Cooperative Deep Unsupervised Anomaly Detection

Fabrizio Angiulli(✉), Fabio Fassetti, Luca Ferragina, and Rosaria Spada

DIMES, University of Calabria, 87036 Rende (CS), Italy
{f.angiulli,f.fassetti,l.ferragina}@dimes.unical.it

Abstract. In last years deep learning approaches to anomaly detection are becoming very popular. In most of the first methods the paradigm is to train neural networks initially designed for compression (Auto Encoders) or data generation (GANs) and to detect anomalies as a collateral result. Recently new architectures have been introduced in which the expressive power of deep neural networks is associated with objective functions specifically designed for anomaly detection. One of these methods is *Deep-SVDD* which, although created for One-Class classification, has been successfully applied to the (semi-)supervised anomaly detection setting. Technically, *Deep-SVDD* technique forces the deep latent representation of the input data to be enclosed into an hypersphere and labels as anomalies data farthest from its center. In this work we introduce *Deep-UAD*, a neural network approach for unsupervised anomaly detection where, iteratively, a network similar to that of *Deep-SVDD* is alternatively trained with an Auto Encoder and the two networks share some weights in order for each network to improve its training by exploiting the information coming from the other network. The experiments we conducted show that the performances obtained by the proposed method are better than the ones obtained both by deep learning methods and standard shallow algorithms.

1 Introduction

Anomaly detection is a fundamental data mining task whose aim is to isolate samples in a dataset that are suspected of being generated by a distribution different from the rest of the data. The presence of anomalies is due to many reasons like mechanical faults, fraudulent behavior, human errors, instrument error or simply through natural deviations in populations.

Depending on the composition of the dataset, anomaly detection settings can be classified as unsupervised, semi-supervised, and unsupervised [1,14]. In the supervised setting the training data are labeled as normal and abnormal and and the goal is to build a classifier. The difference with standard classification problems is that abnormal data form a rare class. In the semi-supervised setting,

P. Pascal and D. Ienco (Eds.): DS 2022, LNAI 13601, pp. 318–328, 2022.
https://doi.org/10.1007/978-3-031-18840-4_23

the training set is composed by both labelled and unlabelled data. A special case of this setting is the one-class classification when we have a training set composed only by normal class items. In the unsupervised setting the goal is to detect outliers in an input dataset by assigning a score or anomaly degree to each object. Several statistical, data mining and machine learning approaches have been proposed to detect anomalies, namely, statistical-based [11,15], distance-based [6,9,10,24], density-based [12,21], reverse nearest neighbor-based [4,5,19,25], SVM-based [30,33], and many others [1,14].

In last years deep learning-based methods for anomaly detection [13,17,27] have shown great performances. Auto encoder(AE) based anomaly detection [3,13,20] consists in training an AE to reconstruct a set of examples and then to detect as anomalies those data that show a large reconstruction error. *Variational auto encoders* (VAE) arise as a variant of standard auto encoders designed for *generative* purposes [23]. The key idea of VAEs is to encode each example as a normal distribution over the latent space and regularize the loss by maximizing similarity of these distributions with the standard normal one. Due to similarities to standard auto encoders, VAEs have also been used to detect anomalies. However, it has been noticed that VAEs share with standard AEs the problem that they generalize so well that they can also well reconstruct anomalies [3,7,8,13,22,32]. Generative Adversarial Networks (GAN) [18] are another tool for generative purposes, aiming at learning an unknown distribution by means of an adversarial process involving a discriminator, that outputs the probability for an observation to be generated by the unknown distribution, and a generator, mapping points coming from a standard distribution to points belonging to the unknown one. GANs have also been employed with success to the anomaly detection task [2,16,29,31,34].

Some authors [26,28] have recently observed that all the above mentioned anomaly detection deep learning based methods are not designed to directly discover anomalies, but their main task is data reconstruction (AE and VAE) or data generation (GAN) and anomaly detection is a collateral result. They introduce new methods, called *Deep-SVDD* and *Deep-SAD*, that combine the expressive power of deep neural networks with a loss inspired from SVM-based methods and specifically designed for anomaly detection. These methods are used for one-class and (semi-)supervised settings but we argue that they do not apply very naturally to the unsupervised setting, thus we introduce *Deep-UAD*, a new unsupervised method that deeply modifies the architectures in [26,28]. In particular we build a new training paradigm for the network in [26] that involves an AE which is trained alternatively with the network and with which the network and exchange the information they obtained during the training. This is done by modifying the losses of both the network and the AE. The proposed approach shows sensible improvements in terms of detection performances over both the standard approach in [26,28] and the baseline shallow methods.

