On the Environmental Impact of the Algorithm LatentOut for Unsupervised Anomaly Detection

Fabrizio Angiulli[1,†], Fabio Fassetti[1,†] and Luca Ferragina[1,*,†]

[1]DIMES Dept., University of Calabria, 87036 Rende (CS), Italy.

Abstract

Because of their astonishing performances, Deep Neural Network-based approaches have become pervasive in many human activities. However, they often require a long, energy-intensive training phase, which has a huge environmental impact.

In recent years, there has been a significant increase in the emphasis placed on environmental themes across various sectors, driven by growing concerns over climate change and sustainability. This heightened focus has led to many initiatives, policies and discussions aimed at addressing ecological challenges and promoting a more sustainable future. For the reasons stated above, Deep Learning cannot be exempted from such initiatives and the literature is starting to pay attention to these issues. This paper aims at contributing to this field, in particular, concerning the Anomaly Detection Task whose environmental impact, due to its widespread employment, deserves to be addressed.

Specifically, we focus on the Anomaly Detection field that, such as many other Data Mining tasks, is not excluded from this analysis. In particular, we consider Latent*Out*, a recently introduced Deep Learning-based framework for unsupervised Anomaly Detection that exploits both the latent space and the baseline anomaly score (i. e. the reconstruction error) of a Variational Autoencoder (VAE) to provide a refined anomaly score performing density estimation in the augmented latent-space/baseline-score feature space.

We analyze the environmental impact of Latent*Out* in terms of carbon footprint by measuring the (estimated) CO_2 consumption through the Python library CodeCarbon. We observe that, with equal CO_2 consumption, Latent*Out* achieves much better performances than the standard VAE. Moreover, we compare Latent*Out* with other Anomaly Detection Neural Network-based methods and we highlight that it is the one that obtains the best results in terms of a balance between high accuracy performance and low carbon footprint.

Keywords

Anomaly Detection, Variational Autoencoder, Carbon Footprint

1. Introduction

Anomalies can be defined as examples that significantly deviate from the majority of the data to arise the suspect of being generated by a different mechanism. Anomaly Detection represents a fundamental task in many human activities, including Healthcare, Cyber-security, Industrial Monitoring, Fraud Detection, and many others.

It is possible to identify three different types of settings of Anomaly Detection [1]. In the *Supervised setting* a dataset whose items are labeled as normal and abnormal is available to build a classifier, typically the dataset is highly unbalanced and the anomalies form a rare class. The *Semi-supervised setting*, also called *one-class*, is characterized by the presence in input of only examples from the normal class that are used to train the detector. In the *Unsupervised setting* the goal is to assign an anomaly score to each object of the input dataset in order to find anomalies in it.

Classical data mining and machine learning algorithms performing the task of detecting outliers include statistical-based [2], distance-based [3, 4, 5, 6], density-based [7, 8], reverse nearest neighbor-based [9, 10, 11], SVM-based [12, 13], and many others [1].

1st Workshop on Green-Aware Artificial Intelligence, 23rd International Conference of the Italian Association for Artificial Intelligence (AIxIA 2024), November 25–28, 2024, Bolzano, Italy

*Corresponding author.
†These authors contributed equally.
f.angiulli@dimes.unical.it (F. Angiulli); f.fassetti@dimes.unical.it (F. Fassetti); luca.ferragina@unical.it (L. Ferragina)
0000-0002-9860-7569 (F. Angiulli); 0000-0002-8416-906X (F. Fassetti); 0000-0003-3184-4639 (L. Ferragina)

Recently, the approaches that have achieved the most success have been those based on deep learning [14], which can be divided into three main families: reconstruction error-based methods employing Autoencoders (AE), models based on Generative Adversarial Networks (GAN), and SVM-like neural architectures.

At the basis of the application of Autoencoders (AE) and Variational Autoencoders (VAE) [15, 16, 14] to Anomaly Detection relies the concept of *reconstruction error*. More in detail, (Variational) Autoencoders are trained to map data into a low dimensional latent space and then map them back into the original space generating in output a reconstruction of the input as similar as possible to it. Since the majority of the data used for training models belongs to the normal class, it is assumed that these networks are able to reconstruct the inliers better than the outliers and, thus, the reconstruction error can be adopted as an *anomaly score*.

