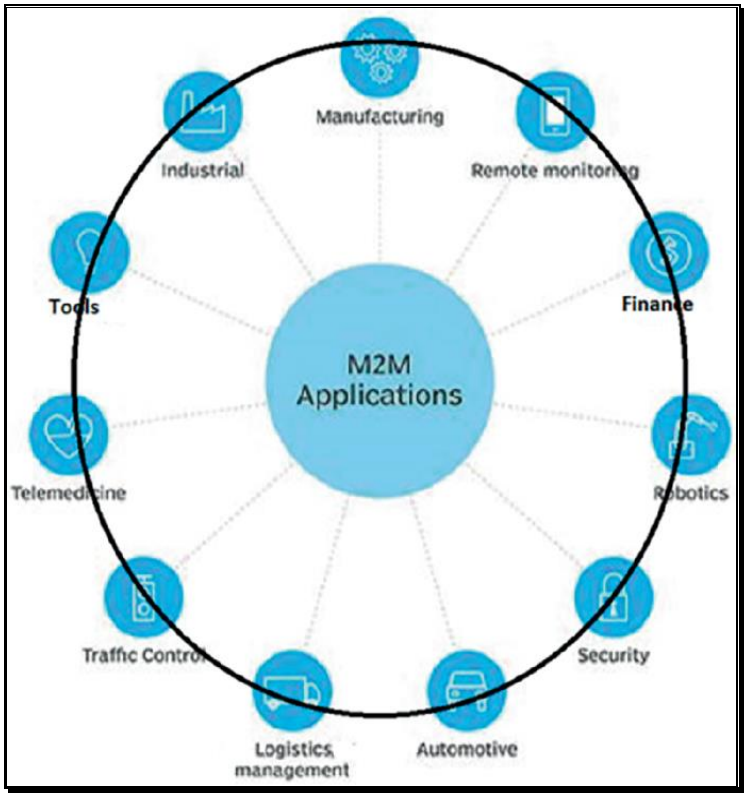
Unit II : Artificial Intelligence for IoT

M2M Communication:

- M2M (Machine-to-Machine) communication in IoT refers to the direct exchange of data between devices without human intervention, enabling autonomous operation and control.
- It's a foundational aspect of the Internet of Things, facilitating seamless data transfer and automated actions within networks of connected devices.
- M2M technology allows device to communicate, collect data from sensors, and set it to a network.
- M2M is great because it uses low-cost networks like cellular or Ethernet rather than expensive tool.
- An m2m includes key parts like sensors, RFID, wifi or cellular network and smart software that helps devices understand data and makes decision.



Dig. M2M applications

Key features of Machine to Machine:

- **Efficiency:**

  M2M enables automation and remote management, leading to increased efficiency and reduced operational costs.

- **Real-time Insights:**

  Continuous data collection and analysis provide real-time insights for better decision-making.

- **Improved Customer Experience:**

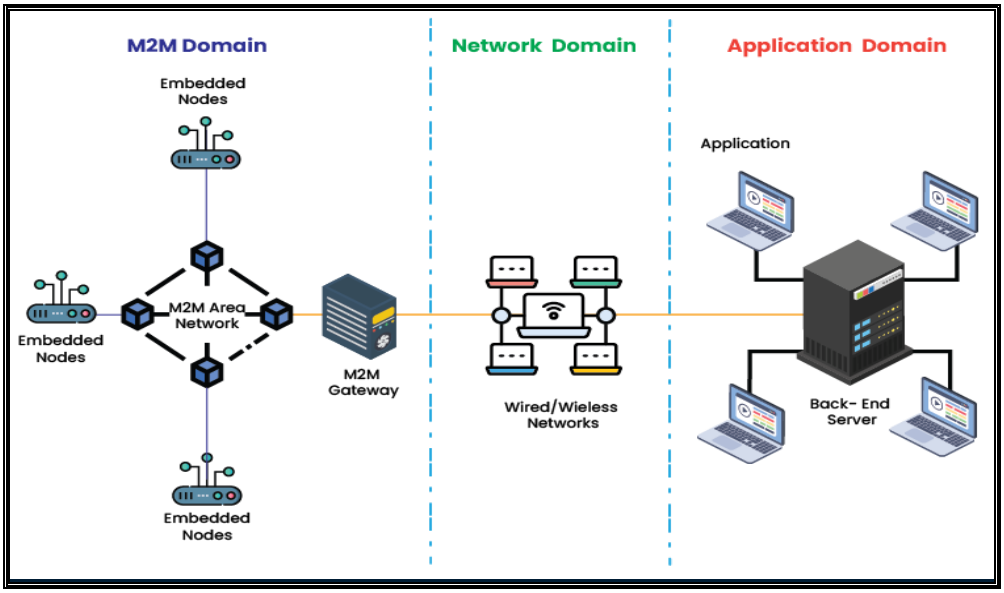
  Proactive maintenance and personalized services are enabled through M2M communication.

- **New Revenue Streams:**

  M2M facilitates the development of new products and services based on connected devices.

