MODULE 5: Vehicular IoT

5.1. Introduction

The use of connected vehicles is increasing rapidly across the globe. The number of on-road accidents and mismanagement of traffic is also increasing. The increasing number of vehicles gives rise to the problem of parking. The evolution of IoT helps to form a connected vehicular environment to manage the transportation systems efficiently. Vehicular IoT systems have penetrated different aspects of the transportation ecosystem, including on-road to off-road traffic management, driver safety for heavy to small vehicles, and security in public transportation.

In a connected vehicular environment, vehicles are capable of communicating and sharing their information. Moreover, IoT enables a vehicle to sense its internal and external environments to make certain autonomous decisions. With the help of modern-day IoT infrastructure, a vehicle owner residing in Earth's northern hemisphere can very easily track his vehicular asset remotely, even if it is in the southern hemisphere. In this module we discuss the importance and applications of IoT in the vehicular systems.
Figure 5.1 represents a simple architecture of a vehicular IoT system.

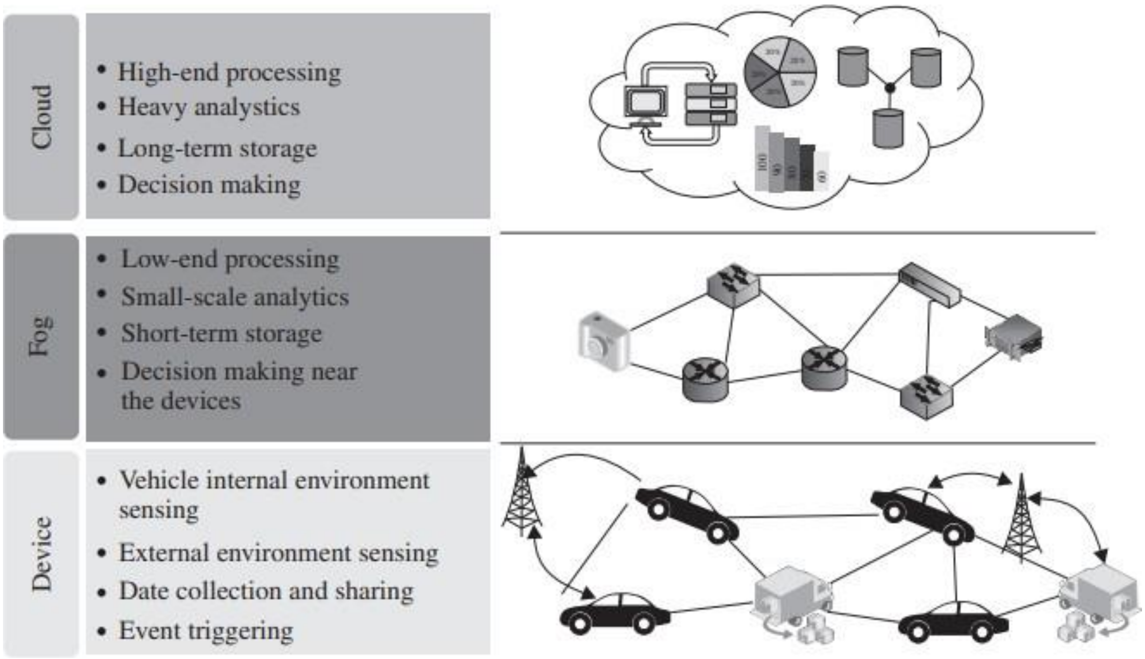


Figure 5.1 Architecture of a vehicular IoT.

The architecture of the vehicular IoT is divided into three sublayers: device, fog, and cloud.

Device: The device layer is the bottom-most layer, which consists of the basic infrastructure of the scenario of the connected vehicle. This layer includes the vehicles and road side units (RSU). These vehicles contain certain sensors which gather the internal information of the vehicles. On the other hand, the RSU works as a local centralized unit that manages the data from the vehicles.

Fog: In vehicular IoT systems, fast decision making is pertinent to avoid accidents and traffic mismanagement. In such situations, fog computing plays a crucial role by providing decisions in real-time, much near to the devices. Consequently, the fog layer helps to minimize data transmission time in a vehicular IoT system.

Cloud: Fog computing handles the data processing near the devices to take decisions instantaneously. However, for the processing of huge data, fog computing is not enough. Therefore, in such a situation, cloud computing is used. In a vehicular IoT system, cloud computing helps to handle processes that involve a huge amount of data. Further, for long-term storage, cloud computing is used as a scalable resource in vehicular IoT systems.

5.1.1 Components of vehicular IoT

Modern cars come equipped with different types of sensors and electronic components. These sensors sense the internal environment of the car and transmit the sensed data to a processor. The on-road deployed sensors sense the external environment and transmit the sensed data to the centralized processor. Thereafter, based on requirements, the processor delivers these sensed data to fog or cloud to perform necessary functions. These processes seem to be simple, but practically, several components, along with their challenges, are involved in a vehicular IoT system. Figure 5.2 depicts the components required for vehicular IoT systems.

