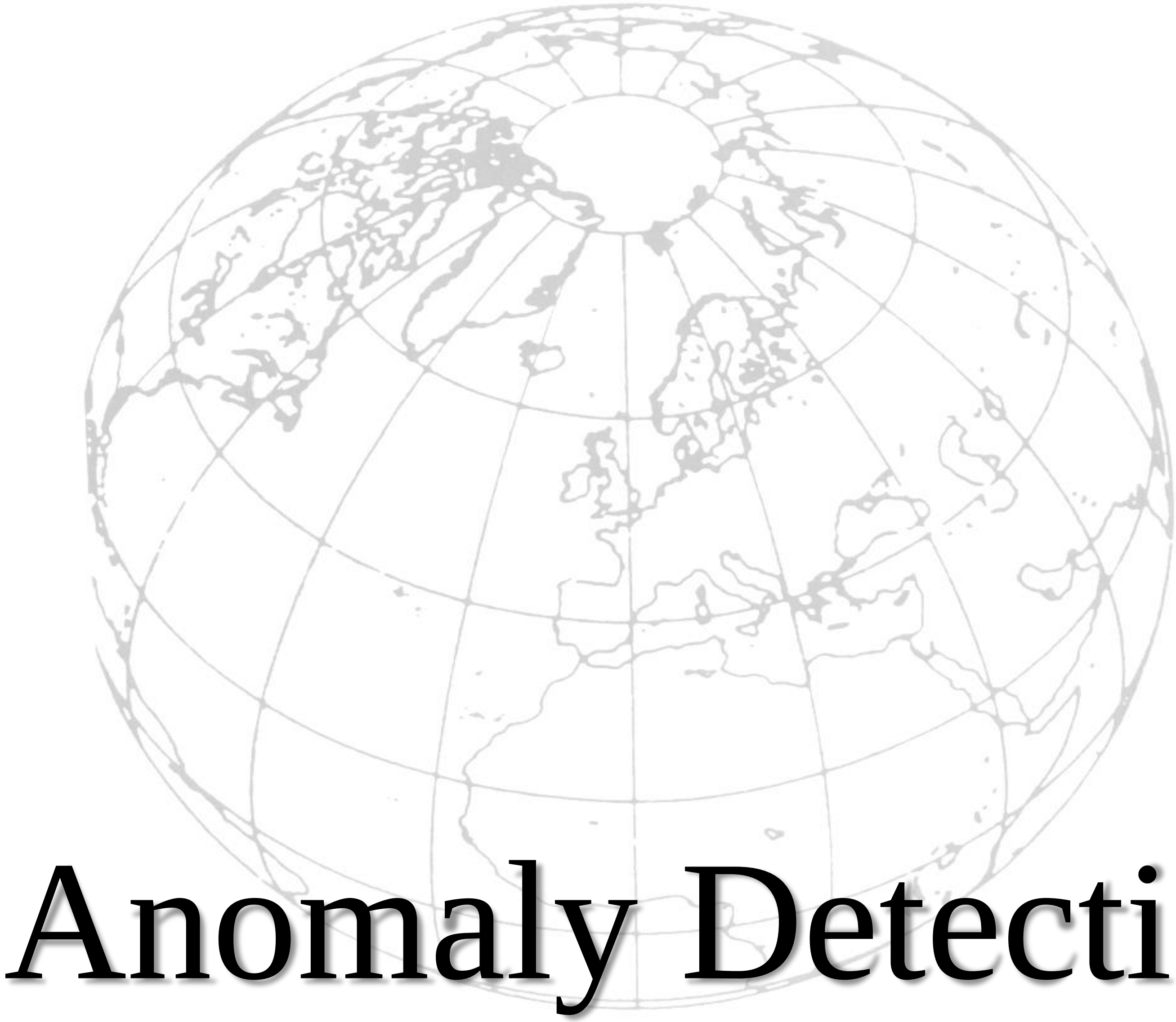


Anomaly Detection

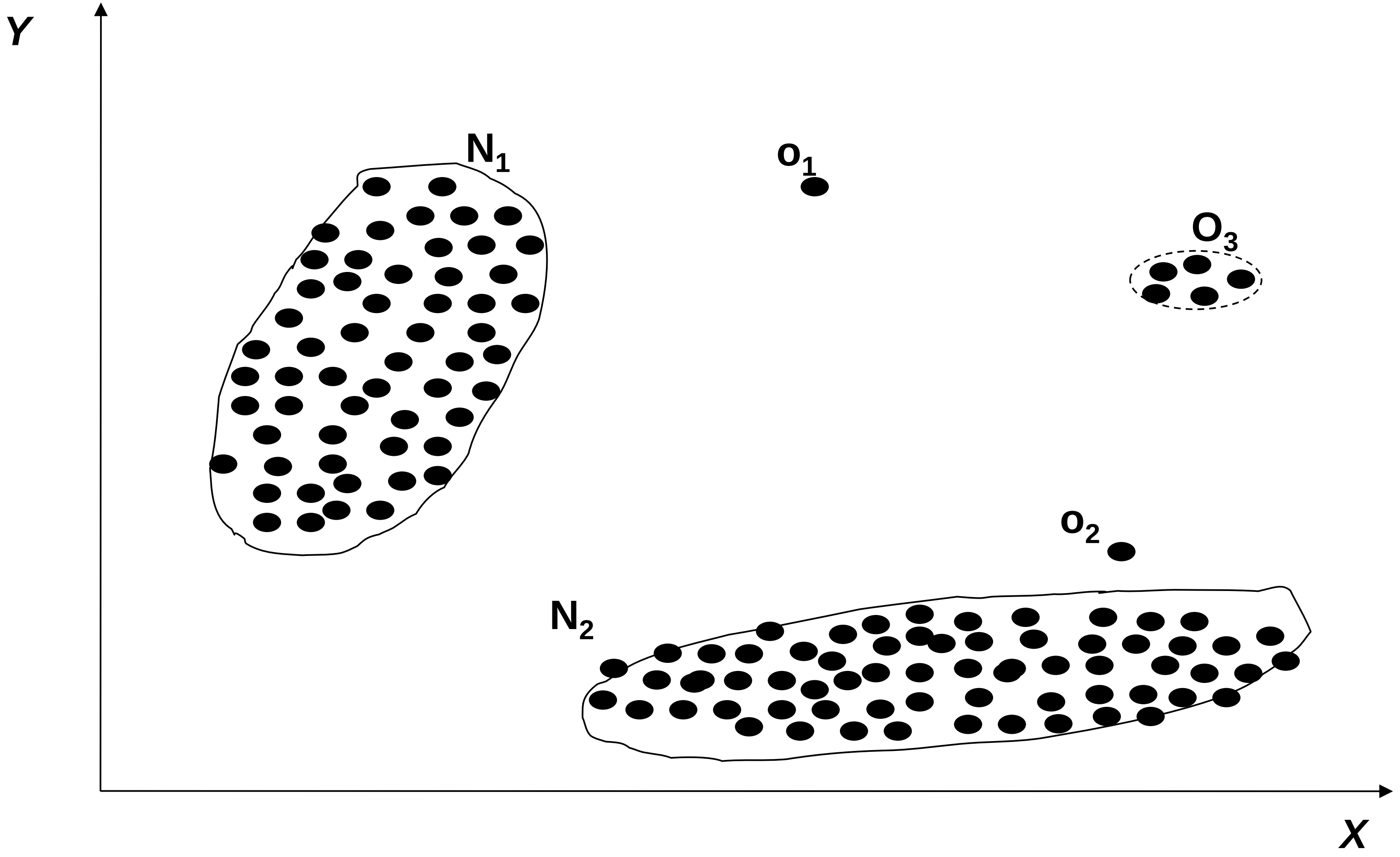
Dr. Cahit Karakuş

What are Anomalies?

- Anomali, beklenen davranışa uymayan verilerdeki bir kalıptır.Aykırı değerler, istisnalar, özellikler, sürpriz vb. olarak da anılır.Anomaliler, önemli (genellikle kritik) gerçek hayattaki varlıklara dönüşür
  - Cyber intrusions
  - Credit card fraud

Simple Example

- N_1 and N_2 are regions of normal behavior
- Points o_1 and o_2 are anomalies
- Points in region O_3 are anomalies



Related problems

- Rare Class Mining
- Chance discovery
- Novelty Detection
- Exception Mining
- Noise Removal
- Black Swan

Temel Zorluklar

- Temsili bir normal bölge tanımlamak zordur
- Normal ve aykırı davranış arasındaki sınır genellikle kesin değildir.
- Bir aykırı değerin tam kavramı, farklı uygulama etki alanları için farklıdır
- Eğitim/doğrulama için etiketlenmiş verilerin kullanılabilirliği
- Kötü niyetli düşmanlar
- Veriler gürültü içerebilir
- Normal davranış gelişmeye devam ediyor

Anormallik Tespit Probleminin Yönleri

- Girdi verilerinin doğası
- Denetimin mevcudiyeti
- Anomali türü: nokta, bağlamsal, yapısal
- Anormallik algılama çıktısı
- Anormallik algılama tekniklerinin değerlendirilmesi

Data Labels

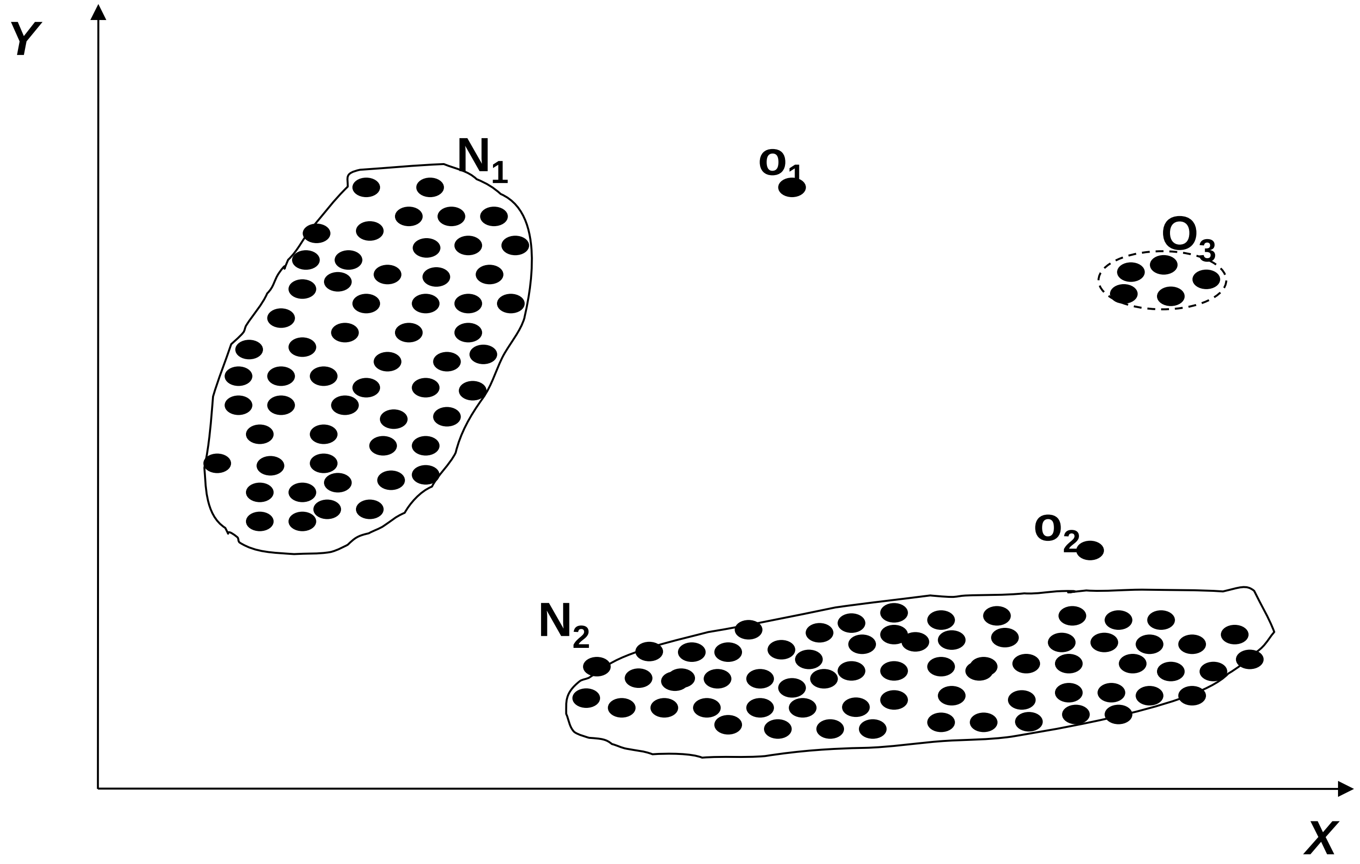
- Supervised Anomaly Detection
  - Labels available for both normal data and anomalies
  - Similar to rare class mining
- Semi-supervised Anomaly Detection
  - Labels available only for normal data
- Unsupervised Anomaly Detection
  - No labels assumed
  - Based on the assumption that anomalies are very rare compared to normal data

Type of Anomaly

- Nokta Anomalileri
- Bağlamsal Anomaliler
- Kolektif Anomaliler

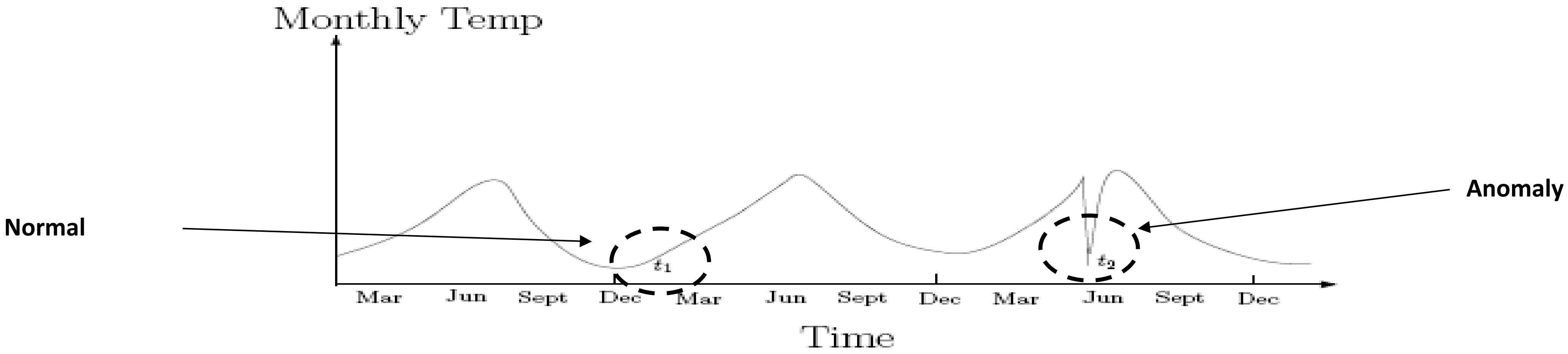
Point Anomalies

- An individual data instance is anomalous w.r.t. the data



Contextual Anomalies

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

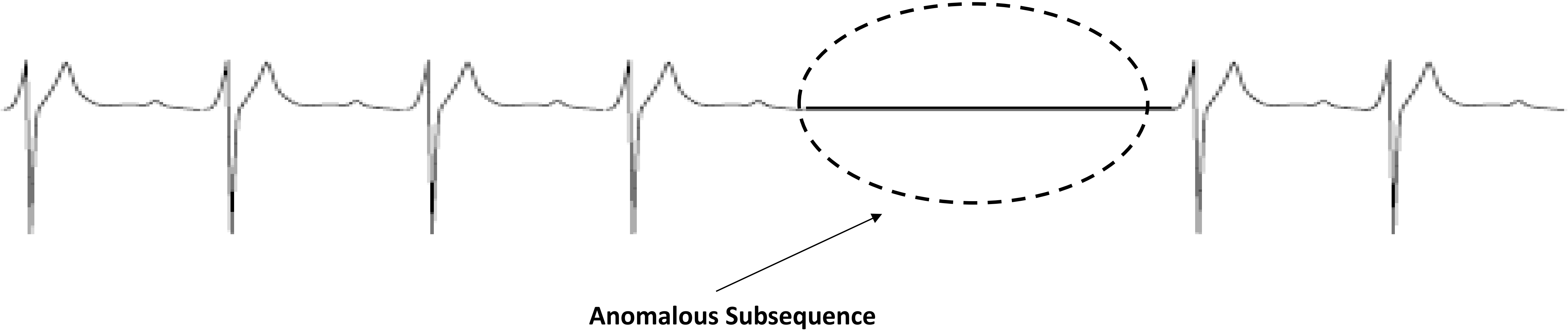


Normal

Anomaly

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



Output of Anomaly Detection

- Label
  - Each test instance is given a *normal* or *anomaly* label
  - This is especially true of classification-based approaches
- Score
  - Each test instance is assigned an anomaly score
    - Allows the output to be ranked
    - Requires an additional threshold parameter

Evaluation of Anomaly Detection – F-value

- Accuracy is not sufficient metric for evaluation
  - Example: network traffic data set with 99.9% of normal data and 0.1% of intrusions
  - Trivial classifier that labels everything with the normal class can achieve 99.9% accuracy !!!!!