GAN-based models [17, 18, 19, 20] basically consist in the combined, adversarial training of two sub-architectures, the *generator* and the *discriminator*. Specifically, the generator network produces artificial anomalies as realistic as possible, and the discriminator assigns an anomaly score to each item.

SVM-like methods [21, 22, 23] leverage the idea of enclosing normal data into a hypersphere employing a One-Class SVM-like loss function combined with a deep neural architecture. A slightly different approach that can be included in this family, is introduced in [24] where the architecture presents an additional final layer composed of just one neuron that produces an anomaly score that, for anomalies, is as far as possible from a value obtained as the average of randomly sampled normal items anomaly scores.

Moreover, in [25] has been introduced Deep Isolation Forest (DIF), a novel methodology that utilizes casually initialized neural networks to map original data into random representation ensembles, where random axis-parallel cuts are subsequently applied to perform data partition.

Nevertheless, the cost of high power and energy combines with the high accuracy and training speed of the Deep Learning models. This is leading researchers to be aware of the environmental impact of deep neural architectures by trading off accuracy against energy consumption and also to perform characterization in terms of performance, power and energy for guiding the architecture design of DNN models [26, 27, 28, 29].

This paper aims to provide a contribution in this direction, and, in particular, to the field of Anomaly Detection by analyzing the behaviour of recent methods from the point of view of the detection performance as well as from the point of view of their carbon footprint. Specifically, we focus on the Latent*Out* algorithm [30, 31, 32, 33], an anomaly detection framework that applies to any deep neural architecture as a baseline to obtain a refined score, and we compare it with the baseline architecture on which it is applied and deep learning-based competitors from the other families.

2. The Latent*Out* algorithm for Unsupervised Anomaly Detection

Due to the quite good performances they obtained as well as their versatility, the ones based on (Variational) Autoencoders have become the most widespread Anomaly Detection approaches relying on Deep Neural Networks.

The main issue about them is that they often generalize so well to reconstruct also anomalies [30], thus worsening the capability of detecting anomalies of the reconstruction error.

In [31] Latent*Out* is introduced. It is a methodology that enhances both the reconstruction error and the latent space distribution of the Variational Autoencoder in order to obtain a refined anomaly score. Specifically, the first variant of the Latent*Out* (Figure 1) algorithm considers the enlarged feature space $F = L \times E$, where L represents the latent space and E is the reconstruction error space (usually $E \subseteq \mathbb{R}$), and performs a k-NN density estimation in the space F.

In Figure 1 the complete workflow of Latent*Out* is showed. Each point of the dataset $x \in X$ is mapped into the latent space L of the VAE (blue points represent inliers, red ones represent anomalies) by means of the encoder ϕ_W and then reconstructed back in the original space $\hat{x} \in X$ by means of the decoder ψ_W. Then, the reconstruction error $E(x) = \|x - \hat{x}\|_2^2$ is computed, the feature space $F = L \times E$ is created, and

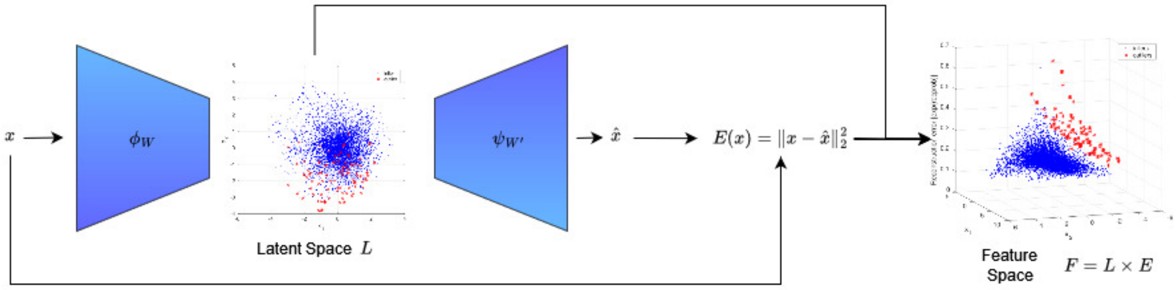

Figure 1: Latent*Out* receives the dataset as input and maps it into F. The transformed dataset is then processed by unsupervised anomaly detection methods which provide an anomaly score for each point.

the k-NN density estimation is performed in it to compute the Latent*Out* anomaly score.

