



2 Days Training on IoT Architecture and Simulation using ns-3

CHAPTER 11 – Data Analytics and Machine Learning in Cloud and Fog Computing

Muhammad Saufy Rohmad
EE, UiTM
CompuThings




Introduction

- The value of an IoT system is not a single sensor event, or a million sensor events archived away.
- A significant value of IoT is in the interpretation and decision made of that data.



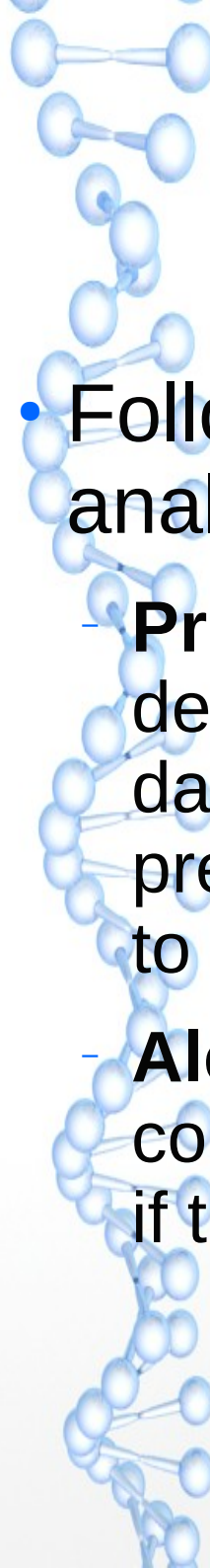
Introduction(2)

- Analytics for the IoT segment deal with:
 - Structured data (SQL storage), a predictable format of data.
 - Unstructured data (raw video data or signals), a high degree of randomness and variance.
 - Semi-structured (Twitter feeds), some degree of variance and randomness in form.



Basic Data Analytics in IoT

- Data analytics intend to find events, usually in a streaming series of data.
- There are multiple types of events and roles that a real-time streaming analysis machine must provide



Basic Data Analytics in IoT

- Following is an enumerated listing of these analytic functions:
 - **Preprocessing:** Filter out events of little interest, denaturing, feature extraction, segmentation, transform data to a more suitable form (although data lakes prefer no immediate transformation), adding attributes to data such as a tag (data lakes do need tags).
 - **Alerting:** Inspect data; if it exceeds some boundary condition, then raise an alert. The simplest example is if the temperature rises above a set limit on a sensor.




Basic Data Analytics in IoT(2)

- Following is an enumerated listing of these analytic functions:
 - **Windowing:** A sliding window of events is created that only draws rules upon that window. Windows can be based on time (for example, one hour), or length (2000 sensor samples). They can be sliding windows (for example, inspect only the 10 latest sensor events and produce a result whenever a new event arises), or batch windows (for example, produce an event only at the end of the window). Windowing is good for rules and for counting events. One could look for the number of temperature spikes in the last hour and resolve that a defect will occur on some machine.



Basic Data Analytics in IoT(3)

- Following is an enumerated listing of these analytic functions:
 - **Joins:** Combine multiple data streams into a new single stream. A scenario where this applies is a logistics example. Say a shipping company tracks their shipments with assets tracking beacons and that their fleet of trucks, planes, and facilities have geolocation information streaming as well. There are initially two streams of data: one for the package, and one for a given truck. When a truck picks up a package, those two streams become joined.



Basic Data Analytics in IoT(4)

- Following is an enumerated listing of these analytic functions:
 - **Errors:** Millions of sensors will generate missing data, garbled data, and data that is out of sequence. This is important in the IoT case with multiple streams of asynchronous and independent data. For example, data may be lost in a cellular WAN if a vehicle enters an underground parking garage. This analytic pattern correlates data within its own stream to attempt to find these error conditions.



Basic Data Analytics in IoT(5)

- Following is an enumerated listing of these analytic functions:
 - **Databases:** The analytics package will need to interact with some data warehouse. For example, if data is streaming in from a number of sensors looking, or in particular, when Bluetooth asset tags if an item is stolen or lost, a database of missing tag IDs would be referenced from all the gateways streaming in tag IDs to the system.: The analytics package will need to interact with some data warehouse. For example, if data is streaming in from a number of sensors looking, or in particular, when Bluetooth asset tags if an item is stolen or lost, a database of missing tag IDs would be referenced from all the gateways streaming in tag IDs to the system.



