Robust, Deep and Inductive Anomaly Detection

Raghavendra Chalapathy[1], Aditya Krishna Menon[2(✉)], and Sanjay Chawla[3]

[1] University of Sydney and Capital Markets Cooperative Research Centre (CMCRC), Sydney, Australia
rcha9612@uni.sydney.edu.au
[2] Data61/CSIRO, Australian National University, Canberra, Australia
aditya.menon@data61.csiro.au
[3] Qatar Computing Research Institute, Al Rayyan, Qatar
schawla@qf.org.qa

Abstract. PCA is a classical statistical technique whose simplicity and maturity has seen it find widespread use for anomaly detection. However, it is limited in this regard by being sensitive to gross perturbations of the input, and by seeking a linear subspace that captures normal behaviour. The first issue has been dealt with by *robust PCA*, a variant of PCA that explicitly allows for some data points to be arbitrarily corrupted; however, this does not resolve the second issue, and indeed introduces the new issue that one can no longer inductively find anomalies on a test set. This paper addresses both issues in a single model, the *robust autoencoder*. This method learns a nonlinear subspace that captures the majority of data points, while allowing for some data to have arbitrary corruption. The model is simple to train and leverages recent advances in the optimisation of deep neural networks. Experiments on a range of real-world datasets highlight the model's effectiveness.

Keywords: Anomaly detection · Outlier detection · Robust PCA Autoencoders · Deep learning

1 Anomaly Detection: Motivation and Challenges

A common need when analysing real-world datasets is determining which instances stand out as being dramatically dissimilar to all others. Such instances are known as *anomalies*, and the goal of *anomaly detection* (also known as *outlier detection*) is to determine all such instances in a data-driven fashion [9]. Anomalies can be caused by errors in the data but sometimes are indicative of a new, previously unknown, underlying process; in fact Hawkins [14] defines an outlier as an observation that *deviates so significantly from other observations as to arouse suspicion that it was generated by a different mechanism.*

Principal Component Analysis (PCA) [15] is a core method for a range of statistical inference tasks, including anomaly detection. The basic idea of PCA is that while many data sets are high-dimensional, they tend to inhabit a low-dimensional manifold. PCA thus operates by (linearly) projecting data into a

M. Ceci et al. (Eds.): ECML PKDD 2017, Part I, LNAI 10534, pp. 36–51, 2017.
https://doi.org/10.1007/978-3-319-71249-9_3

lower-dimensional space, so as to separate the *signal* from the *noise*; a data point which is far away from its projection is deemed as anomalous.

While intuitive and popular, PCA has limitations as an anomaly detection method. Notably, it is highly sensitive to data perturbation: one extreme data point can completely change the orientation of the projection, often leading to the masking of anomalies. A variant of PCA, known as a *robust* PCA (RPCA) limits the impact of anomalies by using a clever decomposition of the data matrix [8]. We will discuss RPCA in detail in Sect. 2, but note here that it still carries out a linear projection, and further cannot be used to make predictions on test instances; that is, we cannot perform *inductive* anomaly detection.

In this paper, we will relax the linear projection limitation of RPCA by using a deep and robust autoencoder [13,30]. The difference between RPCA and a deep autoencoder will be the use of a nonlinear activation function and the potential use of several hidden layers in the autoencoder. While this modification is conceptually simple, we show it yields noticeable improvements in anomaly detection performance on complex real-world image data, where a linear projection cannot capture sufficient structure in the data. Further, the robust autoencoder is capable of performing inductive anomaly detection, unlike RPCA.

In the sequel, we provide an overview of anomaly detection methods (Sect. 2), with a specific emphasis on matrix decomposition techniques such as PCA and its robust extensions. We then proceed to describe our proposed model based on autoencoders (Sect. 3), and present our experiment setup and results (Sects. 4 and 5). Finally, we describe directions for future work (Sect. 6).

2 Background and Related Work on Anomaly Detection

Consider a feature matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$, where N denotes the number of data points and D the number of features for each point. For example, N could be the number of images in some photo collection, and D the number of pixels used to represent each image. The goal of anomaly detection is to determine which rows of $\mathbf{X}$ are anomalous, in the sense of being dissimilar to all other rows. We will use $\mathbf{X}_{i:}$ to denote the ith row of $\mathbf{X}$.

2.1 A Tour of Anomaly Detection Methods

Anomaly detection is a widely researched topic in the data mining and machine learning community [2,9]. The two primary strands of research have been the design of novel algorithms to detect anomalies, and the design *efficient* means of discovering all anomalies in a large dataset. In the latter strand, starting from the work of Bay and Schwabacher [4], several optimisations have been proposed to discover anomalies in near linear time [12].

In the former strand, which is our primary focus, most emphasis has been on non-parametric methods like distance and density based outliers [7,21]. For example, distance-based methods define a domain-dependent dissimilarity metric, and deem a point to be anomalous if it is relatively far away from its

neighbours [35]. Another popular approach is the one-class SVM, which learns a smooth boundary that captures the majority of probability mass of the data [27].

In recent years, matrix factorization methods for anomaly detection have become popular. These methods provide a *reconstruction matrix* $\hat{\mathbf{X}} \in \mathbb{R}^{N \times D}$ of the input $\mathbf{X}$, and use the norm $\|\mathbf{X}_{i:} - \hat{\mathbf{X}}_{i:}\|_2^2$ as a measure of how anomalous a particular point $\mathbf{X}_{i:}$ is; if the reconstruction is close to the input, then it is deemed normal; else, anomalous. We describe several popular examples of this approach, beginning with principal component analysis (PCA).

