An Introduction to Anomaly Detection

COMP90073
Security Analytics

Sarah Erfani, CIS

Semester 2, 2021

Outline

- Using machine learning in cybersecurity
- Basics of machine learning
- Introduction to anomaly detection
- Isolation Forest (iForest)

Why Machine Learning and Security?

Intersecting Machine Learning and Cybersecurity

By WIREs Authors

Posted on May 7, 2019

Artificial intelligence in cyber security market is valued at $4.94bn in 2019, according to Visiongain

GlobeNewswire • May 8, 2019

Applying AI And Machine Learning To Boost Cybersecurity

Forbes Technology Council

Dr. Rao Papolu Forbes Councils
Forbes Technology Council CommunityVoice

Automation in Cybersecurity Key to Addressing Growing Risks

By Simon Eid, Area Vice President, Australia and New Zealand

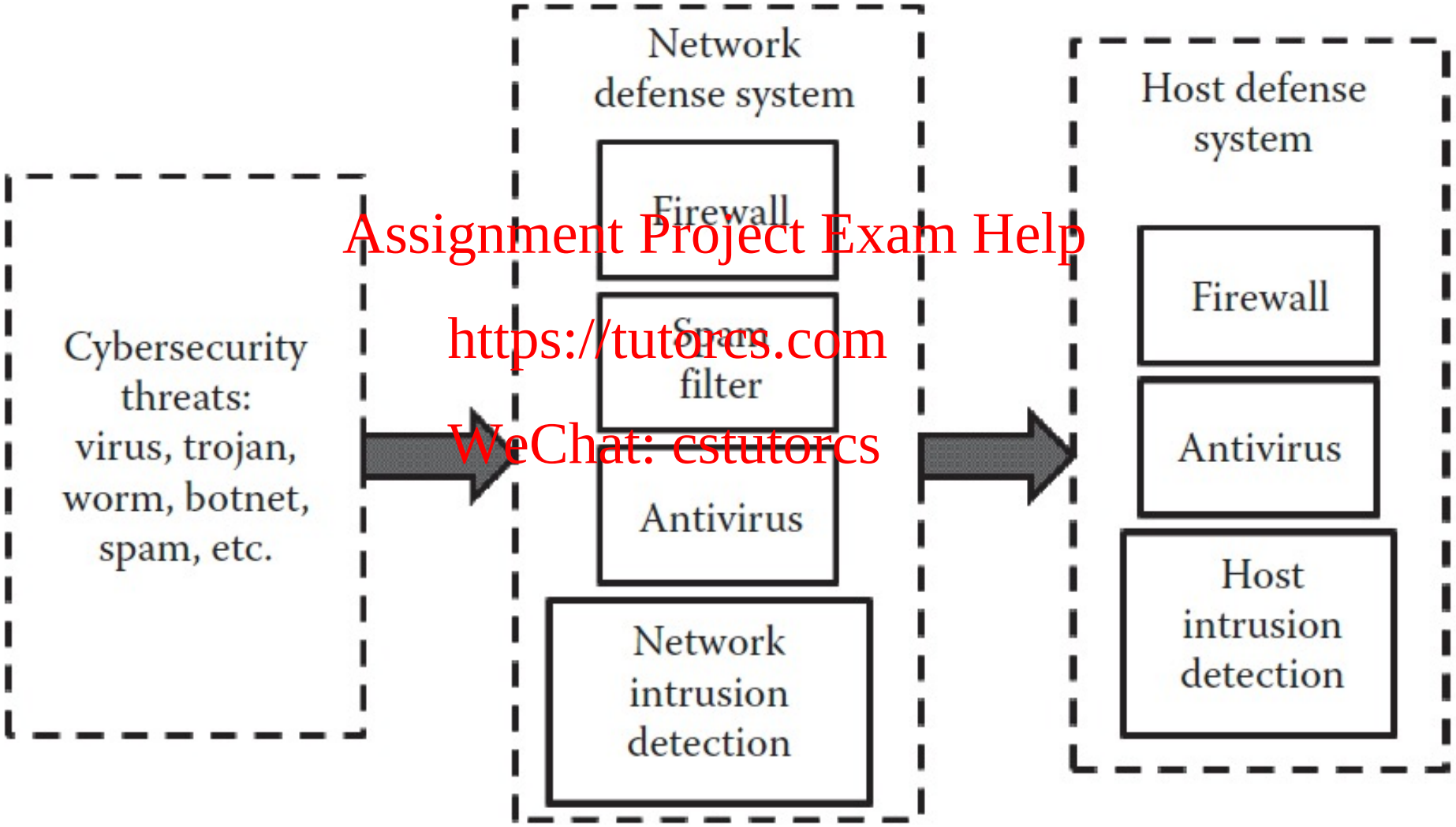
Simon Eid (CSO Online) on 14 May, 2019 14:29

Staples

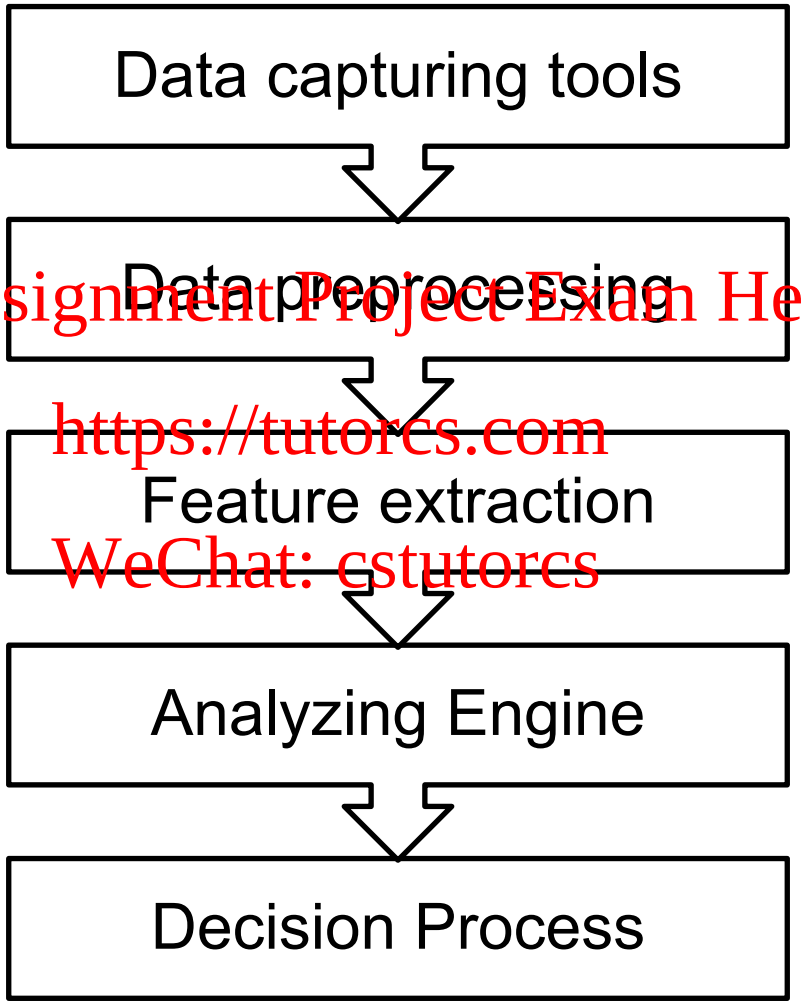
How AI Beefs Up Cybersecurity

Artificial intelligence gives chief information security officers an important new advantage in the ongoing efforts to improve cybersecurity. Find out what to consider when evaluating the latest tools.

