MONASH University

Information Technology

Efficient Anomaly Detection by Isolation using Nearest Neighbour Ensemble

Tharindu Rukshan Bandaragoda

Kai Ming Ting

David Albrecht

Fei Tony Liu

Jonathan R. Wells

Outline

- Overview of anomaly detection
- Existing methods
- Motivation
- iNNE
- Empirical evaluation

Anomaly Detection

- Properties of anomalies
  - Not conforming to the norm in a dataset
  - Rare and different from others
- Applications:
  - Intrusion detection in computer networks
  - Credit card fraud detection
  - Disturbance detection in natural systems (e.g., hurricane)
- Challenges
  - Datasets becoming larger : need efficient methods
  - Datasets increasing in dimensions : need methods effective in high-dimensional scenarios

Existing methods

- Clustering based methods
  - Instances that do not belong to any cluster are anomalies
  - Some measures used:
    - Membership of a cluster (Ester et al., 1996)
    - Distance from closest cluster centroid
    - Ratio between distance to cluster centroid and cluster size (He et al., 2003)
  - Issues
    - Computationally expensive: $O(n^2)$ or higher
    - Do not provide a score to determine the granularity of an anomaly (strong or weak anomaly)

Existing methods

- Distance/density based methods
  - Instances having far neighbours are anomalies
  - Some measures used :
    - k^{th}-nearest neighbour distance (Ramaswamy et al., 2000)
    - Average distance of *k*-nearest neighbours (Angiulli et al., 2002)
    - Number of instances inside an *r* radius hypersphere (Ren et al., 2004)
  - Issues
    - Nearest neighbour search is expensive
      - $O(n^2)$ time complexity
    - Insensitive to locality and thus fail to detect local anomalies

Existing methods

- Relative density based methods
  - Instances having lower density than its neighbourhood are anomalies
  - Measure the ratio between density of a data point and average density of its neighbourhood
  - *k*-nearest neighbour distance (Breunig et al., 2000) or number of instances in *r*-radius neighbourhood (Papadimitriou et al., 2003) are used as proxies to density.
  - Issues
    - Nearest neighbour search is expensive
      - $O(n^2)$ time complexity

Existing methods

- Isolation based methods
  - Attempt to isolate anomalies from others
  - Exploit anomalous properties of being *few and different*

  - iForest (Liu et al., 2008)
    - Partition feature space using axis-parallel subdivisions
    - Instances isolated earlier are anomalies
    - Build an ensemble of binary trees from randomly selected samples
    - **Extremely efficient** : O(ntψ) where t is ensemble size and ψ is subsample size
    - Effective in detection global anomalies of low dimensional datasets

Motivation

- iForest is a highly efficient method
  - Can scale up to very large datasets
- It fails in some scenarios such as:
  - Local anomaly detection
  - Anomaly detection in noisy datasets
  - Axis parallel masking

- Hypothesis : weaknesses of iForest occurs due to its isolation mechanism
- Solution : use a better isolation mechanism to overcome the weaknesses

iNNE

- iNNE : **i**solation using **N**earest **N**eighbour **E**nsembles

- Features:
  - Overcome the identified weaknesses of iForest
  - Retain the efficiency of iForest and scale up to very large datasets
  - Perform competitively with existing methods

Intuition

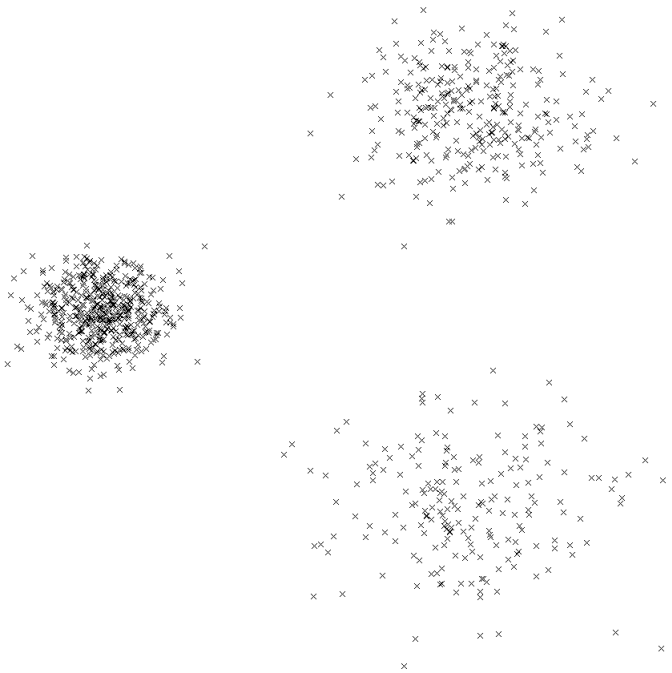
- Anomalies are expected to be far from its Nearest Neighbours
- Isolation can be performed by creating a region around an instance to isolate it from other instances
  - Large regions in sparse areas
  - Small regions in dense areas
- Radius of the region is a measure of **isolation**
- Radius of the region relative to neighbouring region is a measure of **relative-isolation**
- Points that fall into regions with a high relative-isolation are anomalies

