Research in Astronomy and Astrophysics, 22:085006 (11pp), 2022 August https://doi.org/10.1088/1674-4527/ac7386

Identifying Outliers in Astronomical Images with Unsupervised Machine Learning

Yang Han[1], Zhiqiang Zou[1,2], Nan Li[3,4], and Yanli Chen[1,2]
[1] School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China
[2] Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing 210023, China
[3] Key Laboratory of Optical Astronomy, National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100101, China; nan.li@nao.cas.cn
[4] University of Chinese Academy of Sciences, Beijing 100049, China
Received 2022 March 4; revised 2022 May 10; accepted 2022 May 18; published 2022 July 15

Abstract

Astronomical outliers, such as unusual, rare or unknown types of astronomical objects or phenomena, constantly lead to the discovery of genuinely unforeseen knowledge in astronomy. More unpredictable outliers will be uncovered in principle with the increment of the coverage and quality of upcoming survey data. However, it is a severe challenge to mine rare and unexpected targets from enormous data with human inspection due to a significant workload. Supervised learning is also unsuitable for this purpose because designing proper training sets for unanticipated signals is unworkable. Motivated by these challenges, we adopt unsupervised machine learning approaches to identify outliers in the data of galaxy images to explore the paths for detecting astronomical outliers. For comparison, we construct three methods, which are built upon the k-nearest neighbors (KNN), Convolutional Auto-Encoder (CAE) + KNN, and CAE + KNN + Attention Mechanism (attCAE_KNN) separately. Testing sets are created based on the Galaxy Zoo image data published online to evaluate the performance of the above methods. Results show that attCAE_KNN achieves the best recall (78%), which is 53% higher than the classical KNN method and 22% higher than CAE+KNN. The efficiency of attCAE_KNN (10 minutes) is also superior to KNN (4 h) and equal to CAE+KNN (10 minutes) for accomplishing the same task. Thus, we believe that it is feasible to detect astronomical outliers in the data of galaxy images in an unsupervised manner. Next, we will apply attCAE_KNN to available survey data sets to assess its applicability and reliability.

Key words: Galaxy – Physical Data and Processes – Galaxy: fundamental parameters

1. Introduction

Astronomy is stepping into the big data era with the upcoming large-scale surveys (Lochner & Bassett 2021), e.g., Euclid[5], LSST[6] and CSST.[7] Mining knowledge from enormous astronomical data sets has become critical for astrophysical and cosmological investigations. Typically, data mining in astronomy includes object classification, dependency detection, class description, and anomalies/outlier detection. The first three categories of tasks are problem-driven, i.e., once the goals are well-defined, the tasks can be handled in a supervised manner by involving well-designed training sets. These tasks help improve the accuracy and precision of the models for describing mainstream objects, and relevant approaches are relatively mature and widely applied in astronomy (Lukic et al. 2019; Zhu et al. 2019; Cheng et al. 2020; Gupta et al. 2022; Chen et al. 2022; Zhang et al. 2022). On the other hand, astronomical anomalies/outliers constantly lead to unforeseen knowledge in astronomy, which may trigger revolutionary discoveries. Expectedly, more unpredictable outliers should be uncovered in principle with the increment of the coverage and quality of upcoming survey data. Therefore, developing approaches for outlier detection are as important as those for the first three tasks (Reyes & Estévez 2020; Webb et al. 2020; Ishida et al. 2021).

Outliers are defined in various papers (Hawkins 1980; Beckman & Cook 1983; Barnett & Lewis 1996; Pearsons et al. 1995), generally, it is described as: an outlier is an observation that deviates significantly from primary observations so that it aroused suspicions that a different mechanism generates it. In daily life, outlier detection has numerous applications, including credit card fraud detection, the discovery of criminal activities in E-commerce, video surveillance, pharmaceutical research, weather prediction and the analysis of performance statistics of professional athletes. Most of them are relevant to troubles. Nevertheless, the detection of astronomical outliers always leads to the discovery of surprising unforeseen facts and expands the boundaries of human knowledge of the universe (Pruzhinskaya et al. 2019; Sharma et al. 2019). Hence, it is necessary to develop efficient and automated approaches for

[5] https://www.euclid-ec.org/
[6] https://www.lsst.org
[7] http://www.bao.ac.cn/csst/

detecting astronomical outliers and understand their feasibility and reliability thoroughly, particularly in the era of big data (Zhang et al. 2004; Margalef-Bentabol et al. 2020).

Early outlier detection methods (Edgeworth 1888; Zhang et al. 2004; Dutta et al. 2007; Solarz et al. 2017; Giles & Walkowicz 2019; Fustes et al. 2013; Baron & Poznanski 2017) are generally based on traditional unsupervised learning algorithms. For instance, Giles & Walkowicz (2019) employed a variant of the DBSCAN clustering algorithm to detect outliers in derived light curve features. Baron & Poznanski (2017) extracted the feature of galaxy spectra manually first and then adopted an unsupervised Random Forest to detect the most outlying galaxy spectra within the Sloan Digital Sky Survey[8] (SDSS). Moreover, as a well-known clustering algorithm, k-Nearest Neighbor (KNN, Dasarathy 1991) becomes popular for detecting outliers because it operates without assumption about the data distribution. However, these traditional methods become unsuitable when the volume and quality of astronomical images increase greatly. One reason is that the feature extraction routines in traditional methods are too coarse and inflexible to retain details and untypical features of the high-quality astronomical images; another reason is that the efficiency of CPU-based traditional methods is too slow to handle the tremendous volume of future survey data.

Recently, beyond traditional machine learning, deep learning has been utilized to construct programs for detecting outliers (Chalapathy & Chawla 2019; Nadeem et al. 2016; Hendrycks et al. 2018; D'Addona et al. 2021), such as Auto-Encoder (AE, Vincent et al. 2010) and Convolutional Auto-Encoder (CAE, Masci et al. 2011; Storey-Fisher et al. 2020). AE and CAE represent input images with a feature vector which can be used to reconstruct the input images with the most likelihoods. This feature extraction procedure is automated and speedy. Besides, Bayesian Gaussian Mixture is utilized to implement the clustering process and then identify the galaxy images' outlier according to the distribution of the feature vectors in latent space (Cheng et al. 2021). Combining the above two modules, one can classify galaxy images without labels (Cheng et al. 2020), as well as to detect outliers. However, the performance of such unsupervised approaches based on deep learning is above 10%–20% worse than that of supervised approaches due to noisy data (Zhou & Paffenroth 2017).

In this work, we adopt the attention mechanism (Vaswani et al. 2017) to further improve the performance of the unsupervised methods as it can make the CAE pay more attention to the critical features and suppress background noise. To understand the differences from traditional outlier detection methods to state-of-art attention-improved ones systematically, we construct three programs, which are built upon the KNN, CAE + KNN and CAE + KNN + Attention mechanism, separately. We organize two types of data sets based on the galaxy images data published by the Galaxy Zoo Challenge Project on Kaggle[9] to evaluate the performance of various approaches in different cases. The first data sets of galaxy images for testing the above approaches include inliers containing a single type of galaxy morphology plus outliers containing a single type of galaxy morphology; the second data set is similar to the first ones but with multiple types of galaxy morphology in the outliers. After conducting extensive experiments, we find that CAE boosts the clustering process significantly and improves the accuracy of detecting outliers; the attention mechanism increases the accuracy further because it guarantees CAE to extract valuable features only, avoiding noise. It is the first time involving the attention mechanism in the outlier detection of astronomical images, which is worth being included in the program for similar purposes in the future. For the convenience of other researchers, we published the code and data used in this project onine.[10]

This paper is structured as follows. We introduce the data sets used in this work in Section 2. Section 3 describes the methods that we constructed. Details about the experiments, including data processing and the implementation of outliers detection with the above approaches, are shown in Section 4. We summarize and analyze the results in Section 5. Finally, the discussion and conclusions are delivered in Section 6.

