

Received March 12, 2022, accepted April 14, 2022, date of publication April 22, 2022, date of current version April 29, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3169594

ALGAN: Anomaly Detection by Generating Pseudo Anomalous Data via Latent Variables

HIRONORI MURASE¹ AND KENJI FUKUMIZU²

¹Department of Statistical Science, The Graduate University for Advanced Studies, SOKENDAI, Tachikawa 190-8562, Japan

²The Institute of Statistical Mathematics, Tachikawa 190-8562, Japan

Corresponding author: Hironori Murase (hn.murase@gmail.com)

This paper is based on results obtained from a project, JPNP16007, commissioned by the New Energy and Industrial Technology Development Organization (NEDO). This work was supported in part by The Graduate University for Advanced Studies, SOKENDAI.

ABSTRACT In many anomaly detection tasks, where anomalous data rarely appear and are difficult to collect, training using only normal data is important. Although it is possible to manually create anomalous data using prior knowledge, they may be subject to user bias. In this paper, we propose an Anomalous Latent variable Generative Adversarial Network (ALGAN) in which the GAN generator produces pseudo-anomalous data as well as fake-normal data, whereas the discriminator is trained to distinguish between normal and pseudo-anomalous data. This differs from the standard GAN discriminator, which specializes in classifying two similar classes. The training dataset contains only normal data; the latent variables are introduced in anomalous states and are input into the generator to produce diverse pseudo-anomalous data. We compared the performance of ALGAN with other existing methods on the MVTec-AD, Magnetic Tile Defects, and COIL-100 datasets. The experimental results showed that ALGAN exhibited an AUROC comparable to those of state-of-the-art methods while achieving a much faster prediction time.

INDEX TERMS Anomaly detection, computer vision, machine learning, deep learning, generative adversarial networks.

I. INTRODUCTION

Anomaly detection refers to the technique of distinguishing between unexpected and normal data and is closely related to outlier detection and novelty detection [1]. Practical examples include fraud detection to identify unauthorized access [2], medical diagnosis to discover lesion sites from medical images [3], surveillance to find suspicious behavior in real-time videos [4], and optical inspection for detecting defects in industrial products [5]. A common feature of these applications is the discovery of undesirable data.

The difficulty with anomaly detection is that, in many cases, anomalous data are rarely observed and are of a wide variety; hence, the learning of anomaly detection models suffers from the difficulty of imbalanced or one-class classification. Various methods have been proposed to address this issue, such as creating a dataset containing new anomalous data not included in conventional datasets [6], verifying classification performance for anomalous data that are rarely observed [2], and developing a new one-class classifier for complex data such as image and sequence data [7].

The associate editor coordinating the review of this manuscript and approving it for publication was Sabah Mohammed^{1b}.

The augmentation of anomalous data from different sources has also been proposed (e.g., out-of-distribution data [8], [9]); however, in such cases, undesired biases may be introduced.

This paper discusses anomaly detection using only normal training data. Because preparing a large amount of anomalous data is difficult, training without anomalous data is a preferable approach. In this line of research, a wide variety of methods have already been proposed based on traditional machine learning and statistical techniques, such as one-class classification, likelihood, nearest neighbors, and clustering. See Chandola *et al.* [1] for a comprehensive survey of traditional approaches.

Recently, deep learning methods such as representation learning and deep generative models have been successfully applied to anomaly detection without using anomalous data for training. See [10]–[12] for a comprehensive survey of the deep learning approaches. Taking advantage of the effective representation of deep learning, the features obtained by a pre-trained model, such as VGG [13] and ResNet [14], are applied to unsupervised anomaly detection [15]. Because Generative Adversarial Networks (GANs, [16]) can learn a generative model of normal data through a discrimination task, they have been combined for anomaly detection in

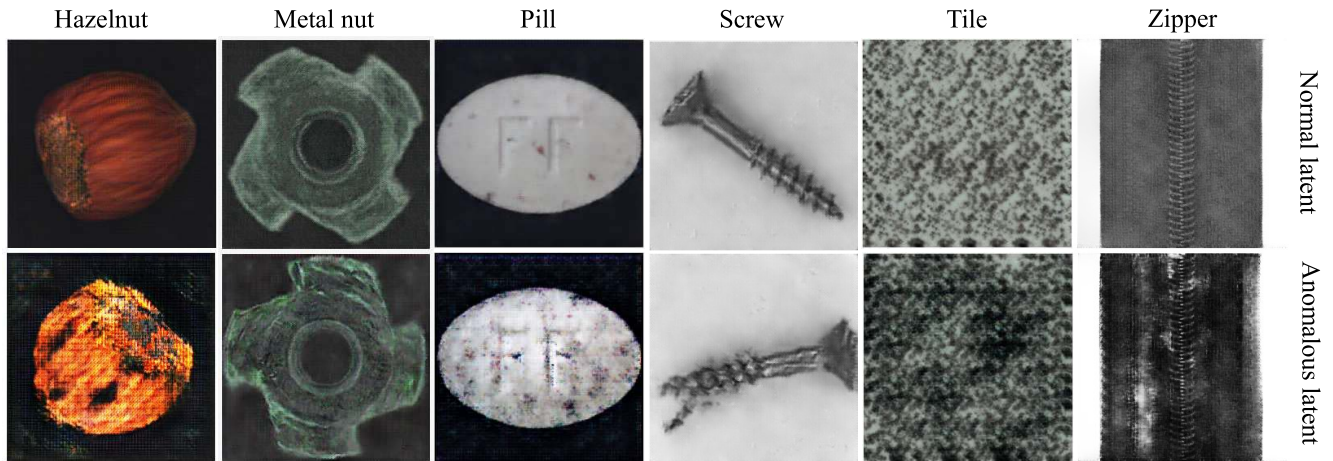


FIGURE 1. Images generated by ALGAN-image. Top row: Fake-normal data from normal latent variables $N(0, I)$. Bottom row: Fake-anomalous data from anomalous latent variables $N(0, \sigma^2 I)$, where $\sigma = 4$.

various ways [3], [17]–[20]. More details on anomaly detection without using anomalous data for training are presented in Section II.

In this paper, we propose a method that improves anomaly detection performance by generating pseudo-anomalous data from only normal training data using GANs. Unlike the standard usage of GANs, the generator used in the proposed method provides pseudo-anomalous data as well as fake-normal data, by introducing anomalous states in the latent variable. We call this model Anomalous Latent GAN (ALGAN). Note that the discriminator of a standard GAN is not necessarily suitable for distinguishing between normal and anomalous data. It is trained to discriminate between real and fake data such that in successful learning, the two classes are almost similar. By contrast, when training is successful, the discriminator of ALGAN distinguishes between the group of real-normal data and the group of fake-normal and pseudo-anomalous data.

Some state-of-the-art anomaly detection methods specialize in product appearance inspection from images. For example, DifferNet [21] concatenates feature vectors from three different resolution images and uses them to train a model; furthermore, PatchCore [22] uses the hierarchical patches of features to achieve effective and fast performance. However, these methods presume a pre-trained model and cannot be trained directly from the image data. By contrast, the proposed ALGAN can be trained using both images and features.

To apply anomaly detection in real-world applications, it is important to consider computational resources [10]. Normal training data can be collected efficiently; however, training time increases with the amount of training data. In addition, real-time prediction is significant in the real world [4], where the data generation speed has increased [23]. Reducing computational costs will contribute to the expansion of the application [24]. The proposed ALGAN achieves fast training and prediction times.

