Check for
updates

A Uniform Framework for Anomaly Detection in Deep Neural Networks

Fangzhen Zhao[1] · Chenyi Zhang[1] · Naipeng Dong[2] · Zefeng You[1] · Zhenxin Wu[1]

Abstract

Deep neural networks (DNN) can achieve high performance when applied to In-Distribution (ID) data which come from the same distribution as the training set. When presented with anomaly inputs not from the ID, the outputs of a DNN should be regarded as meaningless. However, modern DNN often predict anomaly inputs as an ID class with high confidence, which is dangerous and misleading. In this work, we consider three classes of anomaly inputs, (1) natural inputs from a different distribution than the DNN is trained for, known as Out-of-Distribution (OOD) samples, (2) crafted inputs generated from ID by attackers, often known as adversarial (AD) samples, and (3) noise (NS) samples generated from meaningless data. We propose a framework that aims to detect all these anomalies for a pre-trained DNN. Unlike some of the existing works, our method does not require preprocessing of input data, nor is it dependent to any known OOD set or adversarial attack algorithm. Through extensive experiments over a variety of DNN models for the detection of aforementioned anomalies, we show that in most cases our method outperforms state-of-the-art anomaly detection methods in identifying all three classes of anomalies.

Keywords Deep neural networks · Out-of-distribution · Adversarial attack · Anomaly detection

1 Introduction

Although being an emerging technology, Deep Neural Networks (DNN) is valued to be $38.71 billion globally by 2023, with wide range of applications cross sectors like finance, energy & utilities, retail, IT & telecom, manufacturing, aerospace & defence, healthcare etc. (according to Allied Market Research) [36]. Along with DNN's popularity is a growing concern on the safety of DNN models in carrying out the tasks (typically classification)

✉ Chenyi Zhang
chenyi_zhang@jnu.edu.cn

1   College of Computer Science and Technology, Jinan University, Guangzhou, China

2   School of Information Technology and Electric Engineering, University of Queensland, Brisbane, Queensland, Australia

in those applications, especially the security or safety critical ones such as healthcare and self-driving vehicles.

To address the concern, the foremost issue is to ensure the quality of the input data, which the DNN models depend on. As a data-driven technique, a DNN model will only be as good or as bad as the data provided (training data). For instance, a speech recognition system trained on clean speech will not perform well on noisy speech. However, when applied to real-world tasks, it is inevitable that the testing data differs from the training data due to a variety of reasons, such as mis-operations in data collection, natural noise, untrusted data resources etc. Such unavoidable anomaly testing data leads to severe safety problems - the DNN tends to provide high-confidence predictions while being woefully incorrect [16].

A variety of works exist aiming to detect anomaly in testing data, most of which focus on the applications of image classification. We classify these works according to the types of anomaly data they handle: out-of-distribution (OOD) data, adversarial (AD) data, and noise (NS) data.

OOD Out-of-Distribution (OOD) data refers to inputs that do not contain any of the classes modeled in the training distribution. For example, the clothing images from Fashion-MNIST are OOD data for a DNN trained with the MNIST data set which consists of hand-written digits. The OOD data considered in this paper are collections of *meaningful natural* images that are not from ID, excluding those *crafted* images (known as adversarial data) and *meaningless* images (classified as noise data).

AD Adversarial (AD) data is generated by introducing an imperceptible perturbation to an image from the in-distribution (ID), with the intention of inducing a DNN to make wrong judgments. The adversarial methods used in our experiments include FGSM [13], BIM [22], JSMA [33] and CW [5].

NS We consider two types of noise (NS) data. The first type (NS-I) is merely random noise (e.g., Gaussian noise). The second type (NS-II) is often known as fooling images, which are created by evolving meaningless images in order to mislead a DNN to output classes in ID with high confidence, such as the images generated in [30] (examples shown in Figure 9).

Through a comprehensive literature study (see Table 1 in Sect. 2), we observe that existing works either focus only on OOD and NS-I detection, or focus on AD detection. Few works detect both OOD and AD (a notable exception is [25]); and no works aim to detect NS-II.

As suggested by [4], there are no known intrinsic properties that differentiate natural images and adversarial images. We believe that for real-world tasks, all types of anomalies could potentially be fed into DNN models, and there is no effective way to tell ahead of time if they are in-distribution, adversarial, or out-of-distribution images. Many existing works make an implicit assumption that the analysers know which type of anomaly in advance and build detection approaches for the particular type, which is impossible in reality. Therefore, detection approaches should be developed to be able to handle all three types of anomalies that people are aware of.

The uncertainty of the anomaly types leads to a non-trivial challenge in the anomaly detection task. During experiments, we observe that, (1) in general, an approach for OOD performs badly for detecting AD and vice versa. Any combination of the results for each approach would not be able to outperform either the best result for OOD or the best result for AD. In addition, we observe (2) many existing works require pre-processing to the input data. This hinders their applicability in real-world tasks, since one needs to know, in advance, which data is ID and which is OOD in the case of OOD detection, and needs to know by which adversarial algorithm the data is generated in the case of AD detection. Removing

Table 1 A summary of related work

Method	OOD& NS-I	AD	NS-II	Pre-proc.
Baseline [16]	√	×	×	×
ODIN [26]	√	×	×	√
ELO [1]	√	×	×	×
OE [17]	√	×	×	×
G-ODIN [18]	√	×	×	√
KD+BU [11]	×	√	×	√
LID [27]	×	√	×	√
IF [8]	×	√	×	√
FS [46]	×	√	×	×
MD [25]	√	√	×	√

the prior knowledge would largely degrade performance. Therefore, this work improves the anomaly detection accuracy without requiring prior knowledge of input data. Our approach detects anomaly using features from the Most Discriminative Layer (MDL) which has better distinction for the sub-domain of the test input. The anomaly detector then combines the MDL features with the Logit layer to cover all the three types of anomalies.

Contributions This paper proposes a uniform framework for detecting anomaly inputs, which has wider coverage of the anomaly types, easier applicability to various models, and better performance in accuracy, compared to the state-of-the-art approaches.

Coverage	Our approach is able to detect all three types of anomaly inputs, including OOD, AD and NS, which addresses the challenge of unknown anomaly types in real-world applications.
Applicability	Our approach provides a uniform way for the anomaly detection, and thus can be applied to any existing model without requiring extra pre-processing.
Performance	To evaluate our approach, we conduct extensive experiments and a comprehensive comparison with the state-of-the-art approaches. The experiment results show that our approach outperforms the best results in OOD, AD and NS in most of the cases.

2 Related Work

As mentioned in the previous section, existing works either focus on OOD and NS-I detection, e.g., [1, 12, 16–18, 26], or focus on AD detection, e.g., [8, 11, 27, 42, 46]. The only work that can detect both is Mahalanobis distance (*MD*) [25].

OOD+NS-I The seminal work for OOD detection is known as the *baseline* approach [16], which observes that the softmax prediction probability of OOD tends to be lower than the prediction probability for correct examples, and thus a threshold over the predicted softmax probability can be used to detect OOD. A year later, the work *ODIN* observes that temperature scaling and input perturbation (pre-processing) can enlarge the gap between ID and OOD, and thus can be used to improve the detection performance [26]. Meanwhile, the pre-processing of ODIN requires access to OOD samples in advance, which is impossible in reality, to fine-tune the degree of perturbation. To address this limitation, the work Early Layer Output (*ELO*) [1] proposes a one-class classifier trained on the output of an early layer instead of

the softmax layer, which does not need to access to OOD samples. An alternative approach called Generalized ODIN (*G-ODIN*) [18] is later proposed, which only tunes the ID data instead of OOD data. Differing from the above works, the work Outlier Exposure (*OE*) [17] fine-tunes the pre-trained model using an auxiliary data set that is selected from a disjoint set of OOD samples. OE includes an additional loss function to minimize the distance between the output distribution produced by the pre-trained model for the auxiliary data set and the uniform distribution. The softmax values are used as scores for anomaly detection. Similar to ELO and OE, our approach does not require preprocessing and the use of the OOD samples in training. Nevertheless, our approach extracts data from a specific sub-domain, which achieves better separation of ID and OOD (see Fig. 3).

AD A variety of approaches have been proposed to detect adversarial samples from their normal and noisy counterparts. For instance, Feinman *et al.* trains a logistic regression detector using a distance-based generative learning method called kernel density and Bayesian uncertainty features (*KD+BU*) [11]. Ma *et al.* proposes an intrinsic character of the adversarial regions - the local intrinsic dimensionality (*LID*), as the confidence score to separate the adversarial samples [27]. The *IF* approach demonstrates the correspondence between the training data and the classification of the network, which is quantified using the Influence Function, and outperforms LID [8]. Differing from the above works, Xu *et al.* applies Feature Squeezing (*FS*) to distinguish the adversarial samples from ID and does not need special treatment to the input data [46]. These works can only detect AD and are not suitable for detecting OOD and NS.

Finally, the work *MD* measures the probability density of a test sample and uses the Mahalanobis distance as the confidence score to distinguish OOD and AD from ID. It is the first work that can detect both OOD and AD [25]. Our work is able to detect an additional type (NS-II) of anomaly and outperforms MD on most of the test data sets.

3 Preliminaries

In this section, we introduce the notions that are necessary to understand the remaining part of the paper. Let a deep neural network (DNN) of m layers be represented as a function $f : \mathcal{I} \rightarrow \mathcal{O}$, where $\mathcal{I}$ is the input domain and $\mathcal{O}$ is the domain for the output vectors of length d. Given $x \in \mathcal{I}$, we have $f(x) = \langle o_1, \ldots, o_d \rangle \in \mathcal{O}$, and the final classification chosen by the DNN is $C^f(x) = \underset{i \in \{1, \cdots, d\}}{argmax} (o_i)$ for the index of the largest element in vector $f(x)$. For $\ell = 1 \ldots m$, we write $f^{\ell}(x)$ for the output vector in the feature space of layer ℓ. In the literature, the second last layer (i.e., $\ell = m - 1$, right before the softmax layer) is often called the *logit* layer.