2.2 PCA for Anomaly Detection

PCA finds the directions of maximal variance of the data. Supposing without loss of generality that the data matrix $\mathbf{X}$ has zero mean, this may be understood as the result of a matrix factorisation [6]:

$$\min_{\mathbf{W}^T\mathbf{W}=\mathbf{I},\mathbf{Z}} \|\mathbf{X} - \mathbf{W}\mathbf{Z}\|_F^2 = \min_{\mathbf{U}} \|\mathbf{X} - \mathbf{X}\mathbf{U}\mathbf{U}^T\|_F^2. \tag{1}$$

Here, the reconstruction matrix is $\hat{\mathbf{X}} = \mathbf{X}\mathbf{U}\mathbf{U}^T$, where $\mathbf{U} \in \mathbb{R}^{D \times K}$ for some number of *latent dimensions* $K \ll D$. We can interpret $\mathbf{X}\mathbf{U}$ as a projection (or encoding) of $\mathbf{X}$ into a K-dimensional subspace, with the application of $\mathbf{U}^T$ as an inverse projection (or decoding) back into the original D dimensional space.

2.3 Autoencoders for Anomaly Detection

PCA assumes a linear subspace explains the data. To relax this assumption, consider instead

$$\min_{\mathbf{U},\mathbf{V}} \|\mathbf{X} - f(\mathbf{X}\mathbf{U})\mathbf{V}\|_F^2 \tag{2}$$

for some non-decreasing *activation function* $f\colon \mathbb{R} \to \mathbb{R}$, and $\mathbf{U} \in \mathbb{R}^{D \times K}, \mathbf{V} \in \mathbb{R}^{K \times D}$. For the purposes of anomaly detection, one can define the reconstruction matrix as $\hat{\mathbf{X}} = f(\mathbf{X}\mathbf{U})\mathbf{V}$.

Equation 2 corresponds to an autoencoder with a single hidden layer [13]. Popular choices of $f(\cdot)$ include the sigmoid $f(a) = (1 + \exp(-a))^{-1}$ and the rectified linear unit or ReLU $f(x) = \max(0, a)$. As before, we can interpret $\mathbf{X}\mathbf{U}$ as an encoding of $\mathbf{X}$ into a K-dimensional subspace; however, by applying a nonlinear $f(\cdot)$, the projection is implicitly onto a nonlinear manifold.

2.4 Robust PCA

Another way to generalise PCA is to solve, for a tuning parameter $\lambda > 0$,

$$\min_{\mathbf{S},\mathbf{N}} \|\mathbf{S}\|_* + \lambda \cdot \|\mathbf{N}\|_1 : \mathbf{X} = \mathbf{S} + \mathbf{N}, \tag{3}$$

where $\|\cdot\|_*$ denotes the trace or nuclear norm $\|\mathbf{X}\|_* = \mathrm{tr}((\mathbf{X}^T\mathbf{X})^{1/2})$, and $\|\cdot\|_1$ the elementwise ℓ_1 norm. For the purposes of anomaly detection, one can define the reconstruction matrix $\hat{\mathbf{X}} = \mathbf{X} - \mathbf{N} = \mathbf{S}$.

Intuitively, Eq. 3 separates $\mathbf{X}$ into a signal matrix $\mathbf{S}$ and a noise matrix $\mathbf{N}$, where the signal matrix has low-rank structure, and the noise is assumed to not overwhelm the signal for most of the matrix entries. The trace norm may be seen as a convex relaxation of the rank function; thus, this objective can be understood as a relaxed version of PCA.

Equation 3 corresponds to robust PCA (RPCA) [8]. Unlike standard PCA, this objective can effortlessly deal with a single entry perturbed arbitrarily. When $\lambda \to +\infty$, we will end up with $\mathbf{N} = \mathbf{0}, \mathbf{S} = \mathbf{X}$, i.e. we will claim that there is no noise in the data, and so all points are deemed normal. On the other hand, when $\lambda \to 0$, we will end up with $\mathbf{N} = \mathbf{X}, \mathbf{S} = \mathbf{0}$, i.e. we will claim that there is no signal in the data, and so points with high norm are deemed anomalous.

2.5 Direct Robust Matrix Factorization

Building upon RPCA, Xiong et al. [32] introduced the direct robust matrix factorization method (DRMF), where for tuning parameters K, e one solves:

$$\min_{\mathbf{S},\mathbf{N}} \quad \|\mathbf{X} - (\mathbf{N} + \mathbf{S})\|_F^2 \colon \operatorname{rank}(\mathbf{S}) \leq K, \|\mathbf{N}\|_0 \leq e. \tag{4}$$

As before, the matrix $\mathbf{N}$ captures the anomalies and $\mathbf{S}$ captures the signal. Unlike RPCA, one explicitly constraints $\mathbf{S}$ to be low-rank, rather than merely having low trace norm; and one explicitly constraints $\mathbf{N}$ to have a maximal number of nonzeros, rather than merely having bounded ℓ_1 norm. The lack of convexity of the objective requires a bespoke algorithm for the optimisation.

