7 details ROC curves and anomaly score distributions for MNIST and anomaly databases.

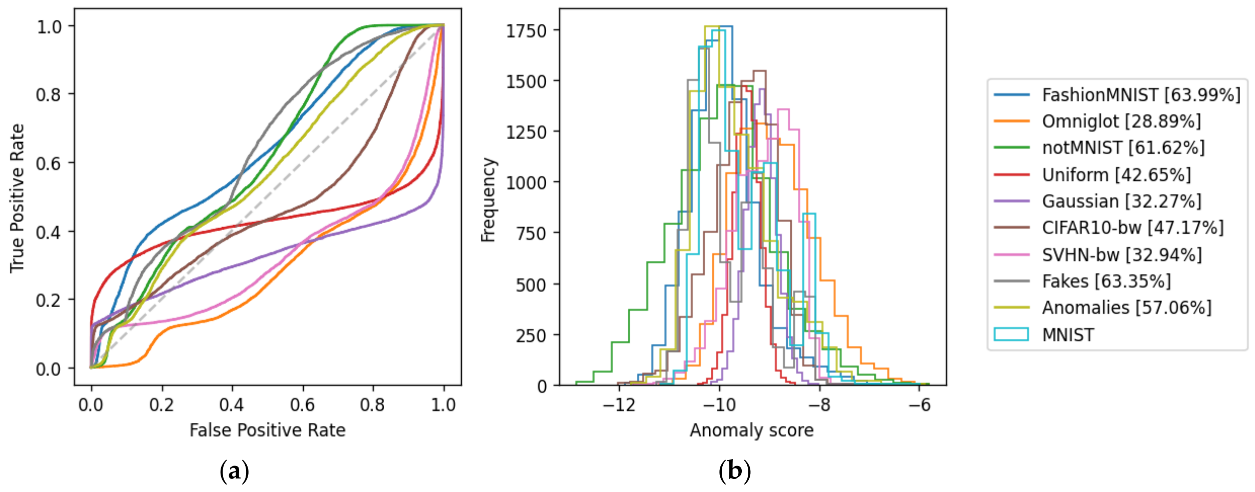


Figure 7. ROC (**a**); and anomaly score distribution (**b**) for the WGAN-GP with a discriminator untrained for anomalies on MNIST.

We obtained a more detailed insight into the test set by combining both outputs (1000 samples in each part of Figure 8). Figure 9 shows improvement when the training for anomalies is applied.

We can see that the classifier better recognizes specific noise properties than the discriminator. It was not only the discriminator score that contributed to correct anomaly detection in this case.

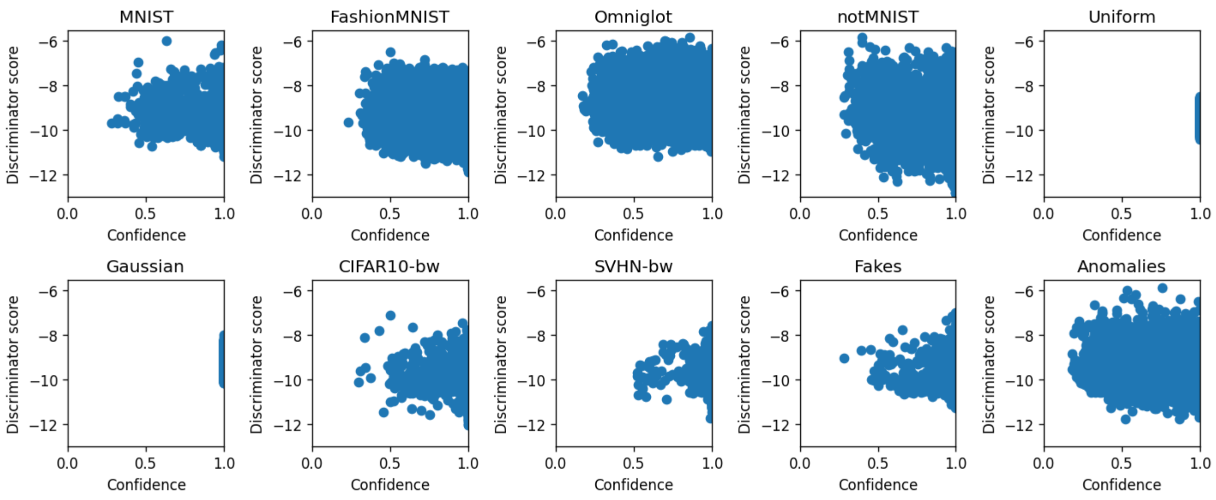


Figure 8. Classifier confidence vs. discriminator score for MNIST and anomaly datasets using the WGAN-GP with a discriminator untrained for anomalies on MNIST.

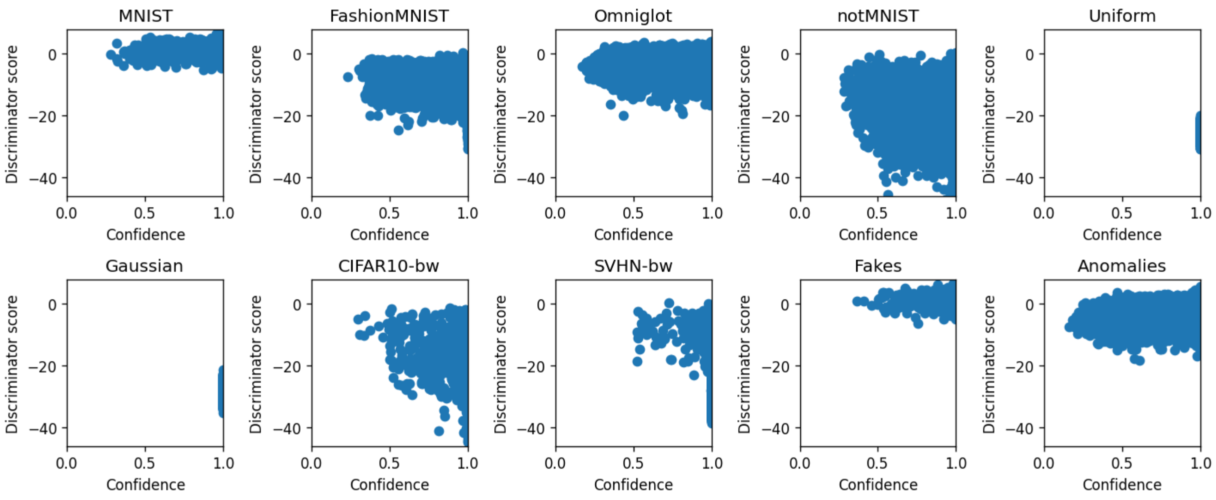


Figure 9. Classifier confidence vs. discriminator score for MNIST and anomaly datasets using the WGAN-GP with a discriminator trained for anomalies on MNIST.

