

# Robust, Deep and Inductive Anomaly Detection

**Raghavendra Chalapathy**

Aditya Krishna Menon

Sanjay Chawla



# Application

# Anomaly Detection: Video Surveillance.

## ■ Detecting: **Background activities.**



# Definition

# Anomaly Detection

- Anomalies are objects : **different from most other objects.**



# Anomaly Detection: By Spectral Techniques

- Analysis based on **Eigen-Decomposition** of data.

# Anomaly Detection: By Spectral Techniques

- Analysis based on **Eigen-Decomposition** of data.
- **Key Idea:**
  - Find combination of attributes **capturing bulk of variability**.

# Anomaly Detection: By Spectral Techniques

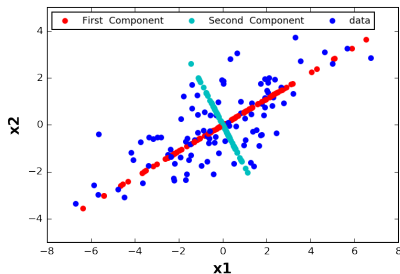
- Analysis based on **Eigen-Decomposition** of data.
- **Key Idea:**
  - Find combination of attributes **capturing bulk of variability**.
  - Reduced set of attributes can **explain only normal data well**.
- Several methods use Principal Component Analysis.
  - **Top few principal components** capture variability: normal data.
  - **Outliers have variability** in the smallest component.



# Motivation and Challenges

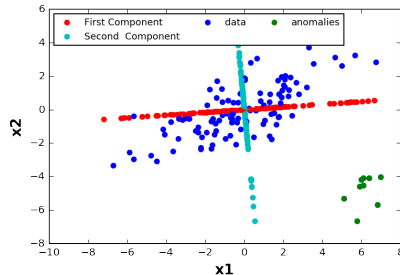
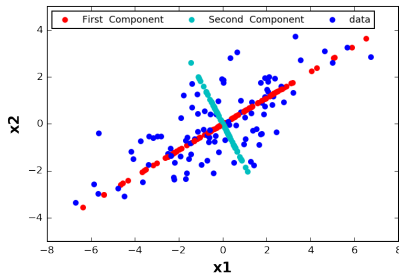
# Anomaly Detection: PCA

- PCA is highly sensitive to **data perturbation**.



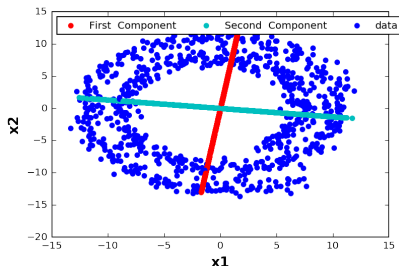
# Anomaly Detection: PCA

- PCA is highly sensitive to **data perturbation**.



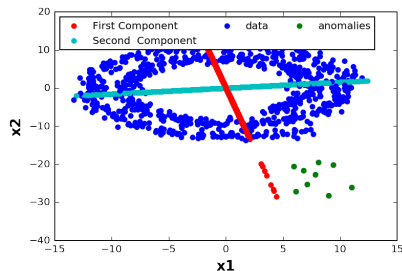
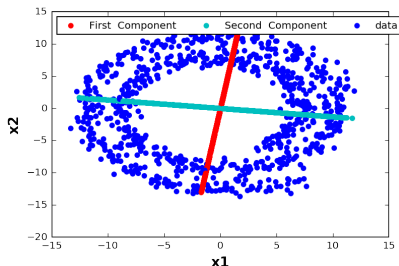
# Anomaly Detection: PCA

- PCA, Robust PCA **fails to capture non-linear projections.**



# Anomaly Detection: PCA

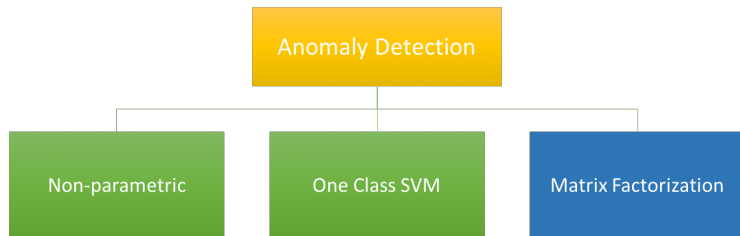
- PCA, Robust PCA fails to capture non-linear projections.