The rest of the paper is organized as follows. Section 2 discusses related work with particular emphasis on *Deep-SVDD* and *Deep-SAD*. Section 3 introduces the *Deep-UAD* unsupervised anomaly detection algorithm. Section 4 illustrates experimental results. Finally, Sect. 5 concludes the work.

2 Preliminaries

In this Section we deepen auto encoder and *Deep-SVDD* which are exploited by our technique as basic components and suitably modified to our purposes.

Auto Encoder. An auto encoder (AE) is a neural network architecture successfully employed for anomaly detection [20]. It aims at providing a reconstruction of the input by exploiting a dimensionality reduction step (the *encoder* ϕ_W) followed by a step mapping back from the compressed space (the *latent space*) to the original space (the *decoder* $\psi_{W'}$). Its ability in detecting anomalies depends on the observation that regularities should be better compressed and, hopefully, better reconstructed [20]. The AE loss is $\mathcal{E}(x) = \|x - \hat{x}\|_2^2$, where $\hat{x} = \psi_{W'}(\phi_W(x))$, and coincides with the standard reconstruction error.

One-class SVM. Before discussing *Deep-SVDD* some preliminary notions about One-Class SVM (OC-SVM) [30] are needed. The original OC-SVM method is designed for the one-class setting and has the objective of finding the hyperplane in a feature space that best separates the mapped data from the origin. Given the data $\{x_1, \ldots, x_n\} \subseteq X$, it is defined by the following optimization problem

$$\begin{aligned} \min_{\mathbf{w},\rho,\xi_i} \; & \frac{1}{2}\|\mathbf{w}\|_F^2 - \rho + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i \\ \text{s. t. } & \langle \phi(x_i), \mathbf{w} \rangle \geq \rho - \xi_i, \\ & \xi_i \geq 0, \quad i = 1, \ldots, n \end{aligned}$$

where ρ is the distance from the origin to the hyperplane $\mathbf{w} \in F$, ξ_i are slack variables and $\nu \in (0, 1]$ is a trade-off hyperparameter. The points in the test set are labelled as normal if they are mapped inside the hyperplane and anomalous if they are mapped outside. Related to OC-SVM, Support Vector Data Descriptor (SVDD) [33] is a method that has the aim of enclosing the input data into a hypersphere of minimum radius. The relative optimization problem is

$$\begin{aligned} \min_{R,\mathbf{c},\xi_i} \; & R^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i \\ \text{s. t. } & \|\phi(x_i) - \mathbf{c}\| \leq R^2 + \xi_i, \\ & \xi_i \geq 0, \quad i = 1, \ldots, n \end{aligned}$$

where $R > 0$ and $\mathbf{c}$ are the radius and the center of the hypersphere and again ξ_i are slack variables and $\nu \in (0, 1]$ is a trade-off hyperparameter.

Deep-SVDD. In [26], authors apply the same idea expressed in SVDD of enclosing the data into an hypersphere performing the mapping into the feature space with the use of a deep neural network. In particular, let $\phi_W : X \to F$ be a mapping obtained with a neural network with weights $W = [W_1, \dots, W_L]$ (W_l are the weights relative to the layer $l \in \{1, \dots, L\}$) from the input space $X \subseteq \mathbb{R}^d$ to the output space $F \subseteq \mathbb{R}^k$, with $k < d$. The loss of the network is given by

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} \|\phi_W\left(x_i\right) - \mathbf{c}\|_2^2 + \frac{\lambda}{2} \sum_{l=1}^{L} \|W_l\|_F^2, \tag{1}$$

where the first term forces the network representation $\phi_W(x)$ to stay close to the center $\mathbf{c}$ of the hypersphere and the second term is a weight decay regularizer with hyperparameter $\lambda > 0$. This loss is used in a One-Class anomaly detection setting to map the training set (composed only by normal items) as close as possible to the center $\mathbf{c}$ so that in the testing phase the network is less able to map the anomalies close to $\mathbf{c}$. Because of this, it is defined as anomaly score of the point x the distance of its network representation from the center: $\mathcal{S}(x) = \|\phi_W(x) - \mathbf{c}\|_2^2$.

