Moving Beyond Traditional Anomaly Detection

Dr Ye Zhu
Senior Lecturer
Deakin University, Australia
ye.zhu@ieee.org

Prof. Gang Li
Professor
Deakin University, Australia
ligang@ieee.org

Yang Cao
PhD Student
Deakin University, Australia
charles.cao@ieee.org

Dr Chen Li
Research Fellow
Nagoya University, Japan
lichen7283@gmail.com

25 May 2023, Osaka, Japan
Dr. Guansong Pang  
Assistant Professor  
Singapore Management University, Singapore  
gspang@smu.edu.sg

Yue Zhao  
PhD student  
Carnegie Mellon University, USA  
zhaoy@cmu.edu

Xin Han  
Research Assistant  
Macau University, China  
xinhan@um.edu.mo

Dr. Sutharshan Rajasegarar  
Senior Lecturer  
Deakin University, Australia  
sutharshan.rajasegarar@deakin.edu.au

Zong-you Liu  
Master student  
Nanjing University, China  
liuzy@lamda.nju.edu.cn

Prof. Ting Kai Ming  
Professor  
Nanjing University, China  
tingkm@nju.edu.cn
Tutorial outline

Overview of challenges and methods
- Problem definition and applications
- Overview of anomaly detection approaches
- Shallow vs deep methods

Part 1
60 min

Shallow anomaly detection models
- Distance/Density/Histogram/PCA-based models
- Isolation-based models
- Code demonstration

Part 2
60 min

Deep anomaly detection models
- The modeling and supervision information
- Anomaly explanation in deep detectors
- Code demonstration

Part 3
60 min

Future opportunities
Practical advices

30 min


**Code for demonstration:**
https://github.com/zhuye88/TAD

https://github.com/yzhao062/pyod
https://sites.google.com/site/gspangsite/sourcecode
https://github.com/IsolationKernel/Codes
Part 1: Overview of Challenges and Methods

- Problem definition and applications
- Challenges
- Overview of anomaly detection approaches
- Deep vs. shallow methods
What are Anomalies?

- Anomalies (a.k.a., outliers, novelties): Points that are significantly different from most of the data
  - Rare
  - Irregular

Binary Output versus scoring
- Binary output generates a yes/no tag
- Preferable and more general: Scoring output generates a real-valued score or rank

Multiple ways to define what makes an anomaly different
Types of Anomalies?

- A **point anomaly** is a single anomalous point.

- A **group anomaly** can be a cluster of anomalies or some series of related points that are anomalous under the joint series distribution.

- A **contextual point anomaly** occurs if a point deviates in its local context, here a spike in an otherwise normal time series.

- A **low-level sensory anomaly** deviates from the low-level features

- A **semantic anomaly** deviates in high-level factors of variation or semantic concepts
Real-World Application Domains

Cybersecurity: attacks, malware, malicious apps/URLs, biometric spoofing

Social Network and Web Security: false/malicious accounts, false/hate/toxic information

Finance: credit card/insurance frauds, market manipulation, money laundering, etc.

Healthcare: lesions, tumours, events in IoT/ICU monitoring, etc.

Video Surveillance: criminal activities, road accidents, violence, etc.

Industrial Inspection: Defects, micro-cracks

Image source: UCF-Crime data, MVTec AD data, etc.
Scientific Application Domains

**Drug Discovery:**
rare active substances

**High-Energy Physics:**
Higgs boson particles

**Astronomy:**
Anomalous events

**Rover-Based Space Exploration:**
unknown textures

**Material Science:**
exceptional molecule graphs
Application-Specific Complexities

Four key complexities

Heterogeneity
• Different anomalies may exhibit completely different expressions, e.g., accidents, robbery vs. explosion events

Application-specific methodologies
• Different methodologies required by different application-specific definitions, e.g., credit card frauds (point anomalies) vs malicious accounts in social media (group anomalies)

Unknown Nature (unsupervised setting)
• Anomalies remain unknown until they actually occur

Coverage
• Difficult to collect data covering all classes of anomalies
Key Challenges