2.6 Robust Kernel PCA

Another way to overcome the linear assumption of PCA is the robust kernel PCA (RKPCA) approach of [25]. For a feature mapping Φ into a reproducing kernel Hilbert space, and projection operator $\mathbf{P}$ of a point into the KPCA subspace, it is proposed to reconstruct an input $\mathbf{x} \in \mathbb{R}^D$ by solving the pre-image problem

$$\hat{\mathbf{x}} = \operatorname*{argmin}_{\mathbf{z} \in \mathbb{R}^D} E_0(\mathbf{x}, \mathbf{z}) + C \cdot \|\Phi(\mathbf{z}) - \mathbf{P}\Phi(\mathbf{z})\|^2, \tag{5}$$

where E_0 is a robust measure of reconstruction error (i.e. not merely the Euclidean norm), and $C > 0$ is a tuning parameter. RKPCA does not explicitly handle gross outliers, unlike RPCA; however, by choosing a rich feature mapping Φ, one can capture nonlinear anomalies. This choice of feature mapping must be pre-specified, whereas autoencoder methods implicitly *learn* a good mapping.

3 From Robust PCA to Robust Autoencoders

We now present our robust (convolutional) autoencoder model for anomaly detection. The method can be seen as an extension of robust PCA to allow for a nonlinear manifold that explains most of the data.

3.1 Robust (Convolutional) Autoencoders

Let $f : \mathbb{R} \to \mathbb{R}$ be some non-decreasing activation function. Now consider the following objective, which combines the salient elements of Eqs. 2 and 3:

$$\min_{\mathbf{U},\mathbf{V},\mathbf{N}} \|\mathbf{X} - (f(\mathbf{X}\mathbf{U})\mathbf{V} + \mathbf{N})\|_F^2 + \frac{\mu}{2} \cdot (\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2) + \lambda \cdot \|\mathbf{N}\|_1, \tag{6}$$

where $f(\cdot)$ is understood to act elementwise, and $\lambda, \mu > 0$ are tuning parameters. This is a form of *robust autoencoder*: one encodes the input into the latent representation $\mathbf{Z} = f(\mathbf{X}\mathbf{U})$, which is then decoded via $\mathbf{V}$. The additional $\mathbf{N}$ term captures gross outliers in the data, as with robust PCA. For the purposes of anomaly detection, we have reconstruction matrix $\hat{\mathbf{X}} = f(\mathbf{X}\mathbf{U})\mathbf{V}$.

When $\lambda \to +\infty$, we get $\mathbf{N} = \mathbf{0}$, and the model reduces to a standard autoencoder (Eq. 2). When $\lambda \to 0$, then one possible solution is $\mathbf{N} = \mathbf{X}$ and $\mathbf{U} = \mathbf{V} = \mathbf{0}$, so that the model memorises the training data. For intermediate λ, the model augments a standard autoencoder with a noise absorption term that endows robustness.

More generally, Eq. 6 can be seen as an instance of

$$\min_{\theta,\mathbf{N}} \|\mathbf{X} - (\hat{\mathbf{X}}(\theta) + \mathbf{N})\|_F^2 + \frac{\mu}{2} \cdot \Omega(\theta) + \lambda \cdot \|\mathbf{N}\|_1, \tag{7}$$

where $\hat{\mathbf{X}}(\theta)$ is some generic predictor with parameters θ, and $\Omega(\cdot)$ a regularisation function. Observe that we could pick $\hat{\mathbf{X}}(\theta)$ to be a convolutional autoencoder [19, 30], which would be suitable when dealing with image data; such a model will be studied extensively in our experiments. Further, the regulariser Ω could involve more general matrix norms, such as the $\ell_{1,2}$ norm [16].

3.2 Training the Model

The objective function of the model of Eqs. 6, 7 is non-convex, but unconstrained and sub-differentiable. There are several ways of performing optimisation. For example, for differentiable activation f, one could compute sub-gradients with respect to all model parameters and apply backpropagation. However, to leverage existing advances in training deep networks, we observe that:

- For fixed $\mathbf{N}$, the objective is equivalent to that of a standard (convolutional) autoencoder on the matrix $\mathbf{X} - \mathbf{N}$. Thus, one can optimise the parameters θ using any modern (stochastic) optimisation tool for deep learning that exploits gradients, such as Adam [20].
- For fixed θ (i.e. $\mathbf{U}, \mathbf{V}$ in the standard autoencoder case), the objective is

$$\min_{\theta,\mathbf{N}} \|\mathbf{N} - (\mathbf{X} - \hat{\mathbf{X}}(\theta))\|_F^2 + \lambda \cdot \|\mathbf{N}\|_1,$$

  which trivially solvable via the soft thresholding operator on the matrix $\mathbf{X} - \hat{\mathbf{X}}(\theta)$ [3], with solution

$$\mathbf{N}_{ij} = \begin{cases} (\mathbf{X} - \hat{\mathbf{X}}(\theta))_{ij} - \frac{\lambda}{2} & \text{if } (\mathbf{X} - \hat{\mathbf{X}}(\theta))_{ij} > \frac{\lambda}{2} \\ (\mathbf{X} - \hat{\mathbf{X}}(\theta))_{ij} + \frac{\lambda}{2} & \text{if } (\mathbf{X} - \hat{\mathbf{X}}(\theta))_{ij} < -\frac{\lambda}{2} \\ 0 & \text{else.} \end{cases}$$

We thus alternately optimise $\mathbf{N}$ and θ until the change in the overall objective is below some threshold. The use of stochastic optimisation for the first step, and the simplicity of the optimisation for the second step, means that we can easily train the model where data arrives in an online or streaming fashion.

3.3 Predicting with the Model

One convenient property of our model is that the anomaly detector will be inductive, i.e. it can generalise to unseen data points. One can interpret the model as learning a robust representation of the input, which is unaffected by gross noise; such a representation should thus be able to accurately model any unseen points that lie on the same manifold as the data used to train the model.