The center $\mathbf{c}$ is not a trainable parameter and is fixed before the training by means of an AE that is composed so that the encoding part has the same structure as the network ϕ and shares with it the weights W, the structure of the decoding part is symmetric to it and thus the latent space coincides with the space F. The training set is given in input to this AE which is trained with the standard loss and subsequently the center $\mathbf{c}$ is defined as $\mathbf{c} = \frac{1}{n} \sum_{i=1}^{n} \phi_W(x_i)$, that is the mean of the latent representations of all the points in the training set. The same architecture has been applied in [28] for the task of semi-supervised anomaly detection with the following natural adaptation of the loss

$$\mathcal{L} = \frac{1}{n+m} \sum_{i=1}^{n} \|\phi_W\left(x_i\right) - \mathbf{c}\|_2^2 + \frac{\eta}{n+m} \sum_{i=1}^{m} \left(\|\phi_W\left(\tilde{x}_i\right) - \mathbf{c}\|_2^2\right)^{\tilde{y}_i} + \frac{\lambda}{2} \sum_{l=1}^{L} \|W_l\|_F^2, \tag{2}$$

where $\tilde{x}_i$ are the m labeled data with the relative labels $\tilde{y}_i$ and η is an hyperparameter handling the trade-off between the contributions of labelled and unlabelled data. Let us observe that data labelled as normal ($\tilde{y}_i = +1$) are treated in the usual way which means that they are forced to be mapped close to $\mathbf{c}$ while for the anomalies ($\tilde{y}_i = -1$) the contribution is inverted and they are force to stay as far as possible from $\mathbf{c}$.

It is important to observe that (2) is designed to consider also unlabelled examples. An extreme case occurs when $m = 0$, when all the training data are unlabelled. This scenario is similar to the unsupervised setting but there is a substantial difference: in one case the objective is to detect anomalies in a test set, in the other the anomalies have to be detected among the same data used for the training phase. In this case the losses (2) and (1) coincide, which means that, even if originally the loss (1) has been designed to deal only with normal class items, it can be used in settings that involve the use of unlabeled anomalies in the training phase, thus it can be applicable also to the unsupervised settings.

3 Method

In this Section the technique *Deep-UAD* proposed in this paper is discussed.

A one class based technique, like *Deep-SVDD*, is aimed at building a model for the normal class exploiting input data by assuming they do not contain anomalies and classifying data of a test set. In particular, *Deep-SVDD* tends to map close to the center all the input data and, then, in the unsupervised setting this technique may fail in correctly separating normal and anomalous samples.

Deep-UAD tackles this issue by providing information to the network about the anomaly degree of each sample in order to force the network to approach normal data to the center and to let anomalies far from the center. This is accomplished by exploiting an AE that provides a level of anomaly suspiciousness. Thus, the proposed architecture consists in two components, a neural network *Deep-UAD$_{NET}$*, and an auto encoder *Deep-UAD$_{AE}$*; *Deep-UAD$_{NET}$* has the same structure of the network of *Deep-SVDD*, thus can be defined by the same mapping function ϕ_W, and it is forced to map the data badly reconstructed by *Deep-UAD$_{AE}$*, namely more suspected to be anomalous, faraway from the center and, conversely, data suggested as normal by *Deep-UAD$_{AE}$* close to the center. Technically, this is done by introducing this novel loss

$$\mathcal{L}_{\text{NET}} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\mathcal{E}\left(x_i\right)} \|\phi_W\left(x_i\right) - \mathbf{c}\|_2^2 + \frac{\lambda}{2} \sum_{l=1}^{L} \|W_l\|_F^2. \tag{3}$$

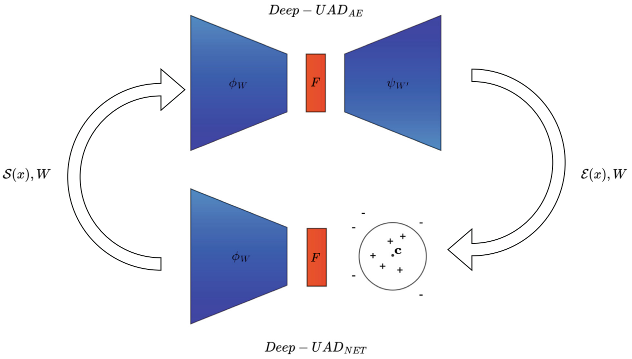

Fig. 1. Diagram of the *Deep-UAD* cooperative strategy: the network *Deep-UAD$_{NET}$* and the auto encoder *Deep-UAD$_{AE}$* refine their capabilities to find anomalies by sharing the encoder weights W and passing to each other the information of their own score.