Local Regions

– Sample $\mathcal{S}$ is selected randomly from the given dataset

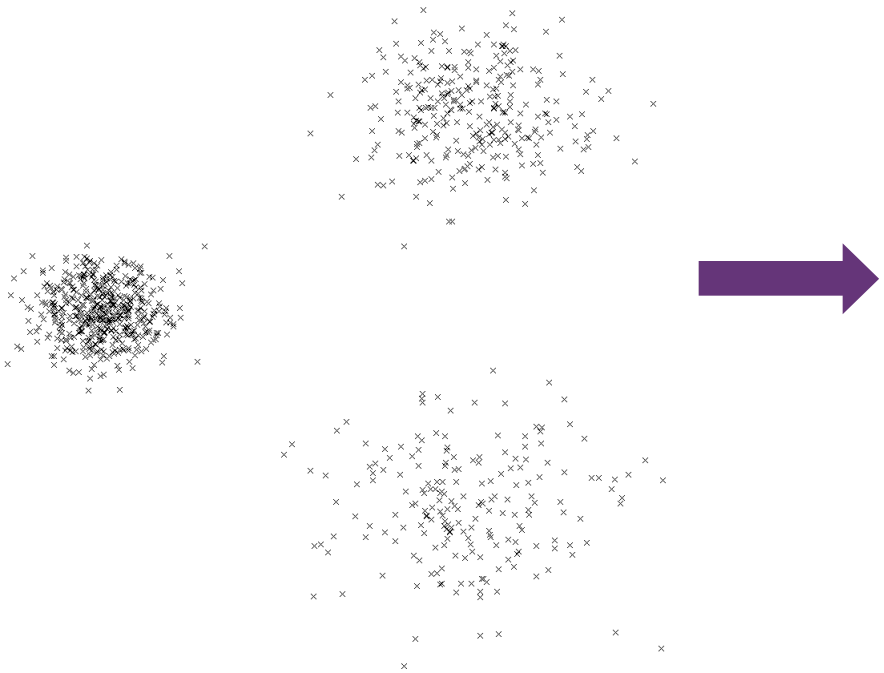
Local Regions

– Sample $\mathcal{S}$ is selected randomly from the given dataset



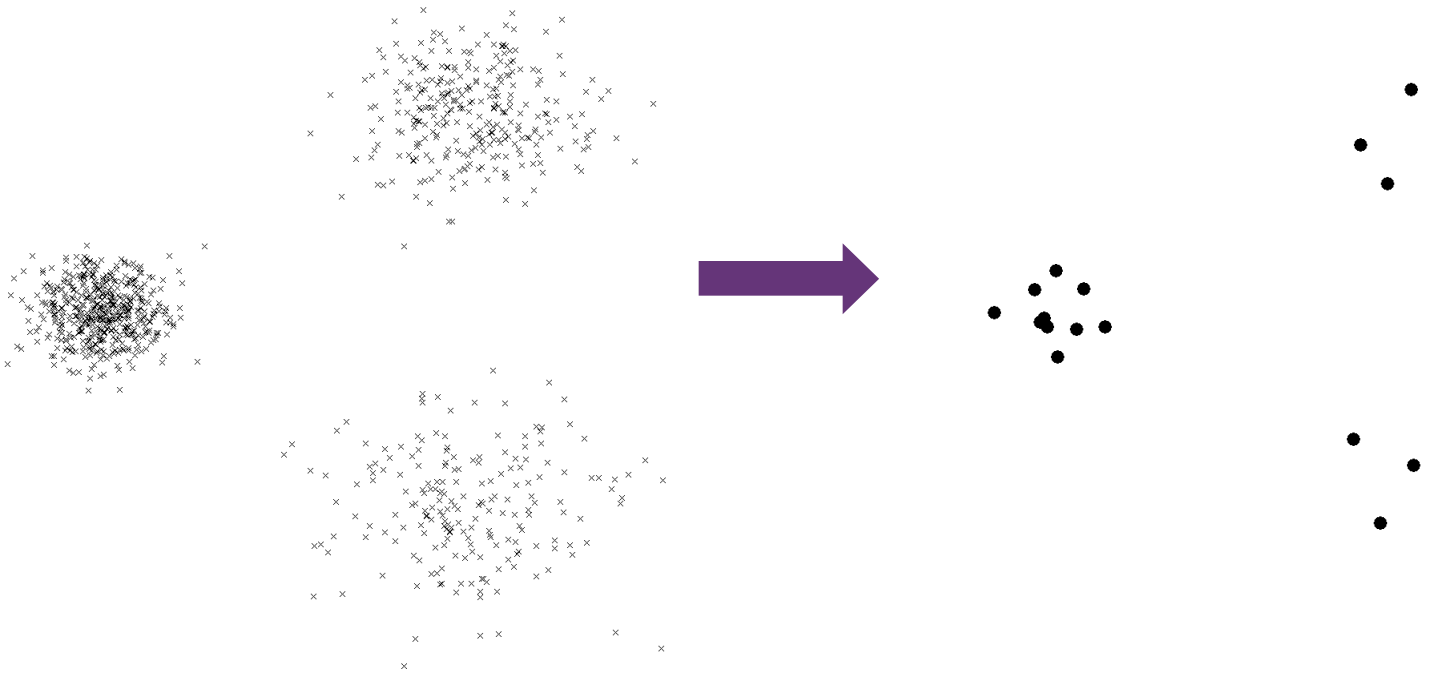
Local Regions

– Sample $\mathcal{S}$ is selected randomly from the given dataset



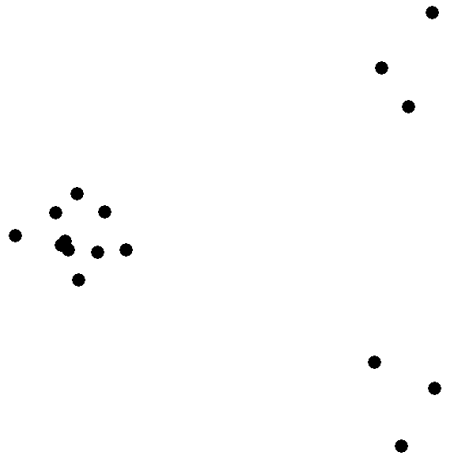
Local Regions

– Sample $\mathcal{S}$ is selected randomly from the given dataset



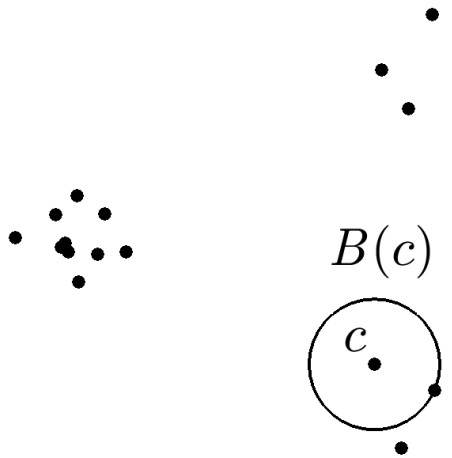
Local Regions

– Sample $\mathcal{S}$ is selected randomly from the given dataset



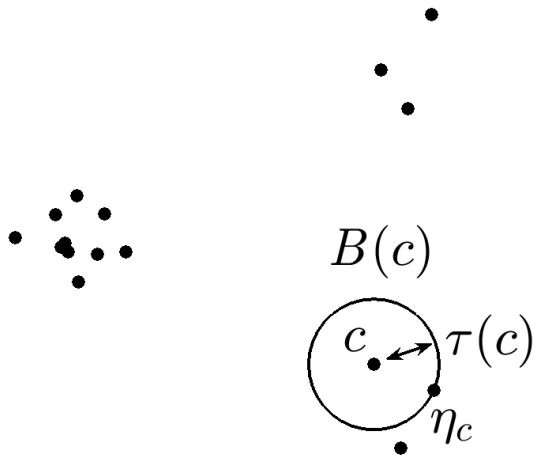
Local Regions

- Sample $\mathcal{S}$ is selected randomly from the given dataset
- Local-regions $B(c)$ are created centering each $c \in \mathcal{S}$



