Data and Analytics for IoT

MODULE 4

- As more and more devices are added to IoT networks, the data generated by these systems becomes overwhelming
- Traditional data management systems are simply unprepared for the demands of what has come to be known as "big data."
- The ***real value of IoT*** is not just in connecting things but rather in the *data produced by those things*, the *new services you can enable* via those connected things, and the business insights that the data can reveal.
- However, to be useful, the data needs to be handled in a way that is *organized and controlled*.
- Thus, a new approach to data analytics is needed for the Internet of Things

An Introduction to Data Analytics for IoT

- In the world of IoT, the creation of massive amounts of data from sensors is common and one of the biggest challenges—not only from a ***transport perspective*** **but also from a *data management* standpoint**
- ***Modern jet engines*** are fitted with thousands of sensors that generate a whopping ***10GB of data per second***
- *Analyzing this amount of data in the most efficient manner possible falls under the umbrella of data analytics*

- Not all data is the same; it can be categorized and thus analyzed in different ways.

- Depending on how data is categorized, various data analytics tools and processing methods can be applied.

- **Two important categorizations from an IoT perspective are** *whether the data is structured or unstructured* and *whether it is in motion or at rest.*

Structured Versus Unstructured Data

- Structured data and unstructured data are important classifications as they typically require *different toolsets* from a data analytics perspective
- **Structured data means** that the *data follows a model or schema that defines how the data is represented or organized, meaning it fits well with a traditional relational database management system (RDBMS).*
- In many cases you will find structured data in a simple ***tabular form***—for example, a ***spreadsheet*** *where data occupies a specific cell and can be explicitly defined and referenced*

- Structured data can be found in most **computing systems** and includes everything from banking transaction and invoices to computer log files and router configurations.
- IoT sensor data often uses **structured values,** such as ***temperature, pressure, humidity, and so on, which are all sent in a known format.***
- Structured data is *easily formatted, stored, queried, and processed*

- Because of the highly organizational format of structured data, a wide array of **data analytics tools** are readily available for processing this type of data.

- From custom scripts to commercial software like **Microsoft Excel** and **Tableau**

- **Unstructured data lacks a logical schema** for understanding and decoding the data through traditional programming means.

- **Examples** of this data type include **text, speech, images, and video.**

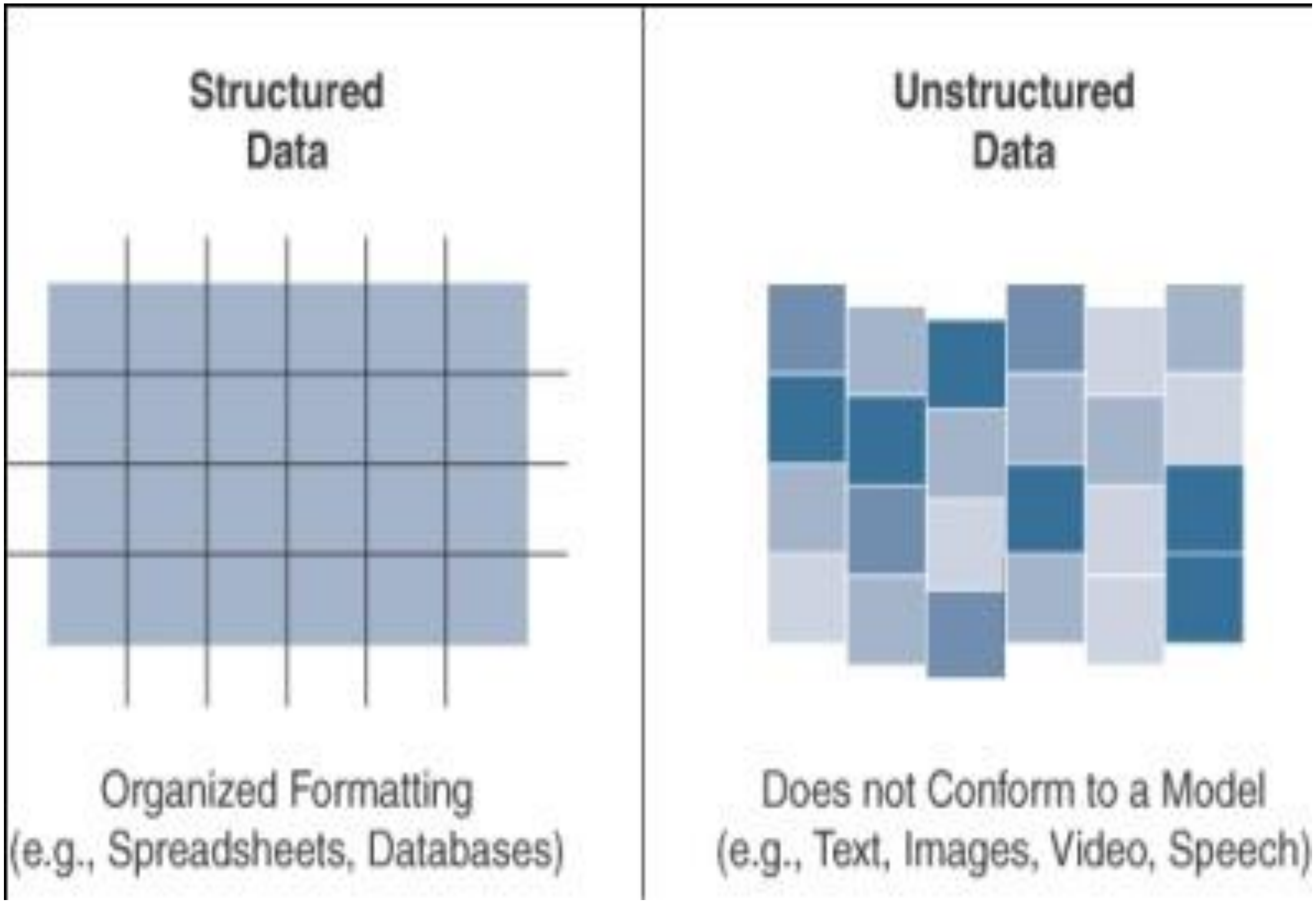
- As a general rule, **any data that does not fit neatly into a predefined data model is classified as unstructured data**

- According to some estimates, ***around 80% of a business's data is unstructured.***
- Because of this fact, **data analytics method**s that can be **applied to unstructured data**, such as cognitive computing and machine learning, are deservedly garnering a lot of attention.
- With machine learning applications, such as **natural language processing (NLP)**, **you can decode speech.**
- With image/facial recognition applications, you can extract critical information from still images and video

- **Smart objects in IoT networks *generate both structured and unstructured data*.**

- Structured data is more easily managed and processed due to its well-defined organization.

- On the other hand, unstructured data can be harder to deal with and typically requires very different analytics tools for processing the data


Structured
Data
Unstructured
Data
Organized Formatting
(e.g., Spreadsheets, Databases)
Does not Conform to a Model
(e.g., Text, Images, Video, Speech)

Data in Motion Versus Data at Rest

- Data in IoT networks is either in transit ("data in motion") or being held or stored ("data at rest").

- Examples of data in motion include ***traditional client/server exchanges, such as web browsing and file transfers, and email.***

- **Data saved to a hard drive, storage array, or USB driv**e is ***data at rest.***

- **From an IoT perspective**, the data from smart objects is considered data in motion as it passes through the network en route to its final destination.
- This is often ***processed at the edge, using fog computing.***
- When data is processed at the edge, it may be filtered and deleted or forwarded on for further processing and possible storage at a fog node or in the data center.
- Data does not come to rest at the edge.
- When data arrives at the data center, it is possible to process it in real-time, just like at the edge, while it is still in motion.
- **Tools** with this sort of capability, are **Spark, Storm, and Flink**