It is inspired by Eq. (1) which is modified by inserting the term $\frac{1}{\mathcal{E}(x_i)}$, directly related to the probability for x_i to be an anomaly according to the AE, and it is used as a weight to control how much is important that the network representation of x_i is mapped close to $\mathbf{c}$. In particular, the smaller is $\mathcal{E}(x_i)$, namely

x_i is probably not an anomaly according to *Deep-UAD*$_{AE}$, the more higher is the weight and thus the network takes more advantage in mapping x_i close to the center; conversely if $\mathcal{E}(x_i)$ is large, x_i is suspected to be an anomaly by the AE, the weight is small and the network has a small advantage in bringing the representation of x_i close to the center.

The strategy of *Deep-UAD* consists in a preliminary phase where the AE, without information by the network *Deep-UAD*$_{NET}$, is trained with standard loss, the center of the hypershpere is computed and the reconstruction error $\mathcal{E}(x_i)$ is evaluated for each sample. Successively, two phases are iteratively executed, during the first one, the network *Deep-UAD*$_{NET}$ is trained with the loss (3) for a certain number of epochs and the score $\mathcal{S}(x_i)$ is calculated, during the second, *Deep-UAD*$_{AE}$ is trained for some epochs with the novel loss

$$\mathcal{L}_{\mathrm{AE}} = \sum_{i=1}^{n} \frac{1}{\mathcal{S}(x_i)} \|x_i - \hat{x_i}\|_2^2. \tag{4}$$

The purpose of $\mathcal{S}(x_i)$ is similar to the one of $\mathcal{E}(x_i)$ in (3), giving a weight to the contribution of each point x_i according to the results obtained by the network. The idea of *Deep-UAD* is that the score obtained from one network improves the training of the other one, the final anomaly score output is $\mathcal{S}(x_i)$.

4 Experimental Results

In this section we report experiments conducted to study the behavior of the proposed method. We focus on three main aspects, namely (*i*) the impact of the dimension of the output space on the performances, (*ii*) the analysis of the cooperative process as the iterations proceed, (*iii*) the comparison with other methods with specific emphasis on *Deep-SVDD*. In our experiments we consider two standard benchmark datasets composed by grayscale images, *MNIST*[1] and *Fashion-MNIST*[2]. They are both composed by 28 × 28 pixels images divided in 10 classes, thus, in order adapt them for anomaly detection, we adopt a *one-vs-all* policy, i.e. we consider one class as normal and all the others as anomalous. For each class, we create a dataset composed by all the examples of the selected class as normal and s random selected examples from each other class as anomalies.

Sensitivity Analysis on the Dimension K of the Output Space. In this section, our aim is to determine how the dimension of the output space F impacts on the behavior of both our method and the original *Deep-SVDD* algorithm. In order to do this we consider the MNIST dataset in the *one-vs-all* setting and, for each class, we train both models with k varying in the interval $[8, 64]$.

From Fig. 2, in which are reported the results after 5 runs, we can see that for both *Deep-UAD* (in red) and *Deep-SVDD* (in black) the trend is increasing

[1] http://yann.lecun.com/exdb/mnist/.

[2] https://github.com/zalandoresearch/fashion-mnist.

