Anomaly Detection

Some slides taken or adapted from:

"Anomaly Detection: A Tutorial"

Arindam Banerjee, Varun Chandola, Vipin Kumar, Jaideep Srivastava, *University of Minnesota*

Aleksandar Lazarevic, *United Technology Research Center*

Anomaly detection

Anomalies and outliers
are essentially
the same thing:

objects that are different from most other objects

The techniques used for detection are the same.

Anomaly detection

- Historically, the field of statistics tried to find and remove outliers as a way to improve analyses.

- There are now many fields where the outliers / anomalies are the objects of greatest interest.
  - The rare events may be the ones with the greatest impact, and often in a negative way.

Causes of anomalies

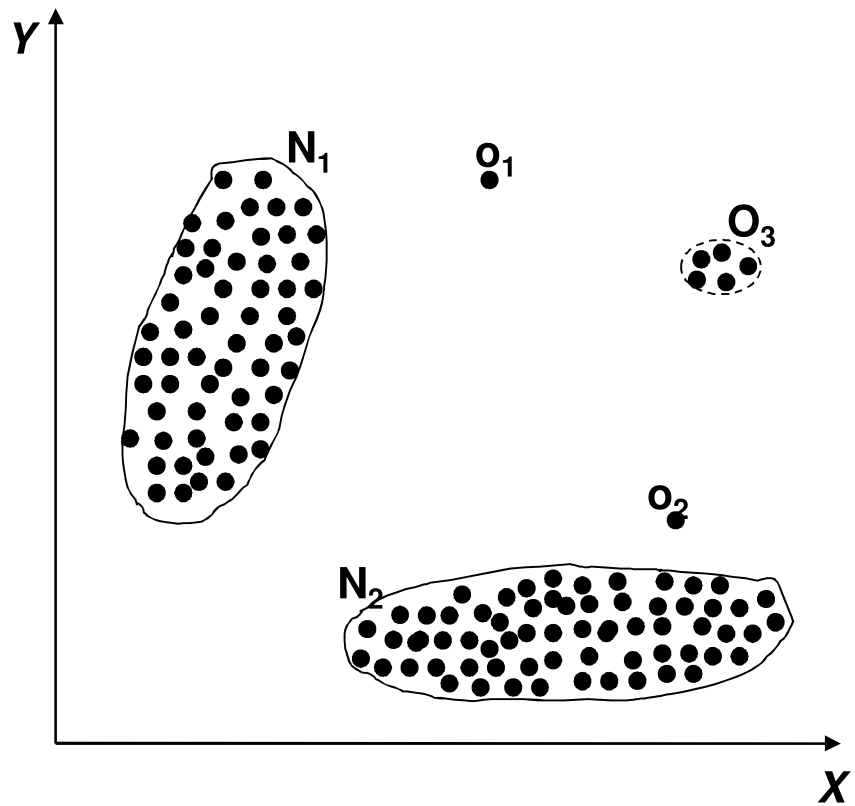
- Data from different class of object or underlying mechanism
  - disease vs. non-disease
  - fraud vs. not fraud
- Natural variation
  - tails on a Gaussian distribution
- Data measurement and collection errors