Local Regions

- Sample $\mathcal{S}$ is selected randomly from the given dataset
- Local-regions $B(c)$, are created centering each $c \in \mathcal{S}$
- Radius $\tau(c) = \|c - \eta_c\|$

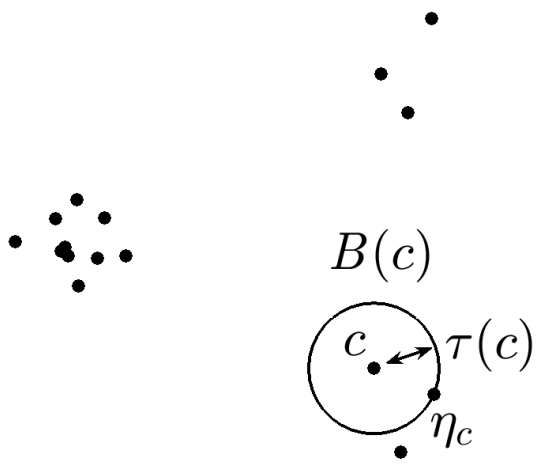


Local Regions

- Sample $\mathcal{S}$ is selected randomly from the given dataset
- Local-regions $B(c)$, are created centering each $c \in \mathcal{S}$
- Radius $\tau(c) = \|c - \eta_c\|$

  η_c is the nearest neighbour

  where $c,\ \eta_c \in \mathcal{S}$

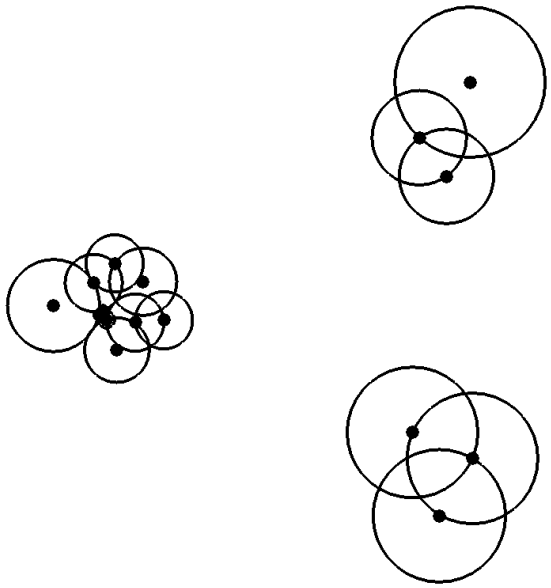


Local Regions

- Sample $\mathcal{S}$ is selected randomly from the given dataset
- Local-regions $B(c)$, are created centering each $c \in \mathcal{S}$
- Radius $\tau(c) = \|c - \eta_c\|$