- We propose **Robust (Convolutional) Auto-encoder** to overcome these limitations.

# Related Work

# Conventional Anomaly Detection Techniques



# Matrix Factorization Approach: PCA



# Matrix Factorization Approach: PCA

- PCA interpreted as **matrix factorisation**.
- Factorise  $\mathbf{X} \in \mathbb{R}^{N \times D}$  into  $\mathbf{Z} \in \mathbb{R}^{N \times K}$  and  $\mathbf{W} \in \mathbb{R}^{K \times D}$

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

# Matrix Factorization Approach: PCA

- PCA interpreted as **matrix factorisation**.
- Factorise  $\mathbf{X} \in \mathbb{R}^{N \times D}$  into  $\mathbf{Z} \in \mathbb{R}^{N \times K}$  and  $\mathbf{W} \in \mathbb{R}^{K \times D}$

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

## Limitations:

- Does not handle data perturbations.
- Does not capture nonlinear projections.

# Robust PCA

- RPCA generalizes PCA with **tuning parameter**  $\lambda > 0$ .

# Robust PCA

- RPCA generalizes PCA with **tuning parameter**  $\lambda > 0$ .

$$\min_{\mathbf{S}, \mathbf{N}} \|\mathbf{S}\|_* + \lambda \|\mathbf{N}\|_1 : \mathbf{X} = \mathbf{S} + \mathbf{N} \quad (1)$$

# Robust PCA

- RPCA generalizes PCA with **tuning parameter**  $\lambda > 0$ .

$$\min_{\mathbf{S}, \mathbf{N}} \|\mathbf{S}\|_* + \lambda \|\mathbf{N}\|_1 : \mathbf{X} = \mathbf{S} + \mathbf{N} \quad (1)$$

- $\mathbf{S}$  is signal,  $\mathbf{N}$  is noise matrix.

# Robust PCA

- RPCA generalizes PCA with **tuning parameter**  $\lambda > 0$ .

$$\min_{\mathbf{S}, \mathbf{N}} \|\mathbf{S}\|_* + \lambda \|\mathbf{N}\|_1 : \mathbf{X} = \mathbf{S} + \mathbf{N} \quad (1)$$

- $\mathbf{S}$  is signal,  $\mathbf{N}$  is noise matrix.

# Robust PCA

- RPCA generalizes PCA with **tuning parameter**  $\lambda > 0$ .

$$\min_{\mathbf{S}, \mathbf{N}} \|\mathbf{S}\|_* + \lambda \|\mathbf{N}\|_1 : \mathbf{X} = \mathbf{S} + \mathbf{N} \quad (1)$$

- **S** is signal, **N** is noise matrix.
- Points with **high value of N** are considered anomalous.

# Robust PCA

- RPCA generalizes PCA with **tuning parameter**  $\lambda > 0$ .

$$\min_{\mathbf{S}, \mathbf{N}} \|\mathbf{S}\|_* + \lambda \|\mathbf{N}\|_1 : \mathbf{X} = \mathbf{S} + \mathbf{N} \quad (1)$$

- **S** is signal, **N** is noise matrix.
- Points with **high value of N** are considered anomalous.



# Robust PCA

- RPCA generalizes PCA with **tuning parameter**  $\lambda > 0$ .

$$\min_{\mathbf{S}, \mathbf{N}} \|\mathbf{S}\|_* + \lambda \|\mathbf{N}\|_1 : \mathbf{X} = \mathbf{S} + \mathbf{N} \quad (1)$$

- **S** is signal, **N** is noise matrix.
- Points with **high value of N** are considered anomalous.

## Limitations:

- Does not capture **nonlinear projections**.

# Robust (Convolutional) Autoencoders

# Auto-encoders for anomaly detection.

- Auto encoder with single **hidden layer**

$$\min_{\mathbf{U}, \mathbf{V}} \|\mathbf{X} - \mathbf{f}(\mathbf{XU})\mathbf{V}\|_{\mathbf{F}}^2 \quad (2)$$

- $\hat{\mathbf{X}} = \mathbf{f}(\mathbf{XU})\mathbf{V}$  is **reconstruction error measure**.