Conventional Cybersecurity System



Adaptive Defense System for Cybersecurity



Cyber Security Solutions

- **Proactive:**

  Maintain the overall security of a system, even if individual components of the system have been compromised by an attack, i.e., *Privacy Preserving Data Mining (PPDM).*

- **Reactive:**

  Identify any unauthorized attempt to access, manipulate, modify, or destroy information or to use a computer system remotely to spam, hack, or modify other computers, i.e., *Intrusion Detection System (IDS).*

Intrusion Detection Systems (IDS)

- **Signature (Misuse) Detection:** Measures the similarity between input events and the signatures of known intrusions

- **Anomaly Detection:** Triggers alarms when the detected object behaves significantly differently from the predefined normal patterns

Overview of ML – Revision

Input to a machine learning system can consist of instance/measurements about individual entities/objects, e.g., *a network packet.*

- **Attribute** (aka Feature, explanatory variable): component of the instances *source IP, destination IP, source port, destination port, etc.*

- **Label** (aka Response, dependent variable): an outcome that is categorical, numeric, etc. *attack vs. legitimate traffic*

- **Models**: discovered relationship between attributes and/or label

Supervised and Unsupervised Learning – Revision

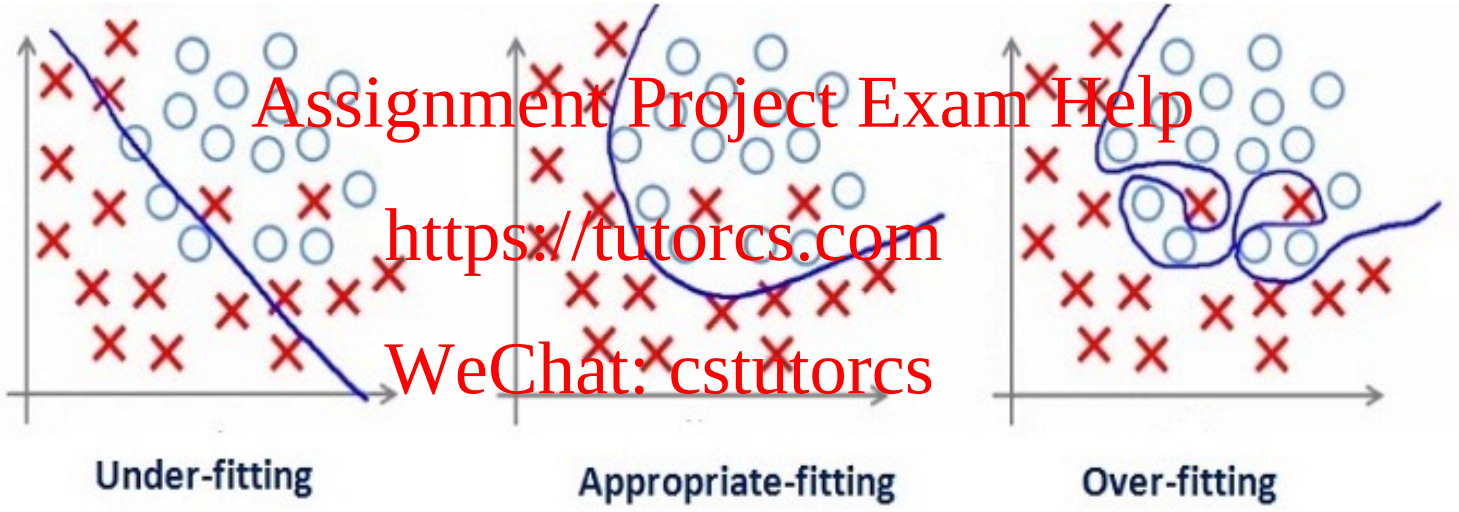
- **Supervised learning**
  - Teach the computer how to do something (by example), then let it
  - Use its new-found knowledge to do it
  - Labelled data: for given inputs, provide the expected output ("the answer")
  - Infer a function mapping from inputs to outputs
- **Unsupervised learning**
  - Let the computer learn how to do something
  - Determine structure and patterns in data
  - Unlabelled data: Don't give the computer "the answer"

Model Validation – Revision

- **Holdout:** Train a classifier over a fixed training dataset, and evaluate it over a fixed held-out test dataset
- **Random Subsampling:** Perform holdout over multiple iterations, randomly selecting the training and test data (maintaining a fixed size for each dataset) on each iteration
- **Leave-One-Out:** Choose each data point as test case and the rest as training data
- **M-fold Cross-Validation:** Partition the data into M (approximately) equal size partitions, and choose each partition for testing and the remaining M-1 partitions for training

Chose a validation model that is efficient, and minimises bias and variance in evaluation.

Generalisation Problem – Revision



Performance Evaluation – Revision

- Confusion Matrix

		Actual	
		Cat	Dog
Predicted	Cat	4 (TP)	3 (FP)
	Dog	2 (FN)	6 (TN)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$= \frac{4 + 6}{4 + 6 + 3 + 2} \cong 67\%$$

Limitation of Accuracy – Revision

- Anomaly detection
  - Number of negative examples = 9990
  - Number of positive examples = 10
- If model predicts everything to be class 0, accuracy is 9990/10000 = 99.9%
  - Accuracy is misleading because model does not detect any positive examples

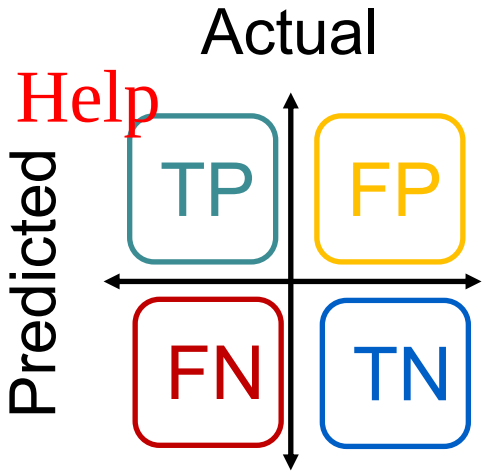
Performance Evaluation – Revision

- **Recall:**

$$\frac{TP}{TP + FN}$$

- **Precision:**

$$\frac{TP}{TP + FP}$$



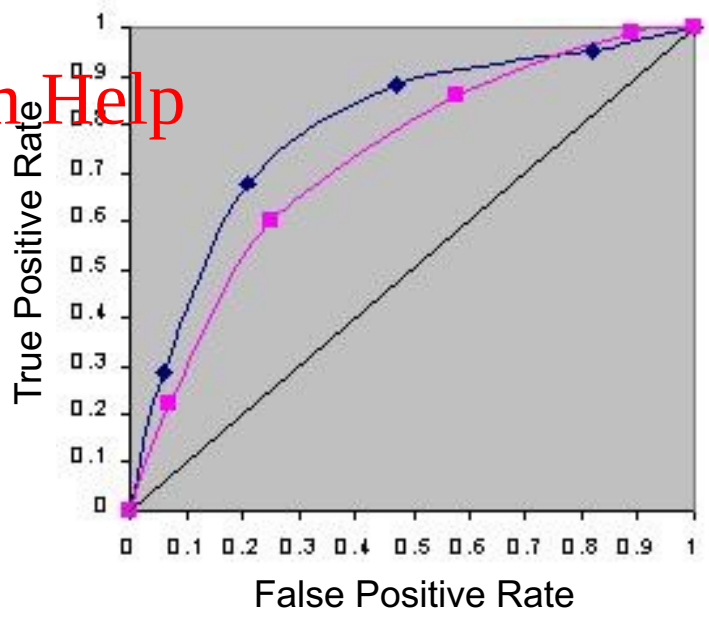
- **F-Score:**

$$(1 + \beta^2)\frac{Per \times Rec}{Rec + \beta^2 Per}$$

Performance Evaluation – Revision

ROC (Receiver Operating Characteristic) Curve

- ROC curve plots TPR (on the y-axis) against FPR (on the x-axis)
- Performance of each classifier represented as a point on the ROC curve
- (TPR,FPR):
  - (0,0): declare everything to be negative class
  - (1,1): declare everything to be positive class
  - (1,0): ideal
- Diagonal line:
  - Random guessing
  - Below diagonal line:
    - prediction is opposite of the true class

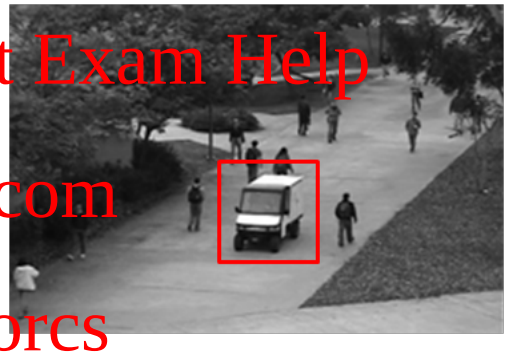


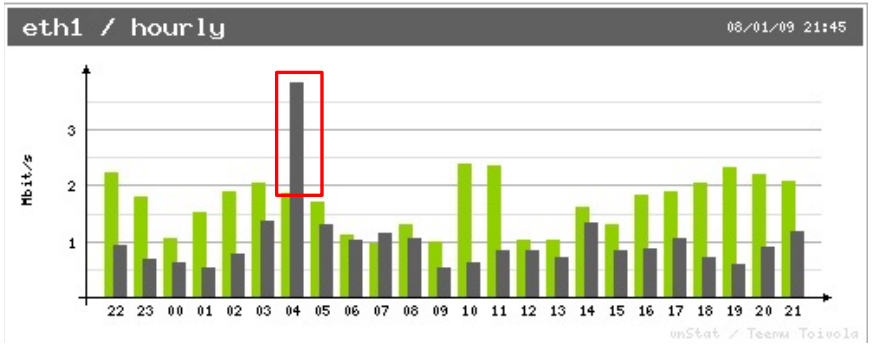
Anomaly (Outlier) Detection

- An anomaly is defined as a pattern in data that *does not conform to the expected behaviours*, including outliers, abbreviations, contaminants, and surprise, etc., in applications.