η_c is the nearest neighbour

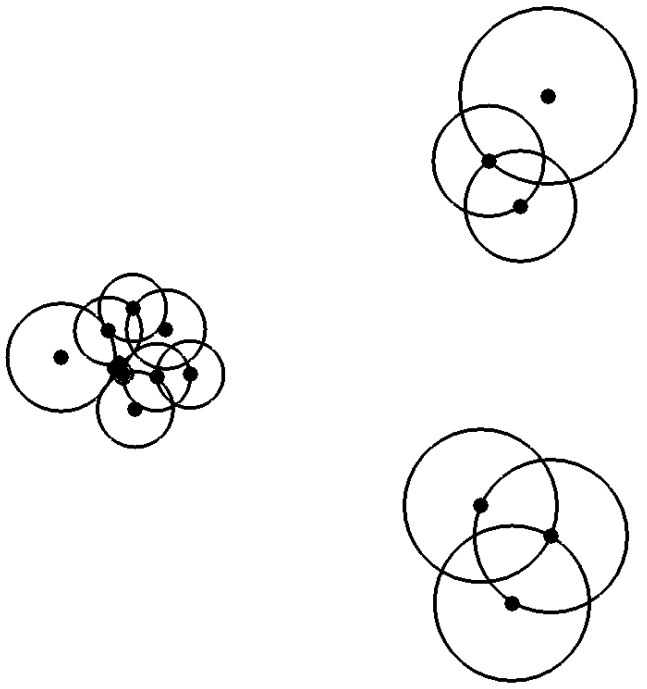
where $c,\ \eta_c \in \mathcal{S}$



Isolation Score

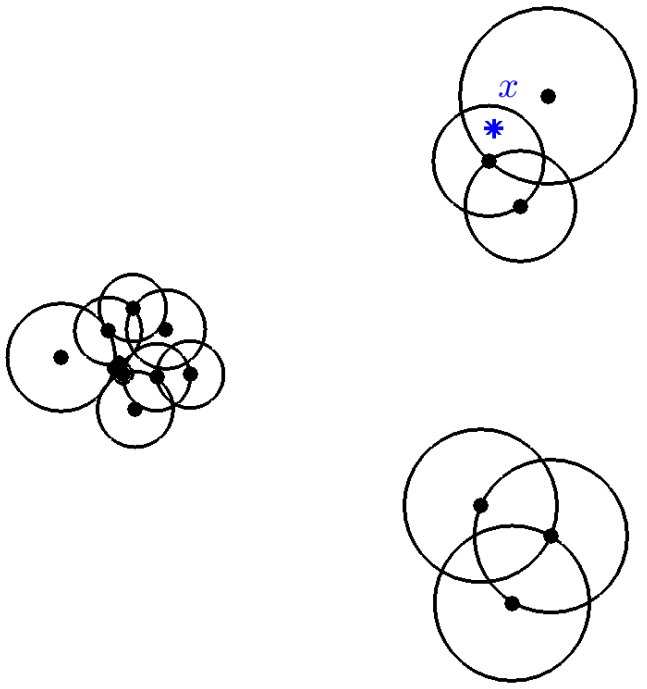
Isolation Score

- Based on $\left\{ B(c) : c \in \mathcal{S} \right\}$



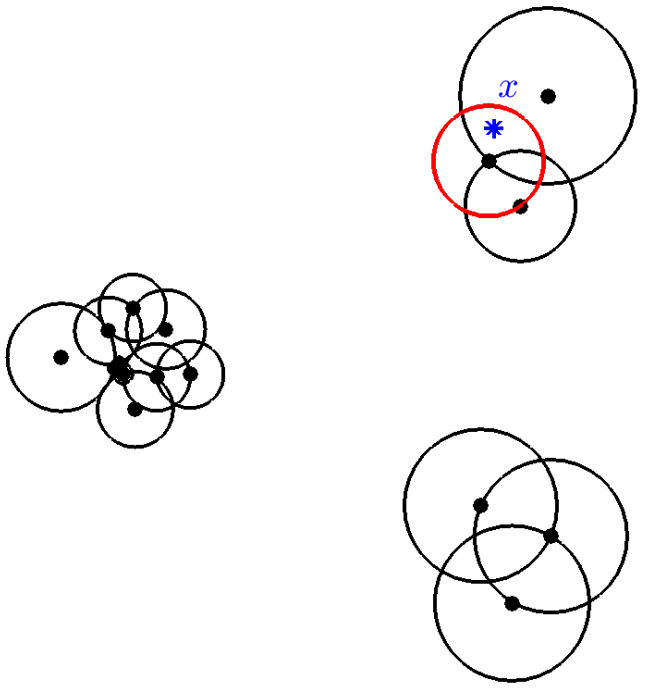
Isolation Score

- Based on $\left\{ B(c) : c \in \mathcal{S} \right\}$
- Isolation score I(x) for x



Isolation Score

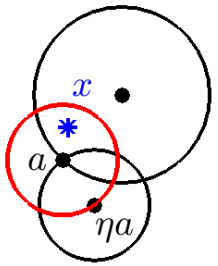
- Based on $\left\{B(c) : c \in \mathcal{S}\right\}$
- Isolation score I(x) for x
  - Find the smallest B(c) s.t. $x \in B(c)$

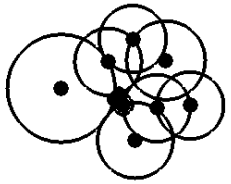


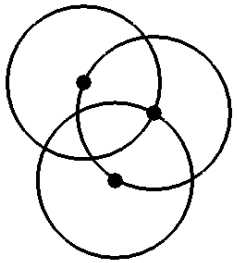
Isolation Score

- Based on $\left\{ B(c) : c \in \mathcal{S} \right\}$
- Isolation score I(x) for x
  - Find the smallest B(c) s.t. $x \in B(c)$
  - Isolation score based on the ratio

$$I(x) = 1 - \frac{\tau(\eta_a)}{\tau(a)}$$



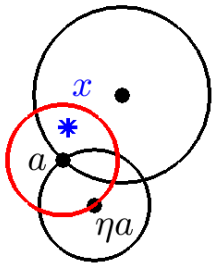


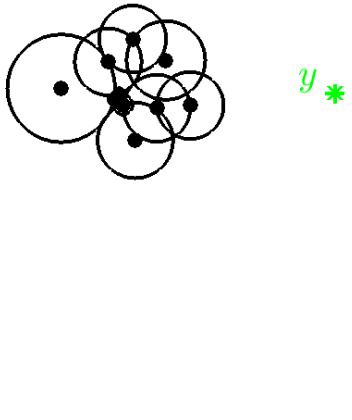


Isolation Score

- Based on $\left\{B(c) : c \in \mathcal{S}\right\}$
- Isolation score I(x) for x
  - Find the smallest B(c) s.t. $x \in B(c)$
  - Isolation score based on the ratio
- Isolation score I(y) for y

$$I(x) = 1 - \frac{\tau(\eta_a)}{\tau(a)}$$

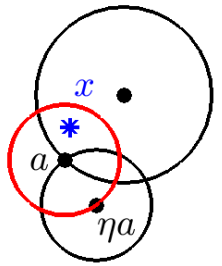


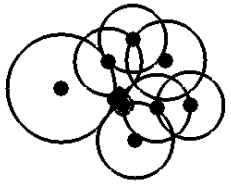


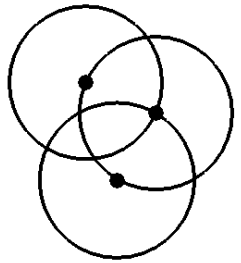
Isolation Score

- Based on $\left\{B(c) : c \in \mathcal{S}\right\}$
- Isolation score I(x) for x
  - Find the smallest B(c) s.t. $x \in B(c)$
  - Isolation score based on the ratio
- Isolation score I(y) for y
  - $\left\{y \in B(c)\right\} = \emptyset$

$$I(x) = 1 - \frac{\tau(\eta_a)}{\tau(a)}$$



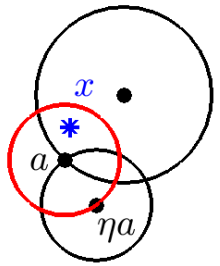


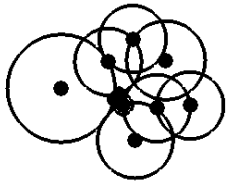


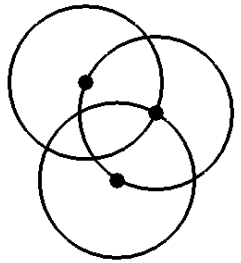
Isolation Score

- Based on $\left\{B(c) : c \in \mathcal{S}\right\}$
- Isolation score I(x) for x
  - Find the smallest B(c) s.t. $x \in B(c)$
  - Isolation score based on the ratio
- Isolation score I(y) for y
  - $\left\{y \in B(c)\right\} = \emptyset$
  - Maximum isolation score