The contributions of this study are as follows:

- We propose a novel method for generating pseudo-anomalous data: adding pseudo-anomalous data to GAN training improves the anomaly detection performance of the discriminator.
- The proposed method can be applied to both images and feature vectors. Experimental results show that it achieves a state-of-the-art performance compared with image-based methods and comparable ability to feature-based methods.
- The proposed ALGAN achieved a remarkably fast prediction time, 10.4 to 54.6 times faster than other image-based methods on the benchmark MVTEC-AD dataset.

The remainder of this paper is organized as follows. We compare our proposed method to relevant anomaly detection methods in Section II. Section III presents preliminaries on standard GANs, details of the proposed method, and intuition for pseudo-anomalous data. Section IV provides the implementation details, datasets, and evaluation method. Section V examines the anomaly detection performances on various datasets and presents the computation time, stability, and ablation study results. Section VI discusses the advantages of this study and possible future work. Section VII presents concluding remarks.

II. RELATED WORK

A. GAN-BASED ANOMALY DETECTION

A vast body of literature already exists on generative models for anomaly detection; therefore, we present only representative studies that use GANs. Anomaly detection methods using GANs can be divided into two categories: those using reconstruction errors and those using one-class classifiers. In early studies [3], [17], the latent variable corresponding to given test data was estimated, and the reconstruction error of the image generated from the latent variable was used as the anomaly score. GANomaly [18] uses two encoders

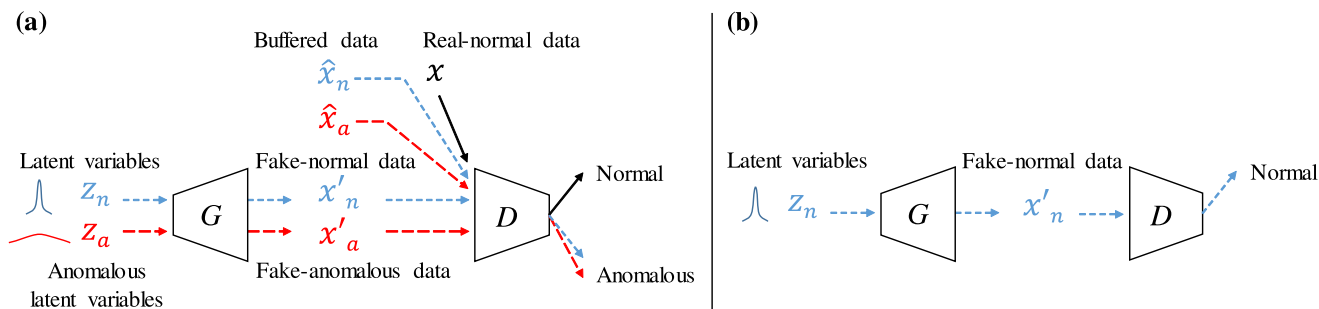


FIGURE 2. Overview of ALGAN training. (a) Procedure for training the discriminator D with real-normal data x (black), latent variables $z_n \sim N(0, I)$ (blue), anomalous latent variables $z_a \sim N(0, \sigma^2 I)$ (red), fake-normal data x'_n (blue), fake-anomalous data x'_a (red), buffered fake-anomalous data \hat{x}_a (red), and buffered fake-normal data \hat{x}_n (blue). (b) Procedure for training the generator G .

that estimate the latent variables of the input and generated images and detects anomalies using the reconstruction error between two latent variables. Skip-GANomaly [19] is an improvement that evaluates the anomaly score using the features extracted from the middle layer of the discriminator. Because a generator with an encoder–decoder structure trained only on normal data may not properly reconstruct anomalous data, ALOCC [20] exploits the one-class classification of the reconstructed images. However, these methods fail to detect anomalies when data reconstruction is successful. By contrast, the proposed method does not depend on the reconstruction error and discriminates directly.

B. ANOMALY DETECTION WITH PRE-TRAINED MODELS

Remarkable performance has been demonstrated in a recent work [25] in which feature representations were exploited from pre-trained models on the ImageNet dataset [26]. The method in [27] extracts the hierarchical features of normal data from a pre-trained EfficientNet [28], fits them with a multivariate normal distribution, and uses the Mahalanobis distance as the anomaly score. SPADE [29] extracts hierarchical features, concatenates them, and performs anomaly detection at the image and fine-grained pixel levels using the k-nearest neighbors. PaDiM [30] separates hierarchical features into patches and stores the mean and covariance in a memory bank to measure the Mahalanobis distance. Patch-Core [22] subsamples patches of hierarchical features to achieve a high performance and fast prediction. Differ-Net [21] uses a flow-based model [31], which typically is computation-intensive, and utilizes pre-trained model features [32] that reduce data dimensionality and computational costs for likelihood-based anomaly detection. Most of these methods use traditional anomaly detection techniques that require the use of features, and images cannot be used directly for training. By contrast, the proposed method can use both types of training data.

C. PSEUDO-ANOMALOUS DATA

Some methods consider the generation of pseudo-anomalous data. Data immaturely generated during the training process of GANs have been used as pseudo-anomalous

data for training [33]–[35]. In CutPaste [36], patches of random sizes and angles are cut out from an image and randomly pasted onto the image. The classifier is trained either from scratch or fine-tuned using normal and pseudo-anomalous data. The feature representation by the classifier is then used to calculate the anomaly score based on the Gaussian density assumption. The proposed method generates pseudo-anomalous data by introducing anomalous states into latent variables other than using data generated by an immature generator. Thus, it is less biased than techniques that generate pseudo-anomalous data using prior knowledge.

III. PROPOSED METHOD

A. OVERVIEW

Fig. 2 illustrates the training procedure of our proposed ALGAN. The generator is trained in the same manner as in standard GANs. Two additional data types are employed to train the discriminator. One of the data types is generated from the anomalous latent variable, and the other is a buffer of data generated during the training process. The buffer size is twice as large as the training data, and in each epoch, a portion of the old buffer is replaced with newly generated data.

B. GENERATIVE ADVERSARIAL NETWORKS

GANs [16] replace distribution modeling with a discrimination problem: the generator $G(z; \theta)$ maps latent variables z to the data space, and the discriminator $D(x; \phi)$ distinguishes between real data x and the generated samples $x' = G(z)$. The discriminator outputs the probability of the input, and real data and generated samples are labeled with probability 1 and 0, respectively. The discriminator is trained to maximize the log-likelihoods $\log(D(x))$ and $\log(1 - D(G(z)))$. Conversely, the generator is trained to minimize $\log(1 - D(G(z)))$ to fool the discriminator. Both network objectives are given by the following equations:

$$\max_D \left(\mathbb{E}_{x \sim p_{data}(x)} [\log(D(x))] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \right), \tag{1}$$

$$\min_G \left(\mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \right), \tag{2}$$

where p_{data} and p_z denote the distributions of real data and latent variables, respectively.

When the discriminator can no longer distinguish between real data and generated samples, the generator approximately realizes a sampler from the real data distribution. The objective function of the GANs is obtained by combining (1) and (2) as follows:

$$\min_G \max_D \left(\mathbb{E}_{x \sim p_{data}(x)} [\log(D(x))] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \right). \quad (3)$$

Given the optimal generator, the discriminator can distinguish between two similar classes. Thus, it is nontrivial to determine whether such a discriminator is suitable for identifying real and anomalous data. When GANs are trained on a dataset $\{x\}$ that includes only normal data, the discriminator has poor discrimination performance on anomalous data [3]. This suggests that the discrimination boundary of GANs is not specialized in the one-class classification of anomaly detection.