Challenge #1: Low Anomaly Detection Accuracy
- Rareness and heterogeneity of anomalies in a dataset
- Many returned anomalies are noise or uninteresting instances

Challenge #2: Contextual and High-Dimensional Data
- Anomalies are visible only in context of implicit relations in temporal, spatial and graph data
- Increased dimensionality also makes anomaly detection difficult

Challenge #3: Sample-Efficient Learning
- Building generalized detection models with a limited amount of labeled anomaly data
Key Challenges

**Challenge #4: Noise-Resilient Anomaly Detection**
- Data may contain normal and anomalous instances with no labels (anomaly contamination)
- Data may contain weak supervision information:
  > Coarse anomaly labels such as leveraging video-level labels to detect anomalous frames

**Challenge #5: Complex Anomalies**
- Conditional/group anomalies
- Multi-modal anomalies

**Challenge #6: Anomaly Explanation**
- Obtaining cues about why a specific instance is detected anomalies by specific methods
- Balancing interpretability and detection accuracy
Overview of Anomaly Detection Approaches

Traditional (Shallow) Methods and Disadvantages

**Statistical/probabilistic-based approaches**
- Statistical test-based, depth-based, deviation-based

**Proximity-based approach**
- Distance-based, density-based, clustering-based

**Shallow ML Models**
- Construct an unsupervised (one-class) analog of a supervised ML model such as the SVM
- Use unsupervised dimensionality reduction methods, PCA, kernel PCA

**Others**
- Information-theoretic, subspace methods

**Weaknesses**
- Weak capability of capturing intricate relationships
- Lots of hand-crafting of algorithms and features [ad hoc]
- Ad hoc nature makes it difficult to incorporate supervision seamlessly
Advantages of Deep Learning

Integrates feature learning and anomaly scoring

• Generates **newly learned feature space** → A uninformative and primitive feature representations [e.g., image pixels]
• End-to-end learning → Can simultaneously learn features and relevant anomaly scores [no hand-crafting of features]
• Strong feature learning → Captures intricate relations [e.g., mid-level image features]
• Diverse neural architectures → Tailor to complex domains [e.g., RNN for time-series]
• Unified detection and explanation of anomalies → Better anomaly explanation guaranteed by integration of detection and localization
• Anomaly-informed models with improved accuracy → Naturally integrates with labeled data (easy to navigate spectrum of supervised and unsupervised models)
Deep vs Shallow [Traditional]: Example

Deep Method - Autoencoder

Shallow Method – iForest
# Deep vs. Shallow [Representation]

<table>
<thead>
<tr>
<th>Feature space</th>
<th>Deep methods</th>
<th>Shallow methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low Level Features</td>
<td>Expressive new space</td>
<td>Primitive space</td>
</tr>
<tr>
<td>Mid Level Features</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High Level Features</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Input---Shallow Layers------Middle Layers------Deeper Layers------> Output

MIT: Alexander Amini, 2018 introtodeeplearning.com
## Deep vs. Shallow: [Algorithm Type]

<table>
<thead>
<tr>
<th></th>
<th>Deep methods</th>
<th>Shallow methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature space</td>
<td>Expressive new space</td>
<td>Primitive space</td>
</tr>
<tr>
<td>Anomaly detection alg.</td>
<td>Defined by NN structure</td>
<td>Heuristic or ad hoc</td>
</tr>
</tbody>
</table>

![Diagram showing the comparison between Deep and Shallow methods](image-url)
# Deep vs. Shallow [Feature Relations]

<table>
<thead>
<tr>
<th></th>
<th>Deep methods</th>
<th>Shallow methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature space</td>
<td>Expressive new space</td>
<td>Primitive space</td>
</tr>
<tr>
<td>Anomaly detection algo.</td>
<td>Defined by NN structure</td>
<td>Heuristic or ad hoc</td>
</tr>
<tr>
<td>Feature relations captured</td>
<td>Intricate</td>
<td>Simple</td>
</tr>
</tbody>
</table>