$$I(x) = 1 - \frac{\tau(\eta_a)}{\tau(a)}$$
$$I(y) = 1$$







Anomaly score

- Average of isolation scores over an ensemble of size *t*

$$\bar{I}(x) = \frac{1}{t}\sum_{i=1}^{t} I_i(x)$$

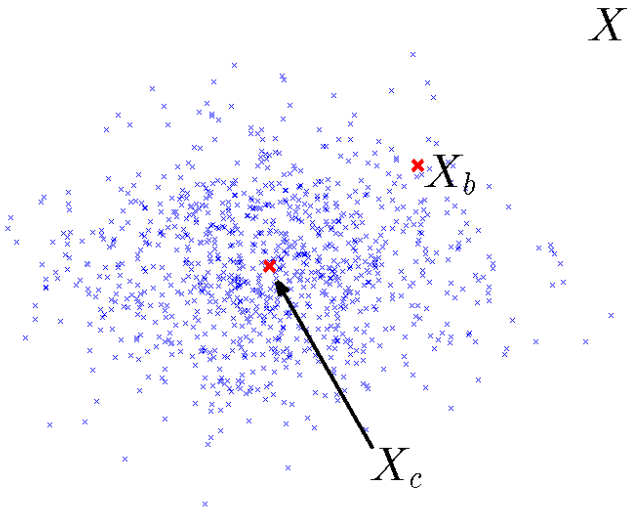
- Instances with high anomaly score are likely to be anomalies

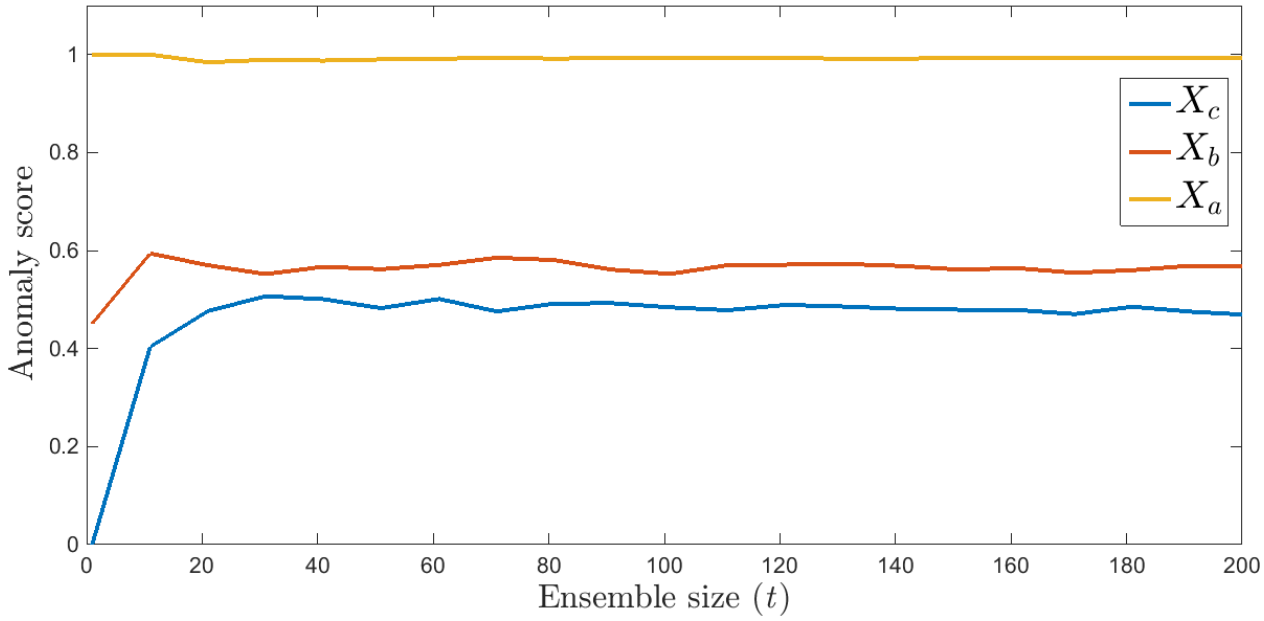
- Accuracy of the anomaly score improve with *t*
  - *t* = 100 is sufficient

- Sample size is a parameter setting
  - Similar to *k* in k-NN based methods
  - Empirical results show that the required sample size is usually in the range **2 - 128**

Example

- X_a get the maximum anomaly score
  - $I(X_a) = 1$
- X_b and X_c get lower anomaly scores





Time and space complexity

- Time complexity
  - Training stage: **O($t\,\Psi^2$)**, t = ensemble size, Ψ = sample size
  - Evaluation stage: **O($n t\,\Psi$)**, n = data size
  - t and Ψ are constants for iNNE, t << n and Ψ << n
    (Default values: t = 100 and Ψ in the range 2 to 128)
  - Thus time complexity of iNNE is **linear with n**

- Space complexity
  - Only need to store the sets of hyperspheres
  - Hence has a constant space complexity: **O($t\,\Psi$)**

iNNE : Advantages over iForest

- **Adapts well to local distribution** better than axis-parallel subdivision
- **Uses all the available attributes** to partition data space into regions
- Isolation score **is a local measure**, which is defined relative to the local neighbourhood

Comparison with LOF

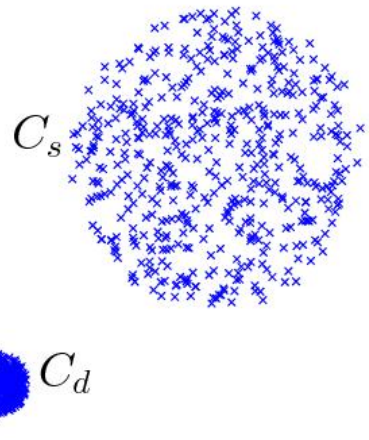
- Similarities
  - Employ NN distance
  - Score based on relative measure to local-neighbourhood

- Differences : O(n) versus $O(n^2)$
  - iNNE : An ensemble based eager learner
  - LOF: Lazy learner