C. PSEUDO-ANOMALOUS DATA

We introduce two types of pseudo-anomalous data for training. They are in addition to the x and $x'_n = G(z_n)$ used in the standard GANs described in Section III-B, where x , x'_n , and z_n are defined as real-normal data, fake-normal data, and latent variables from $N(0, I)$, respectively.

The first type of pseudo-anomalous data is called *fake-anomalous data*. ALGAN utilizes the anomalous latent variables $z_a \sim N(0, \sigma^2 I)$ with a larger variance ($\sigma > 1$) to generate fake-anomalous data $x'_a = G(z_a)$. See Fig. 1 for examples of fake-anomalous data. The fake-anomalous images are slightly degraded compared with the fake-normal images.

The other type of pseudo-anomalous data is called *buffered data* $\hat{x} = \{\hat{x}_n, \hat{x}_a\}$, which are defined as generated samples during the early stage of the training process. These are expected to differ from the real-normal data. During training, the fake samples $x' = \{x'_n, x'_a\}$ are stored and used as buffered data.

D. TRAINING METHODOLOGY

The discriminator is trained to maximize log-likelihoods $\log D(x)$, $\log(1 - D(x'))$, and $\log(1 - D(\hat{x}))$ to distinguish real-normal data x from other data. Conversely, the generator is trained to generate fake-normal data from z_n , and minimize log-likelihood $\log(1 - D(G(z_n)))$ to fool the discriminator. Both network objectives are given by the following equations:

$$\max_D \left(\mathbb{E}_x [\log(D(x))] + \xi \mathbb{E}_{z_n} [\log(1 - D(x'))] + (1 - \xi) \mathbb{E}_{z_n} [\log(1 - D(\hat{x}))] \right), \quad (4)$$

$$\min_G \left(\mathbb{E}_{z_n} [\log(1 - D(G(z)))], \right) \quad (5)$$

where ξ is the ratio of generated data to buffered data. Let the ratio of fake-normal data to fake-anomalous data be α ; then,

Algorithm 1 Training Algorithm of ALGAN

Notation: Number of batches, m ; latent variables for D , z_d ; latent variables for G , z_g .

Hyperparameters: Training epochs (e), update frequency of latent variables (n_z), ratio of normal and anomalous latent variables (α), standard deviation of anomalous latent (σ), and number of updates for D (n_{dis}).

```

1: for  $i = 1, \dots, e$  do
2:   if  $i \bmod n_z = 0$  then
3:     Sample  $z_d \sim \alpha N(0, I)$  and  $(1 - \alpha)N(0, \sigma^2 I)$ 
4:     Sample  $z_g \sim N(0, I)$ 
5:   end if
6:   for  $j = 1, \dots, m$  do
7:     Sample  $x \sim p_{data}$ 
8:     for  $k = 1, \dots, n_{dis}$  do
9:        $x' \leftarrow G_\theta(z_d)$ 
10:      if  $i = 1$  then
11:         $Loss_D \leftarrow D_\phi(x) + D_\phi(x')$ 
12:      else
13:        Sample buffered data  $\hat{x} \sim \text{Buffer}$ 
14:         $Loss_D \leftarrow D_\phi(x) + D_\phi(x') + D_\phi(\hat{x})$ 
15:      end if
16:       $\phi \leftarrow \text{Adam}(Loss_D, \phi)$ 
17:    end for
18:    Buffer  $\leftarrow x'$ 
19:     $Loss_G \leftarrow D_\phi(G_\theta(z_g))$ 
20:     $\theta \leftarrow \text{Adam}(Loss_G, \theta)$ 
21:  end for
22: end for
23: return  $D_\phi, G_\theta$ 

```

the objective of the discriminator is given by the following equation:

$$\max_D \left(\mathbb{E}_x [\log(D(x))] + \alpha \{ \xi \mathbb{E}_{z_n} [\log(1 - D(x'_n))] + (1 - \xi) \mathbb{E}_{\hat{x}_n} [\log(1 - D(\hat{x}_n))] \} + (1 - \alpha) \{ \xi \mathbb{E}_{z_a} [\log(1 - D(x'_a))] + (1 - \xi) \mathbb{E}_{\hat{x}_a} [\log(1 - D(\hat{x}_a))] \} \right). \quad (6)$$

Thus, the discriminator of ALGAN learns the discrimination boundary between real-normal data and the other types of data. The objective function of ALGAN is obtained by combining (5) and (6) as follows:

$$\min_G \max_D \left(\mathbb{E}_x [\log(D(x))] + \alpha \{ \xi \mathbb{E}_{z_n} [\log(1 - D(G(z_n)))] + (1 - \xi) \mathbb{E}_{\hat{x}_n} [\log(1 - D(\hat{x}_n))] \} + (1 - \alpha) \{ \xi \mathbb{E}_{z_a} [\log(1 - D(x'_a))] + (1 - \xi) \mathbb{E}_{\hat{x}_a} [\log(1 - D(\hat{x}_a))] \} \right). \quad (7)$$

The proposed method follows an adversarial training procedure (Fig. 2). It provides a discrimination boundary not only for the real-normal and fake-normal data but also for the real-normal and pseudo-anomalous data, the latter of

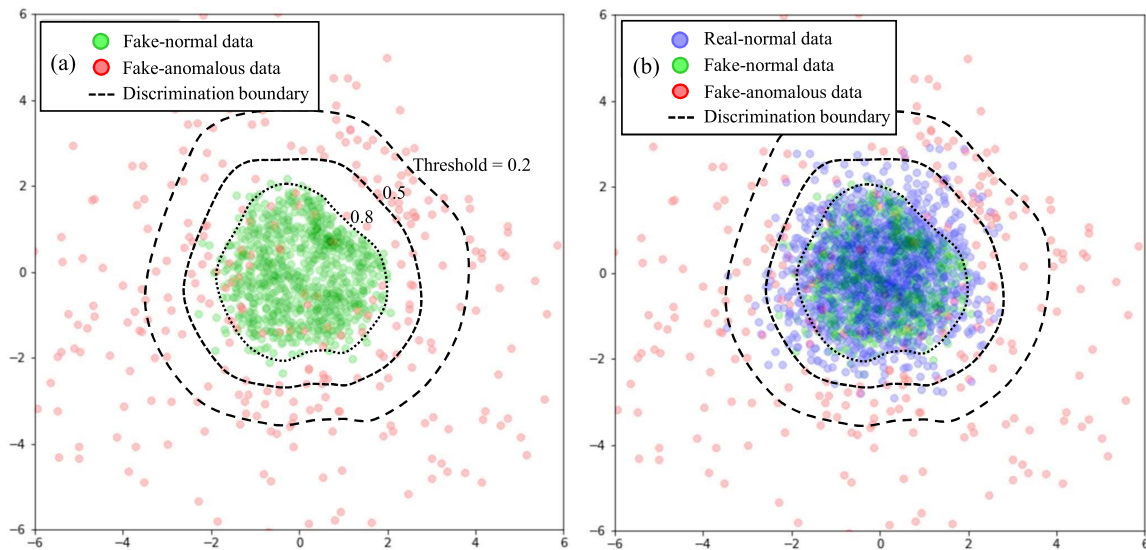


FIGURE 3. Intuitive interpretation of fake-anomalous data in the toy problem: fake-anomalous data are added to training GANs that map 100-dimensional latent variables to 2-dimensional normal distribution. (a) Fake-anomalous data x'_a (red) generated by $\sigma = 4$ are distributed to surround the fake-normal data x'_n (green). (b) Real-normal data x (blue) are overlaid on the left figure. From the figure, it can be seen that the generator has been trained successfully and has generated fake-normal data that approximate the real-normal data distribution. Thus, the discriminator cannot distinguish between the real-normal and fake-normal data. By contrast, the fake-anomalous data can be properly distinguished by adjusting the discrimination threshold because they are distributed outside.

which has a broader support of the distribution. As the training progresses, the generator produces samples that resemble real-normal data, and the discriminator cannot distinguish between real-normal and fake-normal data. The pseudo-anomalous data are clearly different from the real-normal data; therefore, the discrimination boundary of the discriminator is used to classify them.