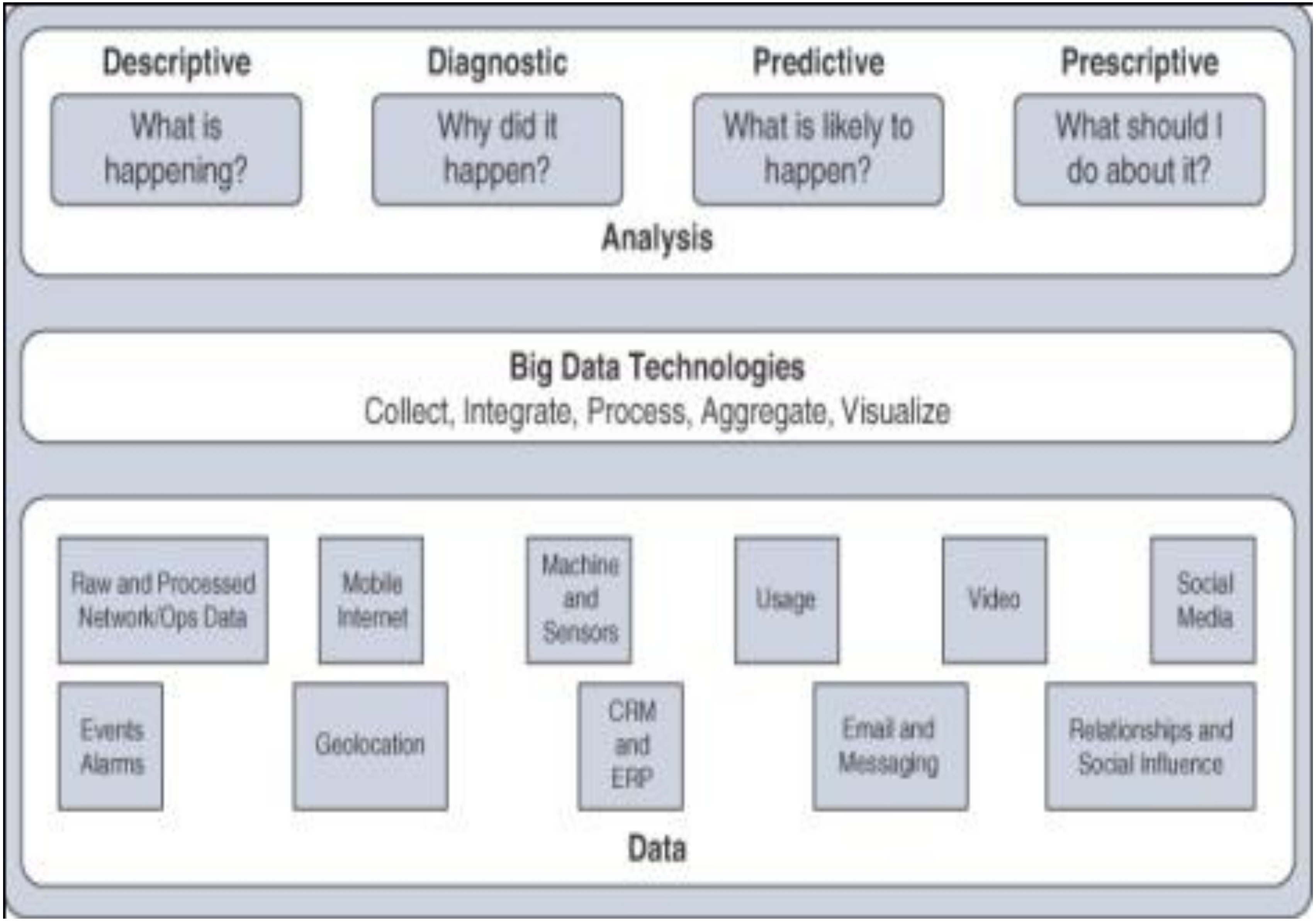
- **Data at rest in IoT networks** can be typically found in ***IoT brokers*** or in ***some sort of storage array at the data center***

- Hadoop not only helps with data processing but also data storage

IoT Data Analytics Overview

- The **true importance of IoT** data from smart objects is realized only when the ***analysis of the data*** leads to ***actionable business intelligence and insights***.

- ***Data analysis is typically broken down by the types of results that are produced***



Types of Data Analysis Results

Four types of data analysis results

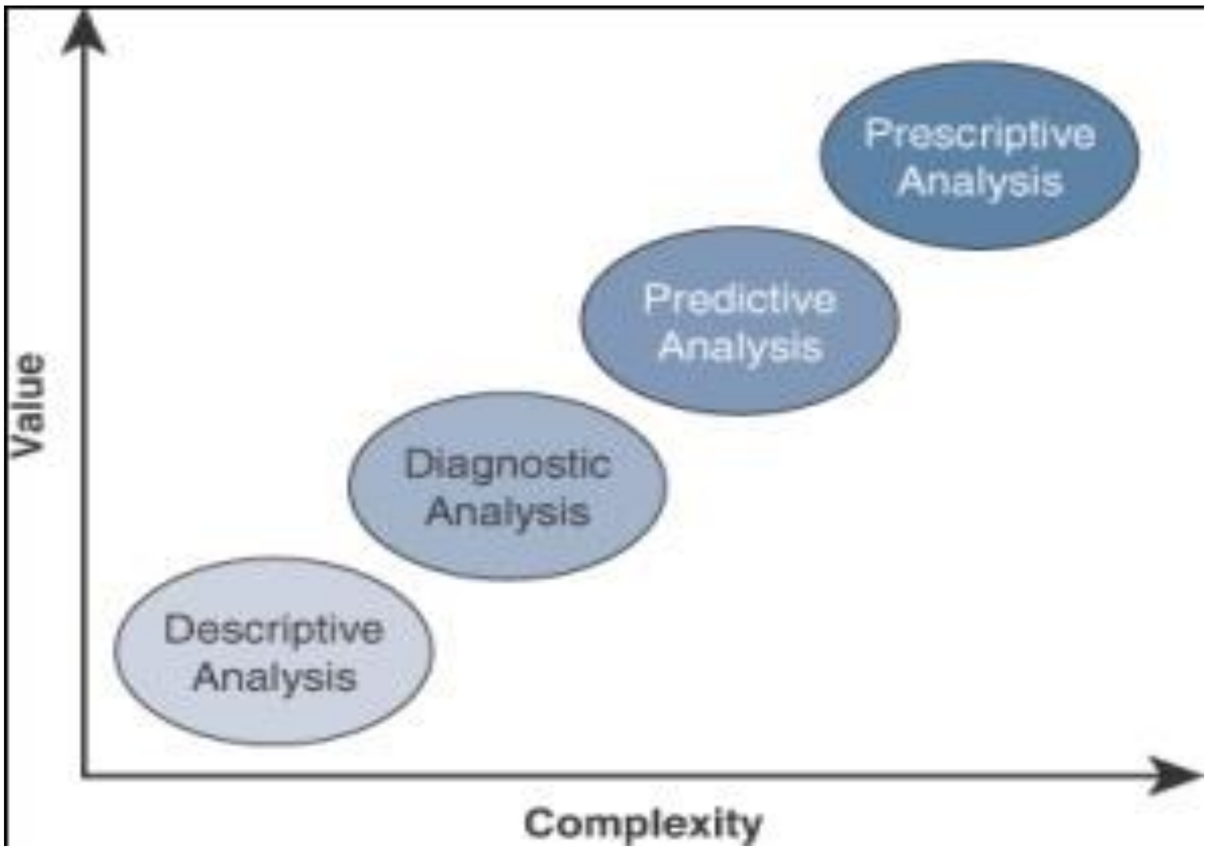
- **Descriptive:**
- Descriptive data analysis tells you what is happening, either now or in the past.
- For example, a thermometer in a truck engine reports temperature values every second.
- From a descriptive analysis perspective, you can pull this data at any moment to gain insight into the **current operating condition of the truck engine.**
- **If the temperature value is too high**, then **there may be a cooling problem** or the ***engine may be experiencing too much load.***

- **Diagnostic:**
- When you are interested in the **"why,"** diagnostic data analysis can provide the answer.
- Continuing with the example of the temperature sensor in the truck engine, ***you might wonder why the truck engine failed.***
- **Diagnostic analysis might show that the temperature of the engine was too high, and the engine overheated**.
- Applying diagnostic analysis across the data generated by a wide range of smart objects can provide a clear picture of why a problem or an event occurred

- **Predictive:**
- **Predictive analysis aims to foretell problems or issues before they occur.**
- For example, with historical values of temperatures for the **truck engine, predictive analysis could provide an estimate on the remaining life of certain components in the engine.**
- These components could then be proactively replaced before failure occurs.
- Or perhaps if temperature values of the truck engine start to rise slowly over time, this could indicate the need for an oil change or some other sort of engine cooling maintenance.

- **Prescriptive:**
- Prescriptive analysis goes a step beyond predictive and recommends solutions for upcoming problems.
- A prescriptive analysis of the temperature data from a truck engine might calculate **various alternatives to cost-effectively maintain our truck**
- *These calculations could range from the cost necessary for more frequent oil changes and cooling maintenance to installing new cooling equipment on the engine or upgrading to a lease on a model with a more powerful engine.*
- Prescriptive analysis looks at a variety of factors and makes the appropriate recommendation

- Both predictive and prescriptive analyses are more resource intensive and increase complexity, but the value they provide is much greater than the value from descriptive and diagnostic analysis



IoT Data Analytics Challenges

Problems by using RDMS in IoT

1. **Scaling Problems** (performance issues, costly to resolve, req more h/w, architechture changes)

2. **Volatility of Data** (change in schema)

Machine Learning

- ML is central to IoT.
- Data collected by smart objects needs to be **analyzed**, and **intelligent actions** need to be taken based on these analyses.
- Performing this kind of operation manually is almost impossible (or very, very slow and inefficient).