Formally, given a new $\mathbf{x}_* \in \mathbb{R}^D$, one simply computes $f(\mathbf{x}_*^T\mathbf{U})\mathbf{V}$ to score this point. The larger $\|\mathbf{x}_* - \mathbf{V}^T f(\mathbf{U}^T\mathbf{x}_*)\|_2^2$ is, the more likely the point is deemed to be anomalous. We emphasise that such inductive predictions are simply not possible with the robust PCA method, as it estimates parameters for the $N \times D$ observations present in $\mathbf{X}$, with no means of generalising to unseen data.

3.4 Connection to Robust PCA

While the robust autoencoder of Eq. 6 has clear conceptual similarity to robust PCA, it may seem that choices such as the ℓ_2 penalty on $\mathbf{U}, \mathbf{V}$ are somewhat arbitrarily used in place of the trace norm. We now show how the objective can in fact be naturally derived as an extension of RPCA.

The trace norm can be represented in the variational form [26] $\|\mathbf{S}\|_* = \min_{\mathbf{WV}=\mathbf{S}} \frac{1}{2} \cdot (\|\mathbf{W}\|_F^2 + \|\mathbf{V}\|_F^2)$. The robust PCA objective is thus equivalently

$$\min_{\mathbf{W},\mathbf{V},\mathbf{N}} \frac{1}{2} \cdot (\|\mathbf{W}\|_F^2 + \|\mathbf{V}\|_F^2) + \lambda \cdot \|\mathbf{N}\|_1 : \mathbf{X} = \mathbf{WV} + \mathbf{N}.$$

This objective has the disadvantage of being non-convex, but the advantage of being amenable to extensions. Pick some $\mu > 0$, and consider a relaxed version of the robust PCA objective:

$$\min_{\mathbf{W},\mathbf{V},\mathbf{N},\mathbf{E}} \|\mathbf{E}\|_F^2 + \frac{\mu}{2} \cdot (\|\mathbf{W}\|_F^2 + \|\mathbf{V}\|_F^2) + \lambda \cdot \|\mathbf{N}\|_1 : \mathbf{X} = \mathbf{WV} + \mathbf{N} + \mathbf{E}.$$

Here, we allow for further systematic errors $\mathbf{E}$ which have low average magnitude. We can equally consider the unconstrained objective

$$\min_{\mathbf{W},\mathbf{V},\mathbf{N}} \|\mathbf{X} - (\mathbf{WV} + \mathbf{N})\|_F^2 + \frac{\mu}{2} \cdot (\|\mathbf{W}\|_F^2 + \|\mathbf{V}\|_F^2) + \lambda \cdot \|\mathbf{N}\|_1 \tag{8}$$

This re-expression of robust PCA has been previously noted, for example in Sprechmann et al. [29]. To derive the robust autoencoder from Eq. 8, suppose now that we constrain $\mathbf{W} = \mathbf{XU}$. This is a natural constraint in light of Eq. 1, since for standard PCA we factorise $\mathbf{X}$ into $\hat{\mathbf{X}} = \mathbf{XUU}^T$. Then, we have the objective

$$\min_{\mathbf{U},\mathbf{V},\mathbf{N}} \|\mathbf{X} - (\mathbf{XUV} + \mathbf{N})\|_F^2 + \frac{\mu}{2} \cdot (\|\mathbf{XU}\|_F^2 + \|\mathbf{V}\|_F^2) + \lambda \cdot \|\mathbf{N}\|_1.$$

Now suppose we modify the regulariser to only operate on $\mathbf{U}$ rather than $\mathbf{XU}$:

$$\min_{\mathbf{U},\mathbf{V},\mathbf{N}} \|\mathbf{X} - (\mathbf{XUV} + \mathbf{N})\|_F^2 + \frac{\mu}{2} \cdot (\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2) + \lambda \cdot \|\mathbf{N}\|_1.$$

This is again natural in the context of standard PCA, since there we have $\mathbf{W} = \mathbf{XU}$ satisfying $\mathbf{W}^T\mathbf{W} = \mathbf{I}$. Observe now that we have derived Eq. 6 for a linear activation function $f(x) = x$. The robust autoencoder thus extends this model by employing a nonlinear activation.

3.5 Relation to Existing Models

Our contribution is a nonlinear extension of RPCA for anomaly detection. As noted above, the key advantages over RPCA are the ability to capture nonlinear structure in the data, as well as the ability to detect anomalies in an inductive setting. The price we have to pay is the lack of convexity of the objective function, unlike RPCA; nonetheless, we shall demonstrate that the model can be effectively trained using the procedure described in Sect. 3.2.

Some works have employed deep networks for anomaly detection [31,34], but without explicitly accounting for gross anomalies. For example, the recent work of [34] employed an autoencoder-inspired objective to train a probabilistic neural network, with extensions to structured data; the use of an RPCA-style noise matrix $\mathbf{N}$ may be useful to explore in conjunction with such methods.

Our method is also distinct to denoising autoencoders (DNA), wherein noise is explicitly added to instances [30], whereas we *infer* the noise automatically. The approaches have slightly different goals: DNAs aim to extract good features from the data, while our aim is to identify anomalies.

Finally, while nonlinear extensions of PCA-style matrix factorisation (including via autoencoders) have been explored in contexts such as collaborative filtering [23,28], we are unaware of prior usage for anomaly detection.

4 Experimental Setup

In this section we show the empirical effectiveness of Robust Convolutional Autoencoder over the state-of-the-art methods on real-world data. Our primary focus will be on non-trivial image datasets, although our method is applicable in any context where autoencoders are useful e.g. speech.