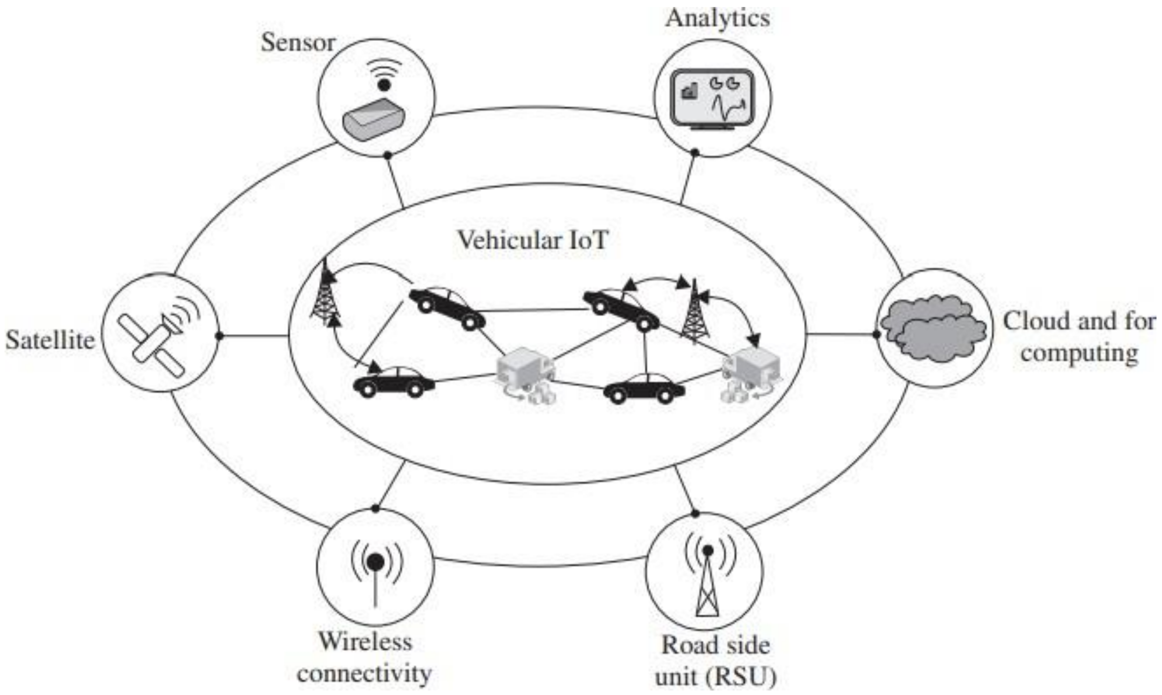


Figure 5.2 Components of vehicular IoT

Sensors: In vehicular IoT, sensors monitor different environmental conditions and help to make the system more economical, efficient, and robust. Traditionally, two types of sensors, internal and external, are used in vehicular IoT systems.

a. **Internal:** These types of sensors are placed within the vehicle. The sensors are typically used to sense parameters that are directly associated with the vehicle. Along with the sensors, the vehicles are equipped with different electronic components such as processing boards and actuators. The internal sensors in a vehicle are connected with the processor board, to which they transmit the sensed data. Further, the sensed data are processed by the board to take certain predefined actions. A few examples of internal sensors are GPS, fuel gauge, ultrasonic sensors, proximity sensors, accelerometer, pressure sensors, and temperature sensors.

b. **External:** External sensors quantify information of the environment outside the vehicle. For example, there are sensors used in the smart traffic system that are capable of sensing vacant parking lots in a designated parking area. The still images and videos from cameras are important inputs to generate decisions in a vehicular IoT system. Therefore, on-road cameras are widely used as external sensors to capture still images and videos. The captured images and videos are processed further, either in the fog or in the cloud layer, to take certain pre-programmed actions. As an example, camera sensor can capture the image of the license plate of an overspeeding vehicle at a traffic signal; the image can be processed to identify the owner of the vehicle to charge a certain amount of fine. Similarly, temperature, rainfall, and light sensors are also used in the vehicular IoT infrastructure.

Satellites: In vehicular IoT systems, automatic vehicle tracking and crash detection are among the important available features. Satellites help the system to track vehicles and detect on-road crashes. The satellite image is also useful for detecting on-road congestions and road blocks.

Wireless connectivity: As vehicular IoT deals with connected vehicles, communication is an important enabling component. For taking any action or making decisions, the collective data from internal and external sensors need processing. For transmitting the sensed data from multiple sensors to RSU (roadside unit) and from RSUs to the cloud, connectivity plays an indispensable role. In the vehicular IoT scenario, the high mobility of the vehicles necessitates the connectivity type to be wireless for practical and real-time data transmission. Different communication technologies, such as Wi-Fi, Bluetooth, and GSM, are common in the vehicular IoT systems.

Road Side Unit (RSU): The RSU is a static entity that works collaboratively with internal and external sensors. Typically, the RSUs are equipped with sensors, communication units, and fog devices. Vehicular IoT systems deal with time critical applications, which need to take decisions in real time. In such a situation, the fog devices attached to the RSUs process the sensed data and take necessary action promptly. If a vehicular system involves heavy computation, the RSU transmits the sensed data to the cloud end. Sometimes, these RSUs also work as an intermediate communication agent between two vehicles.

Cloud and fog computing: In vehicular IoT systems, fog computing handles the light-weight processes geographically closer to the vehicles than the cloud. Consequently, for faster decision making, fog computing is used in vehicular IoT systems. However, for a heavy-weight process, fog computing may not be a suitable option. In such a situation, cloud computing is more adept for vehicular IoT systems. Cloud computing provides more scalability of resources as compared to fog computing. Therefore, the choice of the application of fog and cloud computing depends on the situation. For example, the location and extent of short on-road congestion from a certain location can be determined by fog computing with the help of sensed data.

The congestion information can be shared by the RSU among other onroad vehicles, thereby suggesting that they avoid the congested road. On the other hand, for determining regular on-road congestion, predictions are typically handled with the help of cloud computing. For the regular congestion prediction, the cloud end needs to process a huge amount of instantaneous data, as well as, historical data for that stretch of road spanning back a few months to years.

Analytics: Similar to different IoT application domains, in vehicular IoT, analytics is a crucial component. Vehicular IoT systems can be made to predict different dynamic and static conditions using analytics. For example, strong data analytics is required to predict on-road traffic conditions that may occur at a location after an hour.

