The Internet of Things (IoT)

- Networks of physical objects (aka Things) that are uniquely identifiable in the Internet with embedded sensing and actuating along with programmability capabilities.

- Information about Things can be collected and the state of Things can be changed from anywhere, anytime, by anything!
The “last 100 meters” represent > 90% potential number of connections!
- The IoT has increased the number of connected devices by at least an order of magnitude!

https://www.slideshare.net/mazlan1/iot-and-big-data-the-perfect-marriage

Anything, Everything can be Measured!

Universality of IoT Technology

- The IoT is rapidly transforming the physical world into a large scale information system!

Digital Transformation of the Physical World

<table>
<thead>
<tr>
<th>Industry</th>
<th>Past: Shelling a Product</th>
<th>Future: a Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy &amp; utilities</td>
<td>Power networks/grids</td>
<td>On demand energy production/consumption</td>
</tr>
<tr>
<td>Automotive</td>
<td>Cars</td>
<td>Transportation (assisted, autonomous driving)</td>
</tr>
<tr>
<td>Agriculture</td>
<td>Seeds</td>
<td>Crop Yields</td>
</tr>
<tr>
<td>Healthcare</td>
<td>Diabetes pumps</td>
<td>Diabetes cares</td>
</tr>
<tr>
<td>Food</td>
<td>Packaged goods</td>
<td>Nutrition</td>
</tr>
<tr>
<td>Cities</td>
<td>Physical Urban infrastructure/Facilities</td>
<td>Smart city e-services (street lighting, urban noise/pollution/traffic monitoring, parking/waste management etc.)</td>
</tr>
<tr>
<td>.....</td>
<td>.....</td>
<td>.....</td>
</tr>
<tr>
<td>IT Industry</td>
<td>Computers</td>
<td>Computation</td>
</tr>
</tbody>
</table>

- McKinsey, GE, IBM, Cisco et al. estimate hundreds of billions dollar savings/efficiency improvements in the next 10 years
The Four Industrial Revolutions

- **INDUSTRY 1.0**
  - Mechanical production using the power of water and steam
- **INDUSTRY 2.0**
  - Centralized electric power infrastructure; mass production by division of labor
- **INDUSTRY 3.0**
  - Digital computing & communication technology, enhancing systems' intelligence
- **INDUSTRY 4.0**
  - Everybody & everything is networked – networked information as a "huge brain"

By 2008, there were more Internet-enabled devices than people in the world!

Massive Size and Growth of IoT

- **IoT Market Size (by 2025)**
  - McKinsey & Company: $6.1T
  - IDC: $7.1T
  - Cisco: $14.4T

- **Connected Devices (by 2020)**
  - Gartner: 26B
  - IDC: 32B
  - Cisco: 50B

- **Data Growth (2013 vs 2020)**
  - IDC: Total Data 4.4ZB => 44.4ZB (10x)
  - IoT Data: .09ZB => 4.4ZB (49x)

EMC IDC digital universe of opportunities

The IoT Data Avalanche

- The Internet of Things is more about data, than things!

- ~ 2x more data growth (currently @ 44EB/month) than social (human generated) & transactional/application (computer generated) data

- $10^{21}$ Zettabytes

- $10^{18}$ Exabytes

The IoT Value Chain

- IoT has the potential to transform how and when decisions are made throughout business and our daily lives iff high quality data can be processed efficiently and analyzed effectively!
What Makes IoT Data Processing & Analytics Challenging?

- Continuous data from multiple things in the wild
- Ingest and process data series
- Use multiple data analytics methods
- Data analytics distributed in several locations
- Integration with business operation systems
- Bidirectional communication and control of endpoints

High Volumes of Raw data in Motion
Complex Data Processing Real-time Control
& Analytics Pipelines and Automation

Adapted from https://www.gartner.com/smarterwithgartner/transform-your-business-with-iot-analytics/

IOT DATA CHARACTERISTICS
AND DATA PROCESSING ARCHITECTURES
Types of IoT Data

- **Addresses/Unique Identifiers** (nominal)
  - IP (IPv4, IPv6), Bluetooth, Zigbee, LoRa, RFID

- **Positional Data** (continuous)
  - GPS (Longitude, Latitude, Altitude), WiFi (Longitude, Latitude)

- **Temporal Data** (continuous)
  - Time and Date

- **Sensor Data** (numerical)
  - Output of a device that detects and responds to some type of input from the physical environment (motion, position, environment, mass, biomarker)

- **Descriptive Data** (categorical)
  - Objects, Processes, and Systems

---

IoT Data: Sensing the Physical World!