The pseudo-code for training is presented in Algorithm 1. A feature of ALGAN training is the method for updating the parameter ϕ of the discriminator. In line 3 of the pseudo-code, normal and anomalous latent variables are sampled, and in line 9, fake-normal and fake-anomalous data are produced from the generator. In line 11, loss is calculated by identifying both types of data, and in line 16, the parameter ϕ of the discriminator is updated. Fake-normal and fake-anomalous data are buffered in line 18. As there are no buffered data in the first epoch, the conditional branch for $i = 1$ is required in line 10. The loss of buffered data is calculated in addition to fake-normal and fake-anomalous data in line 14 after the branch. The parameter θ of the generator is updated by only fake-normal data from normal latent variables in lines 4, 19, and 20.

Fig. 3 and the accompanying movie provide an intuitive understanding of the training for anomaly detection using fake-anomalous data. The generator is trained to produce fake-normal data that approximate the distribution of real-normal data. Given anomalous latent variables, the generator produces fake-anomalous data with a large variance. The discriminator is trained to distinguish between real-normal data and the other types of data. Even after real-normal and fake-normal data become indistinguishable, fake-anomalous

data can be identified because the discrimination boundary is laid between real-normal and fake-anomalous data.

E. ANOMALY DETECTION

In ALGAN, the real and fake labels are assigned probability 1 and 0, respectively. The real label corresponds to normal data; thus, the anomaly detection rule is given by the following:

$$\text{ALGAN}(x) = \begin{cases} \text{Normal,} & \text{if } D(x) > \text{threshold,} \\ \text{Anomalous,} & \text{otherwise.} \end{cases} \quad (8)$$

Because the significance of false positives or false negatives depends on their application, the threshold is chosen by the user.

IV. EXPERIMENTS

An advantage of ALGAN is that it can be used for both the images and features extracted from a pre-trained model, whereas some state-of-the-art methods for visual inspection depend on features [21], [22], [29], [30]. To demonstrate this advantage experimentally, we used two different types of implementations. We call them ALGAN-image and ALGAN-feature, and compare them with the relevant methods.

A. IMPLEMENTATION DETAILS

1) NETWORK ARCHITECTURE AND HYPERPARAMETER
ALGAN-image employs an architecture similar to that of DCGAN [37]. The generator and discriminator use seven transposed convolutional and convolutional layers, respectively.

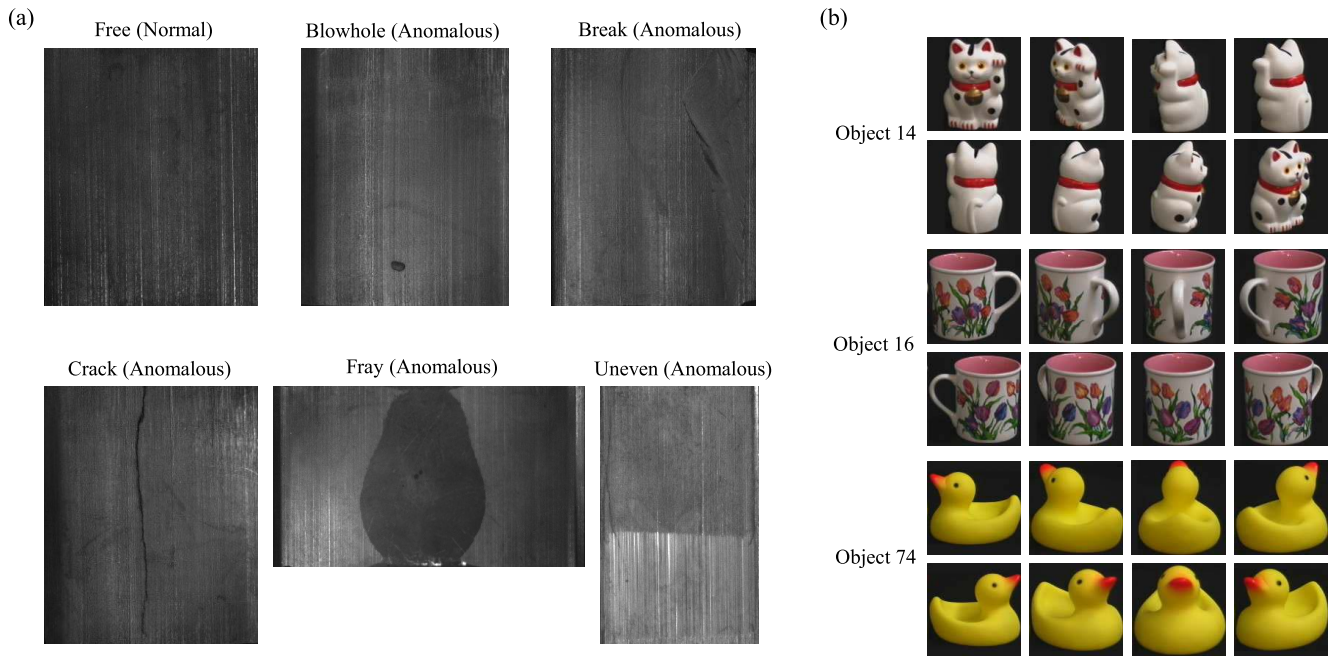


FIGURE 4. Overview of the experimental datasets. (a) Magnetic Tile Defects dataset (MTD); (b) Columbia University Image Library dataset (COIL-100).

For ALGAN-feature, WideResNet101 [38] is applied to the feature extractor to obtain 2048-dimensional vectors from the last block with global average pooling. Both the generator and discriminator have three fully connected layers.

In both architectures, the generator uses batch normalization [39] and the ReLU activation function, whereas batch normalization is removed from the output layer. The discriminator employs spectral normalization [40] and the Leaky-ReLU activation function.

The networks were optimized using Adam [41] with momentum $\beta_1 = 0$ and $\beta_2 = 0.9$, and the learning rates of the generator and discriminator were set to 2×10^{-4} and 1×10^{-4} , respectively. The latent variable z had 100 dimensions, and the standard deviation of the anomalous latent variable used $\sigma = 4$. The parameters that determine the ratio of the pseudo-anomalous data were set to $\alpha = 0.75$ and $\xi = 0.75$. The parameters of Algorithm 1 were set as $n_z = 2$ and $n_{dis} = 2$. The buffer holds twice the batch size, half of which is randomly replaced by newly generated data.

The comparison methods were implemented using a DCGAN-like architecture similar to ALGAN-image. In GANomaly [18] and Skip-GANomaly [19], the dimensions of the latent variables were set to 100 and 512, respectively. The image resolution used for each method was 256×256 . All the models were trained using a batch size of 16.