- **Reduced Downtime:**
  Predictive maintenance and remote troubleshooting minimize downtime and improve system reliability.

M2M Working:



Dig. M2M working

M2M in IoT refers to the technology that allows device to communicate with each other, collect data and transmit it over a network without human intervention. M2M utilizes sensors to gather data and specialized software to analyse and make decisions based on the data, enabling automation and remote monitoring capabilities.

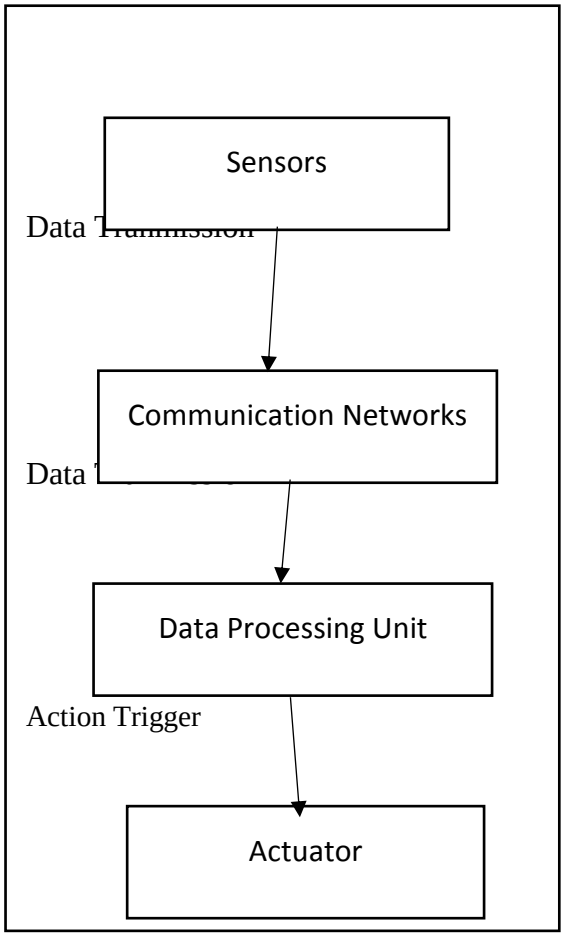
Following are the working principles:

1) **Data Acquisition:** Sensors are attached to machines or devices to collect data about their status, environment or performance. These sensors can measure various parameters like temperature, pressure etc. depending on the application.

2) **Data Transmission:** The collected data is transmitted wirelessly or through wired connections to a central system or other device. Common communication methods include cellular networks, wifi, Ethernet etc.
3) **Data Processing and Analysis:** The data is processed and analysed by software often residing in the cloud or a local server. This software can identify patterns and can predict potential issues.
4) **Action and Decision Making:** Based on the analysed data, the system can trigger automated actions, such as sending data/alters, adjusting machine settings, or initiating maintenance procedure. Human intervention may be required for complex decisions or situations, but m2m aims to minimize this.

Key Components of M2M:

1) Sensors: These devices gather data about the environment or a machines status, such as temperature, pressure or motion.
2) Communication Networks: These networks which can be wired or wireless e.g. wifi, cellular, Bluetooth transmit the sensor data to processing units.
3) Data Processing Units: These units analyse the data and make decisions based on predefined algorithms.
4) Actuators: These components execute actions based on the processed data, such as adjusting temperature or triggering an alarm.
5) Software Applications: These manage the entire system, including device configuration, data visualization and automated actions.



- **Advantages of M2M communication:**

M2M (Machine-to-Machine) communication offers numerous advantages, including increased efficiency, reduced costs, enhanced safety, and improved decision-making. It enables automation, remote monitoring and control of devices, and real-time data insights, leading to optimized operations and proactive problem-solving.

1. Increased Efficiency and Automation:

- M2M facilitates automated data exchange, minimizing the need for manual intervention and streamlining processes.
- This automation can lead to faster responses to changing conditions, improved productivity, and reduced human error.
- For example, a machine can autonomously trigger maintenance requests when needed or a fleet management system can automatically optimize delivery routes.

2. Cost Reduction:

- By enabling remote monitoring and control, M2M reduces the need for physical site visits, lowering travel costs and maintenance expenses.
- Improved efficiency and reduced downtime translate to lower operational costs.
- M2M can also identify potential issues early on, preventing costly repairs or replacements.

3. Enhanced Safety:

- M2M can be used in hazardous environments or for remote monitoring of critical infrastructure, improving worker safety.
- Real-time data and alerts can help prevent accidents and ensure a safer working environment.

4. Improved Decision-Making:

- M2M provides real-time data insights, allowing for more informed and data-driven decision-making.
- This data can be used to optimize processes, predict potential problems, and improve overall system performance.

5. Scalability and Flexibility:

- M2M systems can be easily scaled to accommodate growing business needs and adapt to changing requirements.
- The flexibility of M2M allows for deployment in various locations and environments, both indoors and outdoors.