![Image showing feature levels](image_url)
# Deep vs. Shallow [Feature Learning Methods for Diverse Data Types]

<table>
<thead>
<tr>
<th></th>
<th>Deep methods</th>
<th>Shallow methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature space</td>
<td>Expressive new space</td>
<td>Primitive space</td>
</tr>
<tr>
<td>Anomaly detection algo.</td>
<td>Defined by NN structure</td>
<td>Heuristic or ad hoc</td>
</tr>
<tr>
<td>Feature relations captured</td>
<td>Intricate</td>
<td>Simple</td>
</tr>
<tr>
<td>Extracting features in diverse types of data</td>
<td>Varying on architectures and loss functions [e.g., RNN, CNN]</td>
<td>Hand-crafted feature extractors/off-the-shelf methods</td>
</tr>
</tbody>
</table>

MLP, CNN, RNN, GNN, etc. vs. random projection, PCA, subgraph patterns, optical flow, etc.
Deep vs. Shallow Methods [Explanation]

<table>
<thead>
<tr>
<th>Feature Space</th>
<th>Deep methods</th>
<th>Shallow methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expression new space</td>
<td>Primitive space</td>
<td></td>
</tr>
<tr>
<td>Anomaly detection algo.</td>
<td>Defined by NN structure</td>
<td>Heuristic or ad hoc</td>
</tr>
<tr>
<td>Feature relations captured</td>
<td>Intricate</td>
<td>Simple</td>
</tr>
<tr>
<td>Extracting features in diverse types of data</td>
<td>Varying on architectures and loss functions [e.g., RNN, CNN]</td>
<td>Hand-crafted feature extractors/off-the-shelf methods</td>
</tr>
<tr>
<td>Unified anomaly detection and explanation</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
Part 2: Shallow anomaly detection models

- Distance/Density-based methods
- Histogram-based method
- Principal Component Analysis
- Isolation-based methods
Distance-based method

Nearest Neighbour (kNN) approach
• For each data point \( d \) compute the distance to the \( k \)-th nearest neighbour \( d_k \)
• Sort all data points according to the distance \( d_k \)
• Outliers are points that have the largest distance \( d_k \)
• and therefore, are located in the sparser neighbourhoods
• Usually, data points that have distance \( d_k \) higher than a threshold are identified as outliers
• Not suitable for datasets that have modes with varying density

Density-based method

Local Outlier Factor (LOF)
• Compute the average of the ratios of the density of each point and the density of its nearest neighbors
• Outliers are points with largest ratio value neighbourhoods
• Able to detect local anomalies

Many variants have been proposed to improve efficiency, accuracy and robustness of LOF, such as CBLOF, LDCOF and LDOF.

Histogram-based method

Assume each feature is independent; estimate the histograms separately and combine

Advantages: simple to use; easy to be distributed; suited for large-scale problem

Disadvantages: cannot capture complex feature dependency, while it works well in general

Principal Component Analysis (PCA)

• Calculating eigenvectors using all samples, where outliers are far from the eigenvectors. This distance can be used as the outlier score.

• **Advantages:** easy to understand; moderate running time

• **Disadvantages:** as a linear model, it could not model complex results.
The Isolation Forest ‘isolates’ observations (subsample) by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature.

\[
Score(x) = \frac{1}{t} \sum_{i=1}^{t} \ell_i(x) \quad \text{where } \ell_i(x) \text{ is the path length of test point } x \text{ traversed in tree } i.
\]

Isolation Forest (cont.)

Source: Liu et al. 2008
Isolating partitions

- Large in sparse regions and small in dense regions
- Adapt to local data distribution
- This characteristic is important not only for point anomaly detection, but also for deriving data dependent kernels (to be described later).