4.1 Methods Compared

We compare our proposed Robust Convolutional Autoencoder (RCAE) with the following state-of-the art methods for anomaly detection:

- **Truncated SVD**, which for zero-mean features is equivalent to PCA.
- **Robust PCA (RPCA)** [8], as per Eq. 3.

- **Robust kernel PCA (RKPCA)** [25], as per Eq. 5.
- **Autoencoder (AE)** [5], as per Eq. 2.
- **Convolutional Autoencoder (CAE)**, a convolutional autoencoder without any accounting for gross anomalies i.e. Eq. 7 where $\lambda = +\infty$.
- **Robust Convolutional Autoencoder (RCAE)**, our proposed model as per Eq. 7.

We used TensorFlow [1] for the implementation of AE, CAE and RCAE[1]. For RPCA and RKPCA, we used publicly available implementations[2,3].

4.2 Datasets

We compare all methods on three real-world datasets:

- `restaurant`, comprising video background modelling and activity detection consisting of snapshots of restaurant activities [32].
- `usps`, comprising the USPS handwritten digits [17].
- `cifar-10` consisting of 60000 32×32 colour images in 10 classes, with 6000 images per class [22].

For each dataset, we perform further processing to create a well-posed anomaly detection task, as described in the next section.

4.3 Evaluation Methodology

As anomaly detection is an unsupervised learning problem, model evaluation is challenging. For the `restaurant` dataset, there are no ground truth anomalies, and so we perform a qualitative analysis by visually comparing the anomalies flagged by various methods, as done in the original robust PCA paper [8].

For the other two datasets, we follow a standard protocol (see e.g. [32]) wherein anomalies are explicitly identified in the training set. We can then evaluate the predictive performance of each method as measured against the ground truth anomaly labels, using three standard metrics:

- the area under the precision-recall curve (AUPRC)
- the area under the ROC curve (AUROC)
- the precision at 10 (P@10).

AUPRC and AUROC measure ranking performance, with the former being preferred under class imbalance [11]. P@10 measures classification performance, being the fraction of the top 10 scored instances which are actually anomalous.

For $\texttt{CIFAR} - \texttt{10}$, the labelled dataset is created by combining 5000 images of dog and 50 images of cat; a good anomaly detection method should thus flag the cats to be anomalous. Similarly, for `usps`, the dataset is created by a mixture

[1] https://github.com/raghavchalapathy/rcae.
[2] http://perception.csl.illinois.edu/matrix-rank/sample_code.html.
[3] http://www3.cs.stonybrook.edu/~minhhoai/downloads.html.

Table 1. Summary of datasets used in experiments.

Dataset	# instances	# anomalies	# features
`restaurant`	200	Unknown (foreground)	19200
`usps`	231	11 ('7')	256
`cifar-10`	5000	50 (cats)	1024

of 220 images of '1's, and 11 images of '7'as in [33]. Details of the datasets are summarised in Table 1.

Additionally, we also test the ability of our model to perform denoising of images, as well as detecting inductive anomalies.

4.4 Network Parameters

Although we have observed that deeper RCAE networks tend to achieve better image reconstruction performance, there exist four fold options related to network parameters to be chosen: (a) number of convolutional filters, (b) filter size, (c) strides of convolution operation and (d) activation applied. We tuned via grid search additional hyper-parameters, including the number of hidden-layer nodes $H \in \{3, 64, 128\}$, and regularisation λ within range $[0, 100]$. The learning, drop-out rates and regularization parameter μ were sampled from a uniform distribution in the range $[0.05, 0.1]$. The embedding and initial weight matrices were all sampled from the uniform distribution within range $[-1, 1]$.

5 Experimental Results

In this section, we present experiments for three scenarios:

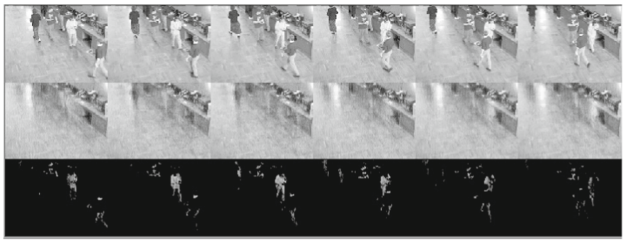
(a) non-inductive anomaly detection,
(b) inductive anomaly detection, and
(c) image denoising.

5.1 Non-inductive Anomaly Detection Results

We present results on the three datasets described in Sect. 4.

(1) `restaurant` dataset. We work with the `restaurant` video activity detection dataset [32], and consider the problem of inferring the background of videos via removal of (anomalous) foreground pixels. Estimating the background in videos is important for tasks such as anomalous activity detection. It is however difficult because of the variability of the background (e.g. due to lighting conditions) and the presence of foreground objects such as moving objects and people.

For this experiment, we only compare the RPCA and RCAE methods, owing to a lack of ground truth labels.



(a) RCAE.

(b) RPCA.

Fig. 1. Top anomalous images containing original image (people walking in the lobby) decomposed into background (lobby) and foreground (people), `restaurant` dataset.

Parameter settings. For RPCA, rank $K = 64$ is used. Per the success of the Batch Normalization architecture [18] and Exponential Linear Units [10], we have found that convolutional+batch-normalization+elu layers provide a better representation of convolutional filters. Hence, in this experiment the RCAE adopts four layers of (conv-batch-normalization-elu) in the encoder part and four layers of (conv-batch-normalization-elu) in the decoder portion of the network. RCAE network parameters such as (number of filter, filter size, strides) are chosen to be (16, 3, 1) for first and second layers and (32, 3, 1) for third and fourth layers of both encoder and decoder layers.

