

NOISE-GUIDED UNSUPERVISED OUTLIER DETECTION

APPENDIX

Anonymous authors

Paper under double-blind review

A THEORETICAL ANALYSIS

For the binary classification problem, we use the binary cross entropy loss to optimize the classifier $f(x)$ by minimizing the loss \mathcal{L}_f^n :

$$\mathcal{L}_f^n = -\left(\sum_{i=0}^{|X|} \log(1 - f(x_i)) + \sum_{k=0}^{|X^-|} \log f(x_k)\right). \quad (1)$$

When we have arbitrarily large samples, the weak law of large numbers shows that the objective function \mathcal{L}_f^n converges in probability to \mathcal{L}_f :

$$\mathcal{L}_f = -\left(E_X(\log(1 - f(x))) + E_{X^-}(\log f(x^-))\right). \quad (2)$$

Let $p(x, y) = p(y)p(x|y)$ be an expanded generative model for x defined as:

$$\begin{aligned} x &\sim a(x) \quad \text{if } y = 0, \\ x &\sim b(x) \quad \text{if } y = 1 \end{aligned} \quad (3)$$

When the number of positive and negative samples is equal, we can express the loss function as:

$$\mathcal{L}_f = -\int (\log(1 - f)a(x) + \log(f)b(x))dx. \quad (4)$$

$$\frac{\partial \mathcal{L}_f}{\partial f} = -\int \left(\frac{1}{f-1}a(x) + \frac{1}{f}b(x)\right)dx. \quad (5)$$

When the derivative is constantly zero, the objective function achieves an extremum. By doing this, we can obtain an optimized classifier:

$$f^* \approx \frac{b(x)}{a(x) + b(x)} = p(y = 1|x) \quad (6)$$

$f(x)$ is the output of the classifier with input x and is the predicted anomaly score of the sample x . We can obtain the optimal classifier $f^*(x) \approx p(y = 1|x)$ after minimizing the loss \mathcal{L}_f . The proof above referenced the counterparts from (Gutmann & Hyvärinen, 2012).

To distinguish inliers and outliers with limited samples, two restrictions are placed, one assumption on the datasets and another on the optimizer. We then provide a simplified proof of the correctness of NOD.

Assumption 1. [Distribution assumption] *Outliers are sparser distributed than inliers and should be sufficiently distant from any inlier.*

Due to the highly unbalanced nature of the sample, we assume that outliers are sparser distributed than inliers and nonoverlapping with inliers. Without this assumption, it would be very hard to differentiate between inliers and outliers. This assumption has been adopted in density-based studies. However, the difficulty lies in how to effectively and efficiently estimate the density of high-dimensional data, due to the "curse of dimensionality". Many UOD calibrate the anomaly score based on the localized distance/density estimation to reduce computation cost. It is difficult for them to use samples beyond their scope. Fig. 1(a) of a toy sample shows the limitations of kNN in the localized calculation.

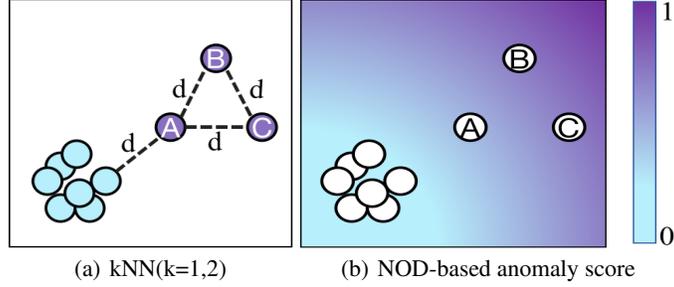


Figure 1: Anomaly score with kNN and NOD on the 2D dataset. Sample A is close to the normal cluster and with lower score than B,C. However, kNN(k=1,2) assign same score to A,B and C.

Certainly, the sparse assumption may not be valid for outliers that are clustered together. However, we argue that we can only effectively address these small clusters with domain-specific knowledge. Section 4.3 in paper contains a discussion on the effects of clustered outliers.

This *Smoothness Prior* (Rosca et al., 2020) specifies that the changing rate of $f(x)$ across the whole value space is below a certain threshold and has been widely used in designing many optimizers, especially those optimizers used in DNNs to estimate a smooth function, e.g., Adam (Kingma & Ba, 2014) and SGD (Bottou & Bousquet, 2007).

Lemma 1. When the value space is limited, using a limited amount of uniform noise, it is ensured that $\rho(x_i) > \rho(x_k) > \rho(x_j)$, where $x_i \in X_n, x_j \in X_o, x_k \in X^-$, and $\rho(\cdot)$ is the density function.

Proof. For $\forall x_{j1}, x_{j2} \in X_o, \forall x_{i1}, x_{i2} \in X_n$ we have $d(x_{j1}, x_{j2}) > 4\sqrt{\dim}d(x_{i1}, x_{i2})$ where \dim is the space dimension for the dataset and the 4 is a scaling factor. We let $D = \max_{i1, i2} d(x_{i1}, x_{i2})$, S be the dataset space, and $\rho(\cdot)$ be the density function. $\rho(x) = \max_y \frac{C(N(x, d(x, y)), x)}{Sqr(N(x, d(x, y)))}$, where x, y come from the same dataset, $N(x, d_x) = \{z | d(x, z) \leq d_x, z \in S\}$, $C(N, x)$ means the number of the data which has the same tag as x and is in the subset N , $Sqr(N)$ means the volume of the subset N . Construct the noise following a uniform distribution, in which the distance between two adjacent points is $4D$; we have: $\min_{j1, k1} d(x_{j1}, x_{k1}) < \frac{\sqrt{\dim}}{2} * 4D < 4\sqrt{\dim}D < \min_{j1, j2} d(x_{j1}, x_{j2})$,

where $x_{j1}, x_{j2} \in X_o, x_{k1} \in X^-$. This indicates that noise is distributed near the outlier instead of the outlier and $\forall x_j \in X_o, \forall x_k \in X^-, \rho(x_k) > \rho(x_j)$. Given that $\max_{x \in S} \min_{x_{k1} \in X^-} d(x, x_{k1}) = 2\sqrt{\dim}D$, we can generate the uniform noise data to guarantee $\exists d(x_{i1}, x_{k1}) \geq 2D$. Then:

$$\min_{i1, k1} d(x_{i1}, x_{k1}) > 2D - \max_{i1, i2} d(x_{i1}, x_{i2}) = \max_{i1, i2} d(x_{i1}, x_{i2}).$$

It means that $\forall x_i \in X_n, \forall x_k \in X^-, \rho(x_k) < \rho(x_i)$. Thus, $\rho(x_j) < \rho(x_k) < \rho(x_i)$. In other words, it is always possible to generate a uniform noise that has a density between that of the inliers and outliers.

Lemma 2. Let $D = \max_j (\min_k d(x_j, x_k))$, where $x_j \in X_o, x_k \in X^-$. There exists an optimized $f^*(x)$ with respect to Equ. 1 that satisfy $\forall x_j \in X_o, f^*(x_j) \geq 1 - MD, MD < 1$.

This lemma shows that the learned optimized function always gives the outlier an anomaly score bigger than a certain positive value.