Key Points:

- The sensors attached to the different parts of a vehicle, such as the battery and fuel pump, transmit the data to the cloud for analyzing the requirements for the maintenance of those parts.
- The evolution of IoT enables a user to lock, unlock, locate their cars, even from a remote location.

5.1.2 Advantages of vehicular IoT

The evolution of IoT resulted in the development of a connected vehicular environment. The typical advantages of IoT architectures directly impact the domain of connected vehicular systems. Therefore, the advantages of IoT are inherently included in vehicular IoT environments. A few selected advantages of vehicular IoT are depicted in Figure 5.3.

(i) Easy tracking: The tracking of vehicles is an essential part of vehicular IoT. Moreover, the system must know from which location and which vehicle the system is receiving the information. In a vehicular IoT system, the tracking of vehicles is straightforward; the system can collect information at a remote location.

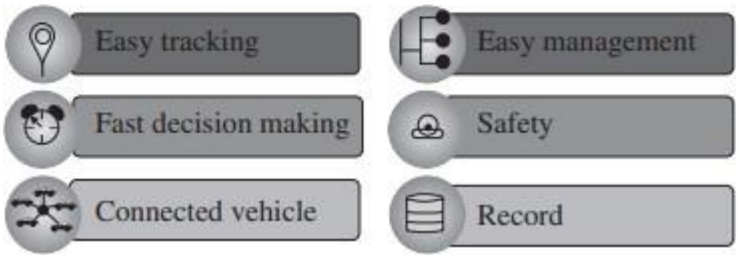


Fig 5.3 : Advantages of vehicular IoT

ii) **Fast decision making:** Most of the decisions in the connected vehicle environment are time critical. Therefore, for such an application, fast and active decision making are pertinent for avoiding accidents. In the vehicular IoT environment, cloud and fog computing help to make fast decisions with the data received from the sensor-based devices.

iii) **Connected vehicles:** A vehicular IoT system provides an opportunity to remain connected and share information among different vehicles.

iv) **Easy management:** Since vehicular IoT systems consist of different types of sensors, a communication unit, processing devices, and GPS, the management of the vehicle becomes easy. The connectivity among different components in a vehicular IoT enables systems to track every activity in and around the vehicle. Further, the IoT infrastructure helps in managing the huge number of users located at different geographical coordinates.

v) **Safety:** Safety is one of the most important advantages of a vehicular IoT system. With easy management of the system, both the internal and external sensors placed at different locations play an important role in providing safety to the vehicle, its occupants, as well as the people around it.

vi) **Record:** Storing different data related to the transportation system is an essential component of a vehicular IoT. The record may be of any form, such as video footage, still images, and documentation. By taking advantage of cloud and fog computing architecture, the vehicular IoT systems keep all the required records in its database.

5.1.3 Crime assistance in a smart IoT transportation system

In this section, we discuss a case study on smart safety in a vehicular IoT infrastructure. The system highlights a fog framework for intelligent public safety in vehicular environments (fog-FISVER). The primary aim of this system is to ensure smart transportation safety (STS) in public bus services. The system works through the following three steps:

(i) The vehicle is equipped with a smart surveillance system, which is capable of executing video processing and detecting criminal activity in real time.

(ii) A fog computing architecture works as the mediator between a vehicle and a police vehicle.

(iii) A mobile application is used to report the crime to a nearby police agent. Architecture The architecture of the fog-FISVER consists of different IoT components. The developers utilized the advantages of the low-latency fog computing architecture for designing their system. Fog-FISVER is based on a three-tiered architecture, as shown in Figure 5.4. Each of the tiers are as follows:

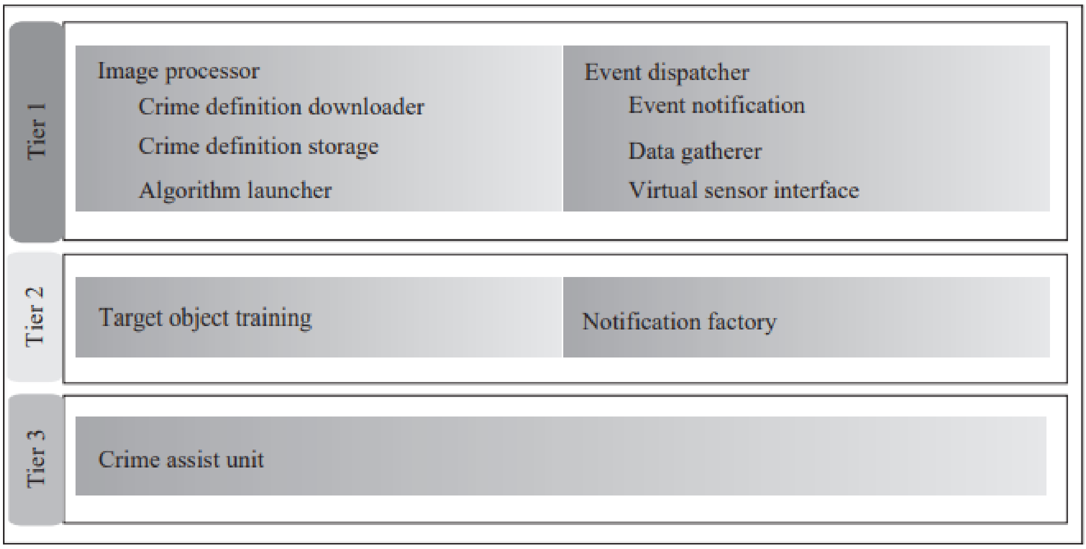


Figure 5.4 Architecture of Fog-FISVER

i) **Tier1—In-vehicle FISVER STS Fog:** In this system component, a fog node is placed for detecting criminal activities. This tier accumulates the real sensed data from within the vehicle and processes it to detect possible criminal activities inside the vehicle. Further, this tier is responsible for creating crime-level metadata and transferring the required information to the next tier. For performing all the activities, Tier 1 consists of two subsystems: Image processor and event dispatcher