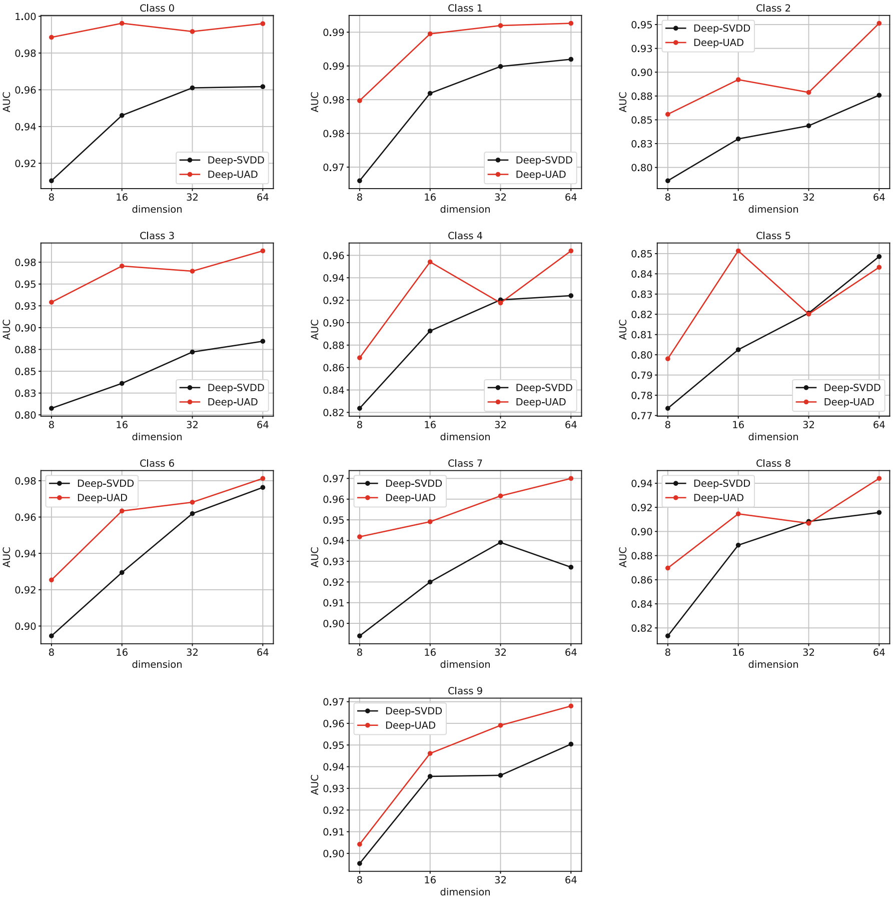

Fig. 2. MNIST dataset ($s = 10$): AUCs of *Deep-UAD* and *Deep-SVDD* varying the dimension of the final space.

which means that a small dimensional space F is not sufficient, in both cases, to separate the anomalies from the normal examples. Moreover it is important to point out that the performances achieved by our method are better than the ones obtained by *Deep-SVDD* for almost each class and each value of k.

Analysis of the Iterative Process. *Deep-UAD* is based on an iterative process in which the network *Deep-UAD*$_{NET}$ and the auto encoder *Deep-UAD*$_{AE}$ share information, because of this it is crucial to investigate how the number of iterations affects the performances of both the architectures. We do this by considering MNIST and Fashion-MNIST datasets in the *one-vs-all* setting, performing 5 runs for each class and computing the AUC for each iteration. In each iteration both the network and the AE are trained for 25 epochs.

In Fig. 3 are reported the trends of the two architectures. As we can see, they are always non decreasing, which means that both the architectures are taking advantage of the cooperative strategy. For what concerns *Deep-UAD*$_{NET}$, which is the one that outputs the score of *Deep-UAD*, the trend becomes substantially stable and constant, sometimes from the very first iteration (as class 0 of MNIST and class *Sandal* of Fashion-MNIST) and other times after a slightly bigger number of iterations (like classes 2 and 7 of MNIST). This means that the parameter of the number of iterations is not hard to fix, since a number around one ten of iterations guarantees always the achievement of a score of *Deep-UAD* close to best possible and an improvement over *Deep-SVDD*.