- ***Machines are needed to process information fast and react instantly when thresholds are met***
  - ***Ex: advances in self-driving vehicle--abnormal pattrn recognition in a crowd and automated intelligent and machine-assisted decision system***

Machine Learning Overview

- Machine learning is, in fact, part of a larger set of technologies commonly grouped under the term ***artificial intelligence (AI)*.**
- AI includes any technology that allows a computing system to **mimic human intelligence** using any technique, from very advanced logic to basic "if-then-else" decision loops.
- Any computer that **uses rules to make decisions** belong to this group

- **A simple example is an app that can help you find your parked car.**
- A GPS reading of your position at regular intervals calculates your speed.
- A basic threshold system determines whether you are driving (for example, "if speed > 20 mph or 30 kmh, then start calculating speed").
- When you park and disconnect from the car Bluetooth system, the app simply records the location when the disconnection happens.
- This is where your car is parked.

- In more **complex cases**, static rules cannot be simply inserted into the program because they require *parameters that can change or that are imperfectly understood*
- A typical example is a **dictation program** that runs on a computer.
- The ***program is configured to recognize the audio pattern of each word in a dictionary,*** but it does not know your voice's specifics—your accent, tone, speed, and so on

- You need to record a **set of predetermined sentences** to help the tool match **well-known words** to **the sounds you make when you say the words.**
- This process is called machine learning.
- **ML is concerned with any process where the computer needs to receive a set of data that is processed to help perform a task with more efficiency.**
- ML is a vast field but can be simply divided in **two main categories: supervised and unsupervised learning**

Supervised Learning

- In supervised learning, **the machine is trained with input for which there is a known correct answer**.
- For example, suppose that you are training a system to recognize when there is a human in a **mine tunnel**.
- **A sensor** equipped with a **basic camera** can **capture shapes** and **return them** to a **computing system** that is responsible for **determining whether the shape is a human or something else** (such as a vehicle, a pile of ore, a rock, a piece of wood, and so on.).

- With supervised learning techniques, hundreds or thousands of images are fed into the machine, and each image is labelled (human or nonhuman in this case).
- This is called the ***training set.***
- An **algorithm** is used to ***determine common parameters and common differences between the images.***
- The **comparison** is usually done at the ***scale of the entire image, or pixel by pixel.***
- Images are **resized** to have the same characteristics (resolution, color depth, position of the central figure, and so on), and each point is analyzed.

- Each new image is compared to the set of known "good images," and a **deviation is calculated** to ***determine how different, the new image*** is from the ***average human image*** and, therefore, the **probability** that what is shown is a human figure. This process is called ***classification.***

- After training, the machine should be able to recognize human shapes. **Before real field deployments,** the *machine is usually tested with* ***unlabeled pictures****— this is called* ***the validation or the test set***, depending on the ML system used—to verify that the recognition level is at acceptable thresholds. ***If the machine does not reach the level of success expected, more training is needed***

- In other cases, the learning process is not about classifying in two or more categories *but about finding a correct value.*
- For example, the **speed of the flow of oil in a pipe** is a function of the *size of the pipe, the viscosity of the oil, pressure, and a few other factors.*
- When you train the machine with measured values, the machine can predict the speed of the flow for a new, and unmeasured, viscosity.
- This process is called ***regression; regression predicts numeric values,*** whereas classification predicts categories

Unsupervised Learning

- In some cases, supervised learning is not the best method for a machine to help with a human decision.
- Suppose that you are processing IoT data from a factory **manufacturing small engines.**
- You know that about 0.1% of the produced engines on average need adjustments to prevent later defects, and your task is to identify them before they get mounted into machines and shipped away from the factory.
- With hundreds of parts, it may be very difficult to detect the potential defects, and it is almost impossible to train a machine to recognize issues that may not be visible

- However, you can test each engine and record multiple parameters, such as sound, pressure, temperature of key parts, and so on.
- Once data is recorded, you can graph these elements in relation to one another (for example, ***temperature as a function of pressure***, ***sound versus rotating speed overtime).***
- You can then input this data into a computer and use mathematical functions to find groups.

- **For example**, **you may decide to group the engines by the sound they make at a given temperature.**
- A standard function to operate this grouping, ***K-means clustering, finds the mean values for a group of engines (for example, mean value for temperature, mean frequency for sound).***
- Grouping the engines this way can quickly reveal several types of engines that all belong to the same category (for example, *small engine of chainsaw typ*e, *medium engine of lawnmower type*).
- All engines of the same type produce sounds and temperatures in the same range as the other members of the same group.

- **There will occasionally be an engine in the group that displays unusual characteristics (slightly out of expected temperature or sound range).**
- **This is the engine that you send for manual evaluation**.
- The computing process associated with this determination is called ***unsupervised learning.***
- *This type of learning is* unsupervised ***because there is not a "good" or "bad" answer known in advance.***
- It is the ***variation from a group behavior*** that allows the **computer to learn that something is different**