Basic Data Analytics in IoT(6)

- Following is an enumerated listing of these analytic functions:
 - **Temporal events and patterns:** This is most often used with the window pattern mentioned previously. Here, a series or sequence of events constitutes a pattern of interest. One can think of this as a state machine. Say we are monitoring the health of a machine based on temperature, vibrations, and noise. A temporal event sequence could be as follows:
 - 1. Detect if the temperature exceeds 100° C
 - 2. Then detect if vibrations exceed 1 m/s
 - 3. Next, detect if the machine is emitting noise at 110 dB
 - 4. If those events take place in that sequence, only then raise an alert



Basic Data Analytics in IoT(7)

- Following is an enumerated listing of these analytic functions:
 - **Tracking:** Tracking involves when or where something exists, an event occurred, or when something doesn't exist where it should have. A very basic example is geolocation of service trucks where a company may need to know exactly where a truck is, and when it was last there. This has application in agriculture, human movement, tracking patients, tracking high-value assets, luggage systems, smart city garbage, snow removal, and so on.



Basic Data Analytics in IoT(8)

- Following is an enumerated listing of these analytic functions:
 - **Trends:** This pattern is particularly useful for predictive maintenance. Here, a rule is designed to detect an event based on time-correlated series data. This is similar to temporal events, but differs in the sense that temporal events have no notion of time, only sequence order. This model uses time as a dimension in the process. A running history of time-correlated data could be used to find patterns like a livestock sensor in farming. Here, a head of cattle may wear a sensor that detects the animal movement and temperature. An event sequence can be constructed to see if the cattle moved in the last day. If there was no movement, the cattle may be sick or dead.



Basic Data Analytics in IoT(9)

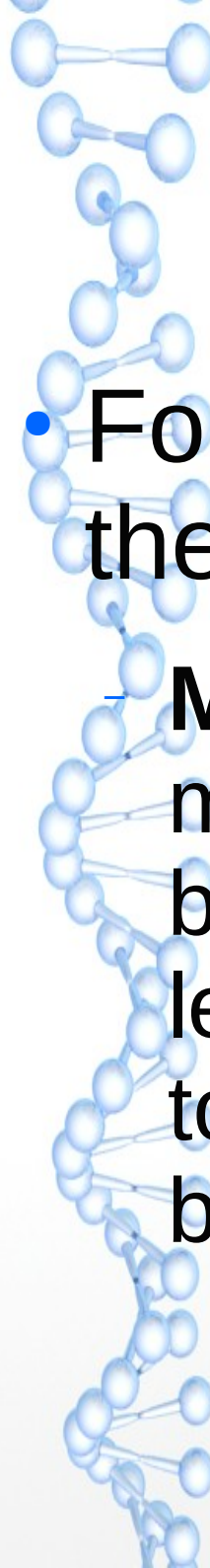
- Following is an enumerated listing of these analytic functions:
 - **Batch queries:** Batch processing typically is more comprehensive and deeper than real-time stream processing. A well-designed streaming platform can fork analysis and call into a batch processing system.



Basic Data Analytics in IoT(10)

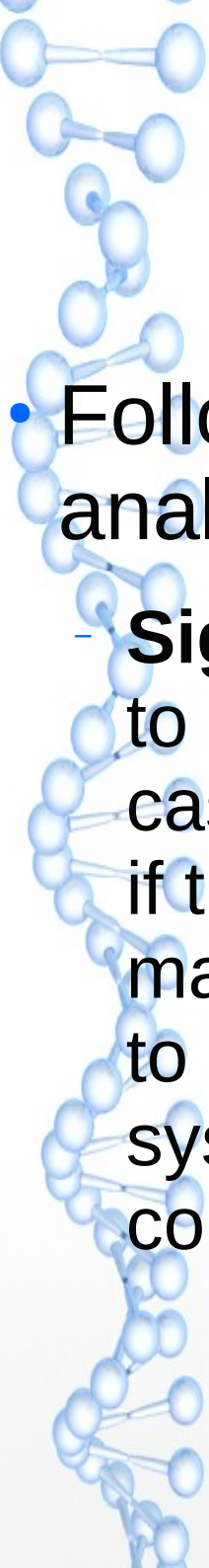
- Following is an enumerated listing of these analytic functions:

Deep analytics pathway: In real-time processing, we make a decision on the fly that some event has occurred. Whether or not that event really should signal an alarm may require further processing that will not operate in real time. This works because these events should be rare, and pass down information to a detailed analysis engine, while new events streaming in real time should be designed within a system. An example is a video surveillance system. Say a smart city issues an amber alert for a lost child. The smart city can issue a simple feature extraction and classification model for the real-time streaming engines. The model would detect license plates for a vehicle the child may be in, or potentially a logo on the child's shirt.



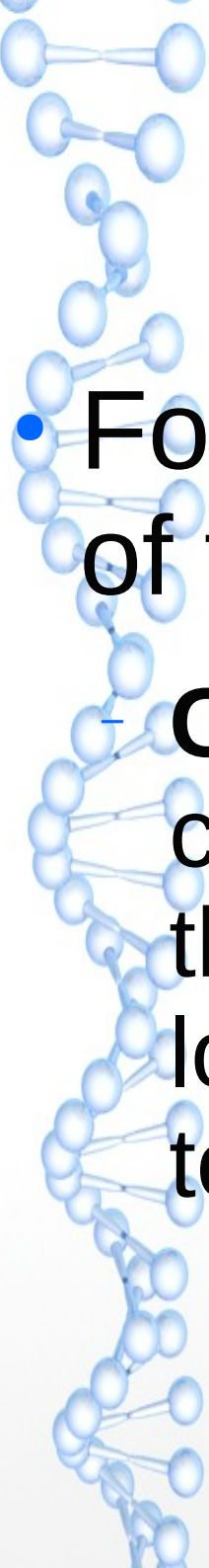
Basic Data Analytics in IoT(11)

- Following is an enumerated listing of these analytic functions:
 - **Models and training:** The first-level model described previously may, in fact, be an inference engine for a machine learning system. These machine learning tools are built on trained models that can be used for in-flight, real-time analysis.



Basic Data Analytics in IoT(12)

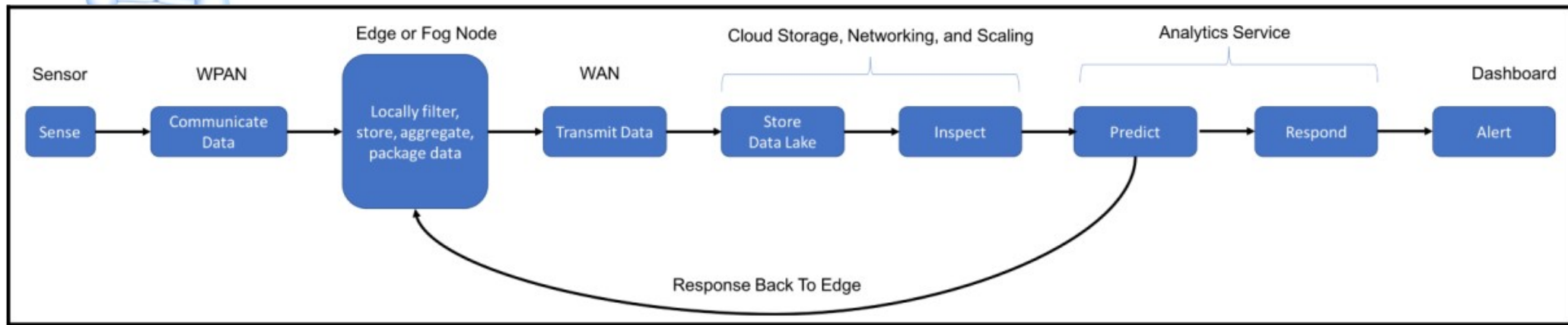
- Following is an enumerated listing of these analytic functions:
 - **Signaling:** It is often the case that an action needs to propagate back to the edge and sensor. A typical case is factory automation and safety. For example, if the temperature rises beyond a certain limit on a machine, log the event, but also send a signal back to the edge device to slow the machine down. The system must be able to be bidirectional in communication.