4.3. WGAN-GP CIFAR-10 Network with a Discriminator Untrained for Anomalies

Similar to Table 3, Table 4 also shows the poor performance in anomaly detection by the WGAN-GP discriminator trained for fakes on the CIFAR-10 database.

Table 4. Anomaly-detection results [%] for the WGAN-GP CIFAR-10 with an untrained discriminator for anomalies.

Dataset	AUROC	FPR95TPR	Accuracy (0.95)	Accuracy (0.99)
CIFAR-10	-	-	94.18	98.58
CIFAR-100	56.41	93.35	7.65	1.90
MNIST-resized	24.09	99.03	1.03	0.52
LSUN	51.15	99.93	0.08	0.01
Uniform	4.33	100.00	0.00	0.00
Gaussian	2.92	100.00	0.00	0.00
SVHN	42.31	94.00	6.78	1.39
Fakes $\|\boldsymbol{n}\| \in [0, 5]$	56.70	93.09	7.77	2.31
Anomalies $\|\boldsymbol{n}\| \in [6, 7]$	54.29	96.31	4.02	1.96

We trained the WGAN-GP CIFAR-10 network under the same settings used for the WGAN-GP MNIST network (Section 4.2), with the only difference being the length of the generator's random seed, which was set to N = 128.

4.4. WGAN-GP MNIST Network with a Discriminator Trained for Generated Anomalies

While the discriminator gives poor anomaly detection performance if the WGAN-GP is trained only for a fake generation, we can add an anomaly detector parallel to the discriminator (see Figure 10; both use the same architecture) and train it for anomaly detection. This means that once the generator part of WGAN-GP is trained for fake generation, it can be used for generating fakes (seeds with the norm $\|\boldsymbol{n}\| < r_A = 5$), and anomalies (seeds with the norm $\|\boldsymbol{n}\| \in [5, 6]$).

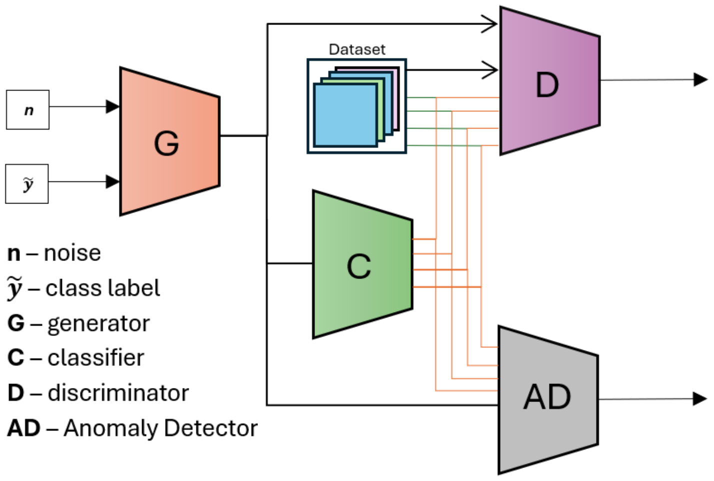


Figure 10. Proposed WGAN-GP architecture for anomaly detection tailored to pattern classifier.

We leveraged this property and trained the anomaly detector for 1200 iterations with the Adam optimizer using a constant learning rate of 1×10^{-3}. In each iteration, we utilized the generator to generate 100 fake and 100 anomalous images (seed norm $\|\boldsymbol{n}\| < 5$ and $\|\boldsymbol{n}\| \in [5, 6]$, respectively). To overcome the class imbalance problem, we used the uniform class label sampling approach and generated ten images from each class. Since we wanted to distinguish between fakes and anomalous images, we formulated this as a binary classification problem; thus, the anomaly detector was trained using binary cross-entropy to classify fake images (class 1) and anomalous ones (class 0). This leads to a high or low logit value, which was later used as an anomaly score.

We assumed that the faithfully generated fakes would augment the training set and can be used instead of the training samples. As representatives of anomalies, we used generated samples in the border seed norm, and we believed that the samples with higher norms were also anomalies (see Figure 3).

Table 5 shows the overall accuracy of the different metrics. Compared with Table 3, we see a substantial improvement in anomaly-detection performance. Figure 5b shows a qualitative improvement in anomaly detection for all classes. Figure 7 shows the ROC curves (a) and anomaly score distribution (b) for the MNIST and anomaly databases.

Fakes obtained the interesting property of a 3.85% (0.79%) misclassification. Compared with 95.67% (99.10%) accuracy in MNIST, the fakes were lying inside the area of training samples. We obtained a more detailed insight into the test set by combining both outputs (1000 samples in each part of Figure 8). Improvement in MNIST sample detection by the anomaly detector improved the decision-making between original and anomaly samples.

We trained eight independent anomaly detectors with random initial weights to obtain better statistical results. We report the average results, with standard deviations in brackets, in Tables 5–10. More details give curves on Figure 11.

Table 5. Anomaly-detection results [%] for the WGAN-GP MNIST with a discriminator trained for generated anomalies. The standard deviation of eight independent trainings is reported in brackets.

Dataset	**AUROC**	**FPR95TPR**	**Accuracy (0.95)**	**Accuracy (0.99)**
MNIST	-	-	95.67 (0.24)	99.10 (0.09)
FashionMNIST	98.55 (2.02)	4.71 (5.19)	94.86 (5.41)	87.84 (8.20)
Omniglot	98.84 (0.46)	5.42 (2.55)	93.48 (2.77)	77.48 (6.58)
notMNIST	99.93 (0.07)	0.24 (0.28)	99.68 (0.36)	98.57 (1.97)
Uniform	100.00 (0.00)	0.00 (0.00)	100.00 (0.00)	100.00 (0.00)
Gaussian	100.00 (0.00)	0.00 (0.00)	100.00 (0.00)	100.00 (0.00)
CIFAR-10-bw	98.03 (2.28)	8.58 (9.73)	90.50 (10.38)	80.95 (17.46)
SVHN-bw	92.33 (7.62)	23.23 (19.28)	75.67 (19.70)	65.26 (22.73)
Fakes $\|\|\boldsymbol{n}\|\| \in [0,5]$	48.48 (1.26)	95.41 (0.21)	3.85 (0.26)	0.79 (0.09)
Anomalies $\|\|\boldsymbol{n}\|\| \in [6,7]$	97.96 (0.40)	9.74 (2.01)	88.90 (1.95)	73.50 (4.44)

Table 6. Anomaly-detection results [%] for the WGAN-GP CIFAR-10 with a discriminator trained for anomalies. The standard deviation of eight independent trainings is reported in brackets.