Structure of anomalies

- Point anomalies
- Contextual anomalies
- Collective anomalies

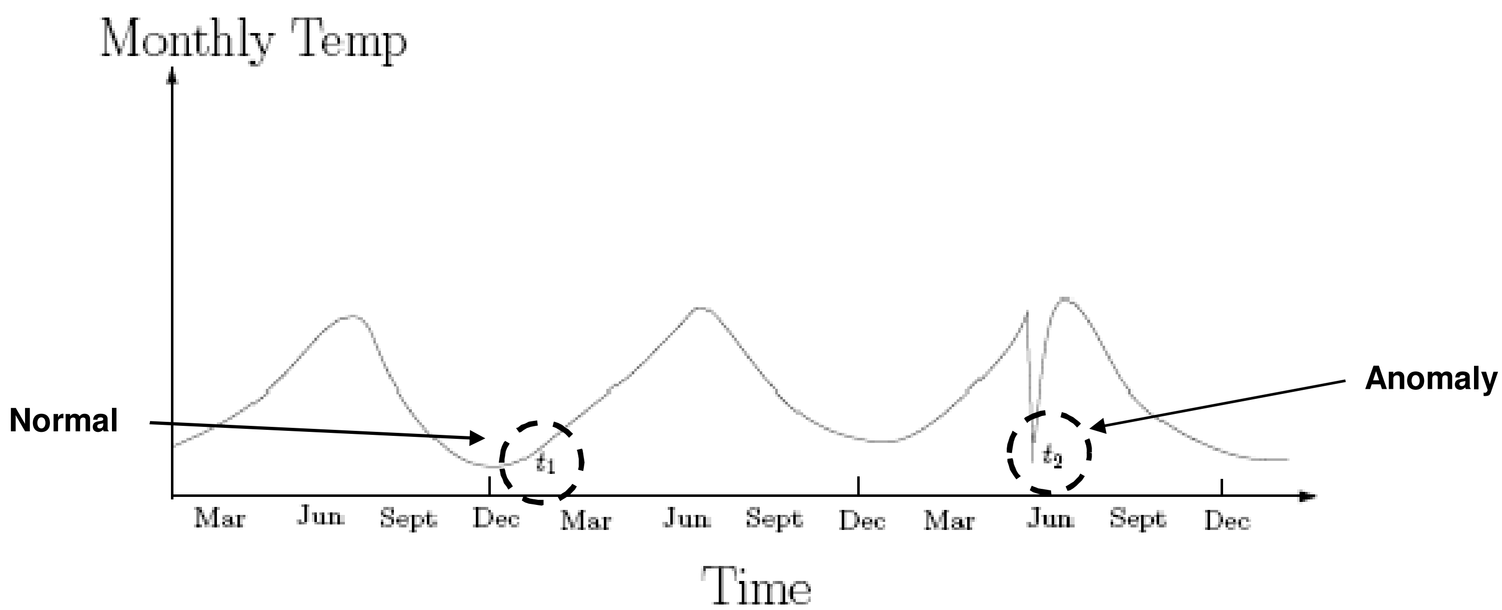
Point anomalies

- An individual data instance is anomalous with respect to the data



Contextual anomalies

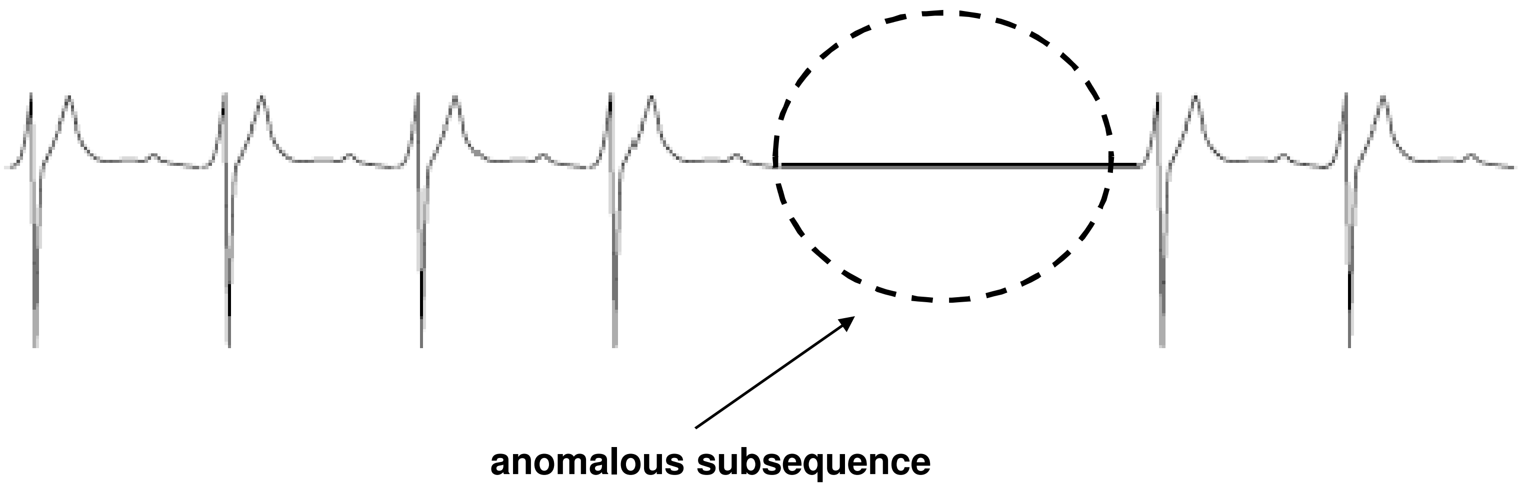
- An individual data instance is anomalous within a context
- Requires a notion of context
- Also referred to as conditional anomalies *



* Song, et al, "Conditional Anomaly Detection", IEEE Transactions on Data and Knowledge Engineering, 2006.

Collective anomalies

- A collection of related data instances is anomalous
- Requires a relationship among data instances
  - Sequential data
  - Spatial data
  - Graph data
- The individual instances within a collective anomaly are not anomalous by themselves



Applications of anomaly detection

- Network intrusion
- Insurance / credit card fraud
- Healthcare informatics / medical diagnostics
- Industrial damage detection
- Image processing / video surveillance
- Novel topic detection in text mining
- …

Intrusion detection

- Intrusion detection
  – Monitor events occurring in a computer system or network and analyze them for intrusions
  – Intrusions defined as attempts to bypass the security mechanisms of a computer or network
- Challenges
  – Traditional intrusion detection systems are based on signatures of known attacks and cannot detect emerging cyber threats
  – Substantial latency in deployment of newly created signatures across the computer system
- Anomaly detection can alleviate these limitations

Fraud detection

- Detection of criminal activities occurring in commercial organizations.
- Malicious users might be:
  - Employees
  - Actual customers
  - Someone posing as a customer (identity theft)
- Types of fraud
  - Credit card fraud
  - Insurance claim fraud
  - Mobile / cell phone fraud
  - Insider trading
- Challenges
  - Fast and accurate real-time detection
  - Misclassification cost is very high

Healthcare informatics

- Detect anomalous patient records
  - Indicate disease outbreaks, instrumentation errors, etc.
- Key challenges
  - Only normal labels available
  - Misclassification cost is very high
  - Data can be complex: spatio-temporal

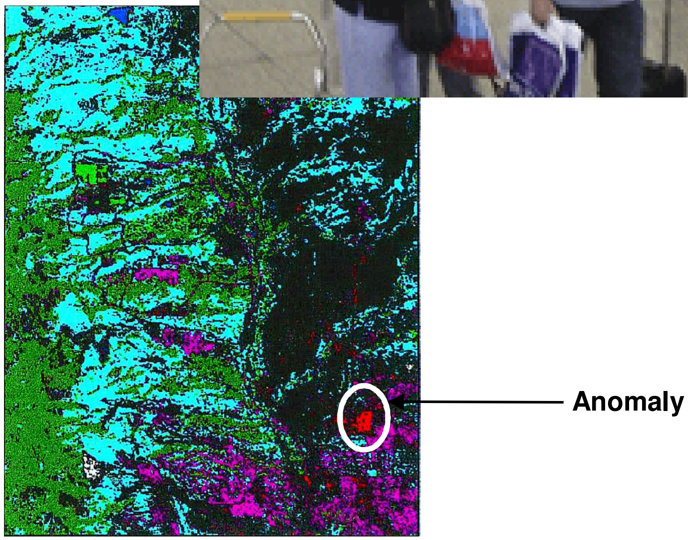
Industrial damage detection

- Detect faults and failures in complex industrial systems, structural damages, intrusions in electronic security systems, suspicious events in video surveillance, abnormal energy consumption, etc.
  - Example: aircraft safety
    - anomalous aircraft (engine) / fleet usage
    - anomalies in engine combustion data
    - total aircraft health and usage management

- Key challenges
  - Data is extremely large, noisy, and unlabelled
  - Most of applications exhibit temporal behavior
  - Detected anomalous events typically require immediate intervention

Image processing

- Detecting outliers in a image monitored over time
- Detecting anomalous regions within an image
- Used in
  - mammography image analysis
  - video surveillance
  - satellite image analysis
- Key Challenges
  - Detecting collective anomalies
  - Data sets are very large



Use of data labels in anomaly detection

- Supervised anomaly detection
  - Labels available for both normal data and anomalies
  - Similar to classification with high class imbalance
- Semi-supervised anomaly detection
  - Labels available only for normal data
- Unsupervised anomaly detection
  - No labels assumed
  - Based on the assumption that anomalies are very rare compared to normal data

Output of anomaly detection

- Label
  - Each test instance is given a *normal* or *anomaly* label
  - Typical output of classification-based approaches
- Score
  - Each test instance is assigned an anomaly score
    - allows outputs to be ranked
    - requires an additional threshold parameter

Variants of anomaly detection problem

- Given a dataset D, find all the data points **x** $\in$ D with anomaly scores greater than some threshold t.
- Given a dataset D, find all the data points **x** $\in$ D having the top-n largest anomaly scores.
- Given a dataset D, containing mostly normal data points, and a test point **x**, compute the anomaly score of **x** with respect to D.