- IoT data is becoming soon “mega” big (i.e. billions of connected Things)
  - 44EB / months corresponds to ~ 100M hard disks
  - Processing 44 EB with 100M servers would still take > one hour

- IoT “data in motion” as opposed to traditional “data at rest”
  - Data speed is bound by the sensing frequency and connectivity
  - Data throughput (vs processing) has become the limiting factor!

- **High variety** data e.g., from heterogeneous Things, embedded in the environment or wearable by persons

- **High veracity** data (dropped or unlivable) as IoT data quality depends on how Things are used and connected in the wild

- **High variability** data as Things frequently change behavior dynamically and in ways that are not fully known in advance

- Data security, privacy, etc. even more important!
IoT Data Life Cycle

How to engender trust in the processes that use/consume the data?

IoT Data Quality (DQ)

- Assess the quality of IoT data created in the wild (vs. controlled setting)
- Various metrics indicating data suitability for decision making & planning
  - **Accuracy**: errors and noise depending on the sensing method calibration and the device resources constraints (power, memory)
  - **Validity**: abnormal data w.r.t. satisfy acceptance requirements
  - **Timeliness**: delayed due to limitations of the transmission protocol or of the processing infrastructure
  - **Completeness**: missing data due to device failures or intermittent communication as well as unevenly sampled data due to different sensing frequencies
  - **Integrity**: inconsistent data due to the dynamic IoT nature in adding or removing Things from existing deployments
  - **Trustworthiness**: maliciously altered data due to security breaches

IoT Data Analytics

- **Use Cases**
  - Real-time monitoring and control
  - Failure detection and predictive maintenance
  - Malicious cyber activity detection
  - Operational efficiency, optimization, self-adaptation

- **Batch and Online Data Processing Techniques**
  - Aggregation and statistics
  - Data series analytics (Spatiotemporal)
  - Machine Learning
    - Supervised: Classification, Regression, Anomaly/Outlier Detection
    - Unsupervised: Clustering, Freq. Itemset Mining, Anomaly/Outlier Detection
  - Deep Learning
  - Domain and data type specific
    - E.g., pressure sensor data from a pipeline system, acoustic sensor data sensors from an engine, vibration sensor data on a suspension bridge

IoT vs Traditional Big Data

- **Different scales of responsiveness**
  - For IoT, the time value of data is of the essence: It collects and uses data in real-time to track and monitor assets and be able to act promptly e.g., detect security breaches, correct malfunctions, etc.
  - Big data focus on the long-term: there is usually a lag between when the data is ingested and when it is analyzed e.g., to support business intelligence, capacity planning, etc.

- **Temporal and spatial dependencies**
  - IoT data come with time and location annotations, which is directly associated with their business value in a given application context and to be considered in IoT data processing

- **Diverging mindsets**
  - IoT targets intensive applications: effective and efficient data ingestion and analysis when and where it is needed
  - Big data targets extensive applications: faster collection and processing of more and more diverge data (4 V's)

What is Real Time Analytics?

Time value of sensor data means that the data you have right now won't mean as much a week, day or even hour from now!

A 3-tier Architecture for IoT Analytics

- To support analytics with different time horizons (near-real time, medium, long terms), we need to a 3-tier architecture for IoT

Insight increases as analytic capability & dataset size / quality increase

Responsiveness increases as network & analytic latencies decrease

Edge Processing/Computing [Cisco, Intel]

- **Objective**: push data processing away from the core and towards the edge of the network
  - help ensuring that the right data processing task takes place at the right time and place
- **Motivating factors**:
  1. **Reduce latency**: run data computations directly on IoT devices or gateways, and only interact with the Cloud off the critical path (e.g. to continuously train ML models with freshly data)
  2. **Be robust to connectivity issues**: applications are not disrupted in case of limited or intermittent network connectivity
  3. **Preserve privacy**: ensures that sensitive data is pre-processed *on-site*, and only data that is *privacy compliant* is sent to the Cloud for further analysis, after having passed through a first layer of anonymizing aggregation
- **Wide range of technologies**: Local Cloud/Fog computing, Grid/Mesh Computing, mobile edge computing, cloudlets, etc.