Results. While there are no ground truth anomalies in this dataset, a qualitative analysis reveals RCAE to outperforms its counterparts in capturing the foreground objects. Figure 1 compares the top 6 most anomalous images for RCAE and RPCA. We see that the most anomalous images contain high foreground activity (which are recognised as anomalous). Visually, we see that the background reconstruction produced by RPCA contains a few blemishes in some cases, while for RCAE the backgrounds are smooth.

(2) `usps` dataset. From the `usps` handwritten digit dataset, we create a dataset with a mixture of 220 images of '1's, and 11 images of '7', as in [33]. Intuitively, the latter images are treated as being anomalous, as the corresponding images have different characteristics to the majority of the training data. Each image is flattened as a row vector, yielding a 231×256 training matrix.

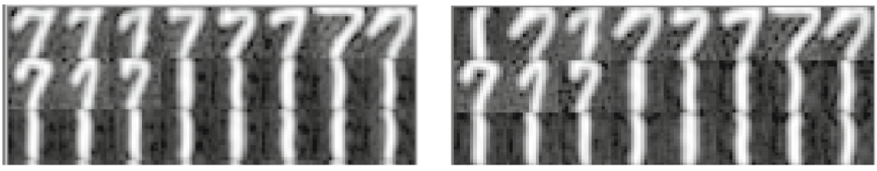
Table 2. Comparison between the baseline (bottom four rows) and state-of-the-art systems (top three rows). Results are the mean and standard error of performance metrics over 20 random training set draws. Highlighted cells indicate best performer.

(a) `usps`

Methods	AUPRC	AUROC	P@10
RCAE	0.9614 ± 0.0025	0.9988± 0.0243	0.9108 ± 0.0113
CAE	0.7003 ± 0.0105	0.9712 ± 0.0002	0.8730 ± 0.0023
AE	0.8533 ± 0.0023	0.9927 ± 0.0022	0.8108 ± 0.0003
RKPCA	0.5340 ± 0.0262	0.9717 ± 0.0024	0.5250 ± 0.0307
DRMF	0.7737 ± 0.0351	0.9928 ± 0.0027	0.7150 ± 0.0342
RPCA	0.7893 ± 0.0195	0.9942 ± 0.0012	0.7250 ± 0.0323
SVD	0.6091 ± 0.1263	0.9800 ± 0.0105	0.5600 ± 0.0249

(b) `cifar-10`

AUPRC	AUROC	P@10
0.9934 ± 0.0003	0.6255 ± 0.0055	0.8716 ± 0.0005
0.9011 ± 0.0000	0.6191 ± 0.0000	0.0000 ± 0.0000
0.9341 ± 0.0029	0.5260 ± 0.0003	0.2000 ± 0.0003
0.0557 ± 0.0037	0.5026 ± 0.0123	0.0550 ± 0.0185
0.0034 ± 0.0000	0.4847 ± 0.0000	0.0000 ± 0.0000
0.0036 ± 0.0000	0.5211 ± 0.0000	0.0000 ± 0.0000
0.0024 ± 0.0000	0.5299 ± 0.0000	0.0000 ± 0.0000



(a) RCAE. (b) RPCA.

Fig. 2. Top anomalous images, `usps` dataset.

Parameter settings. For SVD and RPCA methods, rank $K = 64$ is used. For AE, the inputs are flattened images as a column vector of size 256, and the hidden layer is a column vector of size 64 (matching the rank K).

For DRMF, we follow the settings of [33]. For RKPCA, we used a Gaussian kernel with bandwidth 0.01, a cost parameter $C = 1$, and requested 60% of the KPCA spectrum (which roughly selects 64 principal components).

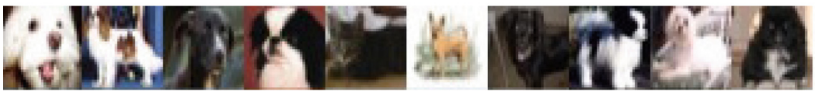
For RCAE, we set two layers of convolution layers with the filter number to be 32, filter size to be 3×3, with number of strides as 1 and rectified linear unit (ReLU) as activation with max-pooling layer of dimension 2×2.

Results. From Table 2, we see that it is a near certainty for all '7' are accurately identified as outliers. Figure 2 shows the top anomalous images for RCAE, where indeed the '7"s are correctly placed at the top of the list. By contrast, for RPCA there are also some '1"s placed at the top.

(3) `cifar-10` dataset. We create a dataset with anomalies by combining 5000 random images of dog and 50 images of cat, as illustrated in Fig. 3. In this scenario the cats are anomalies, and the goal is to detect all the cats in an unsupervised manner.

Parameter settings. For SVD and RPCA methods, rank $K = 64$ is used. We trained a three-hidden-layer autoencoder (AE) (1024-256-64-256-1024 neurons).

(a) RCAE.



(b) RPCA.

Fig. 3. Top anomalous images, `cifar-10` dataset.

The middle hidden layer size is set to be same as rank $K = 64$, and the model is trained using Adam [20]. The decoding layer uses sigmoid function in order to capture the nonlinearity characteristics from latent representations produced by the hidden layer. Finally, we obtain the feature vector for each image by obtaining the latent representation from the hidden layer.

For RKPCA, we used a Gaussian kernel with bandwidth $5 \cdot 10^{-8}$, a cost parameter $C = 0.1$, and requested 55% of the KPCA spectrum (which roughly selects 64 principal components). The RKPCA runtime was prohibitive on the full sample (see Sect. 5.4), so we resorted to a subsample of 1000 dogs and 50 cats.