Unsupervised anomaly detection

- No labels available
- Based on assumption that anomalies are very rare compared to “normal” data

- General steps
  - Build a profile of “normal” behavior
    - summary statistics for overall population
    - model of multivariate data distribution
  - Use the “normal” profile to detect anomalies
    - anomalies are observations whose characteristics differ significantly from the normal profile

Techniques for anomaly detection

- Statistical
- Proximity-based
- Density-based
- Clustering-based

[following slides illustrate these techniques for unsupervised detection of point anomalies]

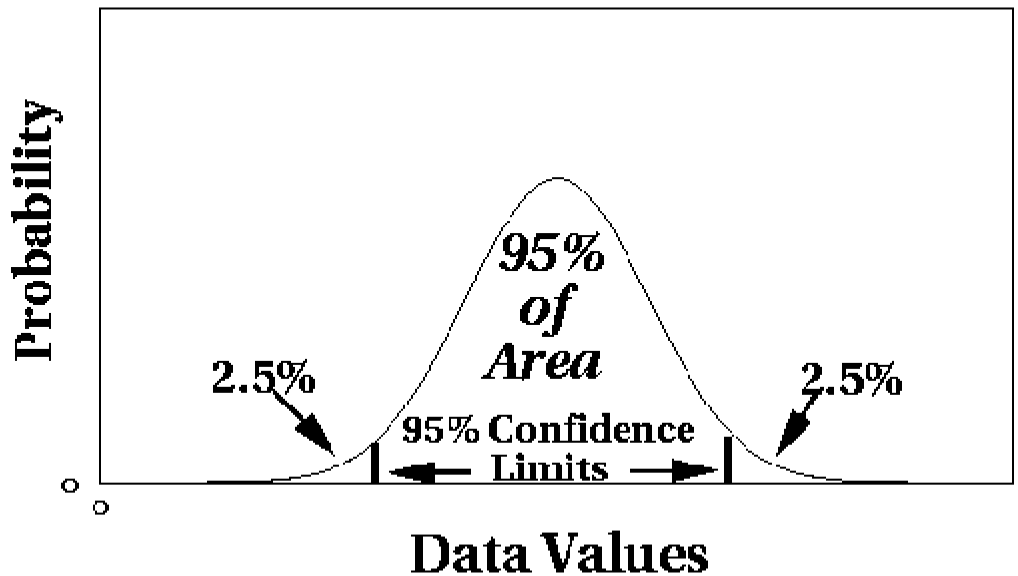
Statistical outlier detection

Outliers are objects that are fit poorly by a statistical model.

- Estimate a parametric model describing the distribution of the data
- Apply a statistical test that depends on
  - Properties of test instance
  - Parameters of model (e.g., mean, variance)
  - Confidence limit (related to number of expected outliers)

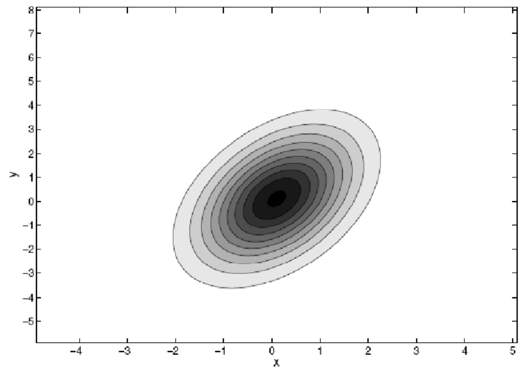
Statistical outlier detection

- Univariate Gaussian distribution
  - Outlier defined by z-score > threshold

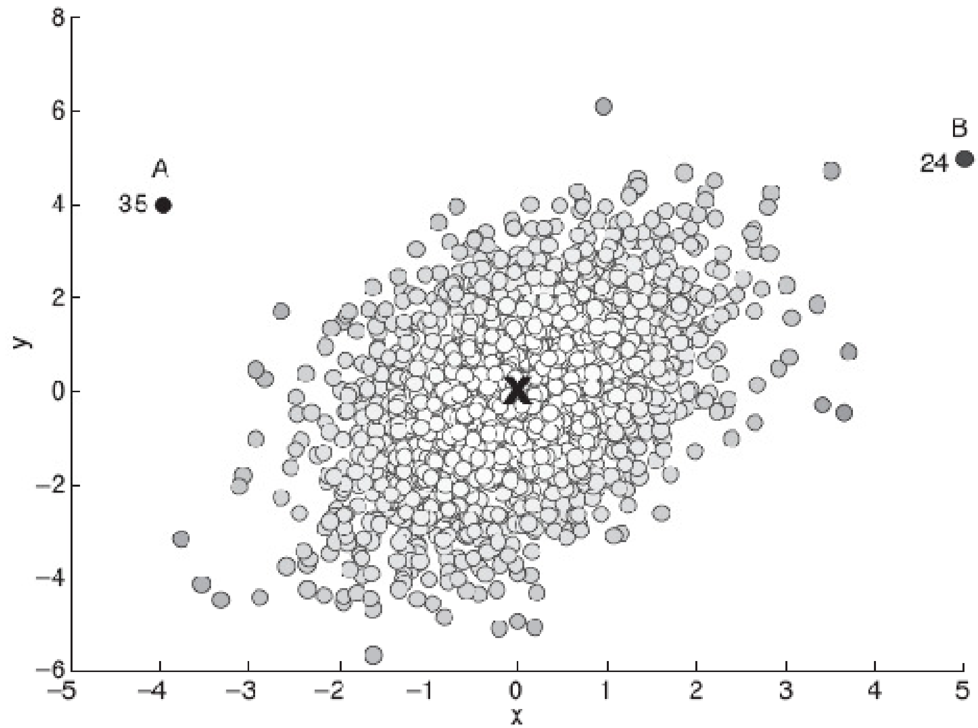


Statistical anomaly detection

- Multivariate Gaussian distribution
  - Outlier defined by Mahalanobis distance > threshold



	Distance	
	Euclidean	Mahalanobis
A	5.7	35
B	7.1	24



Grubbs' test

- Detect outliers in univariate data
- Assume data comes from normal distribution
- Detects one outlier at a time, remove the outlier, and repeat
  - H_0: There is no outlier in data
  - H_A: There is at least one outlier
- Grubbs' test statistic:

$$G = \frac{\max\left|X - \overline{X}\right|}{s}$$

- Reject H_0 if:

$$G > \frac{(N-1)}{\sqrt{N}} \sqrt{\frac{t^2_{(\alpha/N,N-2)}}{N-2+t^2_{(\alpha/N,N-2)}}}$$

Likelihood approach

- Assume the dataset D contains samples from a mixture of two probability distributions:
  - M (majority distribution)
  - A (anomalous distribution)
- General approach:
  - Initially, assume all the data points belong to M
  - Let $L_t(D)$ be the log likelihood of D at time t
  - For each point x_t that belongs to M, move it to A
    - Let L_{t+1} (D) be the new log likelihood.
    - Compute the difference, $\Delta = L_t(D) - L_{t+1}$ (D)
    - If $\Delta > c$ (some threshold), then x_t is declared as an anomaly and moved permanently from M to A

Likelihood approach

- Data distribution, D = (1 – λ) M + λ A
- M is a probability distribution estimated from data
  - Can be based on any modeling method (naïve Bayes, maximum entropy, etc)
- A is initially assumed to be uniform distribution
- Likelihood at time t:

$$L_t(D) = \prod_{i=1}^{N} P_D(x_i) = \left((1-\lambda)^{|M_t|} \prod_{x_i \in M_t} P_{M_t}(x_i) \right) \left(\lambda^{|A_t|} \prod_{x_i \in A_t} P_{A_t}(x_i) \right)$$

$$LL_t(D) = |M_t| \log(1-\lambda) + \sum_{x_i \in M_t} \log P_{M_t}(x_i) + |A_t| \log \lambda + \sum_{x_i \in A_t} \log P_{A_t}(x_i)$$

Statistical outlier detection

- Pros
  - Statistical tests are well-understood and well-validated.
  - Quantitative measure of degree to which object is an outlier.
- Cons
  - Data may be hard to model parametrically.
    - multiple modes
    - variable density
  - In high dimensions, data may be insufficient to estimate true distribution.