Dataset	**AUROC**	**FPR95TPR**	**Accuracy (0.95)**	**Accuracy (0.99)**
CIFAR-10	-	-	95.86 (0.69)	98.61 (0.27)
CIFAR-100	55.42 (0.64)	87.04 (0.59)	11.36 (1.14)	5.57 (0.74)
MNIST-resized	99.54 (0.25)	0.00 (0.00)	99.98 (0.02)	94.72 (5.84)
LSUN	96.28 (2.52)	15.22 (10.47)	82.51 (10.57)	71.47 (15.70)
Uniform	100.00 (0.00)	0.00 (0.00)	100.00 (0.00)	100.00 (0.00)
Gaussian	100.00 (0.00)	0.00 (0.00)	100.00 (0.00)	100.00 (0.00)
SVHN	15.70 (2.59)	99.70 (0.26)	0.23 (0.25)	0.08 (0.08)
Fakes $\|\|\boldsymbol{n}\|\| \in [0,5]$	45.90 (0.35)	96.81 (0.26)	2.64 (0.66)	0.70 (0.20)
Anomalies $\|\|\boldsymbol{n}\|\| \in [6,7]$	99.78 (0.04)	0.80 (0.20)	98.97 (0.27)	97.06 (1.07)

Table 7. Overall anomaly-detection results [%] for the WGAN-GP MNIST with a discriminator trained for generated anomalies mixed with specific anomalies (real SVHN samples). Standard deviation of eight independent trainings is reported in brackets.

Dataset	**AUROC**	**FPR95TPR**	**Accuracy (0.95)**	**Accuracy (0.99)**
MNIST	-	-	95.39 (0.24)	99.07 (0.08)
FashionMNIST	99.97 (0.02)	0.06 (0.06)	99.93 (0.06)	99.46 (0.45)
Omniglot	98.12 (0.50)	9.64 (2.92)	89.58 (3.06)	69.17 (5.99)
notMNIST	99.99 (0.01)	0.02 (0.02)	99.97 (0.02)	99.87 (0.08)
Uniform	100.00 (0.00)	0.00 (0.00)	100.00 (0.00)	100.00 (0.00)
Gaussian	100.00 (0.00)	0.00 (0.00)	100.00 (0.00)	100.00 (0.00)
CIFAR-10-bw	100.00 (0.00)	0.00 (0.00)	100.00 (0.00)	100.00 (0.00)
SVHN-bw	100.00 (0.00)	0.00 (0.00)	100.00 (0.00)	100.00 (0.00)
Fakes $\|\|\boldsymbol{n}\|\| \in [0,5]$	51.21 (0.28)	94.64 (0.21)	4.96 (0.20)	1.04 (0.08)
Anomalies $\|\|\boldsymbol{n}\|\| \in [6,7]$	96.61 (0.36)	16.41 (1.70)	82.75 (1.80)	63.25 (2.69)

Table 8. Overall anomaly-detection results [%] for the WGAN-GP CIFAR-10 with a discriminator trained for generated anomalies mixed with specific anomalies (real SVHN samples). Standard deviation of eight independent trainings is reported in brackets.

Dataset	**AUROC**	**FPR95TPR**	**Accuracy (0.95)**	**Accuracy (0.99)**
CIFAR-10	-	-	95.02 (0.72)	98.24 (0.40)
CIFAR-100	60.82 (1.42)	85.82 (1.17)	14.17 (1.84)	6.69 (0.82)
MNIST-resized	99.46 (0.29)	0.16 (0.37)	99.53 (1.23)	96.35 (4.98)
LSUN	95.16 (1.92)	20.00 (7.54)	80.02 (7.66)	67.44 (9.08)
Uniform	100.00 (0.00)	0.00 (0.00)	100.00 (0.00)	100.00 (0.00)
Gaussian	100.00 (0.00)	0.00 (0.00)	100.00 (0.00)	100.00 (0.00)
SVHN	99.52 (0.27)	1.89 (1.18)	98.02 (1.33)	96.42 (2.10)
Fakes $\|\|\boldsymbol{n}\|\| \in [0,5]$	46.95 (0.53)	96.72 (0.18)	3.30 (0.55)	1.03 (0.27)
Anomalies $\|\|\boldsymbol{n}\|\| \in [6,7]$	99.72 (0.06)	1.00 (0.26)	98.98 (0.27)	96.90 (0.95)

Table 9. Anomaly-detection results [%] for the WGAN-GP MNIST with a discriminator trained by real MNIST samples. The standard deviation of eight independent trainings is reported in brackets.

Dataset	AUROC	FPR95TPR	Accuracy (0.95)	Accuracy (0.99)
MNIST	-	-	95.16 (0.33)	98.97 (0.11)
FashionMNIST	99.99 (0.01)	0.01 (0.01)	99.99 (0.01)	99.87 (0.20)
Omniglot	98.14 (0.96)	9.42 (5.57)	90.49 (5.17)	74.10 (12.57)
notMNIST	99.93 (0.13)	0.30 (0.64)	99.71 (0.61)	99.00 (2.00)
Uniform	100.00 (0.00)	0.00 (0.00)	100.00 (0.00)	100.00 (0.00)
Gaussian	100.00 (0.00)	0.00 (0.00)	100.00 (0.00)	100.00 (0.00)
CIFAR-10-bw	100.00 (0.00)	0.00 (0.00)	100.00 (0.00)	99.99 (0.03)
SVHN-bw	100.00 (0.00)	0.00 (0.00)	100.00 (0.00)	99.97 (0.06)
Fakes [0, 5]	52.51 (0.62)	93.47 (0.35)	6.45 (0.42)	1.66 (0.12)
Anomalies [6, 7]	98.08 (0.80)	8.73 (3.70)	90.97 (3.88)	78.63 (6.21)

Table 10. Anomaly-detection results [%] for the WGAN-GP CIFAR-10 with a discriminator trained by real CIFAR-10 samples. The standard deviation of eight independent trainings is reported in brackets.

Dataset	AUROC	FPR95TPR	Accuracy (0.95)	Accuracy (0.99)
CIFAR-10	-	-	95.43 (0.97)	98.71 (0.38)
CIFAR-100	56.17 (0.87)	87.34 (0.78)	11.75 (1.78)	4.97 (1.20)
MNIST-resized	98.23 (1.33)	7.02 (10.54)	90.83 (11.54)	58.75 (30.85)
LSUN	98.89 (0.58)	4.21 (2.32)	95.54 (2.18)	91.38 (3.94)
Uniform	100.00 (0.00)	0.00 (0.00)	100.00 (0.00)	100.0 (0.00)
Gaussian	100.00 (0.00)	0.00 (0.00)	100.00 (0.00)	100.0 (0.00)
SVHN	18.49 (5.15)	99.76 (0.24)	0.18 (0.27)	0.06 (0.07)
Fakes $\|\boldsymbol{n}\| \in [0,5]$	48.57 (1.26)	94.82 (1.01)	4.58 (1.50)	1.33 (0.27)
Anomalies $\|\boldsymbol{n}\| \in [6,7]$	99.88 (0.03)	0.46 (0.10)	99.43 (0.15)	98.29 (0.36)