The motivation behind this procedure is based on the observation that anomalies tend to lie in the sparsest regions of the augmented feature space F. This happens because even when their reconstruction error is not exceptionally large, is still significantly larger than that of their most similar normal items.

In [32] Latent*Out* has been expanded in order to be potentially applied to any neural architecture that has three fundamental properties:

- it outputs an *anomaly score*,
- it has a *latent space* L,
- it performs a mapping from the original data space X to L through an encoder-shaped module.

In particular, the neural models on which Latent*Out* has actually been tested are AE, VAE, GANomaly, Fast−AnoGAN, SO − GAAL, and MO − GAAL.

Moreover, in [33] it has been showed that the separation properties of the enlarged space F allow any generic anomaly score (not only the k-NN) to perform better when applied on it than on the input data space X.

3. Experimental results

3.1. Experimental setup

In our experiments we consider the tabular datasets *cardio*, *letter*, *lympho*, *mammography*, *pendigits*, *pima*, *satellite*, *satimage-2*, *speech*, *thyroid*, from the ODDS repository [34] as well as the image datasets MNIST [35], Fashion-MNIST [36], and CIFAR10 [37].

The last three datasets (differently from the ones from the ODDS repository) are multi-class, thus to make them suitable for the anomaly detection task we adopt a *one-vs-all* strategy, meaning that we consider one class as normal and we randomly sample s items from each other class. If not otherwise stated, we set $s = 10$. Specifically, we select the class "*0*" as normal for the MNIST dataset, the class "*Sandal*" for Fashion-MNIST, and the class "*deer*" for CIFAR-10.

As for the implementation details of the algorithm, we consider the original version of Latent*Out* with the VAE as baseline architecture, and the k-NN with $k = 50$ as estimator of the density of the feature space F. The latent space dimension ℓ of the VAE is set to $\ell = 2$ for tabular ODDS datasets and to $\ell = 32$ for image datasets. As for the encoder structure (the decoder is symmetric to it) we adopt the same strategy used in [33], i. e. we insert hidden layers of dimension $\ell_i = \left\lfloor \frac{d}{4^i} \right\rfloor$ between the input d-dimensional space and the ℓ-dimensional latent space for each $i \in \mathbb{N}^+$ such that $\left\lfloor \frac{d}{4^i} \right\rfloor > \ell$.

The CO_2 emissions are estimated by means of the Python library *CodeCarbon* [38] which bases its tracking on the power consumption and the geographic location where the code is executed.

3.2. Evolution of performance and emissions of Latent*Out* and VAE during training

The energy consumption of any Deep Learning model is related to the training phase, and, in particular, to the number of training epochs.

Therefore, it is of crucial importance to understand the behavior of these algorithms as the training proceeds to optimize the trade-off between the maximization of the performance and the minimization of energy consumption.

The quantity of CO_2 produced by Latent*Out*, which we represent as $\mathscr{E}_{\text{Latent}Out}$, is fundamentally constituted by two terms:

- the emissions $\mathscr{E}_{VAE}$ needed for the training of the architecture and the computation, which is shared with the Variational Autoencoder,
- the emissions $\mathscr{E}_{k\text{-NN}}$ used for the building of the feature space $\mathscr{F}$ and the computation of the k-NN algorithm in it.

Since the two operations are carried out in sequence and independently of each other, we have that

$$\mathscr{E}_{\text{Latent}Out} = \mathscr{E}_{VAE} + \mathscr{E}_{k\text{-NN}}$$

which means that, with equal training epochs, the carbon footprint of Latent*Out* is always greater than the one of the Variational Autoencoder. Thus, for a fair comparison, we train the Variational Autoencoder for 100 epochs and we stop the training earlier for evaluating the Latent*Out* score.