2. Data

The galaxy morphology data used in this study is collected from the Galaxy Zoo project (Willett et al. 2013; Ventura & D'Antona 2011). In this section, we first introduce the origin and composition of the data set and then present the filtering methods and how to divide the original data in Section 2.1. Section 2.2 describes how to construct experimental data subsets to evaluate the performance of outlier detection with the approaches mentioned in Section 3.

2.1. The Galaxy Zoo Dataset

The SDSS captured around one million galaxy images. To classify these galaxies morphologically, the Galaxy Zoo Project was launched (Lintott et al. 2008), a crowd-sourced astronomy project inviting volunteers to assist in the morphological classification of large numbers of galaxies. The data set we adopted is one of the legacies of the galaxy zoo project, and it is publicly available online with the Galaxy-zoo Data Challenge Project on Kaggle.[11]

The data set provides 28,793 galaxy morphology images with middle filters available in SDSS (g, r, and i) and a truth table including 37 parameters for describing the morphology of each galaxy. The 37 parameters are between 0 and 1 to

[8] https://www.sdss.org/

[9] https://www.kaggle.com/c/galaxy-zoo-the-galaxy-challenge

[10] https://github.com/hanlaomao/hanlaomao.git

[11] https://www.kaggle.com/c/galaxy-zoo-the-galaxy-challenge/data

Table 1
Five Galaxy Morphology Categories from 0 to 4 with 28,793 Samples

Category	Category name	Thresholds	Number
0	Completely round smooth	$f_{\text{smooth}} \geqslant 0.469$	8436
		$f_{\text{completelyround}} \geqslant 0.50$	
1	In-between smooth	$f_{\text{smooth}} \geqslant 0.469$	8069
		$f_{\text{in-between}} \geqslant 0.50$	
2	Cigar-shaped smooth	$f_{\text{smooth}} \geqslant 0.469$	579
		$f_{\text{cigar-shaped}} \geqslant 0.50$	
3	Edge-on	$f_{\text{features/disk}} \geqslant 0.430$	3903
		$f_{\text{edge-on,yes}} \geqslant 0.602$	
4	Spiral	$f_{\text{edge-on,no}} \geqslant 0.715$	7806
		$f_{\text{Spiral,yes}} \geqslant 0.619$	
Total			28,793

Note. The first column is the category id, the second column is the name of category, the third column is the thresholds corresponding to each category, and the last column is the number of galaxy images in each category.

Table 2
Number of Samples in Five Different Experimental Data Subsets

Category	Subset1	Subset2	Subset3	Subset4	Subset5
0	16000	16000	16000	16000	16000
1	0	0	1778	0	445
2	1778	0	0	0	445
3	0	1778	0	0	444
4	0	0	0	1778	444
Total	17778	17778	17778	17778	17778

Note. The first column is the category id and the last five columns are five data subsets. For example, the second column represents the first experimental data subset, it only contains 16,000 samples of category 0 as inliers and 1778 samples of category 2 as outliers. The last column represents the fifth data subset which contains 16,000 samples of category 0 as inliers and contains total 1778 samples from category 1–4.

represent the probability distribution of galaxy morphology in 11 tasks and 37 responses (Willett et al. 2013). Higher response values indicate that more people recognize the corresponding features in the images of given galaxies. The catalog is further debiased to match a more consistent question tree of galaxy morphology classification (Hart et al. 2016).

To make the problem of outlier detection representative, we reorganize 28,793 images into five categories: Completely round smooth, In-between smooth, Cigar-shaped smooth, Edge-on and Spiral according to the 37 parameters in the truth table. The filtering method refers to the threshold discrimination criteria in Zhu et al. (2019). For instance, when selecting the Completely round smooth, values are chosen as follows: f_{smooth} more than 0.469, $f_{\text{complete,round}}$ more than 0.50, as shown in Table 1. The testing sets are constructed by choosing images from the above categories, and details are presented in Section 2.2.

2.2. Experimental Data Subsets

To mimic different cases of outlier detection, we construct two sorts of experimental data subsets by selecting images from the categories of galaxy images described in Section 2.1. One group of testing sets includes inliers containing a single type of galaxy morphology plus outliers containing a single type of galaxy morphology; the other group of testing sets is similar to the first ones for inliers but containing multiple types of galaxy morphology in the outliers.

Implicitly, the first group contains four data subsets, the inliers are all Completely round smooth galaxies, and the outliers are selected from other categories of galaxies separately. The fraction of outliers is 10% in each subset. The second group contains one data subset, the inliers are also Completely round smooth galaxies, but the outliers consist of galaxy images from other categories of galaxies. The total fraction of outliers is 10% as well, and the four types of galaxy images are equally constituted in the outliers.

Table 2 shows an overview of the above five testing sets, and columns denote the structure of each testing set. For instance, the first testing set (subsect1) consists of Completely round smooth galaxies (category 0) as inliers and Cigar-shaped smooth galaxies (category 2) as outliers. There are 16,000 inliers and 1778 outliers. Note that when lacking galaxy images of some categories, we expand the insufficient number of galaxy images by using data augmentation (see Section 4.1).

3. Methodology

For comparing the traditional methods and our state of art method, we build three approaches for outlier detection. The simplest one is based on KNN only, a classic clustering algorithm grounded on distance metrics. The second one involves CAE for feature extraction but still utilizes KNN for the clustering procedure. Finally, we employ the attention mechanism to improve the stability of the feature extraction with CAE. The following subsections demonstrate details of the construction of these approaches.

3.1. The KNN-based Approach

The KNN algorithm is one of the non-parametric classifying algorithms (Dasarathy 1991), whose core idea is to assume that data X has K nearest neighbors in the feature space. If most K neighbors belong to a certain category, the X could also be determined to belong to this category. As shown in Figure 1(a), the yellow rectangle is the data X needs to be predicted. Assuming $K = 3$, as shown in Figure 1(b), then the KNN algorithm will find the three neighbors closest to X (here enclosed in a circle) and select a category with the most elements. For example, in Figure 1(b), there are more elements described by red triangles, so the X is classified to the category containing elements described by red triangles. As shown in