2) SOFTWARE AND HARDWARE

All the models were implemented using Python 3.8.8 and PyTorch 1.8.1 on Ubuntu 20.04 LTS. An AMD EPYC 7542 32-core processor with 512 GB memory and an NVIDIA A100-SXM4 40 GB GPU were used for the computations.

B. DATASETS

For performance evaluation, we used three different datasets.

1) MVTec-AD DATASET

This dataset, designed for visual inspection, consists of 5,354 images, comprising five texture and 10 object categories [42]. The training set contains 3,629 defect-free (normal) images, and the test set contains 467 defect-free (normal) images and 1,258 defective (anomalous) images.

2) MAGNETIC TILE DEFECTS DATASET (MTD)

This dataset consists of grayscale images with different aspect ratios, including 952 defect-free (normal) images and 392 images containing five defect types (anomalous) [43] (Fig. 4).

3) COIL-100 DATASET

This dataset contains 100 different object categories and 7,200 images [44]. Each object category has 72 images rotated every 5° (Fig. 4).

4) SPLITTING INTO TRAINING AND VALIDATION SETS

Because the training sets of MVTec-AD and MTD contain no defective (anomalous) images, 50% of their respective test sets was used as the validation set, which was used as the stopping rule.

In MTD, the test set is not provided separately from the training data; therefore, 50%, 25%, and 25% of the defect-free (normal) images were used for training, validation, and testing, respectively.

For COIL-100, 10 categories were selected for normal data and the remaining 90 categories for anomalous data. We used 60%, 20%, and 20% of the normal data for training, validation, and testing, respectively, and 50% of the anomalous data were used for validation and 50% for testing.

5) PRE-PROCESSING AND DATA AUGMENTATION

In MVTEC-AD using ALGAN-image, images were resized to 256×256 . To highlight defects, two images were concatenated with the original image and used as an input image: one was created with max pooling and the other with average pooling in the channel axis dimension. For the texture categories, vertical and horizontal flips were applied (however, only horizontal flip was applied for wood). In the object categories, vertical flip, horizontal flip, and random rotation were applied to Bottle and Hazelnut. Horizontal flip was applied to Toothbrush, Transistor, and Zipper. Toothbrush was converted into grayscale. Random rotation was applied to Metal nut and Screw. Cable, Capsule, and Pill were only resized.

In MVTEC-AD using ALGAN-feature, the texture categories were resized to 224×224 . Furthermore, the object categories were resized to 256×256 , and then center-cropped to 224×224 . Following Rudolph *et al.*'s procedure [21], we applied 24 and 64 different angular rotations during training and prediction, respectively, but using a single-resolution image.

The images in MTD and COIL-100 were resized to 256×256 for ALGAN-image and 224×224 for ALGAN-feature. For ALGAN-image, a horizontal flip was applied, and for ALGAN-feature, Rudolph *et al.*'s procedure [21] was applied for both training and prediction.

C. EVALUATION METHOD AND METRIC

Accounting for the randomness of the dataset split, we performed 10 experiments with different seed values for each dataset. Performance was evaluated in terms of the area under the receiver operating characteristic (AUROC) curve, which is obtained by moving the classification threshold for the ratio of true-positive and false-positive rates. ALGAN-image was trained on 512 epochs and ALGAN-feature on 192 epochs. The performances were validated every eight epochs on the validation set, and the model that showed the best AUROC was saved. The best model was evaluated on the test set after training.

V. RESULTS

A. ANOMALY DETECTION ON MVTEC-AD

1) TRAINING WITH IMAGE DATA

The results on the test data with the model with the best AUROC for validation (Section IV-C) are listed in Table 1. ALGAN-image significantly outperformed other state-of-the-art image-based methods, such as GANomaly, Skip-GANomaly, and ALOCC. ALGAN-image showed an

TABLE 1. Results obtained on MVTEC-AD. ALGAN-image is compared with methods trained on image data. Top row: mean AUROC. Bottom row: standard deviation. We report the results of 10 experiments using each method. The best performance for each category is indicated in boldface.

	GANomaly	Skip-GANomaly	ALOCC	ALGAN-image
Carpet	0.803 0.070	0.829 0.056	0.736 0.054	0.846 0.041
Grid	0.924 0.088	0.816 0.079	0.847 0.087	0.938 0.057
Leather	0.796 0.079	0.787 0.095	0.768 0.042	0.920 0.041
Tile	0.852 0.034	0.941 0.041	0.648 0.071	0.914 0.034
Wood	0.939 0.029	0.969 0.020	0.849 0.062	0.972 0.022
Bottle	0.704 0.063	0.674 0.096	0.801 0.075	0.948 0.024
Cable	0.686 0.057	0.636 0.064	0.686 0.053	0.877 0.036
Capsule	0.717 0.075	0.691 0.068	0.707 0.073	0.805 0.083
Hazelnut	0.765 0.058	0.955 0.036	0.697 0.111	0.836 0.047
Metal nut	0.698 0.064	0.553 0.105	0.771 0.066	0.811 0.060
Pill	0.772 0.050	0.787 0.083	0.659 0.043	0.819 0.053
Screw	0.567 0.357	1.000 0.001	0.938 0.076	0.811 0.090
Toothbrush	0.774 0.098	0.808 0.084	0.778 0.102	0.933 0.048
Transistor	0.783 0.067	0.769 0.094	0.745 0.085	0.865 0.074
Zipper	0.693 0.052	0.675 0.070	0.656 0.079	0.879 0.025
mean	0.765	0.793	0.752	0.878

average accuracy of more than 10% compared with the others and attained the best accuracy for 13 out of the 15 categories.

Our method uses the discriminator to distinguish between normal and anomalous directly from the images. In the Hazelnut and Screw categories, the rotation angle of the object is different in each image. Thus, small changes in image details caused by slight defects may be buried by large changes in the image caused by rotation. Consequently, the anomalous data detection performance could not be better than that of the comparison methods in these categories.

Because the comparison methods use reconstruction error, they fail to detect anomalous data if the reconstruction is successful. By contrast, our proposed method, ALGAN-image, uses the discriminator to classify the data and does not suffer from detection errors due to reconstruction.

TABLE 2. Results obtained on MVTEC-AD. ALGAN-feature is compared with methods learned from features of pre-trained models. For DifferNet and ALGAN-feature, we report the mean AUROC results of 10 experiments. The best performance for each category is indicated in boldface.

	SPADE	Mah.AD	PaDiM	CutPaste	PatchCore-25	GANomaly	DifferNet	ALGAN-feature
Texture	-	0.978	0.990	0.962	0.991	0.775	0.915	0.925
Object	-	0.950	0.972	0.955	0.992	0.755	0.934	0.905
ALL	0.855	0.958	0.979	0.961	0.991	0.761	0.927	0.911

TABLE 3. Results obtained on Magnetic Tile Defects. For training with the image data, we report the mean AUROC results of 10 experiments. Numbers in boldface indicate the best performance.

AUROC	Training with image data			Training with pre-trained features			
	GANomaly	Skip-GANomaly	ALGAN-image	PatchCore-10	GANomaly-feat.	DifferNet	ALGAN-feature
	0.683	0.504	0.956	0.979	0.766	0.977	0.824

TABLE 4. Results obtained on COIL-100. Top row: mean AUROC. Bottom row: standard deviation. We report the results of 10 experiments using each method. The best performance is indicated in boldface.