Isolation mechanism comparison

iNNE: each region is a hypersphere defined with a center represented by an instance from the subsample, and its boundary is defined by the distance to the nearest neighbor (NN) of the instance at the center.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>iForest</th>
<th>iNNE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition shape</td>
<td>hyper-rectangles</td>
<td>hyper-spheres</td>
</tr>
<tr>
<td>Anomaly score</td>
<td>average measure over $t$ path lengths</td>
<td>average measure over $t$ radiuses of hyper-sphere</td>
</tr>
<tr>
<td>Parameters</td>
<td>$\Psi$ – number of partitioning cells $t$ – number of sets of partitionings</td>
<td></td>
</tr>
</tbody>
</table>

$$I(z) = 1 - \frac{\tau(\eta_x)}{\tau(0)}$$

Source: Tharindu et al 2018

iForest versus iNNE

Source: Tharindu et al 2018
Example handwritten digits: MNIST
top 2 anomalies per digit

iForest

iNNE
Example handwritten digits: MNIST
bottom 2 anomalies (most typical example) per digit

iForest

iNNE
Isolation-based methods are beyond point anomaly detection

Since the idea of Isolation was conceived, it was never confine to point anomaly detection only.

Two notable recent developments:

• **Isolation Kernel (IK):** A data dependent kernel which has a unique characteristic: two points, as measured by IK derived with a dataset in a sparse region, are more similar than the same two points, as measured by IK derived with a dataset in a dense region.

• **Isolation Distributional Kernel (IDK)** measures the similarity of two distributions, based on the framework of kernel mean embedding.
We can use a nearest neighbour method to split a data space into 8 non-overlapping partitions, and independently conduct this partitioning strategy for $t=100$ trials. If two points $x$ and $y$ are located in the same partition (sharing the same nearest subsample point) in 25 out of 100 trials, then the similarity between $x$ and $y$ is estimated as 0.25, i.e., $K_8 (x, y|D) = 0.25$. 

---

36
**Isolation Kernel Feature Map**

$\Phi(x)$ is a binary vector that represents the partitions in all the partitionings, where $x$ falls in to only one of $\psi$ cells in each partitioning.

\[\Phi(x) \rightarrow [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \quad 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \cdots \quad 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0\]

\[\Phi(y) \rightarrow [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \quad 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \cdots \quad 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0\]

\[K_\psi(x, y|D) = \frac{1}{t} \langle \Phi(x), \Phi(y) \rangle >\]
Point-Set Kernel

Given a point $x$ and a set $A = \{y_i\}_{i=1}^p$, and $x, y_i \in R^d$, the point-set similarity between $x$ and $A$ is the average pairwise similarity between $x$ and every point in $A$, defined as follows:

$$\hat{K}_\psi(x, A|D) = \frac{1}{|A|} \sum_{y \in A} K_\psi(x, y|D) = \frac{1}{t} \langle \Phi(x), \hat{\Phi}(A) \rangle$$

Where $\hat{\Phi}(A) = \frac{1}{|A|} \sum_y \Phi(y)$ is the kernel mean map of $K_\psi$. 
Because $\hat{\Phi}(A)$ can be pre-calculated, estimating the similarity between a point and a set points costs constant time $O(1)$.

$$\hat{K}_\psi(x, A|D) = \frac{\langle \hat{\Phi}(x), \hat{\Phi}(A) \rangle}{\sqrt{\langle \hat{\Phi}(x), \hat{\Phi}(x) \rangle} \sqrt{\langle \hat{\Phi}(A), \hat{\Phi}(A) \rangle}}$$

$$\hat{\Phi}(A) = \frac{1}{|A|} \sum_{y \in A} \Phi(y)$$
Isolation Distributional Kernel (IDK)

\[ \hat{K}(P_S, P_T) = \frac{1}{|S||T|} \sum_{x \in S} \sum_{y \in T} \kappa(x, y) \]

1. As \( \kappa \) (Isolation Kernel) is a characteristic kernel, then its kernel mean map is injective, i.e.,
\[ \| \hat{\phi}(P_S) - \hat{\phi}(P_T) \|_H = 0 \text{ if and only if } P_S = P_T. \]

2. Data dependent property: Two distributions, as measured by IDK derived in sparse region, are more similar than the same two distributions, as measured by IDK derived in dense region.
   - Key in improving task-specific performance