Basic Data Analytics in IoT(13)

- Following is an enumerated listing of these analytic functions:
 - **Control:** Finally, we need a way to control these analysis tools. Whether that is starting, stopping, reporting, logging, or debugging, facilities need to be in place to manage this system

IoT Pipeline





Analytics Form

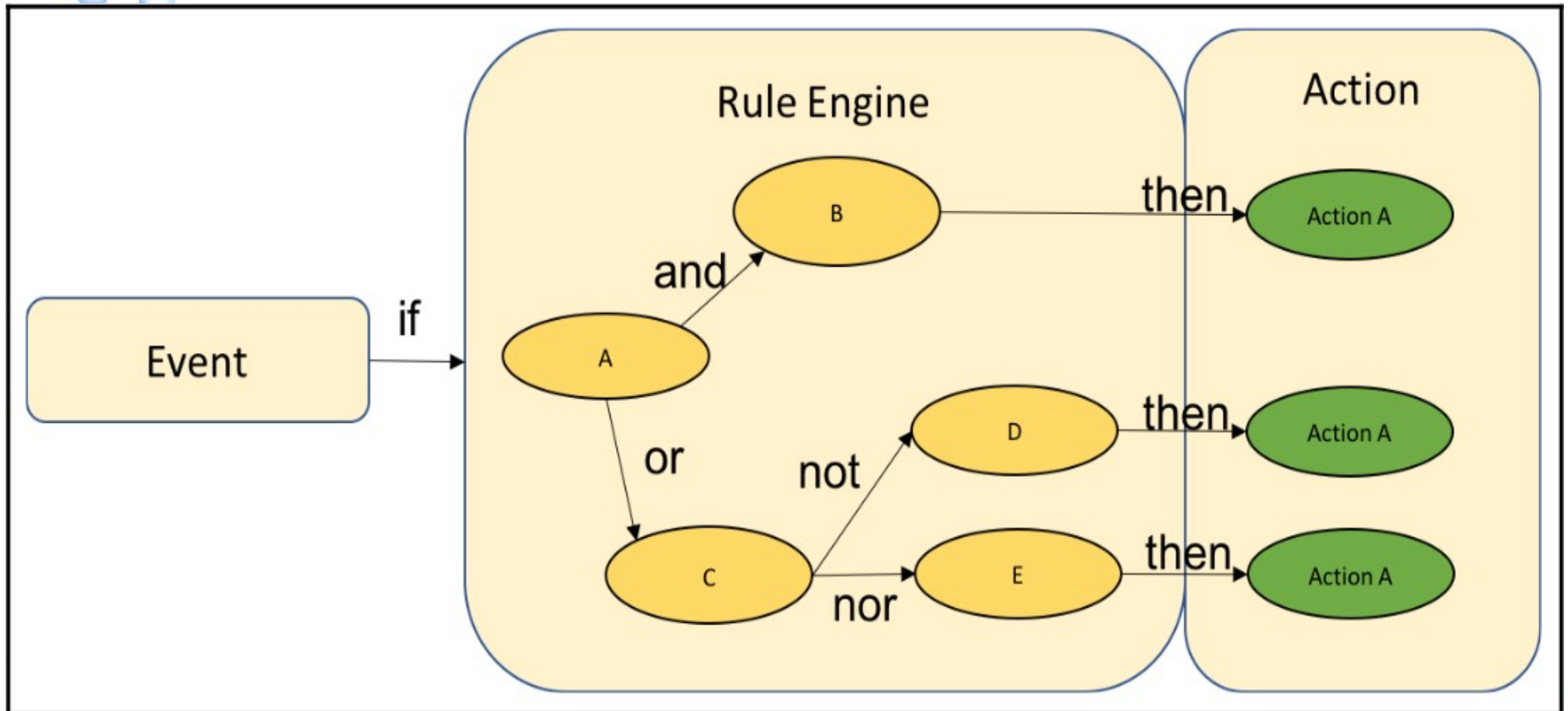
- The analytics (predict-respond) portion of the cloud can take on several forms:
 - Rules engines
 - Stream processing
 - Complex event processing:
 - Lambda architecture



Rule Engines

- A rule engine is simply a software construct that executes actions on events. For example, if the humidity in a room exceeds 50%, send an SMS message to the owner.
- Rules engines may or may not have state and be called stateful.
- That is, it may have a history of the event, and take different actions depending on the order, the amount, or the patterns of events as they occurred historically.