	DifferNet	PatchCore-1	ALGAN-image	ALGAN-feature
AUROC	0.999	1.000	0.999	1.000
	0.003	0.000	0.001	0.000

2) TRAINING WITH PRE-TRAINED FEATURES

The results for the feature-based methods are listed in Table 2. The results for DifferNet are the means of 10 different test datasets (see Section IV-C). The results for GANomaly were obtained from [21] and the other methods from their respective papers [22], [27], [29], [30], [36]. ALGAN-feature achieved results comparable to those of DifferNet, whereas PatchCore-25, PaDiM, and CutPaste achieved better performance. Note, however, that the PatchCore, PaDiM, and CutPaste methods are strongly specialized for pre-trained features.

ALGAN-feature and DifferNet use only a single feature from the last layer of the pre-trained model; better performing methods use features from multiple layers. Using features from multiple layers increases the computational costs owing to their high dimensionality. For example, PatchCore [22] divides high-dimensional features into patches and subsamples them to select useful patches, which results in much higher computational costs. The prediction time is also longer than that of ALGAN-feature (see Section V-C). Furthermore, DifferNet [21] concatenates features from three different image resolutions, whereas our proposed method, ALGAN-feature, achieves comparable performance with only one resolution.

B. ANOMALY DETECTION ON OTHER DATASETS

The results obtained on MTD and COIL-100 are listed in Tables 3 and 4, respectively. These are the means of 10 trials with the same hyperparameters as those used for MVTEC-AD. ALGAN-image achieved state-of-the-art performance

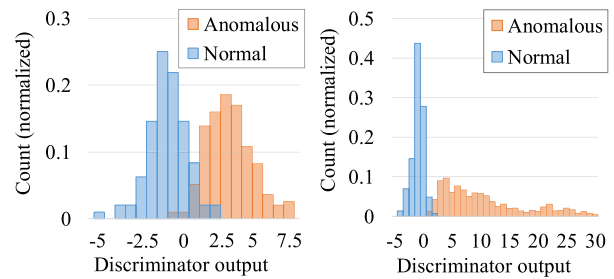


FIGURE 5. Histograms of raw output values of the discriminator before input to the sigmoid function in ALGAN-image. Left: Magnetic Tile Defects. Right: COIL-100. The sign is reversed so that the horizontal axis represents the anomaly score.

on MTD after training with image data and achieved comparable results to that of PatchCore. On MTD, ALGAN-feature performed worse than ALGAN-image. This may be because the features useful for anomaly detection could not be extracted from the last block of WideResNet-101. Because the features from the deep block in ResNet are biased towards ImageNet [22], the features from the shallow block should be used to identify the more abstract features required for MTD. On the COIL-100 benchmark, all the methods performed almost perfectly.

Fig. 5 shows histograms of the raw output values of the discriminator before they are input into the sigmoid function of ALGAN-image. The distributions of normal and anomalous data are significantly separated. In COIL-100, normal data have a peaky distribution with fewer variations, whereas the distribution of anomalous data exhibits a long tail, reflecting the large variations of anomalous data.

C. PREDICTION AND TRAINING TIMES

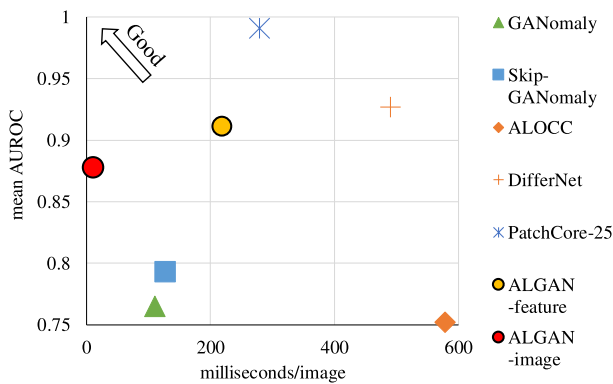
Table 5 compares the prediction times of the models for MVTEC-AD. We can see that ALGAN-image achieved a significantly faster prediction time (10.6 ms), which is 10.4 to 54.6 times faster than those of the other image-based methods. ALGAN-feature is 1.3 to 2.2 times faster than the other methods trained on the feature. Fig. 6 depicts the

TABLE 5. Mean prediction times per single test image for all categories on MVTec-AD. Boldface indicates the best result.

	GANomaly	Skip- GANomaly	ALOCC	DifferNet	PatchCore -25	ALGAN -feature	ALGAN -image
milliseconds/image	109.8	126.3	577.8	489.5	278.6	218.9	10.6

TABLE 6. Mean training times for all categories on MVTec-AD. Boldface indicates the best result.

	GANomaly	Skip- GANomaly	ALOCC	DifferNet	PatchCore -25	ALGAN -feature	ALGAN -image
Training Epochs	512	512	512	192	-	192	512
Training Time (min)	23	47	401	55	98	41	22

**FIGURE 6.** Mean AUROC vs. prediction time for each method on MVTec-AD.

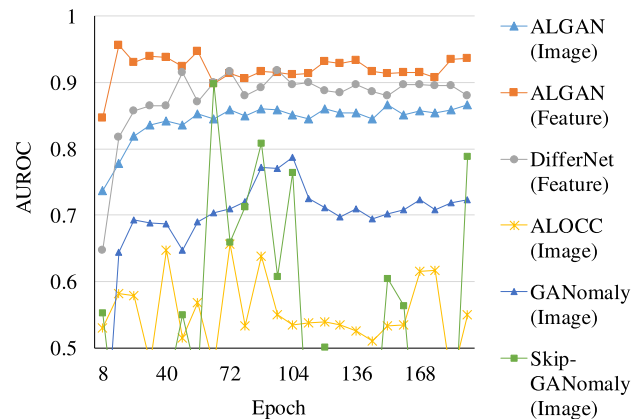
prediction time and AUROC performance of the selected methods. ALGAN-image is the fastest and has a high AUROC, but it is not the highest. ALGAN-feature is faster than the other feature-based methods while maintaining a competitive AUROC. ALGAN and PatchCore exhibit a trade-off between performance and speed. Considering the fast prediction of ALGAN-image, it can be applied to expensive tasks such as real-time prediction with a large number of bounding boxes obtained from object detection [45].

Table 6 compares the training times obtained on MVTec-AD. ALGAN-image is the fastest among the compared methods, with the default number of epochs described in Section IV-C. Because there is no official implementation of PatchCore, we did not use the Faiss library [46],¹ which is used in the original study, but applied our own implementation.

D. TRAINING STABILITY

Zaheer *et al.* reported that GAN-based anomaly detection exhibited unstable validation results during the training process [34]. Upon validation, our proposed method exhibited stable results in terms of AUROC (Fig. 7).

¹Our implementation uses the scikit-learn [47] library for core-set selection [48] and random projection [49]. We confirmed that our implementation produces an AUROC similar to the results in [22].

**FIGURE 7.** Validation results plotted every 8 epochs during the training of the transistor category. ALGAN-image and ALGAN-feature exhibit stable AUROCs compared with other GAN-based methods.**TABLE 7.** Ablation study results obtained on MVTec-AD with ALGAN-image. Each experiment was conducted 10 times. The left side of the mean AUROC column lists the mean and the right side lists the standard deviation. The best performance is indicated in boldface.