3. It has finite-dimensional feature map: \( \hat{K}(P_S, P_T) = \langle \hat{\Phi}(P_S), \hat{\Phi}(P_T) \rangle \)
   - Key in low time complexity
IDK: Group Anomaly Detection

IDK$^2$ : Using two levels of IDK to detect group anomalies [KDD20, TKDE22]
Level-1 maps each group to a point in Level-1 Hilbert space
Level-2 maps level-1 pts and the set of level-1 pts to pts in Level-2 Hilbert space

IDK: Time Series Anomaly Detection

A new treatment for timeseries. This is a paradigm shift from the time domain and frequency domain approaches that have been around for more than 100 years.

IDK is the best for periodic time series because it is more effective in detecting anomalous subsequences that are shortened/lengthened. It also runs orders of magnitude faster because it needs no additional process apart from the feature mapping, e.g., it only costs 661 CPU seconds on 1 million data length.

Figure 1: Example sine waves (with m = 1000) and their pdfs

Kai Ming Ting, Zhongyou Liu, Hang Zhang, Ye Zhu (2022) A New Distributional Treatment for Time Series and an anomaly detection investigation. To appear in VLDB22
Part 3: Deep anomaly detection models

- The modeling perspective
- The supervision information perspective
- Anomaly explanation in deep detectors
Three Principal Categories

- **Simplest approaches**: Anomaly detection-specific feature learning
  - End-to-end optimization of pipeline with score learning
  - Reconstruction/Prediction/Anomaly Measure-driven Loss Function

- **Most methods belong to this category**, e.g., autoencoder-, GANs-, one-class models

- **Often more effective than the other two approaches**

---

Categorization Based on Supervision

Unsupervised approach
• Working on anomaly-contaminated unlabeled data; no manually labeled training data
• Limited work done

Semi-supervised approach
• Assuming the availability of a set of manually labeled normal training data
• Most of current deep methods belong to this approach

Weakly-supervised approach
• Assuming we have some labels for anomaly classes, yet the class labels are **partial** (i.e., they do not span the entire set of anomaly class), **inexact** (i.e., coarse-grained labels), or **inaccurate** (i.e., some given labels can be incorrect)
• Limited work done
Main approach I: Deep learning for feature extraction

Leveraging existing deep models to extract low-dimensional features for downstream anomaly measures

- The feature extraction and the anomaly scoring are fully disjointed
- Assumption: the extracted features preserve the discriminative information that helps separate anomalies from normal instances

General framework

1. Given dataset $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ with $x_i \in \mathbb{R}^D$, the approach is formulated as
   $$ z = \phi(x; \Theta) $$
   where $\phi: \mathcal{X} \rightarrow \mathcal{Z}$ is a deep-neural-network-based feature mapping, with $\mathcal{Z} \in \mathbb{R}^K (K \ll D)$

2. An anomaly measure, i.e., $f$ that has no connection to $\phi$, is then applied onto the new space to calculate anomaly scores

Two directions: pre-trained models vs directly training deep feature extractors on the target data
Main approach II – Learning feature representations of normality

To integrate feature learning with anomaly scoring in some ways, rather than fully decoupling them as in Approach I

- Generic normality feature learning

\[
\{\Theta^*, W^*\} = \arg \min_{\Theta, W} \sum_{x \in X} \ell \left( \psi \left( \phi(x; \Theta); W \right) \right),
\]

\[s_x = f(x, \phi_{\Theta^*}, \psi_{W^*}),\]

(\(\psi\) is a surrogate feature learning function, \(\ell\) is a loss function)

e.g., autoencoder methods

✓ \(\phi\) – encoder, \(\psi\) – decoder, \(f\) – a reconstruction error-based anomaly score
Autoencoders

To learn some low-dimensional feature representation space on which the given data instances can be well reconstructed

- **Assumption**: Normal instances can be better reconstructed from compressed feature space than anomalies