Proximity-based outlier detection

Outliers are objects far away from other objects.

- Common approach:
  - Outlier score is distance to k^{th} nearest neighbor.
  - Score sensitive to choice of k.

Proximity-based outlier detection

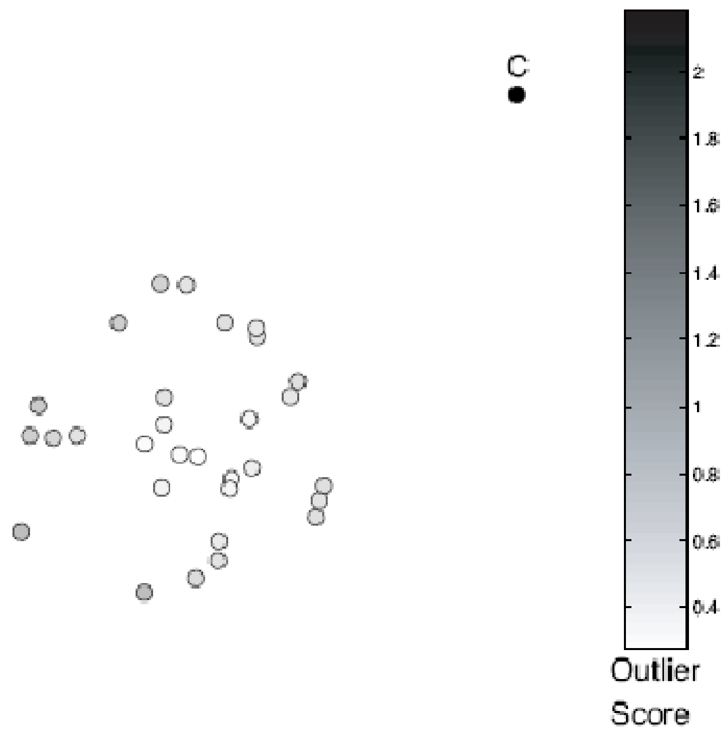


Figure 10.4. Outlier score based on the distance to fifth nearest neighbor.

Proximity-based outlier detection

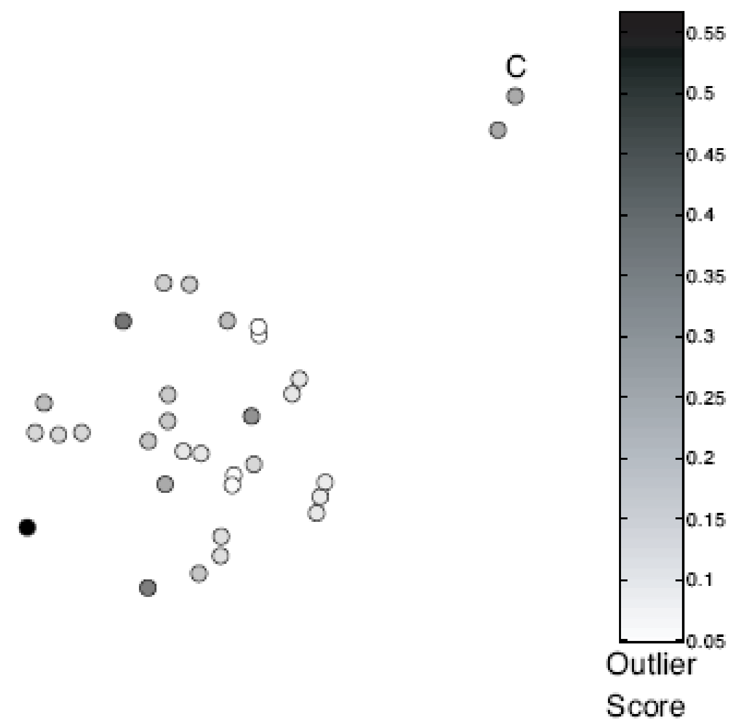


Figure 10.5. Outlier score based on the distance to the first nearest neighbor. Nearby outliers have low outlier scores.

Proximity-based outlier detection

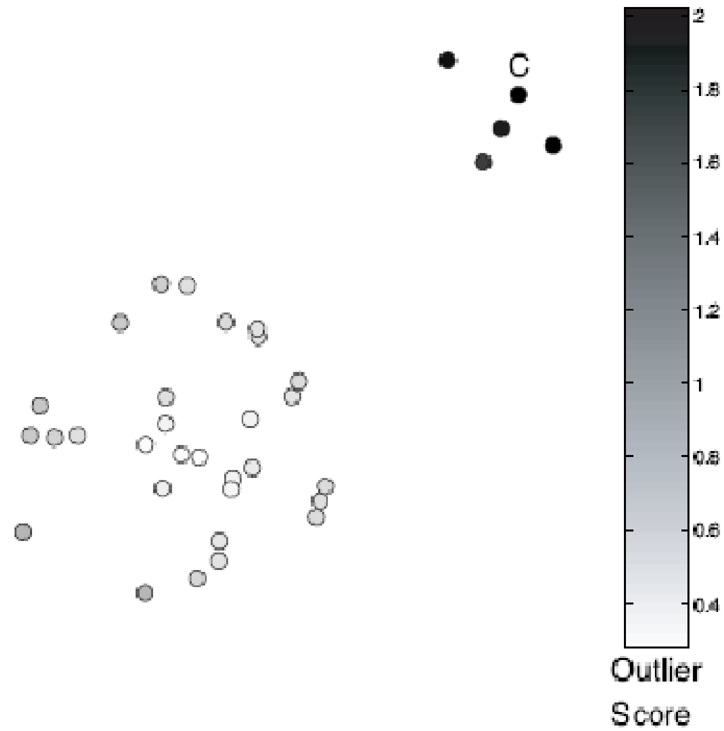


Figure 10.6. Outlier score based on distance to the fifth nearest neighbor. A small cluster becomes an outlier.