Packet List — Packets: 157 · Selected: 1

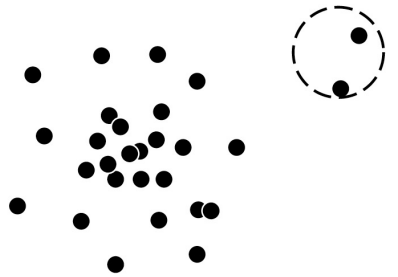
No.	Delta Time	Source	Destination	Protocol
1	0.000000	FC:AA	FF:FF:	ARP Request
2	0.028674	00:1C	FF:FF:	ARP Request
3	0.200358	FC:AA	33:33:	LLMNR
4	0.000001	192.16	224.0.(	LLMNR
5	0.083002	192.16	234.12	UDP
6	0.016290	FC:AA	33:33:	LLMNR
7	0.000001	192.16	224.0.(	LLMNR



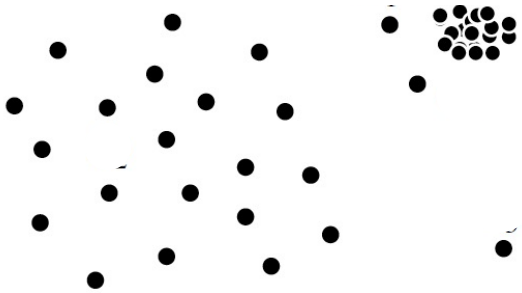


Types of Anomalies

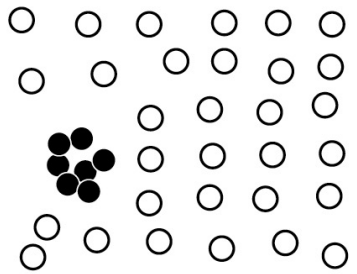
- **Global (Point) Anomalies:** A data object is a global outlier if it *deviates significantly from the rest of the data set*. To detect global anomalies, a critical issue is to find an appropriate measurement of deviation with respect to the application in question.



- **Contextual (Conditional) Anomalies:** A data object is a contextual anomaly if it *deviates significantly with respect to a specific context of the object*. In contextual anomaly detection, the context has to be specified as part of the problem definition.

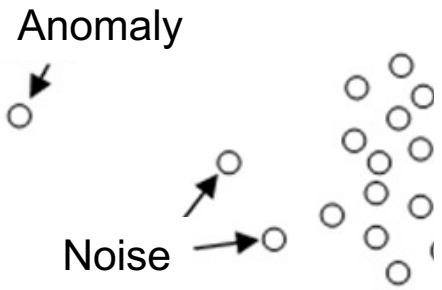


- **Collective Anomalies:** A *subset of data* objects forms a collective anomaly if the objects as *a whole deviate significantly from the entire data set*. Importantly, the individual data objects may not be anomalies.



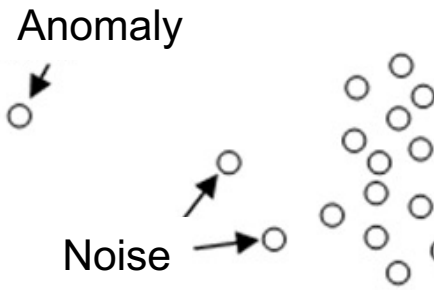
Anomaly, Noise and Novelty Detection

- **Noise vs. Anomaly:**
  - Noise is a random error or variance in an instance variable.
  - In general, noise is not interesting in data analysis, including anomaly detection.
  - Anomalies are interesting because they are suspected of not being generated by the same mechanisms as the rest of the data.

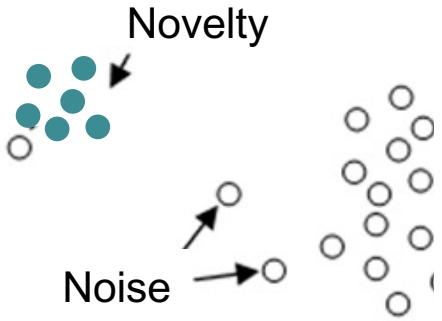


Anomaly, Noise and Novelty Detection

- **Noise vs. Anomaly:**
  - Noise is a random error or variance in an instance variable.
  - In general, noise is not interesting in data analysis, including anomaly detection.
  - Anomalies are interesting because they are suspected of not being generated by the same mechanisms as the rest of the data.



- **Novelty vs. Anomaly:**
  - In evolving datasets, novel patterns may initially appear as anomalies.
  - Once new patterns are confirmed, they are usually incorporated into the model of normal behaviour so that follow-up instances are not treated as anomalies anymore.



Anomaly Detection Schemes

General Steps

- Build a profile of the “normal” behaviour
  - Profile can be patterns or summary statistics for the overall population
- Use the “normal” profile to detect anomalies
  - Anomalies are observations whose characteristics differ significantly from the normal profile

Methods

1. Extreme Value Analysis
2. Proximity-Based
3. Model-based

1. Extreme Value Analysis

- Assume a parametric model describing the distribution of the data (e.g., normal distribution)
- Apply a statistical test (e.g., $z = (x - \mu)/\sigma$) that depends on
  - Data distribution
  - Parameter of distribution (e.g., mean, variance)
  - Number of expected anomalies (confidence limit)

Limitations

- Most of the tests are for a single attribute
- In many cases, data distribution may not be known
- For high dimensional data, it may be difficult to estimate the true distribution
  - Can be used as final steps for interpreting outputs of other anomaly detection methods

2. Proximity-Based

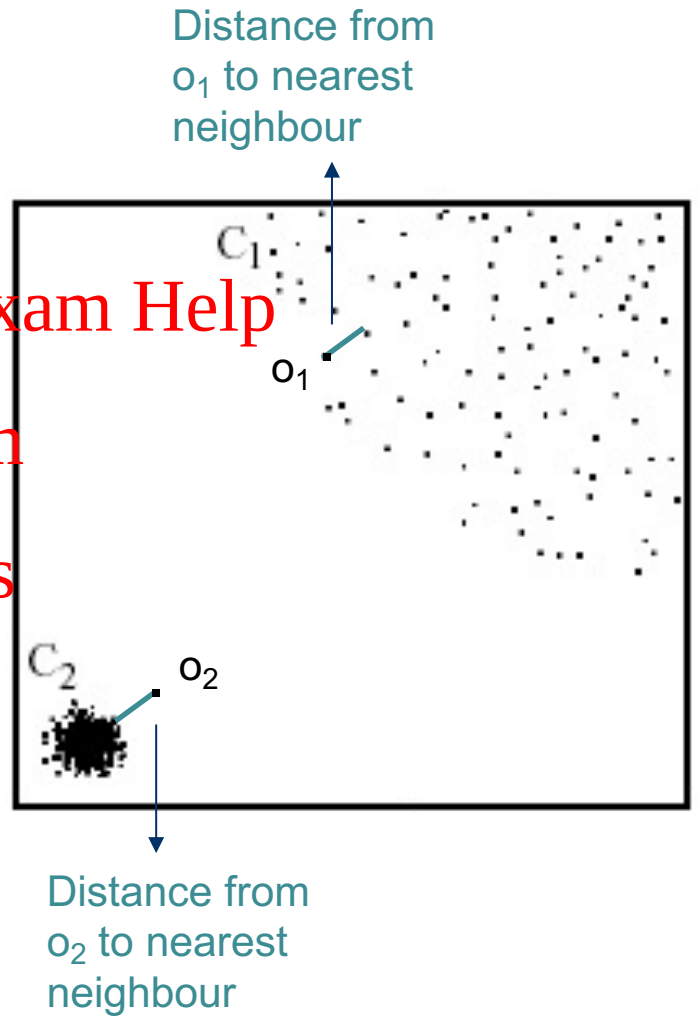
- Data is represented as a vector of features.
- Assumes the proximity of an anomaly to its neighbourhood significantly deviates from the proximity of the object to most of the other objects in the dataset.
- Three major approaches
  - 2.1 Nearest-neighbour based
  - 2.2 Density based
  - 2.3 Clustering based