Confusion matrix		Predicted class	
		NC	C
Actual class	NC	TN	FP
	C	FN	TP

anomaly class – C
normal class – NC

- **Focus on both recall and precision**
  - **Recall (R) = TP/(TP + FN)**
  - **Precision (P) = TP/(TP + FP)**
- **F – measure = 2*R*P/(R+P)**

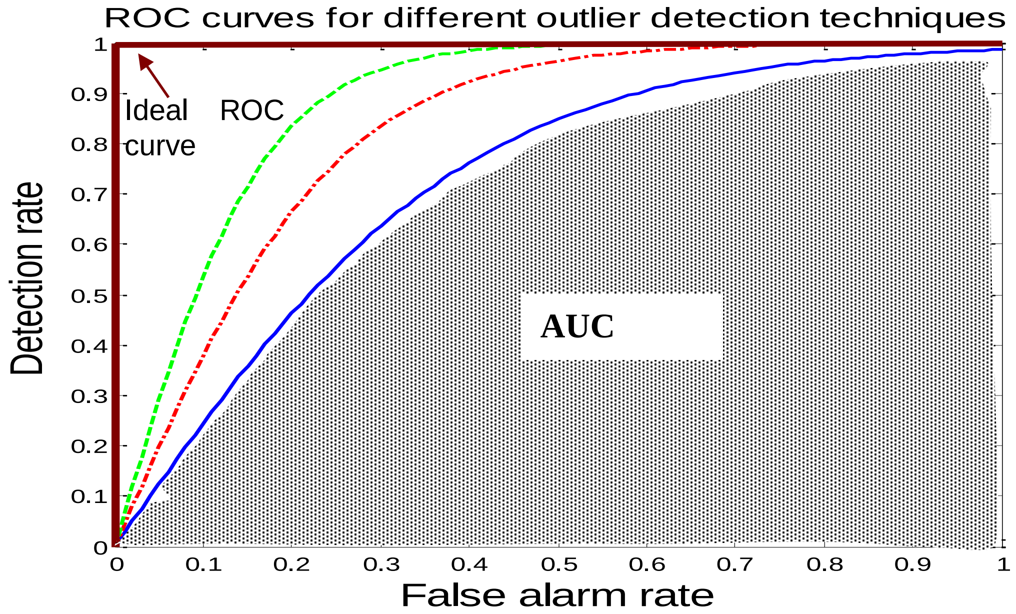
Evaluation of Outlier Detection – ROC & AUC

Confusion matrix		Predicted class	
		NC	C
Actual class	NC	TN	FP
	C	FN	TP

anomaly class – C

normal class – NC

- Standard measures for evaluating anomaly detection problems:
  - *Recall (Detection rate)* - ratio between the number of correctly detected anomalies and the total number of anomalies
  - *False alarm (false positive) rate* – ratio between the number of data records from normal class that are misclassified as anomalies and the total number of data records from normal class
  - *ROC Curve* is a trade-off between detection rate and false alarm rate
  - *Area under the ROC curve (AUC)* is computed using a trapezoid rule

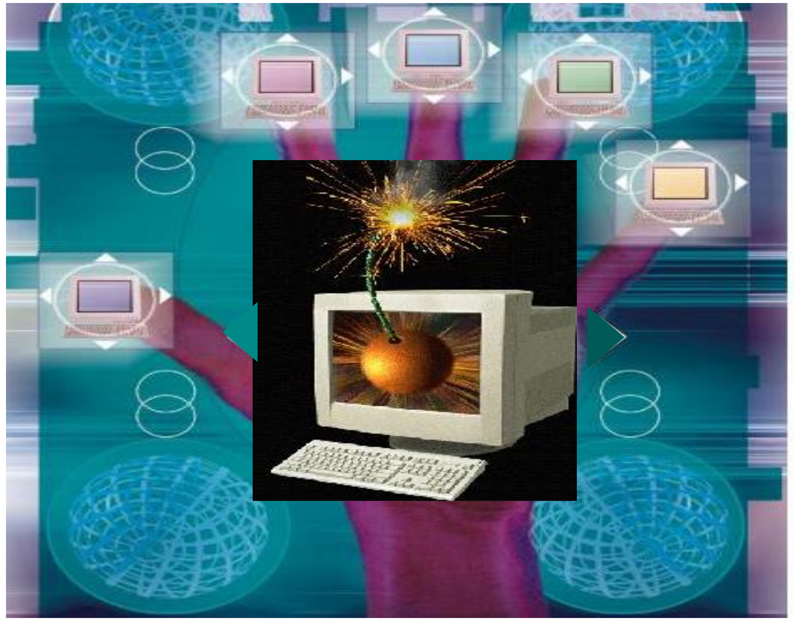


Applications of Anomaly Detection

- Network intrusion detection
- Insurance / Credit card fraud detection
- Healthcare Informatics / Medical diagnostics
- Industrial Damage Detection
- Image Processing / Video surveillance
- Novel Topic Detection in Text Mining
- …

Intrusion Detection

- Intrusion Detection:
  - Process of monitoring the events occurring in a computer system or network and analyzing them for intrusions
  - Intrusions are defined as attempts to bypass the security mechanisms of a computer or network
- Challenges
  - Traditional signature-based 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

- Fraud detection refers to detection of criminal activities occurring in commercial organizations
  - Malicious users might be the actual customers of the organization or might be posing as a customer (also known as 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

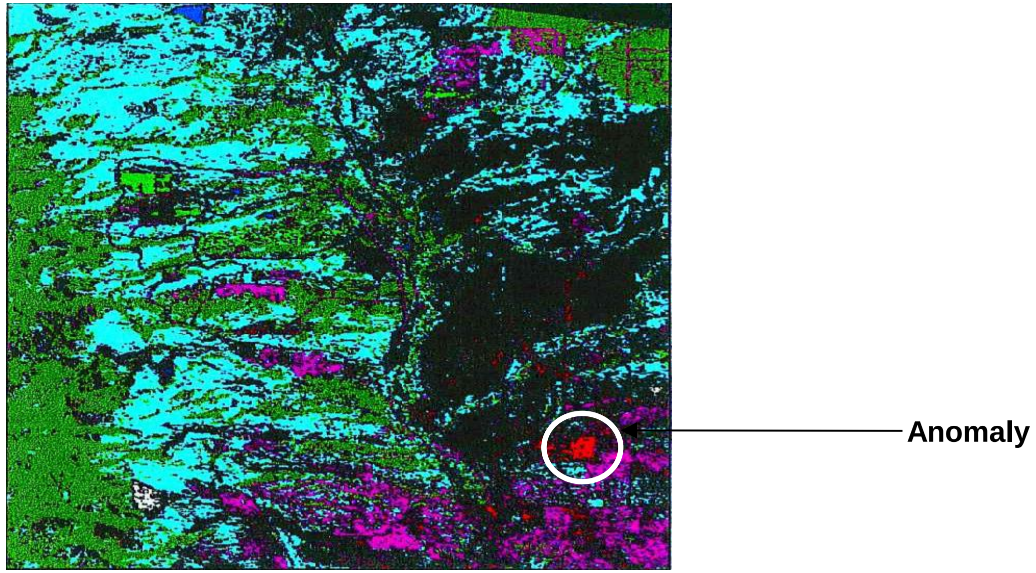
- Detect anomalous patient records
  - Indicate disease outbreaks, instrumentation errors, etc.
- Key Challenges
  - Only normal labels available
  - Misclassification cost is very high
  - Data can be complex: spatio-temporal

Industrial Damage Detection

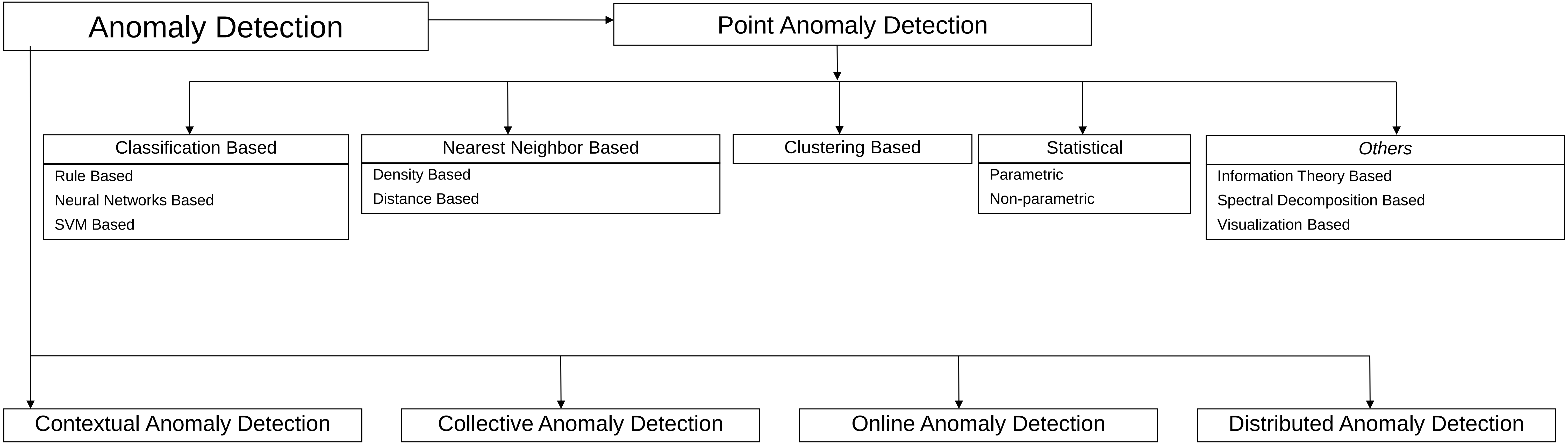
- Industrial damage detection refers to detection of different faults and failures in complex industrial systems, structural damages, intrusions in electronic security systems, suspicious events in video surveillance, abnormal energy consumption, etc.
  - Example: Aircraft Safety
    - Anomalous Aircraft (Engine) / Fleet Usage
    - Anomalies in engine combustion data
    - Total aircraft health and usage management
- Key Challenges
  - Data is extremely huge, noisy and unlabelled
  - Most of applications exhibit temporal behavior
  - Detecting 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



Taxonomy*



Anomaly Detection

Point Anomaly Detection

Classification Based
- Rule Based
- Neural Networks Based
- SVM Based

Nearest Neighbor Based
- Density Based
- Distance Based

Clustering Based

Statistical
- Parametric
- Non-parametric

Others
- Information Theory Based
- Spectral Decomposition Based
- Visualization Based

Contextual Anomaly Detection

Collective Anomaly Detection

Online Anomaly Detection

Distributed Anomaly Detection

Classification Based Techniques

- Main idea: build a classification model for normal (and anomalous (rare)) events based on labeled training data, and use it to classify each new unseen event
- Classification models must be able to handle skewed (imbalanced) class distributions
- Categories:
  - *Supervised classification techniques*
    - Require knowledge of both ***normal*** and ***anomaly*** class
    - Build classifier to distinguish between normal and known anomalies
  - *Semi-supervised classification techniques*
    - Require knowledge of ***normal*** class only!
    - Use modified classification model to learn the normal behavior and then detect any deviations from normal behavior as anomalous

Classification Based Techniques

- Advantages:
  - *Supervised classification techniques*
    - Models that can be easily understood
    - High accuracy in detecting many kinds of known anomalies
  - *Semi-supervised classification techniques*
    - Models that can be easily understood
    - Normal behavior can be accurately learned
- Drawbacks:
  - *Supervised classification techniques*
    - Require both labels from both normal and anomaly class
    - Cannot detect unknown and emerging anomalies
  - *Semi-supervised classification techniques*
    - Require labels from normal class
    - Possible high false alarm rate - previously unseen (yet legitimate) data records may be recognized as anomalies

Supervised Classification Techniques

- Manipulating data records (oversampling / undersampling / generating artificial examples)
- Rule based techniques
- Model based techniques
  - Neural network based approaches
  - Support Vector machines (SVM) based approaches
  - Bayesian networks based approaches
- Cost-sensitive classification techniques
- Ensemble based algorithms (SMOTEBoost, RareBoost

Manipulating Data Records

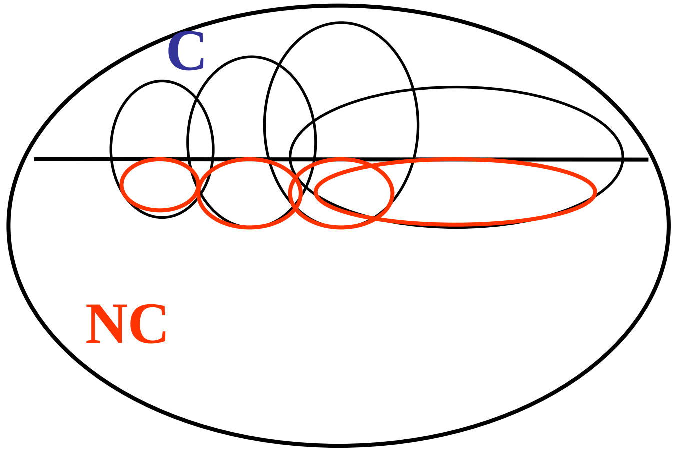
- **Over-sampling the rare class** [Ling98]
  - Make the duplicates of the rare events until the data set contains as many examples as the majority class => balance the classes
  - Does not increase information but increase misclassification cost
- **Down-sizing (undersampling) the majority class** [Kubat97]
  - Sample the data records from majority class (Randomly, Near miss examples, Examples far from minority class examples (far from decision boundaries)
  - Introduce sampled data records into the original data set instead of original data records from the majority class
  - Usually results in a general loss of information and overly general rules
- **Generating artificial anomalies**
  - SMOTE (Synthetic Minority Over-sampling TEchnique) [Chawla02] - new rare class examples are generated inside the regions of existing rare class examples
  - Artificial anomalies are generated around the edges of the sparsely populated data regions [Fan01]
  - Classify synthetic outliers vs. real normal data using active learning [Abe06]

Rule Based Techniques

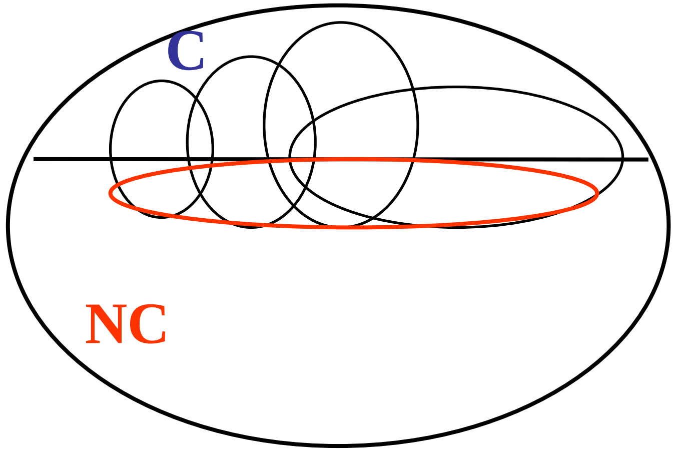
- **Creating new rule based algorithms (PN-rule, CREDOS)**
- **Adapting existing rule based techniques**
  - Robust C4.5 algorithm [John95]
  - Adapting multi-class classification methods to single-class classification problem
- **Association rules**
  - Rules with support higher than pre specified threshold may characterize normal behavior [Barbara01, Otey03]
  - Anomalous data record occurs in fewer frequent itemsets compared to normal data record [He04]
  - Frequent episodes for describing temporal normal behavior [Lee00,Qin04]
- **Case specific feature/rule weighting**
  - Case specific feature weighting [Cardey97] - Decision tree learning, where for each rare class test example replace global weight vector with dynamically generated weight vector that depends on the path taken by that example
  - Case specific rule weighting [Grzymala00] - LERS (Learning from Examples based on Rough Sets) algorithm increases the rule strength for all rules describing the rare class