6. New Business Opportunities:

- M2M can create new business models by enabling remote services, predictive maintenance, and personalized customer experiences.
- Companies can leverage M2M data to identify new product or service offerings.

7. Improved Customer Service:

- Proactive monitoring and servicing of equipment through M2M can lead to fewer disruptions and improved customer satisfaction.
- Real-time data can also be used to provide more personalized and responsive customer support.

In essence, M2M communication empowers businesses to operate more efficiently, reduce costs, enhance safety, and make better decisions, ultimately leading to increased profitability and improved customer satisfaction.

INSHORTS:

Benefits or Advantages of M2M Communication

The following are benefits of M2M Communication:

1. M2M communication is supported by cellular networks either directly or through a gateway.
2. It is easy to deploy and maintain.
3. It is available with fixed and mobile networks, both indoors and outdoors.
4. It offers a higher range, minimum latency, higher throughput, and consumes less energy.
5. It enables communication between smart devices without human intervention.
6. M2M communication helps resolve security and privacy issues in IoT networks.
7. Large-scale protection, data collection, and data processing become possible

- ## Disadvantage of M2M communication:

M2M (Machine-to-Machine) communication, while offering numerous benefits, also presents several disadvantages. These include security vulnerabilities, complexity in implementation, interoperability issues, dependence on network availability, and data privacy concerns.

Here's a more detailed look at the disadvantages:

1. Security Risks:

- M2M devices, often operating without human oversight, are susceptible to cyberattacks, including hacking, data breaches, and unauthorized access.
- The constant exchange of potentially sensitive data requires robust security protocols, which can be complex and costly to implement.
- Remote management of M2M systems, necessary for updates and maintenance, can introduce vulnerabilities if not handled securely.

2. Complexity and Interoperability:

- Implementing M2M technology can be complex, requiring specialized skills and potentially leading to high implementation costs.
- Ensuring compatibility between devices from different manufacturers (interoperability) can be a significant challenge.

3. Dependence on Network Availability:

- M2M devices rely on stable and reliable network connections, and outages or poor network quality can disrupt communication and functionality.
- Constant connectivity is necessary, and interruptions can cause inefficiencies and gaps in data.
- M2M systems generate large amounts of data, and managing and analysing this data can be a challenge.
- Collecting and storing personal data raises concerns about data privacy and security, requiring robust measures to protect sensitive information.

5. Other Considerations:

- **Limited Battery Life:**

  Many M2M devices operate on batteries, which can have limited lifespans and require frequent replacements.

- **Limited Flexibility:**

  Dedicated M2M systems may be designed for specific tasks or industries, limiting their adaptability to new requirements and making upscaling more resource-intensive.

- **Cost:**

  Implementing and maintaining M2M systems can be expensive, especially when considering security measures, network infrastructure, and device maintenance.

- **Dependence on Cloud Providers:**
  Cloud-based M2M solutions can create dependence on external providers, potentially impacting flexibility and innovation.

INSHORTS:

1. Using cloud computing in M2M creates a dependence on external providers, potentially limiting flexibility and innovation.
2. Security and data ownership are significant concerns.
3. Interoperability between cloud/M2M IoT devices remains a challenge.
4. It's designed and optimized for a relatively small number of network devices.
5. M2M communication relies on a constant internet connection with reasonable speed.

Difference between IoT and M2M:

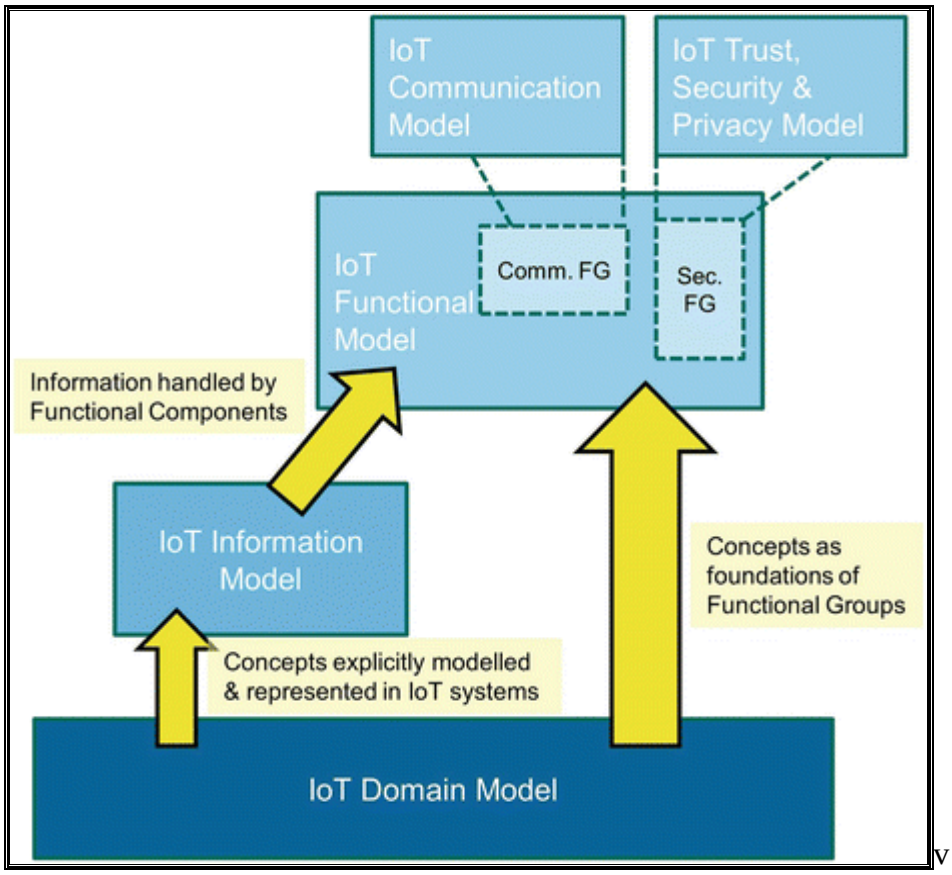
Basis of	IoT	M2M
Abbreviation	Internet of Things	Machine to Machine
Intelligence	Devices have objects that are responsible for decision making	Some degree of intelligence is observed in this.
Connection type used	The connection is via Network and using various communication types.	The connection is a point to point
Communication protocol used	Internet protocols are used such as HTTP, FTP, and Telnet.	Traditional protocols and communication technology techniques are used