Proof.[proof by contradiction] If $\exists x_j \in X_o$, s.t. $f(x_j) < 1 - MD$, with smoothness prior (i.e., x_j, x_k are in a subspace with $d(x_j, x_k) < \epsilon$, thus, $f(x_j) \rightarrow f(x_k)$), $\exists x_k \in X^-, f(x_k) < 1 - MD + Md(x_j, x_{k^*}) < 1$. x_{k^*} is the closest of $x_k \in X^-$ to x_j . Thus, there is an optimal classification value $f^*(x)$, so that $f(x_k) \leq f^*(x_k), \forall x_k \in X^-$, and we further define $f^*(x)$ as:

$$f^*(x) = \begin{cases} 1, & \forall x \in X^-, \\ f(x), & \forall x \in X_n, \\ 1 - MD, & \forall x \in X_o. \end{cases} \quad (7)$$

Here, $D_x = \min_k d(x_k, x)$, $x_k \in X^-$. According to Equ. 1, we define C as a sample set that belongs to the same subspace as x_j and $\forall x \in C, x \in X^-$. Let $g(x) = -(\log(1 - (x - \delta)) + \log(x))$, $0 < \delta < 1$, because $g(1) < g(x)$, $\forall \delta \leq x < 1$. With Assumption 1 and Lemma 1, one outlier has little effect on another outlier. Thus, we can only care about one outlier and noise samples around that outlier. We use $\mathcal{L}_{f(x_j)}$ to represent the loss function, where x_j is the outlier:

$$\begin{aligned} \mathcal{L}_{f(x_j)} &= -\left(\sum_k^{|C|} \log f(x_k) + \log(1 - f(x_j))\right) \\ &\geq -\left(\log(1 - f(x_j)) + \log f(x_{k^*}) + \sum_k^{|C| \setminus \{x_{k^*}\}} \log f^*(x_k)\right) \\ &\geq -\left(\log(1 - f^*(x_j)) + \log f^*(x_{k^*}) + \sum_k^{|C| \setminus \{x_{k^*}\}} \log f^*(x_k)\right) \\ &= \mathcal{L}_{f^*(x_j)}, \end{aligned} \tag{8}$$

where $f^*(x_k) = 1$, $f^*(x_j) = 1 - \min_k Md(x_j, x_k)$, $x_k \in X^-$, $x_j \in X_o$. Therefore, if there exists $\exists x_j \in X_o$, s.t. $f(x_j) < 1 - MD$, it is theoretically possible to find $f^*(x_j)$ that minimizes the loss, which contradicts the fact that $f(x)$ is optimal. Thus Lemma 2 holds true.

Theorem 1. Each predicted value of the outlier is higher than each predicted value of the inlier. $\forall x_i \in X_n, x_j \in X_o$, it holds that $f^*(x_j) > \lambda > f^*(x_i)$, where λ is a boundary value.

Proof. Due to the high density of inliers, $\forall x_i \in X_n$, when $\rho(x_i) \rightarrow +\infty$, we have $f^*(x_i) \rightarrow 0$. Thus, there exists a density value ρ_0 , s.t. $\forall x_i \in X_n$, we have $f^*(x_i) < \tau$. According to Lemma 2, it is possible to learn a classifier $f^*(\cdot)$ that satisfies the following conditions: $\forall x_i \in X_n, x_j \in X_o, f^*(x_i) < \tau \leq \lambda \leq 1 - MD \leq f^*(x_j)$. For instance, when $M < \frac{1}{2D}$, $\tau = 0.5$. Therefore, Theorem 1 holds.

This theorem establishes that the anomaly scores of outliers are higher than those of inliers. If we have an outlier ratio, the classifier $f^*(x)$ can distinguish between X_n and X_o . Fig. 1(b) shows the anomaly score distribution in the toy example. It clearly shows that NOD can effectively balance the impact of both local and remote samples with the support of uniform noise. The anomaly scores span the entire value space and exhibit a gradual increase as the points move farther from the inlier center. Therefore, NOD can identify the anomaly degree of samples A, B, and C.

B SUMMARY OF 22 REAL-WORLD DATASETS

Table 1 summarizes 22 real-world datasets used for evaluating UOD. (mat) represents dataset from ODDS¹ and (arff) from DAMI². These datasets are highly representative in terms of diversity in feature dimensions, data volume, and anomaly proportions. The following experiments are the average results obtained from 20 independent experiments on these 22 datasets.

C PERFORMANCE ON 22 DATASETS USING 22 OUTLIER DETECTORS

NOD is compared with 21 other outlier detectors, including classical methods: kNN(Ramaswamy et al., 2000), LOF(Breunig et al., 2000), HBOS(Goldstein & Dengel, 2012), OC-SVM (Schölkopf et al., 2001), COPOD(Li et al., 2020), ECOD(Li et al., 2022), IForest(Liu et al., 2008), SUOD(Zhao et al., 2021), LSCP(Zhao et al., 2019a) and DNN-based detectors: Deep SVDD (D_SVDD) (Ruff et al., 2018), AE(Xia et al., 2015), VAE (Kingma & Welling, 2013), LUNAR (Goodge et al., 2021), DROCC (Goyal et al., 2020), GOAD (Bergman & Hoshen, 2020), Neutral AD (N_AD) (Qiu et al., 2021), SO-GAAL (Liu et al., 2020), REPEN(Pang et al., 2018), DAGMM(Zong et al., 2018), ICL(Shenkar & Wolf, 2021) and flows_ood(Kirichenko et al., 2020).

¹<http://odds.cs.stonybrook.edu>

²<http://www.dbs.ifi.lmu.de/research/outlier-evaluation/DAMI>

Table 1: Summary of 22 real-world datasets (Ratio means Outlier Ratio).

Dataset	<i>dim</i>	Sample	Ratio (%)	Dataset	<i>dim</i>	Sample	Ratio (%)
pima(mat)	8	768	34.90	breastw(mat)	9	683	34.99
WBC(arff)	9	223	4.48	wine(mat)	13	129	7.75
HeartDisease(arff)	13	270	44.44	pendigits(mat)	16	6870	2.27
Lymphography(arff)	18	148	4.05	Hepatitis(arff)	19	80	16.25
Waveform(arff)	21	3443	2.90	wbc(mat)	30	378	5.56
WDBC(arff)	30	367	2.72	WPBC(arff)	33	198	23.74
satimage-2(mat)	36	5803	1.22	satellite(mat)	36	6435	31.64
KDDCup99(arff)	41	60839	0.40	SpamBase(arff)	57	4207	39.91
optdigits(mat)	64	5216	2.88	mnist(mat)	100	7603	9.21
musk(mat)	166	3062	3.17	Arrhythmia(arff)	259	450	45.78
speech(mat)	400	3686	1.65	InternetAds(arff)	1555	1966	18.72

For kNN, LOF, HBOS, OC-SVM, COPOD, ECOD, IForest, SUOD, LSCP, AE, VAE, and D_SVDD, we use the implementations from PyOD (Zhao et al., 2019b) which is a popular and open-source Python library for Outlier Detection. For others, we use the code given in their papers. In particular, from their source code, D_SVDD, DROCC, GOAD, N_AD, and LUNAR demand pure inliers, i.e., these methods select inliers based on labels and use only inliers as training data. For a fair comparison, we adapt them to the UOD setting by using the original dataset containing both inliers and outliers for model training. The comparison of experimental results using the initial settings of the paper and the UOD settings is presented in Sec. D.

Detailed Hyperparameter Settings. For kNN, LOF, HBOS, OC-SVM, COPOD, ECOD, IForest, SUOD and LSCP, we use default settings in the PyOD library where `n_neighbors` is 5 in kNN, `n_bins` is 10 in HBOS, `n_neighbors` is 20 in LOF and OC-SVM uses the sigmoid kernel. SUOD and LSCP are ensemble learning methods, and their basic detector composition is [LOF, LOF, LOF, LOF, COPOD, IForest, IForest], the parameters `n_neighbors` for the first four LOF algorithms are 15, 20, 25 and 35 respectively.