  - iNNE: Partition the space in to regions based on NN distance
    - Does not relies on the accuracy of underlying k-NN density estimator

  - LOF: Estimates the relative-density based on k-NN distance
    - Heavily relies on the accuracy of underlying k-NN density estimator
    - Hence, ensemble version of LOF (Zimek et al., 2013) requires a larger sample size than iNNE

Detection of local anomalies



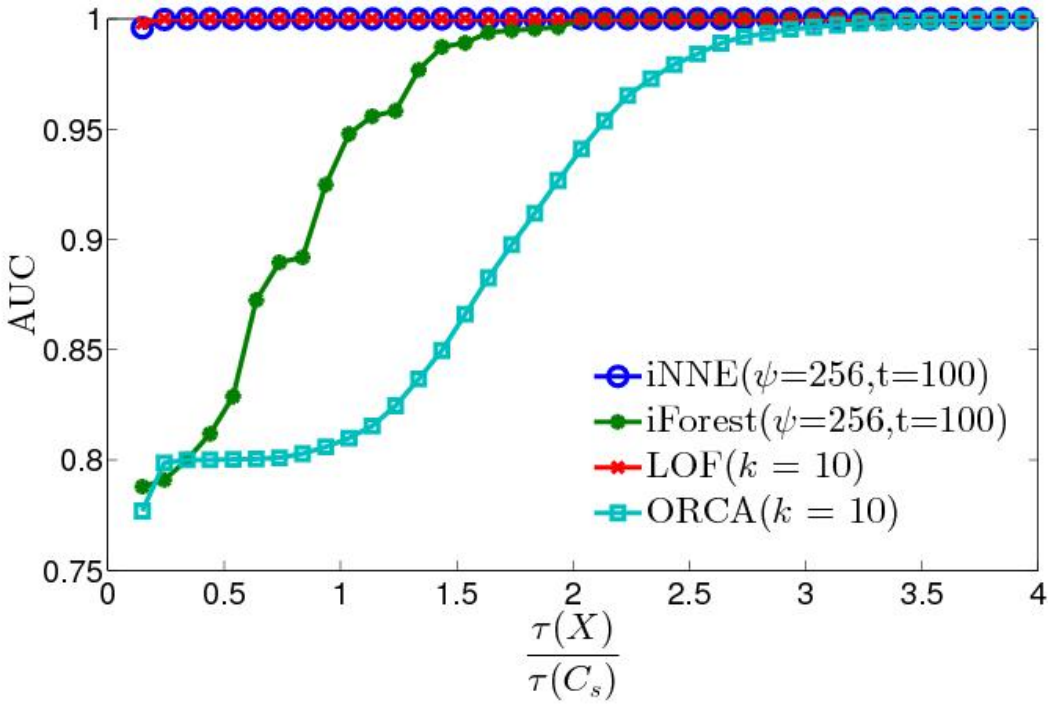
$\tau(C_s)$ *is the average nearest neighbour distance of* C_s
$\tau(C_d)$ *is the average nearest neighbour distance of* C_d
$\tau(X)$ *is the nearest neighbour distance of* X.

X *is an anomaly if and only if*

$$\tau(X) > \tau(C_d)$$

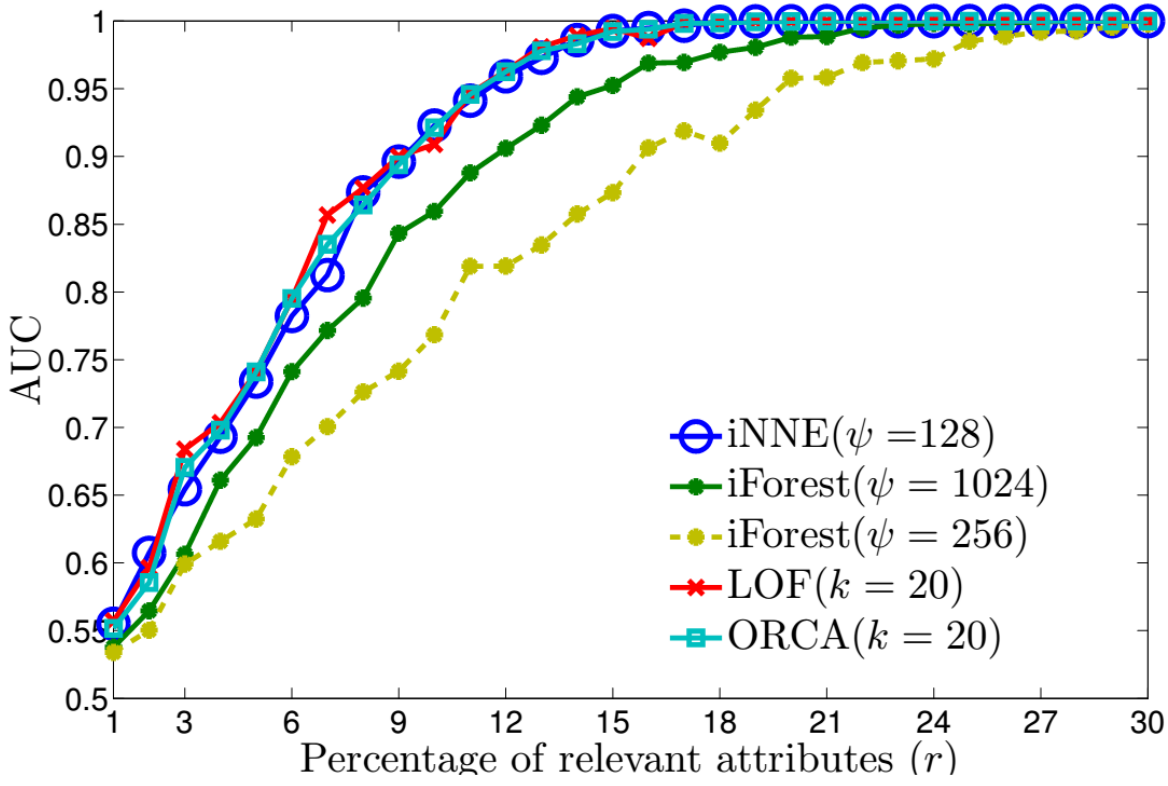
$$\tau(X) < \tau(C_s) \Rightarrow X \text{ is a local anomaly}$$