Rule Engines(2)





Stream Processing

- An IoT device is usually associated with some sensor or a device whose purpose is to measure or monitor the physical world. It does so asynchronously with respect to the rest of the IoT technology stack. That is, a sensor is always attempting to broadcast data, whether or not a cloud or fog node is listening.
- The IoT stream from a sensor to a cloud is assumed to be:
 - Constant and never-ending
 - Asynchronous
 - Unstructured, or structured
 - As close to real-time as possible



Complex Event Processing

- Complex event processing (CEP) is another analytic engine that is often used for pattern detection.
- From its roots in discrete event simulation and stock market volatility trading in the 1990s, it is by nature, a method capable of analyzing a live feed of streaming data in near real time.
- As hundreds and thousands of events enter the system, they are reduced and distilled into higher-level events.
- These are more abstract than raw sensor data. CEP engines have the advantage of a fast turnaround time in real-time analysis over a stream processor.



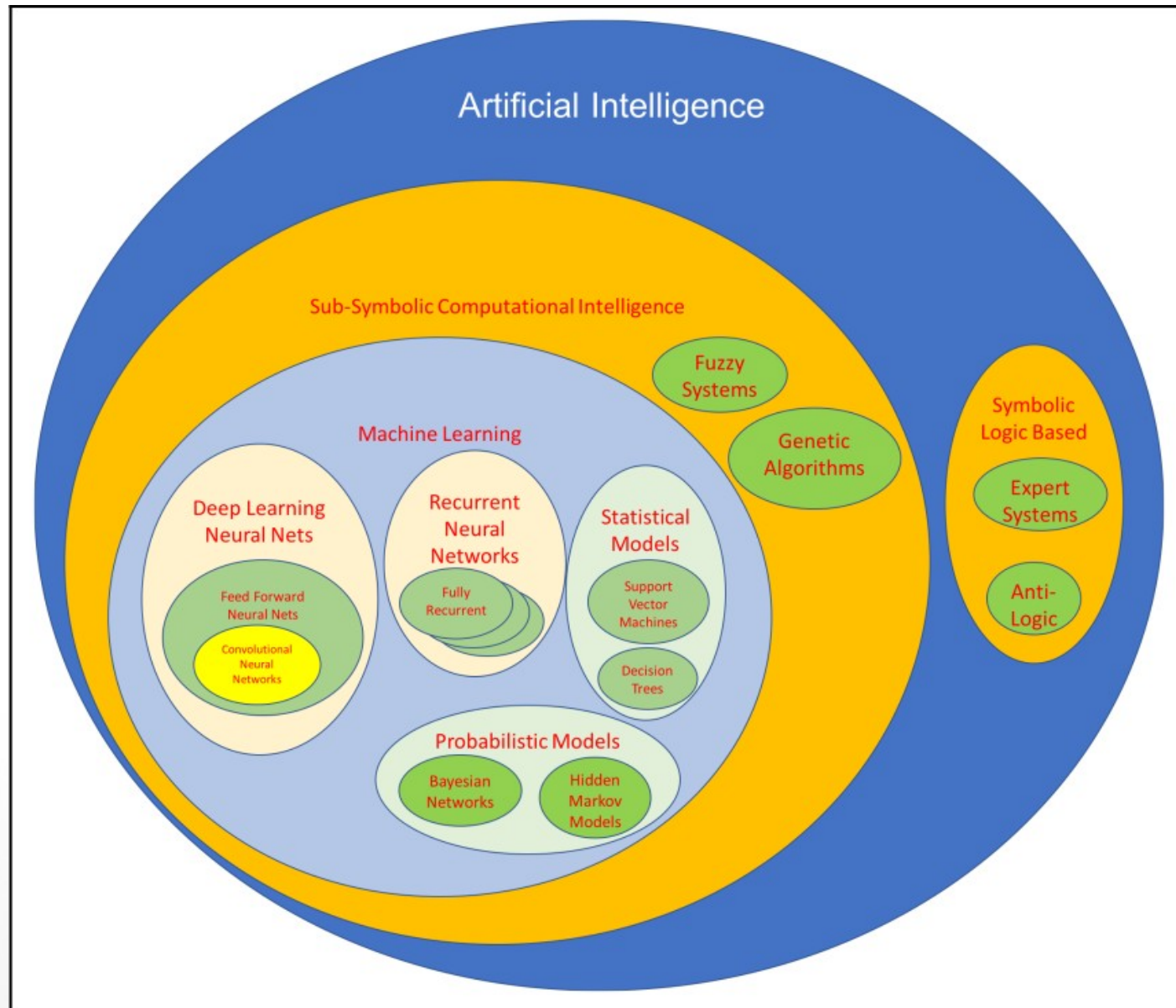
Lambda Architecture

- A Lambda architecture attempts to balance latency with throughput. Essentially, it mixes batch processing with stream processing.

Sector Use Cases

Industry	Use cases	Cloud services	Typical bandwidth	Real time	Analytics
Manufacturing	<ul style="list-style-type: none"> Operational technology Brownfield Asset tracking Factory automation 	<ul style="list-style-type: none"> Dashboards Bulk storage Data lakes SDN (hybrid cloud topology) Low latency 	<ul style="list-style-type: none"> 500 GB/day/factory part produced 2 TB/minute mining operations 	Less than 1s	<ul style="list-style-type: none"> Recurrent neural nets Bayesian networks
Logistics and transport	<ul style="list-style-type: none"> Geolocation tracking Asset tracking Equipment sensing 	<ul style="list-style-type: none"> Dashboards Logging Storage 	<ul style="list-style-type: none"> Vehicles: 4 TB/day/vehicle (50 sensors) Aircraft: 2.5 to 10 TB/day (6000 sensors) Assets tracking: 1 MB/day/beacon 	<ul style="list-style-type: none"> Less than 1s (real-time) Daily (batch) 	Rule engines
Healthcare	<ul style="list-style-type: none"> Asset tracking Patient tracking Home health monitoring Wireless health equipment 	<ul style="list-style-type: none"> Reliability and HIPPA Private cloud option Storage and archival Load balancing 	<ul style="list-style-type: none"> 1 MB/day/sensor 	<ul style="list-style-type: none"> Less than 1s: Life critical Non-life critical: On each change 	<ul style="list-style-type: none"> Recurrent Neural Networks (RNN) Decision trees Rules engines