For DNN models, AE, VAE and D_SVDD use the sigmoid activation function and the SGD optimizer. We train them using 500 epochs with a learning rate of 0.005 and 2 hidden layers. The hidden layer dimensions are $\frac{dim}{2}$ and $\frac{dim}{4}$ for the two models, respectively. We train DROCC 100 epochs where 50 epochs are with CELOSS. The number of hidden nodes for the LSTM model is 128, and the SGD optimizer is used with a 0.005 learning rate and 0.99 momentum. We use the config file “config_arrhy.yml” provided in the source code from N_AD paper where *residual* transformation, Adam with 0.005 learning rate, 5 hidden layers with 64 dimensions and DCL loss are used. For SO-GAAL, the SGD optimizer is used with a 0.0001 learning rate for the generator and a 0.01 learning rate for the discriminator. LUNAR uses kNN with 20 `n_neighbors` to build a graph and constructs a discriminator with 4 layers with the Tanh activation function. SGD optimizer with a learning rate of 0.01 is adopted. REPEN uses the “rankod.py” from the paper code library to evaluate and the number of training epochs is 50. DAGMM adopts the Adam optimizer with a learning rate of 10^{-4} and a training epochs count of 200, where the *gmm.k* parameter is 4. flows_ood uses Adam optimizer with 10^{-3} learning rate and 5×10^{-5} L2 regularization weight decay, and the number of training epochs is 100. For flows_ood, we use the file “train_unsup_ood_uci.py” from the paper code library to train.

Tables 2, 4 and Tables 3, 5 show the results of NOD compared to other outlier detectors, in terms of AUC_ROC, F1-score respectively. Table 2 and Table 3 show the performance comparison between NOD and traditional methods, while Table 4 and Table 5 show the comparison with deep learning methods. The results show that most outlier detectors display significant performance variance on different datasets. Original data distribution highly influences the performances of traditional outlier detection algorithms due to their strong data assumptions, and only a tiny fraction of them can achieve good performance on the 22 datasets. For instance, kNN performs well on the *wine(mat)*, *wbc(mat)*, *breastw(mat)*, *Lymphography(arff)*, *WBC(arff)*, *satimage-2(mat)* and *WDBC(arff)* datasets, but poorly on others as its performance is highly influenced by *k*.

Table 2: Results in ROC_AUC (%) of all 9 compared classical detectors (average of 20 independent trials).

Dataset	kNN	LOF	HBOS	OC-SVM	COPOD	ECOD	IForest	SUOD	LSCP	nod
pima	60.76	53.84	68.6	50	65.4	51.73	67.33±0.9	64.93±0.76	61.7±0.81	62.9±2.95
breastw	97.53	38.32	98.5	0.49	99.44	99.14	98.7±0.16	90.95±1.76	75.69±1.56	99.29±0.24
WBC	98.73	83	98.2	0.75	99.06	99.01	99.04±0.21	98.31±0.31	97.4±0.12	99.16±0.21
wine	99.62	99.75	76.6	50	86.72	71.01	79.23±3.7	98.5±0.19	97.98±0.44	97.21±1.16
HeartDisease	60.53	50.05	74.7	14.78	69.46	58.81	62.22±1.24	61.77±1.01	57.02±0.84	67.08±3.76
pendigits	70.87	47.94	92.8	76.67	90.48	90.9	94.41±1.1	86.96±1.13	69.57±3	91.59±1.84
Lymphography	99.65	97.65	99.8	8.1	99.65	99.53	99.91±0.08	99.54±0.15	98.17±0.66	99.74±0.29
Hepatitis	66.88	62.57	77.7	69.8	80.37	78.65	69.41±1.89	73.08±2.56	72.24±1.78	69.69±3.39
Waveform	73.7	73.41	70	61.03	73.43	72.03	70.79±1.86	75.2±1.56	74.95±0.8	80.74±4.8
wbc	95	92.97	95.8	1.56	96.36	90.01	93.7±0.81	95.06±0.5	94.41±0.5	95.9±0.85
WDDB	99.41	98.15	93.1	50	97.09	91.74	93.53±0.91	96.77±0.22	95.77±0.15	97.39±0.47
WPBC	51.54	51.85	54.5	44.86	52.33	48.01	49±1.52	50.89±0.52	49.94±0.92	57.76±1.34
satimage-2	92.96	53.25	97.2	50	97.45	97.32	99.36±0.1	98.45±0.1	90.04±3.19	99.51±0.1
satellite	67.01	53.95	76.6	50	63.35	74.63	70.75±1.67	69.87±0.44	61.46±0.52	74.43±4.66
KDDCup99	43.9	62.54	98.4	91.33	99.19	99.24	98.91±0.08	99.03±0.05	93.73±1.46	98.94±0.13
SpamBase	48.64	45.13	63.7	30.43	67.71	64.45	62.1±1.96	61.16±0.85	55.97±0.96	68.4±0.98
optdigits	43.57	58.79	87	53.6	68.24	61.53	70.97±4.69	68.5±1.23	60.65±1.87	76.15±5.54
mnist	79.41	64.49	68.7	91.09	77.39	83.81	79.8±1.8	80.26±0.61	72.15±0.61	86.3±2.04
musk	30.38	41.24	99.8	1.1	94.63	95.5	99.97±0.05	91.71±0.81	67.39±9.94	98.18±0.69
Arrhythmia	74.33	72.59	74.8	66.83	75.76	77.37	75.05±1.3	75.22±0.28	73.29±0.33	73.98±0.54
speech	49.29	50.87	47.6	50.57	49.11	48.9	48.12±1.53	49.3±0.58	50.15±0.21	62.02±1.78
InternetAds	71.27	65.54	68.3	38.35	67.64	67.67	68.81±2	74.62±0.83	71.92±2.13	68.71±0.75
AUC_avg	71.59	64.45	81.0	43.24	80.47	78.23	79.6±1.3	80.0±0.7	74.6±1.5	83.0±1.8

Table 3: Results in F1-score (%) of all 9 compared classical detectors (average of 20 independent trials).

Dataset	kNN	LOF	HBOS	OC-SVM	COPOD	ECOD	IForest	SUOD	LSCP	nod
pima	44.8	34.11	50.75	0	48.88	37.31	51.38±1.27	47.82±1.42	44.13±1.19	48.99±1.81
breastw	87.88	13.84	93.5	0	94.56	92.89	92.33±0.63	78.01±3.1	51.26±3.35	94.46±0.9
WBC	70.59	0	70	0	80	80	70±3.16	63.61±6.7	59.85±2.35	75.5±4.97
wine	77.78	66.67	0	0	40	20	14.5±6.69	70.49±1.53	68.76±7.2	67±11
HeartDisease	44.34	45.3	70	15.83	60.83	52.5	51.5±1.25	51.68±1.6	45.75±0.94	59.29±3.59
pendigits	7.25	6.36	32.05	16.03	26.28	25	32.76±3.79	10.7±1.18	15.4±3.3	20.06±7.69
Lymphography	83.33	72.73	83.33	0	83.33	83.33	90±8.16	84.23±2.69	69.46±6.08	85±10.41
Hepatitis	0	17.39	30.77	30.77	46.15	38.46	19.23±3.85	22.01±5.32	21.23±4.53	22.69±7.08
Waveform	19.65	12.09	7	6	4	8	7.1±1.48	6.12±1.3	16.94±1.08	12.5±4.79
wbc	45	43.24	61.9	0	71.43	42.86	53.57±5.4	56.24±5.01	57.53±3.48	65±4.83
WDDB	80	84.21	40	0	80	50	64±4.9	79.43±1.36	62.67±4.39	63.5±4.77
WPBC	13.64	19.15	19.15	14.89	21.28	14.89	14.79±1.84	15.53±2.11	15.67±1.75	30.32±2
satimage-2	40	4.92	64.79	0	74.65	63.38	87.75±2.14	31.99±3.03	36.52±4	89.15±1.42
satellite	49.46	36.22	56.83	0	48.04	55.16	57.59±1.49	56.24±0.64	44.55±0.7	49.7±7.64
KDDCup99	7.74	0	39.02	53.66	45.93	45.53	40.92±1.35	37.29±6.96	30.7±6.67	37.36±1.5
SpamBase	40.04	34.26	51.53	23.94	56.46	54.14	50.21±2	50.41±0.81	43.76±1.28	57.33±0.99
optdigits	3.76	11.43	18.67	10.92	1.33	1.33	2.53±1.19	7.06±0.96	9.49±0.39	5.7±1.67
mnist	37.6	22.63	17.14	56.71	23.57	34.86	29.84±2.17	31.49±1.14	27.77±0.63	39.79±5.29
musk	1.4	3.73	90.72	0	36.08	40.21	96.8±3.86	14.15±3.35	15.02±8.43	65.15±7.09
Arrhythmia	64.82	62.69	64.56	57.28	64.56	66.5	64.95±1.61	66.4±1.04	63.64±0.64	64.49±0.82
speech	1.79	2.38	3.28	3.28	3.28	3.28	3.36±1.68	2.6±0.82	4.07±0.81	1.89±1.4
InternetAds	32.56	39.07	46.47	9.51	44.57	44.57	43.24±2.82	50.67±1.45	48.45±2.28	46.37±1.28
F1_avg	38.79	28.75	45.97	13.58	47.96	43.37	47.2±2.85	42.46±2.43	38.75±2.97	50.06±4.22