Basis of	IoT	M2M
Data Sharing	Data is shared between other applications that are used to improve the end-user experience.	Data is shared with only the communicating parties.
Internet	Internet connection is required for communication	Devices are not dependent on the Internet.
Type of Communication	It supports cloud communication	It supports point-to-point communication.
Computer System	Involves the usage of both Hardware and Software.	Mostly hardware-based technology
Scope	A large number of devices yet scope is large.	Limited Scope for devices.
Business Type used	Business 2 Business(B2B) and Business 2 Consumer(B2C)	Business 2 Business (B2B)
Open API support	Supports Open API integrations.	There is no support for Open APIs
It requires	Generic commodity devices.	Specialized device solutions.
Centric	Information and service centric	Communication and device centric.
Approach used	Horizontal enabler approach	Vertical system solution approach .
Components	Devices/sensors, connectivity, data processing, user interface	Device, area networks, gateway, Application server.
Examples	Smart wearables, Big Data and Cloud, etc.	Sensors, Data and Information, etc.

Reference Model and Architecture

•An ARM consists of two main parts: a Reference model and a Reference Architecture.

•A reference model describes the domain using a number of sub-models



Dig. Reference Model Diagram

- **IoT domain model**

The domain model captures the basic attributes of the main concepts and the relationship between these concepts. A domain model also serves as a tool for human communication between people working in the domain in question and between people who work across different domains.

The IoT is a support infrastructure for enabling objects and places in the physical world to have a corresponding representation in the digital world.

The Devices are physical artefacts with which the physical and virtual worlds interact. Devices as mentioned before can also be Physical Entities for certain types of applications, such as management applications when the interesting entities of a system are the Devices themselves and not the surrounding environment. For the IoT Domain Model, three kinds of Device types are the most important:

1. Sensors:

- These are simple or complex Devices that typically involve a transducer that converts physical properties such as temperature into electrical signals.
- These Devices include the necessary conversion of analog electrical signals into digital signals, e.g. a voltage level to a 16-bit number, processing for simple calculations, potential storage for intermediate results, and potentially communication capabilities to transmit the digital representation of the physical property as well receive commands.
- A video camera can be another example of a complex sensor that could detect and recognise people.

2. Actuators:

- These are also simple or complex Devices that involve a transducer that converts electrical signals to a change in a physical property (e.g. turn on a switch or move a motor).
- These Devices also include potential communication capabilities, storage of intermediate commands, processing, and conversion of digital signals to analog electrical signals.