# Auto-encoders for anomaly detection.

- Auto encoder with single **hidden layer**

$$\min_{\mathbf{U}, \mathbf{V}} \|\mathbf{X} - \mathbf{f}(\mathbf{XU})\mathbf{V}\|_{\mathbf{F}}^2 \quad (2)$$

- $\hat{\mathbf{X}} = \mathbf{f}(\mathbf{XU})\mathbf{V}$  is **reconstruction error measure**.
  - $\mathbf{U}$ : Weights (input  $\rightarrow$  hidden),

# Auto-encoders for anomaly detection.

- Auto encoder with single **hidden layer**

$$\min_{\mathbf{U}, \mathbf{V}} \|\mathbf{X} - \mathbf{f}(\mathbf{XU})\mathbf{V}\|_{\mathbf{F}}^2 \quad (2)$$

- $\hat{\mathbf{X}} = \mathbf{f}(\mathbf{XU})\mathbf{V}$  is **reconstruction error measure**.

- $\mathbf{U}$ : Weights (input  $\rightarrow$  hidden),
- $\mathbf{V}$ : Weights (hidden  $\rightarrow$  output),

- $\mathbf{XU}$  projects  $\mathbf{X}$  into  $K$  dimensional space  
 $\mathbf{U} \in \mathbf{R}^{D \times K}$ ,  $\mathbf{V} \in \mathbf{R}^{K \times D}$ .

# Auto-encoders for anomaly detection.

- Auto encoder with single **hidden layer**

$$\min_{\mathbf{U}, \mathbf{V}} \|\mathbf{X} - \mathbf{f}(\mathbf{X}\mathbf{U})\mathbf{V}\|_{\mathbf{F}}^2 \quad (2)$$

- $\hat{\mathbf{X}} = \mathbf{f}(\mathbf{X}\mathbf{U})\mathbf{V}$  is **reconstruction error measure**.

- $\mathbf{U}$ : Weights (input  $\rightarrow$  hidden),
- $\mathbf{V}$ : Weights (hidden  $\rightarrow$  output),
- activation function:  $f: \mathbb{R} \rightarrow \mathbb{R}$

- $\mathbf{X}\mathbf{U}$  projects  $\mathbf{X}$  into  $K$  dimensional space

$$\mathbf{U} \in \mathbb{R}^{D \times K}, \mathbf{V} \in \mathbb{R}^{K \times D}.$$

- Non linear projection: **sigmoid**  $f(\cdot) := (1 + \exp(-a))^{-1}$

# Comparison: Conventional Anomaly Detection Methods

- Deep (convolution) Robust Auto encoder **are versatile**.

	Handles Data Perturbation	Captures Non-linear Structure	Learns Non-linear Structure from data
PCA	No	No	No
RPCA	Yes	No	No
RKPCA	Yes	Yes	No
<b>RCAE</b>	Yes	Yes	Yes

# Robust (convolution) Auto-Encoders [RCAE]

- For activation function  $f: \mathbb{R} \rightarrow \mathbb{R}$

$$\min_{\mathbf{U}, \mathbf{V}, \mathbf{N}} \|\mathbf{X} - \mathbf{f}(\mathbf{XU})\mathbf{V} + \mathbf{N}\|_{\mathbf{F}}^2 + \frac{\mu}{2} \cdot (\|\mathbf{U}\|_{\mathbf{F}}^2 + \|\mathbf{V}\|_{\mathbf{F}}^2) + \lambda \|\mathbf{N}\|_1 \quad (3)$$



# Robust (convolution) Auto-Encoders [RCAE]

- For activation function  $f: \mathbb{R} \rightarrow \mathbb{R}$

$$\min_{\mathbf{U}, \mathbf{V}, \mathbf{N}} \|\mathbf{X} - \mathbf{f}(\mathbf{XU})\mathbf{V} + \mathbf{N}\|_{\mathbf{F}}^2 + \frac{\mu}{2} \cdot (\|\mathbf{U}\|_{\mathbf{F}}^2 + \|\mathbf{V}\|_{\mathbf{F}}^2) + \lambda \|\mathbf{N}\|_1 \quad (3)$$