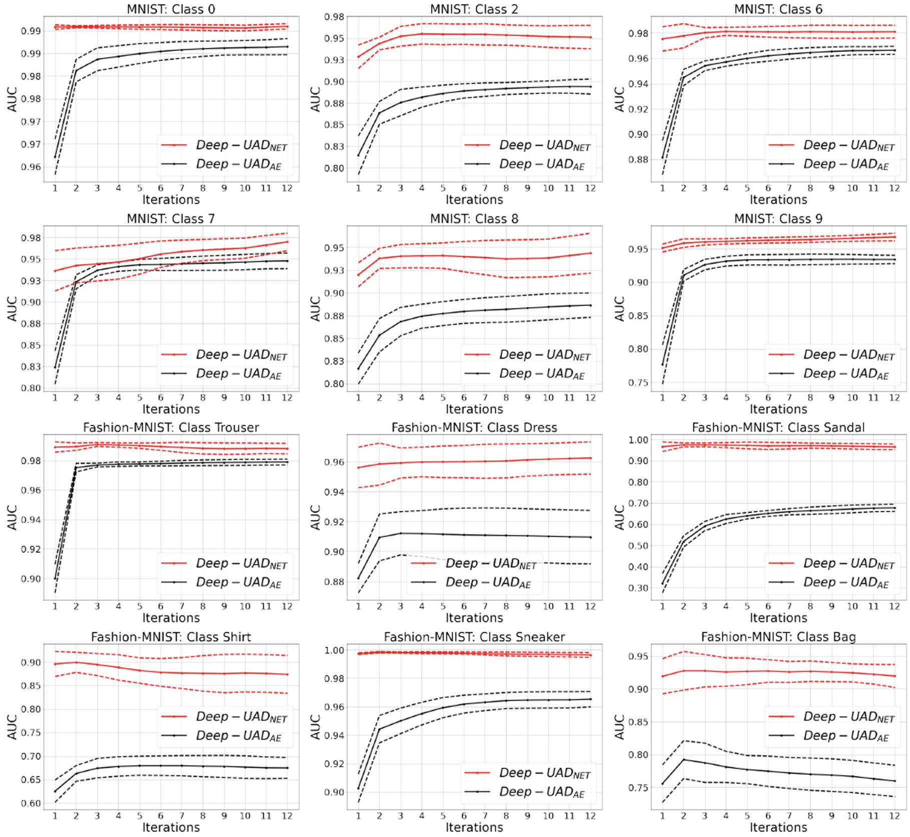

Fig. 3. MNIST and Fashion-MNIST datasets ($s = 10$): AUCs of *Deep-UAD* and AE varying the iterations of the method.

Moreover, *Deep-UAD*$_{AE}$ improves its performances as the iterations proceed. This fact is crucial for the behavior of the whole process, indeed it means that the information provided to *Deep-UAD*$_{NET}$ by *Deep-UAD*$_{AE}$ becomes better at

Table 1. AUC of *Deep-UAD* and competitors on MNIST and Fashion-MNIST with $s = 10$ on the left and $s = 100$ on the right.

MNIST										
	$s = 10$					$s = 100$				
Class	***Deep-SVDD***	***Deep-UAD***	**DCAE**	**IF**	**KNN**	***Deep-SVDD***	***Deep-UAD***	**DCAE**	**IF**	**KNN**
0	.962 ± .018	**.996 ± .001**	.953 ± .008	.951 ± .010	.978 ± .010	.901 ± .015	**.948 ± .006**	.918 ± .005	.868 ± .023	.841 ± .006
1	.991 ± .001	.996 ± .002	.925 ± .013	.994 ± .001	**.998 ± .001**	.974 ± .004	.989 ± .003	.878 ± .014	.991 ± .001	**.996 ± .002**
2	.876 ± .036	**.951 ± .013**	.822 ± .017	.731 ± .024	.891 ± .015	.765 ± .020	**.861 ± .026**	.746 ± .016	.678 ± .028	.703 ± .014
3	.884 ± .014	**.988 ± .006**	.805 ± .026	.804 ± .028	.906 ± .011	.794 ± .027	**.878 ± .026**	.759 ± .027	.762 ± .012	.766 ± .006
4	.924 ± .017	**.964 ± .011**	.772 ± .019	.869 ± .018	.928 ± .008	.827 ± .016	**.918 ± .012**	.698 ± .031	.836 ± .008	.831 ± .007
5	.848 ± .025	.843 ± .037	.728 ± .016	.752 ± .019	**.917 ± .019**	.732 ± .017	**.809 ± .021**	.650 ± .040	.709 ± .020	.765 ± .007
6	.975 ± .007	**.981 ± .005**	.887 ± .019	.894 ± .017	.970 ± .006	.920 ± .014	**.956 ± .007**	.793 ± .018	.848 ± .016	.858 ± .004
7	.927 ± .017	**.970 ± .008**	.853 ± .015	.909 ± .009	.957 ± .007	.875 ± .014	**.946 ± .005**	.781 ± .013	.891 ± .009	.893 ± .003
8	.916 ± .018	**.944 ± .014**	.828 ± .012	.729 ± .021	.857 ± .010	.852 ± .013	**.933 ± .010**	.728 ± .020	.710 ± .014	.731 ± .009
9	.950 ± .009	**.968 ± .014**	.798 ± .019	.872 ± .013	.947 ± .009	.908 ± .011	**.959 ± .005**	.734 ± .012	.858 ± .007	.862 ± .005
Fashion-MNIST										
	$s = 10$					$s = 100$				
Class	***Deep-SVDD***	***Deep-UAD***	**DCAE**	**IF**	**KNN**	***Deep-SVDD***	***Deep-UAD***	**DCAE**	**IF**	**KNN**
0	.868 ± .016	**.929 ± .021**	.793 ± .014	.909 ± .010	.902 ± .010	.770 ± .031	.866 ± .011	.748 ± .012	**.886 ± .008**	.808 ± .006
1	.975 ± .004	**.991 ± .002**	.934 ± .014	.977 ± .003	.987 ± .002	.956 ± .003	**.985 ± .002**	.902 ± .022	.976 ± .002	.967 ± .002
2	.832 ± .016	**.911 ± .013**	.691 ± .021	.873 ± .015	.882 ± .011	.757 ± .013	.828 ± .029	.581 ± .025	**.842 ± .009**	.779 ± .006
3	.923 ± .008	**.963 ± .009**	.898 ± .013	.936 ± .009	.937 ± .011	.867 ± .013	**.955 ± .010**	.858 ± .008	.936 ± .003	.838 ± .004
4	.894 ± .015	**.963 ± .016**	.852 ± .019	.911 ± .014	.888 ± .013	.827 ± .029	**.918 ± .006**	.780 ± .029	.903 ± .005	.807 ± .013
5	.817 ± .025	**.966 ± .013**	.373 ± .041	.928 ± .008	.846 ± .017	.647 ± .041	.801 ± .038	.235 ± .022	**.907 ± .005**	.611 ± .011
6	.756 ± .020	**.874 ± .022**	.619 ± .020	.812 ± .013	.813 ± .012	.691 ± .021	**.781 ± .017**	.537 ± .031	.778 ± .009	.710 ± .005
7	.978 ± .004	**.996 ± .005**	.905 ± .007	.980 ± .007	.979 ± .004	.927 ± .014	**.984 ± .002**	.807 ± .016	.978 ± .002	.915 ± .005
8	.893 ± .015	**.920 ± .018**	.778 ± .016	.886 ± .017	.817 ± .021	.733 ± .025	**.854 ± .017**	.644 ± .025	.822 ± .002	.512 ± .007
9	.980 ± .004	**.991 ± .005**	.970 ± .010	.978 ± .006	.945 ± .013	.917 ± .016	**.983 ± .003**	.912 ± .014	.968 ± .004	.762 ± .014