DNN-based methods: AE, VAE, D_SVDD, DROCC, GOAD, N_AD, SO-GAAL, LUNAR, DAGMM and flows_ood are in a similar situation. In particular, LUNAR relies on the kNN method, SO-GAAL has no clear criteria for the distance between positive and negative samples and VAE is based on the assumption that the inliers can be decoded from the encoding space better than the outliers. Moreover, D_SVDD, DROCC, GOAD, N_AD, and LUNAR need to use pure normal samples for training, contrary to the unsupervised setting. Therefore, we use the original datasets containing both inliers and outliers rather than only containing inliers to train these models. The following experimental results (Sec. D) show that training data mixed with some noise samples hurt their model performance. Except for REPEN, REPEN uses representation learning techniques to map high-dimensional data into low-dimensional embeddings and can be complementary to NOD. One of our future directions is to integrate representation learning techniques into NOD. With the loose assumption, NOD has a rather stable performance and achieves excellent ROC_AUC on almost all the tested datasets. It is worth noting that NOD has 9 average ROC_AUC scores above 0.95 on 22 datasets and NOD performs best among DNN methods with large margins. The results verify the effectiveness and robustness of NOD.

D PERFORMANCE ON DIFFERENT TRAINING SETTINGS

Following the papers of DROCC, GOAD, N_AD and LUNAR, these approaches need pure normal samples (inliers) to train the model. Since we focus on the unsupervised domain, these models are trained using original data (including outliers) as training data. Table 6 shows comparative results using the original paper setting and unsupervised setting. The results show that GOAD, N_AD and LUNAR are interfered by the noise in the training data. On the contrary, DROCC generally performs better in the unsupervised setting. This is because DROCC can be extended to solve One-class Classification with Limited Negatives. For both versions, their results are inferior to NOD.

Table 4: Results in ROC_AUC (%) of all 11 compared DNN-based detectors (average of 20 independent trials, ”/” indicates that the code provided by the paper cannot run on this dataset).

Dataset	D_SVDD	AE	VAE	LUNAR	DROCC	GOAD	N_AD	SOGAAL	REPEN	DAGMM	ICL	flows_ood	nod
pima	48.78±10.83	48.45±1.12	57.71±0.64	50.46±0.07	48.25±30.17	44.96±2.57	49.88±1.45	50.76±1.18	64.4±2.8	59.02±4.97	49.14±3.91	65.47±0.62	62.9±2.95
breastw	78.01±19.58	59.52±4.36	85.61±1.24	49.45±0.13	46.75±31.92	77.15±2.98	70.37±1.98	97.6±0.3	98.8±0.33	96.75±2.69	78.65±3.42	92.23±4.56	99.29±0.24
WBC	89.41±14.21	97.88±0.04	98.08±0.02	47.08±0.31	53.75±31.77	5.69±3.25	85.81±2.31	95.69±0.52	99.16±0.22	84.31±13.01	75.15±8.12	98.08±0.32	99.16±0.21
wine	42.33±28.2	57.18±4.04	74.94±0.64	30.02±0.59	47.9±31.61	71.56±20.32	79.27±4.55	51.13±1.25	99.87±0.08	95.51±9.23	50.62±20.44	65.13±2.98	97.21±1.16
HeartDisease	48.78±17.75	34.33±1.24	49.35±0.9	47.87±0.25	38.6±26.85	47.87±3.23	46.2±2.37	42.45±8.38	66.01±2.75	77.06±4.86	55.16±6.45	66.36±1.4	67.08±3.76
pendigits	49.32±25.64	93.3±0.39	93.69±0.06	56.39±0.09	44.7±30.36	20.14±12.56	78.64±4.94	66.23±9.7	97.69±0.34	91.7±3.65	57.6±8.3	81.68±3.97	91.59±1.84
Lymphography	54.57±25.17	99.75±0.04	99.68±0.05	25.28±1.4	56.35±31.16	21.4±13.69	82.95±9.31	94.89±8.69	99.13±0.51	/	93.57±4.9	97.72±0.4	99.74±0.29
Hepatitis	50.3±18.98	75.87±0.64	74.65±0.3	46.49±4.95	39.7±25.79	39.23±5.63	39.16±11.92	44.29±9.69	76.85±5.61	60.68±12.55	57.7±7.99	59.43±2.59	69.69±3.39
Waveform	54.38±17.43	65.44±0.85	65.52±0.19	49.49±0.07	53.85±35.58	44.17±2.16	76.12±2.1	33.78±3.02	78.01±4.71	60.77±12.11	53.89±1.54	66.83±0.85	80.74±4.8
wbc	64.13±26.84	82.32±0.64	91.56±0.13	42.57±0.31	45.6±31.03	14.98±3.44	85.71±2.63	12.3±6.72	95.81±0.48	94.35±3.54	81.71±3.86	94.66±0.3	95.9±0.85
WDBC	47.26±27.53	82.65±0.38	89.76±0.16	47.63±0.25	62.55±30.12	9.94±3.77	96.65±0.95	50.36±0.31	98.9±0.28	88.97±13.35	92.95±1.68	93.93±0.55	97.39±0.47
WPBC	49.49±6.64	42.81±0.27	46.43±0.15	47.81±0.2	54.15±36.59	51.14±2.47	43.86±3.32	50.15±3.98	52.37±2.2	55.62±5.83	53.72±3.63	50.11±1.11	57.76±1.34
satimage-2	61.82±32.82	97.47±0.04	97.76±0	55.36±0.05	58.6±30.64	87.85±8.24	97.19±0.71	44.75±10.05	99.86±0.05	88.7±10.25	78.45±6.2	99.41±0.21	99.51±0.1
satellite	53.61±13	70.19±0.41	62.22±0.07	50.93±0.01	50.4±33.59	48.23±2.9	70.23±2.23	48.96±3.08	71.93±2.55	55.04±9.62	57.4±7.73	70.7±0.8	74.43±4.66
KDDCup99	55.92±24.26	99.02±0	99.02±0	50.85±0.02	50.4±36.21	89.35±8.13	76.2±14.44	47.42±1.62	65.1±2.79	64.05±9.28	93.82±1.88	/	98.94±0.13
SpamBase	50.54±13.61	50.45±0.39	54.65±0.03	49.18±0.02	52.3±35.76	46.15±3.04	39.08±1.88	33.88±3.02	57.49±2.36	/	47.72±3.61	45.11±2.55	68.4±0.98
optdigits	52.21±23.56	44.43±0.98	50.43±0.09	48.57±0.15	56.8±29.07	58.33±13.18	55.03±4.41	42.44±11.99	89.01±1.22	79.71±8.73	66.01±3.58	/	76.15±5.54
mnist	53.68±12.62	89.1±0.08	85.76±0.01	49.19±0.1	56.2±32.95	44.97±7.91	88.39±1.31	49.36±0.32	86.51±0.65	55.83±6.9	90.63±0.16	/	86.3±2.04
musk	68.44±20.34	100±0	100±0	47.35±0.24	54.15±36.14	83.55±16.24	99.8±0.15	50±0	99.83±0.1	97.03±2.15	99.83±0.05	95.87±3.27	98.18±0.69
Arrhythmia	61.42±5.16	73.3±0.01	73.19±0.02	48.1±0.44	48.05±28.02	42.05±3.09	73.56±0.88	34.16±3.33	74.38±0.98	37.8±2.84	72.23±0.87	75.06±0.32	73.98±0.54
speech	49.48±5.14	46.99±0.02	46.91±0.01	56.78±0.26	58.15±32.29	51.92±3.66	49.97±1.59	48.88±1.77	54.09±1.38	47.48±5.3	54.02±2.03	1.08±0.14	62.02±1.78
InternetAds	70.32±3.64	61.4±0.01	61.46±0.01	51.28±0.11	49.05±36.37	43.05±2.19	67.16±2.76	38.12±5.43	81.22±0.57	/	54.42±3.86	63.87±3.47	68.71±0.75
AUC_avg	57±17.9	71.4±0.7	75.4±0.2	47.6±0.5	51.2±32	47.4±6.6	70.5±3.6	51.3±4.3	82.1±1.5	73.2±7.4	68.84±4.74	72.8±1.6	83±1.8