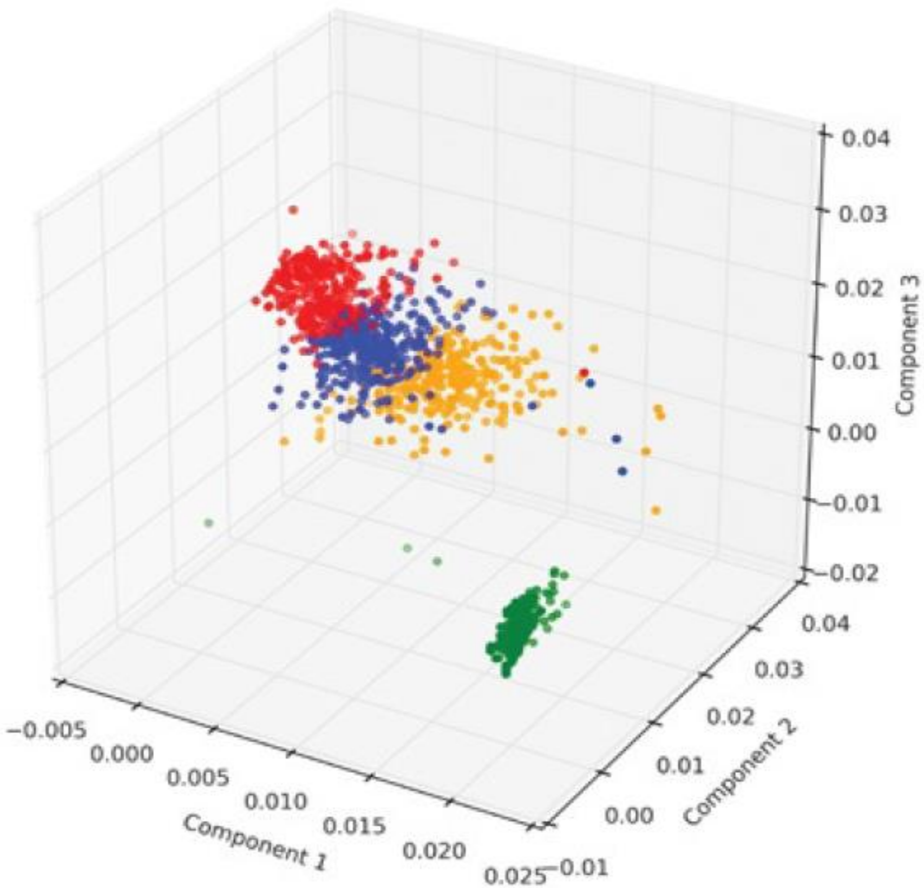


Figure 7-5 *Clustering and Deviation Detection Example*

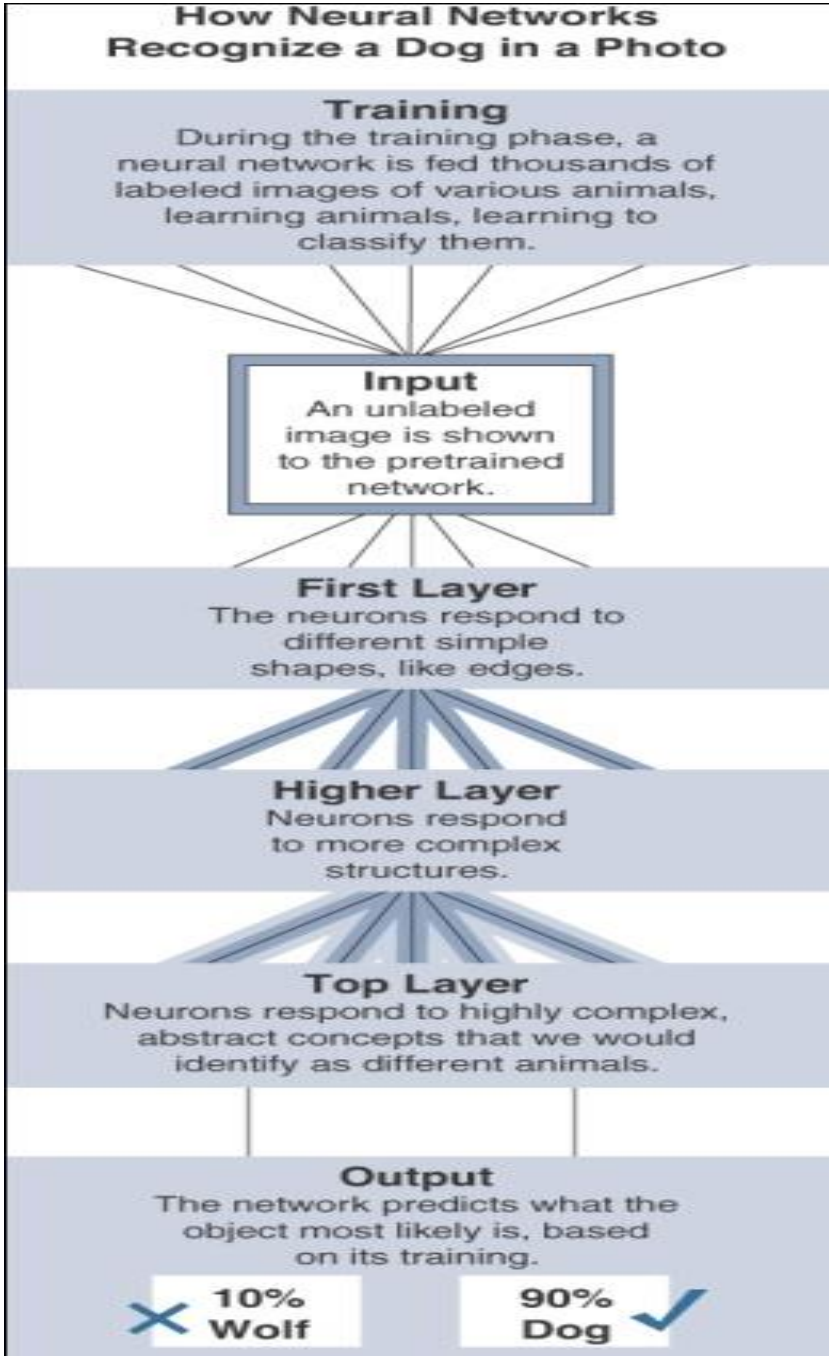
Neural Networks

- Processing multiple dimensions requires a *lot of computing power*.
- It is also difficult to determine *what parameters to input* and *what combined variations should raise red flags.*
- Similarly, supervised learning is efficient only with a large training set; larger training sets usually lead to higher accuracy in the prediction
- Training the machines was often deemed too expensive and complicated.

- **Distinguishing between a human and a car is easy.**
- The computer can recognize that humans have distinct shapes (such as legs or arms) and that vehicles do not.
- But, distinguishing a human from another *mammal* is much more difficult
- The same goes for telling the difference between a pickup truck and a van.
- You can tell when you see one, but training a machine to differentiate them ***requires more than basic shape recognition***

- Neural networks are ML methods that mimic the way the human brain works.
- When you look at a human figure, multiple zones of your brain are activated to recognize colors, movements, facial expressions, and so on.
- Your brain combines these elements to conclude that the shape you are seeing is human
- Neural networks mimic the same logic.
- The information goes through different algorithms (called *units*), each of which is in charge of processing an aspect of the information

- **The resulting value of one unit computation can be used directly or fed into another unit for further processing to occur**
- For example, a ***neural network processing human image recognition*** may have **two units in a first layer** that determines whether the image has straight lines and sharp angles—*because vehicles commonly have straight lines and sharp angles*, *and human figures do not.*
- If the image passes the first layer successfully (because there are no or only a small percentage of sharp angles and straight lines), **a second layer may look for different features (presence of face, arms, and so on)**, and then a **third layer might compare the image to images of various animals and conclude that the shape is a human (or not)**


How Neural Networks
Recognize a Dog in a Photo
Training
During the training phase, a neural network is fed thousands of labeled images of various animals, learning animals, learning to classify them.
Input
An unlabeled image is shown to the pretrained network.
First Layer
The neurons respond to different simple shapes, like edges.
Higher Layer
Neurons respond to more complex structures.
Top Layer
Neurons respond to highly complex, abstract concepts that we would identify as different animals.
Output
The network predicts what the object most likely is, based on its training.
10%
Wolf
90%
Dog

- Neural networks rely on the **idea** that **information is divided into key components, and each component is assigned a weight**.
- The weights compared together decide the classification of this information (no straight lines + face + smile = human).
- When the result of a layer is fed into another layer, the process is called **deep learning** ("deep" because the learning process has more than a single layer).
- One advantage of deep learning is that ***having more layers allows for richer intermediate processing*** and ***representation of the data.***
- At each layer, ***the data can be formatted to be better utilized by the next layer.*** **This process increases the efficiency of the overall result**

Machine Learning and Getting Intelligence from Big Data

- ML operations can be organized into two broad subgroups:
- **Local learning**
  - Data is collected and processed **locally,** either in the **sensor itself (the edge node)** or in the **gateway (the fog node)**
- **Remote learning**
  - Data is collected and **sent to a central computing unit (typically the data center in a specific location or in the cloud), where it is processed.**

- Regardless of the location where data is processed, ***common applications of ML for IoT revolve around*** ***four major domains:***

- **Monitoring**
- Smart objects monitor the environment where they operate
- Example such as air temperature, humidity, or presence of carbon dioxide in a mine etc

- **Behavior control**
- *Monitoring commonly works in conjunction with behavior control.*

- When a given set of parameters reach a target **threshold** —**defined in advance (that is, supervised)** or **learned dynamically through deviation from mean values (that is, unsupervised)—*monitoring functions generate an alarm.***

- This alarm can be relayed to a human, ***but a more efficient and more advanced system would trigger a corrective action***, such as *increasing the flow of fresh air in the mine tunnel*, *turning the robot arm*, or *reducing the oil pressure in the pipe.*

- **Operations optimization**
- Behavior control typically aims at taking corrective actions based on thresholds.
- However, ***analyzing data can also lead to changes that improve the overall process***
- For example, **a water purification plant in a smart city can implement a system to monitor the efficiency of the purification process** ***based on which chemical (from company A or company B) is used, at what temperature, and associated to what stirring mechanism (stirring speed and depth).***

- **Self-healing, self-optimizing**
- The ML engine can be ***programmed to dynamically monitor and combine new parameters (randomly or semi-randomly) and automatically deduce and implement new optimizations when the results demonstrate a possible gain.***
- The system becomes ***self-learning and self optimizing***
- It also detects new K-means deviations that result in predetection of new potential defects, allowing the system to self-heal

Predictive Analytics

- Machine learning and big data processing for IoT fit very well into th digitization
- The ***advanced stages*** of this model **see the network self-diagnose and self-optimize**.
- In the IoT world, this behavior is what the previous section describes
- ***When data from multiple systems is combined and analyzed together, predictions can be made about the state of the system.*** For example,
- case of sensors deployed on locomotives. Multiple smart objects measure the pull between carriages, the weight on each wheel, and multiple other parameters to offer a form of cruise control optimization for the driver.

Predictive Analytics

- At the same time, cameras observe the state of the tracks ahead, audio sensors analyze the sound of each wheel on the tracks, and multiple engine parameters are measured and analyzed.
- All this data can be returned to a data processing center in the cloud that can re-create a virtual twin of each locomotive.
- Modeling the state of each locomotive and combining this knowledge with anticipated travel and with the states (and detected failures) of all other locomotives of the same type circulating on the tracks of the entire city, province, state, or country allows the analytics platform to make very accurate **predictions on what issue is likely to affect each train and each locomotive**.

- **Such predictive analysis allows preemptive maintenance and increases the safety and efficiency of operations.**
- Similarly, sensors combined with big data can anticipate defects or issues in vehicles operating in mines, in manufacturing machines, or any system that can be monitored, along with other similar systems.