It is commonly assumed that the training data of DNN f is drawn from the distribution Δ known as ID. we write $\Delta_{in}(x)$ if $x \in \mathcal{I}$ is from ID given f. An anomaly detector g_f for DNN f is a binary classifier, such that given input $x \in \mathcal{I}$, $g_f(x)$ answers whether x is an anomaly with respect to f. Since it is often difficult to define what an *anomaly distribution* is, we focus on the three types of anomalies (i.e., OOD, AD and NS) in our experiments.

Our anomaly detection algorithm is based on a discriminative model known as Support Vector Domain Description (SVDD) by [41].[1] Similar to the famous Support Vector Machine [43], SVDD defines support vectors for a sphere shaped decision boundary enclosing the class

---

[1] Another important approach for one class classification known as ν-SVC is introduced by Schölkopf et al. [38], which can be shown as equivalent to SVDD when the Gaussian kernel is used [40].

of objects represented by the (unlabeled) training data with minimal space, as shown in the following formulation.

$$\min_{R,\mathbf{a},\xi} \ R^2 + \frac{1}{n\nu} \sum_i \xi_i \tag{1}$$

$$\text{s.t.} \quad \forall i : \quad \mathbf{x}_i - \mathbf{a}^2 \le R^2 + \xi_i, \quad \xi_i \ge 0$$

The solution of the above constraints provides a center vector $\mathbf{a}$, radius R and slack variables ξ_i such that the target term in Eq. (1) is minimized, provided that the square of distance from each training data $\mathbf{x}_i$ to the center $\mathbf{a}$ may exceed R^2 by at most ξ_i. Here ν is a constant in $(0, 1]$ and n is the size of the training set. Intuitively, a smaller ν gives more weight to the right hand side of target term in Eq. (1), which imposes smaller values for ξ_i and larger R. The solution of Eq. (1) allows us to determine if a test input $\mathbf{z}$ is from the ID by checking the following condition.

$$\mathbf{z} - \mathbf{a}^2 = (\mathbf{z} \cdot \mathbf{z}) - 2 \sum_i \alpha_i (\mathbf{z} \cdot \mathbf{x}_i) + \sum_{i,j} \alpha_i \alpha_j (\mathbf{z} \cdot \mathbf{x}_i) \le R^2 \tag{2}$$

Here α_i (α_j) is the Lagrange multiplier associated with the constraint for the i-th (j-th) training input when solving Eq. (1), which is non-zero only if the i-th (j-th) training input is used as a support vector. Given all inputs (including the test input) only appearing in the form of inner product, it is thus viable to replace the inner products by kernel functions, of which the Gaussian Radial Basis Function (RBF) provides the best performance in practice [40]. The RBF kernel is given in the following formulation, where the free parameter s controls the spread, or how tight the density is, of the kernel.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/s^2) \tag{3}$$

Early-Layer Output (ELO) [1], the work most related to ours, trains a one-class SVDD classifier using an early-layer output of ID data in a latent space, based on the observation that there exists an early-layer called the Most Discriminative Layer (MDL), such that in this latent space the ID data and OOD data are well separated.

4 Our Approach

We propose a uniform framework for the anomaly detection task. In the training phase, for a given DNN classifier, we empirically choose the Most Discriminative Layer (MDL) using a randomly picked OOD set (mix data), in the way similar to ELO [1]. Then we use the data generated from the MDL layer (Step 1) and the logit layer of the DNN to train two one-class SVDD classifiers for each known class (Step 2). During the testing phase, an input is first given to the DNN classifier which produces an output class i. The data from the corresponding MDL and logit layers are forwarded to the corresponding SVDD classifiers, i.e., SVDD_i^1 and SVDD_i^2, and the scores obtained from the SVDD classifiers are combined to form a final judgment on whether the given input is anomalous (Step 3). The overview of our approach is sketched in Fig. 1.

Let layer ℓ be the MDL in a reasonable DNN classifier f to ID with sub-domains for the known classes $1 \ldots d$, and the set $\{f^\ell(x) \mid \Delta_{in}(x)\}$ forms a manifold of ID in the latent space of layer ℓ. We conjecture that anomaly detection precision can be further improved
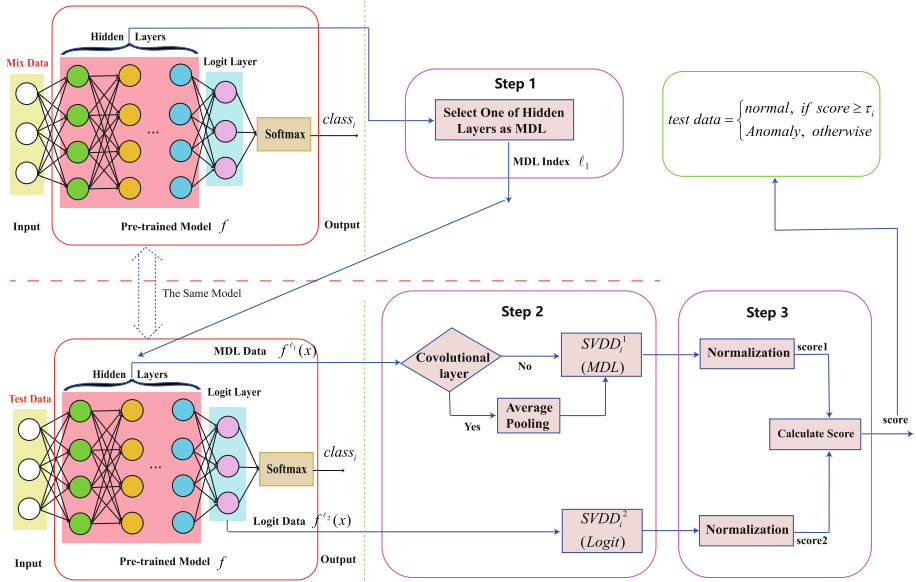
Fig. 1 The overview of the proposed approach. The proposed approach has three steps. *Step 1* Select one of hidden layers as MDL which has the minimum detection errors for mix data. *Step 2* Feed the extracted the data of the MDL and logit layer into the corresponding $SVDD_i^\ell$. *Step 3* Calculate the score of an input sample by using the corresponding $SVDD_i^\ell$ in MDL and logit layer, which helps to detect anomaly inputs
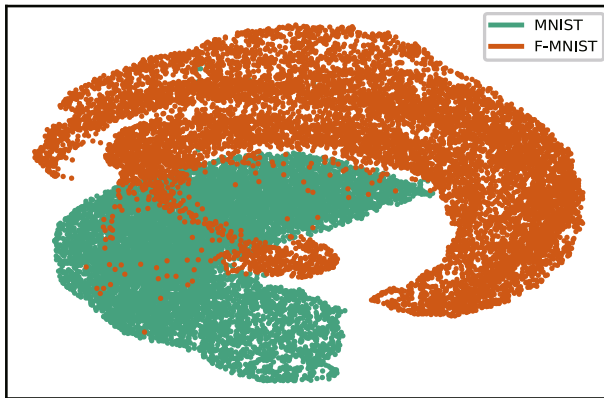

Fig. 2 A two-dimensional representation of features extracted from the MDL of a LeNet model trained on MNIST. The feature cluster consisting of data from all 10 MNIST classes is shown in green dots, and yellow dots represent F-MNIST (OOD) data

by making predictions conditional to output of the DNN. We focus on the sub-domains $\{f^\ell(x) \mid \Delta_{in}(x) \wedge C^f(x) = i\}$ for each known class $i = 1 \ldots d$.

Taking Eq. (1), the approach of ELO empirically chooses $\nu = 0.001$, which imposes a strong penalty on samples with distance larger than R (i.e., the slack variables should be very small) when minimizing the entire term of Eq. (1), which is reasonable if the sub-domains for different classes are relatively apart. Since R becomes relatively large, potentially more
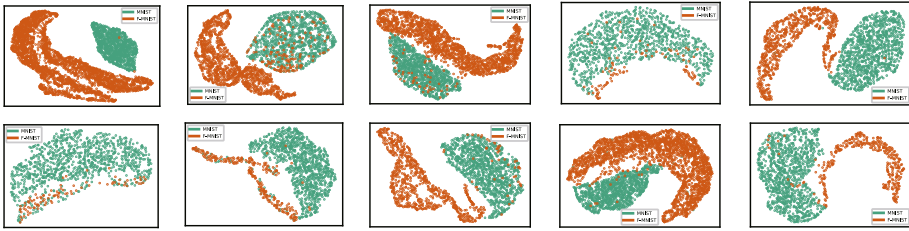
Fig. 3 Two-dimensional representations of features extracted from the MDL of a LeNet model trained on MNIST. The feature clusters for the 10 classes are shown, with green dots for MNIST (ID) data and yellow dots for F-MNIST (OOD) data

Table 2 Comparison of our approach with the baseline, ODIN, Mahalanobis distance (MD) and ELO for OOD data