Proximity-based outlier detection

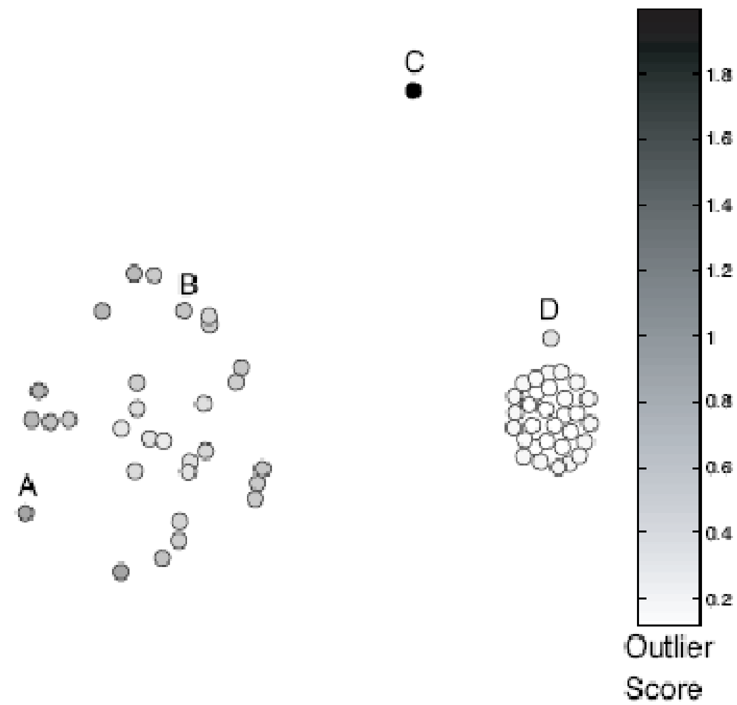


Figure 10.7. Outlier score based on the distance to the fifth nearest neighbor. Clusters of differing density.

Proximity-based outlier detection

- Pros
  - Easier to define a proximity measure for a dataset than determine its statistical distribution.
  - Quantitative measure of degree to which object is an outlier.
  - Deals naturally with multiple modes.

- Cons
  - $O(n^2)$ complexity.
  - Score sensitive to choice of k.
  - Does not work well if data has widely variable density.

Density-based outlier detection

Outliers are objects in regions of low density.

- Outlier score is inverse of density around object.
- Scores usually based on proximities.
- Example scores:
  - Reciprocal of average distance to *k* nearest neighbors:

$$\text{density}(\mathbf{x}, k) = \left(\frac{1}{k} \sum_{\mathbf{y} \in N(\mathbf{x}, k)} \text{distance}(\mathbf{x}, \mathbf{y}) \right)^{-1}$$

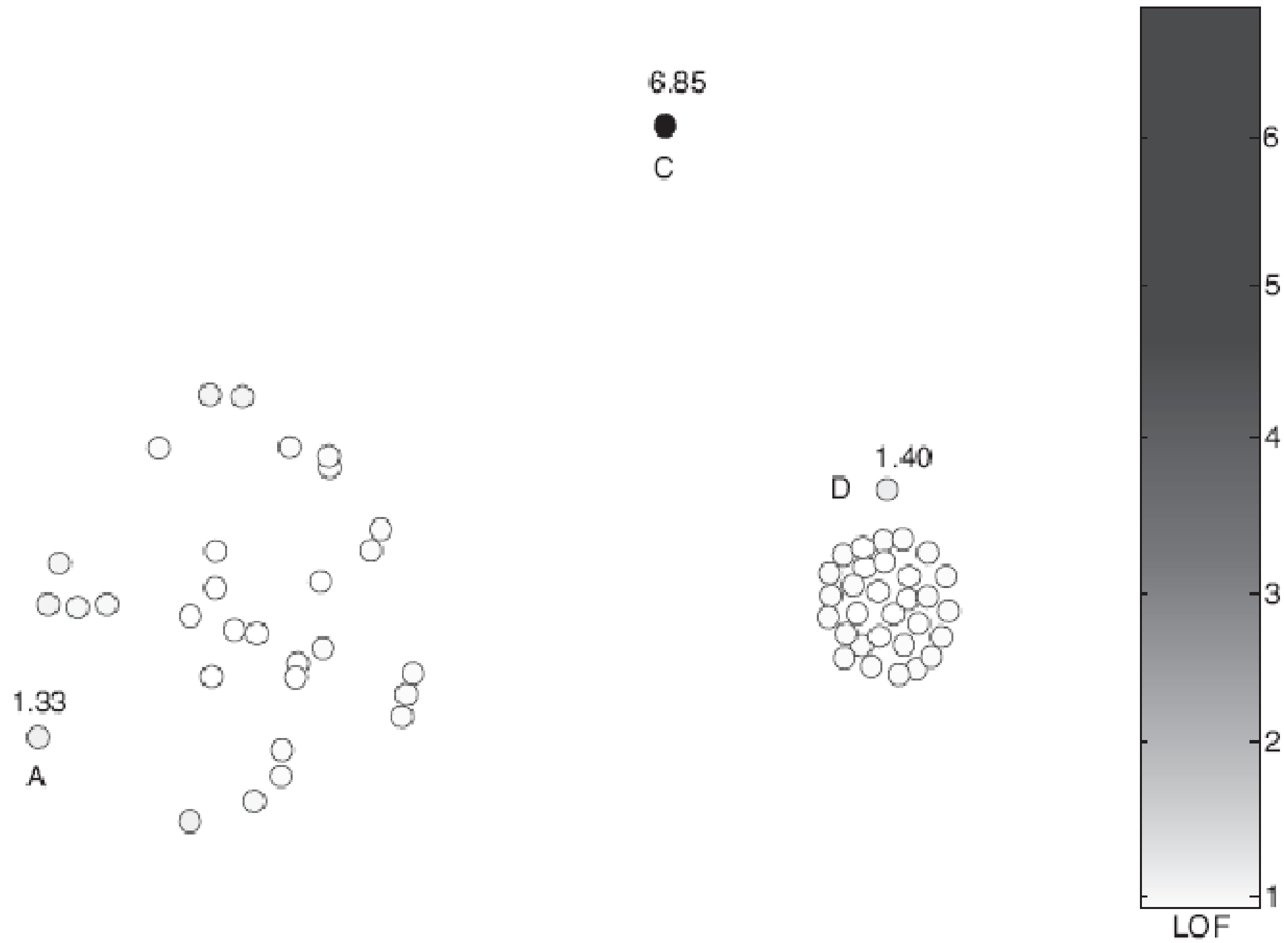
  - Number of objects within fixed radius *d* (DBSCAN).
  - These two example scores work poorly if data has variable density.

Density-based outlier detection

- Relative density outlier score (Local Outlier Factor, LOF)
  - Reciprocal of average distance to *k* nearest neighbors, relative to that of those *k* neighbors.

$$\text{relative density}(\mathbf{x}, k) = \frac{\text{density}(\mathbf{x}, k)}{\frac{1}{k}\sum_{\mathbf{y} \in N(\mathbf{x}, k)} \text{density}(\mathbf{y}, k)}$$

Density-based outlier detection



relative density (LOF) outlier scores

Density-based outlier detection

- Pros
  - Quantitative measure of degree to which object is an outlier.
  - Can work well even if data has variable density.

- Cons
  - $O(n^2)$ complexity
  - Must choose parameters
    - *k* for nearest neighbor
    - *d* for distance threshold

Cluster-based outlier detection

Outliers are objects that do not belong strongly to any cluster.

- Approaches:
  - Assess degree to which object belongs to any cluster.
  - Eliminate object(s) to improve objective function.
  - Discard small clusters far from other clusters.
- Issue:
  - Outliers may affect initial formation of clusters.