IoT Interaction Modes & Data Analytics

<table>
<thead>
<tr>
<th>Centralized</th>
<th>Decentralized</th>
<th>Federated</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CLOUD-CENTRIC</strong></td>
<td><strong>DEVICE-CENTRIC</strong></td>
<td><strong>EDGE, FOG &amp; CLOUD</strong></td>
</tr>
<tr>
<td>Cloud</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fog</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edge</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **for low-cost things** where data can easily migrate e.g. smart meters
- **for standalone things** with strict performance needs e.g. drones
- **for things producing large volumes of data** that are *difficult, costly or sensitive to migrate* e.g. autonomous cars

Distributed Learning

Distributed Test/Training DataSet
- Horizontally vs Vertically Partitioned Feature Space

Cloud-Based IoT Data Analytics

- Device (Local) Nodes
  - Generate raw data
  - Transmit raw data
  - Get trained model
  - Use the model in real-time

- Cloud (Coordinator) Node
  - Collects all data
  - Preprocesses data
  - Extracts features
  - Trains and periodically retrains a model
  - Transmit model to local nodes

- Variation: Local Preprocessing, Central Analysis
  - Local nodes preprocess raw data and extract features
  - Coordinator node gets these features to build a global model
  - Depending on the processing capabilities of local nodes and the particular learning task, such a design might be the only viable option
Edge-Based IoT Data Analytics

- **Edge (local) Nodes**
  - Collect data from local devices
  - Preprocess local data
  - Continuously train a local model
  - Transmit local model (along with feature values)
  - Make predictions online using local models

- **Variation: Fusion of local predictions**
  - Only the predictions are transmitted from local nodes, and fused at the coordinator according to a fusion rule (e.g., majority vote)

- **Cloud (Coordinator) Node**
  - Collects all local models
  - Fusion all local models to a global model
  - Make predictions offline using global models

Scaling Up IoT Data Analytics

- Classical ML approaches are designed to learn from a unique data set and, thus, to be applied to distributed data, they would require the collection of that data in a database for central processing

- A natural solution to scaling up ML algorithms is to allocate the learning process among several computational nodes
  - algorithm complexity and memory limitation are the main obstacles

- Two popular schemes of distributed learning algorithms:
  - **Controlled** (on the Cloud): data are artificially distributed between different computational nodes, while they can move data between them during the execution of a distributed learning algorithm
    - data parallel (horizontal) or model parallel (vertical) settings
  - **Ad hoc** (on the Edge/Fog): data are naturally distributed while moving them between computational nodes is not permitted
    - most distributed ML algorithms lay their foundations in ensemble learning
Summary

- The Internet of Things Is more About Data, than Things!
  - data in motion usually with spatiotemporal semantics
  - produced by several, distributed, heterogeneous devices
  - in the wild, hence data prone to errors and noise
  - data about people (in the consumer IoT), thus data security, privacy, etc. even more important!

- Post-cloud IoT data processing architectures: edge/fog computing for
  - effectively and efficiently ingesting and analyzing data when and where it is needed

- IoT Data Analytics use cases (e.g., real-time monitoring and control, correct malfunctions, detect security breaches, etc.) require streaming / distributed data analytics support
  - horizontal and vertical (or hybrid) distributed data samples/features
  - the size of ML models should be small enough to be used in edge devices and should also have a short inference time for real-time use

Anomaly/Outlier Detection in IoT
Data Outliers /Anomalies

- **What are anomalies/outliers?**
  - "An outlier is an observation in a dataset that that appears to be inconsistent with the remainder of that dataset" [Johnson 1992]
  - "An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism" [Hawkins 1980]

- **Anomaly, or outlier, or deviation, or novelty detection**, is an activity to measure whether a data point (or a subset) considerably differs than the remainder of the data points
  - On a scatter plot of the data, they lie far away from other data

- **Statistics-based intuition:**
  - Normal data follow a “generating mechanism”, e.g. some given statistical process
  - Abnormal data deviate from this generating mechanism

Cause of Anomalies/Outliers

- **Errors in the data collection or measurement process**
  - Because of human error (intentionally or not), a problem with measuring device or the presence of noise
  - Unreliable observations provide no interesting information but only reduce the quality of the data

- **Changes in the data generation process** (not an error, novelties in data)
  - Anomalous data because it is of a different type
  - Novel class of observations is valuable information

- **Insights extracted from “dirty data” are probably erroneous and thus the decisions to be made are likely unsound** (e.g., high rate of false positive and false negative)
  - The importance of reliability of data is even higher when exploited in applications which involve or affect human lives!
Example: Most Unusual Objects?