- $\hat{X} = f(\mathbf{XU})\mathbf{V}$  is **reconstruction error measure**

# Robust (convolution) Auto-Encoders [RCAE]

- For activation function  $f: \mathbb{R} \rightarrow \mathbb{R}$

$$\min_{\mathbf{U}, \mathbf{V}, \mathbf{N}} \|\mathbf{X} - \mathbf{f}(\mathbf{XU})\mathbf{V} + \mathbf{N}\|_{\mathbf{F}}^2 + \frac{\mu}{2} \cdot (\|\mathbf{U}\|_{\mathbf{F}}^2 + \|\mathbf{V}\|_{\mathbf{F}}^2) + \lambda \|\mathbf{N}\|_1 \quad (3)$$

- $\hat{X} = \mathbf{f}(\mathbf{XU})\mathbf{V}$  is **reconstruction error measure**
- $\lambda, \mu > 0$  are tuning parameters.

# Robust (convolution) Auto-Encoders [RCAE]

- For activation function  $f: \mathbb{R} \rightarrow \mathbb{R}$

$$\min_{\mathbf{U}, \mathbf{V}, \mathbf{N}} \|\mathbf{X} - \mathbf{f}(\mathbf{XU})\mathbf{V} + \mathbf{N}\|_{\mathbf{F}}^2 + \frac{\mu}{2} \cdot (\|\mathbf{U}\|_{\mathbf{F}}^2 + \|\mathbf{V}\|_{\mathbf{F}}^2) + \lambda \|\mathbf{N}\|_1 \quad (3)$$

- $\hat{X} = \mathbf{f}(\mathbf{XU})\mathbf{V}$  is **reconstruction error measure**
- $\lambda, \mu > 0$  are tuning parameters.
- $\mathbf{Z} = \mathbf{f}(\mathbf{XU})$  **latent representation decoded by  $\mathbf{V}$  weights.**
- $\mathbf{N}$  captures **gross outliers.**

# Robust (convolution) Auto-Encoders [RCAE]

- For activation function  $f: \mathbb{R} \rightarrow \mathbb{R}$

$$\min_{\mathbf{U}, \mathbf{V}, \mathbf{N}} \|\mathbf{X} - \mathbf{f}(\mathbf{X}\mathbf{U})\mathbf{V} + \mathbf{N}\|_{\mathbf{F}}^2 + \frac{\mu}{2} \cdot (\|\mathbf{U}\|_{\mathbf{F}}^2 + \|\mathbf{V}\|_{\mathbf{F}}^2) + \lambda \|\mathbf{N}\|_1 \quad (3)$$

- $\hat{X} = \mathbf{f}(\mathbf{X}\mathbf{U})\mathbf{V}$  is **reconstruction error measure**
- $\lambda, \mu > 0$  are tuning parameters.
- $\mathbf{Z} = \mathbf{f}(\mathbf{X}\mathbf{U})$  **latent representation decoded by  $\mathbf{V}$  weights.**
- $\mathbf{N}$  captures **gross outliers.**
- $0 < \lambda < +\infty$ , models a standard auto encoder robust to noise.

# RCAE Vs Robust PCA (1)

- **RPCA objective** function with basic equality constraints:

$$\min_{S, N} \|S\|_* + \lambda \|N\|_1 \quad (4)$$

# RCAE Vs Robust PCA (1)

- **RPCA objective** function with basic equality constraints:

$$\min_{S,N} \|S\|_* + \lambda \|N\|_1 \quad (4)$$

- **RCAE objective** function with basic equality constraints:

$$\min_{U,V,N} \|X - f(XU)V + N\|_F^2 + \frac{\mu}{2} \cdot (\|U\|_F^2 + \|V\|_F^2) + \lambda \|N\|_1 \quad (5)$$

# RCAE Vs Robust PCA (1)

- **RPCA objective** function with basic equality constraints:

$$\min_{S,N} \|S\|_* + \lambda \|N\|_1 \quad (4)$$

- **RCAE** objective function with basic equality constraints:

$$\min_{U,V,N} \|X - f(XU)V + N\|_F^2 + \frac{\mu}{2} \cdot (\|U\|_F^2 + \|V\|_F^2) + \lambda \|N\|_1 \quad (5)$$

- Conceptual Similarity between **Equation 4** and **Equation 5** established [5].

## Training RCAE (1)

- Consider the Objective function of **Robust Autoencoder**.

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



# Training RCAE (1)

- Consider the Objective function of **Robust Autoencoder**.

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

- More generally objective could be rewritten as below.