The RCAE architecture in this experiment is same as for `restaurant`, containing four layers of (conv-batch-normalization-elu) in the encoder part and four layers of (conv-batch-normalization-elu) in the decoder portion of the network. RCAE network parameters such as (number of filter, filter size, strides) are chosen to be (16, 3, 1) for first and second layers and (32, 3, 1) for third and fourth layers of both encoder and decoder.

Results. From Table 2, RCAE clearly outperforms all existing state-of-the art methods in anomaly detection. Note that basic CAE, with no robustness (effectively $\lambda = \infty$), is also outperformed by our method, indicating that it is crucial to explicitly handle anomalies with the $\mathbf{N}$ term.

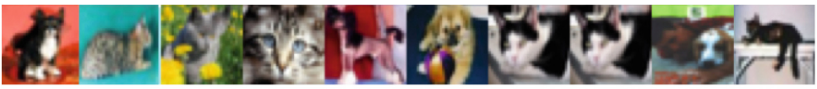
Figure 3 illustrates the most anomalous images for our RCAE method, compared to RPCA. Owing to the latter involving learning a linear subspace, the model is unable to effectively distinguish cats from dogs; by contrast, RCAE can effectively determine the manifold characterising most dogs, and identifies cats to be anomalous with respect to this.

5.2 Inductive Anomaly Detection Results

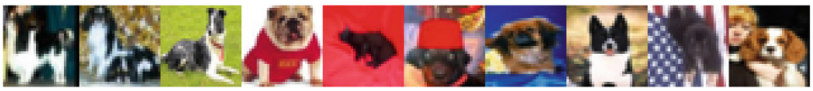
We conduct an experiment to assess the detection of *inductive* anomalies. Recall that this is a capability of our RCAE model, but not e.g. RPCA. We consider the following setup: we train our model on 5000 dog images, and then evaluate

Table 3. Inductive anomaly detection results on `cifar-10`. Note that RPCA and DRMF are inapplicable here. Highlighted cells indicate best performer.

	SVD	RKPCA	AE	CAE	RCAE
AUPRC	0.1752 ± 0.0051	0.1006 ± 0.0045	0.6200 ± 0.0005	0.6423 ± 0.0005	0.6908 ± 0.0001
AUROC	0.4997 ± 0.0066	0.4988 ± 0.0125	0.5007 ± 0.0010	0.4708 ± 0.0003	0.5576 ± 0.0005
P@10	0.2150 ± 0.0310	0.0900 ± 0.0228	0.1086 ± 0.0001	0.2908 ± 0.0001	0.5986 ± 0.0001



(a) RCAE.



(b) CAE.

Fig. 4. Top inductive anomalous images, `cifar-10` dataset.

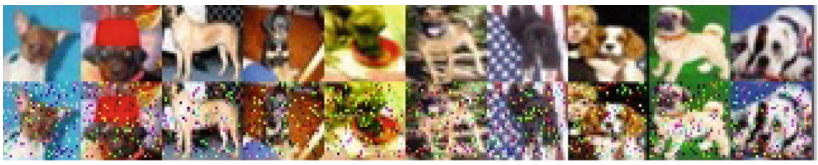
it on a test set comprising 500 dogs and 50 cat images. As before, we wish all methods to accurately determine the cats to be anomalies.

Table 3 summarises the detection performance for all the methods on this inductive task. The lower values compared to Table 2 are indicative that the problem here is more challenging than anomaly detection on a single dataset; nonetheless, we see that our RCAE method manages to convincingly outperform both the SVD and AE baselines. This is confirmed qualitatively in Fig. 4, where we see that RCAE correctly identifies many cats in the test set as anomalous, while the basic CAE method suffers.

5.3 Image Denoising Results

Finally, we test the ability of the model to de-noise images, which is a form of anomaly detection on individual pixels (or more generally, features). In this experiment, we train all models on a set of 5000 images of dog from `cifar-10`. For each image, we then add salt-and-pepper noise at a rate of 10%. Our goal is to recover the original image as accurately as possible.

Figure 5 illustrates that the most anomalous images in the presence of noise contain images of the variations of dog class images (e.g. containing person's face). Further, Fig. 6 illustrates for various methods the mean square error between the reconstructed and original images. RCAE effectively suppresses the noise as evident from the low error. The improvement over raw CAE is modest, but suggests that there is benefit to explicitly accounting for noise.



(a) RCAE.

(b) RPCA.

Fig. 5. Top anomalous images in original form (first row), noisy form (second row), image denoising task on `cifar-10`.

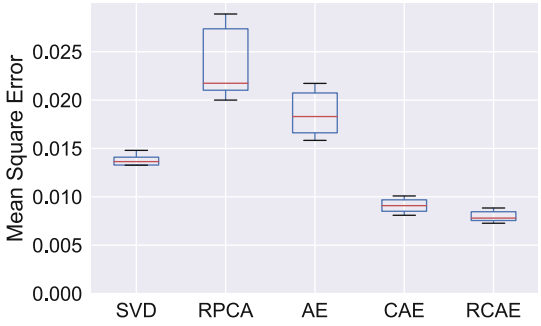


Fig. 6. Illustration of the mean square error boxplots obtained for various models on image denoising task, `cifar-10` dataset. In this setting, RCAE suppresses the noise and detects the background and foreground images effectively.

5.4 Comparison of Training Times

We remark finally that our RCAE method is comparable in training efficiency to existing methods. For example, on the small-scale `restaurant` dataset, it takes 1 min to train RPCA, and 8.5 min to train RKPCA, compared with 10 min for our RCAE method. The ability to leverage recent advances in deep learning as part of our optimisation (e.g. training models on a GPU) is we believe a salient feature of our approach.

