

# Anomaly Detection in Astrophysics: A Comparison Between Unsupervised Deep and Machine Learning on KiDS Data



Maurizio D'Addona, Giuseppe Riccio, Stefano Cavuoti, Crescenzo Tortora, and Massimo Brescia

**Abstract** Every field of Science is undergoing unprecedented changes in the discovery process, and Astronomy has been a main player in this transition since the beginning. The ongoing and future large and complex multi-messenger sky surveys impose a wide exploiting of robust and efficient automated methods to classify the observed structures and to detect and characterize peculiar and unexpected sources. We performed a preliminary experiment on KiDS DR4 data, by applying to the problem of anomaly detection two different unsupervised machine learning algorithms, considered as potentially promising methods to detect peculiar sources, a Disentangled Convolutional Autoencoder and an Unsupervised Random Forest. The former method, working directly on images, is considered potentially able to identify peculiar objects like interacting galaxies and gravitational lenses. The latter instead, working on catalogue data, could identify objects with unusual values of magnitudes and colours, which in turn could indicate the presence of singularities.

---

M. D'Addona (✉) · S. Cavuoti  
Department of Physics, University of Naples Federico II, Strada Vicinale Cupa Cintia, 21,  
80126 Naples, Italy  
e-mail: [mauritiussdadd@gmail.com](mailto:mauritiussdadd@gmail.com)

S. Cavuoti  
e-mail: [stefano.cavuoti@gmail.com](mailto:stefano.cavuoti@gmail.com)

G. Riccio · S. Cavuoti · C. Tortora · M. Brescia  
INAF - Astronomical Observatory of Capodimonte, Salita Moiariello 16, 80131 Naples, Italy  
e-mail: [giuseppe.riccio@inaf.it](mailto:giuseppe.riccio@inaf.it)

C. Tortora  
e-mail: [crescenzo.tortora@inaf.it](mailto:crescenzo.tortora@inaf.it)

M. Brescia  
e-mail: [massimo.brescia@inaf.it](mailto:massimo.brescia@inaf.it)

C. Tortora  
INAF - Osservatorio Astrofisico di Arcetri, Largo Enrico Fermi 5, 50125 Firenze, Italy

## 1 Introduction

Due to the rapid growth in volume and complexity of astronomical datasets, Machine Learning (ML) paradigms are gaining a key role within the data exploration and analysis. They are demonstrated as valid mechanisms to find hidden correlations among data and to discover rare and unexpected structures that do not fit those relations [1–3]. The latter, considered as outliers of a data distribution, can be of various nature and may have different degrees of scientific relevance: they can be artifacts produced by anomalies in the data processing pipelines or in the observing conditions, as well as peculiar objects underlining special and rare astronomical events, whose detection may improve the scientific knowledge of relevant physical phenomena.

Machine learning paradigms are divided into two main classes, respectively, supervised and unsupervised methods. While in the supervised case, an a priori *Knowledge Base* is needed to train the algorithms, unsupervised methods can learn the complex relationships among data, without inferring any known information and with a minimum of human supervision. Therefore, it is evident that unsupervised methods are the most suitable to detect anomalies. In particular, we focus on two specific models: an unsupervised variant of random forests (*Unsupervised Random Forest* or *URF*) [4] and a hybrid type of autoencoder (*Disentangled Convolutional Autoencoder* or *DCA*), which exploits the disentangling property of a variational autoencoder [5] but preserving the structure of a standard convolutional autoencoder [6].

In recent years both methods have successfully been used in the astrophysical context. For example, Tuccillo et al. [7] validated the former method on both analytic profiles and real galaxy images. Baron et al. [1] used a *URF* on galaxy spectra from the *Sloan Digital Sky Survey* (*SDSS*), finding objects with extreme emission line ratios, abnormally strong absorption lines, extremely reddened galaxies and other peculiar objects. Reis et al. [8] applied this method to infrared spectra of stars, showing that the metric defined in this algorithm traced the physical properties of the stars. Finally, Reis et al. [9] also discovered 31 new redshifted broad absorption line quasars within *SDSS* spectral data. Concerning the *DCA* model, a very similar architecture was successfully applied to radio data to disentangle noise signal contamination, revealing emissions from air showers, thus enabling accurate measurements of cosmic particle kinematics and identity [10]. More in general, such models are faster, compared to other traditional profile fitting methods, can be easily adapted to more simple/complex models and could be used to detect peculiar substructures, such as strong gravitational lenses and galaxy mergers.

In this preliminary work, we first use *DCA* on synthetic images in order to evaluate its theoretical performance, then we apply both methods on real image cutouts and catalogue counterparts. In particular, in Sect. 3 we describe the use of a *DCA* to perform an outlier detection using images extracted from the 4th Data Release of the European Southern Observatory (ESO) Kilo Degree Survey (*KiDS*) [11]. Then, for the same purpose, in Sect. 4 we describe the use of an *URF* on the same subset of

objects but using photometric data, always extracted from the KiDS DR4. Finally in Sects. 5 and 6 we discuss the results and compare the performance of the two methods.

## 2 Data Preparation

In order to validate the *DCA* model and assess its performance we generated three sets of 20,000 synthetic images of  $64 \times 64$  pixels, using three different models of surface brightness profile of galaxies that are further described in Sect. 3.1. These images have a dynamic range between 0 and 1. A Gaussian noise, drawn from a folded normal distribution with a standard deviation of  $\sigma_{noise} = 0.09$ , has also been added to each image and the value of the standard deviation has been chosen to maintain the  $\sim 99\%$  of the values within the 30% of the dynamic range. The generated noise has a mean value  $\mu_{noise} \approx 5 \cdot 10^{-2}$  that corresponds to the 5% of the maximum value of the dynamic range of the image.

The real data selected to perform our tests on both methods are extracted from the KIDS Data Release 4 [11]. In particular, we randomly extracted a subset of object cutouts from the tiles that are in common with the DR3 data release [12] and using the DR4 photometry in the related catalogue.

For the photometry we used the Gaussian Aperture and PSF (GAaP) magnitudes in the four bands  $u, g, r, i$ , with the minimum aperture of 1.0 arcsec and the corresponding automatic minimal aperture magnitudes  $u_{auto}, g_{auto}, r_{auto}, i_{auto}$ , which are also corrected for the galactic extinction. In addition to these features, we also included all colours and magnitude ratios [13], derived from all the above magnitudes, resulting in a total of 36 photometric features. From this dataset we excluded all objects with missing data in any of the photometric bands. We also applied a minimum set of magnitude cuts, in order to remove the objects lying in the tails of the distributions:  $16 < i < 22$  and  $16 < r < 22$ . The result is a dataset of 400,000 objects.

For each object, a cutout of  $32 \times 32$  pixels (corresponding to  $\sim 6.7 \times 6.7$  arcsec) has been extracted from the corresponding photometrically and astrometrically calibrated  $r$  band coadded tiles. The size of the cutouts has been chosen so that almost all of them contain only the central object, while preserving a sufficient amount of surrounding pixels and angular size. All pixel values of the cutouts have been normalised between 0 and 1.