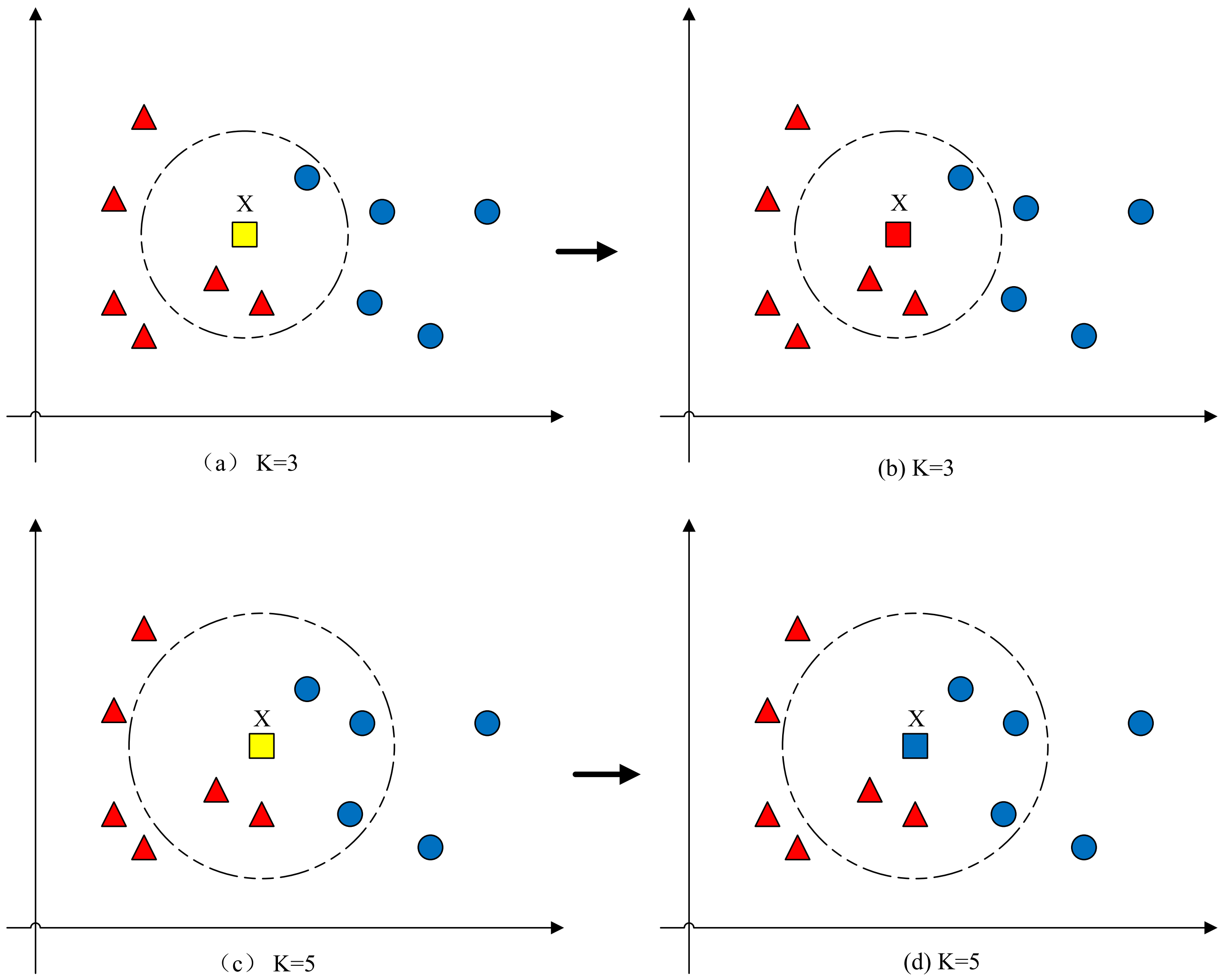

Figure 1. The classifying results based on classical KNN. (a), (b) The procedure of an element with yellow rectangle is classified to the category with red triangle when $K = 3$. (c), (d) The procedure of data X with yellow rectangle is classified to the category with blue circle when $K = 5$.

Figures 1(c) and (d), when $K = 5$, the X is classified to the category containing elements described by blue circles. Hu (2019) used KNN-based algorithms to perform classification experiments on a variety of data sets and achieved good results without any assumptions about the data. However, the KNN-based algorithm would cost considerable computing time due to the high data dimension in the case of astronomical images as input data.

3.2. The CAE-KNN-based Approach

CAE (Masci et al. 2011) is an optimized AE by adopting a convolution operation, which could extract principal features of astronomical image with high dimension. CAE_KNN makes full use of the CAE advantage in reducing the dimension to improve above KNN-based algorithm. We first present the architecture and components of CAE as shown in Figure 2, and then describe the joint of CAE and KNN.

CAE consists of two components: the encoder and the decoder. The first component is the encoder, which is responsible for extracting the representative features from input images. For an input image x, the j^{th} representative feature map h^j is expressed as Equation (1).

$$h^j = f(x^*W^j + b^j), \tag{1}$$

where W^j is the jth filter, * denotes the convolution operation, b^j is the corresponding bias of the feature map and f is an activation function. The activation function $f(z)$, where the input denotes by z used in the convolutional layers, is the Rectified Linear Unit (ReLu) (Bengio & LeCun 2007), as

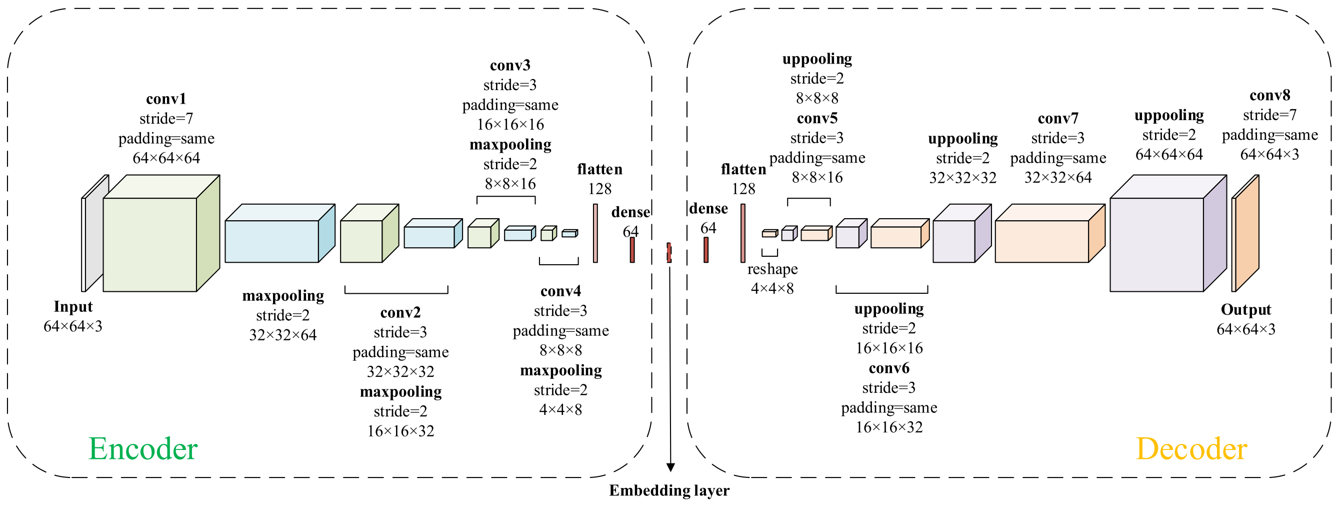

Figure 2. The architecture and components of CAE. CAE consists of two components, the encoder and the decoder. Green cuboids in encoder and orange cuboids in decoder denote convolution layer; blue cuboids indicate maxpooling layer; purple cuboids present uppooling layer.

described in Equation (2).

$$f(z) = \begin{cases} 0 & if\ z\ <\ 0 \\ z & if\ z\ \geqslant\ 0. \end{cases} \quad (2)$$

The encoder in this study is built with four convolutional layers (filter size: 64, 32, 16, and 8) and three dense layers (unit size: 128, 64, 32). A pooling layer follows each convolutional layer with 2 by 2 pixels. The pooling layer is also considered a down-sampling layer, aiming to reduce the volume of parameters involved in the encoder.