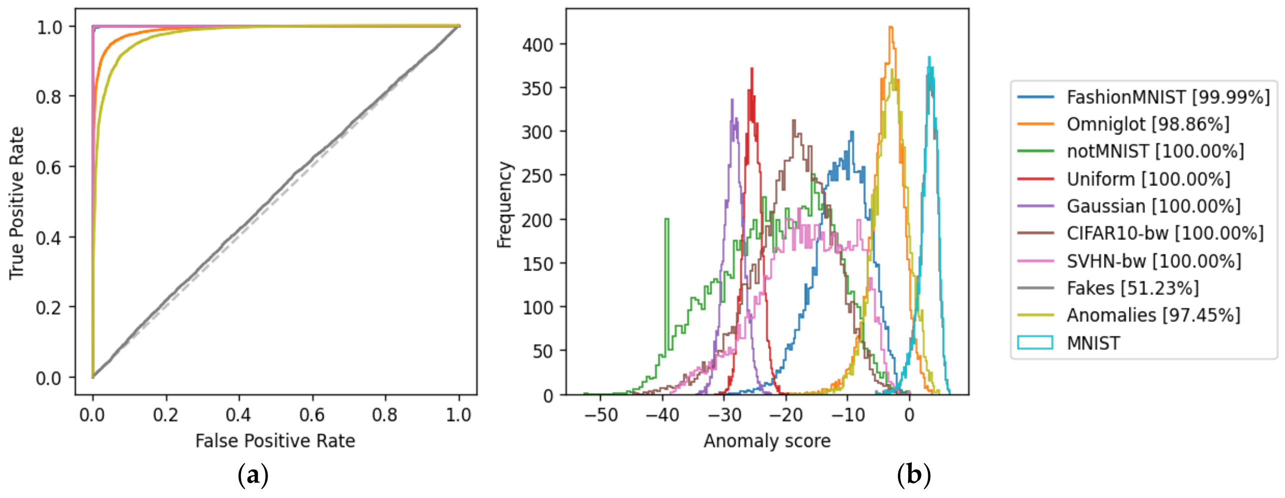
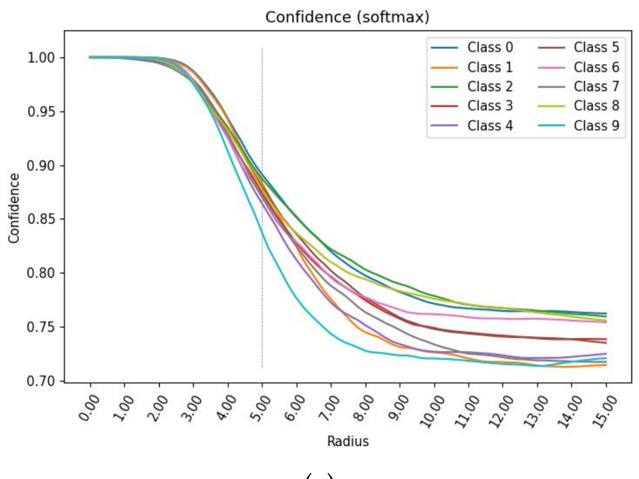


Figure 11. ROC (**a**) and anomaly score distribution (**b**) for the WGAN-GP with a discriminator trained for anomalies on MNIST.

Figure 12 shows the average class values depending on the seed norm.

To visualize classifier behavior under anomaly inputs, we extracted 32 features from the classifier and compressed them by UMAP [63] into 2D. Figure 13 provides a reference of all features within the training set.

We generated five random seeds in class 1 and changed their norm in the range of $\|\boldsymbol{n}\| \in [0,\ 15]$ with a step of 0.5. A visualization of the samples is shown in Figure 14.



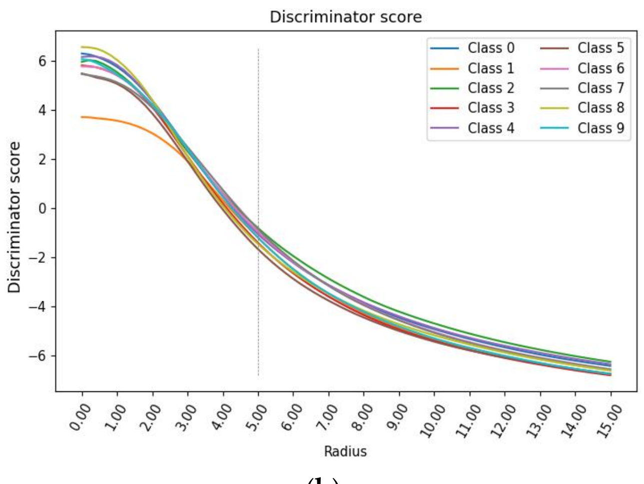


Figure 12. Classifier (**a**) and trained discriminator (**b**) average outputs if the input seed norm (radius) increases. The vertical line gives the border between fakes and anomalies.

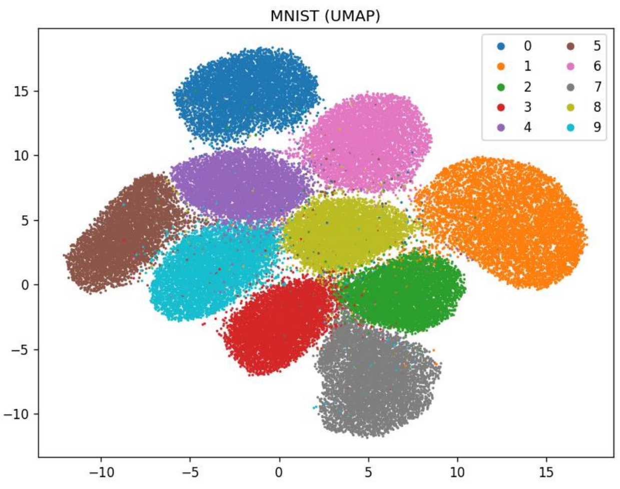


Figure 13. UMAP 2D visualization of the MNIST training set features.

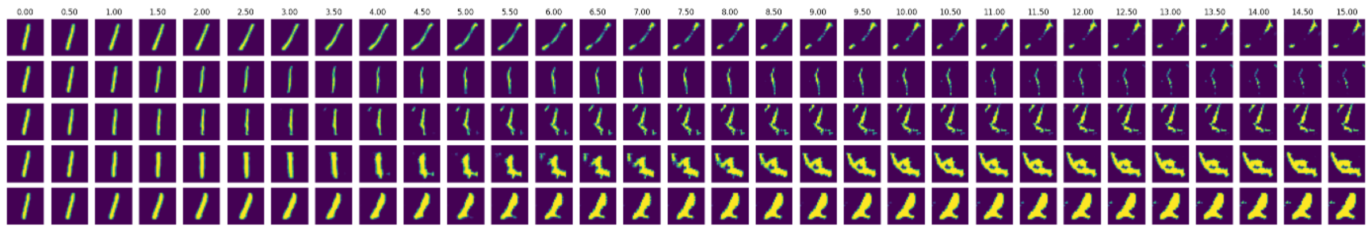


Figure 14. MNIST class 1-based anomalies obtained by the seed norm increase.