every iteration, thus *Deep-UAD* succeeds in mapping the anomalies away from the center better than *Deep-SVDD*, when this information is missing and several anomalies are not detected being closer to the center than some normal samples with consequent worsening of the AUC.

Comparison with Competitors. Finally, in this last section, we compare the results *Deep-UAD* with competitors on MNIST and Fashion-MNIST. The methods taken into account are Isolation Forest (IF) and k-Nearest Neighbor as shallow algorithms and *Deep-SVDD* and Deep Convolutional auto encoder (DCAE) as deep learning methods. To ensure a fair comparison, both *Deep-SVDD* and DCAE have the same structure of *Deep-UAD* and for both of them, as well as for our method, we fix $k = 64$ according to the results of the first experiment.

In Table 1 are reported the results for both datasets with $s = 10$ and $s = 100$. We can see that for almost all classes *Deep-UAD* performs better than all considered competitors and in certain cases the differences with their performances are huge. In particular, in the direct comparison with *Deep-SVDD*, the technique that inspires our method, *Deep-UAD* is always winning, meaning that the cooperative work of the network and the AE succeeds in improving the ability of isolating anomalies.

5 Conclusions

In this work is presented *Deep-UAD*, a deep learning approach for unsupervised anomaly detection. It is based on an alternate and cooperative training of an

AE and a neural network aiming at mapping the data close to a fixed center in the output space. Experimental results show that *Deep-UAD* achieves good performances and that the strategy of alternate training brings benefits to both the neural network and the AE improving their capabilities to isolate anomalies.

In the future our main goals are to investigate the application of a cooperative alternate strategy similar to this one to more complex neural architectures, to study possible modifications to the discussed method that may help in improving performances, and to test our algorithm on dataset of different size and nature.