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

# Training RCAE (1)

- Consider the Objective function of **Robust Autoencoder**.

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

- More generally objective could be rewritten as below.

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

- where  $\hat{\mathbf{X}}(\theta)$  is some generic predictor with parameters  $\theta$ .
- $\Omega(\cdot)$  : regularization function.

# Training RCAE (1)

- Consider the Objective function of **Robust Autoencoder**.

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

- More generally objective could be rewritten as below.

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

- where  $\hat{\mathbf{X}}(\theta)$  is some generic predictor with parameters  $\theta$ .
- $\Omega(\cdot)$  : regularization function.
- Equation 7 is **non-convex but unconstrained and sub-differentiable**

## Training RCAE (2)

- For differentiable function  $\hat{\mathbf{X}}(\theta)$  back-propagation is employed.

---

<sup>1</sup>Bach, F., Jenatton, R., Mairal, J., Obozinski, G. Convex Optimization with Sparsity-Inducing Norms.

## Training RCAE (2)

- For differentiable function  $\hat{\mathbf{X}}(\theta)$  back-propagation is employed.
- We follow **soft thresholding approach** to optimize  $\mathbf{N}$ .

---

<sup>1</sup>Bach, F., Jenatton, R., Mairal, J., Obozinski, G. Convex Optimization with Sparsity-Inducing Norms.

## Training RCAE (2)

- For differentiable function  $\hat{\mathbf{X}}(\theta)$  back-propagation is employed.
- We follow **soft thresholding approach** to optimize  $\mathbf{N}$ .
- For fixed  $\mathbf{N}, \theta, \mathbf{U}, \mathbf{V}$  the objective is:

$$\min_{\theta, \mathbf{N}} \|\mathbf{N} - (\mathbf{X} - \hat{\mathbf{X}}(\theta))\|_1 + \lambda \|\mathbf{N}\|_1 \quad (8)$$

---

<sup>1</sup>Bach, F., Jenatton, R., Mairal, J., Obozinski, G. Convex Optimization with Sparsity-Inducing Norms.

## Training RCAE (2)

- For differentiable function  $\hat{\mathbf{X}}(\theta)$  back-propagation is employed.
- We follow **soft thresholding approach** to optimize  $\mathbf{N}$ .
- For fixed  $\mathbf{N}, \theta, \mathbf{U}, \mathbf{V}$  the objective is:

$$\min_{\theta, \mathbf{N}} \|\mathbf{N} - (\mathbf{X} - \hat{\mathbf{X}}(\theta))\|_1 + \lambda \|\mathbf{N}\|_1 \quad (8)$$

- Applying **Soft-thresholding**<sup>1</sup> to compute  $\mathbf{N}$

$$N_{ij} = \begin{cases} (\mathbf{X} - \hat{\mathbf{X}}(\theta))_{ij} - \frac{\lambda}{2} & \text{if } (\mathbf{X} - \hat{\mathbf{X}}(\theta))_{ij} > \frac{\lambda}{2} \\ (\mathbf{X} - \hat{\mathbf{X}}(\theta))_{ij} + \frac{\lambda}{2} & \text{if } (\mathbf{X} - \hat{\mathbf{X}}(\theta))_{ij} < -\frac{\lambda}{2} \\ 0 & \text{else.} \end{cases} \quad (9)$$

---

<sup>1</sup>Bach, F., Jenatton, R., Mairal, J., Obozinski, G. Convex Optimization with Sparsity-Inducing Norms.

# Experimental Setup



# Summary of Datasets

- We compare all methods on three real-world datasets for anomaly detection:

# Summary of Datasets

- We compare all methods on three real-world datasets for anomaly detection:
  - **Restaurant**, comprising video background modeling and activity detection consisting of snapshots of restaurant activities.

# Summary of Datasets

- We compare all methods on three real-world datasets for anomaly detection:
  - **Restaurant**, comprising video background modeling and activity detection consisting of snapshots of restaurant activities.
  - **USPS**, comprising the USPS handwritten digits.

# Summary of Datasets

- We compare all methods on three real-world datasets for anomaly detection:
  - **Restaurant**, comprising video background modeling and activity detection consisting of snapshots of restaurant activities.
  - **USPS**, comprising the USPS handwritten digits.
  - **CIFAR-10** consisting of 60000  $32 \times 32$  colour images of 10 classes, with 6000 images per class.