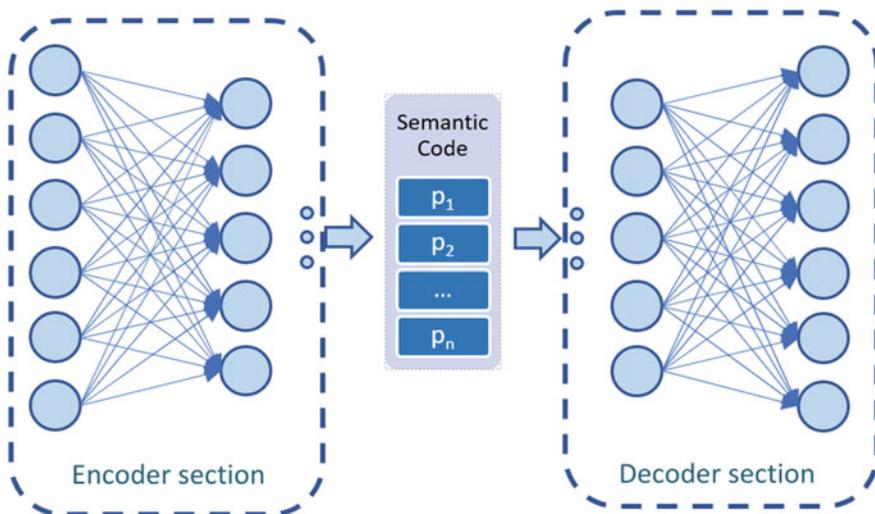
About the 90% of these objects was also present in the candidate quasars (QSOs) catalogue, produced for the 3rd Data Release of KiDS, containing a mixed set of stars, QSOs and galaxies, classified with Machine Learning [14]. The two catalogues were cross-matched, resulting in a subset of  $\sim 1100$  QSOs and  $\sim 260$  stars with a reliable classification, considered useful information to take into account in the evaluation of the anomaly detection experiment results.

### 3 Disentangled Convolutional Autoencoders

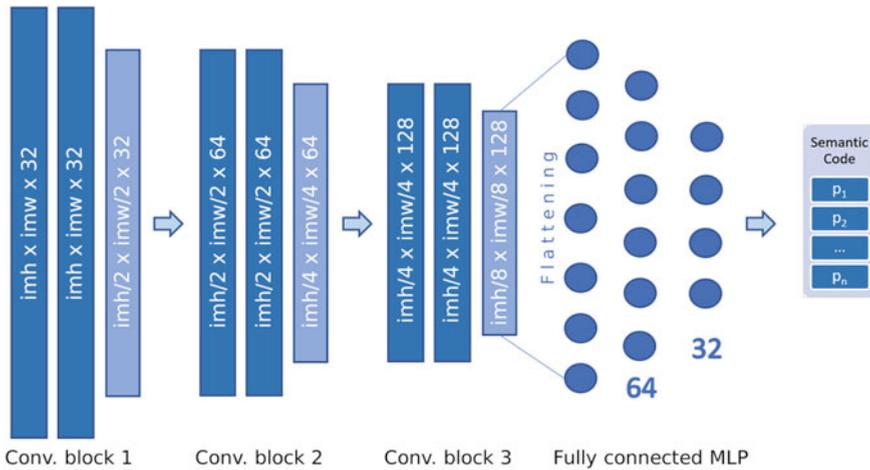
Autoencoders are a particular type of neural network used to learn data codings by efficiently mapping high-dimensional inputs into low-dimensional encoded vectors and reconstructing the input data from the encoded vector only [15]. By forcing the low-dimensional representation, or *latent space*, to have less dimensions than the input data, the network is forced to learn useful features from the data and, through the use of the backpropagation algorithm, in combination with a smooth loss function, the content of the latent space is iteratively adapted, in order to achieve a good reconstruction performance. For such reasons, the autoencoder is able to perform the feature extraction and dimensionality reduction tasks in a completely unsupervised fashion. The basic structure of an autoencoder consists of two sections (Fig. 1):

- An **encoder** that maps the input data into *semantic code vectors* that live in a so called *latent space*.
- A **decoder** that learns to decompress the *semantic code vectors* from the *latent space* back to the input space, producing a reconstructed representation of the input.

In traditional convolutional autoencoders, both the encoder and the decoder are *Convolutional Neural Networks (CNN)* [16]. The convolution operations allow to identify the key features in an image, thus making them well suited for classification, denoising and image compression tasks. However, since there is no direct control on how the input space is mapped to the *latent space*, it is difficult to extract specific and valuable information from the encoded *semantic code vectors*.



**Fig. 1** Representation of the basic structure of an autoencoder



**Fig. 2** The structure of the encoder used in our experiments: there are three convolutional blocks followed by a fully connected MLP with two hidden layers. Each block has two convolutional layers followed by a  $2 \times 2$  max-pooling operation.  $imw$  and  $imh$  are respectively the width and height of the input images

One way to overcome this limitation is to replace the decoder *CNN* with a given function that produces a synthetic model of the input data, as already proposed by Aragon-Calvo [17]. In this way, after a successful training, the *latent space* is forced to coincide with the domain of the model function and each parameter of the *semantic code* controls a different characteristic of the generated model, thus the name *Disentangled Convolutional Autoencoder*. An interesting feature of this type of autoencoders, implicitly deriving from its construction, is that they can successfully represent only objects compatible with the model assumed. Identifying those objects means to detect artifacts, images containing wrong data, but also interesting outliers.

In our experiments we developed a multi-GPU *DCA*, using the Python bindings of *TensorFlow* [18] and its built-in Keras module [19]. The encoder part is made by three convolutional blocks, each one containing two convolution layers, using a ReLU activation function and followed by a  $2 \times 2$  max-pooling layers (Fig. 2). The convolutional layers in the three blocks have respectively 32, 64 and 128 kernels of size  $4 \times 4$ . The last max-pooling layer has 128 channels of size  $\frac{imh}{8} \times \frac{imw}{8}$ , where  $imh$  and  $imw$  are, respectively, the height and width of the input images. This hierarchical module is then flattened and fed to a fully connected *Multi-Layer Perceptron* (MLP) [20], with two hidden layers of 64 and 32 neurons, respectively. The output layer of the MLP section has as many neurons as many parameters there are in the model used by the decoder.

The decoder is a custom TensorFlow layer object that encapsulates a given model and passes to it the encoder output  $p$  along with a pair of coordinate meshes  $X$  and

$Y$ . If needed, the model may also take care of applying some appropriate constraints to the parameters. The coordinate meshes have the same size of the input images and contain, respectively, the  $x$  and  $y$  pixel coordinates.