Image Processor: The image processor inside Tier 1 is a potent component, which has a capability similar to the human eye for detecting criminal activities. Developers of the system used a deep-learning-based approach for enabling image processing techniques in the processor. To implement the fog computing architecture in the vehicle, a Raspberry-Pi-3 processor board is used, which is equipped with a high-quality camera. Further, this

architecture uses template matching and correlation to detect the presence of dangerous articles (such as a pistol or a knife) in the sub-image of a video frame.
Typically, the image processor stores a set of crime object templates in the fog-FISVER STS fog infrastructure, which is present in Tier 2 of the system. The image processor is divided into the following three parts:

(a) Crime definition downloader: This component periodically checks for the presence of new crime object template definitions in fog-FISVER STS fog infrastructure. If a new crime object template is available, it is stored locally.

(b) Crime definition storage: In order to use template matching, the crime object template definition is required to be stored in the system. The crime definition storage is used to store all the possible crime object template definitions.

(c) Algorithm launcher: This component initiates the instances of the registered algorithm in order to match the template with the video captured by the camera attached in the vehicles. If a crime object is matched with the video, criminal activity is confirmed.

Event dispatcher: This is another key component of Tier 1. The event dispatcher is responsible for accumulating the data sensed from vehicles and the image processor. After the successful detection of criminal activity, the information is sent to the fog-FISVER STS fog infrastructure. The components of the event dispatcher are as follows:

(a) Event notifier: It transfers the data to the fog-FISVER STS fog infrastructure, after receiving it from the attached sensor nodes in the vehicle.

(b) Data gatherer: This is an intermediate component between the event notifier and the physical sensor; it helps to gather sensed data.

(c) Virtual sensor interface: Multiple sensors that sense data from different locations of the vehicle are present in the system. The virtual sensor interface helps to maintain a particular procedure to gather data. This component also cooperates to register the sensors in the system.

ii) **Tier 2—FISVER STS Fog Infrastructure:** Tier 2 works on top of the fog architecture. Primarily, this tier has three responsibilities — keep updating the new object template definitions, classifying events, and finding the most suitable police vehicle to notify the event. FISVER STS fog infrastructure is divided into two sub-components:

Target Object Training: Practically, there are different types of crime objects. The system needs to be up-to-dated regarding all crime objects. This subcomponent of Tier 2 is responsible for creating, updating, and storing the crime object definition. The algorithm launcher uses these definitions in Tier 1 for the template matching process. The template

definition includes different features of the crime object such as color gradient and shape format.
A new object definition is stored in the definition database. The database requires to be updated based on the availability of new template definitions.

Notification Factory: This sub-component receives notification about the events in a different vehicle with the installed system. Further, this component receives and validates the events. In order to handle multiple events, it maintains a queue.

iii) Tier 3 consists of mobile applications that are executed on the users' devices. The application helps a user, who witnesses a crime, to notify the police.

II) Healthcare IoT

5.2 Introduction

Internet of Things (IoT) has resulted in the development and emergence of a variety of technologies that has had a huge impact on the medical field, especially wearable healthcare. The salient features of IoT encourage researchers and industries to develop new IoT-based technologies for healthcare. These technologies have given rise to small, power-efficient, health monitoring and diagnostic systems. Consequently, the development of numerous healthcare technologies and systems has rapidly increased over the last few years. Currently, various IoT-enabled healthcare devices are in wide use around the globe for diagnosing human diseases, monitoring human health conditions, caring/monitoring for elders, children, and even infants. Moreover, IoT-based healthcare systems and services help to increase the quality of life for common human beings; in fact, it has a promising scope of revolutionizing healthcare in developing nations.

IoT-based healthcare devices provide access and knowledge about human physiological conditions through hand held devices. With this development, users can be aware of the risks in acquiring various diseases and take necessary precautions to avoid preventable diseases. The basic skeleton of an IoT-based healthcare system is very similar to the conventional IoT architectures. However, for IoT-based healthcare services, the sensors are specifically designed to measure and quantify different physiological conditions of its users/patients. A typical architecture for healthcare IoT is shown in Figure 5.5. We divide the architecture into four layers.
The detailed description of these layers are as follows:

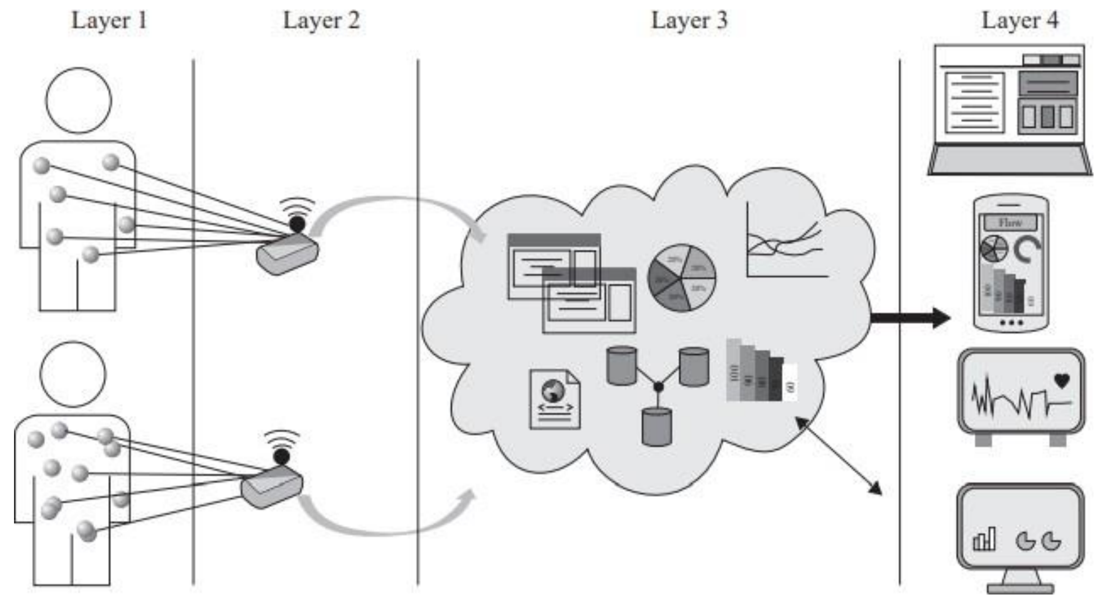


Figure 5.5 Architecture of healthcare IoT

Layer 1: Layer 1 contains different physiological sensors that are placed on the human body. These sensors collect the values of various physiological parameters. The physiological data are analyzed to extract meaningful information.

Layer 2: Layer 1 delivers data to Layer 2 for short-term storage and low-level processing. The devices that belong to Layer 2 are commonly known as local processing units (LPU) or centralized hubs. These units collect the sensed data from the physiological sensors attached to the body and process it based on the architecture's requirement. Further, LPUs or the centralized hubs forward the data to Layer 3.

Layer 3: This layer receives the data from Layer 2 and performs application specific high-level analytics. Typically, this layer consists of cloud architecture or high-end servers. The data from multiple patients, which may be from the same or different locations, are accumulated in this layer. Post analysis of data, some inferences or results are provided to the application in Layer 4.

Layer 4: The end-users directly interact with Layer 4 through receiver-side applications. The modes of accessibility of these services by an end user are typically through cellphones, computers, and tablets.

5.2.1 Components of healthcare IoT

A typical IoT healthcare architecture is composed of several components that are essential to generate the whole architecture. Figure 5.6 depicts different components and their usage in an IoT healthcare system. Each of these components plays a distinct role in the smooth execution of the system as a whole. In this section, we discuss the different components for a basic healthcare IoT system.

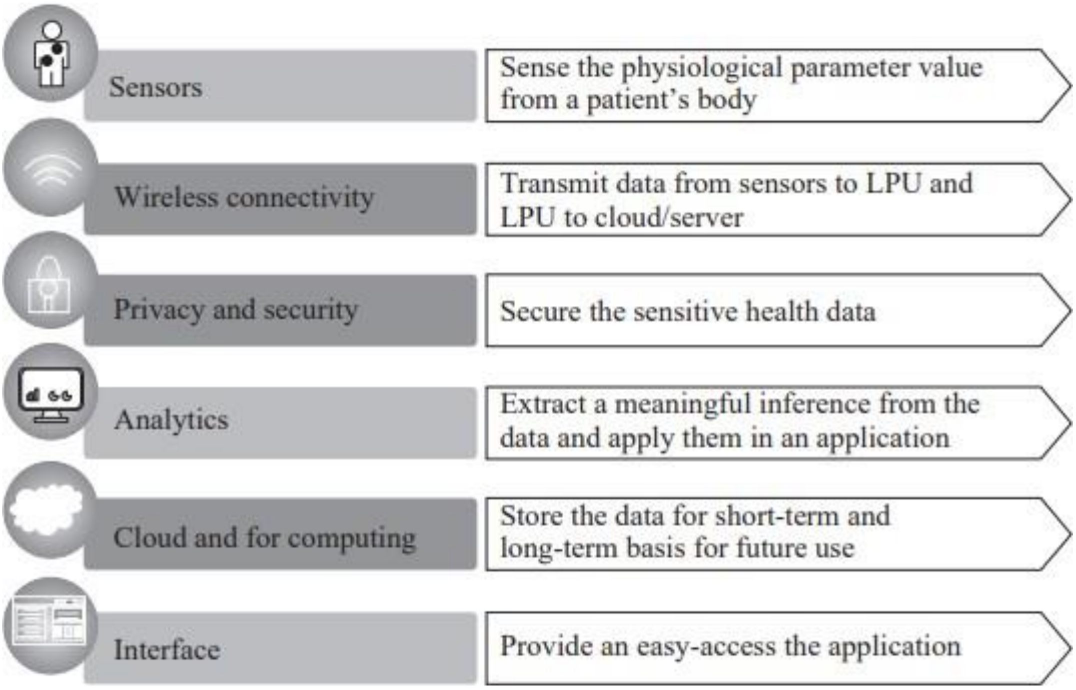
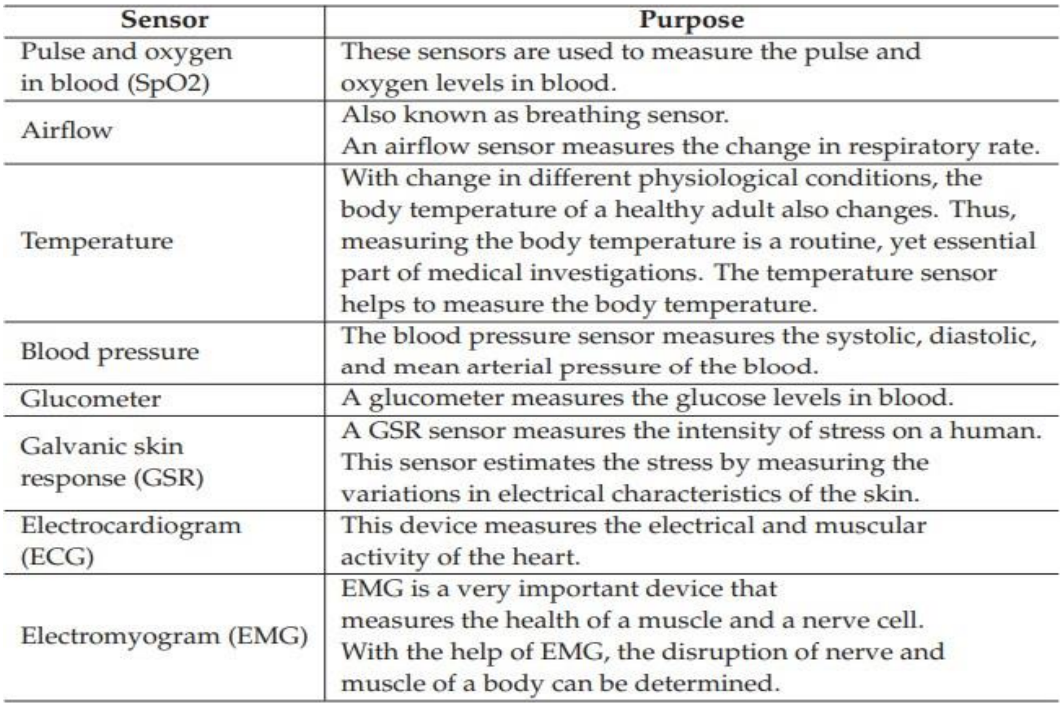