2.1 Nearest-Neighbour Based

- Compute the distance between every pair of data points
- There are various ways to define anomalies:
  - Data points for which there are fewer than k neighbouring points within a distance D
  - The top n data points whose *distance* to the k^{th} nearest neighbour is greatest
  - The top n data points whose *average* distance to the k^{th} nearest neighbours is greatest

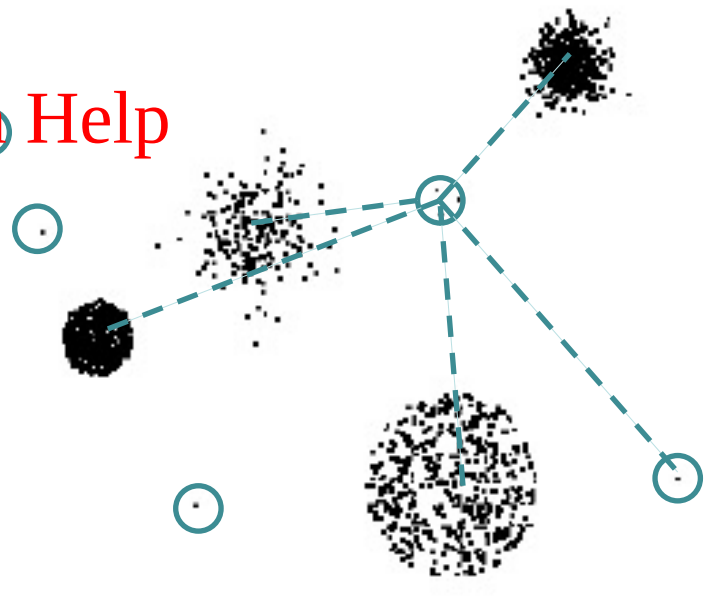
2.2 Density-based

- Estimates the density of objects (using proximity measures between objects).
- Objects that are in regions of low density are relative distant from their neighbours, and can be considered anomalous.
- A more sophisticated approach accommodates the fact that data sets can have regions of widely differing densities.
  - Classifies a point as an outlier only if it has a local density significantly less than that of most of its neighbours.



2.3 Clustering-Based

- Cluster the data into groups of different density
- Choose points in small cluster as candidate anomalies
- Compute the distance between candidate points and non-candidate clusters.
  - If candidate points are far from all other non-candidate points, they are anomalies



3.1 Classification-Based Methods

Idea: Train a classification model that can distinguish "normal" data from anomalies

- Consider a training set that contains samples **labelled** as "normal" and others **labelled** as "anomaly"
  - But, the training set is typically heavily biased: number of "normal" samples likely far exceeds number of anomaly samples
- Handle the imbalanced distribution
  - Oversampling positives and/or under sampling negatives
  - Cost-sensitive learning

3.2 One-Class Model

- One-class model: A classifier is built to describe only the normal class
  - Learn the decision boundary of the normal class using classification methods such as one-class SVM
  - Any samples that do not belong to the normal class (not within the decision boundary) are declared as anomalies
  - Advantage: can detect new anomalies that may not appear close to any anomalous objects in the training set

Output of Anomaly Detection

- **Scoring Techniques:**
  - Assign an anomaly score to each instance in the test data depending on the degree to which that instance is considered an anomaly.
  - The output is a ranked list of anomalies.
  - An analyst may choose to either analyse top few anomalies or use a cut-off threshold (or domain specific threshold) to select the anomalies.

- **Labelling Techniques:**
  - Assign a label (normal or anomalous) to each test instance.
  - Limit the analysts to the binary label, (though this can be controlled indirectly through parameter choices within each technique).

Challenges of Machine Learning in Security

- Modelling data with skewed class distributions (class imbalance)
- Sheer volume and heterogeneous network data
- Difficult to assess the performance of the system, given the vast possibilities of anomalies and lack of label
- Cost of error in IDS is huge
- Large false alarm rate degrades confidence in the system
- Lack of interpretability
- Anomalies may be undetectable at one level of granularity or abstraction but easy to detect at another level
- Evolving patterns (concept drift)

Isolation Forest (iForest) [3]

Isolation Tree (iTree)

- **Objective:** Isolates anomalies rather than profiles normal instances
- **Isolation:** Separating an instance from the rest of the instances

To achieve this, we takes advantage of two anomalies' quantitative properties:

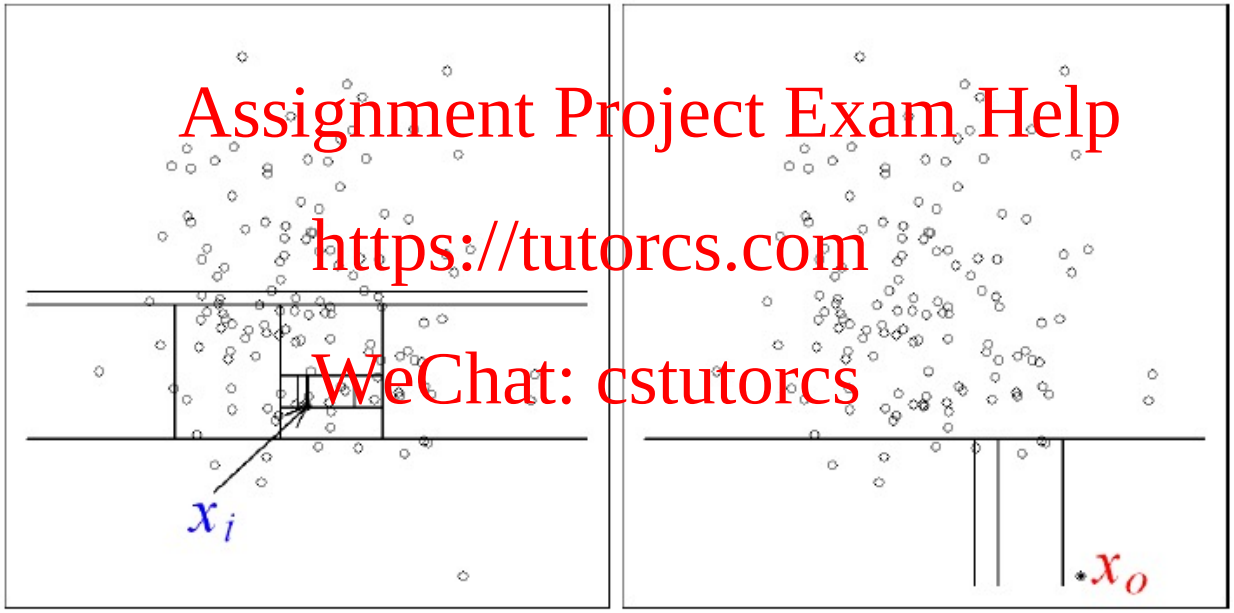
i. They are the minority consisting of fewer instances, and

ii. They have attribute-values that are very different from those of normal instances

- **Isolation Tree (iTree) Intuition:** Because of their susceptibility to isolation, anomalies are isolated *closer to the root* of the tree; whereas normal points are isolated at the *deeper end* of the tree.