Big Data Analytics Tools and Technology

- **Big data analytics can** ***consist of many different software pieces*** **that** ***together collect, store, manipulate, and analyze all different data types.***
- Generally, the industry looks to the **"three *Vs*"** *to categorize* big data:
- **Velocity**
- Refers to ***how quickly data is being collected and analyzed.***
- *Hadoop Distributed File System* is designed to ingest and process data very quickly.
- Smart objects can *generate machine and sensor data at a very fast rate and require database or file systems capable of equally fast ingest functions.*

- **Variety**
- refers to ***different types of data.***
- Often you see *data categorized as structured, semi-structured, or unstructured*.
- *Different database technologies may only be capable of accepting one of these types.*
- ***Hadoop is able to collect and store all three types***

- **Volume**
- refers to the ***scale of the data.***

- Typically, this is ***measured from gigabytes on the very low end to petabytes or even exabytes of data on the other extreme***

- The **characteristics of big data** can be defined by the **sources** and **types of data**.
- First is **machine data,** which is ***generated by IoT devices*** *and* ***is typically unstructured data.***
- Second is ***transactional data***, which is *from sources that produce data from transactions on these systems*, and, *have* ***high volume and structured.***
- Third is ***social data sources***, which are typically ***high volume and structured.***
- Fourth is ***enterprise data***, which is data that is ***lower in volume*** and ***very structured***
- **Hence big data consists of data from all these separate sources**.

Massively Parallel Processing Databases

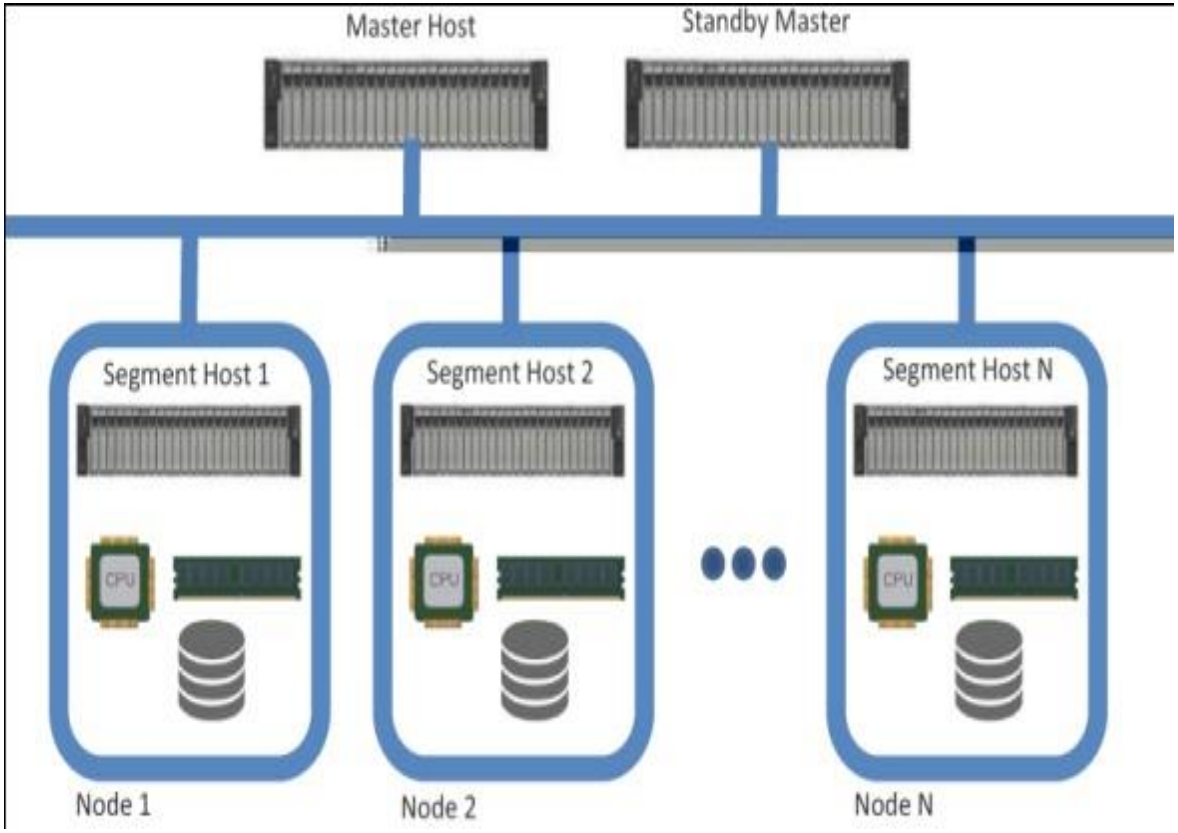
- Enterprises have used relational databases for storing structured, row and column style data types for decades.
- Relational databases are often grouped into a broad data storage category called data warehouses.
- Though they are the centerpiece of most data architectures, they are often used for longer-term archiving and data queries that can often take minutes or hours
- **Massively parallel processing (MPP) databases were built on the concept of the relational data warehouses but are designed to be much faster, to be efficient, and to support reduced query times**

- To accomplish this, MPP **databases take advantage of multiple nodes (computers) designed in a scale-out architecture such that both data and processing are distributed across multiple systems**

- MPPs are sometimes referred to as ***analytic databases*** *because they are designed* to allow for ***fast query processing*** and often ***have built-in analytic functions***

- As the name implies, these database types ***process massive data sets in parallel across many processors and nodes***

- An MPP architecture typically contains **a single master node** that is responsible for the ***coordination of all the data storage and processing across the cluster.***
- It operates in **a "shared-nothing" fashion**, ***with each node containing local processing, memory, and storage and operating independently***.



- **Data storage is optimized** across the nodes in a structured SQL-like format that allows data analysts to work with the data using common SQL tools and applications

NoSQL Databases

- NoSQL ("not only SQL") is a class of databases that support semi-structured and unstructured data, in addition to the structured data handled by data warehouses and MPPs
- NoSQL is ***not a specific database technology***; rather, it is an umbrella term that ***encompasses several different types of databases, including the following***
- **Document stores:**
- This type of database ***stores semi-structured data***, such as ***XML or JSON.***
- Document stores generally have ***query engines and indexing features*** that allow for many optimized queries

- **Key-value stores**

  This type of database ***stores associative arrays*** where ***a key is paired with an associated value***. These databases are ***easy to build and easy to scale***

- **Wide-column stores**

  This type of database ***stores similar to a key-value store***, but the ***formatting of the values can vary from row to row, even in the same table***

- **Graph stores**
  - This type of database is organized based on the ***relationships between elements.***
  - Graph stores are commonly used for ***social media or natural language processing***, ***where the connections between data are very relevant.***

Hadoop

- Hadoop is the most recent entrant into the data management market, but it is arguably the *most popular choice* as a ***data repository*** ***and*** ***processing engine.***

- Hadoop was originally ***developed as a result of projects at Google and Yahoo!***

- The original intent for Hadoop was to ***index millions of websites*** and ***quickly return search results for open source search engines***

- Initially, the project had two key elements:
- **Hadoop Distributed File System (HDFS):**