New Rule-based Algorithms: PN-rule Learning*

- ***P-phase*:**
  - cover most of the positive examples with high support
  - seek good recall
- ***N-phase*:**
  - remove FP from examples covered in P-phase
  - N-rules give high accuracy and significant support



Existing techniques can possibly learn erroneous small signatures for absence of C



PNrule can learn strong signatures for presence of NC in *N-phase*

New Rule-based Algorithms: CREDOS*

- Ripple Down Rules (RDRs) offer a unique tree based representation that generalizes the decision tree and DNF rule list models and specializes a generic form of multi-phase PNrule model
- First use ripple down rules to overfit the training data
  - Generate a binary tree where each node is characterized by the rule R_h, a default class and links to two child subtrees
  - Grow the RDS structure in a recursive manner
  - Induces rules at a node
- Prune the structure to improve generalization
  - Different mechanism from decision trees

Using Neural Networks

- Multi-layer Perceptrons
  - Measuring the activation of output nodes [Augusteijn02]
  - Extending the learning beyond decision boundaries
    - Equivalent error bars as a measure of confidence for classification [Sykacek97]
    - Creating hyper-planes for separating between various classes, but also to have flexible boundaries where points far from them are outliers [Vasconcelos95]
- Auto-associative neural networks
  - Replicator NNs [Hawkins02]
  - Hopfield networks [Jagota91, Crook01]
- Adaptive Resonance Theory based [Dasgupta00, Caudel93]
- Radial Basis Functions based
  - Adding reverse connections from output to central layer allows each neuron to have associated normal distribution, and any new instance that does not fit any of these distributions is an anomaly [Albrecht00, Li02]
- Oscillatory networks
  - Relaxation time of oscillatory NNs is used as a criterion for novelty detection when a new instance is presented [Ho98, Borisyuk00]

Using Support Vector Machines

- SVM Classifiers [Steinwart05,Mukkamala02]
- Main idea [Steinwart05] :
  - Normal data records belong to high density data regions
  - Anomalies belong to low density data regions
  - Use unsupervised approach to learn high density and low density data regions
  - Use SVM to classify data density level
- Main idea: [Mukkamala02]
  - Data records are labeled (normal network behavior vs. intrusive)
  - Use standard SVM for classification

Using Bayesian Networks

- Typical Bayesian networks
  - Aggregates information from different variables and provide an estimate of the expectancy that event belong to one of normal or anomalous classes [Baker99, Das07]
- Naïve Bayesian classifiers
  - Incorporate prior probabilities into a reasoning model that classifies an event as normal or anomalous based on observed properties of the event and prior probabilities [Sebyala02, Kruegel03]
- Pseudo-Bayes estimators [Barbara01]
  - I stage: learn prior and posterior of unseen anomalies from the training data
  - II stage: use Naive Bayes classifier to classify the instances into normal instances, known anomalies and new anomalies

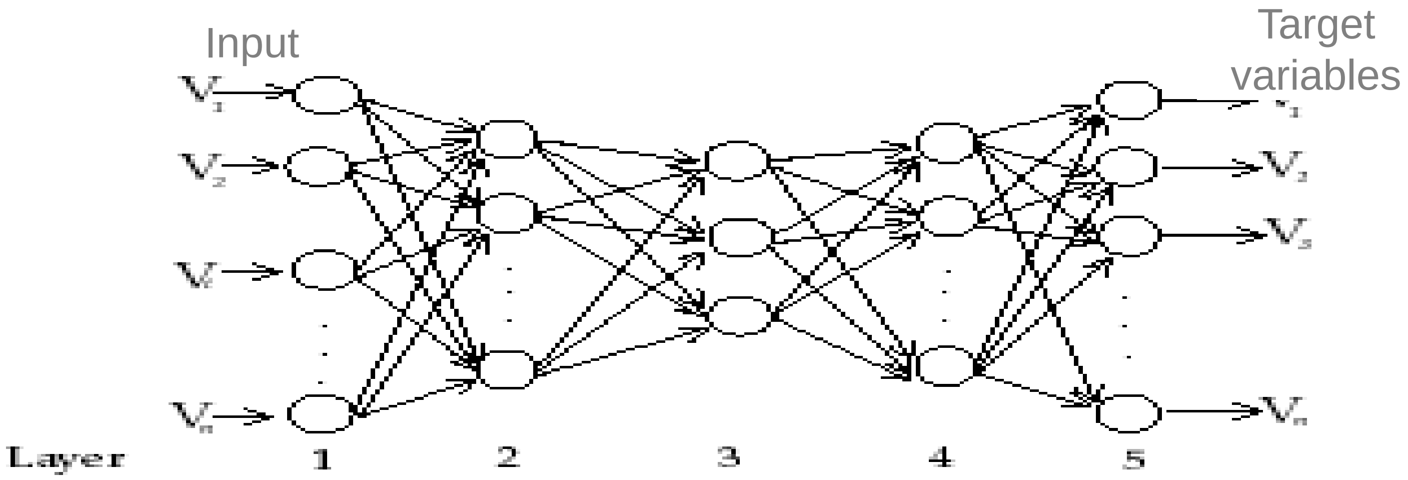
Semi-supervised Classification Techniques

- Use modified classification model to learn the normal behavior and then detect any deviations from normal behavior as anomalous

- Recent approaches:
  - Neural network based approaches
  - Support Vector machines (SVM) based approaches
  - Markov model based approaches
  - Rule-based approaches

Using Replicator Neural Networks*

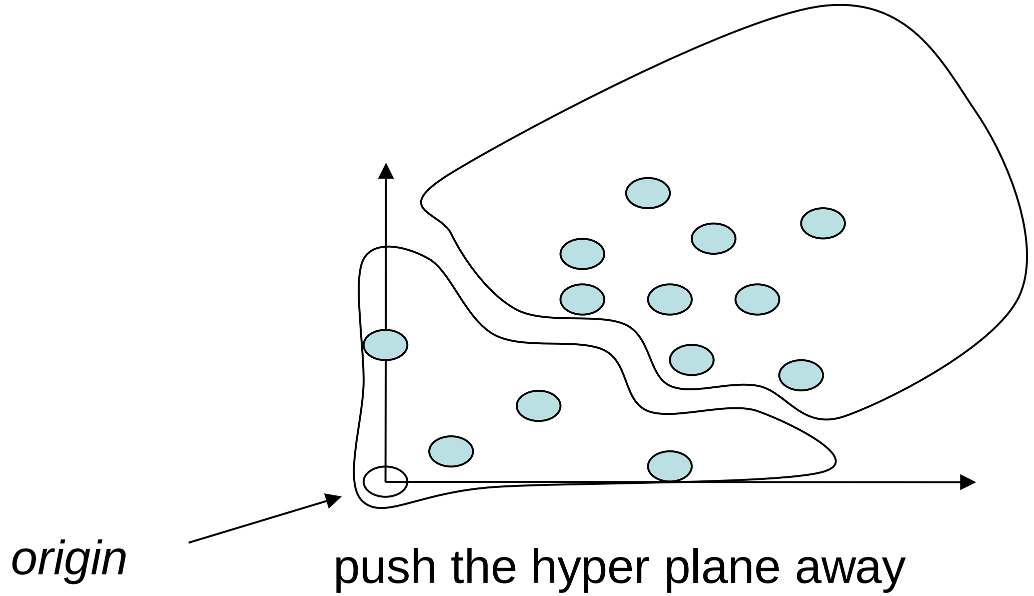
- Use a replicator 4-layer feed-forward neural network (RNN) with the same number of input and output nodes
- Input variables are the output variables so that RNN forms a compressed model of the data during training
- A measure of outlyingness is the reconstruction error of individual data points.



* S. Hawkins, et al. Outlier detection using replicator neural networks, DaWaK02 2002.

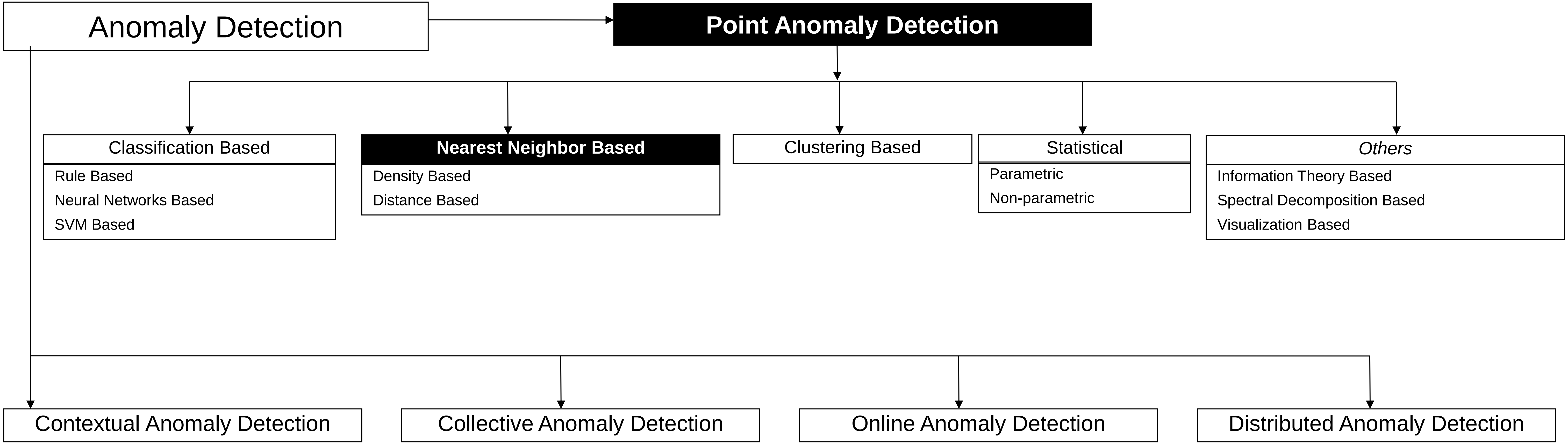
Using Support Vector Machines

- Converting into one class classification problem
  - Separate the entire set of training data from the origin, i.e. to find a small region where most of the data lies and label data points in this region as one class [Ratsch02, Tax01, Eskin02, Lazarevic03]
    - Parameters
      - Expected number of outliers
      - Variance of rbf kernel (As the variance of the rbf kernel gets smaller, the number of support vectors is larger and the separating surface gets more complex)
  - Separate regions containing data from the regions containing no data [Scholkopf99]



push the hyper plane away from origin as much as possible

Taxonomy



Anomaly Detection

Point Anomaly Detection

Classification Based

Rule Based

Neural Networks Based

SVM Based

Nearest Neighbor Based

Density Based

Distance Based

Clustering Based

Statistical

Parametric

Non-parametric

Others

Information Theory Based

Spectral Decomposition Based

Visualization Based

Contextual Anomaly Detection

Collective Anomaly Detection

Online Anomaly Detection

Distributed Anomaly Detection

Nearest Neighbor Based Techniques

- *Key assumption*: normal points have close neighbors while anomalies are located far from other points
- General two-step approach
  1. Compute neighborhood for each data record
  2. Analyze the neighborhood to determine whether data record is anomaly or not
- Categories:
  - Distance based methods
    - Anomalies are data points most distant from other points
  - Density based methods
    - Anomalies are data points in low density regions

Nearest Neighbor Based Techniques

- Advantage
  - Can be used in unsupervised or semi-supervised setting (do not make any assumptions about data distribution)
- Drawbacks
  - If normal points do not have sufficient number of neighbors the techniques may fail
  - Computationally expensive
  - In high dimensional spaces, data is sparse and the concept of similarity may not be meaningful anymore. Due to the sparseness, distances between any two data records may become quite similar => Each data record may be considered as potential outlier!

Nearest Neighbor Based Techniques

- Distance based approaches
  - A point *O* in a dataset is an *DB*(*p*, *d*) outlier if at least fraction *p* of the points in the data set lies greater than distance *d* from the point *O**
- Density based approaches
  - Compute local densities of particular regions and declare instances in low density regions as potential anomalies
  - Approaches
    - Local Outlier Factor (LOF)
    - Connectivity Outlier Factor (COF
    - Multi-Granularity Deviation Factor (MDEF)

Distance based Outlier Detection

- *Nearest Neighbor (NN) approach*[*,**]
  - For each data point *d* compute the distance to the *k-th* nearest neighbor 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 more sparse neighborhoods
  - Usually data points that have top *n*% distance d_k are identified as outliers
    - *n* – user parameter
  - Not suitable for datasets that have modes with varying density

Local Outlier Factor (LOF)*

- For each data point *q* compute the distance to the *k*-th nearest neighbor (*k-distance*)

•Compute *reachability distance* (*reach-dist*) for each data example *q* with respect to data example *p* as:

$$reach\text{-}dist(q, p) = \max\{k\text{-}distance(p), d(q,p)\}$$

•Compute *local reachability density* (*lrd*) of data example *q* as inverse of the average reachabaility distance based on the *MinPts* nearest neighbors of data example *q*

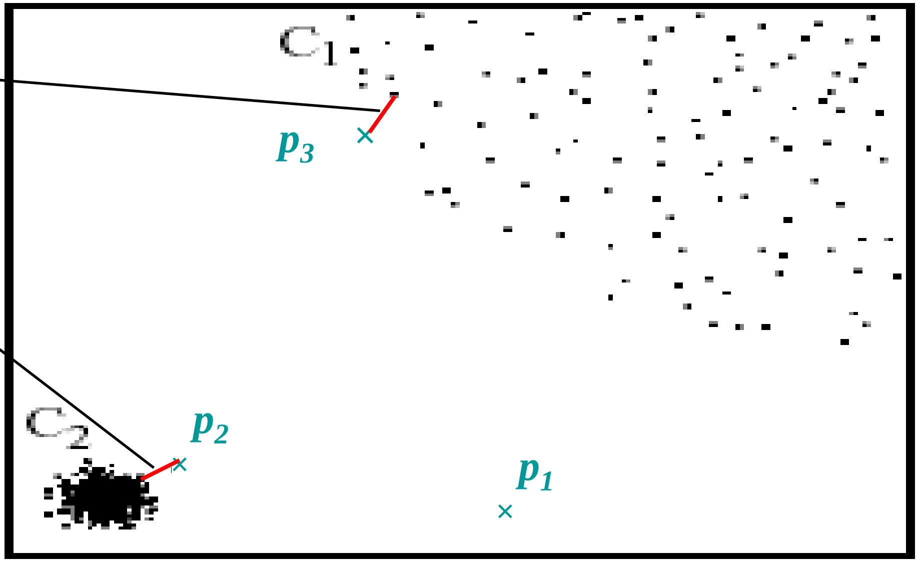
$$lrd(q) = \frac{MinPts}{\sum_{p} reach\_dist_{MinPts}(q, p)}$$

•Compaute *LOF*(*q*) as ratio of average local reachability density of *q*'s *k*-nearest neighbors and local reachability density of the data record *q*

$$LOF(q) = \frac{1}{MinPts} \cdot \sum_{p} \frac{lrd(p)}{lrd(q)}$$

Advantages of Density based Techniques

- *Local Outlier Factor (LOF) approach*
  - Example:



Distance from p_3 to nearest neighbor

Distance from p_2 to nearest neighbor

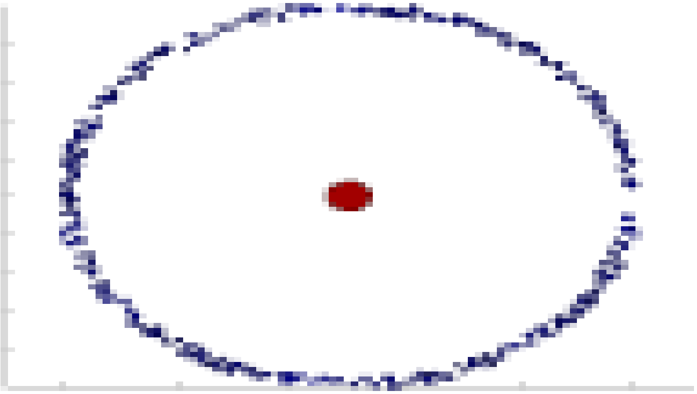
In the *NN* approach, p_2 is not considered as outlier, while the *LOF* approach find both p_1 and p_2 as outliers

NN approach may consider p_3 as outlier, but LOF approach does not

Connectivity Outlier Factor (COF)*

- Outliers are points *p* where average chaining distance $ac\text{-}dist_{kNN(p)}(p)$ is larger than the average chaining distance (*ac-dist*) of their k-nearest neighborhood kNN(p)
- Let p_2 is a point from G-{p_1} closest to p_1, let p_3 be a point from G-{p_1, p_2} closest to the set {p_1,p_2},…, let p_i be a point from G-{p_1,… p_{i-1}} closest to the set {p_1,…,p_{i-1}}, etc. o_i *is* a point from {p_1,… p_i} closest to p_{i+1}.