Agriculture	<ul style="list-style-type: none"> Livestock health and location tracking Soil chemistry analysis 	<ul style="list-style-type: none"> Bulk storage - archiving Cloud-to-cloud provisioning 	<ul style="list-style-type: none"> 512 KB/day/livestock head 1000 to 10000 head of cattle per feedlot 	<ul style="list-style-type: none"> 1 second (real-time) 10 minutes (batch) 	Rules engines
Energy	<ul style="list-style-type: none"> Smart meters Remote energy monitoring (solar, natural gas, oil) Failure prediction 	<ul style="list-style-type: none"> Dashboards Data lakes Bulk storage for Historical rate prediction SDN Low latency 	<ul style="list-style-type: none"> 100-200 GB/day/wind turbine 1 to 2 TB/day/oil rig 100 MB/day/smart meter 	<ul style="list-style-type: none"> Less than 1s: energy production 1 minute: smart meters 	<ul style="list-style-type: none"> RNN Bayesian networks Rules engines
Consumer	<ul style="list-style-type: none"> Real-time health logging Presence detection Lighting and heating/AC Security Connected home 	<ul style="list-style-type: none"> Dashboards PaaS Load balancing Bulk storage 	<ul style="list-style-type: none"> Security camera: 500 GB/day/camera Smart device: 1-1000 KB/day/sensor-device Smart home: 100 MB/day/home 	<ul style="list-style-type: none"> Video: less than 1s Smart home: 1s 	<ul style="list-style-type: none"> Convolutional neural nets (image sensing) Rules engines
Retail	<ul style="list-style-type: none"> Cold chain sensing POS machines Security systems Beaconing 	<ul style="list-style-type: none"> SDN/SDP Micro-segmentation Dashboards 	<ul style="list-style-type: none"> Security: 500 GB/day/camera General: 1-1000 MB/day/device 	<ul style="list-style-type: none"> POS and credit transaction: 100ms Beaconing: 1s 	<ul style="list-style-type: none"> Rules engines Convolutional neural networks for security
Smart City	<ul style="list-style-type: none"> Smart parking Smart trash pickup Environmental sensors 	<ul style="list-style-type: none"> Dashboards Data lakes Cloud-to-cloud services 	<ul style="list-style-type: none"> Energy monitors: 2.5 GB/day/city (70K sensors) Parking spots: 300 MB/day (80,000 sensors) Waste monitors: 350 MB/day (200,000 sensors) Noise monitors: 650 MB/day (30,000 sensors) 	<ul style="list-style-type: none"> Electric meters: 1 minute Temperature: 15 minutes Noise: 1 minute Waste: 10 minutes Parking spots: every change 	<ul style="list-style-type: none"> Rules engine Decision trees