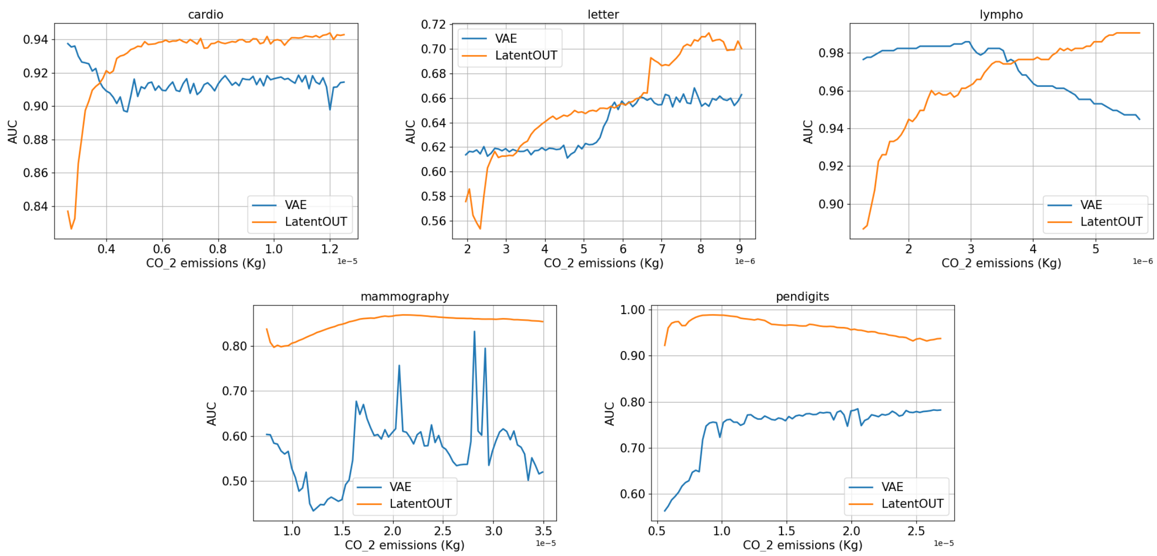

Figure 2: Comparison between the performances of the Variational Autoencoder and Latent*Out* in terms of AUC during the training epochs. ODDS datasets, group 1.

In figures 2, 3, 4, we show the performances of both Latent*Out* (in orange) and the standard Variational Autoencoder (in blue) in terms of Area Under the ROC Curve (AUC) as the training proceeds. Observe that on the horizontal axis is reported the CO_2 emissions (in Kg), which means that, for the reasons stated above, each value of the AUC of Latent*Out* is obtained with fewer epochs than the relative value of the VAE.

As we can see, in almost every plot the curve of Latent*Out* is placed above the curve of the VAE. Moreover, the trend of Latent*Out* is much more regular than the one of the VAE (see in particular the plots of the datasets *cardio*, *mammography*, *satellite*, *satimage-2*, *mnist*, *cifar*). This implies that if we fix a threshold on the amount of CO_2 we want to emit, the score of Latent*Out* always outperforms the standard score of the VAE. In other words, Latent*Out* is able to better exploit the emissions produced than the standard architecture on which it is applied.