### 3.1 Validation with Synthetic Data

In order to evaluate the autoencoder performance, we have first created two simple models of galaxy surface brightness: an exponential and a Sérsic profile model. Then we added a third more complex Bulge/Disk model, which is a linear combination of these two. For each profile, we generate a set of synthetic images as described in Sect. 2 and used half of them as train set, while the rest as a blind test set.

#### 3.1.1 Exponential Profile of Galaxy Surface Brightness

The exponential profile usually well describes the light distribution of the disk of a galaxy as a function of the distance from its centre [21]. The model we implemented has five parameters:

- $x_0$ : the  $x$  coordinate of the center of the galaxy;
- $y_0$ : the  $y$  coordinate of the center of the galaxy;
- $a$ : the size of the semi-major axis in pixels;
- $q$ : the ratio between the minor and major axis;
- $\theta$ : the rotation angle, defined as the angle that the major axis forms with the  $x$  axis of the image.

Using these parameters, we first apply a coordinate transformation to take into account the translation and rotation of the galaxy (Eq. 1).

$$\begin{aligned} x'(x, y) &= (x - x_0) \cdot \cos(\theta) - (y - y_0) \cdot \sin(\theta) \\ y'(x, y) &= (x - x_0) \cdot \sin(\theta) + (y - y_0) \cdot \cos(\theta) \end{aligned} \quad (1)$$

Using the transformed coordinates we then compute the radius value for a give pixel coordinate  $(x, y)$  with the Eq. 2.

$$r'(x, y) = \frac{1}{a} \cdot \sqrt{x'(x, y)^2 + \left(\frac{y'(x, y)}{q}\right)^2} \quad (2)$$

And finally we compute the exponential brightness profile (Eq. 3).

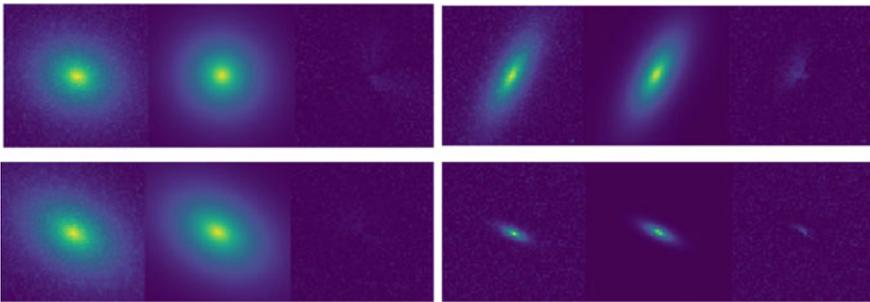
$$f_{exp}(x, y) = \exp(-r'(x, y)) \quad (3)$$

This profile is normalised so that the maximum value is 1 at  $(x = x_0, y = y_0)$  and the minimum value is zero. With this profile and using random parameters, we generated 20,000 synthetic images of  $64 \times 64$  pixels, according to the procedure described in Sect. 2. We then split the images into a train set and test set of 10,000 images each. We run the autoencoder on the train set using different optimizers and loss functions. We obtained the best results using the *Adam* optimizer [22] with a learning rate of  $lr = 1e - 4$ , a batch size of 128 images and a maximum number of 2000 training epochs. We also used a custom loss function defined as follows (Eq. 4):

$$\text{loss}_{mael} = \frac{1}{N \cdot W \cdot H} \sum_{j=0}^N \sum_{x=0, y=0}^{W, H} |\ln(1 + f_j(x, y)) - \ln(1 + I_j(x, y))| \quad (4)$$

where  $W$  and  $H$  are, respectively, the width and height of the input images  $I_j$ ;  $f_j$  is the output image generated by the autoencoder for the corresponding input image and  $N$  is the total number of the images. The logarithmic transformations in Eq. (4) give more weight to the fainter regions of the galaxies that are also the parts more difficult to fit. Using a higher learning rate, the training time decreases, but it also increases the chances that the algorithm will not converge to an optimal solution. Other optimizers like *Adadelta* [23] or *Stochastic gradient descent (SGD)* [24, 25] very often did not converge to an optimal solution, even using different learning rates. Using these training parameters we performed 25 executions and selected the trained model that provided the minimum *mean absolute error (MAE)* between the input and the output images and ran it on the test set (Fig. 3).

As described in Sect. 3, the output of the autoencoder is a reconstruction of the input images, based on the parameters of the model. Therefore, to assess the goodness of the reconstructed image and in turn of the parameters, we computed the *MAE* and the *normalised median absolute deviation (NMAD)* of the residuals for each pair of



**Fig. 3** Comparison between the autoencoder input and output images. The colour-map was chosen for better visualization, so that the brightest pixels are yellow and the darkest ones are in dark blue. The two top images are training examples, while the two on the bottom are examples from the test set. For each image, from left to right panels respectively, there are the original synthetic image with noise, the output of the autoencoder and the residual of the two images

**Table 1** Statistical estimators for the true versus predicted values for each parameter of the exponential galaxy profile model. Note that, although the uncertainty on the size of the galaxy is relatively larger than other parameters, the uncertainty on the axis ratio is small

Parameter	$x_0$	$y_0$	$a$	$q$	$\theta$
NMAE	0.16	0.16	0.32	0.02	0.01
NMAD	0.02	0.02	0.03	0.01	0.01

input-output images, finding an average  $\overline{MAE} = 0.07 \pm 0.02$ , which is compatible with the mean noise level and an average  $\overline{NMAD} = 0.03 \pm 0.01$ , from which we can compute the equivalent standard deviation  $\overline{\sigma_{NMAD}} \approx 1.5 \cdot \overline{NMAD} = 0.05 \pm 0.02$ , which is compatible with the standard deviation of the noise. In Table 1 the normalised *MAE* and *NMAD* for each parameter of the model are also reported, computed using the true parameter values and the ones predicted by the trained encoder. The small values of these statistical indicators show that the autoencoder was able to train the model successfully.

### 3.1.2 Sérsic Profile of Galaxy Surface Brightness

The Sérsic profile usually describes well the light distribution of the bulge of a galaxy and elliptical galaxies in general [26, 27]. This profile is a generalization of the exponential profile, obtained by introducing a parameter  $n$ , called *Sérsic index*, that controls how the light is distributed across the galaxy. The inverse of this parameter  $\beta = 1/n$  is used as exponent of the radius in the surface brightness profile equation. For  $n = 1$  the Sérsic profile coincides with the exponential one. The model we implemented has six parameters:

- $x_0$ : the  $x$  coordinate of the center of the galaxy;
- $y_0$ : the  $y$  coordinate of the center of the galaxy;
- $a$ : the size of the semi-major axis in pixels;
- $c$ : the ratio between the minor and major axis;
- $\theta$ : the rotation angle defined as the angle that the major axis forms with the  $x$  axis of the image;
- $\beta$ : the inverse of the Sérsic index.