iTree Intuition

- Anomalies are more susceptible to isolation under random partitioning

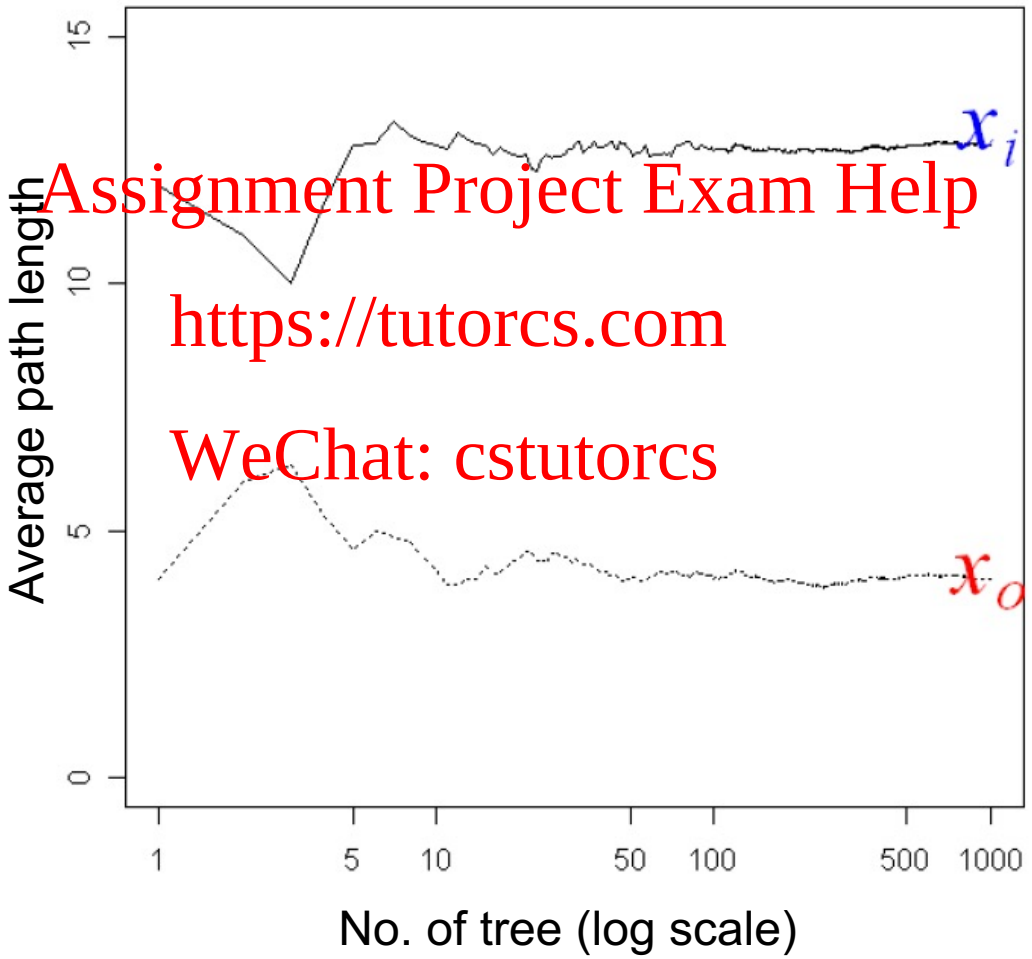


(a) x_i requires only 12 partitions (b) x_0 requires only 4 partitions

Figure. Identifying normal vs. abnormal observations

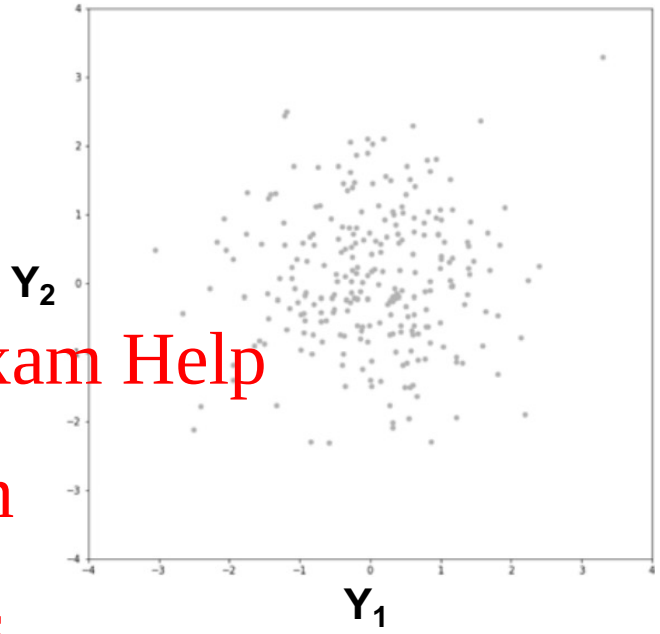
iTree Intuition

- Anomalies are more susceptible to isolation and hence have short path lengths



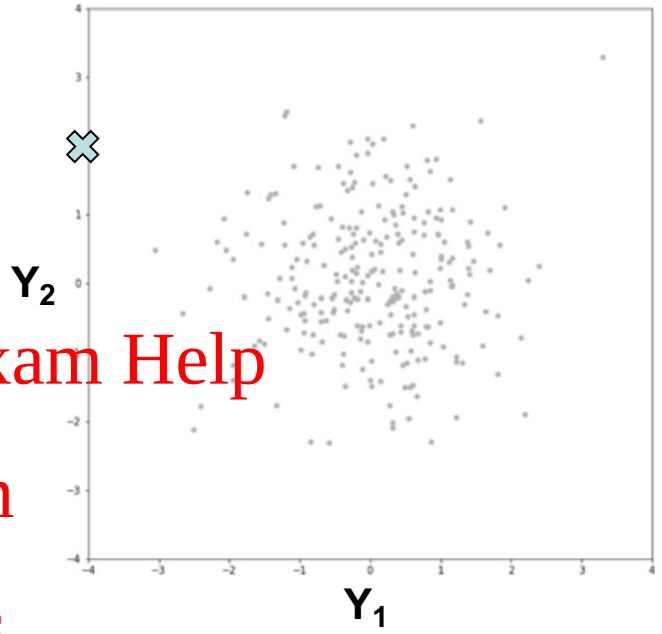
Isolation Forest

- For each tree:
- Get a sample of the data
  - Randomly select a dimension



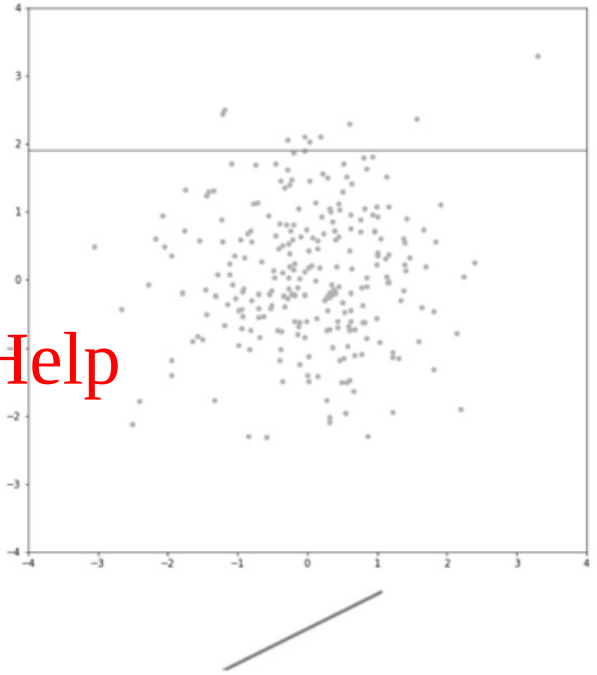
Isolation Forest

- For each tree:
- Get a sample of the data
  - Randomly select a dimension
  - Randomly pick a value in that dimension



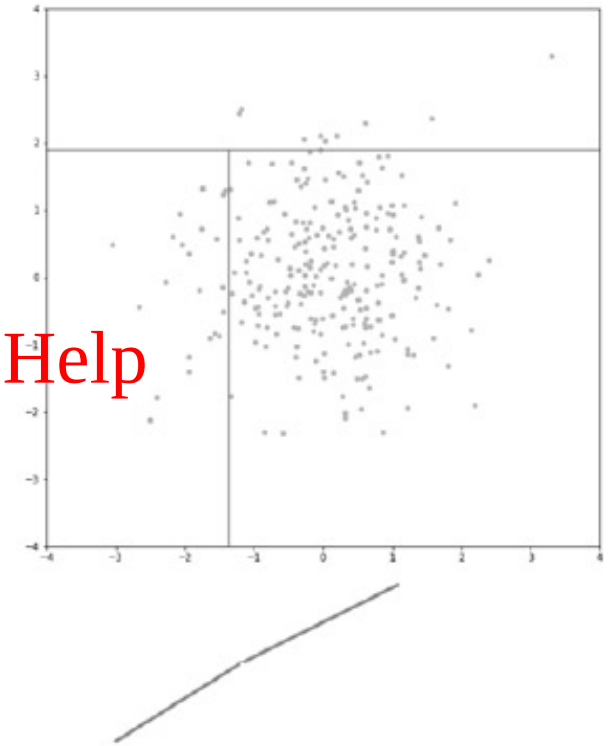
Isolation Forest

- For each tree:
- Get a sample of the data
  - Randomly select a dimension
  - Randomly pick a value in that dimension
  - Draw a straight line through the data at that value and split data