Model	OOD	TNR at 95% TPR ↑	AUROC ↑
		baseline/ODIN/MD/ELO/ours	
LeNet	F-MNIST	97.90/99.59/95.60/99.15/ **99.72**	99.26/99.81/98.81/99.77/**99.83**
(MNIST)	Omniglot	97.03/99.86/93.62/**100.0**/ **100.0**	98.82/99.81/98.48/**100.0**/99.99
VGG-16	TinyIm	36.14/55.76/46.61/84.73/**85.61**	89.79/92.79/90.68/96.45/**97.54**
(CIFAR10)	LSUN	38.82/63.41/56.92/91.97/**92.81**	90.90/94.58/93.05/98.13/**98.98**
	iSUN	39.06/63.65/54.90/89.22/**92.64**	90.89/94.53/92.61/97.77/**98.70**
	SVHN	27.89/43.58/23.28/**87.44**/68.07	89.01/91.92/85.56 /**97.15**/95.66
ResNet	TinyIm	30.32/48.41/15.52/**95.08**/94.24	87.13/90.58/79.46/**98.91**/98.61
(CIFAR10)	LSUN	35.47/67.04/22.84/96.21/**97.34**	89.66/94.72/85.22/99.07/**99.25**
	iSUN	34.25/63.47/22.45/93.61/**95.34**	89.22/94.10/84.41/98.64/**99.01**
	SVHN	38.71/67.27/24.77/85.29/**86.29**	89.74/93.80/82.19/96.74/**97.00**
VGG-16	TinyIm	78.69/88.79/76.37/92.06/**93.97**	96.92/97.99/96.56/98.21/ **98.87**
(SVHN)	LSUN	78.27/88.13/78.42/93.57/**94.80**	96.80/97.83/96.71/98.39/**99.05**
	iSUN	81.48/90.83/79.82/93.55/**95.38**	97.26/98.27/96.95/98.46/**99.08**
	CIFAR10	78.13/88.87/76.39/71.88/**87.37**	97.67/97.91/85.69/94.62/**97.83**
ResNet	TinyIm	75.38/85.39/63.93/93.74/**95.73**	96.21/97.17/94.37/98.62/**99.01**
(SVHN)	LSUN	72.67/83.02/58.96/96.27/**96.58**	95.85/96.72/93.64/99.06/**99.18**
	iSUN	75.65/86.39/63.04/95.75/**96.64**	96.28/97.30/94.20/98.99/**99.21**
	CIFAR10	74.12/85.12/63.91/80.14/**87.60**	96.04/97.07/94.47/95.35/**97.41**

Bold indicates sufficient values

anomaly samples, especially those spatially closer to ID such as adversarial samples, are classified as ID. Moreover, an anomaly input positioned between two ID clusters of distinct classes in the latent space may also be classified as ID. Figure 2 provides a 2-D view of the clusters of MNIST samples (green dots) and Fashion-MNIST samples (yellow dots) in the MDL of a LeNet model.[2] If we split the MNIST samples and Fashion-MNIST samples into 10 classes based on the classification results of the LeNet model, and study the MNIST samples and Fashion-MNIST samples confined to each class, we get the 2-D view in Fig. 3, from which it seems that near perfect separation can be achieved in some classes (e.g., the 1st, 5th, 9th and 10th classes). Our experiment results shown in Tables 2 and 3 confirm that the

---

[2] We choose UMAP [28] as the visualization tool.

Table 3 Comparison of our approach with the KD+BU, LID, MD and ELO for adversarial data

Model	AD	TNR at 95% TPR ↑	AUROC ↑
		KD+BU/LID/MD/ELO/ours	
LeNet	FGSM	83.91/94.65/83.11/**100.0/100.0**	86.86/89.03/96.66/99.97/**99.98**
(MNIST)	BIM-a	**96.29**/51.29/70.82/54.25/95.57	98.88/89.04/95.81/87.82/**99.04**
	BIM-b	28.51/34.70/23.63/99.28/**99.32**	74.07/73.32/77.29/99.57/**99.64**
	JSMA	**96.78**/81.24/89.72/16.77/89.70	**98.90**/94.60/97.78/57.40/98.16
	CW	**97.23**/60.30/66.33/6.77/79.73	**99.07**/90.60/95.12/50.81/97.12
VGG-16	FGSM	77.35/92.81/73.82/92.41/**96.74**	87.66/98.11/95.75/98.59/**99.22**
(CIFAR10)	BIM-a	40.58/7.76/**85.93**/1.47/63.57	76.36/64.40/**97.27**/51.08/94.10
	BIM-b	11.76/3.44/2.34/1.21/**15.80**	52.25/48.98/75.95/57.55/**81.53**
	JSMA	77.20/37.08/**96.91**/4.97/83.60	94.65/67.70/**98.73**/48.49/97.13
	CW	49.92/6.93/**86.22**/7.39/81.18	80.04/66.11/**96.96**/65.27/96.53
ResNet	FGSM	38.93/99.95/9.70/99.98/ **99.96**	75.70/99.98/82.14/**100.0**/99.81
(CIFAR10)	BIM-a	17.48/11.74/6.63/14.56/**35.65**	65.77/60.12/71.41/72.69/**83.67**
	BIM-b	99.89/80.43/**99.99**/95.17/99.98	99.93/90.74/**99.99**/98.66/99.99
	JSMA	60.0/71.80/35.00/83.76/**91.62**	76.38/67.23/92.17/93.44/**97.74**
	CW	26.38/12.49/9.74/9.63/**37.98**	75.75/60.91/82.99/63.21/**86.84**
VGG-16	FGSM	86.91/99.90/83.50/99.91/**97.69**	89.70/99.50/96.93/99.21/**99.46**
(SVHN)	BIM-a	**57.05**/15.64/53.43/4.21/40.81	**91.19**/74.40/89.81/55.35/81.36
	BIM-b	1.35/76.36/22.69/20.09/**98.84**	60.41/91.84/91.85/87.67/**99.17**
	JSMA	**94.05**/27.94/93.81/5.67/76.76	98.33/81.04/**98.52**/55.55/94.72
	CW	75.20/11.28/**100.0**/4.43/66.88	97.03/71.18/**98.48**/49.38/94.54
ResNet	FGSM	73.89/99.56/72.93/**100.0**/99.76	83.01/99.55/95.30/**99.85**/99.80
(SVHN)	BIM-a	50.60/16.81/**54.5**/3.44/24.53	**89.66**/74.60/89.40/43.52/77.33
	BIM-b	**100.0**/99.12/ **100.0**/66.31/**100.0**	100.0/98.50/**100.0**/94.58/99.98
	JSMA	**84.33**/34.73/84.05/8.28/51.04	95.88/78.93/**97.02**/53.45/90.81
	CW	**83.17**/9.26/78.72/4.48/54.53	**97.26**/63.81/96.68/48.40/92.36

Bold indicates sufficient values

method with sub-domain splitting is comparable (if not marginally better) to the ELO method for the detection of OOD inputs, and for the detection of adversarial inputs the sub-domain based method significantly outperforms ELO.[3]

Our second observation is that the features represented by different layers of a DNN indeed represent distinctive discriminative power on anomalies. In the literature the softmax and logit layers [16, 25, 26] are used to distinguish OOD from ID, while some others consider the usage of early layers [1, 19]. Since anomaly data in the class AD are crafted by introducing imperceptible perturbations to images from the ID, they are closer to the ID in the input domain than OOD and NS in the majority of adversarial attack scenarios, especially the AD with relatively small perturbation. Therefore, intuitively, it requires more processing in the original classifier to separate them from ID in the penultimate logit layer, rather than the earlier MDL layer. This has also been described in detail in the literature [32]. Figure 4 provides a 2-D view of the clusters of MNIST ID samples (green dots) and JSMA AD samples

---

[3] Empirically, in our experimental setting, we let $\nu = 0.1$ which ensures a tighter bound (i.e., a smaller R) on each individual class to rule out adversarial samples.

Table 4 Comparison of our approach with the baseline, ODIN and MD for noise data

Model	NS	TNR at 95% TPR ↑	AUROC ↑
		baseline/ODIN/MD/ours	
LeNet	Gaussian	100.0/100.0/100.0/**100.0**	99.58/100.0/99.77/**100.0**
(MNIST)	Uniform	99.27/100.0/99.25/**100.0**	98.31/99.92/98.85/**100.0**
	FoolIm	0.0/0.0/2.07/**100.0**	75.56/86.73/68.61/**99.74**
VGG-16	Gaussian	7.75/58.42/97.43/**100.0**	90.26/95.21/98.83/**100.0**
(CIFAR10)	Uniform	49.56/87.79/99.81/**100.0**	94.72/96.78/99.28/**100.0**
	FoolIm	0.0/0.0/0.0/**97.23**	71.60/79.53/71.82/**98.14**
ResNet	Gaussian	30.22/58.51/33.67/**100.0**	89.08/93.90/89.87/**99.67**
(CIFAR10)	Uniform	24.56/53.40/23.48/**100.0**	87.96/93.38/89.13/**99.77**
	FoolIm	0.0/0.0/16.83/**100.0**	72.84/82.74/76.65/**99.73**
VGG-16	Gaussian	84.76/91.95/80.21/**99.53**	97.67/98.49/97.00/**99.70**
(SVHN)	Uniform	90.49/96.35/83.88/**98.99**	98.35/99.11/97.30/**99.63**
	FoolIm	0.0/0.0/0.0/**100.0**	16.65/19.45/37.91/**98.39**
ResNet	Gaussian	83.52/93.28/66.61/**100.0**	97.28/98.32/95.31/**99.92**
(SVHN)	Uniform	82.68/93.08/62.76/**99.98**	42.09/46.74/94.89/**99.86**
	FoolIm	0.0/0.0/1.30/**96.93**	42.09/46.74/53.18/**98.63**

Bold indicates sufficient values

(yellow dots) in the MDL of a LeNet model. Figure 5 and Figure 6 show the feature clusters for the 10 classes of MNIST ID samples (green dots) and JSMA AD samples (yellow dots) in the MDL and logit layer of a LeNet model, respectively. Such a conjecture is also confirmed by the results in Tables 2 and 3. We believe that combining the power of the early layers and the late layers can achieve better precision on detecting different types of anomalies.