3. Tags:

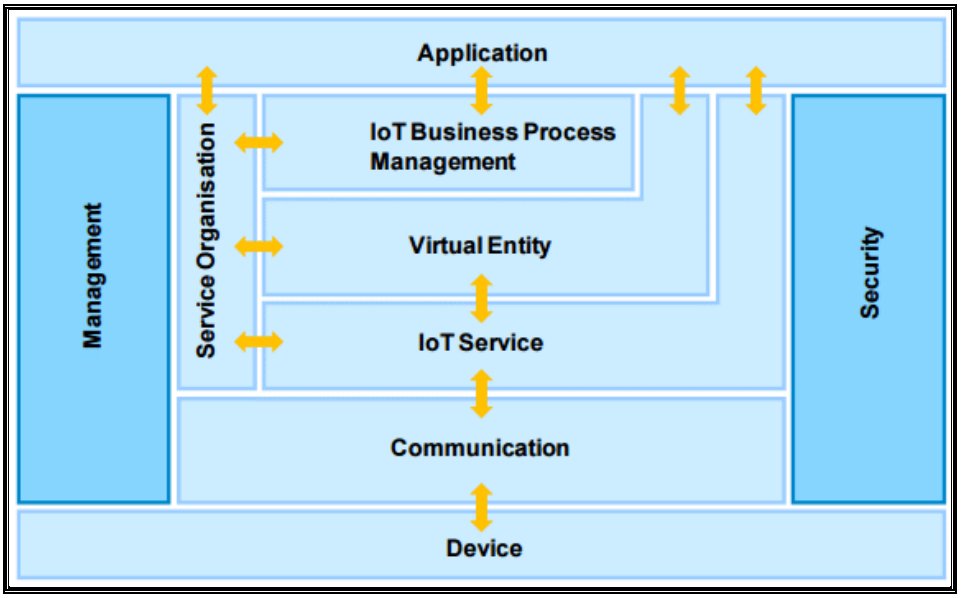
- Tags in general identify the Physical Entity that they are attached to. In reality, tags can be Devices or Physical Entities but not both, as the domain model shows.
- An example of a Tag as a Device is a Radio Frequency Identification (RFID) tag, while a tag as a Physical Entity is a paper-printed immutable barcode or Quick Response (QR) code.
- Either electronic Devices or a paper-printed entity tag contains a unique identification that can be read by optical means (bar codes or QR codes) or radio signals (RFID tags).
- The reader Device operating on a tag is typically a sensor, and sometimes a sensor and an actuator combined in the case of writable RFID tags.

- ***Information Model***

irtual Entity in the IoT Domain Model is the "Thing" in the Internet of Things, the IoT information model captures the details of a Virtual Entity- centric model. Similar to the IoT Domain Model, the IoT Information Model is presented using Unified Modelling Language (UML) diagrams.

- ***Functional model***

The IoT Functional Model aims at describing mainly the Functional Groups (FG) and their interaction with the ARM, while the Functional View of a Reference Architecture describes the functional components of an FG, interfaces, and interactions between the components. The Functional View is typically derived from the Functional Model in conjunction with high-level requirements.



Dig. Functional Model

Device functional group

The Device FG contains all the possible functionality hosted by the physical Devices that are used for increment the Physical Entities. This Device functionality includes sensing, actuation, processing, storage, and identification components, the sophistication of which depends on the Device capabilities

Communication functional group

The Communication FG abstracts all the possible communication mechanisms used by the relevant Devices in an actual system in order to transfer information to the digital world components or other Devices.

IoT Service functional group

The IoT Service FG corresponds mainly to the Service class from the IoT Domain Model, and contains single IoT Services exposed by Resources hosted on Devices or in the Network (e.g. processing or storage Resources).

Virtual Entity functional group

The Virtual Entity FG corresponds to the Virtual Entity class in the IoT Domain Model, and contains the necessary functionality to manage associations between Virtual Entities with themselves as well as associations between Virtual Entities and related IoT Services, i.e. the Association objects for the IoT Information Model. Associations between Virtual Entities can be static or dynamic depending on the mobility of the Physical Entities related to the corresponding Virtual Entities.

IoT Service Organization functional group

The purpose of the IoT Service Organisation FG is to host all functional components that support the composition and orchestration of IoT and Virtual Entity services. Moreover, this FG acts as a service hub between several other functional groups such as the IoT Process Management FG when, for example, service requests from Applications or the IoT Process Management are directed to the Resources implementing the necessary Services.

IoT Process Management functional group

The IoT Process Management FG is a collection of functionalities that allows smooth integration of IoT-related services (IoT Services, Virtual Entity Services, Composed Services) with the Enterprise (Business) Processes.

- **Management functional group**

The Management FG includes the necessary functions for enabling fault and performance monitoring of the system, configuration for enabling the system to be flexible to changing User demands, and accounting for enabling subsequent billing for the usage of the system. Support functions such as management of ownership, administrative domain, rules and rights of functional components, and information stores are also included in the Management FG.

- **Security functional group**

The Security FG contains the functional components that ensure the secure operation of the system as well as the management of privacy. The Security FG contains components for Authentication of Users (Applications, Humans), Authorisation of access to Services by Users, secure communication (ensuring integrity and confidentiality of messages) between entities of the system such as Devices, Services, Applications, and last but not least, assurance of privacy of sensitive information relating to Human Users.

- **Application functional group**

The Application FG is just a placeholder that represents all the needed logic for creating an IoT application. The applications typically contain custom logic tailored to a specific domain such as a Smart Grid

  - ***Communication model***

- **Safety**

the IoT Reference Model can only provide IoT-related guidelines for ensuring a safe system to the extent possible and controllable by a sys- tem designer.
Eg: smart grid.

- **Privacy**

Because interactions with the physical world may often include humans, protecting the User privacy is of utmost importance for an IoT system. The IoT-A Privacy Model depends on the following functional components: Identity Management, Authentication, Authorisation, and Trust & Reputation

- **Trust**

Generally, an entity is said to 'trust' a second entity when the first entity makes the assumption that the second entity will behave exactly as the first entity expects."