**General Framework**

1. Bottleneck architecture + reconstruction loss
2. The larger reconstruction errors the more abnormal

\[
\begin{align*}
    z &= \phi_e(x; \Theta_e), \quad \hat{x} = \phi_d(z; \Theta_d), \\
    \{\Theta_e^*, \Theta_d^*\} &= \arg \min_{\Theta_e, \Theta_d} \sum_{x \in \mathcal{X}} \|x - \phi_d(\phi_e(x; \Theta_e); \Theta_d)\|^2, \\
    s_x &= \|x - \phi_d(\phi_e(x; \Theta_e^*); \Theta_d^*)\|^2,
\end{align*}
\]

Image source: Towards Data Science
Generative Adversarial Networks (GANs)

To adversarially learn a latent space that captures the normality underlying the given data

- **Assumption**: Normal data instances can be better generated than anomalies from the latent feature space of the generative network in GANs

**General framework**

1. Train a GAN-based model
2. Calculate anomaly scores by looking into the difference between an input instance and its counterpart generated from the latent space of the generator
Predictability modeling

Learn representations by using temporally adjacent instances as the context to predict the current/future instances

- **Assumption**: Normal instances are **temporally more predictable** than anomalies

**General framework**

1. Train a current/future instance prediction network
2. Calculate the difference between the predicted instance and the actual instance as anomaly score.
Self-supervised classification

Learn representations of normality by self-supervised classification with different data augmentation operations

• **Assumption**: Normal instances are more consistent to self-supervised classifiers than anomalies

**General framework**

1. Apply different augmentation operations to the data
2. Instances that are augmented with the same operation are treated as from the class, such as flipping, cropping, erasing
3. Learn a multi-class classification model using these synthetic class labels
4. Calculate the inconsistency of the instance to the model as anomaly score

Distance-based measure

Learning representations tailored for distance-based measures

• Assumption: Anomalies are distributed far from their closest neighbors while normal instances are located in dense neighborhoods

The general framework
1. Devise a feature mapping function $\phi$ that maps original data onto a new representation space
2. Optimize the feature representations such that anomalies have larger distance to some reference instances than normal instances
3. Anomaly scoring using the desired distance measure in the new space
One-class classification measure

Learning representations tailored for one-class classification

- **Assumption**: All normal instances come from a single (abstract) class and can be summarized by a compact model, to which anomalies do not conform

The general framework

1. Devise a feature mapping function $\phi$ that maps original data onto a new representation space
2. Optimize the feature representations using one-class classification loss
3. Anomaly scoring using the one-class classification model in the new space
Cluster-based measure

Learning representations so that anomalies are clearly deviated from the clusters in the newly learned representation space

• **Assumption**: Normal instances have stronger adherence to clusters than anomalies

**The general framework**

1. Devise a feature mapping function \( \phi \) that maps original data onto a new representation space
2. Optimize the feature representations using clustering-based loss
3. Anomaly scoring using a cluster-based anomaly measure in the new space
Main approach III – End-to-end anomaly score learning

Directly learn anomaly scores in an end-to-end fashion

- Has a neural network that directly learns scalar anomaly scores
- (surrogate) Loss functions for anomaly ranking/classification
- Generally requiring supervision of (synthetic or real) anomaly data
- Not dependent on existing anomaly measures

Formally, the general formulation is as follows

$$\Theta^* = \arg \min_{\Theta} \sum_{x \in \mathcal{X}} \ell (\tau (x; \Theta)),$$

$$s_x = \tau (x; \Theta^*).$$

where $\tau: \mathcal{X} \rightarrow \mathbb{R}$ is an end-to-end anomaly scoring network
Ranking models

Learn a ranking model that is associated with the absolute/relative ordering relation of the instance abnormality

Assumption: There exists an observable ordinal variable that captures some data abnormality

The general framework

1. Define the (synthetic) ordinal variable
2. Use the variable to define a surrogate loss functions for anomaly ranking and train the detection model
3. Given a test instance, the model directly gives its anomaly score
Prior-driven models

Impose a prior over the anomaly scores to drive the anomaly score learning

• Assumption: The imposed prior captures the underlying (ab)normality of the dataset

The general framework

1. Impose a prior over the weight parameters of a neural network-based anomaly scoring measure, or over the expected anomaly scores
2. Optimize the anomaly ranking/classification with the prior
3. Given a test instance, the model directly gives its anomaly score
End-to-end one-class classification

Train a one-class classifier that discriminates whether a given instance is normal or synthetic outliers in an end-to-end fashion

- **Assumptions**: (i) Data instances that are approximated to anomalies can be effectively synthesized. (ii) All normal instances can be summarized by a discriminative one-class model.