Table 5: Results in F1-score (%) of all 11 compared DNN-based detectors (average of 20 independent trials, ”/” indicates that the code provided by the paper cannot run on this dataset).

Dataset	D_AD	AE	VAE	LUNAR	DROCC	GOAD	N_AD	SOGAAL	REPEN	DAGMM	ICL	flowsood	nod
pima	34.96±10.24	34.4±1.62	43.31±0.71	34.65±0.21	50±25.1	31.53±2.01	34.33±2.03	51.48±0.57	51.72±2.33	54.34±2.89	17.28±3.08	55.16±0.65	48.99±1.81
breastw	69.3±20.43	50.55±4.67	77.58±1.34	33.95±0.38	48±27.68	61.26±2.43	46.44±2.48	95.21±0.56	93.43±1.28	48.42±31.71	35.58±1.81	68.49±1.83	94.46±0.9
WBC	38.5±19.31	60±0	69.02±1.81	10±0	53.5±26.51	0±0	13.5±7.26	52.32±2.93	72.5±5.36	0±0	24.24±12.71	8.97±0	75.5±4.97
wine	9±22.78	10±0	9.93±0.17	0±0	48±28.39	14±18.28	11±3	14.68±0.32	90±3.16	0±0	8.7±13.47	15.5±0	67±11
HeartDisease	43.25±14.83	34.79±1.53	46.59±0.4	42.62±0.54	41.5±24.35	44.08±2.58	41.04±2.02	38.29±6.85	56.62±1.98	73.41±2.93	22.31±5.41	61.19±1.2	59.29±3.59
pendigits	6.86±12.2	31.86±1.55	33.6±0.34	1.92±0	47±26.1	0.1±0.31	6.54±1.74	1.96±1.8	53.75±4.15	2.01±2.23	6.6±1.49	4.54±0	20.06±7.69
Lymphography	15±18.18	83.33±0	83.33±0	0±0	58±25.02	5±10.67	23.33±14.34	71.92±16.02	75.83±8.29	/	45.71±7.13	8.11±0	85±10.41
Hepatitis	19.23±14.49	38.46±0	38.46±0	8.85±7.41	41±21.42	9.62±9.06	8.85±9.51	10.77±10.43	34.62±12.99	24.5±12.41	20.95±7.13	32.5±0	22.69±7.08
Waveform	5±4.99	7.15±0.48	6.98±0.03	4±0	54.5±29.58	1.5±1.2	13.2±1.96	1.64±0.89	10.4±2.37	5.73±0.09	7.37±0.78	5.81±0	12.5±4.79
wbc	24.09±18.81	47.62±0	49.34±1.93	5.24±1.43	47±25.9	0±0	23.57±6.11	0.48±1.43	70.71±1.7	1.67±3.97	39.32±7.61	11.11±0	65±4.83
WDBC	9.5±12.03	60±0	60±0	10±0	62±26	0±0	66.5±9.1	5.34±0.03	80±0	0.27±1.19	33.19±3.18	5.45±0	63.5±4.77
WPBC	22.02±5.64	19.36±0.64	14.89±0	29.79±0	55±32.33	24.89±4.37	16.91±3.71	25.74±5.46	21.81±3.29	39.82±1.85	13.73±3.04	38.99±0.49	30.32±2
satimage-2	27.04±29.61	73.66±1.67	79.08±0.45	3.94±0.56	59±27.37	27.89±16.37	32.56±5.8	2.08±0.62	91.83±1.81	2.2±0.73	10.67±2.94	2.45±0	89.15±1.42
satellite	35.31±12.49	54.9±0.12	51.29±0.12	32.48±0.09	53±26.1	36.96±1.43	52.73±2.99	45.35±6.47	57.8±2.98	48.24±5.92	29.96±8.14	50.56±0.45	49.7±7.64
KDDCup99	10.25±11.99	42.28±0	42.26±0.03	0±0	52±32.34	35.16±13.21	7.24±4.8	0.76±0.03	1.22±0	0.79±0.01	6.13±0.9	/	37.36±1.5
SpamBase	41.05±12.14	41.94±0.2	43.56±0.03	39.75±0.15	53±30.51	36.14±3.11	31.5±1.79	26.07±3.03	45.34±2.55	/	9.18±1.42	46.14±1.85	57.33±0.99
optdigits	2.83±9.71	0±0	0±0	2.17±0.29	57±24.1	1.1±1.83	2.83±1.13	3.54±2.45	8.6±1.87	5.73±0.04	6.73±1.43	/	5.7±1.67
mnist	19.42±8.98	42.64±0.31	39±0	9.49±0.22	55.5±28.19	13.55±4.17	47.4±2.42	16.66±0.1	44.35±1.21	17.36±0.56	56.4±0.73	/	39.79±5.29
musk	19.54±22.12	100±0	100±0	1.96±0.31	53±29.34	30.52±22.12	88.66±4.08	6.14±0	89.79±3.34	0±0	48.02±0	6.34±0	65.15±7.09
Arrhythmia	55.56±4.44	64.08±0	64.17±0.19	44.66±0.75	48.5±22.42	40.12±3.04	64.25±1.56	34.7±2.67	64.66±1.33	46.87±2.24	32.83±0.32	68.89±0.63	64.49±0.82
speech	1.72±1.83	1.64±0	3.28±0	3.2±0.36	56.5±27.44	1.56±1.51	3.03±1.19	2.47±1.55	1.64±0.73	3.28±0.04	2.33±0.42	1.12±0.07	1.89±1.4
InternetAds	40.03±5.27	34.24±0	34.24±0	22.08±0.5	50±30.66	17.54±1.98	30.66±3.85	14.17±3.78	47.07±0.51	/	20.11±4.42	33.1±0.27	46.37±1.28
F1_avg	24.98±13.29	42.4±0.58	44.99±0.34	15.48±0.6	51.95±27.13	19.66±5.44	30.27±4.22	23.71±3.09	52.9±2.87	22.04±4.05	22.61±3.98	27.6±0.39	50.06±4.22

Table 6: Results in ROC_AUC (%) using different training settings (average of 20 independent trials). (S) means using the original settings of the paper and (U) denotes the unsupervised setting.