# Summary of Datasets

- We compare all methods on three real-world datasets for anomaly detection:
  - **Restaurant**, comprising video background modeling and activity detection consisting of snapshots of restaurant activities.
  - **USPS**, comprising the USPS handwritten digits.
  - **CIFAR-10** consisting of 60000  $32 \times 32$  colour images of 10 classes, with 6000 images per class.

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

# Anomaly Detection: Methods Compared

- Compare empirical effectiveness of:

---

<sup>2</sup>Publicly available implementations[3][5][1]

<sup>3</sup>Tensorflow Implementation: <https://github.com/raghavchalapathy/rcae>

# Anomaly Detection: Methods Compared

- Compare empirical effectiveness of:
  - **Truncated SVD**: zero-mean features is equivalent to PCA.<sup>2</sup>

---

<sup>2</sup>Publicly available implementations[3][5][1]

<sup>3</sup>Tensorflow Implementation: <https://github.com/raghavchalapathy/rcae>

# Anomaly Detection: Methods Compared

- Compare empirical effectiveness of:
  - **Truncated SVD**: zero-mean features is equivalent to PCA.<sup>2</sup>
  - **Robust PCA** .<sup>2</sup>

---

<sup>2</sup>Publicly available implementations[3][5][1]

<sup>3</sup>Tensorflow Implementation: <https://github.com/raghavchalapathy/rcae>



# Anomaly Detection: Methods Compared

- Compare empirical effectiveness of:
  - **Truncated SVD**: zero-mean features is equivalent to PCA.<sup>2</sup>
  - **Robust PCA** .<sup>2</sup>
  - **Robust kernel PCA (RKPCA)**<sup>2</sup>.

---

<sup>2</sup>Publicly available implementations[3][5][1]

<sup>3</sup>Tensorflow Implementation: <https://github.com/raghavchalapathy/rcae>

# Anomaly Detection: Methods Compared

- Compare empirical effectiveness of:
  - **Truncated SVD**: zero-mean features is equivalent to PCA.<sup>2</sup>
  - **Robust PCA**.<sup>2</sup>
  - **Robust kernel PCA (RKPCA)**<sup>2</sup>.
  - **Autoencoder (AE)**<sup>3</sup>.

---

<sup>2</sup>Publicly available implementations[3][5][1]

<sup>3</sup>Tensorflow Implementation: <https://github.com/raghavchalapathy/rcae>

# Anomaly Detection: Methods Compared

- Compare empirical effectiveness of:
  - **Truncated SVD**: zero-mean features is equivalent to PCA.<sup>2</sup>
  - **Robust PCA**.<sup>2</sup>
  - **Robust kernel PCA (RKPCA)**<sup>2</sup>.
  - **Autoencoder (AE)**<sup>3</sup>.
  - **Convolutional Autoencoder (CAE)**<sup>3</sup> where  $\lambda = +\infty$ .

---

<sup>2</sup>Publicly available implementations[3][5][1]

<sup>3</sup>Tensorflow Implementation: <https://github.com/raghavchalapathy/rcae>

# Anomaly Detection: Methods Compared

- Compare empirical effectiveness of:
  - **Truncated SVD**: zero-mean features is equivalent to PCA.<sup>2</sup>
  - **Robust PCA**.<sup>2</sup>
  - **Robust kernel PCA (RKPCA)**.<sup>2</sup>
  - **Autoencoder (AE)**.<sup>3</sup>
  - **Convolutional Autoencoder (CAE)**<sup>3</sup> where  $\lambda = +\infty$ .
  - **Robust Convolutional Autoencoder (RCAE)**.<sup>3</sup>

---

<sup>2</sup>Publicly available implementations[3][5][1]

<sup>3</sup>Tensorflow Implementation: <https://github.com/raghavchalapathy/rcae>

# Experiment Settings

- Experiments were conducted **for three scenarios:**

# Experiment Settings

- Experiments were conducted **for three scenarios:**
  - **Non-inductive anomaly detection.**

# Experiment Settings

- Experiments were conducted **for three scenarios:**
  - **Non-inductive anomaly detection.**
  - **Inductive anomaly detection.**

# Experiment Settings

- Experiments were conducted **for three scenarios:**
  - **Non-inductive anomaly detection.**
  - **Inductive anomaly detection.**
  - **Image denoising.**



# Experiment Settings

- Experiments were conducted **for three scenarios:**
  - **Non-inductive anomaly detection.**
  - **Inductive anomaly detection.**
  - **Image denoising.**
- **Four network parameters** were tuned for best performance:

# Experiment Settings

- Experiments were conducted **for three scenarios:**
  - **Non-inductive anomaly detection.**
  - **Inductive anomaly detection.**
  - **Image denoising.**
- **Four network parameters** were tuned for best performance:
  - number of convolutional filters.
  - filter size.
  - strides of convolution operation.
  - activation applied.