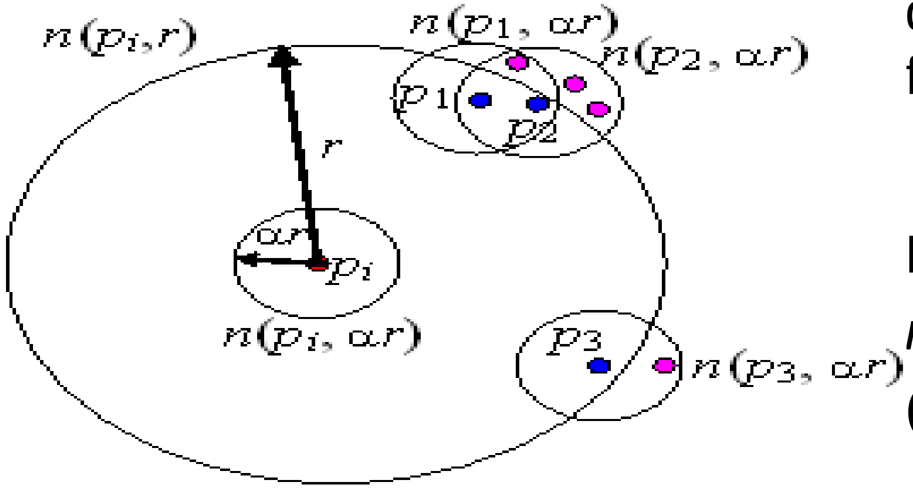
$$ac - dist_G(p_1) = \sum_{i=1}^{r-1} \frac{2(r-i)}{r(r-1)} dist(o_i, p_{i+1})$$



- *dist*(o_i, p_i*+1*) - the single linkage distance between sets {p_1,… p_i} and G-{p_1,… p_i}
- r = |G| is the size of the set G
- $ac\text{-}dist_G(p_1)$ is larger if $dist(o_i, p_{i+1})$ is large for small values of the index *i*, which corresponds to the sparser neighborhood of the point p_1.
- COF identifies outliers as points whose neighborhoods is sparser than the neighborhoods of their neighbors

Multi-Granularity Deviation Factor - LOCI*

- LOCI computes the neighborhood size (the number of neighbors) for each point and identifies as outliers points whose neighborhood size significantly vary with respect to the neighborhood size of their neighbors
- This approach not only finds outlying points but also outlying micro-clusters.
- LOCI algorithm provides LOCI plot which contains information such as inter cluster distance and cluster diameter



Outlier are samples p_i where for any $r \in [r_{min}, r_{max}]$, $n(p_i, \alpha \cdot r)$ significantly deviates from the distribution of values $n(p_j, \alpha \cdot r)$ associated with samples p_j from the r-neighborhood of p_i. Sample is outlier if:

$$n(p_i, \alpha r) < \hat{n}(p_i, r, \alpha) - k_\sigma \sigma_{\hat{n}}(p_i, r, \alpha)$$

Example:

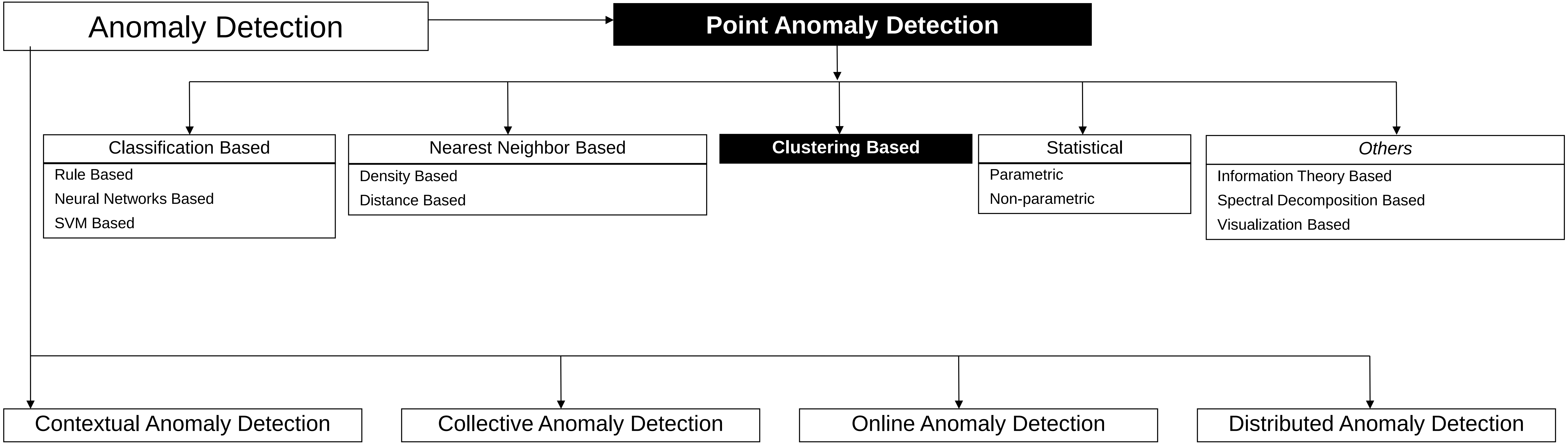
$n(p_i,r)=4$, $n(p_i,\alpha \cdot r)=1$, $n(p_1,\alpha \cdot r)=3$, $n(p_2,\alpha \cdot r)=5$, $n(p_3,\alpha \cdot r)=2$, =

(1+3+5+2) / 4 = 2.75,

; $\alpha = 1/4$. $\hat{n}(p_i, r, \alpha)$

$$\sigma_{\hat{n}}(p_i, r, \alpha) \approx 1.479$$

Taxonomy



Anomaly Detection

Point Anomaly Detection

Classification Based
- Rule Based
- Neural Networks Based
- SVM Based

Nearest Neighbor Based
- Density Based
- Distance Based

Clustering Based

Statistical
- Parametric
- Non-parametric

Others
- Information Theory Based
- Spectral Decomposition Based
- Visualization Based

Contextual Anomaly Detection

Collective Anomaly Detection

Online Anomaly Detection

Distributed Anomaly Detection

Clustering Based Techniques

- *Key assumption*: normal data records belong to large and dense clusters, while anomalies belong do not belong to any of the clusters or form very small clusters
- Categorization according to labels
  - Semi-supervised – cluster normal data to create modes of normal behavior. If a new instance does not belong to any of the clusters or it is not close to any cluster, is anomaly
  - Unsupervised – post-processing is needed after a clustering step to determine the size of the clusters and the distance from the clusters is required fro the point to be anomaly
- Anomalies detected using clustering based methods can be:
  - Data records that do not fit into any cluster (residuals from clustering)
  - Small clusters
  - Low density clusters or local anomalies (far from other points within the same cluster)

Clustering Based Techniques

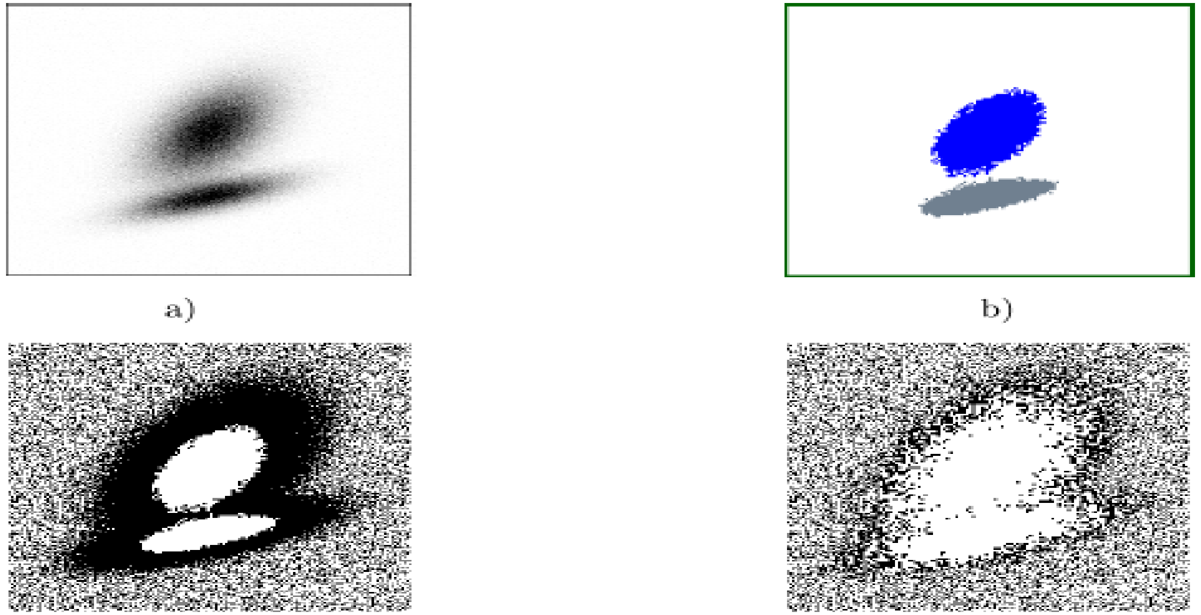
- Advantages:
  - No need to be supervised
  - Easily adaptable to on-line / incremental mode suitable for anomaly detection from temporal data
- Drawbacks
  - Computationally expensive
    - Using indexing structures (k-d tree, R* tree) may alleviate this problem
  - If normal points do not create any clusters the techniques may fail
  - In high dimensional spaces, data is sparse and distances between any two data records may become quite similar.
    - Clustering algorithms may not give any meaningful clusters

Simple Application of Clustering

- Radius ω of proximity is specified
- Two points x_1 and x_2 are "near" if $d(x_1, x_2) \leq \omega$
- Define $N(x)$ – number of points that are within ω of x
- Time Complexity $O(n^2) \Rightarrow$ approximation of the algorithm
- Fixed-width clustering is first applied
  - The first point is a center of a cluster
  - If every subsequent point is "near" add to a cluster
    - Otherwise create a new cluster
  - Approximate $N(x)$ with $N(c)$
  - Time Complexity – $O(cn)$, c - # of clusters
- Points in small clusters - anomalies

FindOut

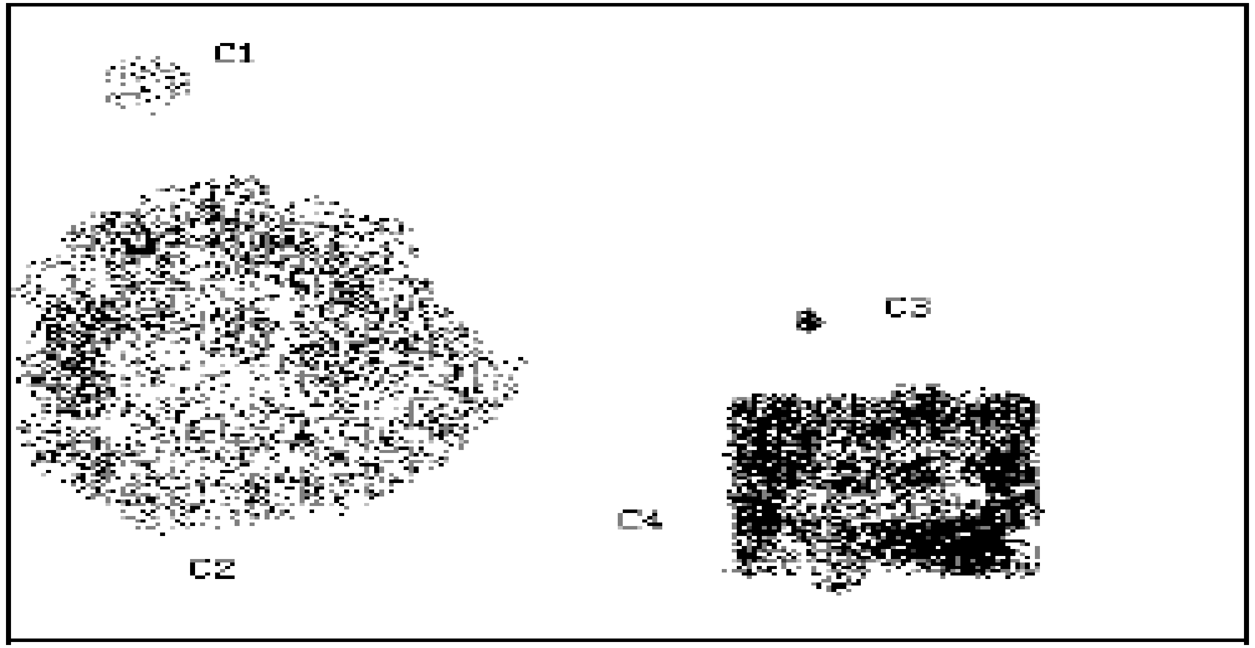
- FindOut algorithm* by-product of *WaveCluster*
- Main idea: Remove the clusters from original data and then identify the outliers
- Transform data into multidimensional signals using wavelet transformation
  - High frequency of the signals correspond to regions where is the rapid change of distribution – boundaries of the clusters
  - Low frequency parts correspond to the regions where the data is concentrated
- Remove these high and low frequency parts and all remaining points will be outliers



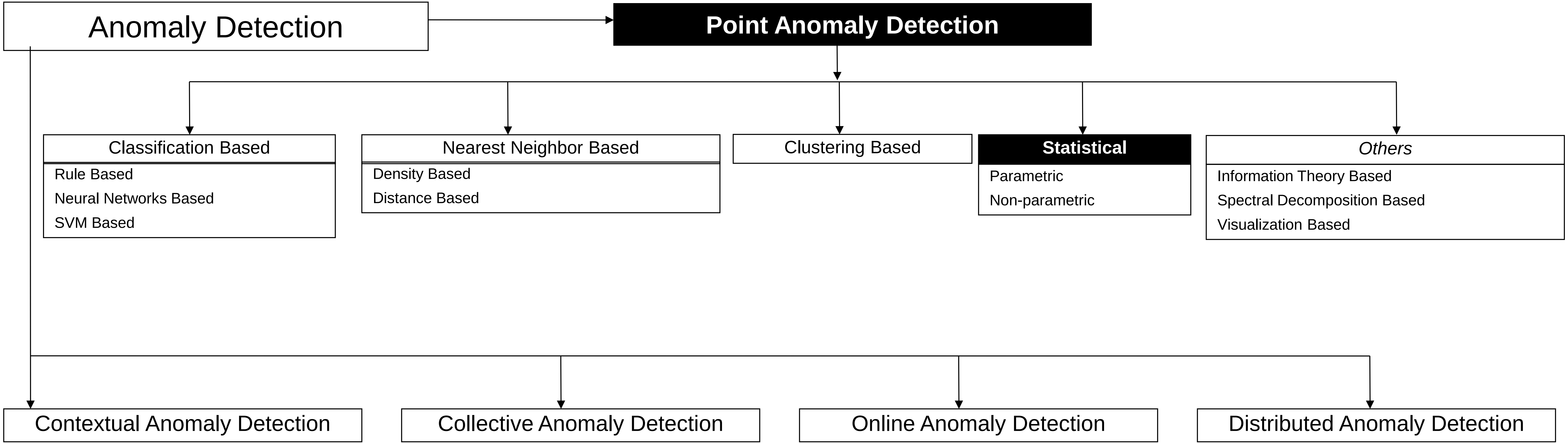
* D. Yu, G. Sheikholeslami, A. Zhang, FindOut: Finding Outliers in Very Large Datasets, 1999.

