Normality-Calibrated Autoencoder for Unsupervised Anomaly Detection on Data Contamination

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this paper, we propose Normality-Calibrated Autoencoder (NCAE), which can boost anomaly detection performance on the contaminated datasets without any prior information or explicit abnormal samples in the training phase. The NCAE adversarially generates high confident normal samples from a latent space having low entropy and leverages them to predict abnormal samples in a training dataset. NCAE is trained to minimise reconstruction errors in uncontaminated samples and maximise reconstruction errors in contaminated samples. The experimental results demonstrate that our method outperforms shallow, hybrid, and deep methods for unsupervised anomaly detection and achieves comparable performance compared with semi-supervised methods using labelled anomaly samples in the training phase. The source code is publicly available on https://github.com/nonamescientist/NCAE_UAD.git.

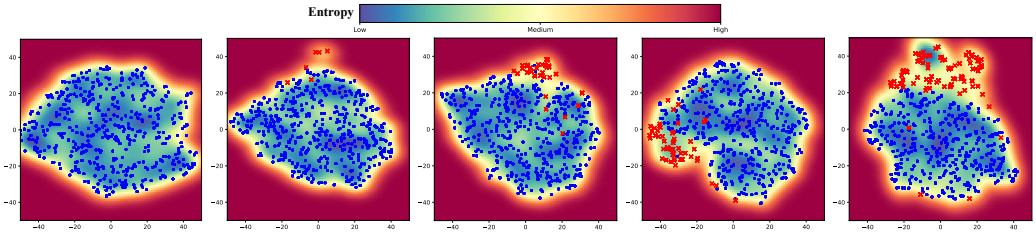
1 Introduction

Most of anomaly detection (AD) methods [Erfani et al., 2016, Zhai et al., 2016, Chen et al., 2017, Ruff et al., 2018, Deecke et al., 2018, Ruff et al., 2019b, Golan and El-Yaniv, 2018, Pang et al., 2019, Hendrycks et al., 2019a,b, Zong et al., 2018] assume that the training dataset only consists of normal samples; however, datasets in real-world are easily *contaminated*, which means that datasets contains both normal and abnormal samples. The contaminated samples significantly degrade the AD performance of models derived based on the assumption.

Various methods have been proposed [Ruff et al., 2019a, Song et al., 2017, Akcay et al., 2018, Chalapathy and Chawla, 2019, Zong et al., 2018] to improve the robustness of AD methods on contaminated datasets. Particularly, filtering contaminated samples based on contamination ratio [Zong et al., 2018, Ruff et al., 2019a], semi-supervised learning approaches that uses explicit abnormal samples in the training step [Wang et al., 2005, Liu and Zheng, 2006, Görnitz et al., 2013, Ruff et al., 2019a], and contamination sample prediction approaches based on geometric distance measurement [Berg et al., 2019, Li et al., 2021, Lai et al., 2020], have been proposed. However, the aforementioned approaches are domain or data-type specific. Additionally, those methods assume that abnormal samples are likely to be located far from the distribution of normal samples, and the entropy of abnormal samples is higher than that of normal samples [Berg et al., 2019, Li et al., 2021, Lai et al., 2020]. Unfortunately, as shown in Figure 1, if a training dataset is highly contaminated, the contaminated samples can also form a low entropy space by themselves.

In this paper, we present Normality-Calibrated Autoencoder (NCAE), which is robust to the training dataset contamination. Our key idea on the NCAE is to adversarially generate high confident normal samples from a low entropy feature space and then to contrastively compare the generated samples with the input samples for estimating contamination score. After identifying the contaminated samples, NCAE is trained to maximise reconstruction error of the found sample.

Submitted to NeurIPS 2021 Workshop on DGMs and Downstream Applications. Do not distribute.



(a) No contamination (b) 1% (c) 5% (d) 10% (e) 20%

Figure 1: Entropy and distribution of latent features under different contamination ratios: (a) No contamination, (b) 1%, (c) 5%, (d) 10%, and (e) 20%. The samples on the '5' class on the MNIST dataset are used as normal (blue dots) and contaminated samples (red x-marks) are randomly picked from the training samples of the remaining classes. The 500 samples are randomly picked for the visualisation. When a dataset is highly contaminated (*i.e.,* contamination ratio over 10%), contaminated samples are also located in a low entropy region.