Machine Learning in IoT





Machine Learning Models

- There are two types of learning systems to consider which are as follows:
 - **Supervised learning:** It simply implies that the training data provided to the model has an associated label with each entry. For example, a set may be a collection of pictures each labeled with the content of that image: for example, cat, dog, banana, car. Many machine learning models today are supervised. Supervised learning allows for classification and regression problems to be solved. We will discuss classification and regression later on in this chapter.
 - **Unsupervised learning:** It has no label for the training data. Obviously, this type of learning cannot resolve an image of a dog to the label dog. This type of learning model uses mathematical rules to reduce redundancy. A typical use case is to find clusters of like things.



Machine Learning Models(2)

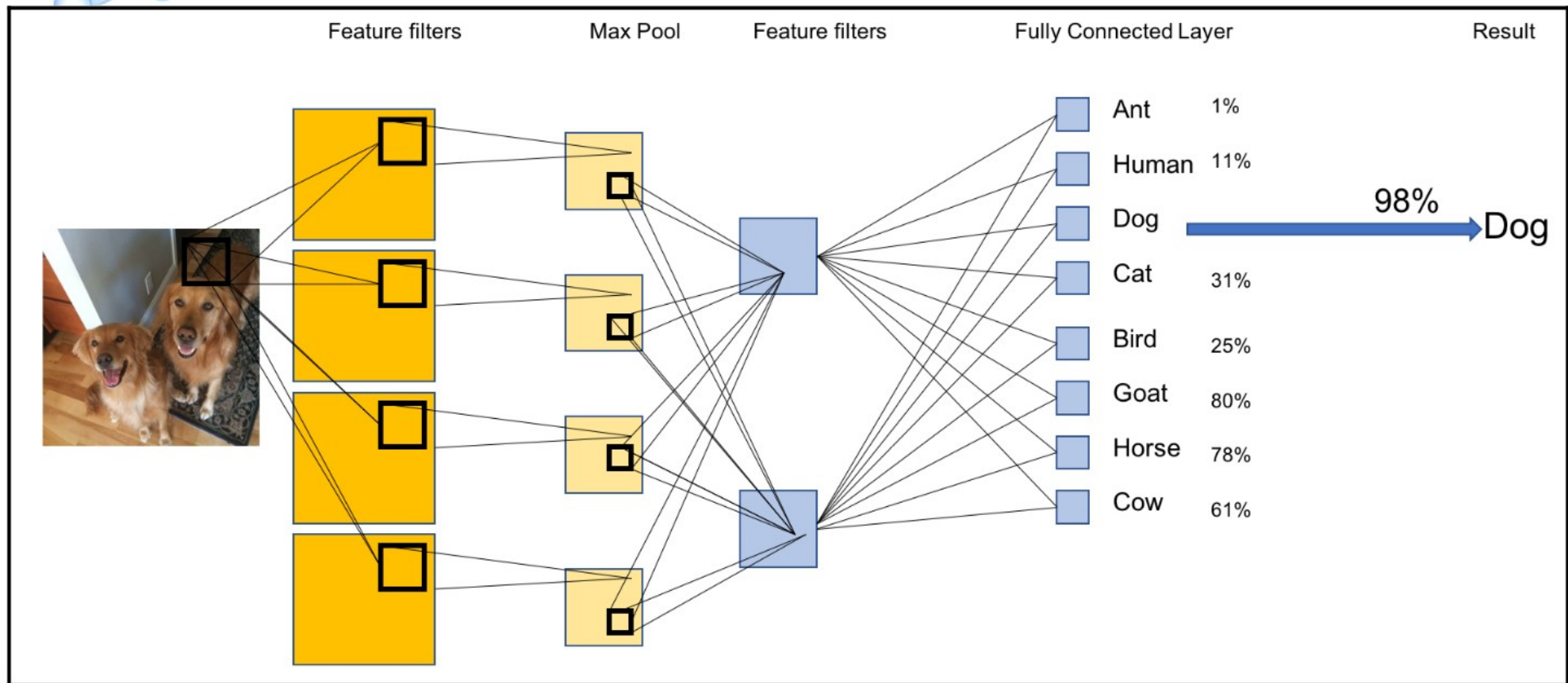
- The three fundamental uses of machine learning are:
 - Classification
 - Regression
 - Anomaly detection