Cluster-based outlier detection

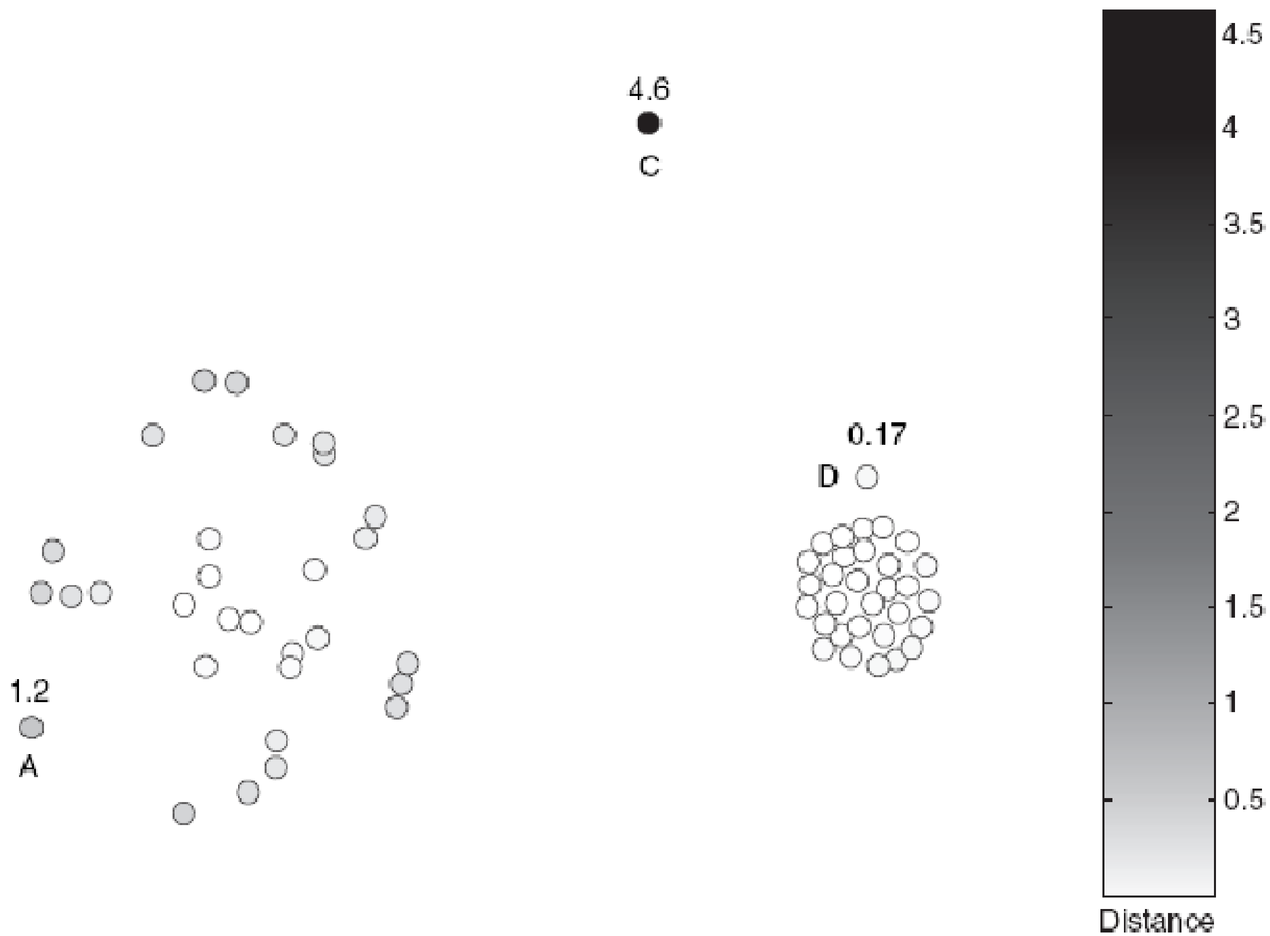
Assess degree to which object belongs to any cluster.

- For prototype-based clustering (e.g. k-means), use distance to cluster centers.
  - To deal with variable density clusters, use relative distance:

$$\frac{\text{distance}(\mathbf{x}, centroid_C)}{\text{median}\left(\left\{\forall_{x' \in C}\ \text{distance}(\mathbf{x}', centroid_C)\right\}\right)}$$

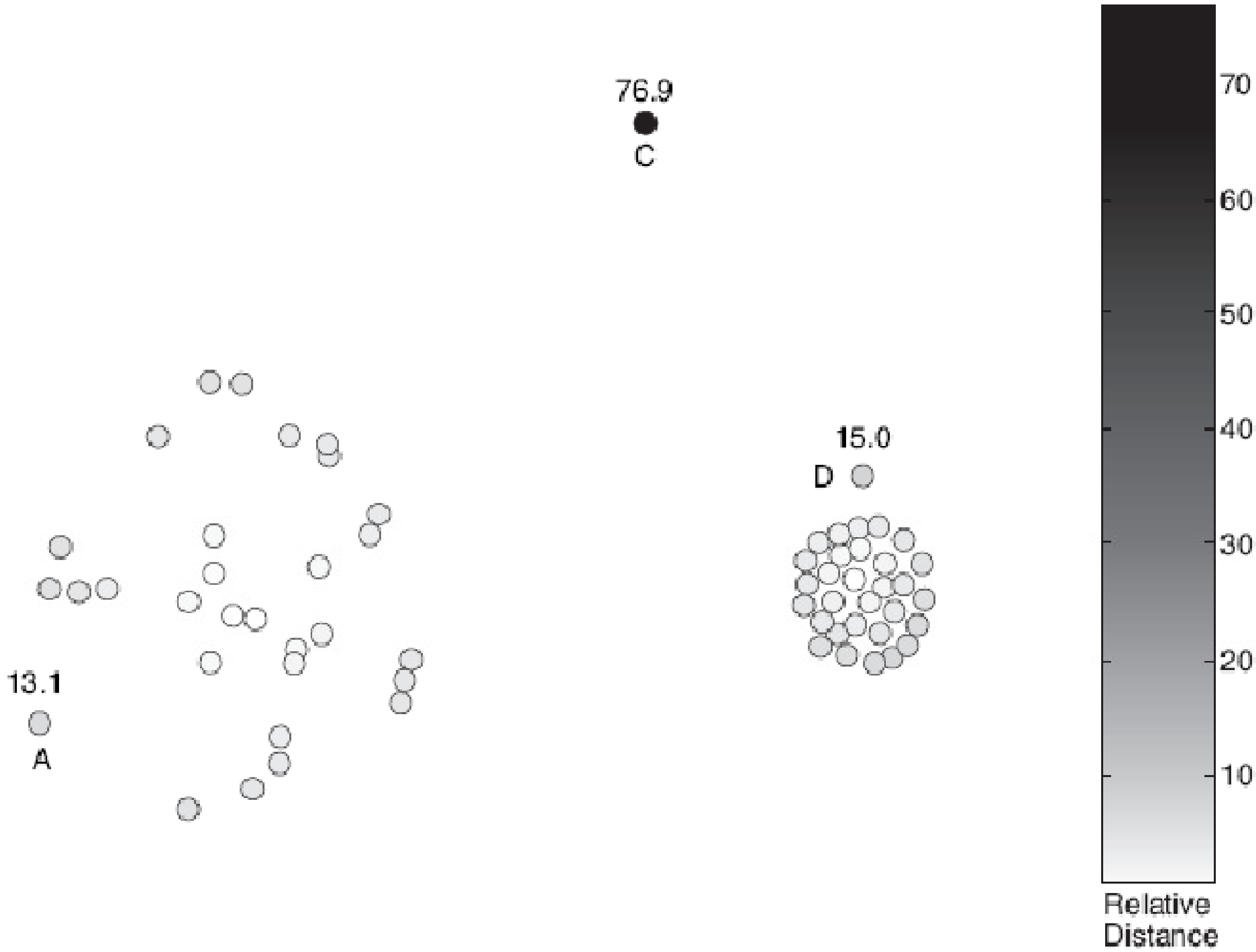
- Similar concepts for density-based or connectivity-based clusters.