Buffered Data	Anomalous Latent	Mean AUROC	
		0.643	0.097
✓		0.761	0.102
	✓	0.779	0.104
✓	✓	0.878	0.049

E. ABLATION STUDY

Ablation studies were performed to verify the effect of the two types of pseudo-anomalous data on performance. In ALGAN, buffered data and the data generated by anomalous latent variables were used as pseudo-anomalous data. The checkmark in Table 7 indicates the type(s) used. Although either of the two can improve performance, using both works best and reduces variance. These results validate that the proposed pseudo-anomalous data are useful for improving anomaly detection performance.

F. HYPERPARAMETER STUDY

We also studied the impact of hyperparameters on anomaly detection performance. ALGAN employs the following

TABLE 8. Impact of σ changes on performance. Top row: mean AUROC. Bottom row: standard deviation. Boldface indicates the best result.

σ	2	3	4	5	6	8
AUROC	0.859	0.932	0.948	0.947	0.937	0.917
	0.058	0.042	0.024	0.025	0.033	0.036

TABLE 9. Impact of α and ξ change on performance. First row: mean AUROC. Second row: standard deviation. Boldface indicates the best result.

		ξ			
		0.25	0.5	0.75	0.85
α	0.25	0.808	0.720	0.749	0.656
		0.085	0.077	0.155	0.162
	0.5	0.755	0.814	0.892	0.794
		0.148	0.081	0.068	0.157
	0.75	0.826	0.860	0.948	0.808
		0.138	0.048	0.024	0.122
	0.85	0.816	0.846	0.910	0.765
		0.094	0.110	0.060	0.153

hyperparameters: σ is the standard deviation of the anomalous latent variable, n_z is the epoch frequency to update the latent variables, n_{dis} is the number of updates per batch on the discriminator, and α and ξ are the fake-anomalous and buffered data balanced parameter, respectively. Each experiment was conducted 10 times using the Bottle category of the MVTEC-AD dataset.

1) STANDARD DEVIATION σ FOR ANOMALOUS LATENT VARIABLE

For a small σ , performance is low and peak performance is reached at $\sigma = 4$ or 5. If the σ value is too large, the support for normal and anomalous data may be separated, which can degrade performance (Table 8).

2) BALANCE PARAMETER α AND ξ FOR FAKE-NORMAL AND BUFFERED DATA

When α is small, the effect of fake-anomalous is large, and when ξ is small, the effect of buffered data is large. The performance is high around the values of $\alpha = 0.75$, $\xi = 0.75$ reported in this study. Recalling (6), ξ is multiplied by α . When $\alpha = 0.25$, $\xi = 0.25$, the effect of the fake-anomalous buffer is greater, and the performance is improved. By contrast, when $\alpha = 0.85$, $\xi = 0.85$, the effect of fake-anomalous and buffered data is smaller, and the performance is lower (Table 9). The results suggest that the effect of fake-anomalous buffer is large, and the parameters α and ξ should be adjusted so that the effect is not too small.

3) UPDATE FREQUENCY FOR LATENT VARIABLE AND DISCRIMINATOR

In this study, n_z peaked at 2, and n_{dis} was good above 2, but performance decreased slightly when discriminator updates became excessive (Tables 10, 11). Therefore, n_z has

TABLE 10. Impact of n_z changes on performance. Top row: mean AUROC. Bottom row: standard deviation. Boldface indicates the best result.

n_z	1	2	3	4
AUROC	0.813	0.948	0.903	0.874
	0.121	0.024	0.035	0.104

TABLE 11. Impact of n_{dis} changes on performance. Top row: mean AUROC. Bottom row: standard deviation. Boldface indicates the best result.

n_{dis}	1	2	3	4
AUROC	0.887	0.948	0.943	0.933
	0.082	0.024	0.031	0.049



FIGURE 8. CutPaste-style processed images. An image patch is sampled from the full image, then color-jittered and pasted onto the original image.

a considerable impact on the performance and should be carefully considered and chosen in practical applications.

G. ROBUSTNESS TO OTHER ANOMALOUS DATA

We also examined the robustness of anomalous data not included in the test data. From Fig. 3, we hypothesized that the generator learns the distribution of real-normal data, and the discriminator lays the discrimination boundary between high-density and low-density regions of normal data. The following two types of data were used for the evaluation: fake-anomalous data generated by $\sigma = 5$ to 8, and CutPaste-style processed images [36] on normal data included in the test data. An image patch was sampled from the full image with an area ratio of 20 to 30% and an aspect ratio between 3 : 1 and 1 : 3, then color-jittered and pasted onto the image.

As in the previous experiments, the discriminator was trained using fake-anomalous data generated by $\sigma = 4$ in the Bottle category. All the anomalous data, generated by $\sigma = 5$ to 8 and CutPaste images, were classified as anomalous. Thus, these results suggest that the discriminator learns similar to one-class classification between normal data and the other data types.

VI. DISCUSSION

The key finding of this study is that adding pseudo-anomalous data to training improves the anomaly detection performance of the discriminator, as shown in Table 7. Fake-anomalous data, one of the pseudo-anomalous data types, are generated from anomalous latent variables with high entropy. This method has fewer concerns about biases than adding

out-of-distribution data to the training [8], [9] or using prior knowledge to generate pseudo-anomalous data [36].

Furthermore, our proposed method uses only the discriminator for anomaly detection, whereas other image-based methods use multiple networks such as encoders, decoders, or both. Thus, the prediction time of our method is faster.

PatchCore [22] is a state-of-the-art feature-based method that uses pre-trained models for images and cannot utilize other types of data. The same is true for other high-performance methods. By contrast, our proposed method can be directly applied to both images and features. Standard GANs have the potential to approximate any data distribution. ALGAN, an extension of standard GANs, has the same potential.

The evaluation in this study focused on image data. However, it would be interesting to see the performance of ALGAN for anomaly or novelty detection in other data types, such as signals [50] and text [51]. This is an important topic for future work.

VII. CONCLUSION

We proposed a novel GAN-based anomaly detection method called ALGAN. The ALGAN generator provides pseudo-anomalous data as well as fake-normal data, by introducing anomalous states in the latent variable. The ALGAN discriminator distinguishes between the group of real-normal data and the group of fake-normal and pseudo-anomalous data.

The proposed method for generating pseudo-anomalous data can be applied to both images and feature vectors. We applied it to three anomaly detection benchmarks and demonstrated its high accuracy.

On MVTec-AD, ALGAN-image achieved more than 10% higher average accuracy than conventional image-based methods, and ALGAN-feature exhibited comparable ability to the feature-based methods. On the COIL-100 dataset, ALGAN performed almost perfectly.

ALGAN exhibited remarkably fast predictions. Compared with methods trained on image data and features, ALGAN-image could predict 10.4 to 54.6 times faster while maintaining high performance, and ALGAN-feature could predict 1.3 to 2.2 times faster.