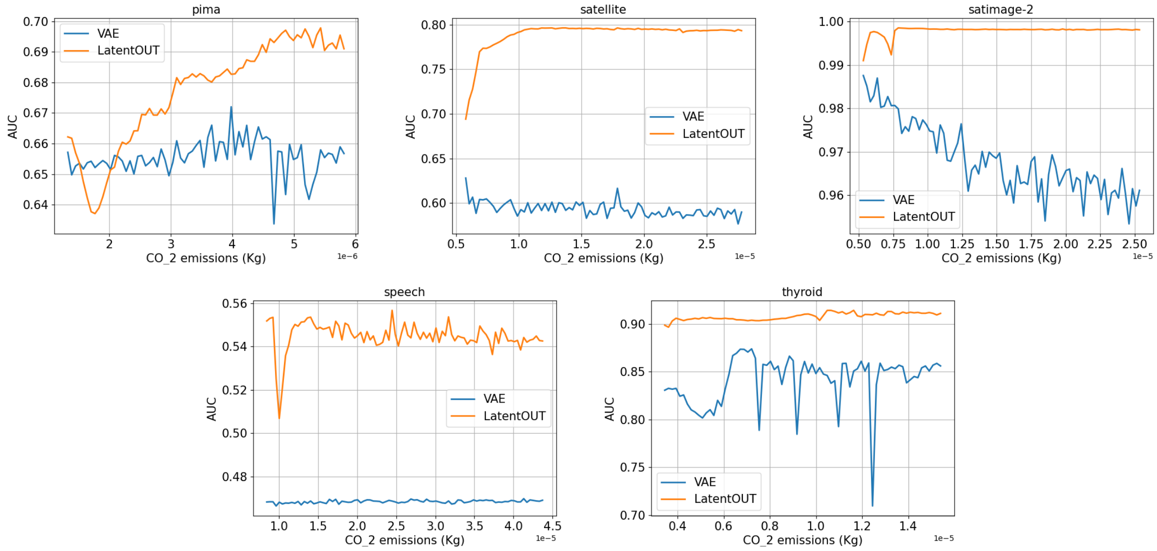

Figure 3: Comparison between the performances of the Variational Autoencoder and Latent*Out* in terms of AUC during the training epochs. ODDS datasets, group 2.

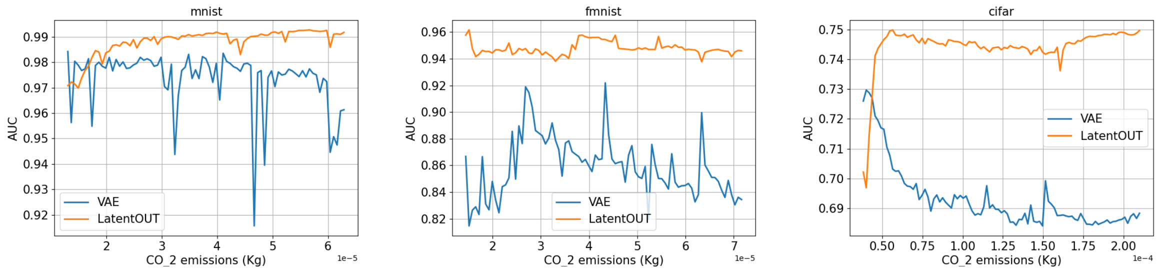

Figure 4: Comparison between the performances of the Variational Autoencoder and Latent*Out* in terms of AUC during the training epochs. MNIST, Fashion-MNIST and CIFAR10 datasets.

This happens because as the training proceeds the reconstruction capabilities of the VAE improve so much that at some point it becomes able to reconstruct also outliers, thus lowering the anomaly detection performances of the model. On the other side Latent*Out* benefits of the latent space organization that produces a progressively better separation between normal examples and anomalies in the feature space F.

3.3. Comparison with competitors

We consider as competitors some of the neural networks algorithm implemented in the Python library *PyOD* [39], namely Deep-SVDD [21], from the SVM-like family, AnoGAN [17] and ALAD [20], from the GAN family, and DIF [25]. For the implementation details (number of layers and neurons, training epochs, learning rate, potential hyperparameters), we refer to the default values fixed in PyOD. As for Latent*Out*, we consider again the setup described in section 3.1 and we perform a few-epochs training, due to the good convergence properties observed in the last section. Specifically, the VAE is trained for 15 epochs.

As evaluation metrics we adopt the standard Area Under the ROC Curve (AUC) and the ratio $\frac{CO_2}{AUC}$ between the emissions of CO_2 (in Kg) produced for the training and the inference of a model, and the AUC. This last value is a measure combining both performance and energy consumption, indeed it indicates how much CO_2 is needed (on average) to obtain a single percentage point of AUC.