Cluster based Local Outlier Factor (CBLOF)

- Use squeezer clustering algorithm to perform clustering
- Determine CBLOF for each data record measured by both the size of the cluster and the distance to the cluster
  - if the data record lies in a ***small*** cluster, CBLOF is measured as a product of the size of the cluster the data record belongs to and the distance to the closest larger cluster
  - if the object belongs to a ***large*** cluster CBLOF is measured as a product of the size of the cluster that the data record belongs to and the distance between the data record and the cluster it belongs to (this provides importance of the local data behavior)



Taxonomy



Anomaly Detection

Point Anomaly Detection

Classification Based
- Rule Based
- Neural Networks Based
- SVM Based

Nearest Neighbor Based
- Density Based
- Distance Based

Clustering Based

Statistical
- Parametric
- Non-parametric

Others
- Information Theory Based
- Spectral Decomposition Based
- Visualization Based

Contextual Anomaly Detection

Collective Anomaly Detection

Online Anomaly Detection

Distributed Anomaly Detection

Statistics Based Techniques

- Data points are modeled using stochastic distribution $\Rightarrow$ points are determined to be outliers depending on their relationship with this model

- Advantage
  - Utilize existing statistical modeling techniques to model various type of distributions
- Challenges
  - With high dimensions, difficult to estimate distributions
  - Parametric assumptions often do not hold for real data sets

Types of Statistical Techniques

- Parametric Techniques
  - Assume that the normal (and possibly anomalous) data is generated from an underlying parametric distribution
  - Learn the parameters from the normal sample
  - Determine the likelihood of a test instance to be generated from this distribution to detect anomalies
- Non-parametric Techniques
  - Do not assume any knowledge of parameters
  - Use non-parametric techniques to learn a distribution – *e.g. parzen window estimation*

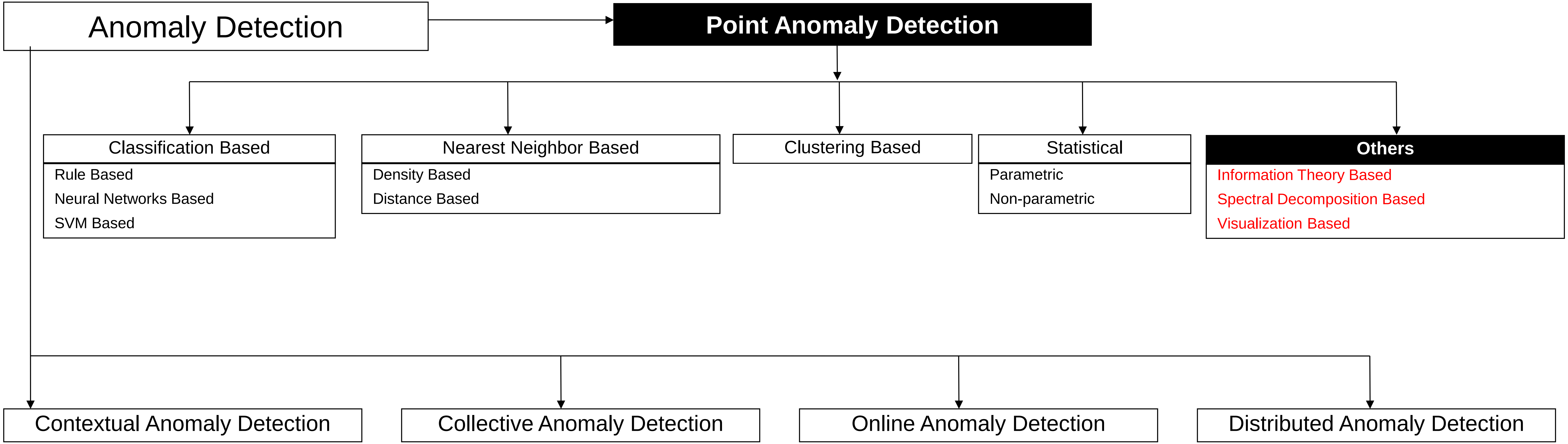
SmartSifter (SS)*

- Uses Finite Mixtures
- SS uses a probabilistic model as a representation of underlying mechanism of data generation.
  - Histogram density used to represent a probability density for categorical attributes
    - SDLE (Sequentially Discounting Laplace Estimation) for learning histogram density for categorical domain
  - Finite mixture model used to represent a probability density for continuous attributes
    - SDEM (Sequentially Discounting Expectation and Maximizing) for learning finite mixture for continuous domain
- SS gives a score to each example x_i on the basis of the learned model, measuring how large the model has changed after the learning

Using Probability Distributions*

- Basic Assumption: # of normal elements in the data is significantly larger then # of anomalies
- Distribution for the data *D* is given by:
  - $D = (1-\lambda)\cdot\mathbf{M} + \lambda\cdot\mathbf{A}$
    M - majority distribution, **A** - anomalous distribution
  - M_t, A_t sets of normal, anomalous elements respectively
  - Compute likelihood $L_t(D)$ of distribution *D* at time *t*
  - Measure how likely each element x_t is outlier:
    - $\mathrm{M_t} = \mathrm{M_{t-1}} \setminus \{\mathrm{x_t}\}$, $\mathrm{A_t} = \mathrm{A_{t-1}} \cup \{\mathrm{x_t}\}$
    - Measure the difference ($L_t - L_{t-1}$)

Taxonomy



Anomaly Detection

Point Anomaly Detection

Classification Based
- Rule Based
- Neural Networks Based
- SVM Based

Nearest Neighbor Based
- Density Based
- Distance Based

Clustering Based

Statistical
- Parametric
- Non-parametric

Others
- Information Theory Based
- Spectral Decomposition Based
- Visualization Based

Contextual Anomaly Detection

Collective Anomaly Detection

Online Anomaly Detection

Distributed Anomaly Detection

Information Theory Based Techniques

- Compute information content in data using information theoretic measures, e.g., entropy, relative entropy, etc.
- Key idea: Outliers significantly alter the information content in a dataset
- Approach: Detect data instances that significantly alter the information content
  - Require an information theoretic measure
- Advantage
  - Operate in an unsupervised mode
- Challenges
  - Require an information theoretic measure sensitive enough to detect irregularity induced by very few outliers

Information Theory Based Techniques

- Using a variety of information theoretic measures [Lee01]
- Kolmogorov complexity based approaches [Arning96]
  - Detect smallest data subset whose removal leads to maximal reduction in Kolmogorov complexity
- Entropy based approaches [He05]
  - Find a k-sized subset whose removal leads to the maximal decrease in entropy

Spectral Techniques

- Analysis based on eigen decomposition of data
- Key Idea
  - Find combination of attributes that capture bulk of variability
  - Reduced set of attributes can explain normal data well, but not necessarily the outliers
- Advantage
  - Can operate in an unsupervised mode
- Disadvantage
  - Based on the assumption that anomalies and normal instances are distinguishable in the reduced space
- Several methods use Principal Component Analysis
  - Top few principal components capture variability in normal data
  - Smallest principal component should have constant values
  - Outliers have variability in the smallest component

Using Robust PCA*

- Variability analysis based on robust PCA
  - Compute the principal components of the dataset
  - For each test point, compute its projection on these components
  - If y_i denotes the *i*th component, then the following has a chi-squared distribution

$$\sum_{i=1}^{q} \frac{y_i^2}{\lambda_i} = \frac{y_1^2}{\lambda_1} + \frac{y_2^2}{\lambda_2} + \ldots + \frac{y_q^2}{\lambda_q}, q \leq p$$

  - An observation is outlier if for a given significance level

$$\sum_{i=1}^{q} \frac{y_i^2}{\lambda_i} > \chi_q^2(\alpha)$$

  - Have been applied to intrusion detection, outliers in space-craft components, etc.

Temporal analysis of dynamic graphs

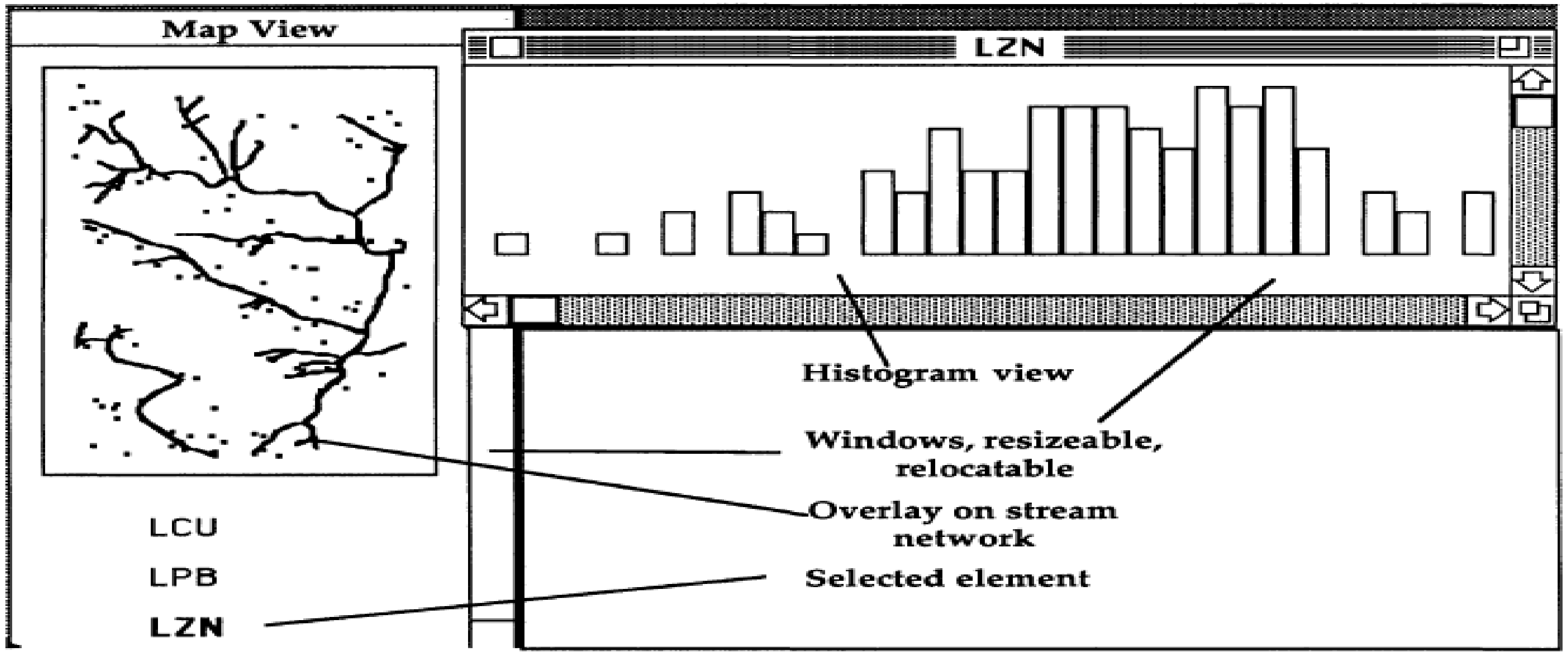
- Based on principal component analysis [Ide04]
  - Applied to network traffic data
  - For each time *t*, compute the principal component
  - Stack all principal components over time to form a matrix
  - Left singular vector of the matrix captures normal behavior
  - For any *t*, angle between principal component and the singular vector gives degree of anomaly
- Matrix approximation based methods [Sun07]
  - Form approximation based on CUR decomposition
  - Track approximation error over time
  - High approximation error implies outlying network traffic

Visualization Based Techniques

- Use visualization tools to observe the data
- Provide alternate views of data for manual inspection
- Anomalies are detected visually
- Advantages
  - Keeps a human in the loop
- Disadvantages
  - Works well for low dimensional data
  - Can provide only aggregated or partial views for high dimension data

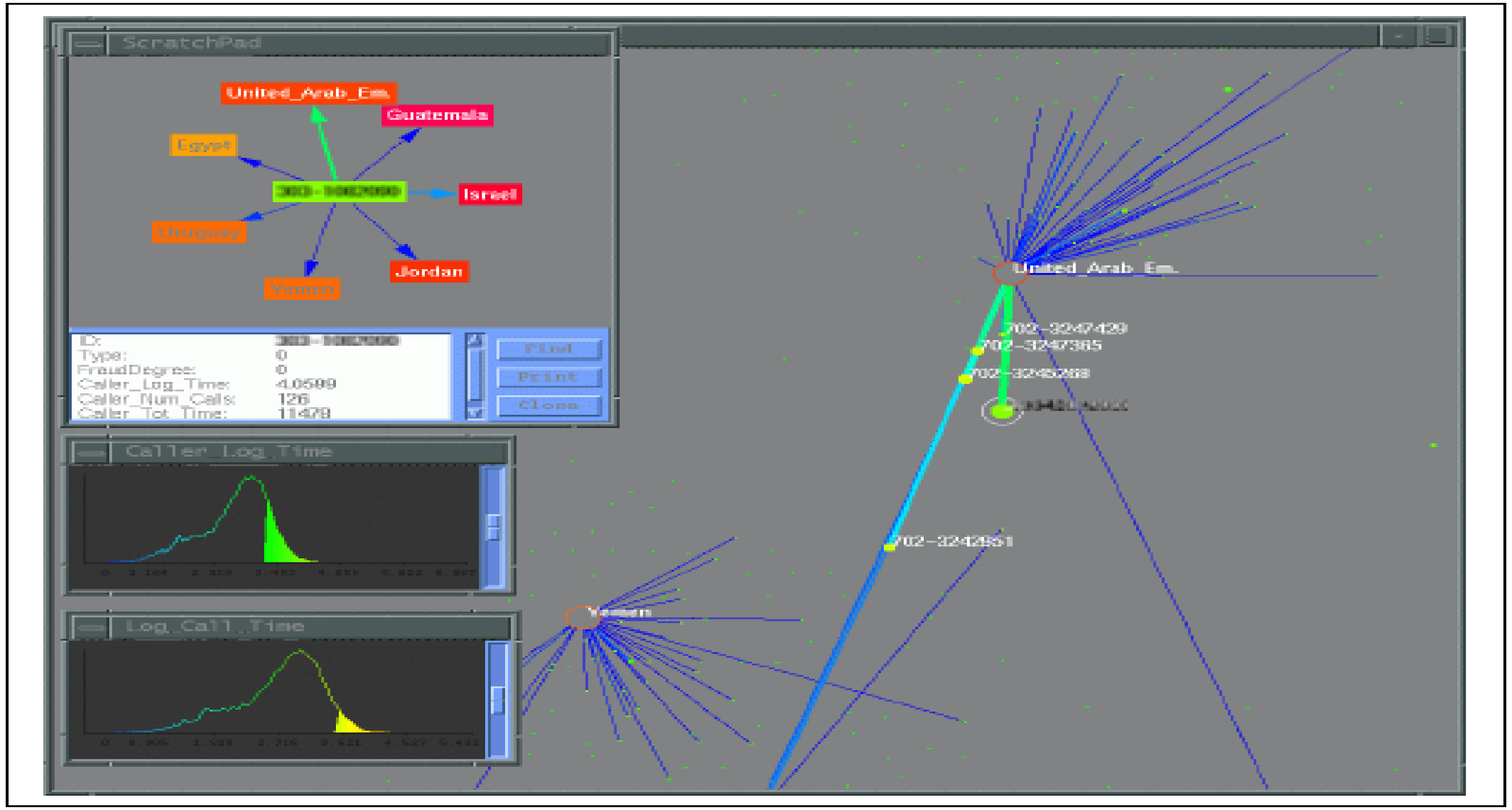
Application of Dynamic Graphics*

- Apply dynamic graphics to the exploratory analysis of spatial data.
- Visualization tools are used to examine local variability to detect anomalies
- Manual inspection of plots of the data that display its marginal and multivariate distributions



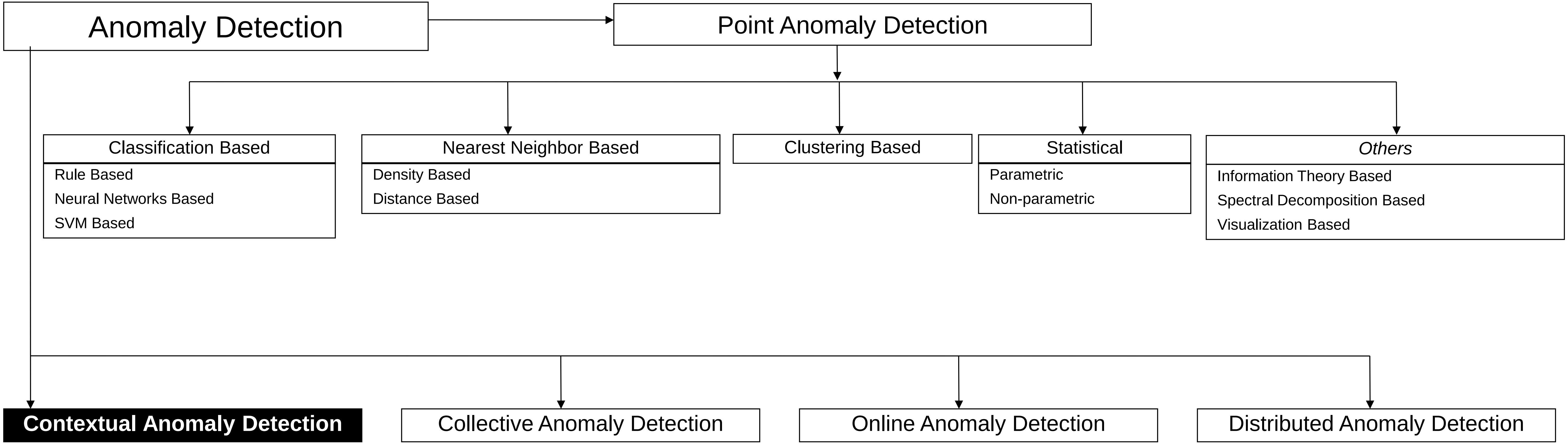
Visual Data Mining*

- Detecting Tele-communication fraud
- Display telephone call patterns as a graph
- Use colors to identify fraudulent telephone calls (anomalies)