- **Security**

The Security Model for IoT consists of communication security that focuses mostly on the confidentiality and integrity protection of interacting entities and functional components such as Identity Management, Authentication, Authorisation, and Trust & Reputation.

Verticals:

IoT (Internet of Things) verticals refer to the diverse range of industries and applications where IoT technologies are being implemented. These verticals encompass various sectors, each with unique requirements and opportunities for leveraging connected devices and data analysis.

Here's a breakdown of some key IoT verticals:

1. Smart Buildings:

- **Description:**

  Monitoring and managing building systems like lighting, HVAC, security, and energy consumption.

- **Benefits:**
  Increased efficiency, reduced energy costs, improved occupant comfort, and enhanced security.
  2. Industrial IoT (IIoT):

- **Description:**

  Applying IoT in manufacturing, supply chain, and logistics to optimize processes, improve asset management, and enhance safety.

- **Benefits:**
  Reduced downtime, increased productivity, improved product quality, and optimized resource utilization.
  3. Transportation & Logistics:

- **Description:**

  Tracking and managing vehicles, optimizing delivery routes, and improving fleet management.

- **Benefits:**
  Reduced fuel costs, improved delivery times, enhanced safety, and better asset utilization.
  4. Healthcare:

- **Description:**

  Remote patient monitoring, wearable health trackers, and connected medical devices for improved healthcare delivery and patient outcomes.

- **Benefits:**
  Early disease detection, personalized treatment plans, reduced hospital readmissions, and improved patient engagement.
  5. Energy:

- **Description:**

  Smart grids, energy management systems, and renewable energy monitoring for efficient energy distribution and consumption.

- **Benefits:**
  Reduced energy waste, improved grid stability, and optimized energy production.
  6. Agriculture:

- **Description:**

  Precision farming, automated irrigation, and livestock monitoring for optimized crop yields and resource management.

- **Benefits:**
  Increased crop yields, reduced water consumption, and improved livestock health.
  7. Retail:

- **Description:**

  Smart shelves, personalized shopping experiences, and inventory management for enhanced customer experience and operational efficiency.

- **Benefits:**
  Increased sales, reduced inventory costs, and improved customer satisfaction.
  8. Finance:

- **Description:** Mobile payments, fraud detection, and risk management for secure and efficient financial transactions.

- **Benefits:** Reduced fraud, improved security, and enhanced customer experience.
  9. Smart Cities:

- **Description:**

  Environmental monitoring, traffic management, and public safety solutions for improving the quality of life in urban areas.

- **Benefits:**
  Reduced traffic congestion, improved air quality, enhanced public safety, and efficient resource management.
  These are just a few examples, and the list of IoT verticals is constantly expanding as new technologies and applications emerge. The key takeaway is that IoT is transforming various industries by enabling data-driven insights and automation, leading to increased efficiency, productivity, and improved outcomes.

Big Data and IoT:

Big data and the Internet of Things (IoT) are closely related concepts, with IoT generating the massive datasets that are then analyzed by big data technologies. IoT refers to the network of interconnected devices that collect and transmit data, while big data encompasses the large volumes of data generated and the techniques used to analyze it. Essentially, IoT provides the data, and big data provides the means to extract valuable insights from it.

Big Data is generally defined by four major characteristics: Volume, Velocity, Variety and Veracity. Let's discuss all 4 V's of Big Data in detail:

Volume

As we know, the first Characteristic of Big Data is its Volume. Trillions of gigabytes of data are created worldwide every day, and the numbers will only rise in the years to come. Most of this massive quantity of data generated daily is due to the widespread use of mobile phones.

Text, photos, videos and applications create lots of data every day, and it will only increase with mobile phone use in the coming years. As the Volume of data sees exponential growth, new Database Management Systems and IT employees will be needed to handle it. As a result, millions of new IT jobs are expected to open in the coming years – thanks to the Volume of Big Data.

Example: Social media platforms like Facebook generate petabytes of data daily through posts, comments, and multimedia uploads.

Velocity

Velocity, or speed as many would call it, is the second characteristic of Big Data. It refers to the unprecedented speed at which data is generated and processed. The data is processed instantaneously if you send a text or post anything on social media such as Facebook, Twitter or Instagram.

Processing and surfacing information once was a time-consuming process, but with the advent of the internet – it now takes next to no time. However, this is not just due to the internet, but also the existence of data. As more data is created, more methods, such as a Big Data Platform, are required to handle it, and the more data is monitored – thereby creating a cycle.

Example: Financial markets require real-time Data Processing to execute trades and manage risk effectively.

Variety

The first and second characteristics of Big Data correlate with the third character, the Variety of Big Data. As we know, data is large in volume and fast to process – and at the same time, data comes in many types. Organisations and individuals generate and process data according to their specific needs, ensuring a wide variety of data on the planet.