Based on the above observations, we train two SVDD detectors for layers ℓ_1 and ℓ_2 for each class $i \in \{1, \ldots d\}$, and combine the results by defining $g_i(x) = \beta_1 \cdot g_{f,i}^{\ell_1}(x)^* + \beta_2 \cdot g_{f,i}^{\ell_2}(x)^*$ as a score to determine if x is an anomaly, given $C^f(x) = i$. In the above formulation, we choose the MDL as ℓ_1 which is empirically determined and it is most of the time an early layer that gives better precision on detection anomalies than any other layers, and ℓ_2 is the logit layer. When used for combining scores, $g_{f,i}^{\ell}(x)^*$ is the normalized value of $g_{f,i}^{\ell}(x)$, which applies here because $g_{f,i}^{\ell_1}$ and $g_{f,i}^{\ell_2}$ tend to produce scores of different scales. Coefficients β_1 and β_2 are used to balance the weights from the two detectors. As shown in Fig. 7 which is the result of a preliminary experiment, setting $\beta_1 = \beta_2 = 0.5$ produces a close to optimal precision on all the given OOD, AD and NS data sets when CIFAR-10 is ID in a ResNet model. This figure also suggests that relying only on the MDL layer may provide acceptable results on detection of Out-of-Distribution anomalies (by treating TinyIm, LSUN, iSUN, SVHN as OOD) and noise detection (e.g., Gaussian noises), while relying only on the logit layer may provide acceptable results on detection of a few adversarial attacks.

Threshold Similar to other methods [16, 26], we need to have thresholds to distinguish normal inputs and anomaly inputs. Different from most other works, we define multiple thresholds based on classes of training samples. In the case of MNIST, there are 10 classes of ID data, so we define a threshold for each class. When a sample x is given to the DNN model which generates output class i, our approach collects data from the MDL layer and logit layer for the SVDDs to generate a score to compare with τ_i. The threshold τ_i is computed in the way to ensure that 95% of the test samples from class i of ID have scores above τ_i. The
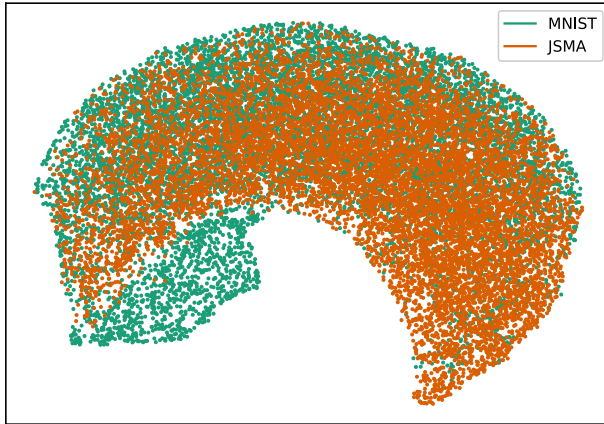
Fig. 4 A two-dimensional representation of features extracted from the MDL of a LeNet model trained on MNIST. The feature cluster consisting of all 10 MNIST classes is shown in green dots, and yellow dots represent JSMA (AD) data
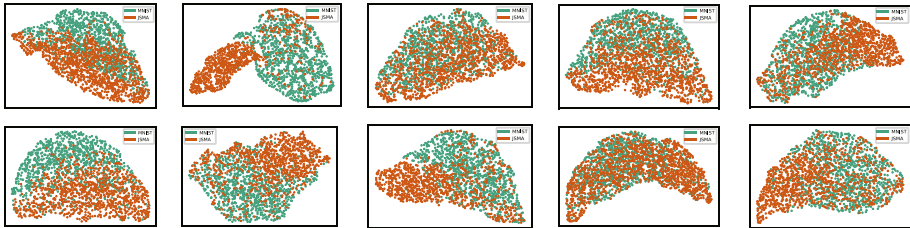

Fig. 5 Two-dimensional representations of features extracted from the MDL of a LeNet model trained on MNIST. The feature clusters for the 10 classes are shown, with green dots for MNIST (ID) data and yellow dots for JSMA (AD) data

Fig. 6 Two-dimensional representations of features extracted from the Logit layer of a LeNet model trained on MNIST. The feature clusters for the 10 classes are shown, with green dots for MNIST (ID) data and yellow dots for JSMA (AD) data
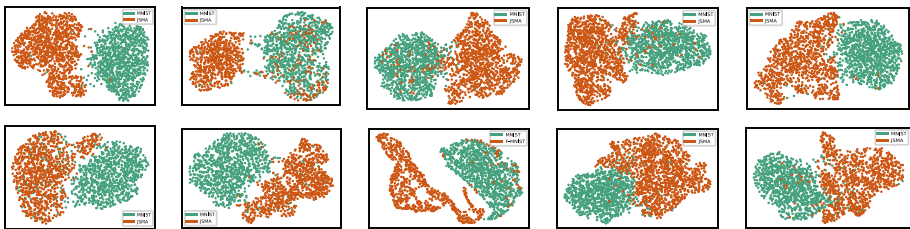
threshold-based discriminator can be formally described as follows.

$$isAnomaly(x) = \begin{cases} True, & if \ \ g_i(x) < \tau_i \\ Fasle, & if \ \ g_i(x) \geq \tau_i \end{cases} \tag{4}$$

Note that SVDD performs poorly on high-dimensional data. Given the MDL layer is usually an early convolutional layer, the feature space of the MDL is often high-dimensional. In this case, we compute the mean of each channel to reduce the dimension of the extracted
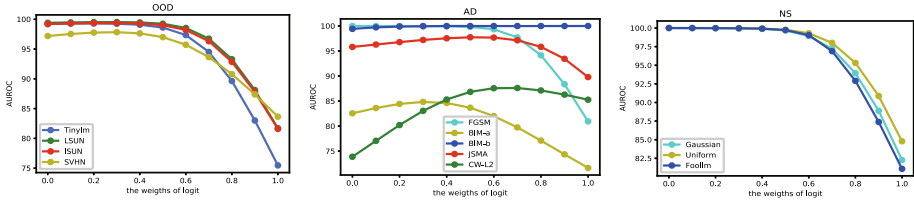
Fig. 7 AUROC of the combined anomaly detection model with different weight coefficients for values from the logit layer of a ResNet model (CIFAR-10 as ID)

features from the MDL layer. More precisely, let $f^\ell \in \mathbb{R}^{d \times w \times h}$ be the feature maps of a convolutional layer, where d, w and h are depth, width and height, respectively. Then the feature size for the SVDD classifier is d, which is reduced from $d \times w \times h$ to d, with each dimension taking the average of all $w \times h$ values of the same depth.

Nomalization As we mentioned before the SVDD classifiers for the MDL layer and the logit layer tend to generate scores of different scales. If we simply combine the scores with incomparable scales, it is likely to weight one SVDD classifier more than the other, leading to undesirable results. Therefore, a normalization process is essential. In our approach, we apply the min-max normalization procedure.

$$score_i^* = \frac{score_i - score_{min}}{score_{max} - score_{min}} \tag{5}$$

where the $score_{min}$ and $score_{max}$ are the minimum and maximum of the score vector, respectively.

In summary, our SVDD detectors are trained from ID data only, i.e., $g_{f,i}^\ell$ only depends on the feature space at layer ℓ of training inputs if $C^f(x) = i$. Most of the ID data are wrapped inside the hypersphere decision boundary in the feature space, as defined by Eq. (1), and the hyperparameter ν controls the relative size R and the percentage of training data to be outside of the boundary. Through some initial experiments we empirically choose an early layer, i.e., the MDL, and determine the coefficients β_1 and β_2, such that the detector $g_i(x) = \beta_1 \cdot g_{f,i}^{\ell_1}(x)^* + \beta_2 \cdot g_{f,i}^{\ell_2}(x)^*$ produces a score by combining information from the MDL (as ℓ_1) and the logit layer (as ℓ_2), if x is likely to be from class i (i.e., $C^f(x) = i$). This score is then used to decide whether x is anomaly.

5 Experiments, Evaluations and Discussions

We conduct experiment on three types of pre-trained DNN models, with three data sets chosen as ID, against various types of OOD, AD and NS data sets. Our testing code is publicly available at https://github.com/fangzhenzhao/AnomalyDetection_NPL.

5.1 Experiment Settings

We choose three popular DNN models used for image classification. All DNN models are pre-trained. The anomaly detection algorithm is run on a Windows 10 desktop equipped with Intel I7-9700 3.0GHz processor, 16G RAM and Nvidia GetForce GTX1660Ti.