The only difference with the exponential model is in the profile function of Eq. (3) that becomes as specified in Eq. (5).

$$f_{\text{Sérsic}}(x, y) = \exp(-r'(x, y)^\beta) \quad (5)$$

As in the previous profile case, by using random parameters we generate 20,000 synthetic images, divided into a training and a test set, each one composed by 10,000 images. Also in this case we obtained the best training results using the *Adam* optimizer with the custom loss function (4), a learning rate of  $lr = 1e - 4$ , a batch size

**Table 2** Statistical estimators for the true versus predicted values, reported for each parameter of the Sérsic galaxy profile model

Parameter	$x_0$	$y_0$	$a$	$q$	$\theta$	$\beta$
NMAE	0.17	0.17	0.35	0.02	0.02	0.33
NMAD	0.02	0.02	0.04	0.01	0.01	0.08

of 128 images, and a maximum number of training epochs of 2000. We computed the *MAE* and the *NMAD* of the residuals for each pair of input-output images, finding an average  $\overline{MAE} = 0.06 \pm 0.03$  that is compatible with the mean noise level and an average  $\overline{NMAD} = 0.03 \pm 0.01$ . In Table 2 the normalised *MAE* and *NMAD* for each parameter of the model are reported, computed using the true parameter values and the ones predicted by the trained encoder. The small values of these statistical estimators show that the autoencoder was able to train the model successfully.

### 3.1.3 Bulge/Disk Profile of Galaxy Surface Brightness

This model is a linear combination of the Exponential and Sérsic profiles, used to mimic a combination of bulge and disk components as well as a uniform background. We also introduced a constant background level to take into account the sky background present in almost all real images. It has eleven parameters:

- $x_0$ : the  $x$  coordinate of the center of the galaxy;
- $y_0$ : the  $y$  coordinate of the center of the galaxy;
- $a_{disk}$ : the size of the semi major axis of the disk component in pixels;
- $c_{disk}$ : the ratio between the minor and major axis of the disk component;
- $\theta_{disk}$ : the rotation angle of the disk component, defined as the angle that the major axis forms with the  $x$  axis of the image;
- $\alpha$ : the fractional ratio between the central brightness of the bulge and the central brightness of the disk;
- $a_{bulge}$ : the size of the semi major axis of the bulge component in pixels;
- $c_{bulge}$ : the ratio between the minor and major axis of the bulge component;
- $\theta_{bulge}$ : the rotation angle of the bulge component, defined as the angle that the major axis forms with the  $x$  axis of the image;
- $\beta$ : the inverse of the bulge Sérsic index;
- $k$ : the background level expressed as the fractional ratio between the brightness of the background and the maximum brightness of bulge+disk.

The profile function of this model is shown in the Eq. (6).

$$f_{bd}(x, y) = (1 - k) \cdot (\alpha \cdot f_{sersic}(x, y) + (1 - \alpha) \cdot f_{exp}(x, y)) + k \quad (6)$$

As done in the previous tests, by using random parameters, we generate 20,000 synthetic images, divided into a training and a test set, each one containing 10,000

images. Also in this case we obtained the best training results using the Adam optimizer with the custom loss function (4), a learning rate of  $lr = 1e - 4$ , a batch size of 128 images, and a maximum number of 2000 training epochs. The results of the test and the training were similar to those found in the previous test, finding an average  $\overline{MAE} = 0.07 \pm 0.04$  that is compatible with the mean noise level and an average  $\overline{NMAD} = 0.03 \pm 0.02$ .

### 3.2 Application to KiDS Data

After having validated the autoencoder model on synthetic data, we tried to apply the *Bulge/Disk* profile model on real data. As already introduced in Sect. 2, the images used in this experiment are cutouts taken from the  $r$  band tiles of the KIDS DR4. We divided them into a training set of 30, 000 images and a set of 370, 000 images used to detect potentially interesting outliers.

We trained the autoencoder by using the *Bulge/Disk* profile model, the optimizer and training parameters validated with synthetic data. 25 training runs were performed, selecting the trained model with the lowest *MAE*. Finally, we ran the best trained model on the image test set.

#### 3.2.1 Anomaly Detection with DCA

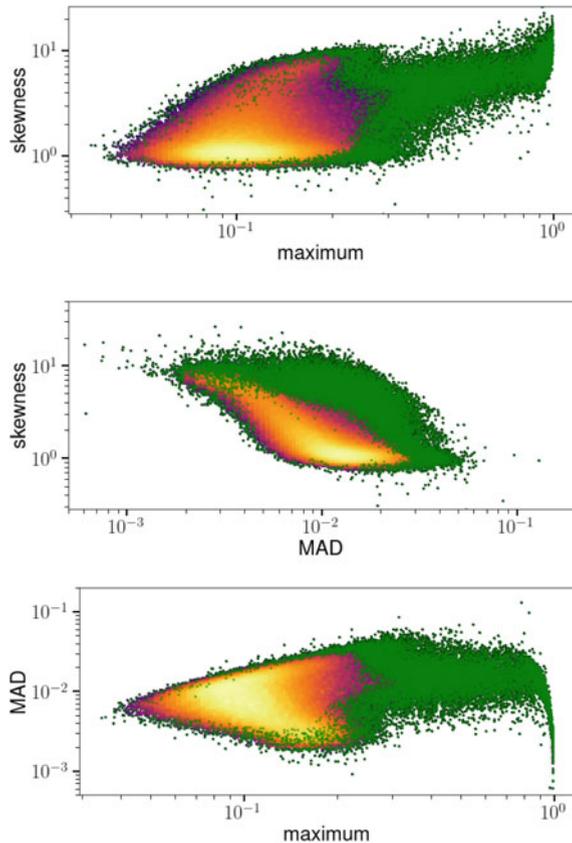
As we said above, if the autoencoder is correctly trained and the chosen model is a valid representation of the input objects, then the residual images—obtained by subtracting the output of the decoder from the corresponding input—should contain only residual noise. Therefore, it is clear that the statistical estimators computed on the residual images have a key role in detecting anomalies that the model is not able to describe. We used the following statistical estimators:

- **MAD:** since it is not very influenced by extreme values, the median of the pixel values in the residual image corresponds approximately to the mean background value. Thus, the *Median Absolute Deviation* is a valid measure of how broadly the residuals are distributed around the background. A high value could indicate the presence of substructures or artifacts.
- **Skewness:** unusually high or low values of this statistical moment could indicate that there is something odd in the image.
- **Maximum:** hot pixels, artifacts but also other objects in the whole image produce very bright pixels in the residual images.