Sensors
Sense the physiological parameter value from a patient's body
Wireless connectivity
Transmit data from sensors to LPU and LPU to cloud/server
Privacy and security
Secure the sensitive health data
Analytics
Extract a meaningful inference from the data and apply them in an application
Cloud and for computing
Store the data for short-term and long-term basis for future use
Interface
Provide an easy-access the application

Figure 5.6 Components of healthcare IoT

Sensors: Layer 1 consists of physiological sensors that collect the physiological parameters of the patient. Few commonly used physiological sensors and their uses are depicted in Table 5.1.

Table 14.1 Commonly used healthcare sensors



Sensor	Purpose
Pulse and oxygen in blood (SpO2)	These sensors are used to measure the pulse and oxygen levels in blood.
Airflow	Also known as breathing sensor. An airflow sensor measures the change in respiratory rate.
Temperature	With change in different physiological conditions, the body temperature of a healthy adult also changes. Thus, measuring the body temperature is a routine, yet essential part of medical investigations. The temperature sensor helps to measure the body temperature.
Blood pressure	The blood pressure sensor measures the systolic, diastolic, and mean arterial pressure of the blood.
Glucometer	A glucometer measures the glucose levels in blood.
Galvanic skin response (GSR)	A GSR sensor measures the intensity of stress on a human. This sensor estimates the stress by measuring the variations in electrical characteristics of the skin.
Electrocardiogram (ECG)	This device measures the electrical and muscular activity of the heart.
Electromyogram (EMG)	EMG is a very important device that measures the health of a muscle and a nerve cell. With the help of EMG, the disruption of nerve and muscle of a body can be determined.

Wireless Connectivity: Without proper connectivity and communication, the data sensed by the physiological sensors are of no use in an IoT-based healthcare system. Typically, the communication between the wearable sensors and the LPU is through either wired or wireless connectivity. The wireless communication between the physiological sensors and LPU occurs with the help of Bluetooth and ZigBee. On the other hand, the communication between the LPU and the cloud or server takes place with Internet connectivity such as WiFi and WLAN.

In Layer 4 of the healthcare IoT architecture, the healthcare data are received by the end users with different devices such as laptops, desktops, and cellphones. These communication protocols vary depending on the type of device in use. For example, when a service is received by a cellphone, it uses GSM (global system for mobile communications). On the other hand, if the same service is received on a desktop, it can be through Ethernet or Wi-Fi. Communication and connectivity in healthcare IoT is an essential component.

Privacy and Security: The privacy and security of health data is a major concern in healthcare IoT services. In a healthcare IoT architecture, several devices connect with the external world. Moreover, between LPU and the server/cloud, different networking devices work via network hops (from one networked device to another) to transmit the data. If any of these devices are compromised, it may result in the theft of health data of a patient, leading to serious security breaches and ensuing lawsuits. In order to increase the security of the healthcare data, different healthcare service providers and organizations are implementing healthcare data encryption and protection schemes.

Analytics: For converting the raw data into information, analytics plays an important role in healthcare IoT. Several actors, such as doctors, nurses, and patients, access the healthcare information in a different customized format. This customization allows each actor in the system to access only the information pertinent to their job/role. In such a scenario, analytics plays a vital role in providing different actors in the system access to meaningful information extracted from the raw healthcare data. Analytics is also used for diagnosing a disease from the raw physiological data available.

Cloud and Fog Computing: In a healthcare IoT system, several physiological sensors are attached to a patient’s body. These sensors continuously produce a huge amount of heterogeneous data. For storing these huge amounts of heterogeneous health data, efficient storage space is essential. These data are used for checking the patient’s history, current health status, and future for diagnosing different diseases and the symptoms of the patient. Typically, the cloud storage space is scalable, where payment is made as per the usage of space. Consequently, to store health data in a healthcare IoT system, cloud storage space is used. Analytics on the stored data in cloud storage space is used for drawing various inferences. The major challenges in storage are security and delay in accessing the data. Therefore, cloud and fog computing play a pivotal role in the storage of these massive volumes of heterogeneous data.