The second component of the CAE is the decoder, and its function is to reconstruct the input image according to the extracted feature map obtained by the encoder. The decoder structure is symmetrical with the structure of the encoder. In other words, its structure is simply the opposite of the encoder structure. As for the detail of reconstructing procedure, please refer to Masci et al. (2011); Cheng et al. (2020). The decoder has three dense layers (unit size: 32, 64, and 128), four convolutional layers (filter size: 8, 16, 32 and 64) using the ReLu activation function (Bengio & LeCun 2007), and an extra convolutional layer (filter size: 3) using the softmax (Ren et al. 2017) function as the output of the decoder. Except for the last output layer, there is an upsampling layer behind each convolutional layer, whose function is to gradually restore the feature maps to the same shape as input images. The layer between the encoder and the decoder is the embedding layer used to reconstruct the input galaxy images.

The loss function L between the two components is given by Equation (3) (Cheng et al. 2020).

$$L = -\frac{1}{N}[t^n \log y^n + (1 - t^n)\log(1 - y^n)], \quad (3)$$

where N is the number of samples, t^n is the target data, and y^n is the reconstructed data. The goal of CAE is to minimize the reconstruction error by using loss function L.

As so far, we could get the low dimension features from galaxy images by using the embedding layer vectors in CAE. Then these features are fed into the KNN algorithm avoiding the time-consuming problem of KNN outlier detection. However, the CAE_KNN has the disadvantage of instability because the background noise of the galaxy image sometimes influences the stability of the outlier detection.

3.3. The Attention-CAE_KNN-based Approach

To increase the stability of the CAE_KNN, we propose a novel algorithm, namely attCAE_KNN, which is the first time to explore the attention strategy to CAE. Attention strategy (Xu et al. 2015; Gregor et al. 2015) makes the attCAE_KNN focus on "what" is meaningful for given astronomical images so that attCAE_KNN could ignore the background noise. We build attCAE_KNN by adopting a convolutional block attention module (CBAM Liu et al. 2019). Its architecture is shown in Figure 3, including encoder, decoder, and KNN module. The decoder and KNN module have been described in Section 3.1 and Section 3.2. Next, we focus on the improved encoder.

The first part is the encoder that consists of the channel attention block and the spatial attention block (Liu et al. 2019), which differs from the classical encoder in inserting CBAM, as shown in Figure 4. These two blocks can extract the meaningful features of astronomical images along the two-dimensions of the channel axis and the spatial axis. The second part is the decoder, in which the CBAM is not inserted after the convolutional layer. This is because through the analysis of experimental results, adding the CBAM after the convolutional layer of the decoder hardly improves the experimental results. To reduce model

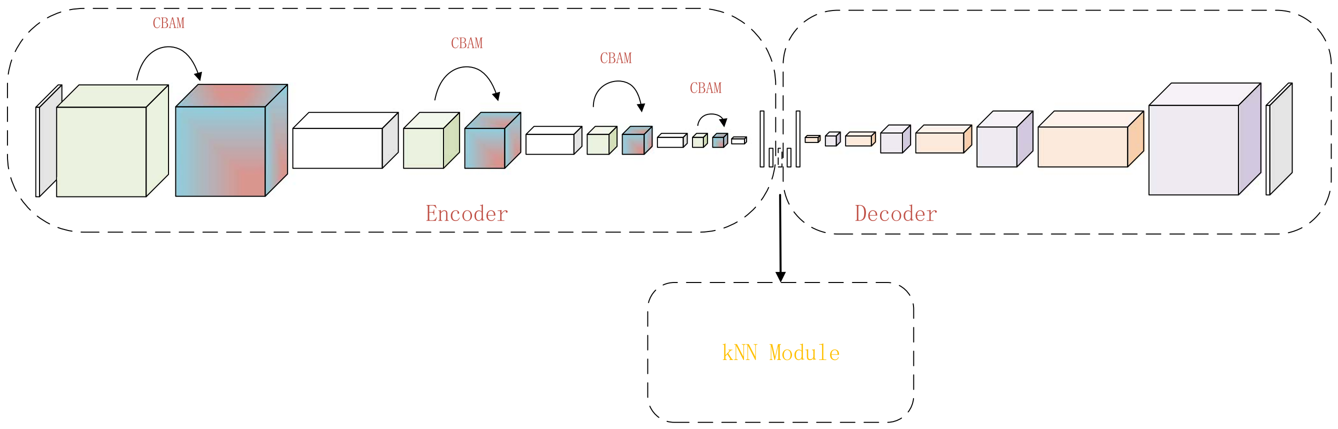

Figure 3. The architecture of the attCAE_KNN, including encoder, decoder and KNN module, where the CBAM attention strategy is added to the encoder.

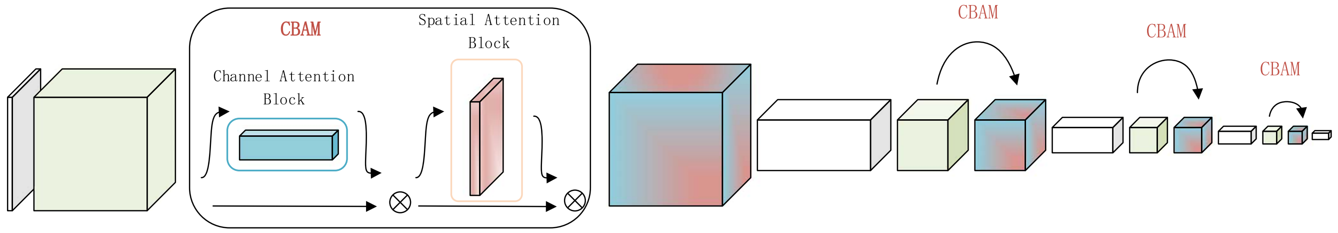

Figure 4. The encoder of the attCAE_KNN, which consists of the channel attention block, the spatial attention block and other CAE blocks.