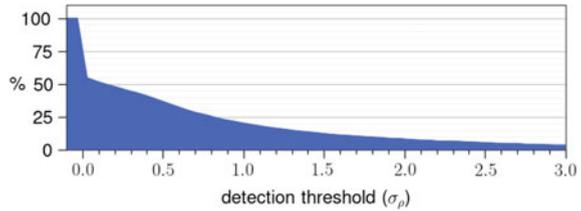
The outliers were selected using the following automated procedure: as first step, the average maximum value of the residuals  $\overline{max}$  was computed along with the respective standard deviation  $\sigma_{max}$  and all objects for which  $max > \overline{max} + 3\sigma_{max}$  were marked as outliers. Then for each unique pair of statistical estimators, the

average number density of the objects,  $\overline{\rho_n}$ , and the corresponding standard deviation  $\sigma_n$  were computed; then this two-dimensional space was divided into 400 tiles of equal size. The local number density  $\rho_n$  was computed in each tile and the resulting density map was smoothed with a gaussian kernel. Finally, each object falling in a sub-region with a density  $\rho < \overline{\rho_n} - 2\sigma_\rho$  was marked as an outlier, as it can be seen in Fig. 4. In Fig. 5 the percentage of objects detected as outliers is reported as a function of the detection threshold previously defined. We note the robustness of the detection that remains approximately constant above the value of  $2\sigma_\rho$ . About the 93% of the objects are concentrated in a quite continuous region with an average of  $\overline{MAD} = 0.011 \pm 0.005$ ,  $\overline{skewness} = 2.3 \pm 1.7$  and  $\overline{max} = 0.12 \pm 0.03$ . These are objects that the autoencoder was able to fit with the model. The low values of the  $\overline{MAD}$  and  $\overline{max}$  indicate that the reconstructions of the autoencoder describe very well the original images and that the residuals contain basically only background noise. This is also confirmed by the value of  $\overline{skewness}$  greater than zero, which is typical of Poissonian distributions, characterized by a low value of the mean, as in the case of the shot noise that affects digital images. We identified few objects

**Fig. 4** From top to bottom: scatter plots of the skewness versus maximum, skewness versus MAD and MAD versus maximum. Axes are in logarithmic scale and the colour indicates the logarithm of the local number density of the points, where a lighter colour means a denser region. Objects identified as outliers are highlighted in green



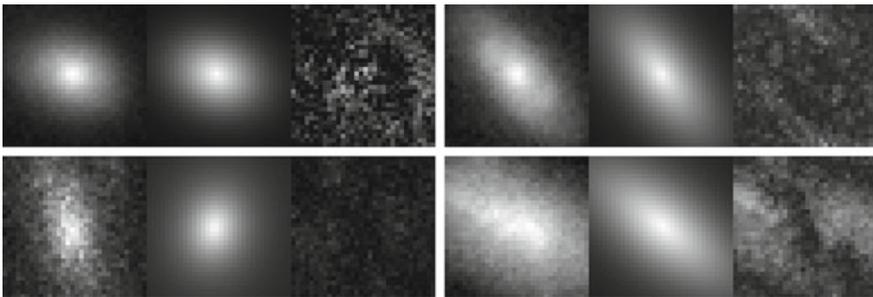
**Fig. 5** Percentages of objects classified as outliers by the DCA, as a function of the detection threshold expressed in units of  $\sigma_\rho$



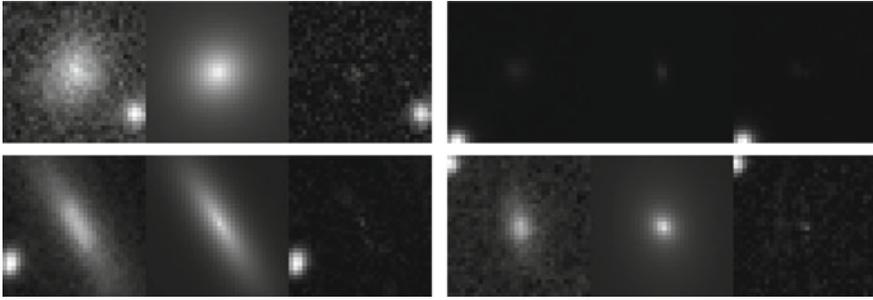
having a very low skewness, which usually indicates a Gaussian-like distribution of the residuals and thus the presence of something else beyond the pure Poissonian background noise.

Some of these objects were bigger than the cutout area (Fig. 6), which the autoencoder was less able to fit properly, while others showed traces of substructures in the residual image, which were hidden by the galaxy light. There was also a small clump of objects, less than 2% of the total amount, having a very high maximum value: these were very faint sources or objects with a very bright companion (Fig. 7).

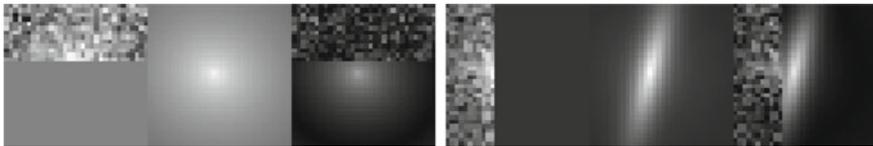
A small set of objects has also a very low *MAD*. Although a low value of this statistical estimator could imply a low dispersion of the residuals, an unusual low value means that most of the pixels in the residual image have the same value, which in turn could indicate some sort of corruption. In fact, most of these objects were located on the edge of the tile, thus resulting in partially corrupted cutouts (Fig. 8). Finally, there was a subset of objects, approximately the 5% of the total amount, having a fairly average of *MAD* and *skewness* values, but with a quite large maximum value ranging from 0.3 to 0.8. Almost all of these objects have one or more than one faint companion, as shown in the examples of Fig. 9.



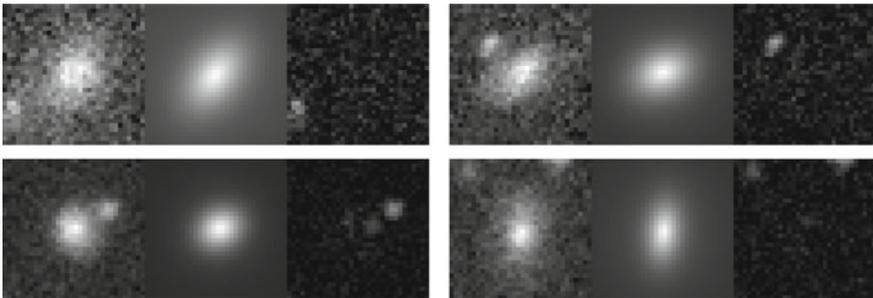
**Fig. 6** Some of these objects show the presence of substructures that were hidden by the light of the galaxy (upper left and upper right), while in other cases the autoencoder failed to fit the surface brightness profile because the objects were bigger than the size of the cutout. For each image, from left to right panels, there is the original image, the images produced by the autoencoder and the residual image, re-scaled to highlight the presence of substructures



**Fig. 7** Examples of KiDS galaxy that are very faint or have a very bright close companion or present artifacts like hot-pixels. For each image, from left to right panels, there is the original image, the images produced by the autoencoder and the residual image



**Fig. 8** Examples of objects that are just on the border of the tile from where the cutouts have been extracted. For each image, from left to right panels, there is the original image, the images produced by the autoencoder and the residual image



**Fig. 9** Examples of objects showing the presence of a faint quite close companion. For each image, from left to right panels, there is the original image, the images produced by the autoencoder and the residual image

## 4 Unsupervised Random Forests

Random Forests are an ensemble of several independently grown decision tree classifiers, where each tree is a non-parametric model organized in a top-bottom tree-like structure and is grown using a random subset of the features of the training dataset [28]. They are usually used to classify objects for which a training set of labelled objects exists so that each tree in the forest learns to map the input features

to the corresponding correct label. When an object identified by a set of feature is passed to the forest, each tree votes for its belonging to one of the given classes, identified by the labels, and the resulting class is usually determined by majority voting.