2 Normality-Calibrated Autoencoder

2.1 Learning normality-calibrated autoencoder

For n number of input samples with D dimensions $\mathcal{X} = \{x_i\}_{i=1:n}$, $x \in \mathbb{R}^D$ and the corresponding latent features with d dimensions $\mathcal{Z} = \{z_i\}_{i=1:n}$, $z \in \mathbb{R}^d$, let an autoencoder is composed of an encoder $f(x) : x \rightarrow z$ and a decoder $g(x) : z \rightarrow \bar{x}$. The general objective of the autoencoder is training f and g to minimise an error between input samples x and the reconstruction results $\bar{x}$, as follows:

$$\min_{f,g} \mathbb{E}_{x \sim p_{\mathcal{X}}} ||x - \bar{x}||^2, \quad \bar{x} = g \cdot f(x), \tag{1}$$

where $p_{\mathcal{X}}$ denotes the entire input samples. However, an autoencoder is known to have an over-confidence issue, *i.e.,* low reconstruction error of unseen samples. AD methods using the autoencoder usually identify abnormal samples using the reconstruction error. Therefore, even if the autoencoder takes anomaly samples as inputs, it may not distinguish whether the samples are abnormal or not [Pidhorskyi et al., 2018, Yu et al., 2021]. This over-confidence issue would be more deepened when a training dataset is contaminated.

One straightforward approach to prevent this issue is adding an extra term to maximise reconstruction error for contaminated samples. We define normality-calibrated reconstruction (NCR) loss as follows:

$$\min_{f,g} \mathbb{E}_{x \sim p_{\mathcal{X}^{\mathrm{N}}}} ||x - \bar{x}||^2 - \mathbb{E}_{x^{\mathrm{c}} \sim p_{\mathcal{X}^{\mathrm{C}}}} ||x^{\mathrm{c}} - \bar{x}^{\mathrm{c}}||^2, \tag{2}$$

where $p_{\mathcal{X}^{\mathrm{N}}}$ and $p_{\mathcal{X}^{\mathrm{C}}}$ denote the normal samples and contaminated samples, respectively, among input samples $p_{\mathcal{X}}$. Now, we should find out which samples are contaminated to optimise autoencoder using Eq. 2 properly.

2.2 high-confidence normal samples generation using Generative Adversarial Network

We find contaminated samples by using high confident normal samples generated from low entropy latent space. We apply the generative adversarial network (GAN) Goodfellow et al. [2014] framework to do this. The high-confidence normal sample generation via the GAN framework is carried out as follows. Initially, we transform a distribution of all latent features, which are encoded from input samples through the encoder f, to a more knowledgeable probabilistic distribution such as Gaussian distribution. And then, we generate samples using noise signals sampled from the centre of the knowledgeable distribution, *i.e.,* the low entropy space. An adversarial loss for transforming a latent feature distribution to a more knowledgeable probabilistic distribution is defined by follow:

$$\min_{f} \max_{D_l} \mathbb{E}_{\omega \sim N(\mu_{\mathcal{Z}}, I_d)} [\log D_l(\omega)] + \mathbb{E}_{x \sim P_{\mathcal{X}}} [\log (1 - D_l(f(x)))], \tag{3}$$

where D_l denotes the discriminator for latent features, and $N(\mu_{\mathcal{Z}}, I_d)$ defines a normal distribution with the mean of latent features $\mu_{\mathcal{Z}} \in \mathbb{R}^d$ and a covariance matrix defined by an identity matrix $I_d \in \mathbb{R}^{d \times d}$. $\mu_{\mathcal{Z}}$ is initialised by the mean value of latent features: $\mu_{\mathcal{Z}} = \frac{1}{n} \sum_{i=1}^{n} z_i$. We would want

each component of z to be maximally informative such as each of them to be an independent random variable. Therefore, the covariance matrix is determined by the $d \times d$ identity matrix.

Since as f and D_l are being updated, $\mu_{\mathcal{Z}}$ would be shifted during the training step. $\mu_{\mathcal{Z}}$ is updated at every training step as follows:

$$\mu_{\mathcal{Z}}^{t+1} = \mu_{\mathcal{Z}}^{t} - \gamma \frac{1}{m} \sum_{i=1}^{m} (\mu_{\mathcal{Z}}^{t} - z_i), \quad \mu_{\mathcal{Z}}^{0} = \frac{1}{n} \sum_{i=1}^{n} z_i^0 \tag{4}$$

where $\mu_{\mathcal{Z}}^{t+1}$ and $\mu_{\mathcal{Z}}^{t}$ denote the $\mu_{\mathcal{Z}}$ on $t+1$-th and t-th training step, respectively. m is the batch size and z_i is i-th latent features on the batch. γ is a learning rate.