The general framework

- **Generate artificial outliers**
- **Train a GAN to discriminate whether a given instance is normal or an artificial outlier**

Softmax likelihood models

Learn anomaly scores by maximizing the likelihood of events in the training data

- **Assumption**: Anomalies and normal instances are respectively low- and high-probability events
- It is primarily designed for categorical data. Different types of interactions can be incorporated.

The general framework

1. The probability of an event is modeled using a softmax function

\[ p(x; \Theta) = \frac{\exp \left( \tau(x; \Theta) \right)}{\sum_{x \in \mathcal{X}} \exp \left( \tau(x; \Theta) \right)} \]

\( \tau \) is an anomaly scoring function

2. The parameters are then learned by a maximum likelihood function

\[ \Theta^* = \arg \max_{\Theta} \sum_{x \in \mathcal{X}} \log p(x; \Theta) \]

3. Given a test instance, the model directly gives its anomaly score by the event probability
Anomaly explanation

To provide tangible explanation of why specific data points are considered as anomalies

Detector-independent outlying aspect mining

Unified anomaly detection and explanation
Unified anomaly detection and explanation in deep detectors

Deep methods

Data reconstruction

Feature-wise reconstruction errors for anomaly explanation. Larger errors indicate more outlying aspects of the anomalies.

Back-propagation

Uses the back-propagation of gradient/activation values to obtain the contribution of features to anomaly scores.
Data reconstruction

$\ell_2$-distance autoencoders vs. SSIM autoencoders

Brighter colors indicate larger dissimilarity between input and reconstruction

Back-propagation approach

This approach uses the back-propagation of gradient/activation values to obtain the contribution of features to anomaly scores

- Gradient back-propagation, such as \textbf{Grad-CAM}, is arguably the most popular method used

\[
\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A^k_{ij}} \\
\text{global average pooling} \\
globally pooled gradients via backprop
\]

\[
L_{\text{Grad-CAM}}^c = \text{ReLU} \left( \sum_k \alpha_k^c A^k \right) \\
\text{linear combination}
\]

Prediction scores (e.g., anomaly scores)

Feature maps

Guided Grad-CAM – Examples

Normality attention learning

Convolutional adversarial variational autoencoder with guided attention (CAVGA)

- Latent representations $z$ preserve normal patterns
- Using attention map derived from Grad-CAM to supervise and localize as much normal regions as possible:

Training:

$$L_{final} = w_r L + w_{adv} L_{adv} + w_{ae} L_{ae}$$

Convolutional VAE: $$L = L_R(x, \hat{x}) + KL(q_\theta(z|x)||p_\theta(z|x))$$

GANs: $$L_{adv} = -\frac{1}{N} \sum_{i=1}^{N} \log(D(x_i)) + \log(1 - D(\hat{x}_i))$$

Attention expansion: $$L_{ae,1} = \frac{1}{|A|} \sum_{i,j} \left(1 - A_{i,j}\right)$$

where $A$ is the attention map gained by using the convolutional representations $z^*$ to back-propagate gradients as in Grad-CAM

• After applying the attention expansion, the model is enforced to attend to the entire images to look for any possible normal patterns

Part 4: Future opportunities and practical advices
Direction #1 – Exploring anomaly-supervisory signals

**Unsupervised**
- Data reconstruction, generator-discriminator, pseudo class labels, etc.

**Self-supervised**
- Self-supervised classification, future prediction, etc.

**Anomaly measure-driven**
- Presuming some distribution of normal/anomalous data, e.g., one-class, cluster, distance, etc.