Dataset	DROCC(S)	DROCC(U)	GOAD(S)	GOAD(U)	N_AD(S)	N_AD(U)	LUNAR(S)	LUNAR(U)
pima	49.6±12.3	48.2±30.2	41.5±3.1	45.0±2.6	60.7±1.3	49.9±1.4	52.1±0	50.5±0.1
breastw	53.4±34.7	46.8±31.9	67.9±16.7	77.2±3.0	96.2±1.0	70.4±2.0	39.3±0.1	49.4±0.1
WBC	54.2±33.6	53.8±31.8	24.9±16.2	5.7±3.2	81.5±4.6	85.8±2.3	35.3±0.8	47.1±0.3
wine	65.5±32.8	47.9±31.6	39.0±18.1	71.6±20.3	95.4±1.9	79.3±4.6	42.8±1.3	30.0±0.6
HeartDisease	46.7±20.0	38.6±26.8	43.7±11.1	47.9±3.2	69.1±4.9	46.2±2.4	50.1±0.3	47.9±0.2
pendigits	16.7±1.3	44.7±30.4	24.8±13.8	20.1±12.6	98.5±0.8	78.6±4.9	51.2±0	56.4±0.1
Lymphography	48.8±28.6	56.4±31.2	98.2±3.6	21.4±13.7	90.0±4.9	83.0±9.3	47.8±1.4	25.3±1.4
Hepatitis	55.2±18.4	39.7±25.8	59.8±10.2	39.2±5.6	63.3±7.9	39.2±11.9	55.5±7.1	46.5±5.0
Waveform	49.2±7.4	53.8±35.6	44.0±2.9	44.2±2.2	80.1±1.3	76.1±2.1	48.0±0.2	49.5±0.1
wbc	47.3±30.1	45.6±31.0	49.5±14.9	15.0±3.4	92.7±2.0	85.7±2.6	96.1±0.1	42.6±0.3
WDBC	37.8±34.8	62.6±30.1	54.8±16.1	9.9±3.8	97.7±0.6	96.6±1.0	54.0±0.7	47.6±0.2
WPBC	58.0±8.7	54.2±36.6	50.3±4.2	51.1±2.5	49.0±7.1	43.9±3.3	49.2±0.4	47.8±0.2
satimage-2	33.4±7.8	58.6±30.6	98.8±0.6	87.8±8.2	99.8±0.1	97.2±0.7	99.9±0	55.4±0
satellite	44.0±1.8	50.4±33.6	70.8±1.2	48.2±2.9	81.1±0.4	70.2±2.2	50.0±0	50.9±0
KDDCup99	4.4±1.4	50.4±36.2	91.3±4.6	89.4±8.1	75.9±12.5	76.2±14.4	49.6±0	50.8±0
SpamBase	28.3±6.5	52.3±35.8	40.0±7.4	46.2±3.0	60.9±3.2	39.1±1.9	28.8±0.1	49.2±0
optdigits	74.3±13.0	56.8±29.1	73.5±15.1	58.3±13.2	82.8±4.6	55.0±4.4	99.5±0.1	48.6±0.2
mnist	24.1±6.1	56.2±33.0	56.5±7.0	45.0±7.9	97.8±0.2	88.4±1.3	92.4±0.3	49.2±0.1
musk	89.4±5.4	54.2±36.1	95.1±9.5	83.6±16.2	99.4±0.1	99.8±0.2	53.1±0.4	47.4±0.2
Arrhythmia	47.9±9.8	48.0±28.0	57.6±3.2	42.0±3.1	69.3±1.8	73.6±0.9	52.0±0.1	48.1±0.4
speech	58.8±5.5	58.2±32.3	51.7±4.3	51.9±3.7	47.9±2.6	50.0±1.6	49.7±0.2	56.8±0.3
InternetAds	13.6±0.5	49.0±36.4	52.4±5.2	43.0±2.2	75.7±1.0	67.2±2.8	40.8±0.1	51.3±0.1
AUC_avg	45.5±15.1	51.2±32.0	58.5±8.6	47.4±6.6	80.2±2.9	70.5±3.6	56.2±0.6	47.7±0.5

E PERFORMANCE ON DIFFERENT NOISE

To verify the effectiveness of uniformly distributed negative sampling under the NOD framework, we conduct experiments on two commonly used negative sampling methods in the field of outlier detection, SUBSPACE (Goodge et al., 2021) and GAN-BASED (Liu et al., 2020). The SUBSPACE method generates noise by adding Gaussian noise to the subset of feature dimensions of real data. The GAN-BASED method uses GAN to generate noise close to the real data.

To generate uniform noise, we first use uniform probability distribution to generate random values that we named Uniform Random (UR). However, the resulting two negative samples may be very close, giving false signals and disturbing model learning. So we adopt the Fast Poisson Disk (FPD) implementation (Bridson, 2007) to generate negative samples. FPD guarantees that the distance between the two samples is at least user-supplied r . But it runs too slowly to generate high-dimensional noise. Thus we only provide results on datasets where dim is below 10 using the FPD method.

In Table 7, we observe that the UR method is more effective than the SUBSPACE and GAN-BASED methods. In addition, the ROC_AUC (AVG_PART) shows no significant performance difference between FPD and UR. Considering the running time in high dimensions, we choose UR as the negative sampling method in NOD. We also incorporated Gaussian noise. NR0.5 means a Gaussian distribution with a mean of 0.5 and a standard deviation of 0.5. NR0.1 means a Gaussian distribution with a mean of 0.5 and a standard deviation of 0.1. The difference between their results is not significant because a Gaussian distribution with a larger std approaches Uniform noise.

F ANALYSIS OF DIFFERENT CLASSIFIERS

We evaluate the performance of different optimizers for the binary classification problem. In addition to the SGD, the SVC with RBF kernel (SVC), Decision Tree (DT), and Random Forest (RF) are tested. Fig. 2 shows the results of four different classifiers in the simulated 2-D OD problem. And the ROC_AUC performance of different classifiers is shown in Table 8.

Although most machine learning algorithms are designed with the so-call ‘‘smoothness prior’’, i.e., the function learn should not vary very much within a small region (Goodfellow et al., 2016), their actual performance in this binary classification task is quite different. As shown in Fig. 2, SVC_RBF, DT and RF try to separate different regions between positive samples and generated noise points with rigid boundaries. However, the inliers may overlap or be close to the random noise points. Thus,

Table 7: ROC_AUC (%) performance under different noise. (“/” indicates that the method did not obtain results within 2 hours)

Dataset	SUBSPACE	GAN	FPD	NR0.5	NR0.1	UR
pima	54.5	26.9	61.9	62.9	62.5	62.9
breastw	47.1	5.8	99.4	99.4	98.7	99.3
WBC	42.4	18.8	99.2	99.2	98.7	99.2
wine	44.3	10.8	/	93.8	94.4	97.2
HeartDisease	44.7	17.7	/	64.5	70.1	67.1
pendigits	56.9	79.2	/	91.8	83.3	91.6
Lymphography	56.2	48.4	/	99.8	99.3	99.8
Hepatitis	47.2	71.8	/	64.4	73.3	69.7
Waveform	51.6	57.3	/	80.8	79.6	80.7
wbc	50.1	1.5	/	95.2	95.6	95.9
WDBC	55.5	2.1	/	97.3	97.2	97.4
WPBC	50.4	44.4	/	57.9	57.0	57.8
satimage-2	48.3	73.1	/	99.0	99.4	99.5
satellite	48.7	72.3	/	78.9	68.6	74.4
KDDCup99	46.7	96.2	/	99.0	98.8	98.9
SpamBase	52.5	38	/	67.9	67.6	68.4
optdigits	50.4	51.8	/	76.0	78.5	76.2
mnist	45.8	82.2	/	86.7	83.6	86.3
musk	53.6	99.2	/	99.3	90.8	98.2
Arrhythmia	51.4	68.9	/	74.0	73.7	74
speech	49.4	49.2	/	60.8	58.7	62
InternetAds	51.8	40.6	/	68.7	68.6	68.7
lympho	52.6	81	/	97.7	96.2	97.3
arrhythmia	49.7	78.3	/	77.9	77.7	77.8
vowels	44	20.1	/	60.7	70.3	63.5
letter	50.7	45.8	/	59.7	58.8	58.9
cardio	58.1	70.9	/	80.4	74.3	77.8
mammography	43	36	84.9	84.8	74.6	81.7
shuttle	35.9	38.8	/	91.2	94.6	95.7
Stamps	48.8	67.7	91.3	90.6	84.6	88.6
Pima	46.5	29	62.8	63.7	60.9	63.4
AUC_avg	49.3	49.2	/	82.6	81.7	83.0
AUC(PART)	47	30.7	83.3	83.4	81.6	82.5