IT solutions are available to almost all sectors, from business to banking and the medical to the sports industry. Nearly every organisation in every sector keeps a database, contributing to a wide Variety of data. When the internet reaches the farthest corners of the world, the Variety of data will only increase.

Example: Emails, social media posts, videos, and sensor data all represent different forms of data collected by companies.

Veracity

The final characteristic attributed to Big Data is its Veracity. Veracity can be defined as conformity to facts or accuracy, and it refers to the element of accuracy that data possesses. The Veracity of Big Data refers to the data's trustworthiness and denotes the data's accuracy and quality.

Data quickly becomes outdated, and with its abundance, it is tricky to determine the authenticity of everything you see. This is why many upper-level businesspeople do not dare make decisions only on data. This also grants impetus to data Scientists and IT Professionals to organise and process the right data so that it can be used correctly. The higher the veracity, the more important the data is as it is set to be analysed and converted to meaningful information.

Example: Customer feedback data may be filled with inconsistencies, biases, or inaccuracies, requiring validation and cleansing to ensure reliability.

Difference between IoT and Big Data:

SR. NO.	IOT	BIG DATA
1	IoT is a global system of interrelated computing devices that are able to sense, collect, and exchange data over the Internet.	Big Data is described as large sets of data generated from a variety of sources that are so large to process using traditional techniques.
2	The concept is to provide interconnection between devices to create a smart environment thereby making machines smart enough to bypass human intermediaries.	The concept is to find insights in new and emerging types of data and content that lead to better decisions and strategic business moves.
3	IoT collects, analyzes, and processes data streams in real-time without any delay to make control decisions in an effective manner.	The data streams are not subjected to processing real-time and there is a delay between when the data is collected and when it is processed.
4	IoT involves analyzing machine-generated data such as sensors in home appliances and so on.	Big Data deals with human-generated data such as social media usage, photos, and videos, etc

SR. NO.	IOT	BIG DATA
5	IoT is about simultaneously collecting and processing data to make real-time decisions.	Big data is more into collecting and accumulating huge data for analysis afterward.
6	Using IoT you can track and monitor assets like trucks, engines, HVAC systems, and pumps. You can correct problems as you detect them.	With big data, you can analyze all the information you have about failures and start to uncover the root causes.

➢ Introduction to tensorflow and keras:

TensorFlow is an open-source framework for machine learning (ML) and artificial intelligence (AI) that was developed by Google Brain. It was designed to facilitate the development of machine learning models, particularly deep learning models by providing tools to easily build, train and deploy them across different platforms.

TensorFlow supports a wide range of applications from natural language processing (NLP) and computer vision (CV) to time series forecasting and reinforcement learning.

Key Features of TensorFlow

1. Scalability

TensorFlow is designed to scale across a variety of platforms from desktops and servers to mobile devices and embedded systems. It supports distributed computing allowing models to be trained on large datasets efficiently.

2. Comprehensive Ecosystem

TensorFlow offers a broad set of tools and libraries including:

- **TensorFlow Core:** The base API for TensorFlow that allows users to define models, build computations and execute them.
- **Keras:** A high-level API for building neural networks that runs on top of TensorFlow, simplifying model development.
- **TensorFlow Lite:** A lightweight solution for deploying models on mobile and embedded devices.
- **TensorFlow.js:** A library for running machine learning models directly in the browser using JavaScript.
- **TensorFlow Extended (TFX):** A production-ready solution for deploying machine learning models in production environments.
- **TensorFlow Hub:** A repository of pre-trained models that can be easily integrated into applications.

3. Automatic Differentiation (Autograd)

TensorFlow automatically calculates gradients for all trainable variables in the model which simplifies the backpropagation process during training. This is a core feature that enables efficient model optimization using techniques like gradient descent.

4. Multi-language Support

TensorFlow is primarily designed for Python but it also provides APIs for other languages like C++, Java and JavaScript making it accessible to developers with different programming backgrounds.

5. TensorFlow Serving and TensorFlow Model Optimization

TensorFlow includes tools for serving machine learning models in production environments and optimizing them for inference allowing for lower latency and higher efficiency.

Advantages:

1. Scalable :Tensorflow is not limited to one specific device. It works as efficiently on a cellular device as it works on any other complex machine. The library is so defined that its deployment isn't limited to any one specific device.

2. Open Source Platform :It is available free of cost to anyone who wants to work with this. This feature makes it possible for any user to employ this module whenever and wherever required.

3. Graphs : Tensorflow has a better data visualization power than any other available library. This makes it working on neural networks easier.

4. Debugging : Tensorflow has Tensorboard which allows easy debugging of nodes. This helps reduce the overhead of visiting the whole code.

5. Parallelism : TensorFlow employs GPU and CPU systems for its functioning. A user is free to use any of the architecture as per requirement. A system uses GPU if not mentioned explicitly. This process reduces memory usage to some extent. Due it this capacity Tensorflow is viewed as a hardware acceleration library.

6. Compatible : It is compatible with many programming languages like Python, C++, JavaScript, etc. This allows users to work in an environment they are comfortable in.