- screw
- bead
- battery
- block
- key
- knob
- dime
- eraser
- box
- penny
- battery

Example: How to Distinguish Objects?

- “Skinny” but not “smooth”
- No “Corners”
- Not “Round”

Key Insight: The “most unusual” object is different in some way from every partition of the features.
Example: How to Distinguish Objects?

- Humans used prior knowledge to select features that separated objects
  - “skinny”, “corners”, “round”, “smooth”
- Objects have been separated based on the chosen features
- Each cluster has been examined to see
  - which object fits less well in its cluster
  - and did not fit any other cluster
- Learning from humans: create features that capture object differences
  - Length/width:
    - $\geq 1 \Rightarrow \text{“skinny”}$
    - $= 1 \Rightarrow \text{“round”}$
    - $< 1 \Rightarrow \text{“tinny”}$
  - Number of surfaces:
    - distinct surfaces requires “edges” that have corners
  - Smooth: true or false

High-Dimensional Data: Example

<table>
<thead>
<tr>
<th>Object</th>
<th>Length/Width</th>
<th>Num Surfaces</th>
<th>Smooth</th>
</tr>
</thead>
<tbody>
<tr>
<td>penny</td>
<td>1</td>
<td>3</td>
<td>true</td>
</tr>
<tr>
<td>dime</td>
<td>1</td>
<td>3</td>
<td>true</td>
</tr>
<tr>
<td>knob</td>
<td>1</td>
<td>4</td>
<td>true</td>
</tr>
<tr>
<td>eraser</td>
<td>2.75</td>
<td>6</td>
<td>true</td>
</tr>
<tr>
<td>box</td>
<td>1</td>
<td>6</td>
<td>true</td>
</tr>
<tr>
<td>block</td>
<td>1.6</td>
<td>6</td>
<td>true</td>
</tr>
<tr>
<td>screw</td>
<td>8</td>
<td>3</td>
<td>false</td>
</tr>
<tr>
<td>battery</td>
<td>5</td>
<td>3</td>
<td>true</td>
</tr>
<tr>
<td>key</td>
<td>4.25</td>
<td>3</td>
<td>false</td>
</tr>
<tr>
<td>bead</td>
<td>1</td>
<td>2</td>
<td>true</td>
</tr>
</tbody>
</table>
High-Dimensional Anomaly Detection

- Anomaly detection in high-dimensional datasets reveals more accurate behavior of the data but at the same time poses various challenges
  - **Space concentration of distances**: the feature-wise distances of i.i.d. data samples approximately converge to a normal distribution
  - **The number of feature subspaces grows exponentially** with the increasing dimensionality of input data
  - **Data-snooping bias**: given enough alternative subspaces, at least one feature subspace can be found for each data point such that it appears as an anomaly

- **Challenge**: select a subspace that highlights relevant features, i.e. in which anomalies exhibit significantly different values from normal data
  - A robust anomaly detector should be able to detect anomalies with a high proportion of irrelevant attributes creating noise in the input data, which masks the true anomalies

---

Anomaly Detection Methods

- **Rare Class Mining**
  - New anomalies will be “similar” to some past anomaly

- **One Class**
  - Anomalies are “rare” in the reference dataset, and i.i.d.

- **IForest**
  - Outliers are "rare" and "isolated" w.r.t. inliers

---

- **Nearest Neighbor Based** (distance, density)
- **Clustering Based**

---


Random Splits

- Partition feature space using axis-parallel subdivisions
  - A random split is performed on a randomly selected attribute
  - The split point is a randomly selected value between the min/max values of the selected attribute
- Knowing that splits matter. The order of splits is unknown!