To generate high confident normal samples, we formulate the following adversarial loss:

$$\min_{g} \max_{D_s} \mathbb{E}_{x \sim P_{\mathcal{X}}}[\log D_s(x)] + \mathbb{E}_{\acute{\omega} \sim N(\mu_{\mathcal{Z}}, \sigma I_d)}[\log(1 - D_s(g(\acute{\omega}))], \tag{5}$$

where D_s denotes the discriminator for samples, and $N(\mu_{\mathcal{Z}}, \sigma I_d)$ is a d-dimensional normal distribution with the mean $\mu_{\mathcal{Z}} \in \mathbb{R}^d$ and the covariance matrix $\sigma I_d \in \mathbb{R}^{d \times d}$. $\mu_{\mathcal{Z}}$ is equivalent to the $\mu_{\mathcal{Z}}$ in Eq. 3. σI_d is defined by multiplication of a scalar value $\sigma \in [0, 1]$ and the identity matrix I_d. σ is a hyperparameter to control the compactness of random noise for generating samples using the decoder g. The smaller σ can give more strong likely to generate high confident normal samples.

2.3 Contaminated sample mining and joint learning

To predict contaminated samples, we use the generated high confident normal samples as a dictionary. With the generation process for high confident normal samples: $g(\acute{\omega}) = \acute{x}$, we construct a latent feature dictionary $\mathcal{M} = [\acute{z}_i]_{i=1:m}$, $\acute{z}_i = f(\acute{x}_i)$ and $\mathcal{M} \in \mathbb{R}^{m \times d}$, where m is the batch size. By leveraging $\mathcal{M}$ and given each training batch $\{x_i\}_{i=1:m}$, we define a pseudo contamination score c_i of each input sample x_i as follow:

$$c_i = \frac{1}{m} \sum_{j=1}^{m} f(x_i) \cdot \acute{z}_j^{\mathrm{T}}, \quad \acute{z}_j \in \mathcal{M}, \tag{6}$$

where T denotes the transpose of the vector. We apply $l2$-normalisation to improve robustness on the variation of the vector scale of the operation.

We predict the contaminated samples by sorting the score in descending order and picking top-τ% samples among the sorted results as the contaminated samples; thus, the number of predicted contaminated samples are decided by τm that is a multiplication of τ and the batch size m. The above process is represented as follow:

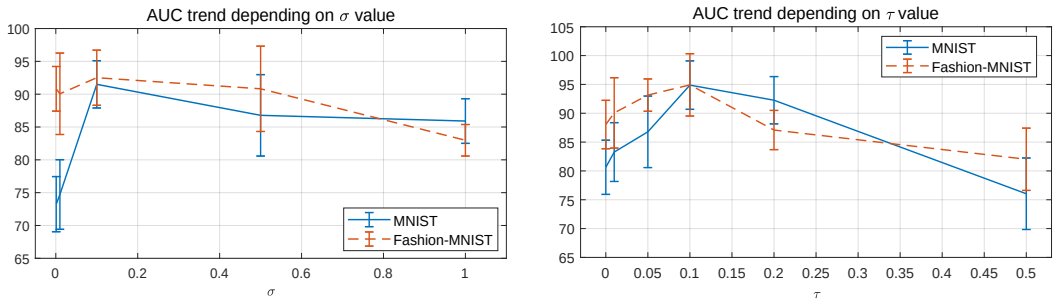
$$\mathcal{X}^{\mathrm{C}} = \{x_t\}_{t \in C[1:\lceil \tau m \rceil]}, \quad C = \arg\operatorname*{sort}_{i} c_i, \quad w.r.t., 1 \leq i \leq m$$

where C is a set of the sorted indices of input batch samples in descending order of the contamination score (Eq. 6), and $\mathcal{X}^{\mathrm{C}}$ is a set of predicted contaminated samples. $\lceil \cdot \rceil$ denotes the ceiling function. τ effects of deciding the number of predicted contaminated samples, so it directly affects the AD performance of our method.