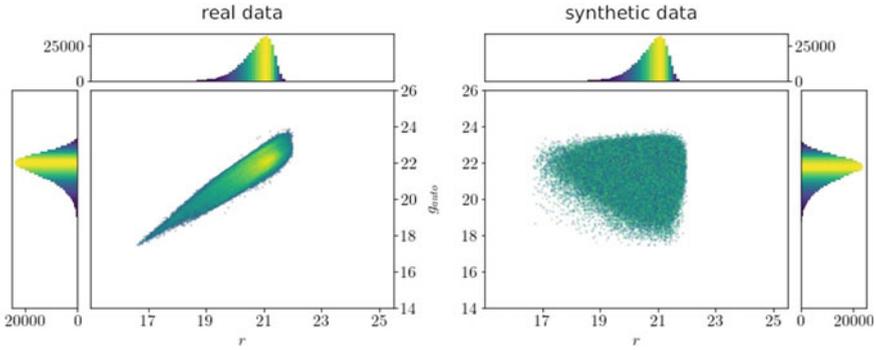
For the problem of outliers detection, where obviously a labelled training set is not available, random forests can also be used in an unsupervised configuration.

A simple but efficient way to use Random Forest as an unsupervised method is to generate a synthetic dataset from the original one, with the same size and the same marginal distribution in all its features, but without the covariance among objects. Then the Random Forest is trained on both datasets to learn to recognize their similarity, thus isolating the outliers. By defining a *similarity index*  $S_{i,j}$  between any two objects as the number of common “real” leaves of the trees, divided by the total number of trees in the forest, a *weirdness score* can be introduced, which describes how distant is, on average, from all the others. This score can assume any value between 0 and 1, but the distribution of its values mostly depends on the specific dataset involved. Therefore, a reasonable way to use it is to impose a certain threshold, based on the distribution of its values for all the objects in the dataset and then to consider as outliers all objects with a weirdness value greater of such threshold. Baron and Poznanski [1] proposed this method that was able to find some galaxies with peculiar spectra in the 12th data release of the Sloan Digital Sky Survey [29].

#### 4.1 Anomaly Detection in KiDS Data Based on the URF

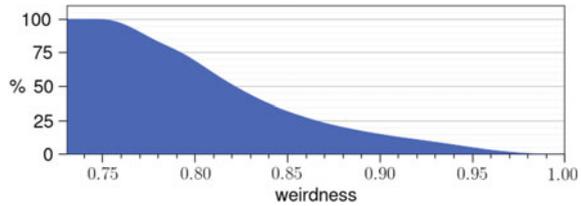
For this experiment we used the photometric catalogue containing the counterparts of the image cutouts, organized as described in Sect. 2. According to what described in Sect. 4, we then created a synthetic dataset of the same size of the real one and with objects drawn randomly from the same marginal distribution of each feature (Fig. 10).

We then merged the two datasets into a single one, labelling the objects depending on whether they were real or synthetic, and used for training and testing a *Random Forest Classifier* containing 800 trees, built using the Python package *scikit-learn* [30]. We remark that in the case of the *URF* model, both training and testing sets coincide, since the same data are used to perform the anomaly detection experiment along the construction of the random forest trees process. We then divided the original dataset into batches of 6000 objects and computed the *weirdness* index for each batch. The size was limited by the amount of memory necessary to compute the *weirdness*. The whole process has been repeated four times and the *weirdness* values have been averaged for each object. The objects show a distribution centred on an average value of *weirdness* of  $\bar{w} = 0.83$  with a standard deviation of  $\sigma_w = 0.06$ , while the number of objects decreases as the *weirdness* value increases (Fig. 11). In analogy to what done in the case of the *DCA* model (Sect. 3.2.1), and to perform a direct comparison between the two models, we imposed a detection threshold of  $2\sigma$  and considered as outliers all objects for which  $w > \bar{w} + 2\sigma_w = 0.95$ .



**Fig. 10** Projection on two features (magnitudes  $r$  and  $g$ ) of the density distributions of the KiDS sample (left panel) and synthetic data (right panel). The latter was generated from the same marginal distributions of the real one, by removing the covariance among original data

**Fig. 11** Percentages of objects classified as outliers by the  $URF$ , as a function of the weirdness



## 5 Discussion

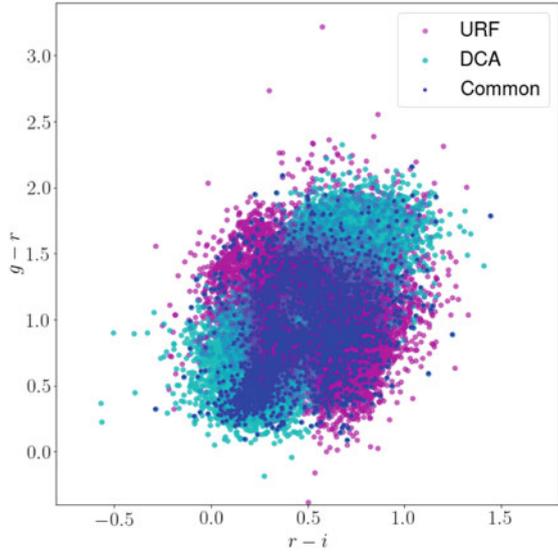
Both chosen algorithms,  $DCA$  and  $URF$ , were tested on a subset of 400, 000 objects extracted from the KiDS survey Data Release 4. The  $DCA$  was used directly on image cutouts extracted from the  $r$  band coadded tiles, while the  $URF$  was used on the catalogue of counterparts, made by magnitudes in the bands  $ugri$ , their derived colours and ratios. The model  $DCA$  required only a minimum of human supervision during the training, just to check the convergence of the algorithm to an optimal solution. It performed very well in both terms of memory requirements and computing time and was able to pinpoint some peculiar sources, about the 5% of the sample, showing substructures that were hidden by the close galaxy light, as well as objects with very small and/or faint close companions.

Since the  $URF$  is based on the computation of a *similarity matrix*, whose size increases as the square of the number of the objects, the dataset has to be analysed in batches and a supplementary amount of human intervention was required in order to determine the optimal batch size.