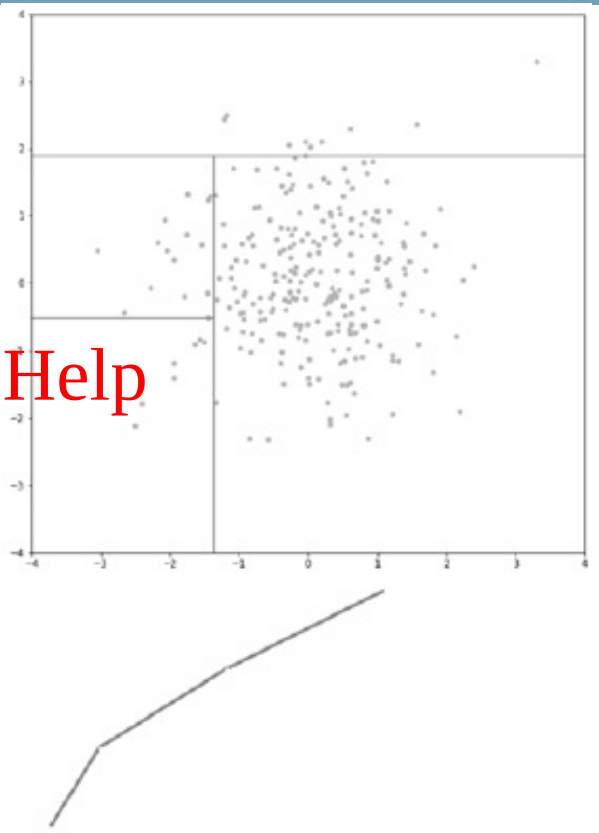
Isolation Forest

- For each tree:
- Get a sample of the data
  - Randomly select a dimension
  - Randomly pick a value in that dimension
  - Draw a straight line through the data at that value and split data
  - Repeat until tree is complete



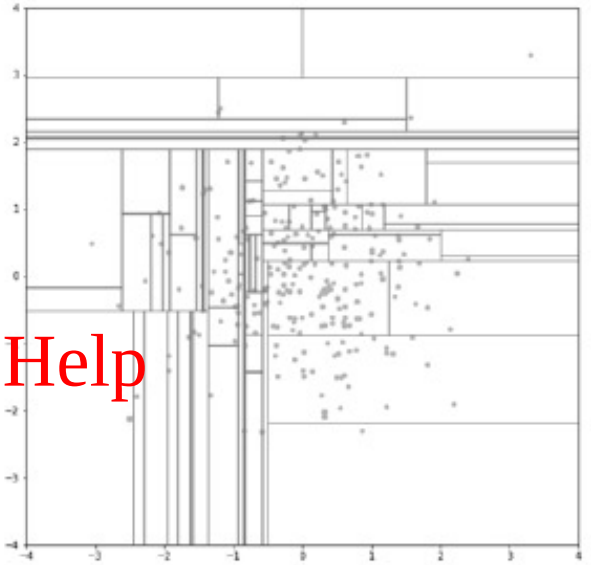
Isolation Forest

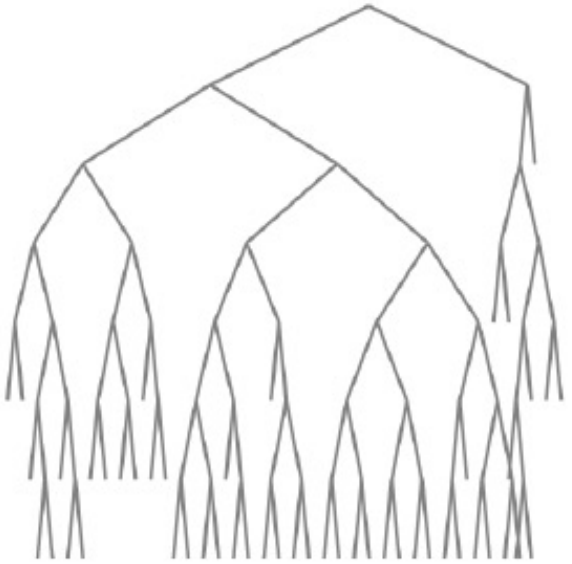
- For each tree:
- Get a sample of the data
  - Randomly select a dimension
  - Randomly pick a value in that dimension
  - Draw a straight line through the data at that value and split data
  - Repeat until tree is complete



Isolation Forest

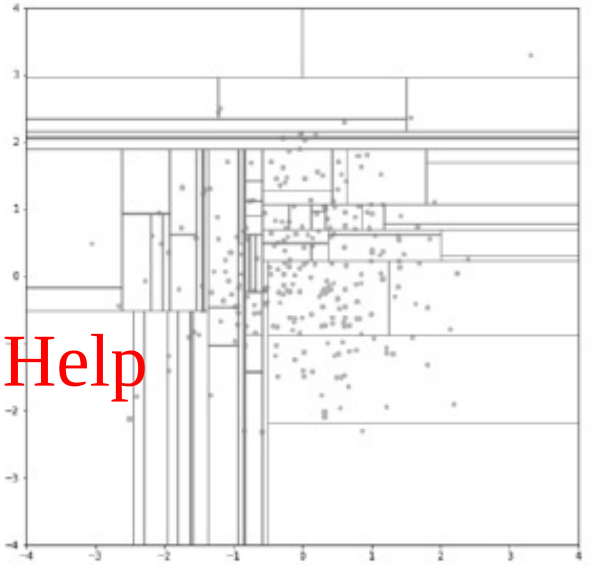
- For each tree:
- Get a sample of the data
  - Randomly select a dimension
  - Randomly pick a value in that dimension
  - Draw a straight line through the data at that value and split data
  - Repeat until tree is complete

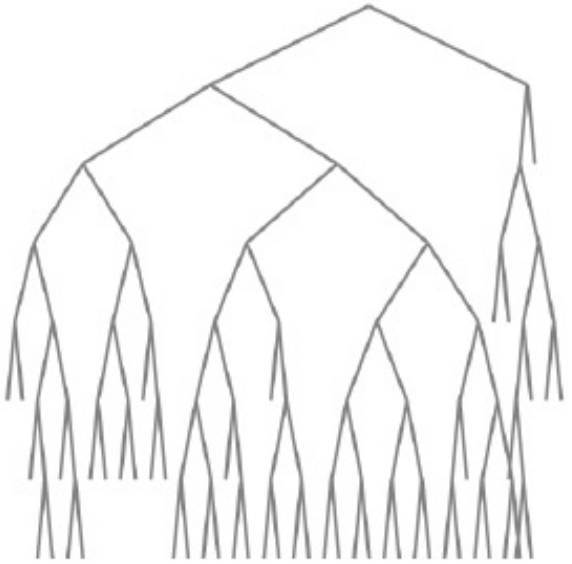




Isolation Forest

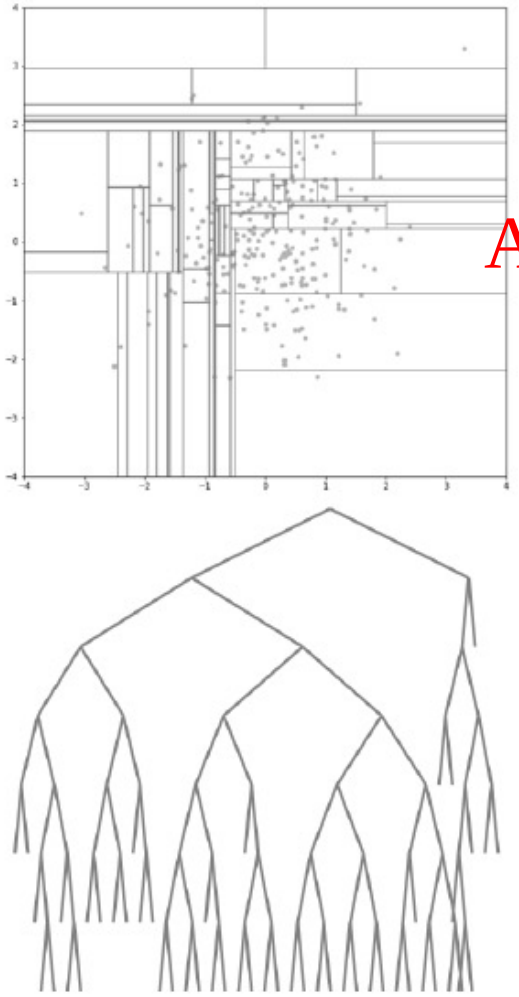
- For each tree:
- Get a sample of the data
  - Randomly select a dimension
  - Randomly pick a value in that dimension
  - Draw a straight line through the data at that value and split data
  - Repeat until tree is complete
- Generate multiple trees → forest