Figure 15 shows the corresponding pathways in 2D for each random seed from Figure 14. For $\|\boldsymbol{n}\| = 0$, they have the same root (class centroid), and for $\|\boldsymbol{n}\| = 15$ they finished in different classes (0, 2, 4). Thus, each pathway consists of images generated from the seed of the corresponding norm in increasing order. This figure underlines the importance of anomaly detection according to individual classes.

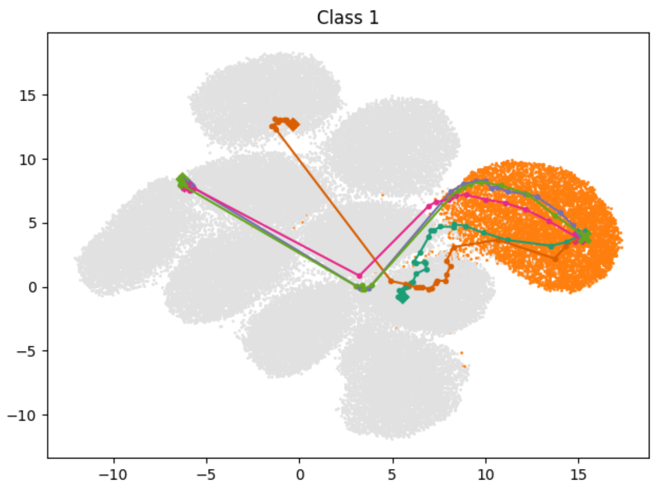


Figure 15. UMAP 2D visualization of the MNIST-based anomaly features generated by increasing the seed norm.

Figure 16 shows the mapping of the notMNIST anomalies (a), "near" anomalies generated by the WGAN-GP generator (with the seed norm $\|\boldsymbol{n}\| \in [6, 7]$) (b), CIFAR-10 anomalies, and Gaussian noise anomalies into 2D compressed feature space.

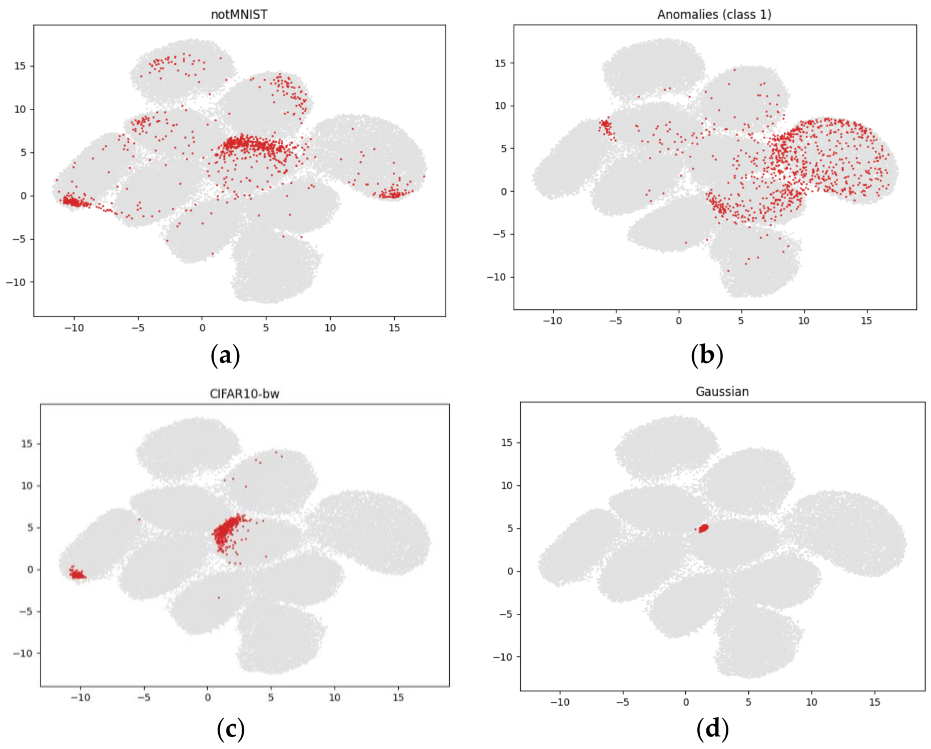


Figure 16. Mapping of the notMNIST anomalies (**a**); class 1 anomalies generated by WGAN-GP generator (**b**); CIFAR-10 (**c**); and Gaussian noise (**d**) into 2D compressed feature space.

4.5. WGAN-GP CIFAR-10 Network with a Discriminator Trained for Generated Anomalies

Although training the discriminator to recognize the generated anomalies improves the anomaly-detection performance, specific anomalies (CIFAR-100, SVHN) remain poorly detected (see Table 6). The SVHN anomalies were even worse than those detected by the WGAN-GP trained on the generated MNIST anomalies. We suspect this is a clever Hans effect because the MNIST and SVHN images have different backgrounds. We used the same approach and hyperparameter settings as in Section 4.4 for the training.

5. Discussion

The hypothesis that a single number can estimate the anomalousness of a sample seems speculative. However, it is no more audacious that a single number can evaluate the similarity of two samples. It turns out that if this number results from a nonlinear transformation, the use of this hypothesis may lead to usable results. To find such a number, we focused on the pattern generation mechanism. We do not consider an anomaly to be a dissimilar pattern but a pattern that is so dissimilar that it is suspected to be generated by a different mechanism [2]. Generative neural networks can generate samples that are like the training set using a generator random seed. Using a normal normalized distribution, we found that random numbers with a relatively narrow probability density distribution of the norm are used to generate fake samples. Therefore, it is natural to assume that for seeds not used in training the generator, the generator will generate samples that do not resemble the training samples. Using the seed norm as a measure of the sample anomaly allowed us to train the anomaly detector to apply smaller seeds for fake generation, and the seeds with the norm slightly exceeded the limit for anomaly sample generation (near anomalies). Experiments have shown that the anomaly detector trained in this way also recognizes anomalies generated by seeds with larger norms or by substantially different mechanisms (anomaly databases). This allows us to solve the curse of dimensionality problem due to orders-of-magnitude higher anomalies than the training samples. However, not all anomalies can be generated in this way. Different training sets and different WGAN-GP architectures can generate different anomalies. This problem is left for further investigation. However, the first results support the hypothesis that the anomaly detector trained to recognize near anomalies from the reference training set will also recognize the anomalies generated on different datasets. Figure 17 shows an average output of the anomaly detector trained for near anomalies of the MNIST training set and anomalies generated by the CIFAR-10 dataset. Most CIFAR-10 real images (90.50%, see Table 5) and generated CIFAR-10 fakes and CIFAR-10 near anomalies (89.75% and 98.52%, respectively) are recognized also as MNIST anomalies.