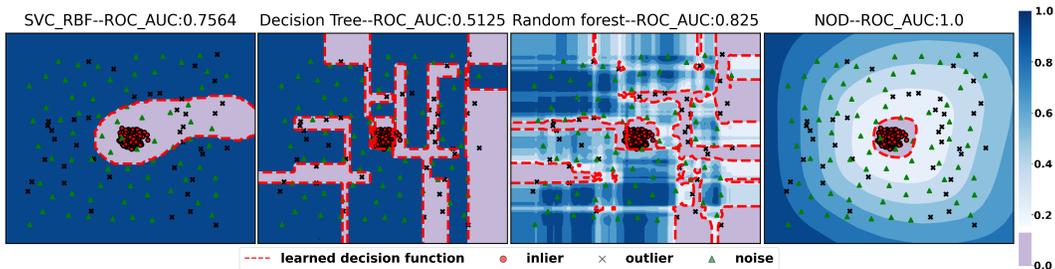


Figure 2: Comparison of different classifiers.

Table 8: ROC_AUC (%) performance under different classifiers.

Dataset	Linear_SVC	DT	RF	LGB	SGD	ADAM
pima	53.7	50.2	50.3	55.5	62.9	58.9
breastw	66.0	50.3	50.6	48.1	99.3	98.6
WBC	83.2	55.0	55.0	91.2	99.2	99.2
wine	81.0	49.8	49.6	67.6	97.2	93.7
HeartDisease	53.5	50.0	50.0	50.0	67.1	63.0
pendigits	63.5	50.3	50.0	59.8	91.6	87.0
Lymphography	98.1	50.0	50.0	98.9	99.7	99.9
Hepatitis	49.1	50.0	50.0	40.0	69.7	53.9
Waveform	75.1	50.2	50.3	59.7	80.7	82.1
wbc	64.2	52.2	52.2	83.8	95.9	84.6
WDBC	74.3	56.7	60.0	82.3	97.4	93.5
WPBC	48.6	50.2	49.5	52.2	57.8	55.3
satimage-2	96.0	50.9	50.6	96.6	99.5	99.7
satellite	59.2	50.0	50.0	52.2	74.4	48.4
KDDCup99	50.0	50.0	50.0	50.0	98.9	98.8
SpamBase	50.0	50.0	50.0	50.0	68.4	65.8
optdigits	50.0	50.0	50.0	50.0	76.2	70.5
mnist	50.0	50.0	50.0	50.0	86.3	88.1
musk	50.0	50.0	50.0	54.4	98.2	91.5
Arrhythmia	50.0	49.8	50.0	38.2	74.0	75.8
speech	49.9	49.9	50.0	49.1	62.0	60.8
InternetAds	50.0	49.9	50.0	51.5	68.7	57.8
AUC_avg	62.1	50.7	50.8	60.5	83.0	78.5

these classifiers cannot produce a smooth distribution estimation with their hard separation methods. SGD, in contrast, can generate smooth boundaries with different levels of abnormality. As seen in NOD, the center of the cluster has a very low anomaly score, and we have high anomaly scores when there are fewer inliers or outliers. In practice, there is often no clear boundary between outliers and outliers. Therefore, our solution can provide more detailed information about the degree of sample abnormality than solutions with only 0,1 labels.

G EXAMPLE OF MULTIPLE CLUSTERING CENTERS ON TWO-DIMENSIONAL DATA

As shown in Fig.3, we constructed some two-dimensional composite datasets with multiple clustering centers and visualized the distribution of NOD anomaly scores on each dataset. From Fig.3, it can be seen that under the premise of complying with the basic assumption of NOD, NOD is effective on datasets with multiple clustering centers.

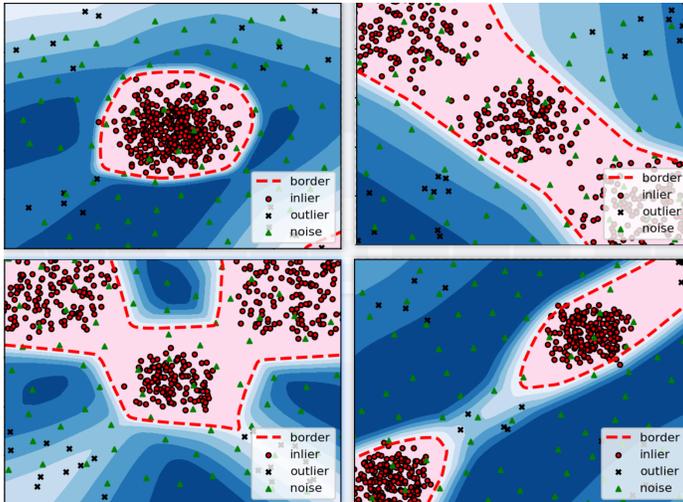


Figure 3: Example of multiple clustering centers on two-dimensional datasets.

H PERFORMANCE OF THE DIFFERENT EMBEDDING METHODS IN IMAGE DATASETS

Table 9 shows the anomaly detection performance of NOD on anomaly detection datasets constructed using different image embedding methods. From the experimental results in the table, it can be seen that different embedding methods can seriously affect the performance of NOD. Different pre-training models have different capabilities to capture intricate presentations or patterns. Resnet152 is much stronger than resnet18. Therefore, Resnet152 embeds more information than resnet18. Thus, the embeddings of outliers from resnet18, due to its lack of interacted patterns, are much more clustered than the ones from resnet152. This might explain the huge performance difference between resnet18 and resnet152 while the small performance difference between resnet50 and resnet152. Therefore, extending NOD to end-to-end anomaly detection solutions is a direction that needs to be explored in the future.

Table 9: ROC_AUC(%) performance of the different embedding methods on Image datasets.

Dataset	resnet18	resnet50	resnet152
airplane	68.22±0.79	91.34±0.18	95.31±0.07
automobile	42.75±0.91	95.94±0.05	96.62±0.08
bird	57.41±0.57	85.56±0.02	87.95±0.21
cat	46.83±1.00	88.87±0.02	90.01±0.13
deer	74.21±0.35	92.91±0.03	95.62±0.09
dog	41.20±1.02	88.95±0.32	92.28±0.26
frog	63.97±0.92	95.42±0.10	96.50±0.06
horse	53.54±0.59	91.14±0.25	95.19±0.11
ship	65.02±0.70	96.20±0.07	97.00±0.10
truck	57.83±0.91	96.89±0.10	97.66±0.01
AUC_avg	57.10±0.78	92.23±0.11	94.41±0.11

I PERFORMANCE COMPARISON BETWEEN NOD AND SOME DENSITY-BASED METHODS

NOD is suitable for density-based scenarios. Here, some classical density-based methods are involved in comparisons including LOF(Breunig et al., 2000), CBLOF(He et al., 2003), COF(Tang et al., 2002) and LOCI(Papadimitriou et al., 2003). Compared to these methods, NOD still has a significant performance lead.