REFERENCES

- [1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, Jul. 2009, doi: [10.1145/1541880.1541882](https://doi.org/10.1145/1541880.1541882).
- [2] S. Rodda and U. S. R. Erothi, "Class imbalance problem in the network intrusion detection systems," in *Proc. Int. Conf. Electr., Electron., Optim. Techn. (ICEEOT)*, Mar. 2016, pp. 2685–2688.
- [3] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *Proc. Int. Conf. Inf. Process. Med. Imag.*, May 2017, pp. 146–157.
- [4] M. Sabokrou, M. Fathy, M. Hoseini, and R. Klette, "Real-time anomaly detection and localization in crowded scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2015, pp. 56–62.
- [5] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, and C. Steger, "Improving unsupervised defect segmentation by applying structural similarity to autoencoders," in *Proc. 14th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl. (VISAPP)*, Feb. 2019, pp. 372–380.
- [6] A. L. P. Gómez, L. F. Maimó, A. H. Celdrán, F. J. G. Clemente, C. C. Sarmiento, C. J. Del Canto Masa, and R. M. Nistal, "On the generation of anomaly detection datasets in industrial control systems," *IEEE Access*, vol. 7, pp. 177460–177473, 2019.
- [7] R. Chalapathy, A. K. Menon, and S. Chawla, "Anomaly detection using one-class neural networks," Feb. 2018, *arXiv:1802.06360*.
- [8] Y. Kawachi, Y. Koizumi, and N. Harada, "Complementary set variational autoencoder for supervised anomaly detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 2366–2370.
- [9] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," in *Proc. Int. Conf. Learning Represent. (ICLR)*, May 2018.
- [10] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," Jan. 2019, *arXiv:1901.03407*.
- [11] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Müller, "A unifying review of deep and shallow anomaly detection," *Proc. IEEE*, vol. 109, no. 5, pp. 756–795, May 2021.
- [12] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, Mar. 2022, doi: [10.1145/3439950](https://doi.org/10.1145/3439950).
- [13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, May 2015.
- [14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [15] J. Andrews, T. Tanay, E. J. Morton, and L. D. Griffin, "Transfer representation-learning for anomaly detection," in *Proc. ICML*, 2016. [Online]. Available: <https://sites.google.com/site/icmlworkshop/onanomalydetection/accepted-papers>
- [16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
- [17] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, "Efficient GAN-based anomaly detection," Feb. 2018, *arXiv:1802.06222*.
- [18] S. Akçay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly: Semi-supervised anomaly detection via adversarial training," in *Proc. Asian Conf. Comput. Vis. (ACCV)*. Cham, Switzerland: Springer, Dec. 2018, pp. 622–637.
- [19] S. Akçay, A. Atapour-Abarghouei, and T. P. Breckon, "Skip-GANomaly: Skip connected and adversarially trained encoder-decoder anomaly detection," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.
- [20] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 3379–3388.
- [21] M. Rudolph, B. Wandt, and B. Rosenhahn, "Same same but DifferNet: Semi-supervised defect detection with normalizing flows," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 1907–1916.
- [22] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, "Towards total recall in industrial anomaly detection," Jun. 2021, *arXiv:2106.08265*.
- [23] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, Nov. 2017.
- [24] G. Menghani, "Efficient deep learning: A survey on making deep learning models smaller, faster, and better," Jun. 2021, *arXiv:2106.08962*.
- [25] L. Bergman, N. Cohen, and Y. Hoshen, "Deep nearest neighbor anomaly detection," Feb. 2020, *arXiv:2002.10445*.
- [26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 248–255.
- [27] O. Rippel, P. Mertens, and D. Merhof, "Modeling the distribution of normal data in pre-trained deep features for anomaly detection," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 6726–6733.
- [28] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jun. 2019, pp. 6105–6114.

- [29] N. Cohen and Y. Hoshen, "Sub-image anomaly detection with deep pyramid correspondences," May 2020, *arXiv:2005.02357*.
- [30] T. Defard, A. Setkov, A. Loesch, and R. Audigier, "PaDiM: A patch distribution modeling framework for anomaly detection and localization," in *Pattern Recognition. ICPR International Workshops and Challenges*. Cham, Switzerland: Springer, Mar. 2021, pp. 475–489.
- [31] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Apr. 2017.
- [32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, Dec. 2012, pp. 1097–1105.
- [33] P. Chatillon and C. Ballester, "History-based anomaly detector: An adversarial approach to anomaly detection," in *Proc. SAI Intell. Syst. Conf.* Cham, Switzerland: Springer, Aug. 2020, pp. 761–776.
- [34] M. Z. Zaheer, J.-H. Lee, M. Astrid, and S.-I. Lee, "Old is gold: Redefining the adversarially learned one-class classifier training paradigm," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14183–14193.
- [35] M. Pourreza, B. Mohammadi, M. Khaki, S. Bouindour, H. Snoussi, and M. Sabokrou, "G2D: Generate to detect anomaly," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 2003–2012.
- [36] C.-L. Li, K. Sohn, J. Yoon, and T. Pfister, "CutPaste: Self-supervised learning for anomaly detection and localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9664–9674.
- [37] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, May 2016.
- [38] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Sep. 2016, pp. 87.1–87.12. [Online]. Available: <http://www.bmva.org/bmvc/2016/papers/paper087/index.html>
- [39] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2015, pp. 448–456.
- [40] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Apr. 2018.
- [41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, May 2015.
- [42] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD—A comprehensive real-world dataset for unsupervised anomaly detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9592–9600.
- [43] Y. Huang, C. Qiu, and K. Yuan, "Surface defect saliency of magnetic tile," *Vis. Comput.*, vol. 36, no. 1, pp. 85–96, 2020.
- [44] S. A. Nene, S. K. Nayar, and H. Murase. (1996). *Columbia Object Image Library (Coil-100)*. [Online]. Available: <https://www.cs.columbia.edu/CAVE/software/softlib/coil-100.php>
- [45] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, 2020.
- [46] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, Jul. 2021.
- [47] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. 10, pp. 2825–2830, Jul. 2017.
- [48] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Apr. 2018.
- [49] S. Sinha, H. Zhang, A. Goyal, Y. Bengio, H. Larochelle, and A. Odena, "Small-GAN: Speeding up GAN training using core-sets," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Apr. 2020, pp. 9005–9015.
- [50] E. Brophy, Z. Wang, Q. She, and T. Ward, "Generative adversarial networks in time series: A survey and taxonomy," Jul. 2021, *arXiv:2107.11098*.
- [51] G. H. de Rosa and J. P. Papa, "A survey on text generation using generative adversarial networks," *Pattern Recognit.*, vol. 119, Nov. 2021, Art. no. 108098.



HIRONORI MURASE received the B.E. and M.E. degrees in engineering from Tohoku University, Japan, in 2008 and 2010, respectively. He is currently pursuing the Ph.D. degree in statistical science with The Graduate University for Advanced Studies, SOKENDAI, Japan. Since 2010, he has been working as an Engineer with Aichi Steel Corporation. His current research interests include computer vision, deep neural networks, and anomaly detection.



KENJI FUKUMIZU received the B.S. and Ph.D. degrees from Kyoto University, in 1989 and 1996, respectively. He is a Professor with The Institute of Statistical Mathematics, where he is the Director of the Research Center for Statistical Machine Learning. He joined the Research and Development Center, Ricoh Company, Ltd., in 1989. He was a Researcher at the Institute of Physical and Chemical Research (RIKEN) and became an Associate Professor at the Institute of Statistical Mathematics, in 2000. He was a Visiting Scholar at the Department of Statistics, UC Berkeley, from 2002 to 2003, and a Humboldt Fellow at the Max Planck Institute for Biological Cybernetics, from 2006 to 2007. He has been a Technical Advisor with Preferred Networks, Inc., since 2018. He has served as an organizing committee member for many conferences, including a Program Co-Chair of AISTATS 2021, and an Area Chair on the Program Committees of NeurIPS and ICML. He was the Chief Editor of the Annals of the Institute of Statistical Mathematics, from 2011 to 2017 and currently serves as an Action Editor for the *Journal of Machine Learning Research*.

• • •