7. Architectural Support : The TensorFlow architecture uses TPU which makes computation faster than CPU and GPU. Models which are built over TPU can be easily deployed over clouds and works faster compared to the other two.

8. Library management : Being backed by google, Tensorflow is updated frequently and is capable of displaying outstanding performance.

Disadvantages:

1. No windows support : Besides all the advantages possessed by Tensorflow, it has a very limited set of features for Windows users. For Linux users this isn't the case there is a wide arcade of features when it comes to them.

2. Slow : It is comparatively slower and less usable compared to its competing frameworks.

3. GPU support : Tensorflow has only NVIDIA support for GPU and Python programming language support for GPU programming.

4. Frequent updates : Tensorflow undergoes frequent updates making it overhead for a user to time to time uninstall and reinstall it so that it can bind and be blended with its latest updates.

5. Architectural limitation : Tensorflow's TPU architecture allows only execution of models and doesn't allow its training.

6. Inconsistent : Tensorflow contains homonyms as names of its contents which makes it difficult for a user to remember and use. A single name is used for various different purposes and this is where the confusion starts.

7. Dependency : Even though TensorFlow reduces the size of the program and makes it user-friendly, it adds a layer of complexity to it. Every code needs some platform for its execution which increases dependency.

8. Symbolic loops : Tensorflow lags at providing symbolic loops for indefinite sequences. Its support for definite sequences makes it a useful resource.

Keras:

Keras is a deep learning API that simplifies the process of building deep neural networks. Initially it was developed as an independent library, Keras is now tightly integrated into TensorFlow as its official high-level API. It supports multiple backend engines like TensorFlow, Theano and Microsoft Cognitive Toolkit (CNTK).

Keras makes it easier to train and evaluate deep learning models without requiring extensive knowledge of low-level operations.

Key Features of Keras Library

Aspect	Key Points
Simplicity	High-level API for the full ML workflow and reduces manual effort with clear, productive interfaces.
Customizability	Easily extendable with custom layers, losses and preprocessing, It supports both functional API and low-level model building.
Compatibility	Runs on TensorFlow, Theano and CNTK allowing consistent performance across CPU and GPU platforms.
Scalability and Performance	Leverages scalable backends like TensorFlow

Aspect	Key Points
Fast Experimentation	Uses simple tools to build models quickly, making it great for research and testing

Advantages of Keras in IoT:

- **Ease of Use and Rapid Prototyping:**

  Keras is known for its simple, intuitive API, making it relatively easy to learn and use, especially for beginners in deep learning. This allows for quick development and testing of models for IoT applications.

- **Cross-Platform Compatibility:**

  Keras can be deployed on various platforms, including those commonly used in IoT, such as Raspberry Pi and other embedded systems.

- **Large Community Support:**

  Keras boasts a large and active community, providing ample resources, tutorials, and support for developers working on IoT projects.

- **Backend Flexibility:**

  Keras can run on different backend engines like TensorFlow, Theano, or CNTK, allowing developers to choose the best fit for their specific needs and hardware.

- **Pre-trained Models:**
  Keras offers a wide range of pre-trained models, which can be fine-tuned for specific IoT tasks, reducing development time and effort.

Disadvantages of Keras in IoT:

- **Performance Overhead:**

  Keras, being a high-level API, can introduce some performance overhead compared to lower-level frameworks like TensorFlow or PyTorch directly. This can be a concern for resource-constrained IoT devices with limited processing power and memory.

- **Limited Low-Level Control:**

  Keras abstracts away many low-level details of the underlying hardware and computation. This can be a limitation when specific optimizations or fine-grained control are needed for optimal performance on IoT devices.

- **Debugging Challenges:**

  While Keras has debugging features, some users report that error messages can be cryptic and debugging can be challenging, especially when dealing with low-level issues or complex model architectures.

- **Not Optimized for All IoT Scenarios:**

  Keras might not be the best choice for all IoT applications. For example, some IoT devices might require specialized algorithms or hardware acceleration that are not directly supported by Keras.

- **Potential Bottlenecks:**
  While Keras can be faster for prototyping, the actual training and inference on IoT devices might be slower compared to more optimized solutions.

Difference between TensorFlow and Keras:

S.No	TensorFlow	Keras
1.	Tensorhigh-performanceFlow is written in C++, CUDA, Python.	Keras is written in Python.
2.	TensorFlow is used for large datasets and high performance models.	Keras is usually used for small datasets.
3.	TensorFlow is a framework that offers both high and low-level APIs.	Keras is a high-Level API.
4.	TensorFlow is used for high-performance models.	Keras is used for low-performance models.
5.	In TensorFlow performing debugging leads to complexities.	In Keras framework, there is only minimal requirement for debugging the simple networks.
6.	TensorFlow has a complex architecture and not easy to use.	Keras has a simple architecture and easy to use.
7.	TensorFlow was developed by the Google Brain team.	Keras was developed by François Chollet while he was working on the part of the research effort of project ONEIROS.