# Experiment Settings

- Experiments were conducted **for three scenarios**:
  - **Non-inductive anomaly detection.**
  - **Inductive anomaly detection.**
  - **Image denoising.**
- **Four network parameters** were tuned for best performance:
  - number of convolutional filters.
  - filter size.
  - strides of convolution operation.
  - activation applied.
- Number of **hidden nodes**  $H \in [3, 64, 128]$ , regularisation parameters  $\lambda \in [0, 100]$  and  $\mu \in [0.05, 0.1]$ .

# Experiment Settings

- Experiments were conducted **for three scenarios**:
  - **Non-inductive anomaly detection.**
  - **Inductive anomaly detection.**
  - **Image denoising.**
- **Four network parameters** were tuned for best performance:
  - number of convolutional filters.
  - filter size.
  - strides of convolution operation.
  - activation applied.
- Number of **hidden nodes**  $H \in [3, 64, 128]$ , regularisation parameters  $\lambda \in [0, 100]$  and  $\mu \in [0.05, 0.1]$ .
- Initial **weight matrices** were from uniform distribution in  $[-1, 1]$ .

# Evaluation Methodology

- Predictive performance is measured against the ground truth anomaly labels **using three standard metrics:**

# Evaluation Methodology

- Predictive performance is measured against the ground truth anomaly labels **using three standard metrics:**
  - **Area under the precision-recall curve (AUPRC)**
  - **Area under the ROC curve (AUROC)**
  - **Precision at 10 (P@10)**

# Evaluation Methodology

- Predictive performance is measured against the ground truth anomaly labels **using three standard metrics:**
  - **Area under the precision-recall curve (AUPRC)**
  - **Area under the ROC curve (AUROC)**
  - **Precision at 10 (P@10)**
- **AUPRC and AUROC** measure ranking performance.

# Evaluation Methodology

- Predictive performance is measured against the ground truth anomaly labels **using three standard metrics:**
  - **Area under the precision-recall curve (AUPRC)**
  - **Area under the ROC curve (AUROC)**
  - **Precision at 10 (P@10)**
- **AUPRC and AUROC** measure ranking performance.
- **P@10** measures classification performance.(actual anomalies among top-10 scored instances).



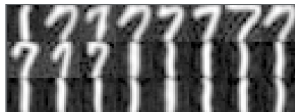
# Results

# Non Inductive: Top anomalous Images Detected

- USPS : 220 images of '1's, and 11 images of '7' (anomalous)



(a) RCAE



(b) RPCA

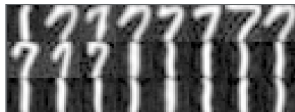
- CIFAR: 5000 images of 'dog's, and 50 images of 'cat's (anomalous)

# Non Inductive: Top anomalous Images Detected

- USPS : 220 images of '1's, and 11 images of '7'(anomalous)



(a) RCAE



(b) RPCA

- CIFAR: 5000 images of 'dog's, and 50 images of 'cat's(anomalous)



(a) RCAE



(b) RPCA

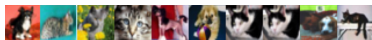
# Non Inductive Anomaly Detection: Performance

- The **Robust convolution autoencoder** outperforms the state of the art methods.