Table 1 shows the results in terms of AUC. As we can see, Latent*Out* is the best method for half the datasets, achieving performances close to the best also in the other half. In particular, confirming the

Dataset (d)	Latent*Out*	Deep-SVDD	AnoGAN	ALAD	DIF
cardio (21)	0.9300	**0.9509**	0.4460	0.4885	0.9129
letter (32)	0.6206	0.5189	0.5118	0.5094	**0.6557**
lympho (18)	0.9495	0.9460	**0.9847**	0.6549	0.8650
mammography (6)	0.8326	**0.8767**	0.1366	0.5450	0.7415
pendigits (16)	**0.9880**	0.9748	0.9729	0.4785	0.9363
pima (8)	0.6598	0.6289	**0.7571**	0.5472	0.6071
satellite (36)	**0.7911**	0.6460	0.5432	0.4037	0.7574
satimage-2 (36)	**0.9984**	0.9682	0.0165	0.4292	0.9935
speech (400)	**0.5504**	0.4968	0.4658	0.4906	0.4633
thyroid (6)	0.9055	0.8743	0.8967	0.4837	**0.9613**
MNIST (28×28)	**0.9863**	0.9321	0.2176	0.3350	0.9572
Fashion-MNIST (28×28)	**0.9444**	0.9392	0.6634	0.6623	0.6269
CIFAR-10 ($32 \times 32 \times 3$)	**0.7474**	0.6624	0.5756	0.5363	0.6383

Table 1
Comparison with competitors in terms of AUC.

Dataset (d)	Latent*Out*	Deep-SVDD	AnoGAN	ALAD	DIF
cardio (21)	**4.7158e-6**	9.6679e-6	1.2619e-3	2.0648e-5	4.0021e-5
letter (32)	**5.7428e-6**	1.8790e-5	1.3014e-3	1.9605e-5	5.6887e-5
lympho (18)	**2.6640e-6**	2.9348e-6	5.2290e-5	1.3394e-5	9.8577e-6
mammography (6)	**1.5830e-5**	4.8771e-5	2.4759e-2	2.9251e-5	1.7729e-4
pendigits (16)	**9.2478e-6**	3.7444e-5	2.1541e-3	2.7159e-5	1.0738e-4
pima (8)	4.1708e-6	9.1278e-6	**1.9493e-6**	1.6284e-5	3.3011e-5
satellite (36)	**1.1943e-5**	4.0915e-5	4.7031e-3	3.1390e-5	1.2655e-4
satimage-2 (36)	**9.1152e-6**	2.4921e-5	1.4122e-1	2.9071e-5	8.5686e-5
speech (400)	**1.9139e-5**	5.9722e-5	4.3628e-3	5.4631e-5	1.7098e-4
thyroid (6)	**7.5721e-6**	1.9487e-5	1.2720e-3	2.2425e-5	5.6633e-5
MNIST (28×28)	**2.1834e-5**	3.7648e-5	1.7076e-2	8.5111e-5	1.3168e-4
Fashion-MNIST (28×28)	**2.3119e-5**	4.6431e-5	5.5211e-3	3.7217e-5	1.9408e-4
CIFAR-10 ($32 \times 32 \times 3$)	**4.9952e-5**	6.9862e-5	7.7896e-3	5.8859e-5	2.1652e-4

Table 2
Comparison with competitors in terms of $\frac{CO_2}{AUC}$.

observation made in [31], Latent*Out* is especially effective on higher dimensional, structured data (for example *speech* and the image datasets). In Table 2 are shown the results of the experiment in terms of the ratio $\frac{CO_2}{AUC}$. Here, Latent*Out* outperforms its competitors in all but one dataset, exhibiting the best trade-off between performances obtained and the emissions of CO_2 produced.

4. Conclusion

In this paper, we have focused on the algorithm Latent*Out* for unsupervised anomaly detection in order to evaluate its performances and measure the environmental impact of its executions. When compared to the standard architecture on which it is applied, i. e. the Variational Autoencoder, Latent*Out* shows that low energy-consumptive training can lead it to conspicuously better results. Moreover, in comparison with other neural network-based anomaly detection approaches it has shown superior performances both in terms of absolute AUC and, most importantly, in terms of the ratio between the emitted CO_2 and the AUC obtained.

As future development, we intend to expand the discussion about the environmental impact of Latent*Out* by including a more profound analysis of all its several variants and an investigation specialized on the hardware type (e.g., CPU vs. GPU), as well as propose novel measures to better capture the trade-off between emissions and performances. Finally, as a more ambitious goal, we aim at introducing a mechanism enabling Latent*Out* to consider the green-aware aspect at training time.

Acknowledgments

We acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), Spoke 9 - Green-aware AI, under the NRRP MUR program funded by the NextGenerationEU.