Cluster-based outlier detection



distance of points from nearest centroid

Cluster-based outlier detection



relative distance of points from nearest centroid

Cluster-based outlier detection

Eliminate object(s) to improve objective function.

1) Form initial set of clusters.
2) Remove the object which most improves objective function.
3) Repeat step 2) until …

Cluster-based outlier detection

Discard small clusters far from other clusters.

- Need to define thresholds for “small” and “far”.

Cluster-based outlier detection

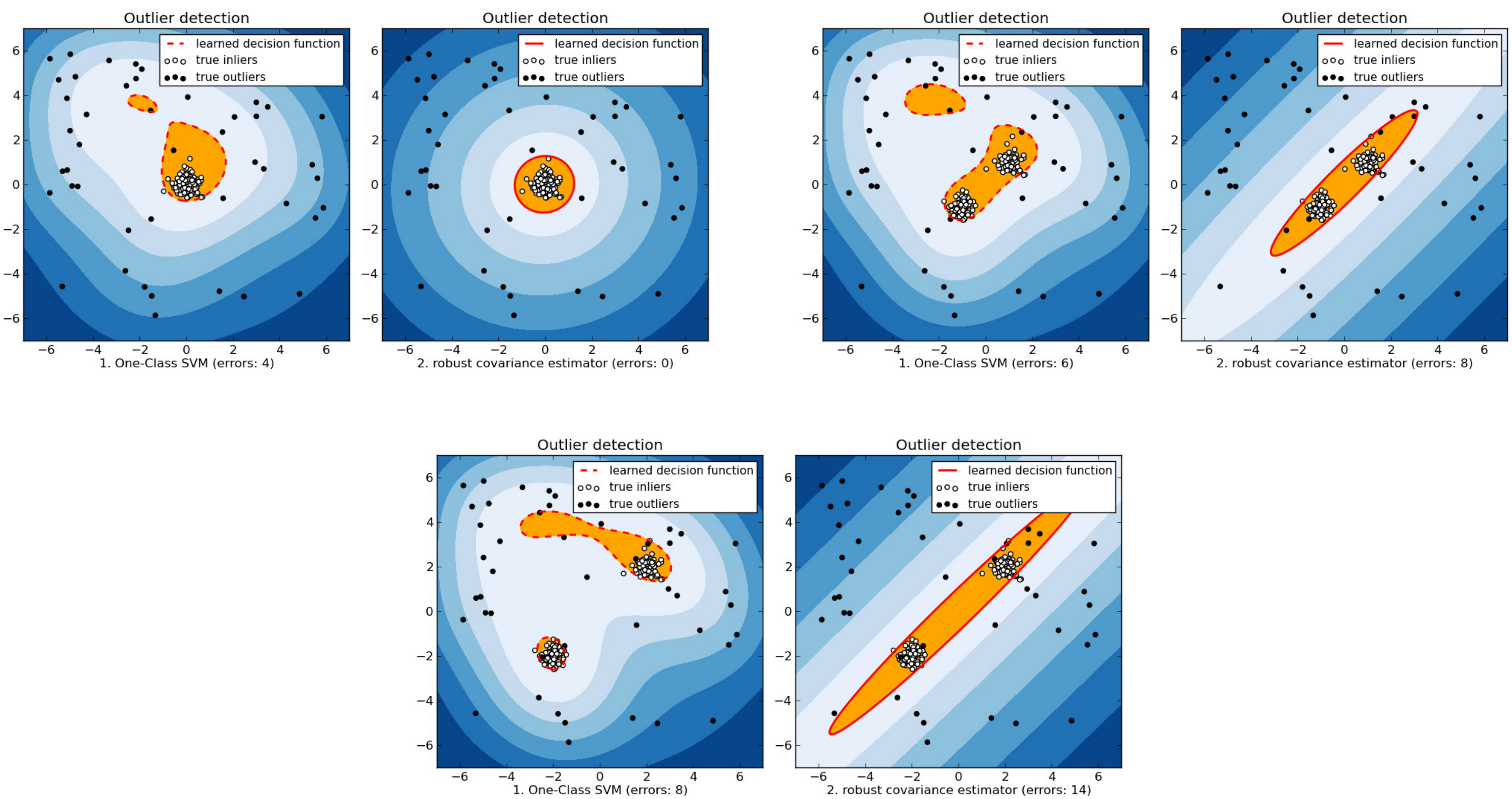
- Pro:
  - Some clustering techniques have $O(n)$ complexity.
  - Extends concept of outlier from single objects to groups of objects.
- Cons:
  - Requires thresholds for minimum size and distance.
  - Sensitive to number of clusters chosen.
  - Hard to associate outlier score with objects.
  - Outliers may affect initial formation of clusters.

One-class support vector machines

- Data is unlabelled, unlike usual SVM setting.

- Goal: find hyperplane (in higher-dimensional kernel space) which encloses as much data as possible with minimum volume.
  - Tradeoff between amount of data enclosed and tightness of enclosure; controlled by regularization of slack variables.

One-class SVM vs. Gaussian envelope



Images from http://scikit-learn.org/stable/modules/outlier_detection.html

One-class SVM demo

LIBSVM

http://www.csie.ntu.edu.tw/~cjlin/libsvm/

-s 2 -t 2 -g 50 -n 0.35

Anomaly detection on real network data

- Three groups of features
  - Basic features of individual TCP connections
    - source & destination IP *Features 1 & 2*
    - source & destination port *Features 3 & 4*
    - Protocol *Feature 5*
    - Duration *Feature 6*
    - Bytes per packets *Feature 7*
    - number of bytes *Feature 8*

dst …	*service …*	*flag*
h1	http	S0
h1	http	S0
h1	http	S0
h2	http	S0
h4	http	S0
h2	ftp	S0

existing features useless

syn flood

normal

dst …	*service …*	*flag*	*%S0*
h1	http	S0	70
h1	http	S0	72
h1	http	S0	75
h2	http	S0	0
h4	http	S0	0
h2	ftp	S0	0

construct features with high information gain

  - **Time based features**
    - For the same source (destination) IP address, number of unique destination (source) IP addresses inside the network *in last T seconds – Features 9 (13)*
    - Number of connections from source (destination) IP to the same destination (source) port *in last T seconds – Features 11 (15)*
  - Connection based features
    - For the same source (destination) IP address, number of unique destination (source) IP addresses inside the network *in last N connections - Features 10 (14)*
    - Number of connections from source (destination) IP to the same destination (source) port *in last N connections - Features 12 (16)*