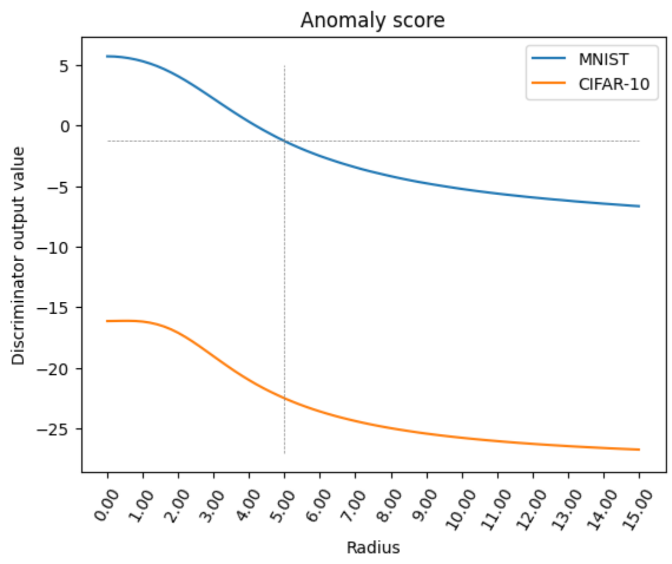


Figure 17. Average output for WGAN-GP MNIST anomaly detector if CIFAR-10-generated anomalies tested the anomaly detector. The dotted line gives the level for decision between the normal and anomaly samples.

Sometimes, it is up to the user's preference as what is considered an anomaly. If MNIST contains handwritten digits, are the house numbers in the Street View House Numbers (SVHN) database (some handwritten) anomalies? If we consider the answer to be yes, we can add this database to the anomalies generated by the fake generator and use them in anomaly-detector training. In this way, we can strengthen the training of the anomaly detector with arbitrary images that we want to be considered as anomalies (see Tables 9 and 10). As expected, it will improve SVHN anomaly detection and other "near" anomalies like CIFAR-100.

On the other hand, real samples from the training set can also support anomaly-detector training. By replacing fakes with real samples, we support correct decisions for real data (see Tables 9 and 10). There were no significant changes for real samples, but the anomaly-detection accuracy was influenced in both directions. When using an MNIST detector, anomaly detection of CIFAR-10-bw and SVHN-bw samples improved, while Omniglot detection was degraded. Similarly, with a CIFAR-10-based detector, anomaly detection of LSUN samples improved, while the detection of MNIST-resized samples was degraded. Hence, supplementing the training with real data leads to only slightly better anomaly resolution than when generated fakes alone are used to train the anomaly detector.

Tailoring for the given classifier means taking recognized classes into account. As Figure 15 shows, the anomalies regarding class 1 can be accepted as the normal samples of other classes. Therefore, the post hoc anomaly detection tailored for the classifier is more precise. Table 11 compares the obtained results with those from the literature [64]. Comparing results with those presented in [64], we can see that CAVGA gives the best accuracy in normal MNIST samples, but anomaly detector AD outperforms other methods in terms of anomaly detection (bold indicates the best results). The proposed anomaly detector provides a conservative strategy that rejects some normal samples but is very sensitive to anomalous ones. This strategy was set by a threshold of 95% TPR in Figure 11b. Setting a more aggressive strategy with an MNIST threshold of 99% TPR, AD gives accuracy = 87.84% for FashionMNIST and accuracy = 80.95% for CIFAR-10 anomalies.

Table 11. MNIST accuracy and anomaly-detection AUROC results [%].

Anomaly Dataset	AD (0.95)	γ-VAEg	LSA	OCGAN	ULSLM	Caps Net$_{PP}$	Caps Net$_{RE}$	AnoGAN	ADGAN	CAVGA
MNIST accuracy	95.7	98.2	97.5	97.5	94.9	97.7	92.5	93.7	96.8	**98.6**
FashionMNIST	**94.96**	87.3	87.6	-	-	76.5	67.9	-	-	88.5
CIFAR-10-bw	**90.5**	71.7	64.1	65.6	73.6	61.2	53.1	61.2	63.4	73.7

Our proposed method was tested on visual data (images), as we believe this will allow the user to gain more insights into the "anomaly generation" process and evaluate it intuitively. The main advantage of the proposed approach is that it can be tailored to practically any type of classifier. The classifier only provides the predicted class for the anomaly detector. The anomaly-detection part is solely conducted by an additional anomaly detector that acts as a discriminator between non-anomalous and anomalous examples. There is also no restriction on the data type or specific database selection. The only constraint our approach requires is that the tested example must be processed independently by the given classifier, and the given AD, twice.

6. Conclusions

Experiments support the hypothesis that:

- WGAN-GP generator seeds with the norm inside the hypercube (hypersphere with the Chebyshev norm) generate fakes, and seeds with the norm outside the hypercube generate anomalies;
- An anomaly detector trained for anomalies with seeds at the border of the hypercube can detect the anomalies generated by seeds anywhere outside the hypercube. This partially solves the anomaly/real samples dimensionality curse problem because we can obtain a huge number of anomalies from the fake generator trained on the given real sample set. Of course, we cannot obtain anomalies generated from the fake generator trained on different training sets or generators with different architectures.

Still, the first results support the hypothesis of recognition anomalies obtained in these ways.

Anomaly detectors trained by WGAN-GP-generated samples (fakes/anomalies) also performed well for databases different to the training database and applied as anomaly databases. Using specific databases in anomaly-detector training to support positive or negative decisions can help with specific user needs.