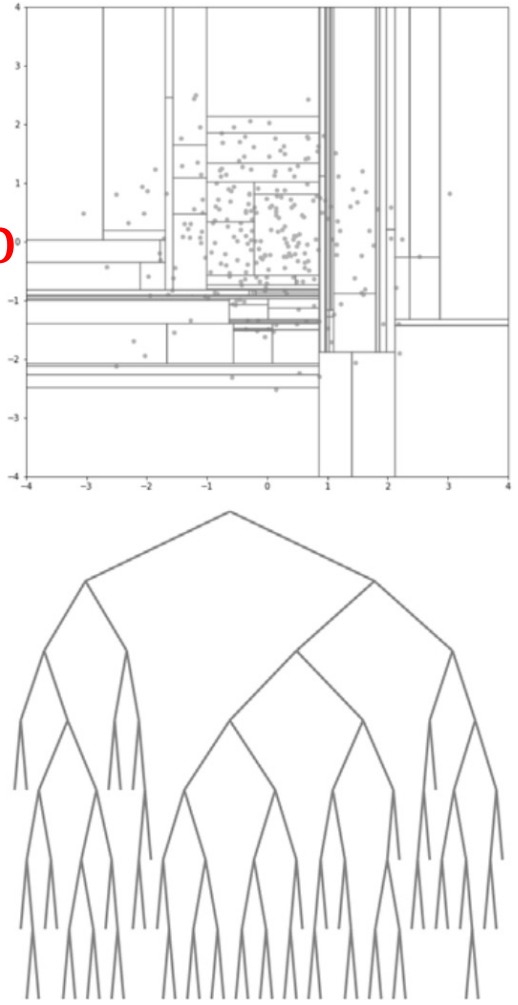
Isolation Forest

iTree 1



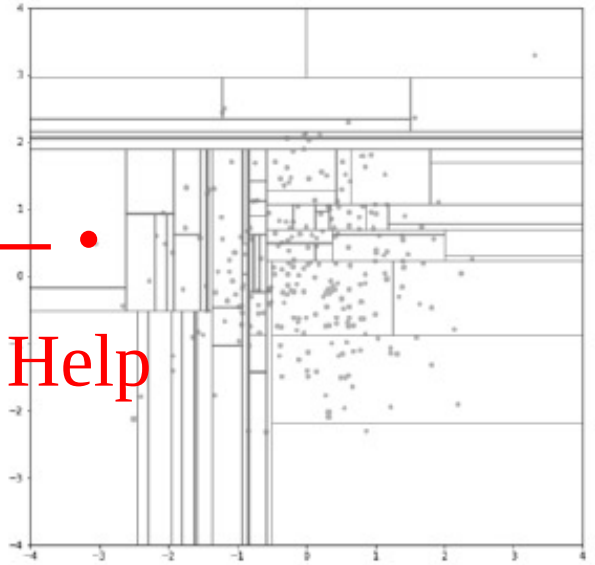
iTree 2

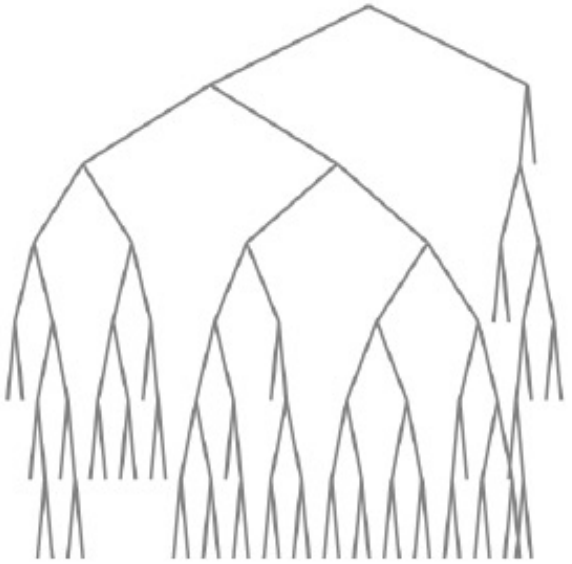
iTree 3



Isolation Forest

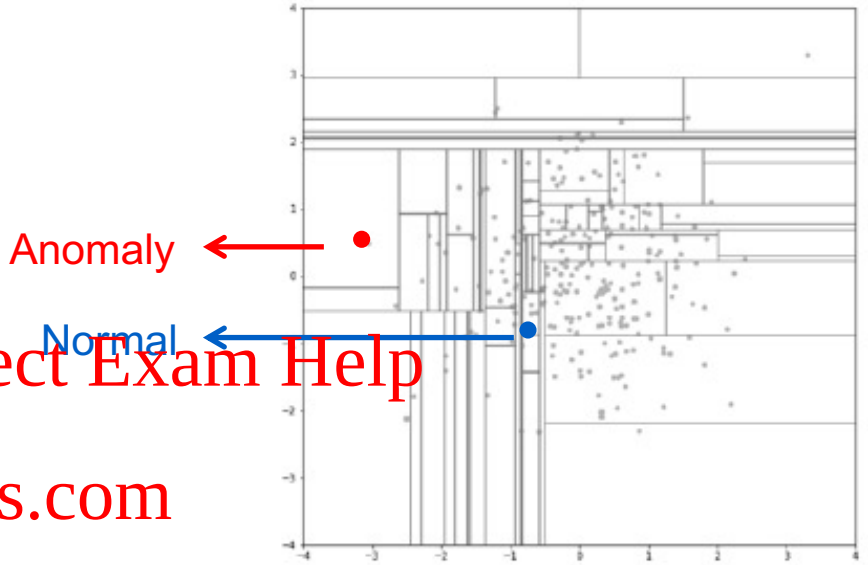
- For each tree:
- Get a sample of the data
  - Randomly select a dimension
  - Randomly pick a value in that dimension
  - Draw a straight line through the data at that value and split data
  - Repeat until tree is complete
- Generate multiple trees → forest
- Anomalies will be isolated in only a few steps

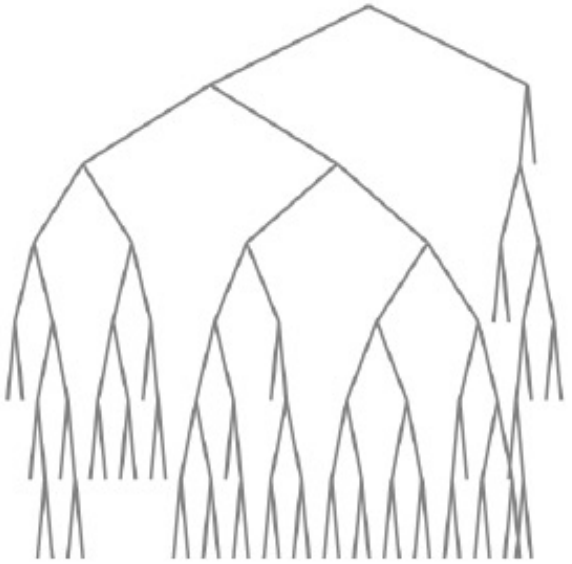




Isolation Forest

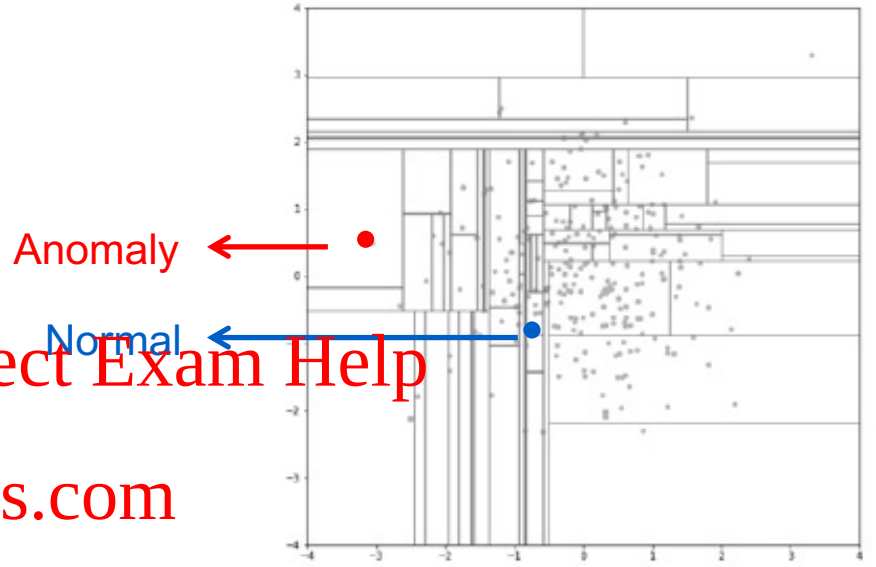
- For each tree:
- Get a sample of the data
  - Randomly select a dimension
  - Randomly pick a value in that dimension
  - Draw a straight line through the data at that value and split data
  - Repeat until tree is complete
- Generate multiple trees → forest
- Anomalies will be isolated in only a few steps
- Nominal points in more