Isolation Forest [Liu et al. 2008]

- Each isolation tree is build from a random data subsample of size n
- Build an ensemble of binary trees from randomly selected subsamples

- Grow a random decision tree until each data point is in its own leaf
  - To score a data point, find the depth of the terminal node
  - The smaller the depth the more anomalous is the data
Isolation Forest Anomaly Scores

- Use average depth to compute the anomaly score:
  - 0 (normality) 1 (outlyingness)

- Score
  - Ensemble average path length to the point
  - Normalized by the expected path length of balanced binary search tree

\[
S(x) = 2 \frac{\mathbb{E}(h(x))}{c(n)}
\]

Isolation Forest Scores

(a) Isolating \( x_i \)
- 12 partitions (not an anomaly)

(b) Isolating \( x_o \)
- 4 partitions (anomaly)
iForest: Pros and Cons

- **Pros**
  - Very easy to construct (no distance function needed) avoiding hard decisions whether a data point is an anomaly or not
    - assigns an anomalous score to each of the testing data point
  - Achieve a **low linear time-complexity** and a **small memory-footprint**
    - By exploiting **subsampling**
    - By eliminating major computational cost of distance calculation in all the distance-based and density-based AD methods
  - Can provide **anomaly explanations** [Siddiqui et al. 2015]

- **Cons**
  - Hyper-parameter tuning (e.g. number/height of trees, sample size)
    - Large datasets will need more isolation trees (how many?)
  - Need extensions to handle discrete values and nominal data
  - Requires a **high percentage of relevant features to identify anomalies** [Bandaragoda et al. 2018]

Towards Distributed AD Methods

- **Communication cost per local node:** $O(km)$ where $k$ is the number of iTrees and $m$ the number of its leafs

- **Computation cost per local node:**
  - $O(km)$ for forest construction where $k$ is the number of iTrees and $m$ the number of its leafs and
  - $O(\log_2 m)$ for node scoring

One-Class Support Vector Machines
[Scholkopf et al, 1999]

- Instead of dividing data samples by class as SVM algorithms, 1CSVMs consider all samples as members of the normal class, and find the boundary that best fits the given data, dividing it from the origin
  - model the underlying distribution of normal data while being insensitive to noise or anomalies in the training data
  - after the training is finished, the testing simply involves classifying a sample as normal or anomalous depending on which side of the found boundary they are mapped

- A kernel function implicitly maps the input data space to a higher dimensional feature space in order to clearly separate between normal and anomalous data
  - When properly applied, in principle a kernel-based method is able to model any non-linear pattern of normal behavior

Various 1CSVM Methods

Find a hyperplane (in higher-dimensional kernel space) such that most of the data are separated from the origin with maximum margin

- PSVM (Plane-based Support Vector Machine)
- SVDD (Support Vector Data Description)
- QSSVM (Quarter-Sphere Support Vector Machine)
- CESVM (Centered Hyperellipsoidal Support Vector Machine)
### Performance Measures of AD Algorithms

<table>
<thead>
<tr>
<th>Confusion Matrix</th>
<th>Actual Normal Data ($n_n$)</th>
<th>Actual Anomalous Data ($n_a$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted Non-</td>
<td>TN</td>
<td>FN</td>
</tr>
<tr>
<td>Anomalies</td>
<td>FP</td>
<td>TP</td>
</tr>
<tr>
<td>Predicted</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anomalies</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Accuracy Rate (ACC)** = \( \frac{TP + TN}{TN + FP + FN + TP} \)
- **False Alarm –Positive– Rate (FAR)** = \( \frac{FP}{FP + TN} \)
- **True Positive –Detection– Rate (TPR)** = \( \frac{TP}{TP + FN} \)
- **Receiver Operating Characteristic (ROC)** = tradeoff between TPR & FAR
- **Area Under the ROC Curve (AUC)**: can be computed by a slight modification of the algorithm for constructing ROC curves

### ROC Spaces and AUC

- ROC curves are two dimensional plots in which the true positive (TP) rate is plotted on the Y axis and the false positive (FP) rate on the X axis.
- Area Under the ROC Curve (AUC) has an important statistical property:
  - The AUC of a classifier is equivalent with the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance.
  - When comparing classifiers, the bigger AUC the better!
- **AUC** = \( \frac{S - \frac{n_a^2 + n_a}{2}}{n_a \times n_n} \)
  \( \sum_{i=1}^{n_a} r_i \) where \( r_i \) is the rank of the i-th anomalous point list which is sorted by the anomaly score in ascending order, \( S \) is the sum of the ranks of the actual anomalous points.