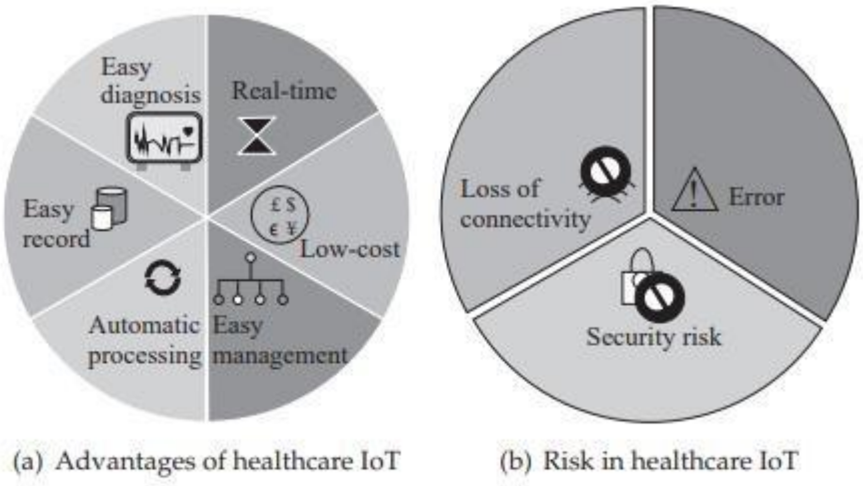
Interface: The interface is the most important component for users in a healthcare IoT system. Among IoT applications, healthcare IoT is a very crucial and sensitive application. Thus, the user interface must be designed in such a way that it can depict all the required information clearly and, if necessary, reformat or represent it such that it is easy to understand. Moreover, an interface must also contain all the useful information related to the services.

Key points:

- As healthcare data is private, a popular US legislation - Health Insurance Portability and Accountability (HIPAA) - protects through data privacy and security provisions.
- Drones are used to deliver medicines in disaster rescue and management scenarios.

5.2.2 Advantages and risk of healthcare IoT

IoT has already started to penetrate the domain of medical science. In healthcare, IoT has become significantly popular due to its various features, which have been covered previously in this book. Healthcare IoT helps in managing different healthcare subsystems efficiently. Although it has many advantages, healthcare IoT has some risks too, which may be crucial in real-life applications. In this section, we discuss the different advantages and risks of healthcare IoT as depicted in Figure 5.7.



(a) Advantages of healthcare IoT (b) Risk in healthcare IoT

Figure 5.7 Advantages and risk in healthcare IoT

Advantages of healthcare IoT The major advantages of healthcare IoT can be listed as follows:

Real-time: In healthcare sectors, different components, such as the condition of the patients, availability of doctors and beds in a hospital, medical facilities with their monetary charges, can vary dynamically with time. In such a dynamic scenario, one of the important characteristics of an IoT-based healthcare system is real-timeliness.

A healthcare IoT system enables users, such as doctors, end users at the patient-side, and staff in a healthcare unit, to receive real-time updates about the healthcare IoT components, as mentioned earlier. Moreover, a healthcare IoT system can enable a doctor to observe a patient's health condition in real-time even from a remote location, and can suggest the type of care to be provided to the patient.

On the other hand, users at the patient-end can easily take different decisions, such as where to take a patient during critical situations. Moreover, the staff in a healthcare unit are better aware of the current situation of their unit, which includes the number of patients admitted, availability of the doctors and bed, total revenue of the unit, and other such information.

Low cost: Healthcare IoT systems facilitate users with different services at low cost. For example, an authorized user can easily find the availability of the beds in a hospital with simple Internet connectivity and a web-browser-based portal. The user need not visit the hospital physically to check the availability of beds and facilities. Moreover, multiple registered users can retrieve the same information simultaneously.

Easy management: Healthcare IoT is an infrastructure that brings all its end users under the same umbrella to provide healthcare services. On the other hand, in such an infrastructure, the management of numerous tangible and intangible entities (such as users, medical devices, facilities, costs, and security) is a challenging task. However, healthcare IoT facilitates easy and robust management of all the entities.

Automatic processing: A healthcare unit consists of multiple subsystems, for which manual interventions are required. For example, to register a patient with a hospital, the user may be required to enter his/her details manually. However, automatic processing features can remove such manual intervention with a fingerprint sensor/device. Healthcare IoT enables end-to-end automatic processing in different units and also consolidates the information across the whole chain: from a patient's registration to discharge.

Easy record-keeping: The healthcare IoT system, includes a huge number of patients, doctors, and other staff. Different patients suffer from different types of diseases. A particular disease requires particular treatment, which requires knowledge of a patient's health history, along with other details about them. Therefore, the timely delivery of health data of the patient to the doctor is important. In such a situation, the permanent storage of the patients' health data along with their respective details is essential.

Similarly, for the smooth execution of the healthcare unit, details of the staff with their daily activity in a healthcare unit are also required for storage. A healthcare unit must also track its condition and financial transactions for further development of the unit. A healthcare IoT enables the user to keep these records in a safe environment and deliver them to the authorized user as per requirement. Moreover, these recorded data are accessible from any part of the globe.

Easy diagnosis: The healthcare IoT system stores the data of the patient in a secure manner. Sometimes, for diagnosing a disease, a huge chunk of prior data is required. In a healthcare IoT system, the diagnosis of the disease becomes easier with the help of certain learning mechanisms along with the availability of prior datasets.

Risk in healthcare IoT

In a healthcare IoT system, there are multiple risks as well. Here, we discuss the various risks associated with a healthcare IoT system.