We note that while the RKPCA method is fast to train on smaller datasets, on larger datasets it suffers from the $O(n^2)$ complexity of kernel methods; for example, it takes over an hour to train on the `cifar-10` dataset. It is plausible that one could leverage recent advances in fast approximations of kernel methods [24], and studying these would be of interest in future work. Note that the issue of using a fixed kernel function would remain, however.

6 Conclusion

We have extended the robust PCA model to the nonlinear autoencoder setting. To the best of our knowledge, ours is the first approach which is *robust*, *nonlinear* and *inductive*. The robustness ensures that the model is not over-sensitive to anomalies; the nonlinearity helps discover potentially more subtle anomalies; and being inductive makes it possible to deploy our model in a live setting.

While autoencoders are a powerful mechansim for data representation they suffer from their "black-box" nature. There is a growing body of research on outlier description, i.e., explain the reason why a data point is anomalous. A direction of future reason is to extend deep autoencoders for outlier *description*.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al.: TensorFlow: large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 (2016)
2. Aggarwal, C.C.: Outlier Analysis. Springer, New York (2016)
3. Bach, F., Jenatton, R., Mairal, J., Obozinski, G.: Convex optimization with sparsity-inducing norms. In: Optimization for Machine Learning. MIT Press (2011)
4. Bay, S.D., Schwabacher, M.: Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In: International Conference on Knowledge Discovery and Data Mining (KDD) (2003)
5. Bengio, Y., et al.: Learning deep architectures for AI. Found. Trends® Mach. Learn. **2**(1), 1–127 (2009)
6. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
7. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. In: ACM SIGMOD record, vol. 29, pp. 93–104. ACM (2000)
8. Candés, E., Li, X., Ma, Y., Wright, J.: Robust principal component analysis?: recovering low-rank matrices from sparse errors. In: 2010 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM), pp. 201–204. IEEE (2010)
9. Chandola, V., Banerjee, A., Kumar, V.: Outlier detection: a survey. ACM Comput. Surv. (2007)
10. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (ELUs). arXiv preprint arXiv:1511.07289 (2015)
11. Davis, J., Goadrich, M.: The relationship between Precision-Recall and ROC curves. In: International Conference on Machine Learning (ICML) (2006)
12. Ghoting, A., Parthasarathy, S., Otey, M., Ghoting, A., Parthasarathy, S., Otey, M.E.: Fast mining of distance-based outliers in high-dimensional datasets. Data Min. Knowl. Disc. **16**(3), 349–364 (2008)
13. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016). http://www.deeplearningbook.org
14. Hawkins, D.: Identification of Outliers. Chapman and Hall, London (1980)
15. Hotelling, H.: Analysis of a complex of statistical variables into principal components. J. Educ. Psychol. **24**, 417–441 (1933)
16. Huang, J., Zhang, T.: The benefit of group sparsity. Ann. Stat. **38**(4), 1978–2004 (2010)

17. Hull, J.J.: A database for handwritten text recognition research. IEEE Trans. Pattern Anal. Mach. Intell. **16**(5), 550–554 (1994)
18. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
19. Jain, V., Seung, S.: Natural image denoising with convolutional networks. Adv. Neural Inf. Process. Syst. **21**, 769–776 (2008)
20. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
21. Knorr, E.M., Ng, R.T.: A unified notion of outliers: properties and computation. In: KDD, pp. 219–222 (1997)
22. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Technical report (2009)
23. Lawrence, N.D., Urtasun, R.: Non-linear matrix factorization with Gaussian processes. In: International Conference on Machine Learning (ICML) (2009)
24. Lopez-Paz, D., Sra, S., Smola, A.J., Ghahramani, Z., Schölkopf, B.: Randomized nonlinear component analysis. In: International Conference on Machine Learning (ICML) (2014)
25. Nguyen, M.H., Torre, F.: Robust kernel principal component analysis. In: Advances in Neural Information Processing Systems (NIPS) (2009)
26. Recht, B., Fazel, M., Parrilo, P.A.: Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM Rev. **52**(3), 471–501 (2010)
27. Schlkopf, B., Platt, J.C., Shawe-Taylor, J.C., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural Comput. **13**(7), 1443–1471 (2001)
28. Sedhain, S., Menon, A.K., Sanner, S., Xie, L.: AutoREC: autoencoders meet collaborative filtering. In: International Conference on World Wide Web (WWW) (2015)
29. Sprechmann, P., Bronstein, A.M., Sapiro, G.: Learning efficient sparse and low rank models. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1821–1833 (2015)
30. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. JMLR **11**, 3371–3408 (2010)
31. Williams, G., Baxter, R., He, H., Hawkins, S., Gu, L.: A comparative study of RNN for outlier detection in data mining. In: International Conference on Data Mining (ICDM) (2002)
32. Xiong, L., Chen, X., Schneider, J.: Direct robust matrix factorizatoin for anomaly detection. In: International Conference on Data Mining (ICDM). IEEE (2011)
33. Xu, H., Caramanis, C., Sanghavi, S.: Robust PCA via outlier pursuit. In: Advances in Neural Information Processing Systems, pp. 2496–2504 (2010)
34. Zhai, S., Cheng, Y., Lu, W., Zhang, Z.: Deep structured energy based models for anomaly detection. In: International Conference on Machine Learning (ICML) (2016)
35. Zhao, M., Saligrama, V.: Anomaly detection with score functions based on nearest neighbor graphs. In: Advances in Neural Information Processing Systems (NIPS), pp. 2250–2258 (2009)