ID: [redacted]
Type: 0
FraudDegree: 0
Caller_Log_Time: 4.0589
Caller_Num_Calls: 126
Caller_Tot_Time: 11478

Taxonomy



Anomaly Detection

Point Anomaly Detection

Classification Based
- Rule Based
- Neural Networks Based
- SVM Based

Nearest Neighbor Based
- Density Based
- Distance Based

Clustering Based

Statistical
- Parametric
- Non-parametric

Others
- Information Theory Based
- Spectral Decomposition Based
- Visualization Based

Contextual Anomaly Detection

Collective Anomaly Detection

Online Anomaly Detection

Distributed Anomaly Detection

Contextual Anomaly Detection

- Detect context anomalies
- General Approach
  - Identify a context around a data instance (using a set of *contextual attributes*)
  - Determine if the data instance is anomalous w.r.t. the context (using a set of *behavioral attributes*)
- Assumption
  - All normal instances within a context will be similar (in terms of behavioral attributes), while the anomalies will be different

Contextual Anomaly Detection

- Advantages
  - Detect anomalies that are hard to detect when analyzed in the global perspective
- Challenges
  - Identifying a set of good contextual attributes
  - Determining a context using the contextual attributes

Contextual Attributes

- Contextual attributes define a neighborhood (context) for each instance
- For example:
  - Spatial Context
    - *Latitude, Longitude*
  - Graph Context
    - *Edges, Weights*
  - Sequential Context
    - *Position, Time*
  - Profile Context
    - *User demographics*

Contextual Anomaly Detection Techniques

- Techniques
  - Reduction to point outlier detection
    - Segment data using contextual attributes
    - Apply a traditional point outlier within each context using behavioral attributes
  - Utilizing structure in data
    - Build models from the data using contextual attributes
      - E.g. – Time series models (ARIMA, etc.)

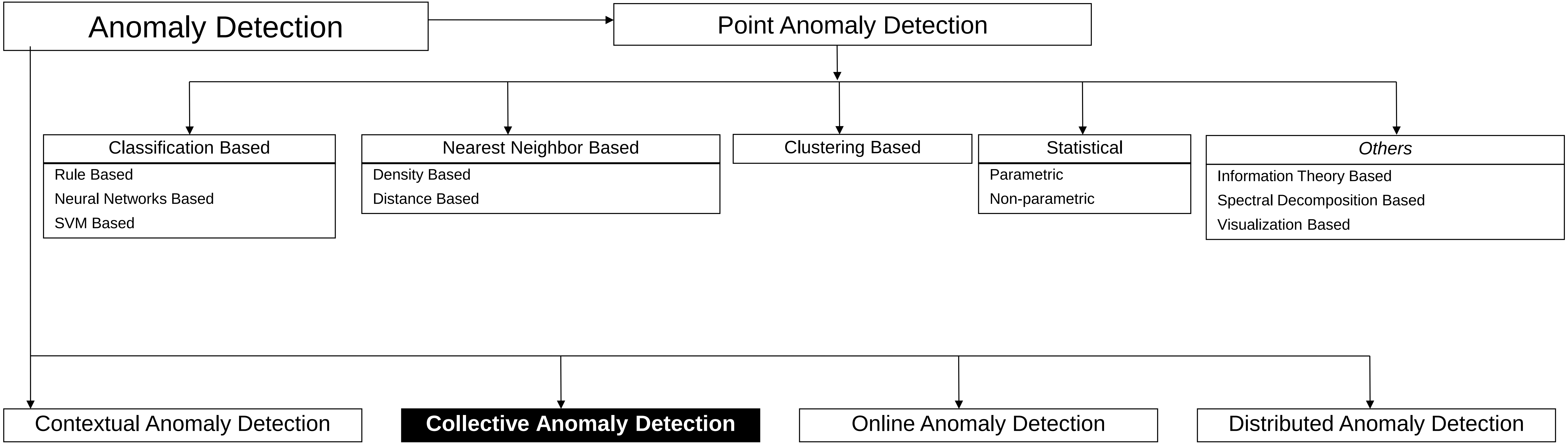
Conditional Anomaly Detection*

- Each data point is represented as [x,y], where x denotes the *environmental (contextual) attributes* and y denotes the *indicator (behavioral) attributes*
- A mixture of N_U Gaussian models, U is learnt from the contextual data
- A mixture of N_V Gaussian models, V is learn from the behavioral data
- A mapping $p(V_j|U_i)$ is learnt that indicates the probability of the behavioral part to be generated by component V_j when the contextual part is generated by component U_i

- Outlier Score of a data instance ([x,y]):

$$Outlier\ Score = \sum_{i=1}^{n_U} p(x \in U_i) \sum_{j=1}^{n_V} p(y \in V_j) p(V_j|U_i)$$

Taxonomy



Anomaly Detection
Point Anomaly Detection
Classification Based
Rule Based
Neural Networks Based
SVM Based
Nearest Neighbor Based
Density Based
Distance Based
Clustering Based
Statistical
Parametric
Non-parametric
Others
Information Theory Based
Spectral Decomposition Based
Visualization Based
Contextual Anomaly Detection
Collective Anomaly Detection
Online Anomaly Detection
Distributed Anomaly Detection

Collective Anomaly Detection

- Detect collective anomalies
- Exploit the relationship among data instances
- Sequential anomaly detection
  - Detect anomalous sequences
- Spatial anomaly detection
  - Detect anomalous sub-regions within a spatial data set
- Graph anomaly detection
  - Detect anomalous sub-graphs in graph data

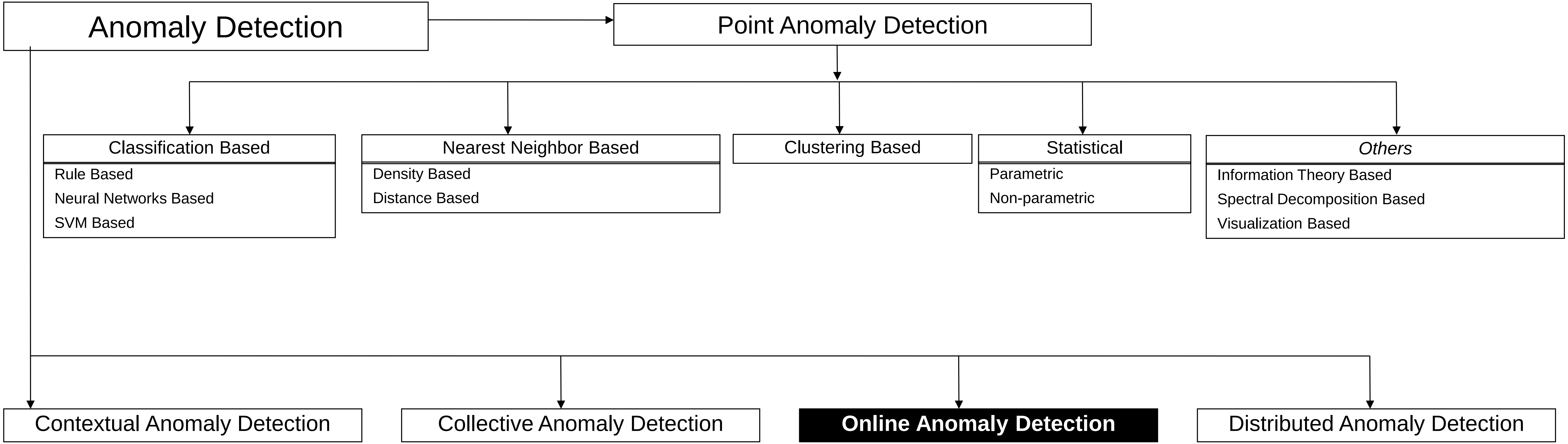
Sequential Anomaly Detection

- Detect anomalous sequences in a database of sequences, or
- Detect anomalous subsequence within a sequence
- Data is presented as a set of symbolic sequences
  - System call intrusion detection
  - Proteomics
  - Climate data

Sequence Time Delay Embedding (STIDE)*

- Assumes a training data containing normal sequences
- Training
  - Extracts fixed length (k) subsequences by sliding a window over the training data
  - Maintain counts for all subsequences observed in the training data
- Testing
  - Extract fixed length subsequences from the test sequence
  - Find empirical probability of each test subsequence from the above counts
  - If probability for a subsequence is below a threshold, the subsequence is declared as anomalous
  - Number of anomalous subsequences in a test sequence is its anomaly score
- Applied for system call intrusion detection

Taxonomy



Anomaly Detection

Point Anomaly Detection

Classification Based
- Rule Based
- Neural Networks Based
- SVM Based

Nearest Neighbor Based
- Density Based
- Distance Based

Clustering Based

Statistical
- Parametric
- Non-parametric

Others
- Information Theory Based
- Spectral Decomposition Based
- Visualization Based

Contextual Anomaly Detection

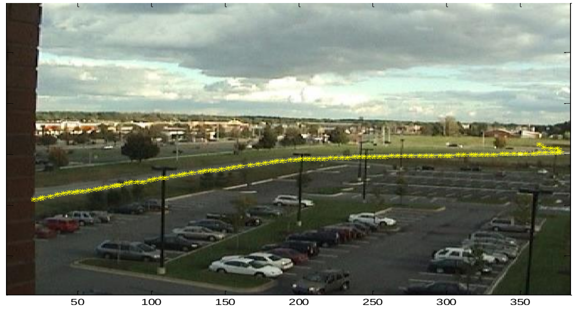
Collective Anomaly Detection

Online Anomaly Detection

Distributed Anomaly Detection

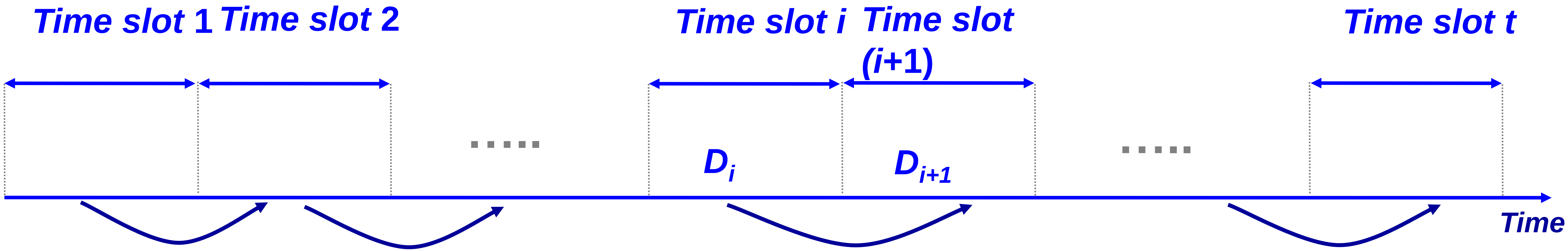
Motivation for On-line Anomaly Detection

- Data in many rare events applications arrives continuously at an enormous pace
- There is a significant challenge to analyze such data
- Examples of such rare events applications:
  - Video analysis
  - Network traffic monitoring
  - Aircraft safety
  - Credit card fraudulent transactions



On-line Anomaly Detection – Simple Idea

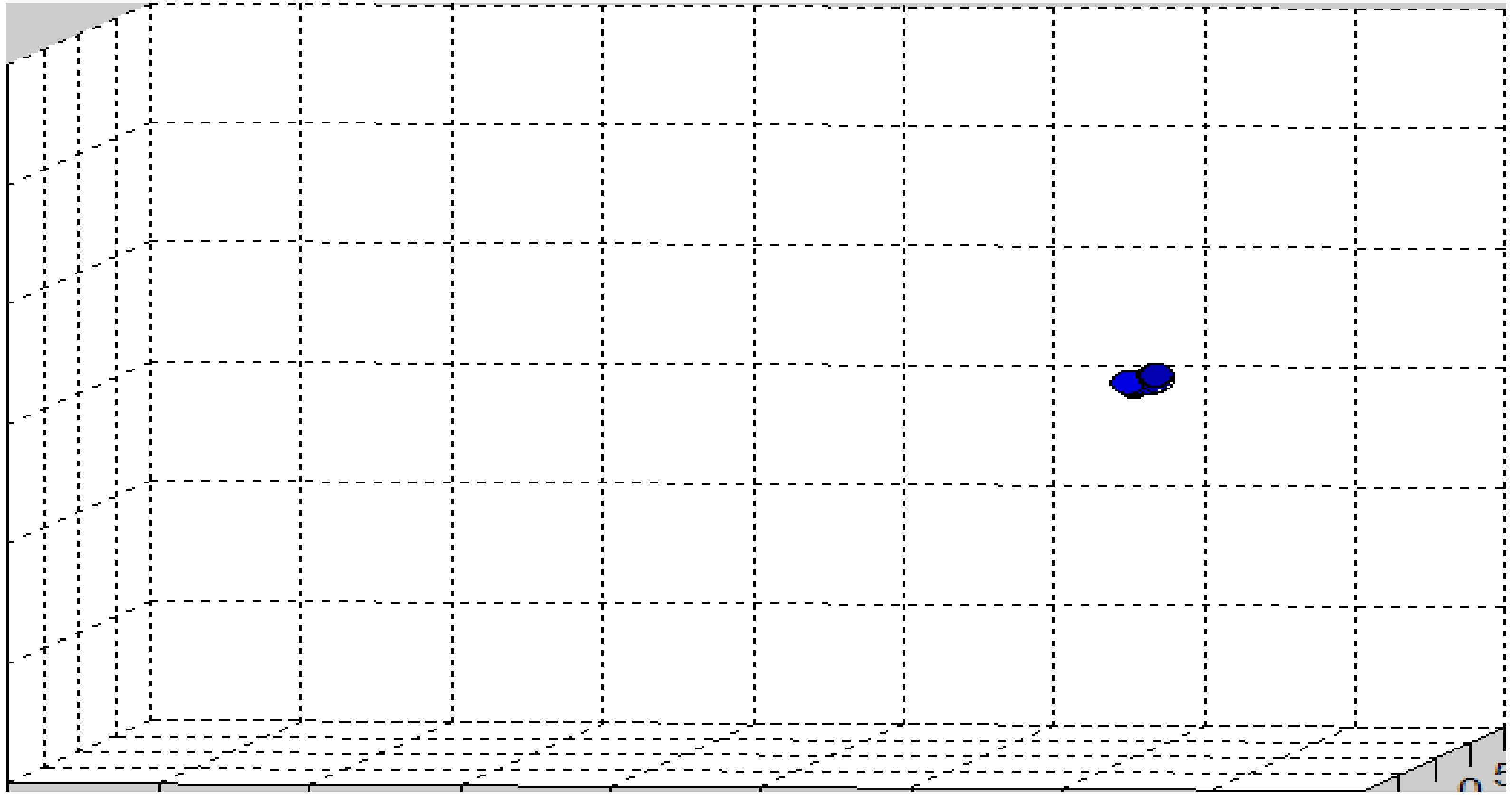
- The normal behavior is changing through time
- Need to update the "normal behavior" profile dynamically
  - Key idea: Update the normal profile with the data records that are "probably" normal, i.e. have very low anomaly score



  - Time slot *i* – Data block D_i – model of normal behavior M_i
  - Anomaly detection algorithm in time slot (*i+1*) is based on the profile computed in time slot *i*