References

[1] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, K. Müller, A unifying review of deep and shallow anomaly detection, Proc. IEEE 109 (2021) 756–795.

[2] L. Davies, U. Gather, The identification of multiple outliers, Journal of the American Statistical Association 88 (1993) 782–792.

[3] E. Knorr, R. Ng, V. Tucakov, Distance-based outlier: algorithms and applications, VLDB Journal 8 (2000) 237–253.

[4] F. Angiulli, C. Pizzuti, Outlier mining in large high-dimensional data sets, IEEE Trans. Knowl. Data Eng. 2 (2005) 203–215.

[5] F. Angiulli, S. Basta, C. Pizzuti, Distance-based detection and prediction of outliers, IEEE Trans. on Knowledge and Data Engineering 2 (2006) 145–160.

[6] F. Angiulli, F. Fassetti, DOLPHIN: an efficient algorithm for mining distance-based outliers in very large datasets, ACM Trans. Knowl. Disc. Data (TKDD) 3(1) (2009) Article 4.

[7] M. M. Breunig, H. Kriegel, R. Ng, J. Sander, Lof: Identifying density-based local outliers, in: Proc. Int. Conf. on Managment of Data (SIGMOD), 2000.

[8] W. Jin, A. Tung, J. Han, Mining top-n local outliers in large databases, in: Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD), 2001.

[9] V. Hautamäki, I. Kärkkäinen, P. Fränti, Outlier detection using k-nearest neighbour graph, in: International Conference on Pattern Recognition (ICPR), Cambridge, UK, August 23-26, 2004, pp. 430–433.

[10] M. Radovanović, A. Nanopoulos, M. Ivanović, Reverse nearest neighbors in unsupervised distance-based outlier detection, IEEE Transactions on Knowledge and Data Engineering 27 (2015) 1369–1382.

[11] F. Angiulli, CFOF: A concentration free measure for anomaly detection, ACM Transactions on Knowledge Discovery from Data (TKDD) 14 (2020) 4:1–4:53.

[12] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, R. C. Williamson, Estimating the support of a high-dimensional distribution, Neural Computation (2001).

[13] D. M. J. Tax, R. P. W. Duin, Support vector data description, Mach. Learn. (2004).

[14] R. Chalapathy, S. Chawla, Deep learning for anomaly detection: A survey, 2019. arXiv:1901.03407.

[15] S. Hawkins, H. He, G. Williams, R. Baxter, Outlier detection using replicator neural networks, in: International Conference on Data Warehousing and Knowledge Discovery (DAWAK), 2002, pp. 170–180.

[16] J. An, S. Cho, Variational autoencoder based anomaly detection using reconstruction probability, Technical Report 3, SNU Data Mining Center, 2015.

[17] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, G. Langs, Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, 2017. arXiv:1703.05921.

[18] S. Akcay, A. Atapour-Abarghouei, T. P. Breckon, Ganomaly: Semi-supervised anomaly detection via adversarial training, 2018. arXiv:1805.06725.

[19] Y. Liu, Z. Li, C. Zhou, Y. Jiang, J. Sun, M. Wang, X. He, Generative adversarial active learning for unsupervised outlier detection, IEEE Trans. Knowl. Data Eng. 32 (2020) 1517–1528.

[20] H. Zenati, M. Romain, C.-S. Foo, B. Lecouat, V. Chandrasekhar, Adversarially learned anomaly detection, in: 2018 IEEE International conference on data mining (ICDM), IEEE, 2018, pp. 727–736.

[21] L. Ruff, N. Görnitz, L. Deecke, S. A. Siddiqui, R. A. Vandermeulen, A. Binder, E. Müller, M. Kloft,

Deep one-class classification, in: J. G. Dy, A. Krause (Eds.), Proceedings of the 35th ICML 2018, Stockholm, Sweden, 2018.

[22] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K. Müller, M. Kloft, Deep semi-supervised anomaly detection, in: 8th ICLR 2020, Addis Ababa, Ethiopia, OpenReview.net, 2020.

[23] F. Angiulli, F. Fassetti, L. Ferragina, R. Spada, Cooperative deep unsupervised anomaly detection, in: Discovery Science - 25th International Conference, DS 2022, Montpellier, France, October 10-12, 2022, Proceedings, volume 13601 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 318–328.