Typical anomaly detection output

score	srcIP	sPort	dstIP	dPort	protocol	flags	packets	bytes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
37674.69	63.150.X.253	1161	128.101.X.29	1434	17	16	[0,2)	[0,1829)	0	0	0	0	0	0	0	0	0.81	0	0.59	0	0	0	0	0
26676.62	63.150.X.253	1161	160.94.X.134	1434	17	16	[0,2)	[0,1829)	0	0	0	0	0	0	0	0	0.81	0	0.59	0	0	0	0	0
24323.55	63.150.X.253	1161	128.101.X.185	1434	17	16	[0,2)	[0,1829)	0	0	0	0	0	0	0	0	0.81	0	0.58	0	0	0	0	0
21169.49	63.150.X.253	1161	160.94.X.71	1434	17	16	[0,2)	[0,1829)	0	0	0	0	0	0	0	0	0.81	0	0.58	0	0	0	0	0
19525.31	63.150.X.253	1161	160.94.X.19	1434	17	16	[0,2)	[0,1829)	0	0	0	0	0	0	0	0	0.81	0	0.58	0	0	0	0	0
19235.39	63.150.X.253	1161	160.94.X.80	1434	17	16	[0,2)	[0,1829)	0	0	0	0	0	0	0	0	0.81	0	0.58	0	0	0	0	0
17679.1	63.150.X.253	1161	160.94.X.220	1434	17	16	[0,2)	[0,1829)	0	0	0	0	0	0	0	0	0.81	0	0.58	0	0	0	0	0
8183.58	63.150.X.253	1161	128.101.X.108	1434	17	16	[0,2)	[0,1829)	0	0	0	0	0	0	0	0	0.82	0	0.58	0	0	0	0	0
7142.98	63.150.X.253	1161	128.101.X.223	1434	17	16	[0,2)	[0,1829)	0	0	0	0	0	0	0	0	0.82	0	0.57	0	0	0	0	0
5139.01	63.150.X.253	1161	128.101.X.142	1434	17	16	[0,2)	[0,1829)	0	0	0	0	0	0	0	0	0.82	0	0.57	0	0	0	0	0
4048.49	142.150.Y.101	0	128.101.X.127	2048	1	16	[2,4)	[0,1829)	0	0	0	0	0	0	0	0	0.83	0	0.56	0	0	0	0	0
4008.35	200.250.Z.20	27016	128.101.X.116	4629	17	16	[2,4)	[0,1829)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
3657.23	202.175.Z.237	27016	128.101.X.116	4148	17	16	[2,4)	[0,1829)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
3450.9	63.150.X.253	1161	128.101.X.62	1434	17	16	[0,2)	[0,1829)	0	0	0	0	0	0	0	0	0.82	0	0.57	0	0	0	0	0
3327.98	63.150.X.253	1161	160.94.X.223	1434	17	16	[0,2)	[0,1829)	0	0	0	0	0	0	0	0	0.82	0	0.57	0	0	0	0	0
2796.13	63.150.X.253	1161	128.101.X.241	1434	17	16	[0,2)	[0,1829)	0	0	0	0	0	0	0	0	0.82	0	0.57	0	0	0	0	0
2693.88	142.150.Y.101	0	128.101.X.168	2048	1	16	[2,4)	[0,1829)	0	0	0	0	0	0	0	0	0.83	0	0.56	0	0	0	0	0
2683.05	63.150.X.253	1161	160.94.X.43	1434	17	16	[0,2)	[0,1829)	0	0	0	0	0	0	0	0	0.82	0	0.57	0	0	0	0	0
2444.16	142.150.Y.236	0	128.101.X.240	2048	1	16	[2,4)	[0,1829)	0	0	0	0	0	0	0	0	0.83	0	0.56	0	0	0	0	0
2385.42	142.150.Y.101	0	128.101.X.45	2048	1	16	[0,2)	[0,1829)	0	0	0	0	0	0	0	0	0.83	0	0.56	0	0	0	0	0
2114.41	63.150.X.253	1161	160.94.X.183	1434	17	16	[0,2)	[0,1829)	0	0	0	0	0	0	0	0	0.82	0	0.57	0	0	0	0	0
2057.15	142.150.Y.101	0	128.101.X.161	2048	1	16	[0,2)	[0,1829)	0	0	0	0	0	0	0	0	0.83	0	0.56	0	0	0	0	0
1919.54	142.150.Y.101	0	128.101.X.99	2048	1	16	[2,4)	[0,1829)	0	0	0	0	0	0	0	0	0.83	0	0.56	0	0	0	0	0
1634.38	142.150.Y.101	0	128.101.X.219	2048	1	16	[2,4)	[0,1829)	0	0	0	0	0	0	0	0	0.83	0	0.56	0	0	0	0	0
1596.26	63.150.X.253	1161	128.101.X.160	1434	17	16	[0,2)	[0,1829)	0	0	0	0	0	0	0	0	0.82	0	0.57	0	0	0	0	0
1513.96	142.150.Y.107	0	128.101.X.2	2048	1	16	[0,2)	[0,1829)	0	0	0	0	0	0	0	0	0.83	0	0.56	0	0	0	0	0
1389.09	63.150.X.253	1161	128.101.X.30	1434	17	16	[0,2)	[0,1829)	0	0	0	0	0	0	0	0	0.82	0	0.57	0	0	0	0	0
1315.88	63.150.X.253	1161	128.101.X.40	1434	17	16	[0,2)	[0,1829)	0	0	0	0	0	0	0	0	0.82	0	0.57	0	0	0	0	0
1279.75	142.150.Y.103	0	128.101.X.202	2048	1	16	[0,2)	[0,1829)	0	0	0	0	0	0	0	0	0.83	0	0.56	0	0	0	0	0
1237.97	63.150.X.253	1161	160.94.X.32	1434	17	16	[0,2)	[0,1829)	0	0	0	0	0	0	0	0	0.83	0	0.56	0	0	0	0	0
1180.82	63.150.X.253	1161	128.101.X.61	1434	17	16	[0,2)	[0,1829)	0	0	0	0	0	0	0	0	0.83	0	0.56	0	0	0	0	0

- Anomalous connections that correspond to the “slammer” worm
- Anomalous connections that correspond to the ping scan
- Connections corresponding to Univ. Minnesota machines connecting to “half-life” game servers

Real-world issues in anomaly detection

- Data often streaming, not static
  - Credit card transactions
- Anomalies can be *bursty*
  - Network intrusions

Quote of the day

An excerpt from advice given by a machine learning veteran on StackOverflow:

“ … you are training and testing on the same data. A kitten dies every time this happens.”