Author Contributions: Conceptualization, M.K. (Martin Kontšek) and L'.K.; methodology, M.K. (Martin Klimo); software, L'.K.; validation, L'.K., O.Š. and M.K. (Martin Kontšek); resources, L'.K.; data curation, L'.K.; writing—original draft preparation, M.K. (Martin Klimo), O.Š., L'.K. and M.K. (Martin Kontšek); writing—review and editing, M.K. (Martin Kontšek); visualization, L'.K.; supervision, M.K. (Martin Klimo); project administration, M.K. (Martin Kontšek); funding acquisition, M.K. (Martin Kontšek). All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the University of Žilina, Žilina, Slovakia grant scheme number 1/2021/FRI/MVP and by the Slovak Grant Agency VEGA project Fast Reroute, No. 1/0316/24.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author, L'.K., upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Aggarwal, C.C. *An Introduction to Outlier Analysis*; Springer: Berlin/Heidelberg, Germany, 2017.
2. Hawkins, D.M. *Identification of Outliers*, 1st ed.; Springer: Dordrecht, The Netherlands, 1980. [CrossRef]
3. Lu, S.; Zhang, W.; Guo, J.; Liu, H.; Li, H.; Wang, N. PatchCL-AE: Anomaly detection for medical images using patch-wise contrastive learning-based auto-encoder. *Comput. Med Imaging Graph.* **2024**, *114*, 102366. [CrossRef]
4. Shvetsova, N.; Bakker, B.; Fedulova, I.; Schulz, H.; Dylov, D.V. Anomaly Detection in Medical Imaging with Deep Perceptual Autoencoders. *IEEE Access* **2021**, *9*, 118571–118583. [CrossRef]
5. Nakao, T.; Hanaoka, S.; Nomura, Y.; Murata, M.; Takenaga, T.; Miki, S.; Watadani, T.; Yoshikawa, T.; Hayashi, N.; Abe, O. Unsupervised Deep Anomaly Detection in Chest Radiographs. *J. Digit. Imaging* **2021**, *34*, 418–427. [CrossRef] [PubMed]
6. Kim, M.; Moon, K.-R.; Lee, B.-D. Unsupervised anomaly detection for posteroanterior chest X-rays using multiresolution patch-based self-supervised learning. *Sci. Rep.* **2023**, *13*, 3415. [CrossRef]
7. Gouda, W.; Tahir, S.; Alanazi, S.; Almufareh, M.; Alwakid, G. Unsupervised Outlier Detection in IOT Using Deep VAE. *Sensors* **2022**, *22*, 6617. [CrossRef] [PubMed]
8. Abdallah, M.; Le Khac, N.A.; Jahromi, H.; Jurcut, A.D. A hybrid CNN-LSTM based approach for anomaly detection systems in SDNs. In Proceedings of the 16th International Conference on Availability, Reliability and Security, Vienna, Austria, 17–20 August 2021; pp. 1–7.
9. Crépey, S.; Lehdili, N.; Madhar, N.; Thomas, M. Anomaly Detection in Financial Time Series by Principal Component Analysis and Neural Networks. *Algorithms* **2022**, *15*, 385. [CrossRef]
10. Okechukwu, O.P.; Okechukwu, G.N.; Mbonu, C.E.; Paul, R.U. A Deep Learning Model for Detecting Anomalies in the Banking Sector Using a Feed-Forward Neural Network. *Int. J. Sci. Eng. Res.* **2023**, *14*, 322–327.
11. Karthikeyan, T.; Govindarajan, M.; Vijayakumar, V. An effective fraud detection using competitive swarm optimization based deep neural network. *Meas. Sens.* **2023**, *27*, 100793. [CrossRef]
12. Tosunoğlu, N.; Abacı, H.; Ateş, G.; Akkaya, N.S. Artificial neural network analysis of the day of the week anomaly in cryptocurrencies. *Financ. Innov.* **2023**, *9*, 88. [CrossRef]
13. Darabi, N.; Tayebati, S.; Ravi, S.; Tulabandhula, T.; Trivedi, A.R. STARNet: Sensor Trustworthiness and Anomaly Recognition via Approximated Likelihood Regret for Robust Edge Autonomy. *arXiv* **2023**, arXiv:2309.11006.
14. Zhang, J.; Chen, X.; Jandaghi, E.; Zeng, W.; Zhou, M.; Yuan, C. Dynamics Learning-Based Fault Isolation for A Soft Trunk Robot. In Proceedings of the 2023 American Control Conference (ACC), San Diego, CA, USA, 31 May–2 June 2023; pp. 40–45. [CrossRef]
15. Jandaghi, E.; Chen, X.; Yuan, C. Motion Dynamics Modeling and Fault Detection of a Soft Trunk Robot. In Proceedings of the 2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Seattle, WA, USA, 28–30 June 2023; pp. 1324–1329. [CrossRef]
16. Ramachandra, B.; Jones, M. Street scene: A new dataset and evaluation protocol for video anomaly detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 2569–2578.