The objective function for joint learning the entire components on our method is as follows:

$$\begin{aligned} \min_{f,g} \max_{D_l, D_s} \quad & \underbrace{\mathbb{E}_{x \sim p_{\mathcal{X}^{\mathrm{N}}}} ||x - f \cdot g(x)||^2 - \mathbb{E}_{x \sim p_{\mathcal{X}^{\mathrm{C}}}} ||x - \bar{x}'||^2}_{\text{(a)}} \\ & + \underbrace{\mathbb{E}_{\omega \sim N(\mu_{\mathcal{Z}}, I_d)}[\log D_l(\omega)] + \mathbb{E}_{x \sim P_{\mathcal{X}}}[\log (1 - D_l(f(x)))]}_{\text{(b)}} \\ & + \underbrace{\mathbb{E}_{x \sim P_{\mathcal{X}}}[\log D_x(\omega)] + \mathbb{E}_{\omega' \sim N(\mu_{\mathcal{Z}}, \sigma I_d)}[\log(1 - D_s(g(\omega')))]}_{\text{(c)}}, \end{aligned} \tag{7}$$

where $\bar{x}'$ is defined by the nearest sample from the given contaminated samples among the generated high confident normal samples $g(N(\mu_{\mathcal{Z}, \sigma I_d}))$ on the latent feature space. (a), (b), and (c) denote the NCR loss and the two adversarial losses, respectively.



(a) Effects of σ (b) Effects of τ

Figure 2: Ablation studies about unsupervised AD performance depending on σ and τ. (a) and (b) represent the trends of AUC with respect to the setting of σ and τ, respectively, on the MNIST and Fashion-MNIST datasets.

3 Experiments

3.1 Experiment setting and Dataset

We follow the unsupervised AD protocol described by Ruff *et al.* [Ruff et al., 2019a]. MNIST and Fashion-MNIST datasets are used for the experiments. We set one of the classes provided by a dataset as normal and others as abnormal. After we decide contamination ration $\rho = \frac{A}{N+A}$, where N and A are the numbers of normal and abnormal samples, respectively, we pick normal samples from the chosen class and contaminated samples from the remaining classes. In the test phase, the samples of the normal class are labelled by 0, and other samples are labelled by 1. For the performance analysis, Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC) are used.

We employ LeNet-type convolutional neural networks (CNNs) on MNIST and Fashion-MNIST datasets, where each convolutional module consists of a convolutional layer followed by leaky ReLU activation functions with leakiness of 0.1. We use the Adam optimiser with the recommended default hyperparameters [Kingma and Ba, 2015]. The batch size is set to 128. The initial learning rate is 0.01 and decayed every 10 epochs by multiplying 0.1. σ and τ are decided by 0.1 and 0.1 (based on the results from the ablation study), respectively.

3.2 Ablation study

We analyse unsupervised AD performance depending on the setting of σ and τ. MNIST and Fashion-MNIST datasets are used for the ablation study. Ablation studies are conducted based on the experimental protocol described in the previous section. The contamination ratio ρ is fixed to 0.2.

Parameter analysis on σ: When σ is too small, then the distribution of sample noise for generating samples would be too compact so that the generated samples can not provide comprehensive information to cover the diverse patterns of normal samples. On the other hand, when σ is too large, then there is a possibility that the noise can be sampled from low entropy space (*i.e.,* abnormal samples also can be generated).

Figure 2(a) shows the AUC trends depending on the σ. The AUC increases rapidly in the case of sigma is less than 0.1, and then decreases gradually. This can be interpreted as follows. If the sampling space is too compact (*i.e.,* when σ is too small), it means that the generated normal sample does not provide enough information to distinguish the contaminated sample. When sampling space is too broad (*i.e.,* when σ is too large), it also degrades performance, but the impacts of the braoder sampling space are relatively less than that of the smaller sampling space (e.g., when $\sigma \leq 0.1$). The best performance is obtained by σ of 0.1.

Parameter analysis on τ: τ decides the number of predicted contaminated samples per training batch. The lower τ can provide more precise prediction performance but may not enough to provide more comprehensive prediction performance. In contrast, when τ is too large, the predicted results possibly more accurate but also may have a lot of false-positive results.

Table 1: Performance comparison on unsupervised anomaly detection in terms of various contamination ratios ρ. MNIST and Fashion-MNIST datasets are used for the comparison. The **bolded** figures indicate the best performances.