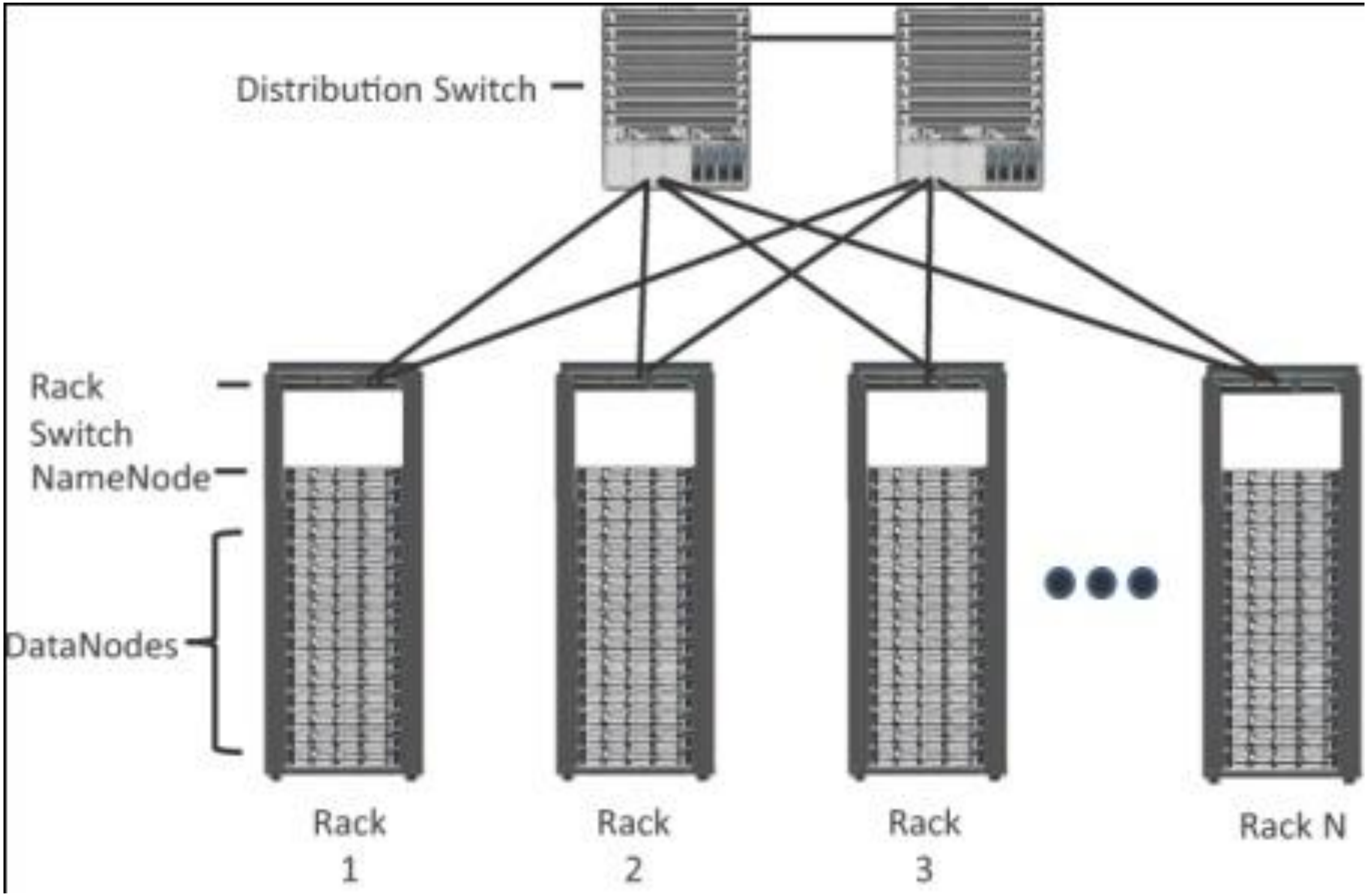
  A system for storing data across multiple nodes

- **MapReduce:**
  - ***A distributed processing engine*** that ***splits a large task into smaller ones that can be run in parallel.***

  - Hadoop relies on a ***scale-out architecture*** that ***leverages local processing, memory, and storage to distribute tasks and provide a scalable storage system for data.***

- Both MapReduce and HDFS take advantage of this ***distributed architecture to store and process massive amounts of data and are thus able to leverage resources from all nodes in the cluster.***

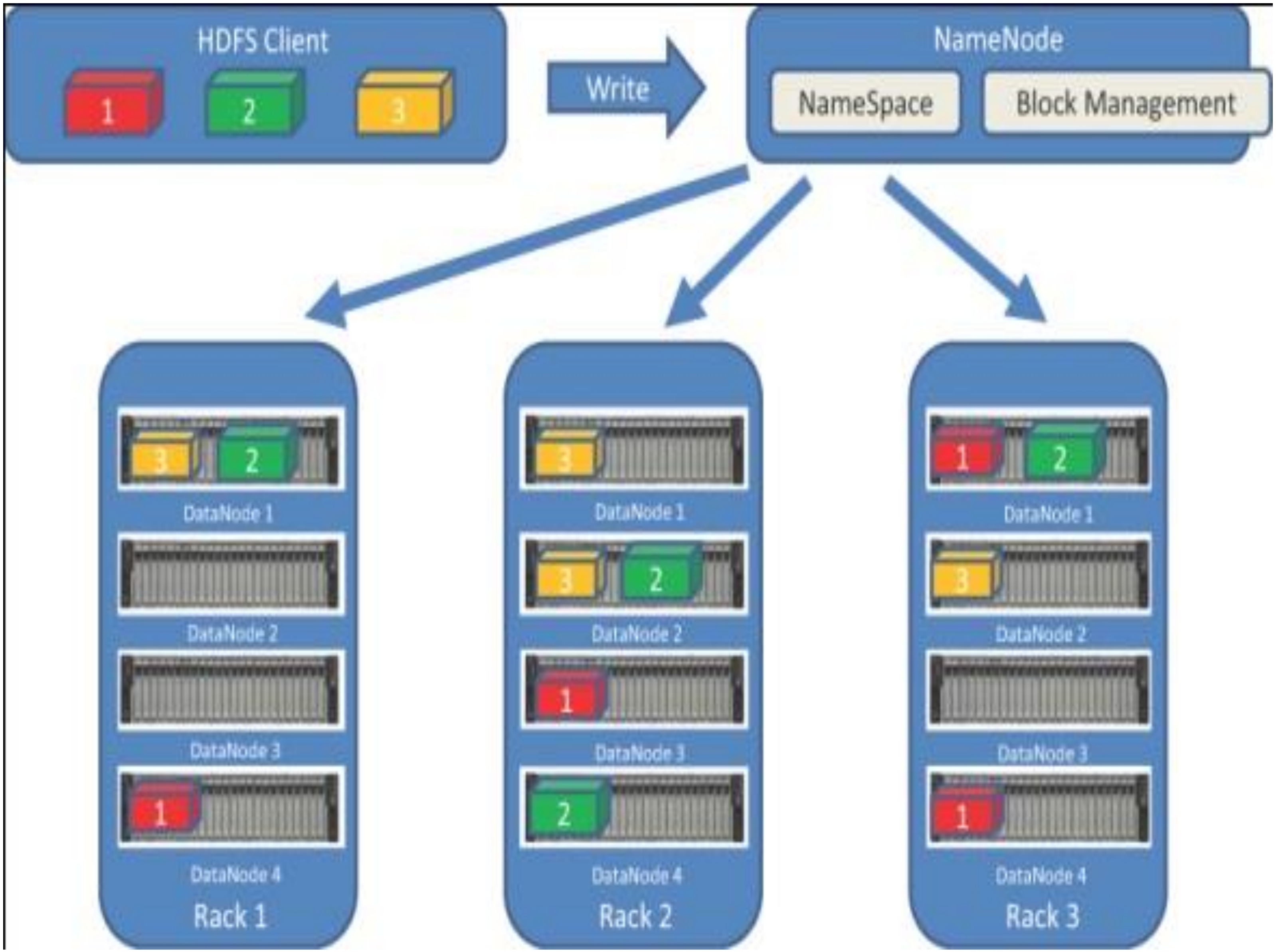
- For HDFS, this capability is handled by specialized nodes in the cluster, including **NameNodes** and **DataNodes**



Fig; Distributed Hadoop Cluster

- **NameNodes**
- These are a critical piece in data adds, moves, deletes, and reads on HDFS.
- They coordinate where the data is stored, and maintain a map of where each block of data is stored and where it is replicated.
- All interaction with HDFS is coordinated through the primary (active) NameNode, with a secondary (standby) NameNode notified of the changes in the event of a failure of the primary.
- The NameNode takes write requests from clients and distributes those files across the available nodes in configurable block sizes, usually 64 MB or 128 MB blocks.
- The NameNode is also responsible for instructing the DataNodes where replication should occur.

- **DataNodes**
- These are the servers where the data is stored at the direction of the NameNode.
- It is common to have many DataNodes in a Hadoop cluster to store the data.
- Data blocks are distributed across several nodes and often are replicated three, four, or more times across nodes for redundancy.
- Once data is written to one of the DataNodes, the DataNode selects two (or more) additional nodes, based on replication policies, to ensure data redundancy across the cluster


HDFS Client
1
2
3
Write
NameNode
NameSpace
Block Management
3
2
DataNode 1
DataNode 2
DataNode 3
1
DataNode 4
Rack 1
3
DataNode 1
3
2
DataNode 2
1
DataNode 3
2
DataNode 4
Rack 2
1
2
DataNode 1
3
DataNode 2
DataNode 3
1
DataNode 4
Rack 3

YARN

- Introduced with version 2.0 of Hadoop, YARN (Yet Another Resource Negotiator) was designed to enhance the functionality of MapReduce.
- With the initial release, MapReduce was responsible for batch data processing and job tracking and resource management across the cluster.
- YARN was developed to take over the resource negotiation and job/task tracking, allowing MapReduce to be responsible only for data processing.