17. Chang, Y.; Tu, Z.; Xie, W.; Yuan, J. Clustering Driven Deep Autoencoder for Video Anomaly Detection. In *Computer Vision, Proceedings of the 16th European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 329–345.
18. Sun, X.; Chen, J.; Shen, X.; Li, H. Transformer with Spatio-Temporal Representation for Video Anomaly Detection. In *Structural, Syntactic, and Statistical Pattern Recognition*; Krzyzak, A., Suen, C.Y., Torsello, A., Nobile, N., Eds.; Springer International Publishing: Cham, Switzerland, 2022; pp. 213–222.
19. Gao, Y.; Yin, X.; He, Z.; Wang, X. A deep learning process anomaly detection approach with representative latent features for low discriminative and insufficient abnormal data. *Comput. Ind. Eng.* **2023**, *176*, 108936. [CrossRef]
20. Liu, W.; Yan, L.; Ma, N.; Wang, G.; Ma, X.; Liu, P.; Tang, R. Unsupervised Deep Anomaly Detection for Industrial Multivariate Time Series Data. *Appl. Sci.* **2024**, *14*, 774. [CrossRef]
21. Jeong, J.; Zou, Y.; Kim, T.; Zhang, D.; Ravichandran, A.; Dabeer, O. WinCLIP: Zero-/Few-Shot Anomaly Classification and Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023.
22. You, Z.; Cui, L.; Shen, Y.; Yang, K.; Lu, X.; Zheng, Y.; Le, X. A Unified Model for Multi-class Anomaly Detection. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 4571–4584.
23. Liu, F.T.; Ting, K.M.; Zhou, Z.-H. Isolation forest. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; pp. 413–422.
24. Xu, H.; Pang, G.; Wang, Y.; Wang, Y. Deep isolation forest for anomaly detection. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 12591–12604. [CrossRef]
25. Muandet, K.; Schölkopf, B. One-Class Support Measure Machines for Group Anomaly Detection. *arXiv* **2013**, arXiv:1303.0309.
26. Yang, K.; Kpotufe, S.; Feamster, N. An Efficient One-Class SVM for Anomaly Detection in the Internet of Things. *arXiv* **2021**, arXiv:2104.11146.
27. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
28. Chen, Z.; Yeo, C.K.; Lee, B.S.; Lau, C.T. Autoencoder-based network anomaly detection. In Proceedings of the 2018 Wireless Telecommunications Symposium (WTS), Phoenix, AZ, USA, 17–20 April 2018; pp. 1–5. [CrossRef]
29. Torabi, H.; Mirtaheri, S.L.; Greco, S. Practical autoencoder based anomaly detection by using vector reconstruction error. *Cybersecurity* **2023**, *6*, 1. [CrossRef]
30. Guha, D.; Chatterjee, R.; Sikdar, B. Anomaly Detection Using LSTM-Based Variational Autoencoder in Unsupervised Data in Power Grid. *IEEE Syst. J.* **2023**, *17*, 4313–4323. [CrossRef]
31. Wei, Y.; Jang-Jaccard, J.; Xu, W.; Sabrina, F.; Camtepe, S.; Boulic, M. LSTM-autoencoder-based anomaly detection for indoor air quality time-series data. *IEEE Sens. J.* **2023**, *23*, 3787–3800. [CrossRef]
32. Li, Z.; Sun, Y.; Yang, L.; Zhao, Z.; Chen, X. Unsupervised Machine Anomaly Detection Using Autoencoder and Temporal Convolutional Network. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 3525813. [CrossRef]
33. He, Z.; Chen, Y.; Zhang, D.; Abdulaal, M. Vehicle Anomaly Detection by Attention-Enhanced Temporal Convolutional Network. In Proceedings of the 2023 IEEE 6th International Conference on Industrial Cyber-Physical Systems, ICPS 2023, Wuhan, China, 8–11 May 2023. [CrossRef]
34. Kim, J.; Kang, H.; Kang, P. Time-series anomaly detection with stacked Transformer representations and 1D convolutional network. *Eng. Appl. Artif. Intell.* **2023**, *120*, 105964. [CrossRef]
35. Goodfellow, I. Nips 2016 tutorial: Generative adversarial networks. *arXiv* **2016**, arXiv:1701.00160.
36. Brownlee, J. *Generative Adversarial Networks with Python: Deep Learning Generative Models for Image Synthesis and Image Translation*; Machine Learning Mastery: San Juan, Puerto Rico, 2019.
37. Aggarwal, A.; Mittal, M.; Battineni, G. Generative adversarial network: An overview of theory and applications. *Int. J. Inf. Manag. Data Insights* **2021**, *1*, 100004. [CrossRef]
38. Sabuhi, M.; Zhou, M.; Bezemer, C.-P.; Musilek, P. Applications of Generative Adversarial Networks in Anomaly Detection: A Systematic Literature Review. *IEEE Access* **2021**, *9*, 161003–161029. [CrossRef]
39. Xia, X.; Pan, X.; Li, N.; He, X.; Ma, L.; Zhang, X.; Ding, N. GAN-based anomaly detection: A review. *Neurocomputing* **2022**, *493*, 497–535. [CrossRef]
40. Pang, G.; Shen, C.; Cao, L.; Van Den Hengel, A. Deep Learning for Anomaly Detection: A Review. *ACM Comput. Surv.* **2021**, *54*, 1–38. [CrossRef]
41. Di Mattia, F.; Galeone, P.; De Simoni, M.; Ghelfi, E. A survey on gans for anomaly detection. *arxiv* **2019**, arXiv:1906.11632.
42. Li, H.; Li, Y. Anomaly detection methods based on GAN: A survey. *Appl. Intell.* **2023**, *53*, 8209–8231. [CrossRef]
43. Schlegl, T.; Seeböck, P.; Waldstein, S.M.; Schmidt-Erfurth, U.; Langs, G. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *Information Processing in Medical Imaging*; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Cham, Switzerland, 2017. [CrossRef]
44. Zenati, H.; Foo, C.S.; Lecouat, B.; Manek, G.; Chandrasekhar, V.R. Efficient GAN-Based Anomaly Detection. *arXiv* **2019**, arXiv:1802.06222.

45. Akçay, S.; Atapour-Abarghouei, A.; Breckon, T.P. GANomaly: Semi-supervised Anomaly Detection via Adversarial Training. In *Computer Vision–ACCV 2018*; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Cham, The Netherlands, 2019. [CrossRef]
46. Adari, S.K.; Alla, S. *Beginning Anomaly Detection Using Python-Based Deep Learning*, 2nd ed.; Apress Publishers: Berkeley, CA, USA, 2024. [CrossRef]
47. Kopčan, J.; Klimo, M.; Škvarek, O. Do Neural Networks Recognize Patterns as well as Students? In Proceedings of the 2022 20th International Conference on Emerging ELearning Technologies and Applications (ICETA), Stary Smokovec, Slovakia, 20–21 October 2022; pp. 338–343. [CrossRef]
48. LeCun, Y.; Cortes, C. The Mnist Database of Handwritten Digits. 2005. Available online: https://api.semanticscholar.org/CorpusID:60282629 (accessed on 7 March 2024).
49. Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. 2009. Available online: https://api.semanticscholar.org/CorpusID:18268744 (accessed on 7 March 2024).
50. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv* **2017**, arXiv:1708.07747.
51. Lake, B.M.; Salakhutdinov, R.; Tenenbaum, J.B. Tenenbaum, Human-level concept learning through probabilistic program induction. *Science* **2015**, *350*, 1332–1338. [CrossRef]
52. Bulatov, Y. Notmnist Dataset, Google (Books/OCR), Tech. Rep. 2011. Available online: https://yaroslavvb.blogspot.com/2011/09/notmnist-dataset.html (accessed on 7 March 2024).
53. Yuval, N. Reading digits in natural images with unsupervised feature learning. In Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning, Granada, Spain, 16–17 December 2011.
54. Yu, F.; Seff, A.; Zhang, Y.; Song, S.; Funkhouser, T.; Xiao, J. LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop. *arXiv* **2016**, arXiv:1506.03365.
55. Zagoruyko, S.; Komodakis, N. Wide residual networks. *arXiv* **2016**, arXiv:1605.07146.
56. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A. Improved Training of Wasserstein GANs. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5767–5777.
57. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein GAN. *arXiv* **2017**, arXiv:1701.07875.
58. Liang, S.; Li, Y.; Srikant, R. Enhancing the Reliability of Out-of-distribution Image Detection in Neural Networks. *arXiv* **2020**, arXiv:1706.02690.
59. Winkens, J.; Bunel, R.; Roy, A.G.; Stanforth, R.; Natarajan, V.; Ledsam, J.R.; MacWilliams, P.; Kohli, P.; Karthikesalingam, A.; Kohl, S.; et al. Contrastive training for improved out-of-distribution detection. *arXiv* **2020**, arXiv:2007.05566.
60. Liu, W.; Wang, X.; Owens, J.D.; Li, Y. Energy-based out-of-distribution Detection. *Adv. Neural Inf. Process. Syst.* **2021**, *33*, 21464–21475.
61. Masana, M.; Ruiz, I.; Serrat, J.; van de Weijer, J.; Lopez, A.M. Metric Learning for Novelty and Anomaly Detection. *arXiv* **2018**, arXiv:1808.05492.
62. Fawcett, T. An introduction to ROC analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874. [CrossRef]
63. McInnes, L.; Healy, J.; Saul, N.; Großberger, L. UMAP: Uniform Manifold Approximation and Projection. *J. Open Source Softw.* **2018**, *3*, 861. [CrossRef]
64. Venkataramanan, S.; Peng, K.C.; Singh, R.V.; Mahalanobis, A. Attention Guided Anomaly Localization in Images. In *Computer Vision, Proceedings of the 16th European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020.