Dataset	ρ	OC-SVM	IF	KDE	CAE	Deep SVDD	SSAD	SS-DGM	Deep SAD	Classification	NCAE
MNIST	.00	96.0±2.9	85.4±8.7	95.0±3.3	92.9±5.7	92.8±4.9	**97.9±1.8**	92.2±5.6	96.7±2.4	94.5±4.6	94.0±4.2
	.01	94.3±3.9	85.2±8.8	91.2±4.9	91.3±6.1	92.1±5.1	96.6±2.4	92.0±6.0	95.5±3.3	91.5±5.9	**97.2±5.2**
	.05	91.4±5.2	83.9±9.2	85.5±7.1	87.2±7.1	89.4±5.8	93.4±3.4	91.0±6.9	93.5±4.1	86.7±7.4	**97.0±7.1**
	.10	88.8±6.0	82.3±9.5	82.1±8.5	83.7±8.4	86.5±6.8	90.7±4.4	89.7±7.5	91.2±4.9	83.6±8.2	**92.6±5.7**
	.20	84.1±7.6	78.7±10.5	77.4±10.9	78.6±10.3	81.5±8.4	87.4±5.6	87.4±8.6	86.6±6.6	79.7±9.4	**89.8±7.4**
F-MNIST	.00	92.8±4.7	91.6±5.5	92.0±4.9	90.2±5.8	89.2±6.2	**94.0±4.4**	71.4±12.7	90.5±6.5	76.8±13.2	91.5±9.7
	.01	91.7±5.0	91.5±5.5	89.4±6.3	87.1±7.3	86.3±6.3	92.2±4.9	71.2±14.3	87.2±7.1	67.3±8.1	**94.5 ±4.7**
	.05	90.7±5.5	90.9±5.9	85.2±9.1	81.6±9.6	80.6±7.1	88.3±6.2	71.9±14.3	81.5±8.5	59.8±4.6	**92.4± 8.2**
	.10	89.5±6.1	90.2±6.3	81.8±11.2	77.4±11.1	76.2±7.3	85.6±7.0	72.5±15.5	78.2±9.1	56.7±4.1	**91.5±5.7**
	.20	86.3±7.7	88.4±7.6	77.4±13.6	72.5±12.6	69.3±6.3	81.9±8.1	70.8±16.0	74.8±9.4	53.9±2.9	**88.9±9.2**

As shown in Figure 2(b), the AUC increases rapidly with τ from 0 to 0.1 and then decreases slowly. The results can be interpreted as follows. Finding contaminated samples themselves has a large impact on the AD performance, but the quantity of found samples affects less to AD performance. But, predicting too many samples may degrade AD performance by taking a great number of false positives. The best performance is obtained by τ of 0.1.

3.3 Comparison with other methods

We consider the OC-SVM [Schölkopf et al., 2001] , isolation forest (IF) [Liu et al., 2008], and KDE [Parzen, 1962] for shallow unsupervised baselines. For deep unsupervised competitors, we consider general binary classifider (supervised), convolutional autoencoders (CAE), deep support vector data description (Deep SVDD) [Ruff et al., 2018], semi-supervised anomaly detection (SSAD) [Ruff et al., 2019a], semi-supervised deep generative model (SS-DGM) [Kingma and Ba, 2015], and deep semi-supervised anomaly detection (Deep SAD) [Ruff et al., 2019a]. We repeat this training set generation process 10 times per AD setup over all the nine respective anomaly classes and report the average results over the resulting 90 experiments per contamination ratio.

Table 1 shows the quantitative performance comparison depending on the contamination ratio ρ. In the comparison using the MNIST dataset, the proposed NCAE achieves the best performances except when the dataset is not contaminated ($\rho = 0.0$). Even compared with semi-supervised approaches [Ruff et al., 2018, 2019a] which use explicit anomaly samples in the training phase, the NCAE shows outstanding performances. This trend is also shown in the performance comparison using the Fashion-MNIST dataset. The NCAE produces the AUC of 91.57 and 88.97 for the Fashion-MNIST dataset with 0.1 and 0.2 contamination ratios, respectively. Those figures are the best performance among the listed methods when a dataset is contaminated.