The Hadoop Ecosystem

- Since the initial release of Hadoop in 2011, many projects have been developed to add incremental functionality to Hadoop and have collectively become known as the *Hadoop ecosystem*.
- **Apache Kafka**
- **Apache Spark**
- **Apache Storm and Apache Flink**
- **Lambda Architecture**

Comparing Big Data and Edge Analytics

- When you hear the term *big data, it is usually in reference to unstructured data* that has been collected and stored in the cloud
- Tools like Hadoop and MapReduce are great at tackling problems that require deep analytics on a large and complex quantity of unstructured data;
- However, due to their distance from the IoT endpoints and the bandwidth required to bring all the data back to the cloud, they are generally not well suited to real-time analysis of data as it is generated.

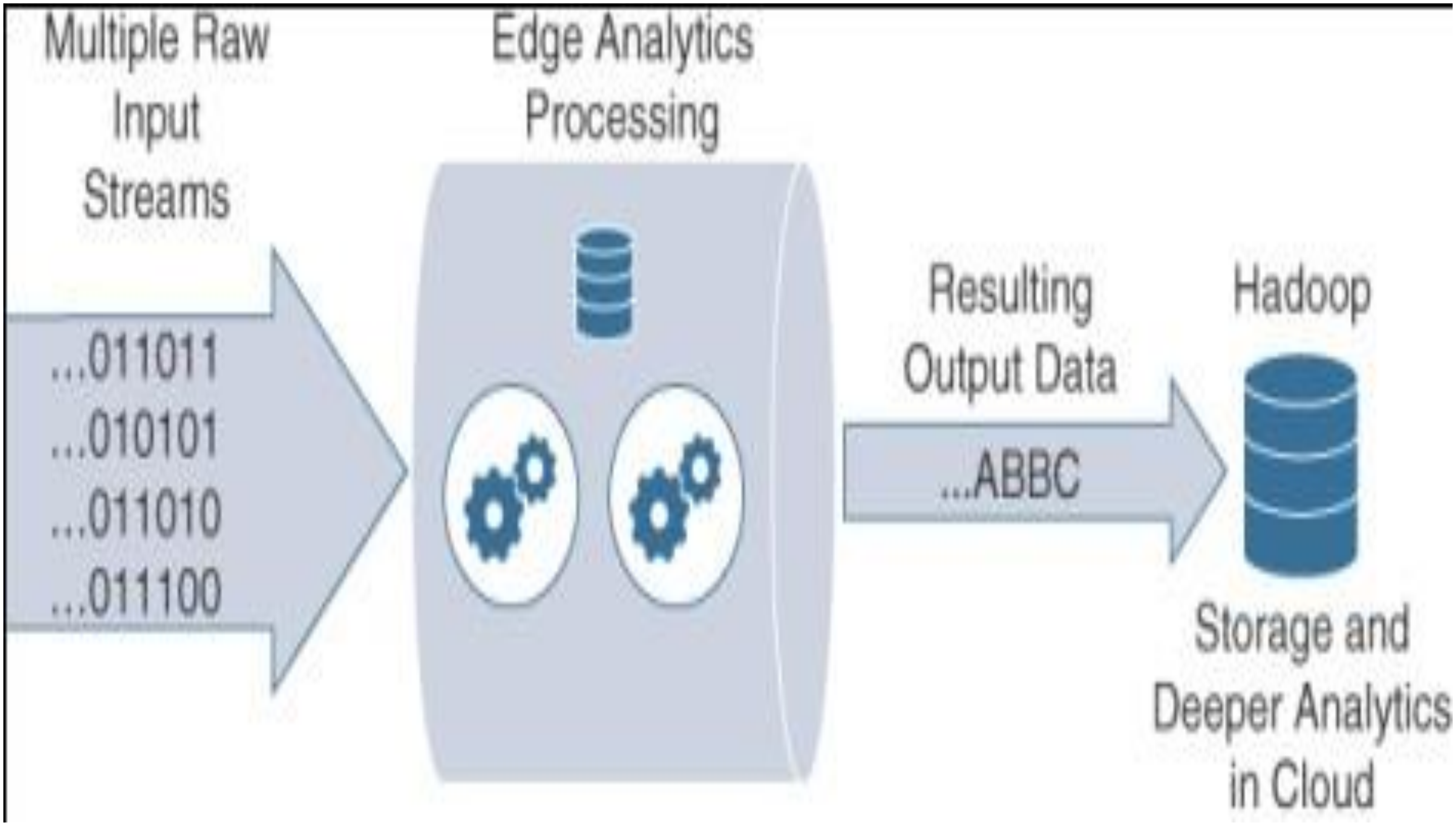
- In applying data analytics to the car racing example, big data analytics is used to examine all the statistics of the racing team and players based on their performance in the data center or cloud
- Streaming analytics involves analyzing a race while it is happening and trying to figure out who is going to win based on the actual performance in real-time—and this analysis is typically performed as close to the edge as possible.
- Streaming analytics allows you to continually monitor and assess data in real-time so that you can adjust or fine-tune your predictions as the race progresses.

- In the context of IoT, with streaming analytics performed at the edge (either at the sensors themselves or very close to them, in a fog node that is, for example, integrated into the gateway), it is possible to process and act on the data in realtime without waiting for the results from a future batch-processing job in the cloud.

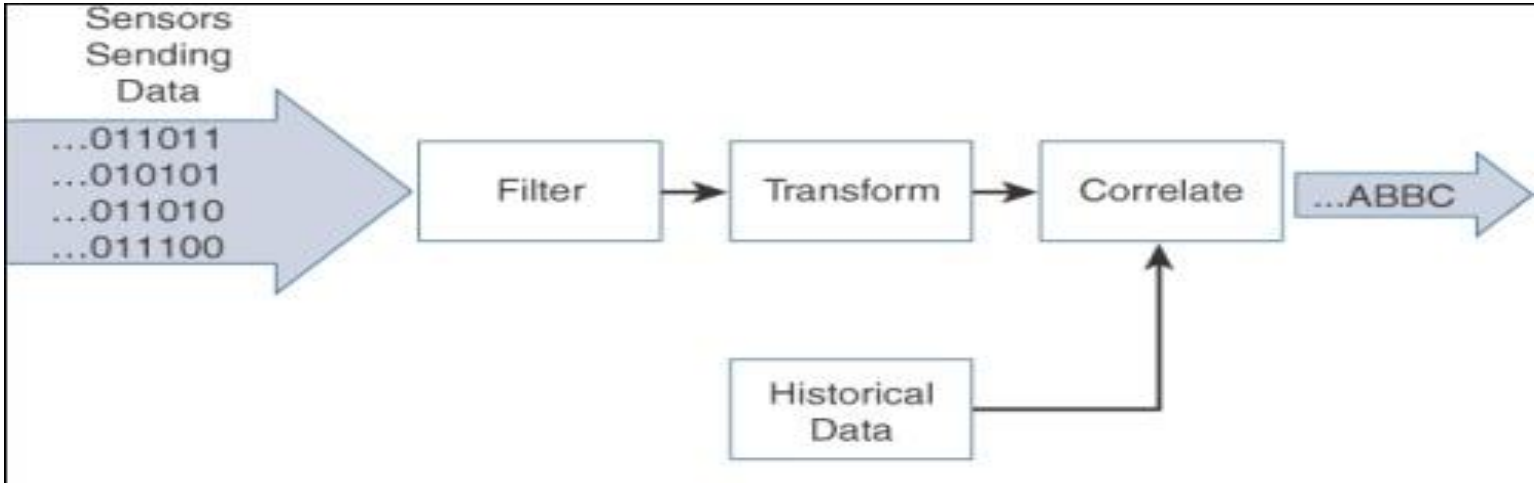
- The key values of edge streaming analytics include the following:
- **Reducing data at the edge**
- **Analysis and response at the edge**
- **Time sensitivity**

Edge Analytics Core Functions

- To perform analytics at the edge, data needs to be viewed as real-time flows.
- Whereas big data analytics is focused on large quantities of data at rest, edge analytics continually processes streaming flows of data in motion
- Streaming analytics at the edge can be broken down into three simple stages:
- **Raw input data**
- **Analytics processing unit (APU)**
- **Output streams**


Multiple Raw Input Streams
...011011
...010101
...011010
...011100
Edge Analytics Processing
Resulting Output Data
...ABBC
Hadoop
Storage and Deeper Analytics in Cloud

- In order to perform analysis in real-time, the APU needs to perform the following functions:
- **Filter**
- **Transform**
- **Time**
- **Correlate**



- **Match patterns**
- **Improve business intelligence**

Distributed Analytics Systems

- Depending on the application and network architecture, analytics can happen at any point throughout the IoT system.
- Streaming analytics may be performed directly at the edge, in the fog, or in the cloud data center.
- There are no hard and- fast rules dictating where analytics should be done, but there are a few guiding principles
- Sometimes better insights can be gained and data responded to more intelligently when we step back from the edge and look at a wider data set.
- This is the value of fog computing.