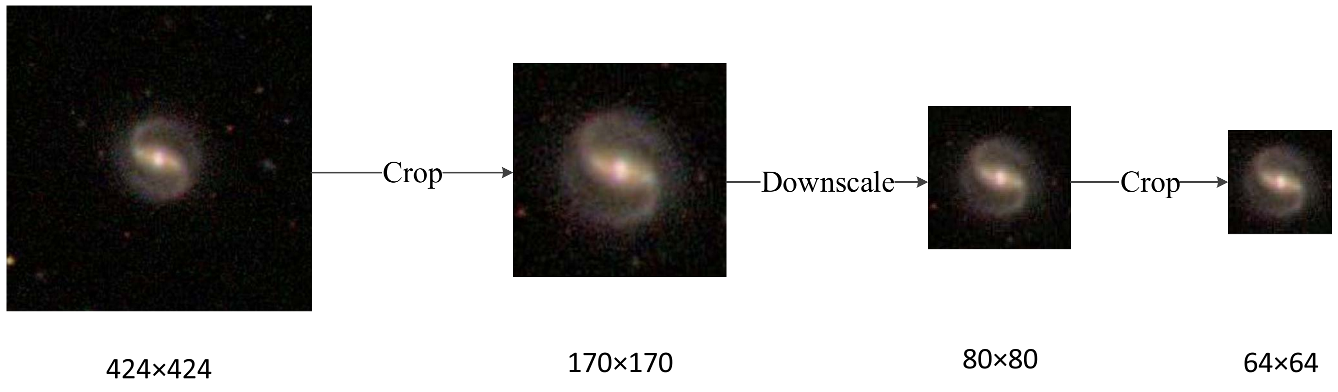

Figure 5. The procedure of data preprocessing on the original galaxy image.

complexity and decrease model training time, we only add the CBAM after the convolutional layer in the encoder of the CAE. The last part is the KNN module, whose input data is the latent features from the embedding layer of attCAE.

4. Experiment

We present the details of experiments with the data and methods described in Sections 2 and 3 here. It includes data processing, parameters of the machine learning models, evaluation metrics and experimental environments.

4.1. Data Pre-processing

As is shown in Section 2, we obtain 28,793 RGB color images with a size of $424 \times 424 \times 3$ pixels. Considering the valuable features of these images are concentrated at the central part, we conduct some pre-processing operations (see Figure 5). The first step is to crop the images with a box of 170×170 pixels in all channels. The second step is to downscale images from $170 \times 170 \times 3$ pixels to $80 \times 80 \times 3$ pixels. The last step is to crop images from $80 \times 80 \times 3$ pixels to $64 \times 64 \times 3$ pixels further. The detailed operations refer to the process in (Zhu et al. 2019).

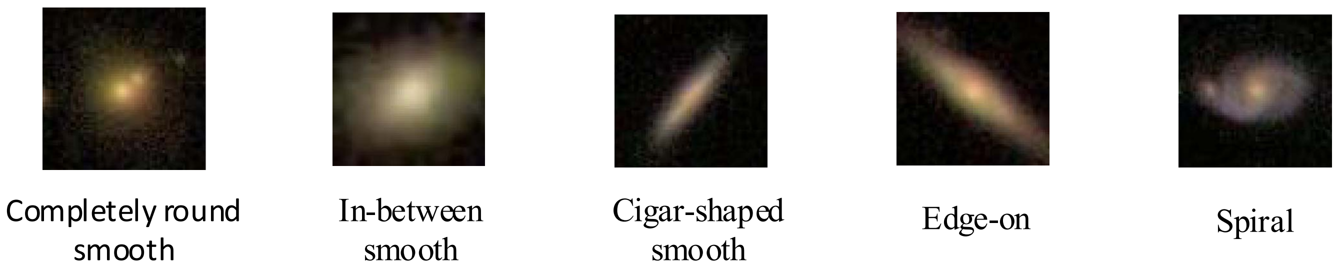

Figure 6. Five representative examples from five categories.

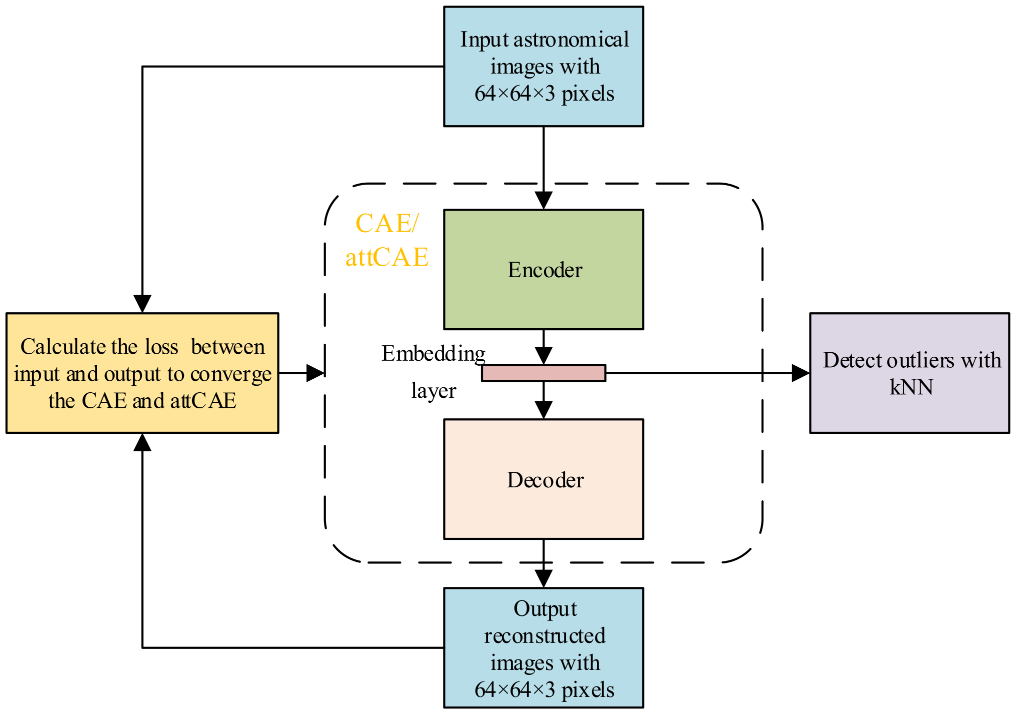
Input astronomical images with 64×64×3 pixels
CAE/ attCAE
Encoder
Embedding layer
Calculate the loss between input and output to converge the CAE and attCAE
Detect outliers with kNN
Decoder
Output reconstructed images with 64×64×3 pixels

Figure 7. The flow chart of the attCAE_KNN for detecting outliers in astronomical images.

Five processed examples from five categories described in Section 2 are displayed in Figure 6.

The number of images in some categories is too small to be outliers for supporting machine learning algorithms for outlier detection, for instance, there are only 579 Cigar-shaped smooth galaxies. Thus, we make data augmentation by rotating these images randomly and finally obtain 17,778 images in five data subsets, where each data subset consists of 16,000 inliers and 1778 outliers (see Table 2).

4.2. Training And Clustering

We apply the three methods (KNN, CAE_KNN, and attCAE_KNN) to the data subsets separately. The training process consists of auto-encoder training and KNN training. The former is for extracting the representative features of the astronomical images, while the latter is for detecting outliers. The flow chart of the attCAE_KNN for detecting outliers in astronomical images is shown in Figure 7.

The training process in this paper is entirely different from the training process in the context of supervised learning. We train CAE to extract features by comparing the input images and generated images, so no labels are included in the whole process. To avoid overfitting, we divide each data set shown in Table 2 into training sets and testing sets with a ratio of 7:3, and the images in the training set and test set are randomly selected from the whole set with 17,778 images. Considering that the number of outliers always accounts for a small part of the total data set, we set the proportion of outlier data to account for 10% of the whole data set for detecting outliers. For example, the number of outliers in the test set is 533, which can be calculated by $17,778 \times 0.3 \times 0.1$.