References

1. Aggarwal, C.C.: Outlier Analysis. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-47578-3
2. Akcay, S., Atapour-Abarghouei, A., Breckon, T.P.: GANomaly: semi-supervised anomaly detection via adversarial training (2018)
3. An, J., Cho, S.: Variational autoencoder based anomaly detection using reconstruction probability. Technical Report 3, SNU Data Mining Center (2015)
4. Angiulli, F.: Concentration free outlier detection. In: European Conference on Machine Learning and Knowledge Discovery in Databases, Skopje, Macedonia (2017)
5. Angiulli, F.: CFOF: a concentration free measure for anomaly detection. ACM Trans. Knowl. Discov. Data (TKDD) **14**(1), 4:1-4:53 (2020)
6. Angiulli, F., Basta, S., Pizzuti, C.: Distance-based detection and prediction of outliers. IEEE Trans. Knowl. Data Eng. **2**(18), 145–160 (2006)
7. Angiulli, F., Fassetti, F., Ferragina, L.: Improving deep unsupervised anomaly detection by exploiting VAE latent space distribution. In: Discovery Science (2020)
8. Angiulli, F., Fassetti, F., Ferragina, L.: Latent*Out*: an unsupervised deep anomaly detection approach exploiting latent space distribution. Machine Learning (2022). https://doi.org/10.1007/s10994-022-06153-4
9. Angiulli, F., Pizzuti, C.: Fast outlier detection in large high-dimensional data sets. In: Principles of Data Mining and Knowledge Discovery (PKDD) (2002)
10. Angiulli, F., Pizzuti, C.: Outlier mining in large high-dimensional data sets. IEEE Trans. Knowl. Data Eng. **2**(17), 203–215 (2005)
11. Barnett, V., Lewis, T.: Outliers in Statistical Data. Wiley (1994)
12. Breunig, M.M., Kriegel, H., Ng, R., Sander, J.: LOF: identifying density-based local outliers. In: Proceedings of the International Conference on Managment of Data (SIGMOD) (2000)
13. Chalapathy, R., Chawla, S.: Deep learning for anomaly detection: a survey (2019)
14. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. ACM Comput. Surv. **41**(3), 1–15 (2009)
15. Davies, L., Gather, U.: The identification of multiple outliers. J. Am. Statist. Assoc. **88**, 782–792 (1993)
16. Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial feature learning (2017)
17. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
18. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, vol. 27 (2014)
19. Hautamäki, V., Kärkkäinen, I., Fränti, P.: Outlier detection using k-nearest neighbour graph. In: ICPR, Cambridge, UK (2004)

20. Hawkins, S., He, H., Williams, G., Baxter, R.: Outlier detection using replicator neural networks. In: International Conference on Data Warehousing and Knowledge Discovery (DAWAK), pp. 170–180 (2002)
21. Jin, W., Tung, A., Han, J.: Mining top-n local outliers in large databases. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD) (2001)
22. Kawachi, Y., Koizumi, Y., Harada, N.: Complementary set variational autoencoder for supervised anomaly detection. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2366–2370 (2018)
23. Kingma, D.P., Welling, M.: Auto-encoding variational bayes (2013)
24. Knorr, E., Ng, R., Tucakov, V.: Distance-based outlier: algorithms and applications. VLDB J. **8**(3–4), 237–253 (2000)
25. Radovanović, M., Nanopoulos, A., Ivanović, M.: Reverse nearest neighbors in unsupervised distance-based outlier detection. IEEE Trans. Knowl. Data Eng. **27**(5), 1369–1382 (2015)
26. Ruff, L., et al.: Deep one-class classification. In: Proceedings of the 35th ICML, Stockholm, Sweden (2018)
27. Ruff, L., et al.: A unifying review of deep and shallow anomaly detection. Proc. IEEE **109**(5), 756–795 (2021)
28. Ruff, L., et al.: Deep semi-supervised anomaly detection. In: 8th ICLR, Addis Ababa, Ethiopia. OpenReview.net (2020)
29. Schlegl, T., Seeböck, P., Waldstein, S., Langs, G., Schmidt-Erfurth, U.: f-AnoGAN: fast unsupervised anomaly detection with generative adversarial networks. In: Medical Image Analysis 54 (2019)
30. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural Comput. **13**(7), 1443–1471 (2001)
31. Sáinchez-Martín, P., Olmos, P.M., Perez-Cruz, F.: Improved BIGAN training with marginal likelihood equalization (2020)
32. Sun, J., Wang, X., Xiong, N., Shao, J.: Learning sparse representation with variational auto-encoder for anomaly detection. IEEE Access **6**, 33353–33361 (2018)
33. Tax, D.M.J., Duin, R.P.W.: Support vector data description. Mach. Learn. **54**, 45–66 (2004). https://doi.org/10.1023/B:MACH.0000008084.60811.49
34. Zenati, H., Foo, C.S., Lecouat, B., Manek, G., Chandrasekhar, V.R.: Efficient GAN-based anomaly detection (2019)