Table 5 Comparison of results with OE about the baseline, ODIN, Mahalanobis distance (MD) and ELO for OOD data

Model	OOD	TNR at 95% TPR ↑	AUROC ↑
		baseline+OE/ODIN+OE/MD+OE/ELO+OE/ours+OE	
LeNet	F-MNIST	98.98/99.49/94.85/99.13/**99.73**	99.58/99.84/98.53/99.76/**99.87**
(MNIST)	Omniglot	99.91/99.97/98.82/**100.0/100.0**	99.75/99.92/99.27/**100.0/100.0**
VGG-16	TinyIm	55.74/66.46/65.85/81.24/**81.93**	92.57/93.90/93.07/95.63/**96.74**
(CIFAR10)	LSUN	60.56/75.72/77.23/90.20/**91.81**	94.02/95.98/96.10/97.68/**98.66**
	iSUN	60.07/74.49/75.26/87.91/**90.20**	93.77/95.78/95.56/97.41/**98.33**
	SVHN	60.85/74.65/78.96/75.53/**95.88**	94.80/95.92/94.58/95.94/**98.22**
ResNet	TinyIm	46.50/65.42/13.90/**94.43**/93.55	90.78/93.60/80.95/**98.76**/98.47
(CIFAR10)	LSUN	53.60/80.98/21.51/96.22/**97.43**	93.02/96.74/86.31/99.06/**99.27**
	iSUN	53.23/80.94/21.69/93.74/**95.55**	92.84/96.66/86.12/98.62/**99.04**
	SVHN	71.10/81.27/51.46/85.44/**89.41**	95.61/97.35/91.22/96.78/**97.66**
VGG-16	TinyIm	87.98/96.48/94.57/87.09/**96.65**	97.78/99.17/98.54/97.36/**99.30**
(SVHN)	LSUN	85.50/95.48/93.33/84.92/**95.75**	97.54/99.00/98.37/97.12/**99.28**
	iSUN	89.31/97.10/95.05/85.77/**96.97**	97.96/99.26/98.70/97.29/**99.35**
	CIFAR10	84.54/92.22/93.56/59.61/**92.30**	97.40/98.07/98.33/92.52/**98.52**
ResNet	TinyIm	87.82/91.29/77.02/92.95/**95.56**	97.52/98.10/95.75/98.45/**98.91**
(SVHN)	LSUN	86.20/90.38/74.55/95.79/**96.68**	97.30/97.97/95.29/98.96/**99.19**
	iSUN	88.50/92.43/78.39/94.87/**96.53**	97.66/98.33/95.82/98.84/**99.08**
	CIFAR10	88.41/**91.95**/78.26/78.98/87.94	97.59/**98.27**/95.89/95.10/97.35

Bold indicates sufficient values

1. A LeNet [24] model with two convolutional layers and three fully connected layers. The model is trained for the **MNIST** data set [24] and achieves 99.20% accuracy on the testing set. **MNIST** consists of 60, 000 28 × 28 grayscale images of hand-written digits in the training sets and 10, 000 images in the testing set.
2. A ResNet [15] model for the **CIFAR-10** [21] data set and another ResNet model for the **SVHN** [29] data set, achieving accuracies of 91.65% and 96.12%, respectively. **CIFAR-10** consists of 50, 000 and 10, 000 32 × 32 color images in its training set and testing set, respectively, with each image belonging to one of the ten classes. **SVHN** consists of 73, 257 and 26, 032 colored house numbers from Google Street View images in its training set and testing set, respectively.
3. A VGG [39] model for the **CIFAR-10** data set and another VGG model for the **SVHN** data set, achieving accuracies of 93.47% and 95.56%, respectively.

Outlier Exposure (OE) has been shown as an effective fine-tuning method for improving the performance of existing anomaly detectors [17, 31, 34]. In this work, we also present the experimental results that combines OE and our approach, which are shown in the 'ours + OE' column of Tables 5, 6 and 7. The authors of [17] demonstrate that only 50, 000 samples from auxiliary data set to be used to fine-tune the pre-trained model is enough to improve the performance of existing anomaly detectors. In this work, we use 50, 000 English letters from E-MNIST [7] as the auxiliary data set to fine-tune the LeNet model. For VGG and ResNet models, we use 50, 000 samples from the TinyImageNet data set [9] to perform the fine-tuning. Note that the training data from the auxiliary data set, the anomaly testing data and ID testing data are pairwise disjoint.

Table 6 Comparison of results with OE about the KD+BU, LID, MD and ELO for adversarial data

Model	AD	TNR at 95% TPR ↑	AUROC ↑
		KD+BU+OE/LID+OE/MD+OE/ELO+OE/ours+OE	
LeNet	FGSM	78.21/95.85/86.18/**99.99**/**99.99**	93.91/95.93/97.23/99.93/**99.98**
(MNIST)	BIM-a	51.58/52.72/49.85/43.53/**97.09**	88.31/87.49/91.59/84.01/**99.29**
	BIM-b	28.95/34.98/18.33/**96.99**/96.69	78.28/77.90/72.48/99.10/**99.22**
	JSMA	79.40/81.08/80.64/17.19/**91.84**	95.12/92.89/96.69/58.80/**98.42**
	CW	46.74/37.65/37.62/6.21/**92.46**	84.28/75.22/88.08/49.57/**98.44**
VGG-16	FGSM	81.83/92.85/78.42/89.44/**96.58**	66.79/98.73/96.07/97.99/**99.14**
(CIFAR10)	BIM-a	73.95/12.55/**76.07**/1.34/67.55	78.86/67.37/**96.17**/49.81/95.24
	BIM-b	5.77/8.47/7.03/1.05/**10.63**	47.04/55.96/70.12/55.02/**74.51**
	JSMA	87.92/35.01/**87.66**/5.05/80.88	94.55/69.82/**97.60**/47.95/96.90
	CW	75.53/10.33/**77.29**/4.79/74.75	80.02/67.87/96.27/63.06/**96.36**
ResNet	FGSM	28.32/100.0/7.02/**100.0**/**100.0**	75.40/99.99/80.22/**100.0**/99.79
(CIFAR10)	BIM-a	12.14/11.12/5.78/18.05/**38.84**	63.96/60.58/71.26/74.63/**84.73**
	BIM-b	99.84/66.52/99.96/95.93/**99.98**	99.93/85.00/**99.99**/98.84/**99.99**
	JSMA	32.13/68.30/18.65/83.39/**91.69**	73.45/67.78/89.57/93.36/**97.83**
	CW	13.11/11.23/7.19/13.98/**48.12**	68.51/60.73/78.72/66.74/**89.21**
VGG-16	FGSM	91.72/**99.68**/91.32/95.08/96.83	89.18/**99.56**/97.77/97.69/99.40
(SVHN)	BIM-a	68.60/18.85/68.93/4.17/**75.45**	**94.03**/75.77/93.83/53.17/93.10
	BIM-b	15.26/**57.61**/21.04/8.63/52.43	69.16/88.40/85.54/81.62/**93.30**
	JSMA	80.22/28.44/78.11/4.76/**91.64**	95.10/80.66/95.57/53.04/**97.65**
	CW	22.67/8.04/23.43/4.43/**91.99**	78.16/57.97/76.28/49.18/**97.97**
ResNet	FGSM	61.24/99.96/70.00/**100.0**/99.63	81.94/**99.86**/94.52/99.83/99.72
(SVHN)	BIM-a	48.86/17.20/**59.02**/3.40/31.95	90.86/75.94/**91.51**/43.49/81.22
	BIM-b	99.98/96.71/99.98/52.93/**100.0**	99.99/97.54/**100.0**/92.24/99.97
	JSMA	68.19/31.36/**73.28**/8.00/58.82	94.08/79.76/**95.59**/52.52/92.05
	CW	19.70/6.75/20.83/4.50/**80.93**	80.88/58.54/81.35/48.71/**96.53**

Bold indicates sufficient values

Evaluation Metrics

Given a (binary) anomaly detector, we define true positive (TP) as the number of cases when an input from ID is correctly reported as $isAnomaly(x) = False$, and false negative (FN) as the number of cases when an input from ID is incorrectly reported as $True$, for anomaly. Similarly, true negative (TN) is the number of cases when an anomaly input is correctly reported as $True$, and false positive (FP) is the number of cases when an anomaly is incorrectly reported as $False$, for data from ID. We adopt two commonly used metrics, TNR (True Negative Rate) at 95% TPR (True Positive Rate) and Area Under the Receiver Operating Characteristic curve (AUROC), to evaluate the effectiveness of our method.

Since we have a detector g_i for each class i, all counted values need to be taken weighted average for each class. For example, we have TPR $= \sum_{i=1,...d} \gamma_i \cdot \text{TPR}_i$, where TPR_i is the true positive rate calculated for inputs that are classified as i by the DNN and γ_i is the percentage of sample cases being classified as i.

Existing works mostly focus on detecting either OOD only, or AD only. Therefore, we compare our results on OOD data sets and Noise data with models designed for OOD

Table 7 Comparison of results with OE about the baseline, ODIN and MD for noise data

Model	NS	TNR at 95% TPR ↑	AUROC ↑
		baseline+OE/ODIN+OE/MD+OE/ours+OE	
LeNet	Gaussian	100.0/100.0/100.0/**100.0**	99.99/100.0/99.63/**100.0**
(MNIST)	Uniform	100.0/100.0/100.0/**100.0**	99.95/99.99/99.59/**100.0**
	FoolIm	30.57/51.29/30.44/**100.0**	90.65/93.20/83.78/**99.95**
VGG-16	Gaussian	97.47/99.56/99.97/**100.0**	98.38/98.79/99.42/**100.0**
(CIFAR10)	Uniform	97.82/99.60/99.96/**100.0**	98.49/98.78/99.42/**100.0**
	FoolIm	2.92/5.96/11.14/**98.37**	82.51/84.05/86.42/**99.31**
ResNet	Gaussian	54.79/83.82/31.24/**100.0**	93.58/97.28/91.33/**99.76**
(CIFAR10)	Uniform	54.68/88.33/25.16/**100.0**	93.77/97.62/90.77/**99.73**
	FoolIm	25.51/32.02/32.97/**100.0**	84.59/86.04/85.34/**99.86**
VGG-16	Gaussian	99.19/99.91/99.42/**99.98**	99.68/99.95/99.38/**99.97**
(SVHN)	Uniform	99.58/99.96/99.69/**99.99**	99.83/99.98/99.54/**99.97**
	FoolIm	7.44/24.98/15.68/**68.13**	63.92/69.08/62.91/**96.25**
ResNet	Gaussian	94.31/96.49/83.93/**100.0**	98.51/99.04/96.76/**99.92**
(SVHN)	Uniform	93.47/95.74/81.94/**99.98**	98.43/98.89/96.57/**99.89**
	FoolIm	7.97/14.17/8.81/**99.38**	75.51/78.40/74.01/**99.11**

Bold indicates sufficient values

detection, and compare our results on AD data with models designed for adversarial attack detection, in separate.