$$\tau(X) > \tau(C_s) \Rightarrow X \text{ is a global anomaly}$$



Resilient to low relevant dimensions

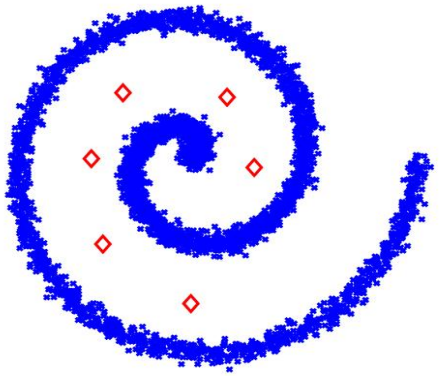
- 1000 dimensional dataset used, while changing percentage of relevant dimensions from 1% to 30%
- Irrelevant dimensions have random noise
- iNNE is more resilient than iForest



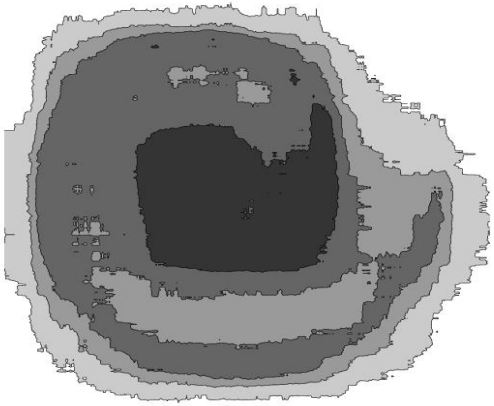
Axis parallel masking

- iNNE produces better contour maps of anomaly scores, tightly fitted to the data distribution

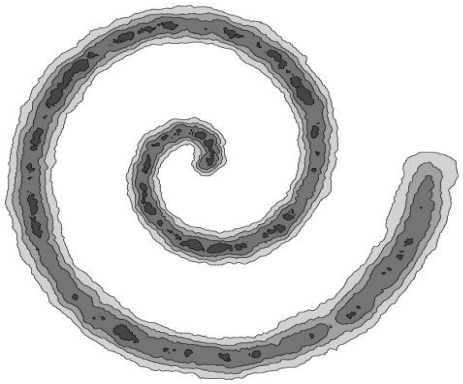
Dataset



iForest



iNNE



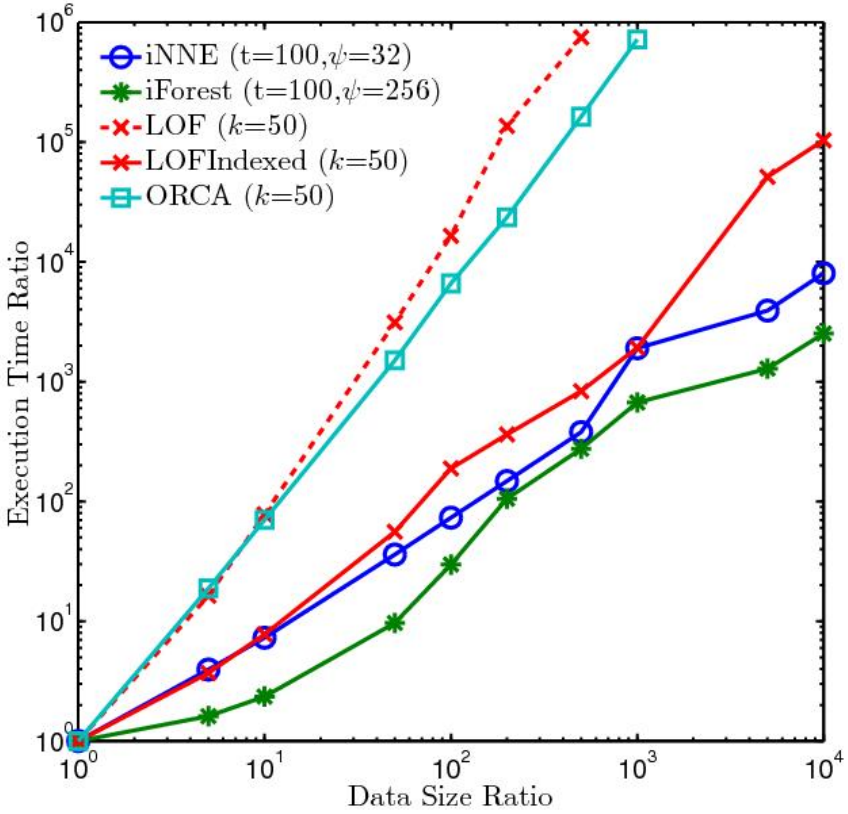
- Spiral dataset with 4000 normal instances (blue cross) and 6 anomaly instances (red diamond)
- **iNNE**: AUC = 1:00, Anomaly Ranking: 1 - 6
- **iForest** : AUC = 0:86, Anomaly Ranking: 75, 320, 345, 354, 563, 1802

Scaleup test: Increasing size of dataset

- Compared execution time against iForest , LOF and ORCA
- 5 dimensional datasets are used with increasing size
- iNNE can efficiently scale up to very large datasets
- For a 10-million dataset

  iForest : 9 m
  iNNE : 1 h 40 m
  LOF: 220 d (projected)
  LOFIndexed: 7 h 30 m
  ORCA: 15 d (projected)