The interpretation of the relatively low performance on the uncontaminated dataset ($\rho = 0.0$) is as follows. Basically, our method is derived under the assumption that a training dataset is contaminated. Therefore, even if the dataset is not contaminated, the NCAE tries to find some anomaly samples and maximise the reconstruction errors of the samples during the model training. This process degrades the performance of our methods as shown in the experimental results. This is a critical defect of our method;

Overall, the comparison results demonstrate the advantage of the proposed NCAE that can detect anomaly samples on data contamination without prior knowledge or explicit abnormal samples in the training phase.

4 Conclusion

In this work, we have proposed NCAE that is a generative method for fully unsupervised anomaly detection on contaminated data. The experimental results have suggested that the NCAE outperforms existing methods for fully unsupervised anomaly detection with a large margin, and they have also provided competitive performances compared with semi-supervised methods using explicit abnormal samples to train their AD model.

References

S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon. GANomaly: Semi-supervised anomaly detection via adversarial training. In *ACCV*, pages 622–637, 2018.

A. Berg, J. Ahlberg, and M. Felsberg. Unsupervised learning of anomaly detection from contaminated image data using simultaneous encoder training. *arXiv preprint arXiv:1905.11034*, 2019.

R. Chalapathy and S. Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

J. Chen, S. Sathe, C. C. Aggarwal, and D. S. Turaga. Outlier Detection with Autoencoder Ensembles. In *SDM*, pages 90–98, 2017.

L. Deecke, R. A. Vandermeulen, L. Ruff, S. Mandt, and M. Kloft. Image anomaly detection with generative adversarial networks. In *ECML-PKDD*, 2018.

S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognition*, 58: 121–134, 2016.

I. Golan and R. El-Yaniv. Deep anomaly detection using geometric transformations. In *NeurIPS*, pages 9758–9769, 2018.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. In *NIPS*, pages 2672–2680, 2014.

N. Görnitz, M. Kloft, K. Rieck, and U. Brefeld. Toward supervised anomaly detection. *Journal of Artificial Intelligence Research*, 46:235–262, 2013.

D. Hendrycks, M. Mazeika, and T. G. Dietterich. Deep anomaly detection with outlier exposure. In *ICLR*, 2019a.

D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song. Using self-supervised learning can improve model robustness and uncertainty. In *NeurIPS*, pages 15637–15648, 2019b.

D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015.

C.-H. Lai, D. Zou, and G. Lerman. Robust subspace recovery layer for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=rylb3eBtwr`.

T. Li, Z. Wang, S. Liu, and W.-Y. Lin. Deep unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3636–3645, 2021.

F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation Forest. In *ICDM*, pages 413–422, 2008.

Y. Liu and Y. F. Zheng. Minimum enclosing and maximum excluding machine for pattern description and discrimination. In *ICPR*, pages 129–132, 2006.

G. Pang, C. Shen, and A. van den Hengel. Deep anomaly detection with deviation networks. In *KDD*, pages 353–362, 2019.

E. Parzen. On Estimation of a Probability Density Function and Mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

S. Pidhorskyi, R. Almohsen, and G. Doretto. Generative probabilistic novelty detection with adversarial autoencoders. In *NeurIPS*, pages 6822–6833, 2018.

L. Ruff, R. A. Vandermeulen, N. Görnitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft. Deep one-class classification. In *ICML*, volume 80, pages 4390–4399, 2018.

L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft. Deep semi-supervised anomaly detection. *arXiv preprint arXiv:1906.02694*, 2019a.

L. Ruff, Y. Zemlyanskiy, R. Vandermeulen, T. Schnake, and M. Kloft. Self-attentive, multi-context one-class classification for unsupervised anomaly detection on text. In *ACL*, pages 4061–4071, 2019b.

B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the Support of a High-Dimensional Distribution. *Neural computation*, 13(7):1443–1471, 2001.

H. Song, Z. Jiang, A. Men, and B. Yang. A hybrid semi-supervised anomaly detection model for high-dimensional data. *Computational Intelligence and Neuroscience*, 2017.

J. Wang, P. Neskovic, and L. N. Cooper. Pattern classification via single spheres. In *International Conference on Discovery Science*, pages 241–252. Springer, 2005.

J. Yu, Y. Lee, K. C. Yow, M. Jeon, and W. Pedrycz. Abnormal event detection and localization via adversarial event prediction. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

S. Zhai, Y. Cheng, W. Lu, and Z. Zhang. Deep structured energy based models for anomaly detection. In *ICML*, volume 48, pages 1100–1109, 2016.

B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018.