5.2 Experiment Results

OOD Detection

We consider several OOD data sets for evaluating the effectiveness of our methods. In particular, **Fashion-MNIST (F-MNIST)** [44] and **Omniglot** [23] are used as OOD for the LeNet model trained with MNIST. For the ResNet model trained with **CIFAR-10**, the OOD sets are **TinyImageNet** [9], **LSUN** [47], **iSUN** [45] and **SVHN** [29]. For the ResNet model trained with **SVHN**, the OOD sets are **TinyImageNet**, **LSUN**, **iSUN** and **CIFAR-10**. The experiments for the two VGG models are treated in the same way as the ResNet models. Note that we do not test MD with *feature ensemble* which uses output from all layers but involves tuning with particular OOD sets. We only apply the version of MD which uses the logit layer of the DNN instead.

The results for OOD detection of these models are presented in Table 2, where the data set enclosed by the brackets next to the model denotes the ID set, e.g., MNIST is the ID for the LeNet model. As shown in the results, our method has the best precision for detection of OOD anomalies in most cases. Even for the one case when ELO is better, the percentage difference is minor.

AD Detection

We compare with the works that are designed for adversarial detection, including KD+BU [11], LID [27] and MD [25]. We include the ELO method because this is the most related work to ours. The adversarial samples used in the experiment are generated by various well-known methods, including FGSM [13], BIM [22], JSMA [33] and CW [5]. For the BIM attack, we consider two scenarios: BIM-a, which stops iterating as soon as the attack is successful
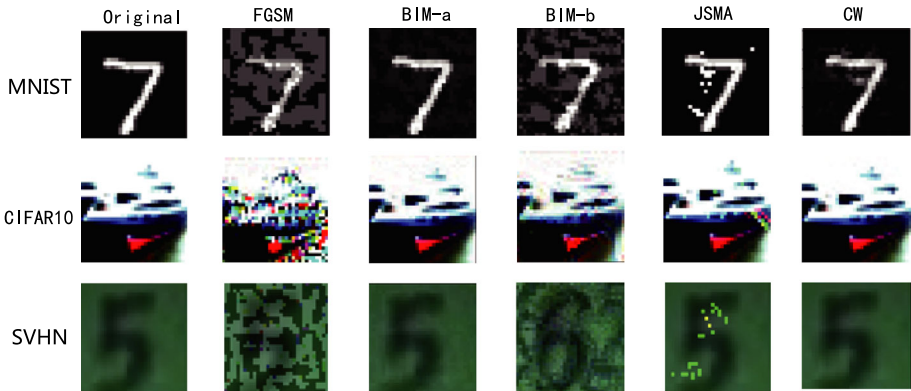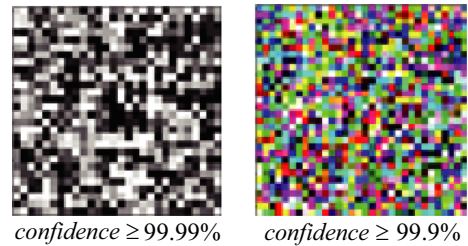
Fig. 8 Some normal images and AD anomaly images for MNIST, CIFAR10 and SVHN

Fig. 9 NS-II anomaly inputs

confidence ≥ 99.99%   *confidence ≥ 99.9%*

('at the decision boundary'), and BIM-b, which attacks for a fixed number of iterations that is well beyond the average misclassification point ('eyond the decision boundary'. Some normal inputs and adversarial inputs are displayed in Fig. 8. Since both KD+BU and LID require training with adversarial inputs, in this experiment, both detectors are trained with FGSM.

The results in Table 3 have shown that our method produces the best precision in about half of the cases regarding the AUROC values. For the rest cases where other methods have better precision, our scores are not much behind except for the two BIM-a cases for the SVHN data set.

NS Detection In this experiment we have prepared three types of noise images.

1. The **Gaussian noise (NS-I)** set consists of 10, 000 images of which every pixel is sampled from random Gaussian distribution with the mean $\mu = 0.5$ and the variance $\sigma = 1$, clipped to [0,1];
2. The **Uniform noise (NS-I)** set consists of 10, 000 images of which every pixel is sampled from a random uniform distribution between [0, 1];
3. The **Fooling images (NS-II)** are generated by evolving meaningless images in order to mislead a DNN to output classes in ID with high confidence. We adopt the algorithm from Section 3 of [30]. The Fooling Image sets feeding to the LeNet model consists of 10, 000 28×28 images (confidence≥ 99.99%). The Fooling Image sets feeding to ResNet and VGG models both consist of 10, 000 32×32 images (confidence≥ 99.9%). Those images are totally unrecognizable to human eyes. Two examples from NS-II are displayed in Fig. 9.

The results shown in Table 4 indicate that only our method achieves near 100% precision regarding detection of the given Noise inputs. Note that some methods, including baseline [16] and ODIN [26], which take the classification probability of an input to discriminate whether this image is abnormal, fail to detect anomalies from NS-II and in most cases classify them as ID with high confidence. In particular, these methods mostly report 0 in the column of TNR at 95% TPR. One possible explanation is that inputs from NS-II are quite different from any known anomaly distributions that have been considered by these approaches, therefore, those images are relatively hard for them to detect.

5.3 Comparison of Results with Outlier Exposure

Since OE is designed and proved to be effective for enhancing Out-of-Distribution (OOD) detection tasks [17], we need to check if it is also useful in our approach by using OE to fine-tune the models. However, in our experiment, it is inconclusive whether OE is an effective way to enhance precision when the target anomaly includes not only OOD, but also AD. First, out of the 18 OOD detection benchmarks, OE has improved the results of our method for 11 benchmarks regarding the AUROC metric. For adversarial (AD) detection tasks, our method combined with OE has produced equal or better performance in 15/25 cases. Nevertheless, performance of our method has been enhanced in a few benchmarks, e.g., for the SVHN data set, the results of the two BIM-a are significantly improved (11.74% gain in VGG model and 3, 89% gain in ResNet model). For Noise (NS) detection, the performance of our method with OE is also improved in most cases (equal or better in 13/15 cases), although the improvements are mostly minor.

Regarding the other methods, similar improvement in performance has been observed for basline, ODIN and MD regarding OOD detection tasks. For AD detection, the improvement of performance for KD+BU, LID, MD, ELO are in general insignificant (some cases even suffering performance degradation). The effect of OE for NS detection is in general positive for all the tested methods.

5.4 Comparison of Results with AEs

Another class of approach for anomaly detection, which is substantially different from the main methodology presented in this paper, applies autoencoder (AE) based structures, such as AE [3] and VAE [20], to measure the difference between the reconstruction loss of normal and anomalous examples. In general, AEs are used to detect anomaly data in an unsupervised manner. Some more recent works include MemAE [12], which augments the autoencoder with a memory module and targets OOD, and AE&KL [42], which measures the difference between the outputs of the DNN with original data and reconstructed data as inputs and targets AD detection mostly. To achieve a better coverage over the anomaly detection works in the literature, we compare the performance of our method and a few recent AE-based methods, and demonstrate that our method delivers more stable results when facing OOD and AD.

In the literature, the AEs are often applied in the following two ways for anomaly detection tasks. First, an AE is used to reconstruct a test sample and the reconstruction error can be calculated and directly used for detection [6, 12, 14, 49]. Second, an AE can be used to learn a low-dimensional representation of the input data in its latent space, then distance-based metrics are applied to measure the difference between a test example and the ID dataset [10, 37, 42, 50]. The reconstruction error or distance metrics are used as an anomaly score and

Table 8 Comparison of results with AE, VAE, AE&KL and MemAE for OOD data

Model	OOD	TNR at 95% TPR ↑	AUROC ↑
		AE / VAE / AE&KL / MemAE / ours	
LeNet	F-MNIST	**100.0** / 99.64 / 99.60 / 82.20 / 99.72	**100.0** / 99.89 / 99.82 / 93.89 / 99.83
(MNIST)	Omniglot	100.0 / 100.0 / 99.86 / 100.0 / **100.0**	**100.0** / 100.0 / 99.83 / 100.0 / 99.99
VGG-16	TinyIm	**88.17** / 49.86 / 57.03 / 32.90 / 85.61	96.84 / 83.17 / 92.81 / 78.03 / **97.54**
(CIFAR10)	LSUN	**95.79** / 65.44 / 65.10 / 38.12 / 92.81	98.96 / 90.90 / 94.63 / 81.98 / **98.98**
	iSUN	91.44 / 53.39 / 65.55 / 30.15 / **92.64**	97.64 / 84.81 / 94.60/ 75.33 / **98.70**
	SVHN	0.01 / 1.12 / 47.82 / 1.62 / **68.07**	2.62 / 13.47 / 92.54 / 28.15 / **95.66**
ResNet	TinyIm	88.53 / 49.65 / 48.77 / 29.03 / **94.24**	96.88 / 83.10 / 90.62 / 78.69 / **98.61**
(CIFAR10)	LSUN	95.99 / 64.78 / 67.79 / 31.18 / **97.34**	99.00 / 90.79 / 94.80 / 83.17 / **99.25**
	iSUN	91.73 / 52.77 / 64.21 / 24.39 / **95.34**	97.69 / 84.55 / 94.17 / 78.21 / **99.01**
	SVHN	0.01 / 1.25 / 67.77 / 1.35 / **86.29**	2.33 / 13.95 / 93.86 / 23.74 / **97.00**
VGG-16	TinyIm	**99.94** / 94.79 / 90.04 / 43.93 / 93.97	**99.97** / 98.77 / 98.17 / 90.21 / 98.87
(SVHN)	LSUN	**99.98** / 98.30 / 89.43 / 47.73 / 94.80	**99.99** / 99.49 / 97.97 / 92.16 / 99.05
	iSUN	**99.94** / 95.94 / 92.00 / 44.15 / 95.38	**99.97** / 99.09 / 98.42 / 90.08 / 99.08
	CIFAR10	**97.91** / 79.46 / 90.45 / 33.00 / 87.37	**99.42** / 96.13 / 98.08 / 83.47 / 97.83
ResNet	TinyIm	**99.93** / 94.87 / 85.78 / 65.45 / 95.73	**99.97** / 98.83 / 97.21 / 92.14 / 99.01
(SVHN)	LSUN	**99.97** / 98.50 / 83.46 / 71.54 / 96.58	**99.99** / 99.51 / 96.76 / 94.30 / 99.18
	iSUN	**99.93** / 96.17 / 86.76 / 63.26 / 96.64	**99.97** / 99.11 / 97.36 / 92.25 / 99.21
	CIFAR10	**98.15** / 80.18 / 85.63 / 43.24 / 87.60	**99.46** / 96.20 / 97.13 / 85.39 / 97.41

Bold indicates sufficient values

compared with a given threshold, and the samples above the threshold are considered as anomalous.