Table 10: ROC_AUC(%) performance comparison between NOD and some density-based methods. (“/” indicates that the method did not obtain results within 2 hours, OOM denotes the out-of-memory error with 512G memory)

Dataset	LOF	CBLOF	COF	LOCI	NOD
pima	53.84	60.52	51.86	44.45	62.9
breastw	38.32	96.27	33.22	17.03	99.29
WBC	83	98.73	73.94	86.2	99.16
wine	99.75	99.92	97.9	65.46	97.21
HeartDisease	50.05	57.92	52.7	35.35	67.08
pendigits	47.94	92.2	52.37	/	91.59
Lymphography	97.65	99.88	99.41	83.92	99.74
Hepatitis	62.57	63.61	51.09	39.27	69.69
Waveform	73.41	74.97	70.03	/	80.74
wbc	92.97	94	87.13	/	95.9
WDBC	98.15	98.18	99.1	78.99	97.39
WPBC	51.85	46.78	47.43	43.61	57.76
satimage-2	53.25	99.86	55.83	/	99.51
satellite	53.95	73.2	53.55	/	74.43
KDDCup99	62.54	OOM	60.86	/	98.94
SpamBase	45.13	55.08	43.49	/	68.4
optdigits	58.79	88.28	57.29	/	76.15
mnist	64.49	80.43	62	/	86.3
musk	41.24	100	40.7	/	98.18
Arrhythmia	72.59	73.45	71.91	64.95	73.98
speech	50.87	47.28	52.98	/	62.02
InternetAds	65.54	71.42	67.86	/	68.71
AUC_avg	64.45	79.62	62.85	55.92	82.96

J PARAMETER SENSITIVITY ANALYSIS

This section examines the effects of various settings in NOD, including the ratios of negative samples, hidden layer dimensions, number of layers, and the usage of early stopping. Fig. 4(a) shows relative ROC_AUC change rates (Y-axis) with different ratios of negative samples (e.g., $0.1 * |X|$). X-axis values denote the dataset index ordered as Table 1. $|X|$ is normalized to 1 for the 22 datasets. Dots above 1 mean improved performance, while those below 1 indicate underperformance. We observe that the performance generally deteriorates when there are too many/small negative samples (e.g., the brown dots). Fig. 4(b) shows the effect of varying the hidden layer dimension. NOD is insensitive to changes in the hidden layer dimension. Fig. 4(c) shows the impacts of the number of layers with average ROC_AUC and standard deviations across 20 runs. The results indicate that a model with two layers outperforms a single-layer model. However, as the number of layers increases, the model’s fitting ability increases while the risk of overfitting also rises. Fig. 4(d) shows the comparison with(out) the proposed early stop. For most datasets, the early stop can effectively reduce the impact of overfitting and achieve better performance.

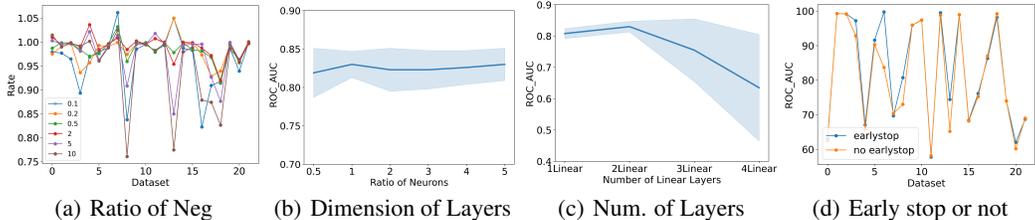


Figure 4: Performance under different settings. Shaded areas indicate standard deviations. X axis of (a) and (d) is the number of datasets with the same order as Table 2

REFERENCES

- Liron Bergman and Yedid Hoshen. Classification-based anomaly detection for general data. *arXiv preprint arXiv:2005.02359*, 2020.
- Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. *Advances in neural information processing systems*, 20, 2007.
- Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104, 2000.
- Robert Bridson. Fast poisson disk sampling in arbitrary dimensions. *SIGGRAPH sketches*, 10(1):1, 2007.
- Markus Goldstein and Andreas Dengel. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: poster and demo track*, 9, 2012.
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- Adam Goodge, Bryan Hooi, See Kiong Ng, and Wee Siong Ng. Lunar: Unifying local outlier detection methods via graph neural networks. *arXiv preprint arXiv:2112.05355*, 2021.
- Sachin Goyal, Aditi Raghunathan, Moksh Jain, Harsha Vardhan Simhadri, and Prateek Jain. DROCC: Deep Robust One-Class Classification, August 2020. URL <http://arxiv.org/abs/2002.12718>.
- Michael U Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of machine learning research*, 13(2), 2012.
- Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering cluster-based local outliers. *Pattern recognition letters*, 24(9-10):1641–1650, 2003.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- Polina Kirichenko, Pavel Izmailov, and Andrew G Wilson. Why normalizing flows fail to detect out-of-distribution data. *Advances in neural information processing systems*, 33:20578–20589, 2020.
- Zheng Li, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu. Copod: copula-based outlier detection. In *2020 IEEE International Conference on Data Mining (ICDM)*, pp. 1118–1123. IEEE, 2020.
- Zheng Li, Yue Zhao, Xiyang Hu, Nicola Botta, Cezar Ionescu, and George H Chen. Ecod: Un-supervised outlier detection using empirical cumulative distribution functions. *arXiv preprint arXiv:2201.00382*, 2022.
- Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pp. 413–422. IEEE, 2008.
- Yezheng Liu, Zhe Li, Chong Zhou, Yuanchun Jiang, Jianshan Sun, Meng Wang, and Xiangnan He. Generative adversarial active learning for unsupervised outlier detection. *IEEE Transactions on Knowledge and Data Engineering*, 32(8):1517–1528, 2020.
- Guansong Pang, Longbing Cao, Ling Chen, and Huan Liu. Learning representations of ultrahigh-dimensional data for random distance-based outlier detection. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2041–2050, 2018.

- Spiros Papadimitriou, Hiroyuki Kitagawa, Phillip B Gibbons, and Christos Faloutsos. Loci: Fast outlier detection using the local correlation integral. In *Proceedings 19th international conference on data engineering (Cat. No. 03CH37405)*, pp. 315–326. IEEE, 2003.
- Chen Qiu, Timo Pfroemer, Marius Kloft, Stephan Mandt, and Maja Rudolph. Neural transformation learning for deep anomaly detection beyond images. In *International Conference on Machine Learning*, pp. 8703–8714. PMLR, 2021.
- Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 427–438, 2000.
- Mihaela Rosca, Theophane Weber, Arthur Gretton, and Shakir Mohamed. A case for new neural network smoothness constraints. In *"I Can't Believe It's Not Better!" at NeurIPS Workshops*, volume 137 of *Proceedings of NeurIPS Workshops, Machine Learning Research*, pp. 21–32. PMLR, 2020.
- Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4393–4402. PMLR, 10–15 Jul 2018.
- Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.
- Tom Shenkar and Lior Wolf. Anomaly detection for tabular data with internal contrastive learning. In *International Conference on Learning Representations*, 2021.
- Jian Tang, Zhixiang Chen, Ada Wai-Chee Fu, and David W Cheung. Enhancing effectiveness of outlier detections for low density patterns. In *Advances in Knowledge Discovery and Data Mining: 6th Pacific-Asia Conference, PAKDD 2002 Taipei, Taiwan, May 6–8, 2002 Proceedings 6*, pp. 535–548. Springer, 2002.
- Yan Xia, Xudong Cao, Fang Wen, Gang Hua, and Jian Sun. Learning discriminative reconstructions for unsupervised outlier removal. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1511–1519, 2015.
- Yue Zhao, Zain Nasrullah, Maciej K Hryniewicki, and Zheng Li. Lscp: Locally selective combination in parallel outlier ensembles. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pp. 585–593. SIAM, 2019a.
- Yue Zhao, Zain Nasrullah, and Zheng Li. Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20:1–7, 2019b.
- Yue Zhao, Xiyang Hu, Cheng Cheng, Cong Wang, Changlin Wan, Wen Wang, Jianing Yang, Haoping Bai, Zheng Li, Cao Xiao, et al. Suod: Accelerating large-scale unsupervised heterogeneous outlier detection. *Proceedings of Machine Learning and Systems*, 3:463–478, 2021.
- Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*, 2018.