[http://slideplayer.com/slide/5379166/]
Synthetic Datasets for AD Evaluation

J. Sopheap Semi-Supervised Outlier Detection Algorithms Ms Thesis UCL 2018

dataset1

Iforest vs 1CSVN: Outlier Score Contour Plots

dataset2
**iForest vs 1CSVN: Outlier Score Contour Plots**

**F₁ and AUC**

<table>
<thead>
<tr>
<th>Methods</th>
<th>F₁</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOF</td>
<td>0.99</td>
<td>0.9994</td>
</tr>
<tr>
<td>Robust PCA</td>
<td>1.00</td>
<td>1.0000</td>
</tr>
<tr>
<td>Autoencoder</td>
<td>1.00</td>
<td>1.0000</td>
</tr>
<tr>
<td>SOM</td>
<td>0.99</td>
<td>0.9994</td>
</tr>
<tr>
<td>One-Class SVM</td>
<td>0.98</td>
<td>0.9993</td>
</tr>
<tr>
<td>Isolation Forest</td>
<td>0.97</td>
<td>0.9977</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>F₁</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOF</td>
<td>0.98</td>
<td>0.9985</td>
</tr>
<tr>
<td>Robust PCA</td>
<td>0.90</td>
<td>0.9244</td>
</tr>
<tr>
<td>Autoencoder</td>
<td>0.80</td>
<td>0.8773</td>
</tr>
<tr>
<td>SOM</td>
<td>0.97</td>
<td>0.9994</td>
</tr>
<tr>
<td>One-Class SVM</td>
<td>0.98</td>
<td>0.9991</td>
</tr>
<tr>
<td>Isolation Forest</td>
<td>0.93</td>
<td>0.9967</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>F₁</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOF</td>
<td>0.96</td>
<td>0.9895</td>
</tr>
<tr>
<td>Robust PCA</td>
<td>0.91</td>
<td>0.9895</td>
</tr>
<tr>
<td>Autoencoder</td>
<td>0.89</td>
<td>0.9793</td>
</tr>
<tr>
<td>SOM</td>
<td>0.98</td>
<td>0.9967</td>
</tr>
<tr>
<td>One-Class SVM</td>
<td>0.96</td>
<td>0.9993</td>
</tr>
<tr>
<td>Isolation Forest</td>
<td>0.95</td>
<td>0.9942</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>F₁</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOF</td>
<td>0.90</td>
<td>0.9878</td>
</tr>
<tr>
<td>Robust PCA</td>
<td>0.86</td>
<td>0.9717</td>
</tr>
<tr>
<td>Autoencoder</td>
<td>0.87</td>
<td>0.9781</td>
</tr>
<tr>
<td>SOM</td>
<td>0.90</td>
<td>0.9680</td>
</tr>
<tr>
<td>One-Class SVM</td>
<td>0.92</td>
<td>0.9762</td>
</tr>
<tr>
<td>Isolation Forest</td>
<td>0.90</td>
<td>0.9896</td>
</tr>
</tbody>
</table>
Summary

- **Outlier** is a data point that deviates significantly from the rest of the points, as if it were generated by a different mechanism.
- **Anomaly detection** finds patterns in data that do not correspond to normal system behavior.
- **Novelty detection** aims to detect unobserved (emergent) patterns in data.
- Which anomaly detection method to use depends on:
  - **Data characteristics**: *dimensionality* (Univariate, Multivariate) and *type* (categorical numerical).
  - **Anomaly characteristics**: *type* (Point, Collective), *semantics* (Set-based, Sequence-based), *kind* (Binary, Score).
  - **Availability of labels**: *supervised*, *unsupervised*, *semi-supervised*.
  - **Algorithmic properties**: *computational cost* (exponential vs linear), *prior knowledge* (parametric, non-parametric), *distributional/incremental computation potential*, …

---

IoT Solutions of Major Cloud Providers

<table>
<thead>
<tr>
<th>Service</th>
<th>AWS</th>
<th>Microsoft</th>
<th>IBM</th>
<th>Google</th>
<th>Alibaba</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Collection</td>
<td>HTTP, WebSockets, MQTT</td>
<td>HTTP, AMQP, MQTT (and custom protocols, using protocol gateway project)</td>
<td>MQTT, HTTP</td>
<td>Google IoT</td>
<td>Alibaba IoT</td>
</tr>
<tr>
<td>Security</td>
<td>Link Encryption (TLS), Authentication (SIAV4, X.599)</td>
<td>Link Encryption (TLS), Authentication (Per-device with SAS tokens)</td>
<td>Link Encryption (TLS), Authentication (IBM Cloud SSO), Identity Management (LDAP)</td>
<td>Link Encryption (TLS)</td>
<td>Link Encryption (TLS)</td>
</tr>
<tr>
<td>Integration</td>
<td>REST APIs</td>
<td>REST APIs</td>
<td>REST and Real-time APIs</td>
<td>REST APIs, gRPC</td>
<td>REST APIs</td>
</tr>
<tr>
<td>Data Analytics</td>
<td>Amazon Machine Learning model (Amazon QuickSight)</td>
<td>Stream Analytics, Machine Learning</td>
<td>IBM Bluemix Data Analytics</td>
<td>Cloud Dataflow, BigQuery, Datalab, Dataprep</td>
<td>ManufCloud</td>
</tr>
</tbody>
</table>
Questions?