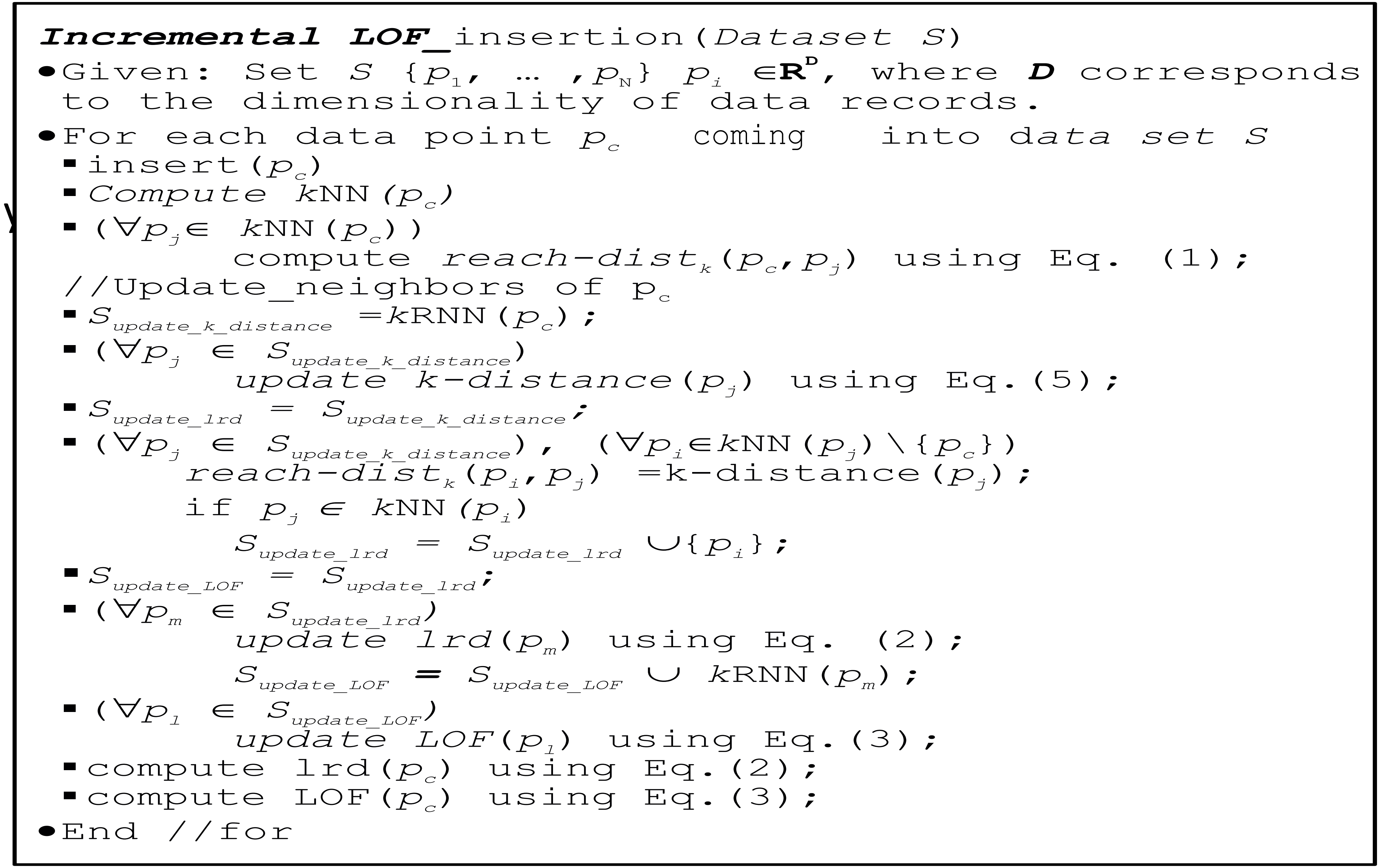
Drawbacks of simple on-line anomaly detection algorithm

- If arriving data points start to create a new data cluster, this method will not be able to detect these points as outliers neither the time when the change occurred



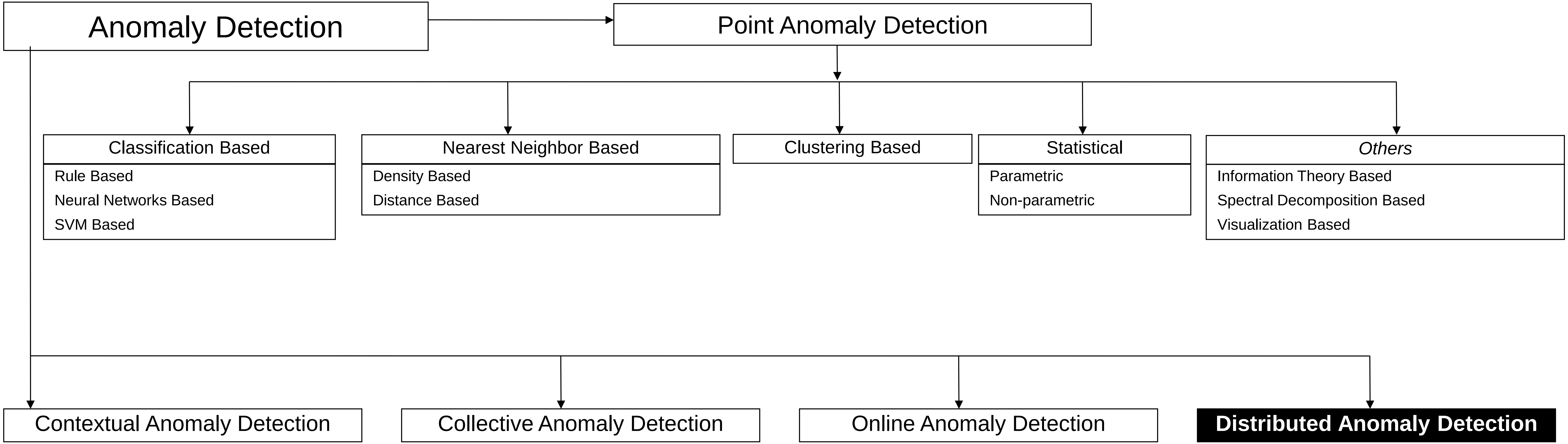
Incremental LOF algorithm

- Incremental *LOF* algorithm computes *LOF* value for each inserted data record and instantly determines whether that data record is outlier
- *LOF* values for existing data records are updated if necessary



```
Incremental LOF_insertion(Dataset S)
• Given: Set $S$ $\{p_1, \dots ,p_N\}$ $p_i \in \mathbf{R}^D$, where $D$ corresponds
  to the dimensionality of data records.
• For each data point $p_c$ coming into data set $S$
  ▪ insert($p_c$)
  ▪ Compute kNN($p_c$)
  ▪ ($\forall p_j \in$ kNN($p_c$))
        compute reach-dist$_k$($p_c$,$p_j$) using Eq. (1);
  //Update_neighbors of p$_c$
  ▪ $S_{update\_k\_distance}$ =kRNN($p_c$);
  ▪ ($\forall p_j \in S_{update\_k\_distance}$)
        update k-distance($p_j$) using Eq.(5);
  ▪ $S_{update\_lrd}$ = $S_{update\_k\_distance}$;
  ▪ ($\forall p_j \in S_{update\_k\_distance}$), ($\forall p_i \in$kNN($p_j$)$\setminus\{p_c\}$)
      reach-dist$_k$($p_i$,$p_j$) =k-distance($p_j$);
      if $p_j \in$ kNN($p_i$)
        $S_{update\_lrd}$ = $S_{update\_lrd} \cup \{p_i\}$;
  ▪ $S_{update\_LOF}$ = $S_{update\_lrd}$;
  ▪ ($\forall p_m \in S_{update\_lrd}$)
        update lrd($p_m$) using Eq. (2);
        $S_{update\_LOF}$ = $S_{update\_LOF} \cup$ kRNN($p_m$);
  ▪ ($\forall p_l \in S_{update\_LOF}$)
        update LOF($p_l$) using Eq.(3);
  ▪ compute lrd($p_c$) using Eq.(2);
  ▪ compute LOF($p_c$) using Eq.(3);
• End //for
```

Taxonomy



Anomaly Detection

Point Anomaly Detection

Classification Based
- Rule Based
- Neural Networks Based
- SVM Based

Nearest Neighbor Based
- Density Based
- Distance Based

Clustering Based

Statistical
- Parametric
- Non-parametric

Others
- Information Theory Based
- Spectral Decomposition Based
- Visualization Based

Contextual Anomaly Detection

Collective Anomaly Detection

Online Anomaly Detection

Distributed Anomaly Detection

Need for Distributed Anomaly Detection

- Data in many anomaly detection applications may come from many different sources
  - Network intrusion detection
  - Credit card fraud
  - Aviation safety
- Failures that occur at multiple locations simultaneously may be undetected by analyzing only data from a single location
  - Detecting anomalies in such complex systems may require integration of information about detected anomalies from single locations in order to detect anomalies at the global level of a complex system
- There is a need for the high performance and distributed algorithms for correlation and integration of anomalies

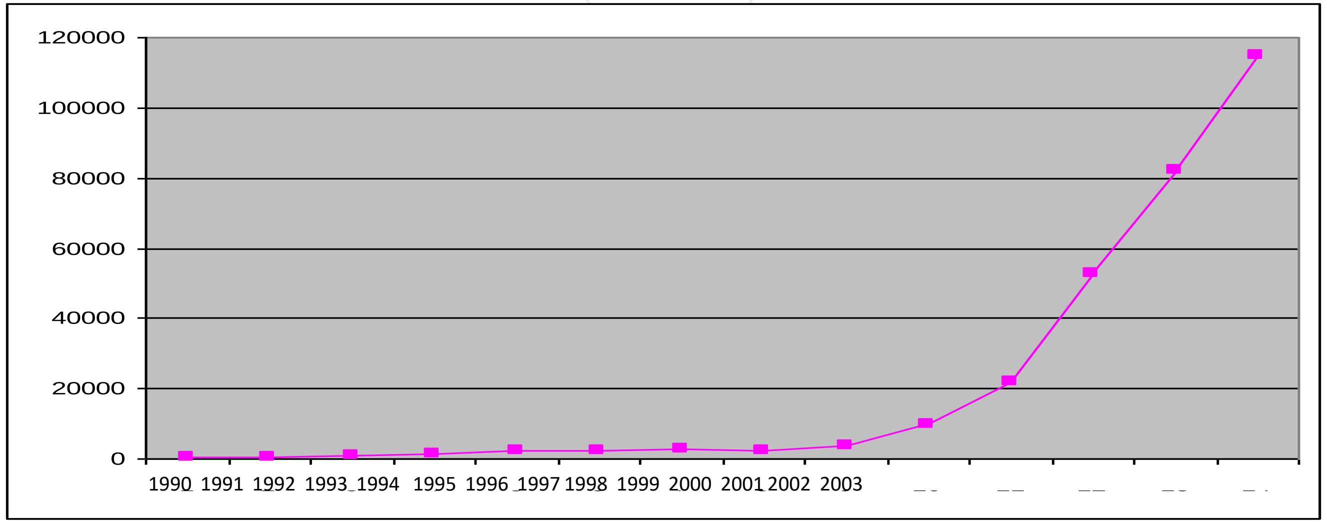
Distributed Anomaly Detection Techniques

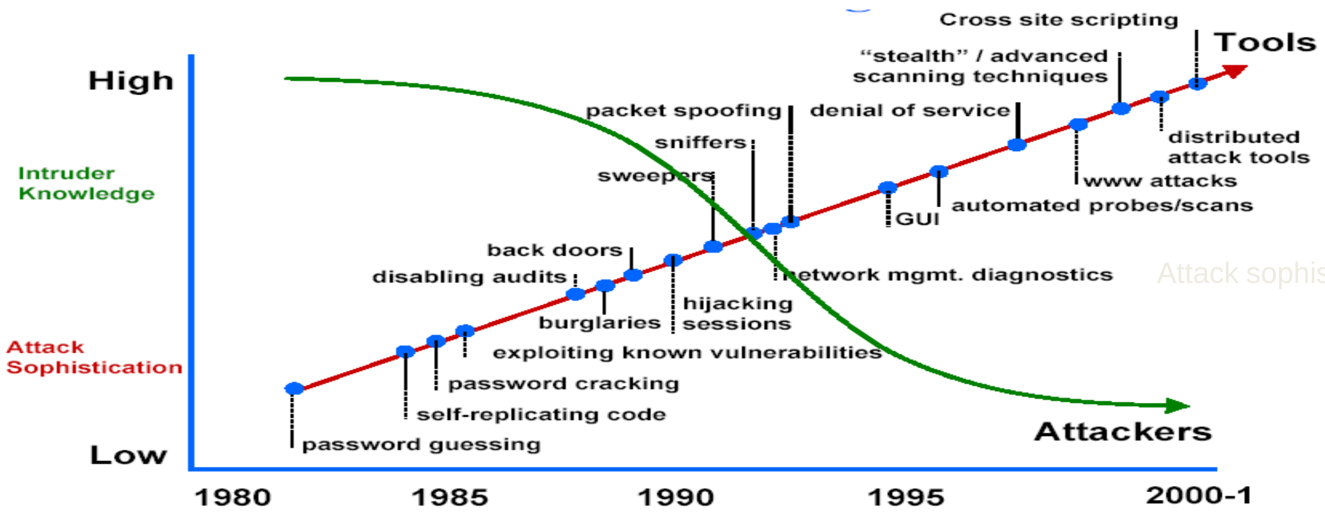
- Simple data exchange approaches
  - Merging data at a single location
  - Exchanging data between distributed locations
- Distributed nearest neighboring approaches
  - Exchanging one data record per distance computation – computationally inefficient
  - privacy preserving anomaly detection algorithms based on computing distances across the sites
- Methods based on exchange of models
  - explore exchange of appropriate statistical / data mining models that characterize normal / anomalous behavior
    - identifying modes of normal behavior;
    - describing these modes with statistical / data mining learning models; and
    - exchanging models across multiple locations and combing them at each location in order to detect global anomalies

Case Study: Data Mining in Intrusion Detection

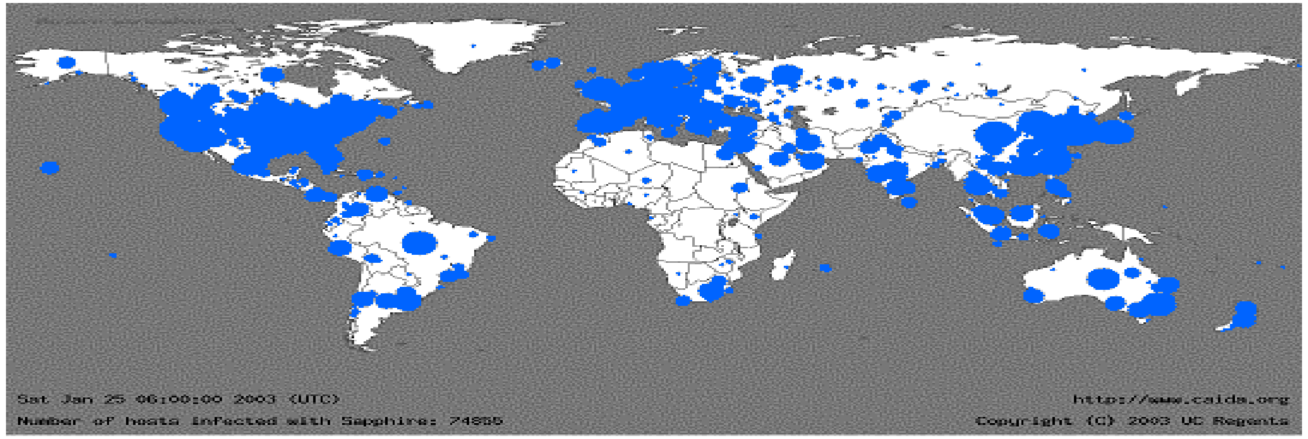
- **Due to the proliferation of Internet, more and more organizations are becoming vulnerable to cyber attacks**
- **Sophistication of cyber attacks as well as their severity is also increasing**

Incidents Reported to Computer Emergency Response Team/Coordination Center (CERT/CC)





Attack sophistication vs. Intruder technical knowledge, source: www.cert.org/archive/ppt/cyberterror.ppt


Sat Jan 25 06:00:00 2003 (UTC)
Number of hosts infected with Sapphire: 74855
http://www.caida.org
Copyright (C) 2003 UC Regents

- **Security mechanisms always have inevitable vulnerabilities**
  - **Firewalls are not sufficient to ensure security in computer networks**
  - **Insider attacks**

What are Intrusions?