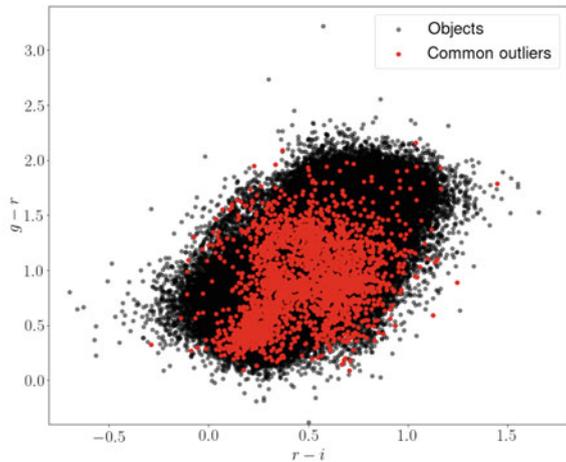
To perform a comparative analysis of the results obtained by the two methods, we imposed a similar criterion to extract candidate outliers, for instance, a common value of  $2\sigma$  with  $w > \bar{w} + 2\sigma_w = 0.95$  in terms of weirdness  $w$  for the  $URF$  and object density  $\rho < \bar{\rho}_n - 2\sigma_\rho$  in the case of  $DCA$ . With such outlier detection thresholds,

both methods found a comparable amount of peculiar objects,  $\sim 7\%$  of the test set for *DCA* and  $\sim 5$  for *URF*. Among the objects considered as an anomaly by at least one of the two methods, the  $\sim 7\%$  were detected as peculiar objects by both of them. The distributions of the outliers (Fig. 12) shows that most of the peculiar objects found by the two models cover a wide and uncorrelated area of the parameter space, with a limited overlapping region in which most of the common outliers lay. This seems to suggest a certain amount of complementarity of the two methods in

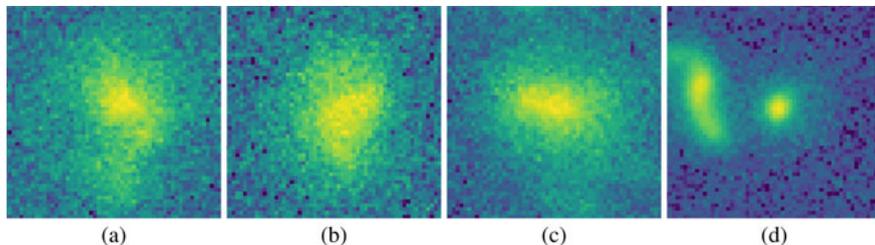
**Fig. 12** Upper panel: colour-colour diagram of all the candidate outliers detected by *URF* (in magenta) and *DCA* (in cyan). Common outliers found by both methods are coloured in blue. Lower panel: colour-colour diagram of the outliers detected by both *DCA* and *URF* (in red), plotted against all the objects in the dataset (in black)



(a)



(b)



**Fig. 13** Examples of cutouts including irregular galaxies (a, b and c) and interacting galaxies (d), detected as anomalies by the two methods

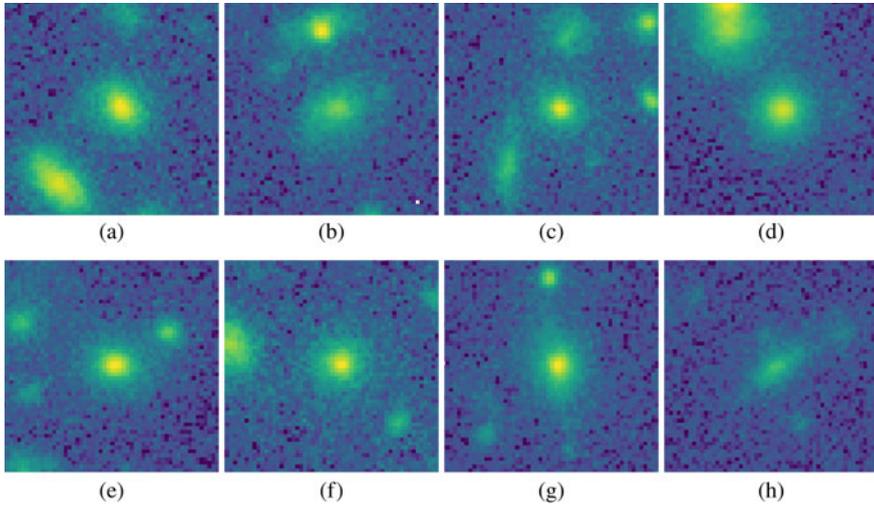
detecting peculiarities, according to a similar behaviour found in [31], concerning the analysis of outliers identified from a distribution of photometric redshifts, estimated by different methods, however no any particular evidence of interesting peculiarity seems to emerge.

By analyzing the detected peculiar objects having a class label provided in [14], only about the  $\sim 27\%$  of stars and QSOs were detected as anomalies. Most of these objects, in fact, were not confirmed as peculiar by *DCA* and appear uniformly distributed with respect to the different thresholds of weirdness calculated by the *URF*. This behaviour was expected for *DCA* because no any limitation was imposed on the value of the Sérsic index nor on the galaxy size, thus the model should be able to fit also star-like objects. Through visual inspection of the cutouts for the peculiar objects detected, we observe that both methods tend to assign as peculiar the irregular and interacting galaxies (see examples in Fig. 13), as well as objects that are in more crowded fields, like the ones showed in Fig. 14.

## 6 Conclusions

The identification of anomalies in Astronomy has always played a major role in making new scientific discoveries. Nowadays, the shift to more large and complex surveys makes essential the use of robust and efficient automated algorithms to identify peculiar patterns. In this context, we performed a preliminary set of anomaly detection experiments, by testing two different unsupervised machine learning algorithms, a *Disentangled Convolutional Autoencoder* and an *Unsupervised Random Forest*, using the former on real image cutouts and the latter on the catalogue of their counterparts, which includes measured magnitudes, derived colours and magnitude ratios, both extracted from the 4th KiDS Data Release.

We performed a comparative analysis of the peculiar objects detected by both methods, by analyzing their colour distribution in the parameter space and their capability to disentangle the presence of QSOs and stars from galaxies within a mixed dataset. The results of this preliminary experiment revealed that most of the



**Fig. 14** Examples of cutouts for sources within crowded fields, detected as anomalies by the two methods

anomalies detected by both methods involve irregular and interacting galaxies and sources located in more crowded fields. Further experiments are then required on these models, especially in terms of their setup and configuration, to investigate their real capability to isolate peculiar types of sources. In particular, since the *DCA* is mainly a method to estimate the goodness of a fit to the data, it may result affected by the presence of nearby objects, not taken into account by the model. Improving the detection criteria for *DCA* is thus one of the future enhancements of this method, as well as to take into account the PSF and seeing in the Bulge/Disk model, which should achieve a more accurate estimation of the structural parameters. Regarding *URF*, on the other hand, a further step is the introduction of the infrared bands in the photometric dataset, as well as the search for spectroscopic counterparts, which can improve the classification accuracy and the validation of the method.