Machine Learning Construct

- Random forests: Statistical models (fast model, good for systems with many attributes needed for anomaly detection)
- Bayesian networks: Probabilistic models
- Convolutional Neural Network: Deep learning (deep learning model for unstructured image data)
- RNN: Recurrent neural nets (deep learning model for time series analysis)

CNN Example



Comparison

Model	Best application	Worse fit and side effects	Resource demands	Training
Random forests (statistical models)	<ul style="list-style-type: none"> • Anomaly detection • Systems with 1000's of choice points and hundreds of inputs • Regression and classification • Handles mixed data types • Ignores missing values • Scales linearly with input 	<ul style="list-style-type: none"> • Feature extraction • Time and sequence analysis 	Low	<ul style="list-style-type: none"> • Training based on bagging techniques. for maximum effectiveness • Training fairly resource light • Mainly supervised
RNN (temporal and sequence-based neural networks)	<ul style="list-style-type: none"> • Prediction of an event based on a sequence • Streaming data patterns • Time-correlated series data • Maintains knowledge of past states to predict new states (electrical signals, audio, speech recognition) • Unstructured data • Input variables may or may not be dependent 	<ul style="list-style-type: none"> • Image and video analysis • Systems of requiring thousands of features 	<ul style="list-style-type: none"> • Very high for training • High for inference execution 	<ul style="list-style-type: none"> • Training more cumbersome than CNN backpropagation • Very hard to train • Supervised
CNN (deep learning)	<ul style="list-style-type: none"> • Prediction of an object based on surrounding values • Pattern and feature identification • 2D image recognition • Unstructured Data • Input variables may or may not be dependent 	<ul style="list-style-type: none"> • Time-based and sequential predictions • Systems of requiring thousands of features 	<ul style="list-style-type: none"> • Very high for Training (floating point precision, large training sets, large memory demands) • High for inference execution 	Supervised and unsupervised
Bayesian networks (probabilistic models)	<ul style="list-style-type: none"> • Noisy and incomplete data sets • Streaming data patterns • Time correlated series • Structured data • Signal analysis • Models developed quickly 	<ul style="list-style-type: none"> • Assumes all input variables are independent • Perform poorly with high orders of data dimensions 	Low	<ul style="list-style-type: none"> • Little training data need with respect to other artificial neural networks



Summary

- This chapter was a brief introduction to data analytics for IoT in the cloud and in the fog.
- Data analytics is where the value is extracted out of the sea of data produced by millions or billions of sensors.
- Analytics is the realm of the data scientist in attempts to find the hidden patterns and develop predictions from an overwhelming amount of data