LOFIndexed = LOF with R*-Tree indexing

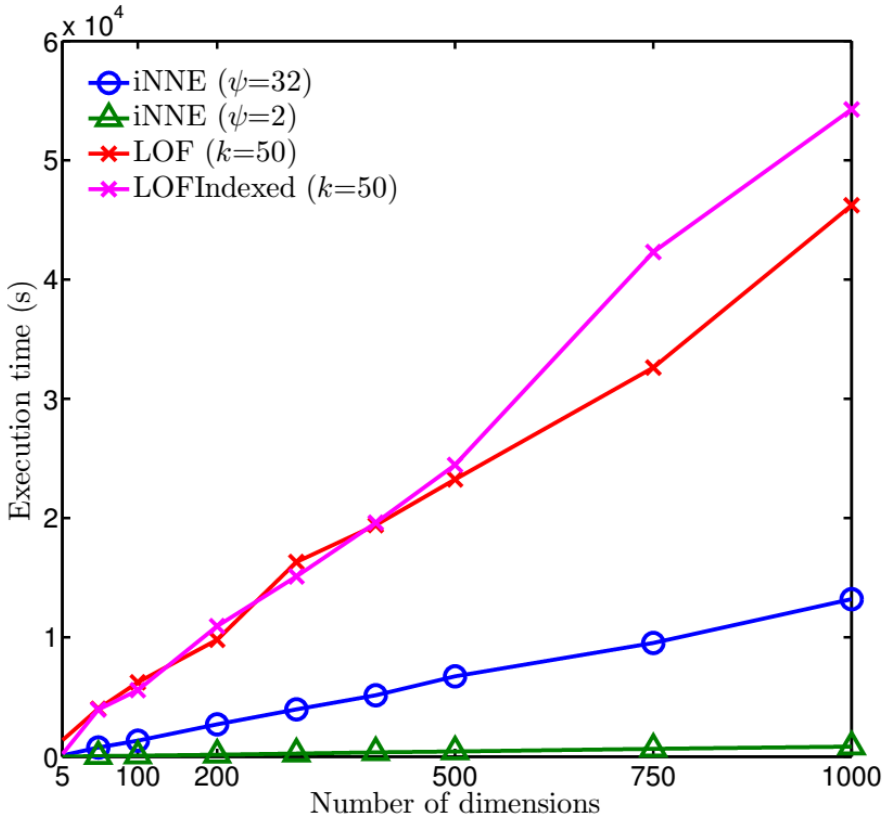


Scaleup test: Increasing dimensions of dataset

- Compared execution time against LOF and ORCA
- 100,000 instance datasets are used with increasing dimensions
- For 1000-dimension dataset

  iNNE(Ψ = 2): 14m
  iNNE(Ψ = 32): 3 h 40 m
  LOF: 12h 50m
  LOFIndexed: 15h

- iNNE efficiently scales up to high dimensional datasets
- An indexing scheme becomes more expensive in high dimensions



Performance in Benchmark datasets

Dataset	Data Size (anomaly %)		d	AUC iNNE	LOF	ORCA	iForest	EnLOF
http	567,497	(0.4)	3	1.00	1.00	1.00	1.00	1.00
cover	286,048	(0.9)	10	0.97	0.94	0.88	0.94	0.98
mulcross	262,144	(1.0)	4	1.00	1.00	1.00	1.00	1.00
smtp	95,156	(0.03)	3	0.95	0.95	0.74	0.92	0.95
shuttle	49,097	(7.0)	9	0.99	0.98	0.99	1.00	0.99
mnist	20,444	(3.3)	96	0.87	0.87	0.88	0.85	0.87
har	5,272	(11.4)	561	0.99	0.99	0.99	0.94	0.99
isolet	730	(1.4)	617	1.00	1.00	1.00	1.00	1.00
mfeat	410	(2.4)	649	0.98	0.98	0.98	0.95	0.98
p53Mutant	31,159	(0.5)	5408	0.73	0.75	NA	0.61	0.75

Dataset	Execution Time(CPU seconds) iNNE	LOF	ORCA	iForest	EnLOF	Best Parameter iNNE ψ	LOF k	ORCA k	iForest ψ	EnLOF k
http	23	19,965	78,931	66	295,564	2	500	3000	256	300
cover	202	2,918	94,336	52	78,373	16	1000	3000	512	200
mulcross	13	2,169	56,372	5	74,581	2	2000	3000	32	200
smtp	481	373	125	13	2,789	128	1000	40	512	100
shuttle	5	656	16,137	3	1,729	2	4000	4000	64	500
mnist	275	678	111	2	285	32	300	60	512	50
har	25	193	409	0.4	76	2	4000	4000	32	400
isolet	4	2	1	0.3	2	2	40	10	32	2
mfeat	10	1	0.5	0.6	1	8	80	40	128	5
p53Mutant	8,275	43,235	NA	19	57,166	16	2000	NA	512	200

Summary

- iNNE performs isolation by creating local regions based on the NN distance
- It overcomes the identified weaknesses of iForest to detect
  - local anomalies
  - anomalies with low relevant dimensions
  - anomalies masked by axis parallel normal clusters
- Has a linear time complexity with data size, thus can scaleup efficiently
- Efficiency does not degrade with the increase of dimensions

Thank you !!!

Any Questions ?

References

- *M. Ester, H. peter Kriegel, J. S, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining, AAAI Press, 1996.*
- *Z. He, X. Xu, and S. Deng. Discovering cluster-based local outliers. Pattern Recognition Letters, 24(9-10):1641-1650, June 2003.*
- *S. Ramaswamy, R. Rastogi, and K. Shim. Efficient Algorithms for Mining Outliers from Large Data Sets. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, SIGMOD '00, ACM, 2000.*
- *F. Angiulli and C. Pizzuti. Fast Outlier Detection in High Dimensional Spaces. In T. Elomaa, H. Mannila, and H. Toivonen, editors, Principles of Data Mining and Knowledge Discovery, volume 2431 of Lecture Notes in Computer Science, pages 15-27. Springer Berlin Heidelberg, 2002.*
- *D. Ren, I. Rahal, W. Perrizo, and K. Scott. A Vertical Distance-based Outlier Detection Method with Local Pruning. In Proceedings of the 13th ACM International Conference on Information and Knowledge Management, CIKM '04, ACM, 2004.*
- *M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: Identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, ACM, 2000.*
- *S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos. LOCI: Fast Outlier Detection Using the Local Correlation Integral. In Proceedings of 19th International Conference on Data Engineering, 2003.*
- *F. Liu, K. M. Ting, and Z.-H. Zhou. Isolation Forest. In Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on, pages 413-422, Dec 2008.*
- *S. D. Bay and M. Schwabacher. Mining Distance-based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule. In Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2003.*
- *A. Zimek, M. Gaudet, R. J. Campello, and J. Sander, Subsampling for Efficient and Effective Unsupervised Outlier Detection Ensembles, In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2013.*