**Acknowledgements** Based on observations made with ESO Telescopes at the La Silla Paranal Observatory under programme IDs 177.A-3016, 177.A-3017, 177.A-3018 and 179.A-2004, and on data products produced by the KiDS consortium. The KiDS production team acknowledges support from: Deutsche Forschungsgemeinschaft, ERC, NOVA and NWO-M grants; Target; the University of Padova, and the University Federico II (Naples). MB acknowledges financial contributions from the agreement ASI/INAF 2018-23-HH.0, *Euclid* ESA mission - Phase D. MB and CT acknowledge the INAF PRIN-SKA 2017 program 1.05.01.88.04. SC acknowledges the financial contribution from FFABR 2017.

## References

1. Baron, D., Poznanski, D.: *Mon. Not. R. Astron. Soc.* **465**(4), 4530 (2017). <https://doi.org/10.1093/mnras/stw3021>, <https://academic.oup.com/mnras/article-lookup/doi/10.1093/mnras/stw3021>
2. Brescia, M., Cavuoti, S., Amaro, V., Riccio, G., Angora, G., Vellucci, C., Longo, G.: Data Analytics and Management in Data Intensive Domains. In: Kalinichenko, L., Manolopoulos, Y., Malkov, O., Skvortsov, N., Stupnikov, S., Sukhomlin, V. (eds.) *Communications in Computer and Information Science*, vol. 822, pp. 61–72. Springer International Publishing, Berlin (2018)
3. Fluke, C.J., Jacobs, C.: arXiv e-prints [arXiv:1912.02934](https://arxiv.org/abs/1912.02934) (2019)
4. Shi, T., Horvath, S.: *J. Comput. Graph. Stat.* (2006). <https://doi.org/10.1198/106186006X94072>
5. Chen, R.T.Q., Li, X., Grosse, R.B., Duvenaud, D.K.: Advances in Neural Information Processing Systems 31. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.), pp. 2610–2620. Curran Associates Inc., New York (2018). <http://papers.nips.cc/paper/7527-isolating-sources-of-disentanglement-in-variational-autoencoders.pdf>
6. Guo, X., Liu, X., Zhu, E., Yin, J.: *Neural Information Processing*. In: Liu, D., Xie, S., Li, Y., Zhao, D., El-Alfy, E.S.M. (eds.), pp. 373–382. Springer International Publishing, Cham (2017)
7. Tuccillo, D., Huertas-Company, M., Decencière, E., Velasco-Forero, S., Domínguez Sánchez, H., Dimauro, P.: *Mon. Not. R. Astron. Soc.* **475**(1), 894 (2018). <https://doi.org/10.1093/mnras/stx3186>
8. Reis, I., Baron, D., Shahaf, S.: *Astron. J.* (2018). <https://doi.org/10.3847/1538-3881/aaf101>
9. Reis, I., Poznanski, D., Hall, P.B.: *MNRAS* **480**(3), 3889 (2018). <https://doi.org/10.1093/mnras/sty2127>
10. Erdmann, M., Schlüter, F., Šmída, R.: *J. Instrum.* **14**(4), P04005 (2019). <https://doi.org/10.1088/1748-0221/14/04/P04005>
11. Kuijken, K., Heymans, C., Dvornik, A., Hildebrandt, H., De Jong, J.T., et al.: *Astron. Astrophys.* (2019). <https://doi.org/10.1051/0004-6361/201834918>
12. de Jong, J.T.A., Verdoes Kleijn, G.A., Erben, T., Hildebrandt, H., Kuijken, K., et al.: *A&A* **604**, A134 (2017). <https://doi.org/10.1051/0004-6361/201730747>
13. D’Isanto, A., Cavuoti, S., Gieseke, F., Polsterer, K. L.: *A&A* **616** (2018). <https://doi.org/10.1051/0004-6361/201833103>, [arXiv:1904.07248](https://arxiv.org/abs/1904.07248)
14. Nakoneczny, S., Bilicki, M., Solarz, A., Pollo, A., Maddox, N., Spiniello, C., Brescia, M., Napolitano, N.R.: *Astron. Astrophys.* (2019). <https://doi.org/10.1051/0004-6361/201834794>
15. Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. MIT Press, Cambridge (2016). <http://www.deeplearningbook.org>
16. Fukushima, K.: *Biol. Cybern.* (1980). <https://doi.org/10.1007/BF00344251>
17. Aragon-Calvo, M.A.: arXiv e-prints [arXiv:1907.03957](https://arxiv.org/abs/1907.03957) (2019)
18. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., et al.: *TensorFlow: large-scale machine learning on heterogeneous systems* (2015). Software available from <https://www.tensorflow.org/>
19. Chollet, F., et al.: *Keras*. <https://keras.io> (2015)
20. Van Der Malsburg, C.: *Brain Theory*, pp. 245–248. Springer, Berlin (1986). [https://doi.org/10.1007/978-3-642-70911-1\\_20](https://doi.org/10.1007/978-3-642-70911-1_20), [http://link.springer.com/10.1007/978-3-642-70911-1\\_20](http://link.springer.com/10.1007/978-3-642-70911-1_20)
21. Binney, J., Tremaine, S.: *Galactic Dynamics*, 2nd edn. Princeton Series in Astrophysics. Princeton University Press, Princeton (2008). <http://gen.lib.rus.ec/book/index.php?md5=cd0fd2e719d8966f78eee1f04eee540e>
22. Kingma, D.P., Ba, J.L.: In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings (2015)
23. Zeiler, M.D.: arXiv e-prints [arXiv:1212.5701](https://arxiv.org/abs/1212.5701) (2012)
24. Kiefer, J., Wolfowitz, J.: *Ann. Math. Stat.* **23**(3), 462 (1952). <https://doi.org/10.1214/AOMS/1177729392>
25. Robbins, H., Monro, S.: *Ann. Math. Stat.* **22**(3), 400 (1951). <https://doi.org/10.1214/AOMS/1177729586>

26. Graham, A.W., Driver, S.P.: *Publ. Astron. Soc. Aust.* (2005). <https://doi.org/10.1071/AS05001>
27. Roy, N., Napolitano, N.R., La Barbera, F., Tortora, C., Getman, F., et al.: *Mon. Not. R. Astron. Soc.* (2018). <https://doi.org/10.1093/mnras/sty1917>
28. Breiman, L.: *Mach. Learn.* **45**(1), 5 (2001). <https://doi.org/10.1023/A:1010933404324>
29. Alam, S., Albareti, F.D., Allende Prieto, C., Anders, F., Anderson, S.F., et al.: *The Eleventh and Twelfth Data Releases of the Sloan Digital Sky Survey: Final Data from SDSS-III* (2015). <https://doi.org/10.1088/0067-0049/219/1/12>
30. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., et al.: *J. Mach. Learn. Res.* **12**, 2825 (2011)
31. Brescia, M., Salvato, M., Cavuoti, S., Ananna, T.T., Riccio, G., LaMassa, S.M., Urry, C.M., Longo, G.: *MNRAS* **489**(1), 663 (2019). <https://doi.org/10.1093/mnras/stz2159>