- Fog analytics allows you to see beyond one device, giving you visibility into an aggregation of edge nodes and allowing you to correlate data from a wider set
- Example of an oil drilling company that is measuring both pressure and temperature on an oil rig

Network Analytics

- Another form of analytics that is extremely important in managing IoT systems is network-based analytics
- Network analytics is concerned with discovering patterns in the communication flows from a network traffic perspective.
- Network analytics has the power to analyze details of communications patterns made by protocols and correlate this across the network.
- It allows you to understand what should be considered normal behavior in a network and to quickly identify anomalies that suggest network problems due to suboptimal paths, intrusive malware, or excessive congestion.

Securing IoT

- Information technology (IT) environments have faced active attacks and information security threats for many decades, and the incidents and lessons learned are well-known and documented.
- Operational technology (OT) environments were traditionally kept in silos and had only limited connection to other networks.
- Thus, the history of cyber attacks on OT systems is much shorter and has far fewer incidents documented
- Security in the OT world also addresses a wider scope than in the IT world. For example, in OT, the word *security is almost synonymous with safety*

- A Brief History of OT Security
- Common Challenges in OT Security
- How IT and OT Security Practices and Systems Vary
- Formal Risk Analysis Structures: OCTAVE and FAIR
- The Phased Application of Security in an Operational Environment

A Brief History of OT Security

- Cybersecurity incidents in industrial environments can result in physical consequences that can cause threats to human lives as well as damage to equipment, infrastructure, and the environment.
- While there are certainly traditional IT-related security threats in industrial environments, it is the physical manifestations and impacts of the OT security incidents that capture media attention and elicit broad-based public concern.

- Historically, attackers were skilled individuals with deep knowledge of technology and the systems they were attacking.
- However, as technology has advanced, tools have been created to make attacks much easier to carry out.
- To further complicate matters, these tools have become more broadly available and more easily obtainable.
- Compounding this problem, many of the legacy protocols used in IoT environments are many decades old, and there was no thought of security when they were first developed.
- This means that attackers with limited or no technical capabilities now have the potential to launch cyber attacks, greatly increasing the frequency of attacks and the overall threat to end operators.

Common Challenges in OT Security

- **Erosion of Network Architecture**
- Two of the major challenges in securing industrial environments have been initial design and ongoing maintenance.
- The initial design challenges arose from the concept that networks were safe due to physical separation from the enterprise with minimal or no connectivity to the outside world, and the assumption that attackers lacked sufficient knowledge to carry out security attacks.

- In many cases, the initial network design is sound and even follows well-defined industrial best practices and standards
- The challenge, and the biggest threat to network security, is standards and best practices either being misunderstood or the network being poorly maintained.
- It is more common that, over time, what may have been a solid design to begin with is eroded through ad hoc updates and individual changes to hardware and machinery without consideration for the broader network impact

Pervasive Legacy Systems

- Due to the static nature and long lifecycles of equipment in industrial environments, many operational systems may be deemed legacy systems
- In many cases, legacy components are not restricted to isolated network segments but have now been consolidated into the IT operational environment.
- From a security perspective, this is potentially dangerous as many devices may have historical vulnerabilities or weaknesses that have not been patched and updated

- Beyond the endpoints, the communication infrastructure and shared centralized compute resources are often not built to comply with modern standards.
- In fact, their communication methods and protocols may be generations old and must be interoperable with the oldest operating entity in the communications path.
- This includes switches, routers, firewalls, wireless access points, servers, remote access systems, patch management, and network management tools.
- All of these may have exploitable vulnerabilities and must be protected

Insecure Operational Protocols

- Many industrial control protocols, were designed without inherent strong security requirements
- Furthermore, their operation was often within an assumed secure network.
- In addition to any inherent weaknesses or vulnerabilities, their operational environment may not have been designed with secured access control in mind

- Industrial protocols, such as supervisory control and data acquisition (SCADA) ,particularly the older variants, suffer from common security issues.
- Three examples of this are, lack of authentication between communication endpoints, no means of securing and protecting data at rest or in motion, and insufficient granularity of control to properly specify recipients or avoid default broadcast approaches.

- The structure and operation of most of these protocols is often publicly available.
- While they may have been originated by a private firm, for the sake of interoperability, they are typically published for others to implement.
- Thus, it becomes a relatively simple matter to compromise the protocols themselves and introduce malicious actors that may use them to compromise control systems

Some common industrial protocols and their respective security concerns

- **Modbus**
- Modbus is commonly found in many industries, such as utilities and manufacturing environments, and has multiple variants (for example, serial, TCP/IP).
- It was created by the first programmable logic controller (PLC) vendor, Modicon, and has been in use since the 1970s.
- It is one of the most widely used protocols in industrial deployments, and its development is governed by the Modbus Organization.

- Authentication of communicating endpoints was not a default operation
- Some older and serial-based versions of Modbus communicate via broadcast.
- The ability to curb the broadcast function does not exist in some versions. There is potential for a recipient to act on a command that was not specifically targeting it
- Validation of the Modbus message content is also not performed by the initiating application

DNP3 (Distributed Network Protocol)

- DNP3 has placed great emphasis on the reliable delivery of messages
- In the case of DNP3, participants allow for unsolicited responses, which could trigger an undesired response.
- The missing security element here is the ability to establish trust in the system's state and thus the ability to trust the veracity of the information being presented

ICCP (Inter-Control Center Communications Protocol)

- Initial versions of ICCP had several significant gaps in the area of security.
- One key vulnerability is that the system did not require authentication for communication.
- Second, encryption across the protocol was not enabled as a default condition, thus exposing connections to man-in-the-middle (MITM) and replay attacks

International Electrotechnical Commission (IEC) Protocols

- Three message types were initially defined: MMS (Manufacturing Message Specification), GOOSE (Generic Object Oriented Substation Event), and SV (Sampled Values).
- Both GOOSE and SV operate via a publisher/subscriber model, with no reliability mechanism to ensure that data has been received
- Authentication is embedded in MMS, but it is based on clear-text passwords, and authentication is not available in GOOSE or SV

Device Insecurity

- Beyond the communications protocols that are used and the installation base of legacy systems, control and communication elements themselves have a history of vulnerabilities
- It is not difficult to understand why such systems are frequently found vulnerable.
- First, many of the systems utilize software packages that can be easily downloaded and worked against.
- Second, they operate on common hardware and standard operating systems, such as Microsoft Windows.
- Third, Windows and the components used within those applications are well known to traditionally IT-focused security researchers, There is little need to develop new tools or techniques

Dependence on External Vendors

- Modern IT environments may be outsourcing business operations or relegating certain processing or storage functions to the cloud

How IT and OT Security Practices and Systems Vary

- The differences between an enterprise IT environment and an industrial-focused OT deployment are important to understand because they have a direct impact on the security practice applied to them
- **The Purdue Model for Control Hierarchy**
- Regardless of where a security threat arises, it must be consistently and unequivocally treated
- IT information is typically used to make business decisions, such as those in process optimization, whereas OT information is instead characteristically leveraged to make physical decisions, such as closing a valve, increasing pressure, and so on

- The operational domain must also address physical safety and environmental factors as part of its security strategy —and this is not normally associated with the IT domain

The Purdue Model for Control Hierarchy

Enterprise Zone	Enterprise Network	Level 5
	Business Planning and Logistics Network	Level 4
DMZ	Demilitarized Zone — Shared Access	
Operations Support	Operations and Control	Level 3
Process Control / SCADA Zone	Supervisory Control	Level 2
	Basic Control	Level 1
	Process	Level 0
Safety	Safety-Critical	

OT Network Characteristics Impacting Security

- While IT and OT networks are beginning to converge, they still maintain many divergent characteristics in terms of how they operate and the traffic they handle.
- These differences influence how they are treated in the context of a security strategy.
- **IT networks**
- traffic traverse far
- They frequently traverse the network through layers of switches and eventually make their way to a set of local or remote servers, which they may connect to directly

- **OT networks**
- By comparison, in an OT environment (Levels 0–3), there are typically two types of operational traffic.
- The first is local traffic that may be contained within a specific package or area to provide local monitoring and closed-loop control.
- This is the traffic that is used for real time (or near-real-time) processes and does not need to leave the process control levels.
- The second type of traffic is used for monitoring and control of areas or zones or the overall system.