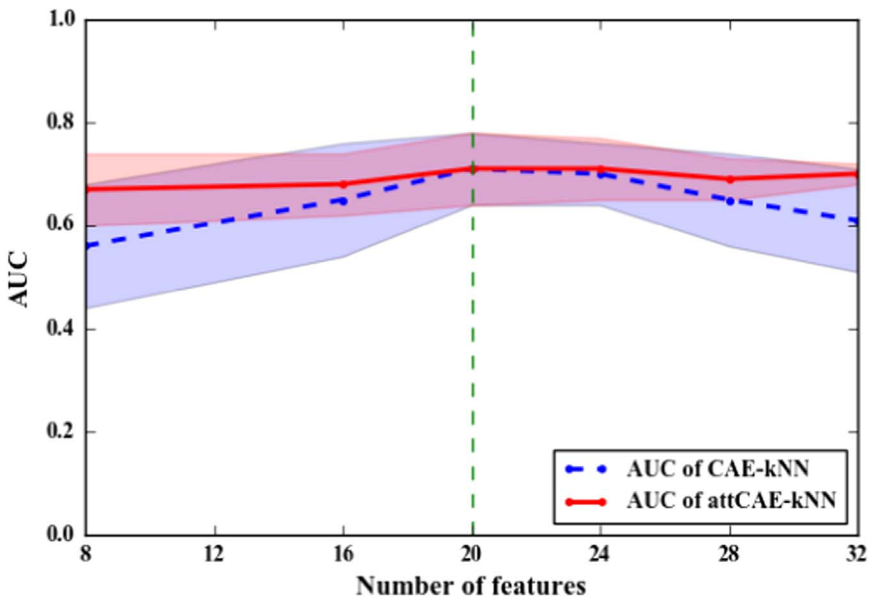

Figure 8. Effect on AUC mean values of various feature numbers in embedding layer.

During the training procedure of CAE, parameters of the embedded layer need to be optimized in a data-driven manner. We use area under the receiver operating characteristic curve (AUC Bradley 1997; Fawcett 2006) as the criteria. The receiver operating characteristic (ROC Fawcett 2006; Cheng et al. 2020) can be drawn with false-positive rates (FPR) and true-positive rates (TPR), which are given by Equations (4) and (5),

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}, \tag{4}$$

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{5}$$

where TP means true positive, TN means true negative, FP denotes false positive and FN means false negative, respectively. We then repeat the outlier detection process and compare the AUC of each classification to find the most optimal number of extracted features within the embedding layer in the CAE. In Figure 8, the blue dashed line shows the mean AUC of the outlier detection with CAE_KNN, while the solid red line shows the mean AUC of the outlier detection with attCAE_KNN. The lighter shadings present the standard deviation of the three results from the three training processes. One can see that the AUC of CAE_KNN and attCAE_KNN reach the maximum values when the feature number of the embedding layer is set to 20, which is, therefore, chosen to be the number of latent features in CAE and attCAE. In addition, it can also be seen that the stability of attCAE_KNN is higher than that of CAE_KNN.

The detailed implementation of KNN outlier detection refers to the modules in (Zhao et al. 2019; Ramaswamy et al. 2000), where there is a core procedure, namely *computeOutliersIndex*. The output data of procedure *computeOutliersIndex* is stored in a heap structure (Lattner & Adve 2003). We take the top 533 galaxy images with the largest values in a heap as outliers and then evaluate the model's performance based on the 533 outliers.

4.3. Evaluation Metrics

Besides AUC, we also employ Recall, F1 score, and Accuracy to estimate the performance of outlier detection (Cheng et al. 2020; Zhu et al. 2019; Hou 2019; Kamalov & Leung 2020) , which are given by Equations (6), (7), (8) and (9).

$$\mathrm{precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{6}$$

$$\mathrm{recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{7}$$

$$f1 = 2 \times \frac{\mathrm{precision} \times \mathrm{recall}}{presion + recall}, \tag{8}$$

$$\mathrm{accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}}. \tag{9}$$

Be worth mentioning, though Accuracy and F1 score are two of major performance metrics in many applications, they are considered supplements to AUC and Recall because the data distributions in this study are unbalanced (the ratio of the outliers is only 10%). In addition, TP+FN is equal to the TP +FP in all experiments, resulting in the values of *recall* being equal to the values of F1.

Table 3
Results of Experiment 1 (The Bold Entries Highlight our Results.)

	AUC	recall	f1	acc	Time
KNN	0.83	0.26	0.26	0.85	>4 h
CAE_KNN	0.94	0.57	0.57	0.91	10 min
attCAE_KNN	**0.97**	**0.76**	**0.76**	**0.95**	**10 min**

Table 4
Results of Experiment 2 (The Bold Entries Highlight our Results.)

	AUC	recall	f1	acc	Time
KNN	0.83	0.25	0.25	0.85	>4 h
CAE_KNN	0.95	0.56	0.56	0.91	10 min
attCAE_KNN	**0.98**	**0.78**	**0.78**	**0.96**	**10 min**

4.4. Implementation Details

The experimental environment of this study is as follows. We mainly use an Intel Xeon E5-2690 CPU and an Nvidia Tesla K40 GPU. Software environment includes python 3.5, Keras 2.3.1, NumPy 1.16.2, Matplotlib 3.0.3, scikit_learn 0.19.1, and pyod 0.8.4. It takes less than half an hour to train 17,778 images running on two NVIDIA Tesla K40 GPUs.

When training the CAE and attCAE, we set the batch_size to 128 and set epoch to 100, use the binary_crossentropy as described in Equation (3) and the Adam as optimizer. One can refer to the settings of the CBAM in Woo et al. (2018).

5. Results

This section presents the results of experiments described in Section 4. The outcomes of each experiment and the comparative analysis are listed in the following two subsections.

5.1. The Case of Single Type Inliers *and* Single Type Outliers

Experiment 1. We apply three methods illustrated in Section 3 to the testing set comprising images of Completely round smooth galaxies as inliers and images of Cigar-shaped smooth galaxies as outliers. Five metrics, i.e., Area under the ROC (AUC), Recall, F1 score, accuracy, and runtime, are utilized to evaluate outlier detection performance of the three methods. The results are shown in Table 3. Apparently, the attCAE–KNN approach obtains the best performance in all metrics. For instance, the *recall* using CAE–KNN is ∼31% higher than KNN, reaching 57%, while the *recall* using attCAE–KNN is ∼19% higher than CAE–KNN, which can reach 76%. Notably, the runtime of attCAE–KNN is also superior to other methods, and it is ∼4% of that of KNN alone.

Experiment 2. The second testing set contains images of Completely round smooth galaxies as inliers and images of Edge-on galaxies as outliers, as for the experimental parameters are similar to experiment 1. As is shown in Table 4, the results are also similar to experiment 1. One of the reasons is that the differences between inliers and outliers are both well distinguished in the first and second experiments.

Table 5
Results of Experiment 3 (The Bold Entries Highlight our Results.)

	AUC	recall	f1	acc	Time
KNN	0.64	0.11	0.11	0.82	>4 h
CAE_KNN	0.68	0.15	0.15	0.83	10 min
attCAE_KNN	**0.71**	**0.22**	**0.22**	**0.84**	**10 min**

Table 6
Results of Experiment 4 (The Bold Entries Highlight our Results.)