- Intrusions are actions that attempt to bypass security mechanisms of computer systems. They are usually caused by:
  - Attackers accessing the system from Internet
  - Insider attackers - authorized users attempting to gain and misuse non-authorized privileges
- Typical intrusion scenario

Intrusion Detection

- Intrusion Detection System
  - combination of software and hardware that attempts to perform intrusion detection
  - raises the alarm when possible intrusion happens
- Traditional intrusion detection system IDS tools (e.g. SNORT) are based on signatures of known attacks
  - Example of SNORT rule (MS-SQL “Slammer” worm)

```
any -> udp port 1434 (content:"|81 F1 03 01 04 9B 81 F1 01|";
content:"sock"; content:"send")
```

- Limitations
  - Signature database has to be manually revised for each new type of discovered intrusion
  - They cannot detect emerging cyber threats
  - Substantial latency in deployment of newly created signatures across the computer system
- Data Mining can alleviate these limitations

Data Mining for Intrusion Detection

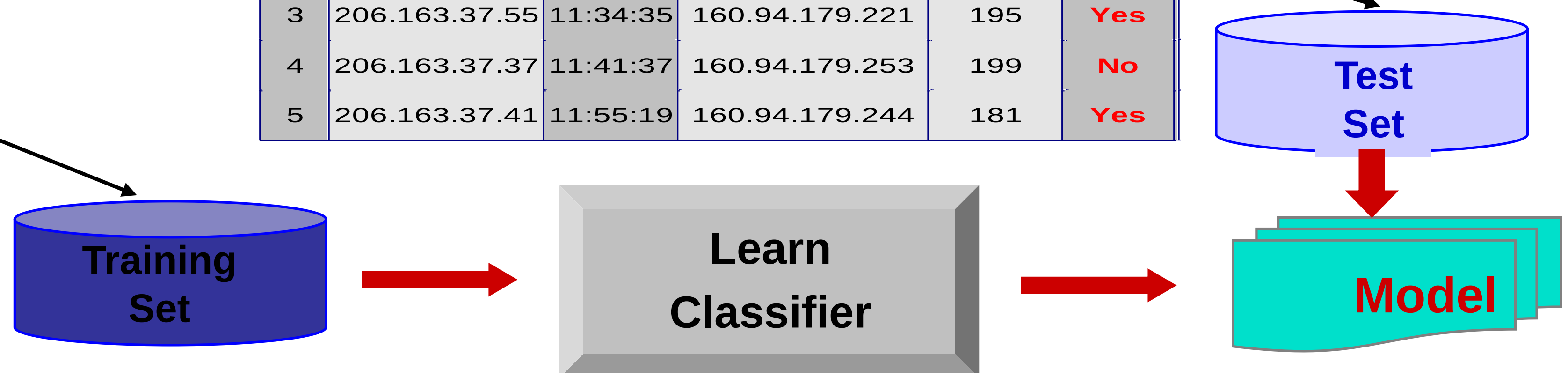
- Increased interest in data mining based intrusion detection
  - Attacks for which it is difficult to build signatures
  - Attack stealthiness
  - Unforeseen/Unknown/Emerging attacks
  - Distributed/coordinated attacks
- Data mining approaches for intrusion detection
  - *Misuse detection*
    - Building predictive models from labeled labeled data sets (instances are labeled as "normal" or "intrusive") to identify known intrusions
    - High accuracy in detecting many kinds of known attacks
    - Cannot detect unknown and emerging attacks
  - *Anomaly detection*
    - Detect novel attacks as deviations from "normal" behavior
    - Potential high false alarm rate - previously unseen (yet legitimate) system behaviors may also be recognized as anomalies
  - *Summarization of network traffic*

Data Mining for Intrusion Detection

Misuse Detection – Building Predictive Models

Tid	SrcIP	Start time	Dest IP	Dest Port	Number of bytes	Attack
1	206.135.38.95	11:07:20	160.94.179.223	139	192	No
2	206.163.37.95	11:13:56	160.94.179.219	139	195	No
3	206.163.37.95	11:14:29	160.94.179.217	139	180	No
4	206.163.37.95	11:14:30	160.94.179.255	139	199	No
5	206.163.37.95	11:14:32	160.94.179.254	139	19	Yes
6	206.163.37.95	11:14:35	160.94.179.253	139	177	No
7	206.163.37.95	11:14:36	160.94.179.252	139	172	No
8	206.163.37.95	11:14:38	160.94.179.251	139	285	Yes
9	206.163.37.95	11:14:41	160.94.179.250	139	195	No
10	206.163.37.95	11:14:44	160.94.179.249	139	163	Yes

	categorical	temporal	categorical	continuous	class
Tid	SrcIP	Start time	Dest IP	Number of bytes	Attack
1	206.163.37.81	11:17:51	160.94.179.208	150	No
2	206.163.37.99	11:18:10	160.94.179.235	208	No
3	206.163.37.55	11:34:35	160.94.179.221	195	Yes
4	206.163.37.37	11:41:37	160.94.179.253	199	No
5	206.163.37.41	11:55:19	160.94.179.244	181	Yes

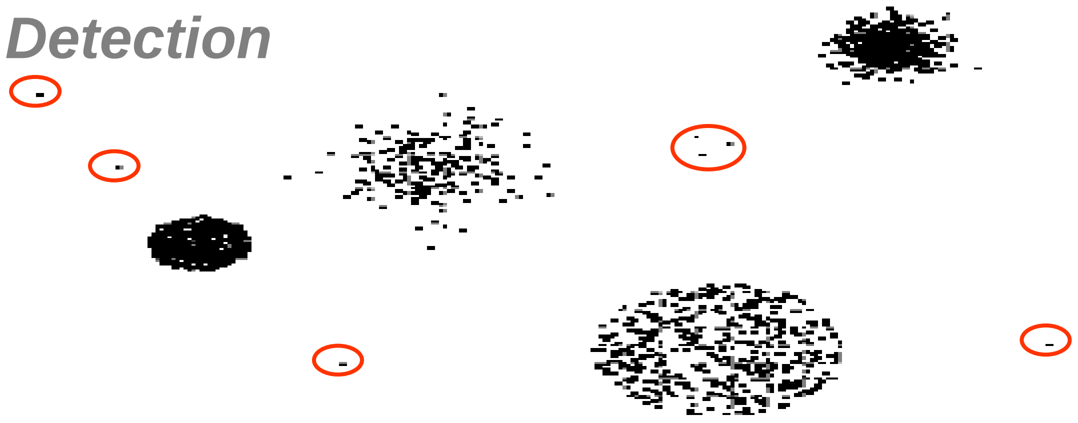


Summarization of attacks using association rules

Rules Discovered:

{Src IP = 206.163.37.95,
Dest Port = 139,
Bytes ∊ [150, 200]} --> {ATTACK}

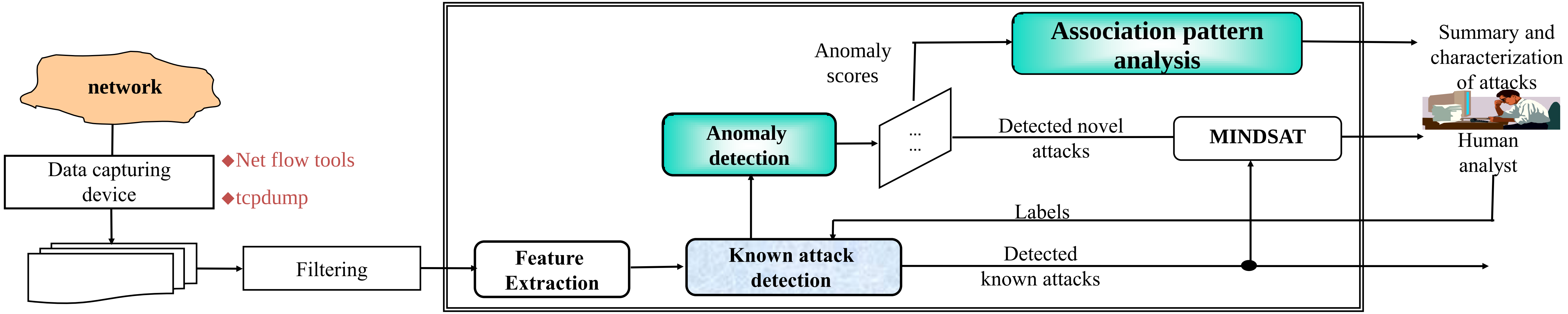
Anomaly Detection



Anomaly Detection on Real Network Data

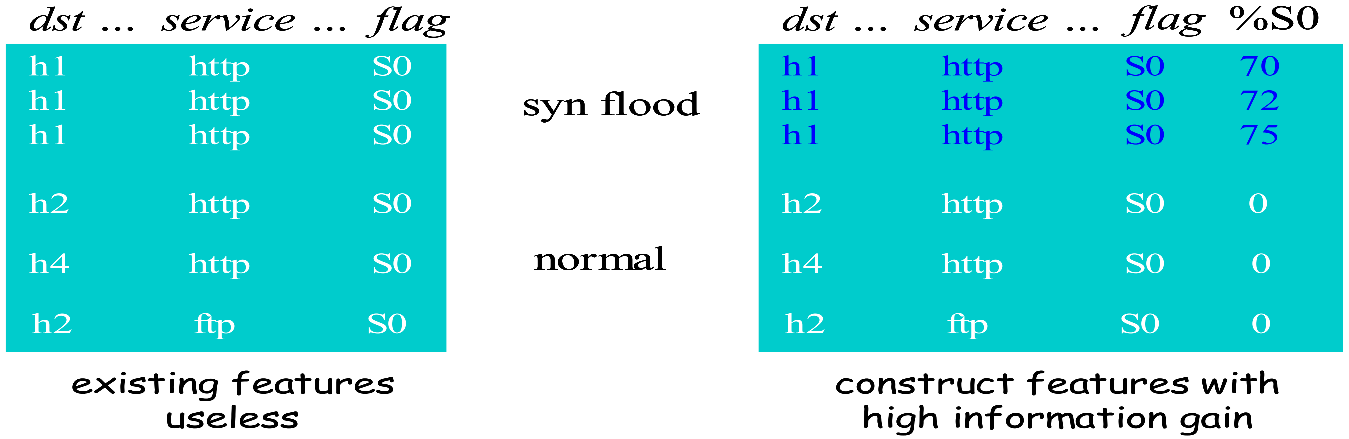
- Anomaly detection was used at U of Minnesota and Army Research Lab to detect various intrusive/suspicious activities
- Many of these could not be detected using widely used intrusion detection tools like SNORT
- Anomalies/attacks picked by *MINDS*
  - Scanning activities
  - Non-standard behavior
    - Policy violations
    - Worms

MINDS – Minnesota Intrusion Detection System



Feature Extraction

- 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

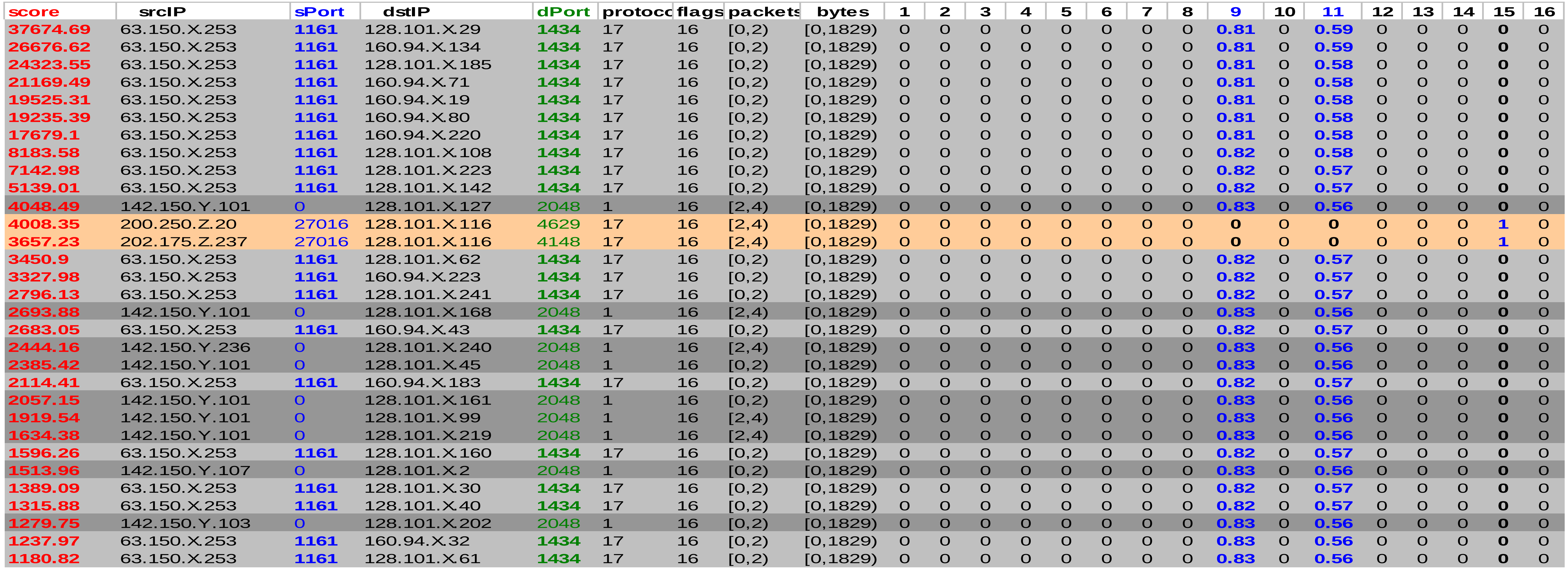
dst ...	*service* ...	*flag*	%S0
h1	http	S0	70
h1	http	S0	72
h1	http	S0	75
h2	http	S0	0
h4	http	S0	0
h2	ftp	S0	0

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

– 48 hours after the “slammer” worm



score	srcIP	sPort	dstIP	dPort	protocc	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 UM machines connecting to “half-life” game servers

Detection of Anomalies on Real Network Data

- Anomalies/attacks picked by MINDS include scanning activities, worms, and non-standard behavior such as policy violations and insider attacks. Many of these attacks detected by MINDS, have already been on the CERT/CC list of recent advisories and incident notes.
- Some illustrative examples of intrusive behavior detected using MINDS at U of M

- Scans
  - August 13, 2004, **Detected scanning for Microsoft DS service on port 445/TCP (Ranked#1)**
    - Reported by CERT as recent DoS attacks that needs further analysis (CERT August 9, 2004)
    - Undetected by SNORT since the scanning was non-sequential (very slow). Rule added to SNORT in September 2004
  - August 13, 2004, Detected scanning for Oracle server (Ranked #2), Reported by CERT, June 13, 2004
    - Undetected by SNORT because the scanning was hidden within another Web scanning
  - October 10, 2005, Detected a distributed windows networking scan from multiple source locations (Ranked #1)
- Policy Violations
  - August 8, 2005, Identified machine running Microsoft PPTP VPN server on non-standard ports (Ranked #1)
    - Undetected by SNORT since the collected GRE traffic was part of the normal traffic
  - August 10 2005 & October 30, 2005, Identified compromised machines running FTP servers on non-standard ports, which is a policy violation (Ranked #1)
    - Example of anomalous behavior following a successful Trojan horse attack
  - February 6, 2006, The IP address 128.101.X.0 (not a real computer, but a network itself) has been targeted with IP Protocol 0 traffic from Korea (61.84.X.97) (bad since IP Protocol 0 is not legitimate)
  - February 6, 2006, Detected a computer on the network apparently communicating with a computer in California over a VPN or on IPv6
- Worms
  - October 10, 2005, Detected several instances of slapper worm that were not identified by SNORT since they were variations of existing worm code
  - February 6, 2006, Detected unsolicited ICMP ECHOREPLY messages to a computer previously infected with Stacheldract worm (a DDos agent)

Conclusions

- Anomaly detection can detect critical information in data
- Highly applicable in various application domains
- Nature of anomaly detection problem is dependent on the application domain
- Need different approaches to solve a particular problem formulation

Usage Notes

- A lot of slides are adopted from the presentations and documents published on internet by experts who know the subject very well.
- I would like to thank who prepared slides and documents.
- Also, these slides are made publicly available on the web for anyone to use
- If you choose to use them, I ask that you alert me of any mistakes which were made and allow me the option of incorporating such changes (with an acknowledgment) in my set of slides.

Sincerely,
Dr. Cahit Karakuş
cahitkarakus@gmail.com