[24] G. Pang, C. Shen, A. van den Hengel, Deep anomaly detection with deviation networks, in: A. Teredesai, V. Kumar, Y. Li, R. Rosales, E. Terzi, G. Karypis (Eds.), Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019, ACM, 2019, pp. 353–362. URL: https://doi.org/10.1145/3292500.3330871. doi:10.1145/3292500.3330871.

[25] H. Xu, G. Pang, Y. Wang, Y. Wang, Deep isolation forest for anomaly detection, IEEE Transactions on Knowledge and Data Engineering 35 (2023) 12591–12604.

[26] A. E. Brownlee, J. Adair, S. O. Haraldsson, J. Jabbo, Exploring the accuracy–energy trade-off in machine learning, in: 2021 IEEE/ACM International Workshop on Genetic Improvement (GI), IEEE, 2021, pp. 11–18.

[27] Y. Sun, Z. Ou, J. Chen, X. Qi, Y. Guo, S. Cai, X. Yan, Evaluating performance, power and energy of deep neural networks on cpus and gpus, in: Theoretical Computer Science: 39th National Conference of Theoretical Computer Science, NCTCS 2021, Yinchuan, China, July 23–25, 2021, Revised Selected Papers 39, Springer, 2021, pp. 196–221.

[28] R. Schwartz, J. Dodge, N. A. Smith, O. Etzioni, Green ai, Communications of the ACM 63 (2020) 54–63.

[29] R. Verdecchia, J. Sallou, L. Cruz, A systematic review of green ai, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 13 (2023) e1507.

[30] F. Angiulli, F. Fassetti, L. Ferragina, Improving deep unsupervised anomaly detection by exploiting vae latent space distribution, in: Discovery Science, Springer International Publishing, Cham, 2020, pp. 596–611.

[31] F. Angiulli, F. Fassetti, L. Ferragina, Latent*Out*: an unsupervised deep anomaly detection approach exploiting latent space distribution, Machine Learning (2022).

[32] F. Angiulli, F. Fassetti, L. Ferragina, Detecting anomalies with rmlatentout: Novel scores, architectures, and settings, in: M. Ceci, S. Flesca, E. Masciari, G. Manco, Z. W. Ras (Eds.), Foundations of Intelligent Systems - 26th International Symposium, ISMIS 2022, Cosenza, Italy, October 3-5, 2022, Proceedings, volume 13515 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 251–261. URL: https://doi.org/10.1007/978-3-031-16564-1_24. doi:10.1007/978-3-031-16564-1_24.

[33] F. Angiulli, F. Fassetti, L. Ferragina, Enhancing anomaly detectors with latentout, Journal of Intelligent Information Systems (2023) 1–19.

[34] S. Rayana, Odds library, 2016. URL: http://odds.cs.stonybrook.edu.

[35] L. Deng, The mnist database of handwritten digit images for machine learning research, IEEE Signal Processing Magazine 29 (2012) 141–142.

[36] H. Xiao, K. Rasul, R. Vollgraf, Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, CoRR abs/1708.07747 (2017). URL: http://arxiv.org/abs/1708.07747. arXiv:1708.07747.

[37] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images (2009).

[38] B. Courty, V. Schmidt, S. Luccioni, Goyal-Kamal, MarionCoutarel, B. Feld, J. Lecourt, LiamConnell, A. Saboni, Inimaz, supatomic, M. Léval, L. Blanche, A. Cruveiller, ouminasara, F. Zhao, A. Joshi, A. Bogroff, H. de Lavoreille, N. Laskaris, E. Abati, D. Blank, Z. Wang, A. Catovic, M. Alencon, M. Stęchły, C. Bauer, L. O. N. de Araújo, JPW, MinervaBooks, mlco2/codecarbon: v2.4.1, 2024. URL: https://doi.org/10.5281/zenodo.11171501. doi:10.5281/zenodo.11171501.

[39] Y. Zhao, Z. Nasrullah, Z. Li, Pyod: A python toolbox for scalable outlier detection, Journal of Machine Learning Research 20 (2019) 1–7. URL: http://jmlr.org/papers/v20/19-011.html.