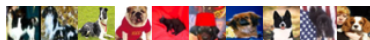
(a) USPS				(b) cifar-10		
Methods	AUPRC	AUROC	P@10	AUPRC	AUROC	P@10
RCAE	0.9614 $\pm$ 0.0025	0.9988 $\pm$ 0.0243	0.9108 $\pm$ 0.0113	0.9934 $\pm$ 0.0003	0.6255 $\pm$ 0.0055	0.8716 $\pm$ 0.0005
CAE	0.7003 $\pm$ 0.0105	0.9712 $\pm$ 0.0002	0.8730 $\pm$ 0.0023	0.9011 $\pm$ 0.0000	0.6191 $\pm$ 0.0000	0.0000 $\pm$ 0.0000
AE	0.8533 $\pm$ 0.0023	0.9927 $\pm$ 0.0022	0.8108 $\pm$ 0.0003	0.9341 $\pm$ 0.0029	0.5260 $\pm$ 0.0003	0.2000 $\pm$ 0.0003
RKPCA	0.5340 $\pm$ 0.0262	0.9717 $\pm$ 0.0024	0.5250 $\pm$ 0.0307	0.0557 $\pm$ 0.0037	0.5026 $\pm$ 0.0123	0.0550 $\pm$ 0.0185
DRMF	0.7737 $\pm$ 0.0351	0.9928 $\pm$ 0.0027	0.7150 $\pm$ 0.0342	0.0034 $\pm$ 0.0000	0.4847 $\pm$ 0.0000	0.0000 $\pm$ 0.0000
RPCA	0.7893 $\pm$ 0.0195	0.9942 $\pm$ 0.0012	0.7250 $\pm$ 0.0323	0.0036 $\pm$ 0.0000	0.5211 $\pm$ 0.0000	0.0000 $\pm$ 0.0000
SVD	0.6091 $\pm$ 0.1263	0.9800 $\pm$ 0.0105	0.5600 $\pm$ 0.0249	0.0024 $\pm$ 0.0000	0.5299 $\pm$ 0.0000	0.0000 $\pm$ 0.0000

# Inductive: Top anomalous Images Detected

- First train model on **only normal** 5000 dog images.
- Evaluate it on a test set 500 dogs and 50 'cat's(anomalous).



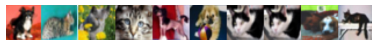
(a) RCAE



(b) CAE

# Inductive: Top anomalous Images Detected

- First train model on **only normal** 5000 dog images.
- Evaluate it on a test set 500 dogs and 50 'cat's(anomalous).



(a) RCAE



(b) CAE

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

# Image De-noising Capability: RCAE vs RPCA

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



(a) RCAE



(b) RPCA

# Image De-noising Capability: RCAE vs RPCA

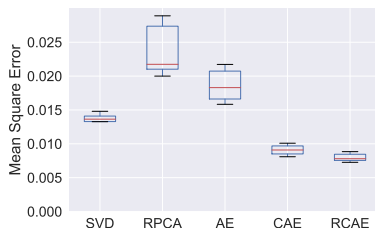
- Top anomalous images in original form (first row), noisy form (second row), image denoising task on cifar-10.



(a) RCAE



(b) RPCA





# Conclusion

# Conclusion

- Extended robust PCA model **to the nonlinear autoencoder setting.**

# Conclusion

- Extended robust PCA model **to the nonlinear autoencoder setting**.
- Our approach is **robust, deep and inductive**.

# Conclusion

- Extended robust PCA model **to the nonlinear autoencoder setting.**
- Our approach is **robust, deep and inductive.**
- Not oversensitive besides captures **subtle anomalies.**

# Conclusion

- Extended robust PCA model **to the nonlinear autoencoder setting.**
- Our approach is **robust, deep and inductive.**
- Not oversensitive besides captures **subtle anomalies.**
- Extend deep autoencoders for **outlier description.**

# References

- [1] Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289 (2015)
- [2] Goodfellow, I., Bengio, Y., Courville, A. Deep Learning. MIT Press (2016), <http://www.deeplearningbook.org>
- [3] Xiong, L., Chen, X., Schneider, J. Direct robust matrix factorization for anomaly detection. In International Conference on Data Mining (ICDM). IEEE (2011)
- [4] 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)
- [5] Chalapathy, Raghavendra, Aditya Krishna Menon, and Sanjay Chawla. Robust, Deep and Inductive Anomaly Detection. arXiv:1704.06743 (2017).
- [6] Candes, E., Li, X., Ma, Y., Wright, J.: Robust principal component analysis: Recovering low-rank matrices from sparse errors. In: Sensor Array and Multichannel Signal Processing Workshop (SAM), 2010 IEEE. pp. 201204. IEEE (2010)