	AUC	recall	f1	acc	Time
KNN	0.68	0.15	0.15	0.83	>4 h
CAE_KNN	0.77	0.24	0.24	0.84	10 min
attCAE_KNN	**0.81**	**0.29**	**0.29**	**0.86**	**10 min**

Experiment 3. This experiment is similar to the previous one, except we adopt images of In-between smooth galaxies as outliers. However, as is shown in Table 5, the experimental results are different from previous ones because the similarity between inliers and outliers in this test set is less significant than the previous ones. Though including CAE and attention mechanism brings improvement, it is less considerable than the first two cases. For example, concerning *recall*, the CAE–KNN is ∼4% higher than KNN and only reaches 15%, while the *recall* of using attCAE–KNN is higher than CAE–KNN by ∼7% and reaches 22% only. It reveals that the definition of the outliers detection problem is crucial for outlier detection. According to the results, identifying smooth elliptical galaxies with specific ellipticity is not a practical outlier detection problem.

Experiment 4. Similarly, we adopt images of Spiral galaxies as outliers in this experiment. As expected (see Table 6), this experiment's results are better than experiment 3 but worse than experiments 1 and 2 because the distinguishability between inliers and outliers in this testing set is more noticeable than that in case 3 but less than cases 1 and 2 (mainly due to the PSF smearing). The most noteworthy difference between completely round-smooth and face-on Spiral galaxies is detailed structures and colors; thus, we hope the improvement of CAE and attention mechanism would be more significant when applying the methods to data from space-born telescopes.

Table 7
Results of Experiment 5 (The Bold Entries Highlight our Results.)

	AUC	recall	precision	f1	acc	Time
KNN	0.77	0.22	0.22	0.22	0.84	>4 h
CAE_KNN	0.85	0.43	0.43	0.43	0.87	10 min
attCAE_KNN 5%	**0.87**	**0.37**	**0.74**	**0.50**	**0.92**	**10 min**
attCAE_KNN 10%	**0.87**	**0.53**	**0.53**	**0.53**	**0.92**	**10 min**
attCAE_KNN 15%	**0.87**	**0.67**	**0.44**	**0.53**	**0.88**	**10 min**

5.2. *The Case of* Single Type Inliers And Multiple Type Outliers

The above experiments primarily explore the feasibility of unsupervised approaches for outlier detection with testing sets containing single type inliers and single type outliers. This subsection demonstrates an experiment in a more realistic case, i.e., the testing set contains a single type of inliers plus multiple types of outliers.

Experiment 5. We consider images of Completely round smooth galaxies as inliers and images of In-between smooth, Cigar-shaped smooth, Edge-on and Spiral as outliers. The experimental results are shown in Table 7, the attCAE_KNN still achieves the best performance. The *recall* of CAE_*k*NN reaches 43%, ∼21% higher than KNN, and the *recall* of attCAE–KNN is ∼10% higher than CAE–KNN, reaching to 53%. It is easy to conclude that the missing points in this experiment are dominated by In-between smooth galaxies.

Notably, recall and f1 values are the same in all the experiments because we define the most distant 10% objects to the center of the cluster of inliers in feature space as outliers during the detection of outliers, while the fraction of outliers in the testing set is 10%. Consequently, FN equals FP, then *recall* will equal *precision*, and hence *recall* equals *f*1 as well. However, when the chosen fraction does not equal the actual value, *recall* and *f*1 are not the same. The actual fraction of outliers is unknown in real cases; thus, it is impossible to choose a perfect fraction, and one needs to choose a rational fraction to define outliers according to specific scientific goals. We set the fraction to be 5% and 15%, in addition to illustrating comparative results. As is shown in Table 7, when the definition of outliers is the most distant 5% objects to the center of the inlier cluster, *recall* decreases to 0.37, *precision* increases to 0.74, and *f*1 is 0.5. Whereas, when the definition of outliers is the most distant 15% objects to the center of the inlier cluster, *recall*, *precision*, and *f*1 become 0.67, 0.44, and 0.53 separately. Accordingly, if one plans to obtain a sample of outliers with high completeness, a greater fraction (e.g., 15%) is needed, while if the goal is to find rare objects with noticeable and wired features efficiently, a lower fraction (e.g., 5%) is practical.

6. Discussion and Conclusions

In this study, we explore the feasibility of applying unsupervised learning to detect outliers in the data of galaxy images. First, we construct three methods, which are built upon the KNN, CAE + KNN, and attCAE_KNN separately. To evaluate the performance of the approaches, we organize two sorts of data sets based on the data of galaxy images given by the project of galaxy zoo challenge published on Kaggle. One group of testing sets includes inliers containing a single type of galaxy morphology plus outliers containing a single type of galaxy morphology; the other group of testing sets is similar to the first ones for inliers but with multiple types of galaxy morphology in the outliers. Comparing the results of applying three approaches to all the testing sets, we find that attCAE_KNN achieves the best performance and costs the least runtime, though its superiority is limited in the case of the testing set with a substantial similarity between inliers and outliers.

Specifically, KNN is usable for outlier detection, but its performance and efficiency are deficient. For instance, the best recall is 0.25, even when the testing set (testing set 1) has significant differences between inliers and outliers. The main reason for the shortcomings is the outdated procedure for extracting features. Therefore, we involve CAE as a module for feature extractions, and then the recall reaches 0.56 in the case of testing set 1. We further employ the attention mechanism to improve the stability of the feature extraction module, and the best recall goes to 0.78 in the case of testing set 1. Repeating the above process in other testing sets that contain single type inliers and single outliers, attCAE_KNN performs the best and costs the least runtime, and one can see more details in Tables 3–6.

To test the feasibility of the three methods in a more realistic context, we create testing set 5, containing single type inliers and multiple types outliers. As is expected, attCAE_KNN is still superior to the other two methods. For instance, its recall is 0.53, but the recalls of CAE_KNN and KNN are 0.43 and 0.22, respectively. As is shown in Table 7, the advantage of attCAE_KNN is evident over all five metrics. Hence, we can conclude that outlier detection in galaxy images is feasible by combing CAE and KNN, and the performance can be enhanced by involving the attention mechanism further. Besides, we implement a comparative investigation with different definitions of outliers when detecting them with our methods. The results in Table 7 demonstrate that a tighter definition of outliers leads to higher precision but lower recall, while a looser definition of outliers leads to lower precision but higher recall; nevertheless, the overall AUC is stable.

The structures of the testing sets used in the paper are relatively simple compared to real observations because we focus on assessing the feasibility of unsupervised approaches. To make our unsupervised approach suitable for real observations, we are forming a module to reduce any complex case (multiple types inliers + multiple types outliers) to the simple one employed in this paper (single type inliers + multiple types outliers) by combining human inspection and supervised learning. Then, we will apply the pipeline to actual survey data, such as KiDs, DES, and DESI legacy imaging surveys, to test its applicability and reliability. Also, to further improve the performance of approaches, particularly attCAE_kNN, we plan to optimize the architectures and hyper-parameters while applying them to observational data. Last but not least, defining the boundary of inliers and outliers is key to the outlier detection task, as is shown in the results in testing set 3. Hence, we will adopt a data-driven strategy to investigate the optimal definition of the boundaries according to specific scientific purposes.