Bibliography

https://iot.ieee.org/definition.html

- IEC IoT 2020: Smart and secure IoT platform Geneva, Switzerland 2016
- S. Weber, R. Wong. The new world of data: Four provocations on the Internet of Things

- Edge Computing & Analytics
  - J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, W. Zhao. A Survey on Internet of Things:
  Architecture, Enabling Technologies, Security and Privacy, and Applications. IEEE Internet of
  Things Journal, 2017. ISSN 2327-4662.
    Commun. Rev. 45, 5, September 2015, 37-42
  - J. Ren, H. Guo, C. Xu, Y. Zhang. Serving at the Edge: A Scalable IoT Architecture Based on
  - M. Stolpe. 2016. The Internet of Things: Opportunities and Challenges for Distributed Data
  - M. Satyanarayanan, P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, W. Hu, B. Amos. Edge
  - E. Gaura, J. Brusey, M. Allen, R. Wilkins, D. Goldsmith, R. Rednic. Edge Mining the Internet
Bibliography


Big Sensor Data Applications


Anomaly Detection

- C. Aggarwal. Outlier Analysis. 2013 Springer.
- A. Muallem, S. Shetty, J. Pan, J. Zhao, B. Biswal Hoeffding Tree Algorithms for Anomaly Detection in Streaming Datasets: A Survey JIS. Vol.8 No.4, October 2017
Bibliography

- A. Siddiqui, A. Fern, T. Dietterich, W.-K. Wong Sequential Feature Explanations for Anomaly Detection. Workshop on Outlier Definition, Detection, and Description (ODDx3), August 2015, Sydney, AU

Related Tutorials

- IoT Analytics from Edge to Cloud - using IBM Informix
  https://www.slideshare.net/mlprague/adam
- G. Schmutz Introduction to Streaming Analytics https://www.slideshare.net/gschmutz/introduction-to-streaming-analytics
- Beyond Distributed ML Algorithms: Techniques for Learning from Large Data Sets https://iwinger.wordpress.com/2015/10/06/techniques-for-learning-from-large-amounts-of-data/

Datasets:
- IoT datasets available on data.world https://data.world/datasets/iot
Anomaly Detection Software Libraries

- Collection of anomaly detection examples: https://github.com/shubhomoydas/ad_examples
  - Isolation Forest (iForest) implementation
    - In Python: https://github.com/mgckind/iso_forest
    - In spark: https://github.com/titicaca/spark-forest
    - In Java: https://github.com/bnjmn/weka/blob/master/packages/internal/isolationForest/src/main/java/weka/classifiers/misc/IsolationForest.java
    - In Go: https://github.com/rieXpertSolutions/go-forest
  - One Class SVN (1CSVN) implementation
    - in R: https://gumroad.com/l/nbri (download the supplemental zip file at http://univprofblog.html.xdomain.jp/code/R_scripts_functions.zip)
    - In Python: https://gum.co/oPLZ (download the supplemental zip file at http://univprofblog.html.xdomain.jp/code/supportingfunctions.zip)
    - In Java: https://github.com/jnioche/libsvm-java
  - Unsupervised Anomaly detection algorithms
    - In Rapid Miner: http://madm.dfki.de/rapidminer/anomalydetection
  - Anomaly Detection using One-Class Neural Networks: https://github.com/raghavchalapathy/ocnn

License

- These slides are made available under a Creative Commons Attribution-ShareAlike license (CC BY-SA 3.0): http://creativecommons.org/licenses/by-sa/3.0/
- You can share and remix this work, provided that you keep the attribution to the original authors intact, and that, if you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one
Interpretability of ML Algorithms