We compare our proposed method with a few standard or recent anomaly detectors using AE models, including autoecoder (AE) [3], variational autoencoder (VAE) [20], memory-augmented autoencoder (MemAE) [12] and autoecoder combined with KL divergence (AE&KL) [42]. The AE [3] and VAE [20] were widely used for anomaly detection [2], while these models may also reconstruct well anomaly data, causing unsatisfactory detection performance [35, 48]. The MemAE [12] is a more recent work that augments the autoencoder with a memory module which is used to represent the prototypical elements of the normal data, so that the reconstruction is obtained from only a restricted set of memory records. The above three works belong to the first category, i.e., they use the reconstruction error as the anomaly detection metrics. The data sample with reconstruction error above a given threshold is anomalous. The AE&KL defines the KL divergence as distance measure between the output distributions of the classified model on input data and reconstructed data. The AE&KL belongs to the second category.

The detailed implementation and parameter settings are given in our public code. The experimental results are given in Tables 8 and 9. The results show that AE and VAE perform better than our model in a few experiments, e.g., on detecting OOD on the SVHN dataset. However, it does not perform stably, because it also reconstruct well on some anomaly data, e.g., the BIM-a and CW for CIFAR-10 (ResNet) and SVHN (VGG/ResNet). For MemAE, we train 10 anomaly detectors according to 10 subclasses, which is the same as the original work [12]. The difference is that the original work selects one class as ID, and the remaining classes from the same dataset are considered to be anomalous,

Table 9 Comparison of our approach with the AE, VAE, AE&KL and MemAE for adversarial data

Model	AD	TNR at 95% TPR ↑	AUROC ↑
		AE / VAE / AE&KL / MemAE / ours	
LeNet	FGSM	100.0 / 100.0 / 91.79 / 99.66 / **100.0**	**100.0** / 100.0 / 98.21 / 99.32 / 99.98
(MNIST)	BIM-a	**97.15** / 30.87 / 82.52 / 34.10 / 95.57	**99.40** / 82.53 / 97.03/ 78.44 / 99.04
	BIM-b	**100.0** / 100.0 / 1.89 / 67.81 / 99.32	**100.0** / 100.0 / 37.34 / 94.78 / 99.64
	JSMA	**98.91** / 93.37 / 91.72 / 77.42 / 89.70	**99.76** / 98.80 / 98.44 / 94.94 / 98.16
	CW	70.11 / 9.16 / 68.34 / 35.01 / **79.73**	94.08 / 65.19 / 95.08 / 78.07 / **97.12**
VGG-16	FGSM	77.12 / 76.73 / 39.13 / 1.05 / **96.74**	**99.98** / 99.78 / 94.69 / 20.75 / 99.22
(CIFAR10)	BIM-a	15.01 / 4.91 / 40.11 / 0.14 / **63.57**	73.31 / 54.27 / **95.72** / 14.92 / 94.10
	BIM-b	**23.02** / 4.27 / 0.36 / 0.15 / 15.80	**84.63** / 54.33 / 33.95 / 14.16 / 81.53
	JSMA	17.77 / 5.98 / 64.16 / 11.29 / **83.60**	76.57 / 54.10 / 93.50 / 59.24 / **97.13**
	CW	63.98 / 64.35 / 49.59 / 0.15 / **81.18**	95.82 / 91.79 / 96.14 / 13.62 / **96.53**
ResNet	FGSM	60.14 / 60.53 / 37.30 / 44.60 / **99.96**	99.98 / 98.43 / 93.24 / 83.76 / 99.81
(CIFAR10)	BIM-a	8.75 / 4.71 / 30.10 / 0.14 / **35.65**	60.40 / 43.51 / **86.48** / 18.38 / 83.67
	BIM-b	34.96 / 0.97 / 0.0 / 0.15 / **99.98**	91.65 / 34.16 / 0.01 / 18.47 / **99.99**
	JSMA	60.77 / 5.81 / 56.70 / 40.22 / **91.62**	97.61 / 62.45 / 93.95 / 83.84 / **97.74**
	CW	20.96 / 19.93 / 33.34 / 0.12 / **37.98**	76.00 / 63.27 / **92.80** / 18.23 / 86.84
VGG-16	FGSM	**100.0** / 82.55 / 68.26 / 11.50 / 97.69	**100.0** / 98.91 / 97.12 / 80.05 / 99.46
(SVHN)	BIM-a	11.44 / 5.69 / 40.12 / 5.77 / **40.81**	69.55 / 53.20 / **89.35** / 52.49 / 81.36
	BIM-b	**100.0** / 16.51 / 0.0 / 6.44 / 98.84	**99.93** / 87.01 / 1.98 / 62.41 / 99.17
	JSMA	**83.48** / 14.10 / 71.47 / 7.38 / 76.76	**98.91** / 73.12 / 97.92 / 59.09 / 94.72
	CW	5.60 / 4.98 / 37.36 / 5.58 / **66.88**	52.37 / 50.28 / 94.46 / 51.75 / **94.54**
ResNet	FGSM	**100.0** / 91.25 / 74.89 / 14.23 / 99.76	**100.0** / 99.01 / 97.62 / 81.93 / 99.80
(SVHN)	BIM-a	11.66 / 5.29 / **53.08** / 5.64 / 24.53	75.50 / 52.80 / **89.40** / 52.93 / 77.33
	BIM-b	100.0 / 15.90 / 0.0 / 6.49 / **100.0**	99.97 / 86.34 / 0.0 / 63.26 / **99.98**
	JSMA	**89.84** / 16.73 / 70.12 / 7.44 / 51.04	**99.29** / 77.23/ 97.04 / 59.92 / 90.81
	CW	5.25 / 4.96 / 38.31 / 5.66 / **54.53**	51.77 / 50.17 / 91.82 / 52.07 / **92.36**

Bold indicates sufficient values

while in our approach, ID and anomalous data (OOD and AD) are different datasets. In addition, our approach detects anomalies in given classes, meaning that for each input, we apply the target DNN to return a class, based on which we select the corresponding MemAE-based detector for testing, whether the input is anomalous. The results of MemAE are significantly inferior to the results of our method and other methods for OOD and AD, as shown in Tables 8 and 9. The AE&KL demonstrates better performance than ours in some cases for AD, e.g., the results of the BIM-a for SVHN. However, the results for BIM-b are not as good. Besides, the results for OOD are significantly inferior to our results. In summary, the performance of the above AE-based anomaly detectors are unstable and cannot be used as a general method to tackle both OOD and AD.

5.5 Discussion on Preprocessing

As shown in Table 1, both ODIN and MD apply *input pre-processing* to improve their precision. To illustrate the performance of our work, we further compare our results with their

Table 10 The results of ODIN and MD for pre-processing

Model	OOD	TNR at 95% TPR ↑	AUROC ↑
		ODIN / MD / ours	
VGG-16	TinyIm	73.76 / 63.64 / **85.61**	95.66 / 92.14 / **97.54**
(CIFAR10)	LSUN	83.80 / 73.74 / **92.81**	97.36 / 94.87 / **98.98**
	iSUN	84.06 / 72.56 / **92.64**	97.27 / 94.51 / **98.70**
	SVHN	**84.06** / 35.61 / 68.07	**96.54** / 84.39 / 95.66
ResNet	TinyIm	65.35 / 41.32 / **94.24**	93.57 / 84.67 / **98.61**
(CIFAR10)	LSUN	85.40 / 58.12 / **97.34**	97.42 / 91.58 / **99.25**
	iSUN	82.33 / 56.00 / **95.34**	96.98 / 90.65 / **99.01**
	SVHN	67.27 / 60.26 / **86.29**	93.80 / 87.44 / **97.00**

Bold indicates sufficient values

Table 11 Comparison of our approach with the KD+BU, LID for the known adversarial data

Model	AD	TNR at 95% TPR ↑	AUROC ↑
		KD+BU/LID/ours	
VGG-16	FGSM	77.35/92.81 /**96.74**	87.66/98.11/**99.22**
(CIFAR10)	BIM-a	86.38/**82.79**/63.57	83.74/84.15/**94.10**
	BIM-b	0.05/**45.40**/15.80	49.11/**86.86**/81.53
	JSMA	**97.23**/94.81/ 83.60	**97.29**/97.15/97.13
	CW	87.01/**95.94**/81.18	85.80/88.75/ **96.53**
ResNet	FGSM	38.93/99.95/**99.96**	75.70/99.98/**99.81**
(CIFAR10)	BIM-a	17.48/**65.25**/35.65	65.77/83.52/**83.67**
	BIM-b	99.89/99.89 /**99.98**	99.93/99.93/**99.99**
	JSMA	60.0/81.13/**91.62**	76.38/90.69/**97.74**
	CW	26.38/**49.92**/37.98	75.75/72.55/**86.84**

Bold indicates sufficient values

best performance i.e., ODIN and MD with pre-processing, using VGG and ResNet models on the CIFAR-10 data set. The results (see Table 10) confirm that the performance of ODIN and MD with pre-processing is obviously better than those without pre-processing (Table 2), and demonstrate that our results are better than both ODIN and MD with pre-processing. Similarly, for AD detection, knowing the adversarial attack strategy, the performance of LID can be significantly improved and the performance of KD+BU can also be improved to some extent (excluding BIM-b for VGG model) as shown in Table 11. More importantly, we show that our results outperform both KD+BU and LID with known adversarial samples in majority cases, when applied to VGG and ResNet models on the CIFAR-10 data set (see Table 11).