In summary, unsupervised approaches, especially when we involve CAE and the attention mechanism, are feasible for outlier detection in the data sets of galaxy images. It is foreseen that unsupervised approaches can mine astronomical outliers so as to expand the boundary of human knowledge of the Universe in the big data era. On the other hand, the unsupervised approaches can also detect misclassified samples in standard supervised classification, similar to outlier detection, with no additional efforts. Accordingly, an ideal pipeline for classifying astronomical objects might need to combine supervised and unsupervised manners.

Acknowledgments

The data set used in this work is collected from the Galaxy-Zoo-Challenge-Project posted on the Kaggle platform. We acknowledge the science research grants from the China Manned Space Project with Nos. CMS-CSST-2021-A01 and CMS-CSST-2021-B05. Y.H., Z.Q.Z., and Y.L.C. are thankful for the funding and technical support from the Jiangsu Key Laboratory of Big Data Security and Intelligent Processing.

References

Barnett, V., & Lewis, T. 1996, International Journal of Forecasting, 12, 175
Baron, D., & Poznanski, D. 2017, MNRAS, 465, 4530
Beckman, R. J., & Cook, R. D. 1983, Technometrics, 25, 119
Bengio, Y., & LeCun, Y. 2007, Scaling learning algorithms toward AI, Large-Scale Kernel Machines (Cambridge, MA: MIT Press), 321
Bradley, A. P. 1997, PatRe, 30, 1145
Chalapathy, R., & Chawla, S. 2019, arXiv:1901.03407
Chen, S.-X., Sun, W.-M., & He, Y. 2022, RAA, 22, 025017
Cheng, T.-Y., Li, N., Conselice, C. J., et al. 2020, MNRAS, 494, 3750
Cheng, T. Y., Marc, H. C., Conselice, C. J., et al. 2021, MNRAS, 503, 4446
D'Addona, M., Riccio, G., Cavuoti, S., et al. 2021, in Intelligent Astrophysics, ed. I. Zelinka, M. Brescia, & D. Baron, Vol 39 (Switzerland AG: Springer Nature), 225
Dasarathy, B. V. 1991, Nearest Neighbor (NN) Norms: Nn Pattern Classification Techniques (Los Alamitos, CA: IEEE Computer Society Press)
Dutta, H., Giannella, C., Borne, K., & Kargupta, H. 2007, in Proc. 2007 SIAM Int. Conf. on Data Mining (SDM '07) (Philadelphia, PA: SIAM), 473
Edgeworth, F. Y. 1888, J. Roy. Stat. Soc., 51, 113
Fawcett, T. 2006, PatRe, 27, 861
Fustes, D., Manteiga, M., Dafonte, C., et al. 2013, A&A, 559, A7
Giles, D., & Walkowicz, L. 2019, MNRAS, 484, 834
Gregor, K., Danihelka, I., Graves, A., Rezende, D., & Wierstra, D. 2015, in Proc. 32nd Int. Conference on Machine Learning, 37 (Lille, France: PMLR), 1462
Gupta, R., Srijith, P. K., & Desai, S. 2022, A&C, 38, 100543
Hart, R. E., Bamford, S. P., Willett, K. W., et al. 2016, MNRAS, 461, 3663
Hawkins, D. M. 1980, Identification of Outliers, Vol. 11 (Berlin: Springer)
Hendrycks, D., Mazeika, M., & Dietterich, T. 2018, arXiv:1812.04606
Hu, D. 2020, in Proc. SAI Intelligent Systems Conf. (Berlin: Springer), 432
Ishida, E. E. O., Kornilov, M. V., Malanchev, K. L., et al. 2021, A&A, 650, A195
Kamalov, F., & Leung, H. H. 2020, Journal of Information & Knowledge Management, 19, 2040013
Lattner, C., & Adve, V. 2003, Data structure analysis: a fast and scalable context-sensitive heap analysis UIUCDCS-R-2003-2340, Computer Science Dept., Univ. of Illinois at Urbana-Champaign
Lintott, C. J., Schawinski, K., Slosar, A., et al. 2008, MNRAS, 389, 1179
Liu, T., Kong, J., Jiang, M., & Huo, H. 2019, J. Electron. Imaging, 28, 023012
Lochner, M., & Bassett, B. A. 2021, A&C, 36, 100481
Lukic, V., de Gasperin, F., & Brüggen, M. 2019, Galax, 8, 3
Margalef-Bentabol, B., Huertas-Company, M., Charnock, T., et al. 2020, MNRAS, 496, 2346
Masci, J., Meier, U., Cireşan, D., & Schmidhuber, J. 2011, in Int. Conf. on Artificial Neural Networks (Berlin: Springer), 52
Nadeem, M., Marshall, O., Singh, S., Fang, X., & Yuan, X. 2016, in CCERP 2016 (Kennesaw State University)
Pearsons, K. S., Barber, D. S., Tabachnick, B. G., & Fidell, S. 1995, JASA, 97, 331
Pruzhinskaya, M. V., Malanchev, K. L., Kornilov, M. V., et al. 2019, MNRAS, 489, 3591
Ramaswamy, S., Rastogi, R., & Shim, K. 2000, SIGMOD Rec., 29, 427–438
Ren, Y., Zhao, P., Sheng, Y., Yao, D., & Xu, Z. 2017, in Proc. 26th Int. Joint Conf. on Artificial Intelligence, 2641
Reyes, E., & Estévez, P. A. 2020, arXiv:2005.07779
Sharma, K., Kembhavi, A., Kembhavi, A., Sivarani, T., & Abraham, S. 2019, Bull. Soc. R. Sci. Liege, 88, 174
Solarz, A., Bilicki, M., Gromadzki, M., et al. 2017, A&A, 606, A39
Storey-Fisher, K., Huertas-Company, M., Ramachandra, N., et al. 2020, arXiv:2012.08082
Vaswani, A., Shazeer, N., Parmar, N., et al. 2017, Attention is all you need, in NeurIPS 2017, 6000
Ventura, P., & D'Antona, F. 2011, MNRAS, 410, 2760
Vincent, P., Larochelle, H., Lajoie, I., et al. 2010, JMLR, 11, 3371
Webb, S., Lochner, M., Muthukrishna, D., et al. 2020, MNRAS, 498, 3077
Willett, K. W., Lintott, C. J., Bamford, S. P., et al. 2013, MNRAS, 435, 2835
Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. 2018, in Proc. European Conf. on Computer Vision (ECCV), 3
Xie, J., Hou, Q., & Cao, J. 2019, Journal of Frontiers of Computer Science and Technology, 13, 586
Xu, K., Ba, J., Kiros, R., et al. 2015, in Proc. 32nd Int. Conf. on Machine Learning, 37 (Lille, France: PMLR), 2048
Zhang, Y.-X., Luo, A. L., & Zhao, Y.-H. 2004, Proc. SPIE, 5493, 521
Zhang, Z., Zou, Z., Li, N., & Chen, Y. 2022, RAA, 22, 055002
Zhao, Y., Nasrullah, Z., & Li, Z. 2019, arXiv:1901.01588
Zhou, C., & Paffenroth, R. C. 2017, in Proc. 33rd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, 665
Zhu, X.-P., Dai, J.-M., Bian, C.-J., et al. 2019, Ap&SS, 364, 1