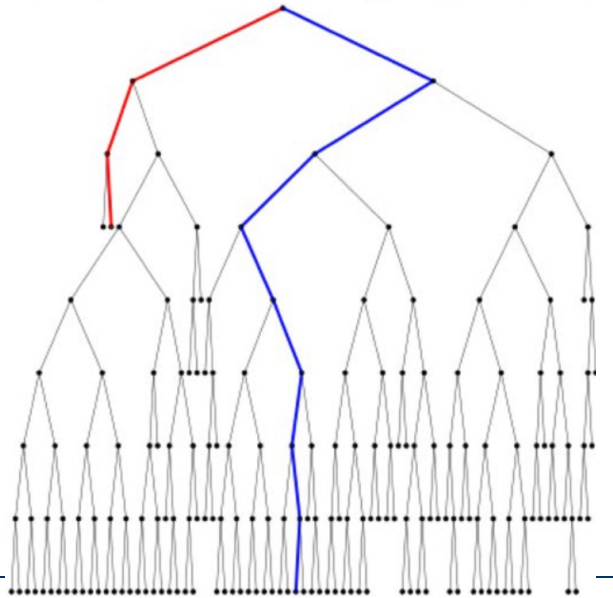




Isolation Forest

- For each tree:
- Get a sample of the data
  - Randomly select a dimension
  - Randomly pick a value in that dimension
  - Draw a straight line through the data at that value and split data
  - Repeat until tree is complete
- Generate multiple trees → forest
- Anomalies will be isolated in only a few steps
- Nominal points in more





Anomaly Detection with iForest

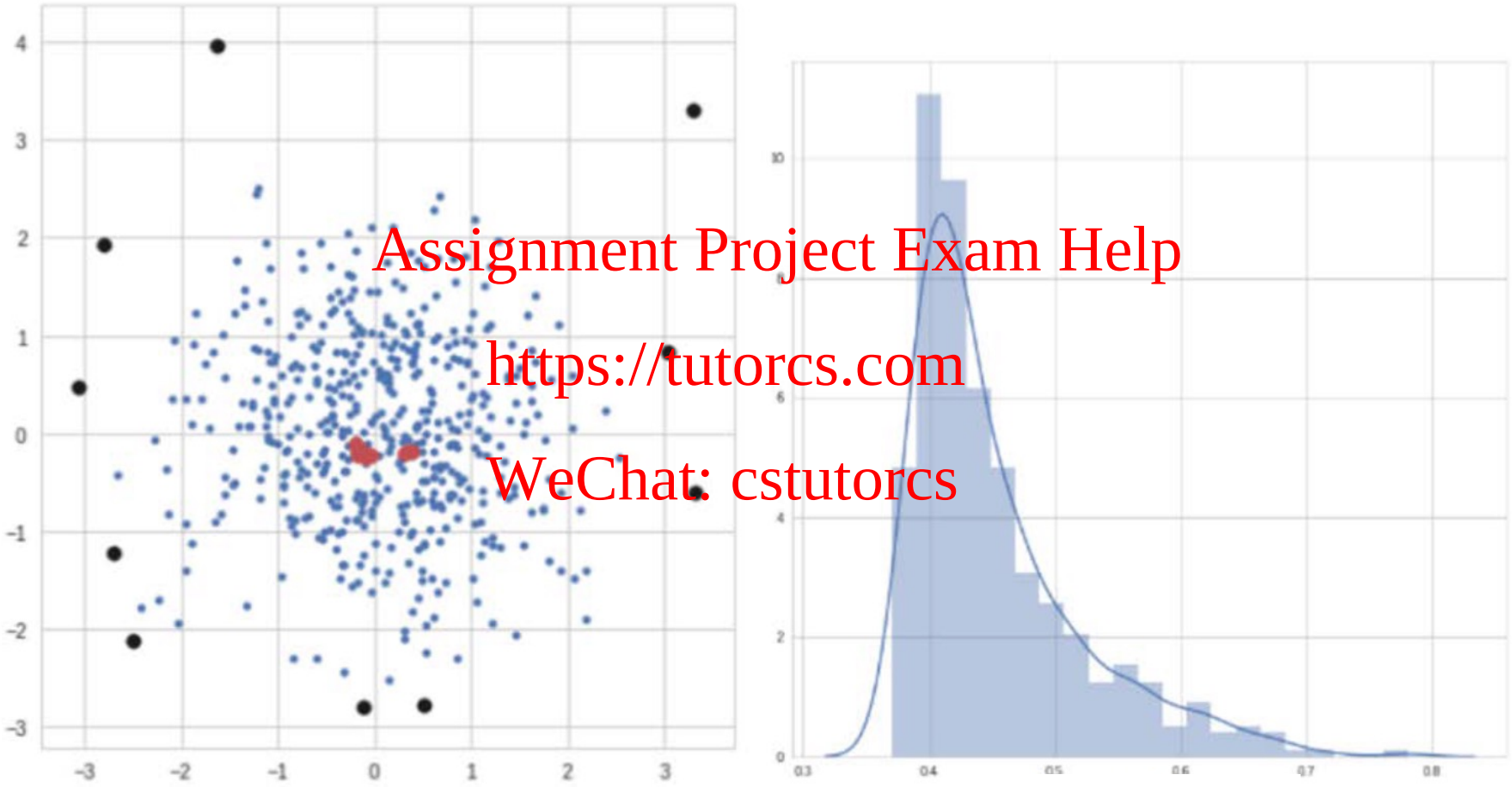
- Isolation Forest score:

$$s(x,n) = 2^{-\frac{E(h(x))}{c(n)}}$$

- Where,
  - $h(x)$ is the *path length* of observation x from the root node,
  - $\mathrm{E}(h(x))$ is the average of $h(x)$ from a collection of isolation trees
  - n is the number of data points
  - $c(n) = 2H(n-1) - (\frac{2(n-1)}{2})$, where Euler's constant

- $0 < s \leq 1$
  - $s \to 1$, then samples are definitely anomalies,
  - $s \ll 0.5$, then samples are quite safe to be regarded as normal,
  - $s = 0.5$, then the entire sample does not really have any distinct anomaly.

iForest Score – Case Study



Advantages of iForest

- Requires two parameters, the number of trees to build and the sub-sampling size
- Converges quickly with a very small number of trees, and it only requires a small sub-sampling size to achieve high detection performance with high efficiency
- The isolation characteristic of iTrees enables them to *build partial models* and exploit sub-sampling to an extent that is not feasible in existing methods.
- Utilizes no distance or density measures to detect anomalies.
- Has a linear time complexity with a low constant and a low memory requirement.
- Scales up to handle extremely large and high-dimensional datasets

Summary

- What is anomaly detection and what are different types of anomalies?
- How we can evaluation the performance of anomaly detection techniques?
- How anomaly detection is different from other machine learning problems?
- How does the iForest algorithm operates, and what are its advantages of this method?

Next: Clustering and Density-based Anomaly Detection

References

1. Data Mining and Machine Learning in Security, Chapters 1,3.
2. Machine Learning and Security, Chapter 1.
3. Fei Tony Liu, Kai Ming Ting, Zhi-Hua Zhou, “Isolation Forest”, IEEE International Conference on Data Mining, 2008.