6 Conclusion and Future Work

To enhance the applicability of DNN input anomaly detection in real-world tasks, we have proposed a novel approach that is able to detect all three types of anomalies, namely Out-of-Distribution (OOD) data, Adversarial (AD) data and Noise (NS) data. By combining the early and late layers of pre-trained DNN models, and deepening to a fine-grained level of each

sub-class, our approach generally outperforms the state-of-the-art approaches for detection of all aforementioned anomaly types, to the best of our knowledge, which has been evidenced by the experiments.

One limitation is that this work and other anomaly detection works focus only on image classification. As the application domains of DNN is expanding fast, it is interesting and necessary to explore whether the existing methodology can be adopted to other applications beyond image processing, such as speech recognition, natural language processing, and intrusion detection with network traffic monitoring.

References

1. Abdelzad V, Czarnecki K, Salay R, et al. (2019) Detecting out-of-distribution inputs in deep neural networks using an early-layer output arXiv:1910.10307
2. An J, Cho S (2015) Variational autoencoder based anomaly detection using reconstruction probability. Special Lect IE 2(1):1–18
3. Bengio Y, Lamblin P, Popovici D et al (2006) Greedy layer-wise training of deep networks. In: Schölkopf B, Platt J, Hoffman T (eds) Advances in neural information processing systems (NIPS2006). MIT Press, USA, pp 153–160
4. Carlini N, Wagner D (2017) Adversarial examples are not easily detected: bypassing ten detection methods In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (AISec-2017), p 3-14
5. Carlini N, Wagner D (2017) Towards evaluating the robustness of neural networks In: Proceedings of 2017 IEEE Symposium on Security and Privacy (IEEE S& P-2017), pp 39–57
6. Chen J, Sathe S, Aggarwal C, et al. (2017) Outlier detection with autoencoder ensembles In: Proceedings of the 2017 SIAM international conference on data mining, SIAM, pp 90–98
7. Cohen G, Afshar S, Tapson J, et al. (2017) Emnist: extending mnist to handwritten letters In: Proceedings of 2017 International Joint Conference on Neural Networks (IJCNN-2017), pp 2921–2926
8. Cohen G, Sapiro G, Giryes R (2020) Detecting adversarial samples using influence functions and nearest neighbors In: Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR-2020), pp 14,441–14,450
9. Deng J, Dong W, Socher R, et al (2009) Imagenet: a large-scale hierarchical image database In: Proceedings of 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR-2009), pp 248–255
10. Denouden T, Salay R, Czarnecki K, et al (2018) Improving reconstruction autoencoder out-of-distribution detection with mahalanobis distance arXiv preprint arXiv:1812.02765
11. Feinman R, Curtin RR, Shintre S, et al (2017) Detecting adversarial samples from artifacts arXiv:1703.00410
12. Gong D, Liu L, Le V, et al (2019) Memorizing normality to detect anomaly: memory-augmented deep autoencoder for unsupervised anomaly detection In: IEEE International Conference on Computer Vision (ICCV-2019)
13. Goodfellow IJ, Shlens J, Szegedy C (2015) Explaining and harnessing adversarial examples In: Proceedings of International Conference on Learning Representations (ICLR-2015)
14. Goodge A, Hooi B, Ng SK, et al (2020) Robustness of autoencoders for anomaly detection under adversarial impact In: Proceedings of the 29 International Joint Conference on Artificial Intelligence (IJCAI-2020), pp 1244–1250
15. He K, Zhang X, Ren S, et al (2016) Deep residual learning for image recognition In: Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR-2016), pp 770–778
16. Hendrycks D, Gimpel K (2017) A baseline for detecting misclassified and out-of-distribution examples in neural networks In: Proceedings of International Conference on Learning Representations (ICLR-2017)
17. Hendrycks D, Mazeika M, Dietterich T (2019) Deep anomaly detection with outlier exposure In: Proceedings of International Conference on Learning Representations (ICLR-2019)
18. Hsu YC, Shen Y, Jin H, et al (2020) Generalized ODIN: detecting out-of-distribution image without learning from out-of-distribution data In: Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR-2020), pp 10,948–10,957
19. Jiang H, Kim B, Guan M, et al (2018) To trust or not to trust a classifier In: Proceedings of the 32nd International Conference on Neural Information Processing SystemsDecember (NeurIPS-2018), pp 5546–5557
20. Kingma DP, Welling M (2013) Auto-encoding variational bayes arXiv:1312.6114 [stat.ML]

21. Krizhevsky A, Hinton G (2009) Learning multiple layers of features from tiny images University of Toronto, Tech rep
22. Kurakin A, Goodfellow IJ, Bengio S (2017) Adversarial examples in the physical world In: Proceedings of International Conference on Learning Representations(ICLR-2017 Workshop)
23. Lake BM, Salakhutdinov R, Tenenbaum JB (2015) Human-level concept learning through probabilistic program induction. Science 350(6266):1332–1338
24. Lecun Y, Bottou L, Bengio Y et al (1998) Gradient-based learning applied to document recognition Proceedings of the IEEE 86(11):2278–2324
25. Lee K, Lee K, Lee H, et al (2018) A simple unified framework for detecting out-of-distribution samples and adversarial attacks In: Proceedings of the 32nd International Conference on Neural Information Processing SystemsDecember (NeurIPS-2018), p 7167-7177
26. Liang S, Li Y, Srikant R (2018) Enhancing the reliability of out-of-distribution image detection in neural networks In: Proceedings of International Conference on Learning Representations (ICLR-2018)
27. Ma X, Li B, Wang Y, et al (2018) Characterizing adversarial subspaces using local intrinsic dimensionality In: Proceedings of International Conference on Learning Representations (ICLR-2018)
28. McInnes L, Healy J, Saul N et al (2018) Umap: uniform manifold approximation and projection. J Open Sour Softw 3(29):861
29. Netzer Y, Wang T, Coates A, et al (2011) Reading digits in natural images with unsupervised feature learning In: Proceedings of NIPS Workshop on Deep Learning and Unsupervised Feature Learning (NIPS-2011 Workshop)
30. Nguyen A, Yosinski J, Clune J (2015) Deep neural networks are easily fooled: high confidence predictions for unrecognizable images In: Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR-2015), pp 427–436
31. Papadopoulos AA, Rajati MR, Shaikh N et al (2021) Outlier exposure with confidence control for out-of-distribution detection. Neurocomputing 441:138–150
32. Papernot N, McDaniel PD (2018) Deep k-nearest neighbors: towards confident, interpretable and robust deep learning CoRR abs/1803.04765 http://arxiv.org/abs/1803.04765, arXiv:1803.04765
33. Papernot N, McDaniel PD, Jha S, et al (2016) The limitations of deep learning in adversarial settings In: Proceedings of 2016 IEEE European Symposium on Security and Privacy (EuroS&P-2016), pp 372–387
34. Reiss T, Cohen N, Bergman L, et al. (2020) Panda–adapting pretrained features for anomaly detection arXiv preprint arXiv:2010.05903
35. Ren J, Liu PJ, Fertig E, et al. (2019) Likelihood ratios for out-of-distribution detection Adv Neural Information Process Syst (NIPS2019) 32:14,707–14,718
36. Research AM (2020) Neural network market to reach $38.71 billion, globally, by 2023, says allied market research Retrieved 22 Dec 2020
37. Sarafijanovic-Djukic N, Davis J (2019) Fast distance-based anomaly detection in images using an inception-like autoencoder In: International Conference on Discovery Science, Springer, pp 493–508
38. Schölkopf B, Platt JC, Shawe-Taylor JC et al (2001) Estimating the support of a high-dimensional distribution. Neural Comput 13(7):1443–1471
39. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition In: Proceedings of International Conference on Learning Representations (ICLR-2020)
40. Tax D (2001) One-class classification: concept-learning in the absence of counter-examples PhD thesis, Delft University of Technology
41. Tax DMJ, Duin RPW (1999) Support vector domain description. Pattern Recognit Lett 20(11–13):1191–1199
42. Vacanti G, Van Looveren A (2020) Adversarial detection and correction by matching prediction distributions arXiv preprint arXiv:2002.09364
43. Vapnik V (2000) The Nature of Statistical Learning Theory. In series of Statistics for Engineering and Information Science
44. Xiao H, Rasul K, Vollgraf R (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms arXiv:1708.07747
45. Xu P, Ehinger KA, Zhang Y, et al. (2015) Turkergaze: crowdsourcing saliency with webcam based eye tracking arXiv:1504.06755
46. Xu W, Evans D, Qi Y (2018) Feature squeezing: detecting adversarial examples in deep neural networks In: Proceedings of The Network and Distributed System Security (NDSS-2018)
47. Yu F, Zhang Y, Song S, et al (2015) LSUN: construction of a large-scale image dataset using deep learning with humans in the loop arXiv:1506.03365
48. Zhang S, Pan C, Song L, et al. (2021) Label-assisted memory autoencoder for unsupervised out-of-distribution detection In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, pp 795–810

49. Zhou C, Paffenroth RC (2017) Anomaly detection with robust deep autoencoders In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp 665–674
50. Zong B, Song Q, Min MR, et al. (2018) Deep autoencoding gaussian mixture model for unsupervised anomaly detection In: International conference on learning representations

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.