Are there other more effective sources of supervisory signals?

**Domain-driven anomaly detection?**
- Application-specific knowledge of anomaly
- Expert rules, etc.
Direction #2 – Deep weakly-supervised anomaly detection

**Few-shot anomaly detection or data-efficient anomaly detection**
- Leveraging a few anomaly examples to perform anomaly-informed detection
- Data efficiency?
- Overfitting?

**Unknown anomaly detection**
- To generalize from the limited labeled anomalies to novel classes of anomaly

**Learning detection models with coarse-grained anomaly labels**
- How to effectively leverage such label information
Direction #3 – Large-scale normality learning

Large-scale unsupervised/self-supervised representation learning specifically designed for anomaly detection

• Any anomaly contamination in the large-scale data?

• Knowledge transferable across different domains?

• How about different types of datasets or anomalies?
Direction #4 – Deep detection of complex anomalies

Deep models for conditional/group anomalies

• Capturing complex temporal/spatial dependence
• Learning representations of a set of unordered data points

Multimodal anomaly detection

• Excellent capability in learning feature representations from different types of raw data
• Flexible feature representation fusion
Direction #5 – Interpretable and actionable deep anomaly detection

Interpretable deep anomaly detection
- Intrinsically interpretable deep detection models?

Actionable deep anomaly detection
- Quantifying the impact of detected anomalies and mitigation actions
Direction #6 – Novel applications and settings

Out-of-distribution (OOD) detection
• Accurate classification while being able to detect any data instances that are drawn far away from the given training distribution

Curiosity learning
• Curiosity-driven exploration: Encouraging reinforcement learning agents to explore novel states

Non-i.i.d. anomaly detection

Detection of adversarial examples

Anti-spoofing in biometric systems

Anomaly detection in scientific data

Safety in autonomous systems

Montezuma’s Revenge
Practical Advices

No free lunch theorem
No single anomaly detector can always outperform. Even we know the *best* anomaly detection algorithm for a task, we need to set the hyperparameters for it. Consequently, we need to select for both:
• the detection algorithm and
• its corresponding hyperparameters (default is insufficient)

It is often necessary to try many algorithms

Characteristics of Ideal Anomaly Detectors

**Few parameters**
- parameter-free the best
- Easy to tune; not too sensitive to parameter setting

**Fast runtime (Scalability)**
Can scale up to large datasets and high dimensional datasets

**Known behaviours under different data properties (Interpretability)**
Can explain the prediction results matters in many applications

**Can deal with different types of anomalies**
Factors influencing a performance assessment

The nature of the anomaly detection problem
The data properties of benchmark datasets
The characteristics of algorithms
number of parameters, sensitivity to parameter setting, ensemble or not; how it performs under different conditions

Evaluation methodology
Best performance, test accuracy, AUC
A good measure in assessing the “goodness” of the ranking outcome
Recent Advance of Model Selection

Ensemble learning
Combining more than one ML models, leading to potentially better results and more robust models at higher cost. Some common operations include averaging, maximization, and more. Some of them are introduced in a later page.

Model selection
Only pick a model but it is challenging under the unsupervised setting since we could not do any model evaluation.
• Based on internal model evaluation
• Selecting more reliable and stable algorithms

Automating Outlier Detection via Meta-Learning
MetaOD is trained on extensive OD benchmark datasets to capitalize the prior experience so that it could select the potentially best performing model for unseen datasets.

Further tips

Utilising labels/domain knowledge

• If there are available labels, use (semi-)supervised models first

Further tips (cont.)

Starting from rule-based and scalable method
- Try to combine the rule-based models and ML models. Keep the rule-based models at least use as baselines.
- Try to use rule-based models to explain ML results; try to use ML results to discover new anomalous patterns. If possible, analyse on which samples they agree & disagree.
- If your data is extremely large with many features, then use neural networks.
- If your data can be viewed as either tabular data and/or time-series/graph, try tabular first.

Selecting faster tools/packages
If you have GPUs, consider using TOD other than PyOD— the former is 10x faster.
Thank you!

Q & A