Loss of connectivity: A healthcare IoT system consists of different physiological sensors that sense and transmit the sensed data to a centralized unit. Moreover, continuous data transmission from the patient is expected in a good healthcare system. Intermittent connectivity may result in data loss, which may result in a life-threatening situations for the patient. Proper and continuous connectivity is essential in a healthcare IoT system.

Security: A healthcare IoT system contains the health data of different patients associated with the system. The healthcare system must keep the data confidential. This data should not be accessible to any unauthorized person. On the other hand, different persons and devices are associated with a healthcare IoT system. In such a system, the risk of data tampering and unauthorized access is quite high.

Error: Data analytics helps a healthcare IoT system to predict the patients’ condition and diagnosis of diseases. A huge amount of data needs to be fed into the system in order to perform accurate analytics. Moreover, the management of a huge amount of data is a crucial task in any IoT-based system. Particularly, in the healthcare system, errors in data may lead to misinterpretation of symptoms and lead to the wrong diagnosis of the patient. It is a challenging task to construct an error-free healthcare IoT architecture.

5.3 Case Studies

5.3.1 AmbuSens system

In many developing countries, patients need to be transferred from primary-care to tertiary-care hospitals for proper diagnosis and treatment. During the transit, the hospitals at both ends—the referring one as well as the referred one—do not have any information about the patient’s health condition during transit. In such situations, the hospitals are unable to suggest any precautionary measures in the event of some emergency during transit. Consequently, many patients die during the transit due to lack of proper suggestive care by medical experts.

To overcome these shortcomings, the Smart Wireless Applications and Networking (SWAN) laboratory at the Indian Institute of Technology Kharagpur developed a system: AmbuSens. The system was primarily funded by the Ministry of Human Resource and Development (MHRD) of the Government of India. This product system is a very crucial part of the healthcare IoT system.

The primary objectives of the AmbuSens system are summarized as follows:

- Digitization and standardization of the healthcare data, which can be easily accessed by the registered hospital authorities.

- Real-time monitoring of the patients who are in transit from one hospital to another. At both hospitals, doctors can access the patients' health conditions.

- Accessibility by which multiple doctors can access the patient's health data at the same time.

- Provision of confidentiality to the health data of the patients in the cloud.

- In the AmbuSens system, wireless physiological sensor nodes are used. These sensor nodes make the system flexible and easy to use.

Architecture

The AmbuSens system is equipped with different physiological sensors along with a local hub. These sensors sense the physiological parameters from the patient's body and transmit those to a local data processing unit (LDPU). The physiological sensors and LDPU form a wireless body area network (WBAN). Further, this local hub forwards the physiological data to the cloud for storing and analyzing the health parameters. Finally, the data are accessed by different users. The detailed layered architecture of the AmbuSens system is depicted in Figure 5.8.

Layer 1: This layer consists of multiple WBANs attached to a patient's body. These WBANs acquire the physiological data from the patient and transmit them to the upper layer. The physiological sensors are heterogeneous, that is, each of these sensors senses different parameters of the body. Moreover, the physiological sensors require calibration for acquiring the correct data from a patient's body. Layer 1 takes care of the calibration of the physiological sensor nodes. Further, in order to deliver the patient's physiological data from the sensor node to the LDPU, it is essential to form a proper WBAN. The formation of WBAN takes place by connecting multiple physiological sensor nodes to the LDPU so that the sensors can transmit the data to the LDPU, simultaneously.

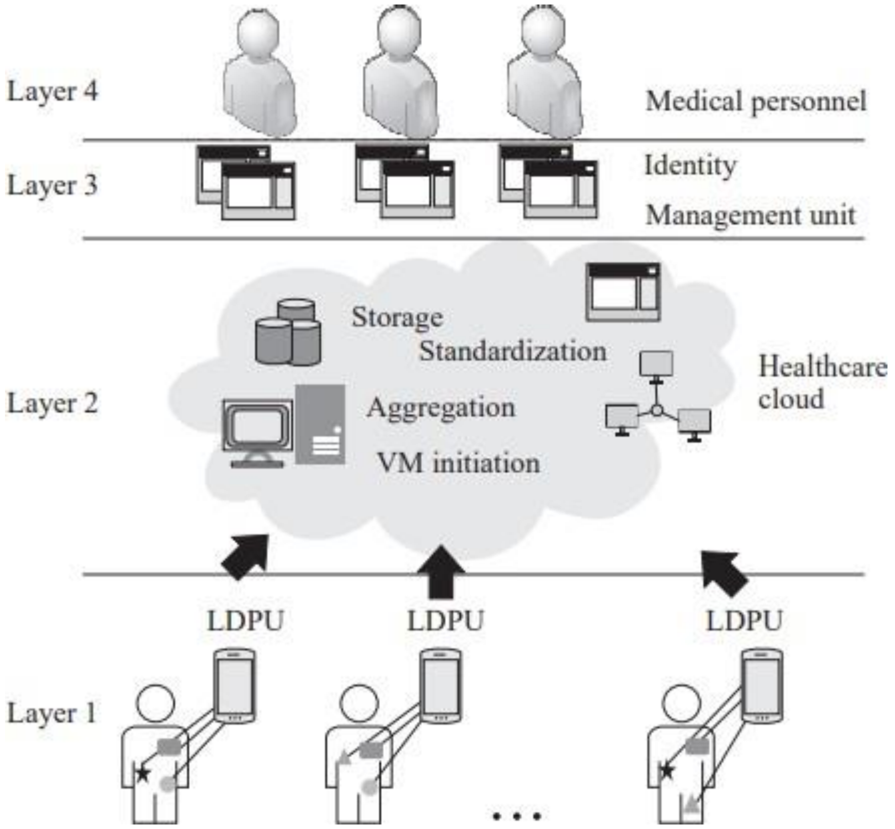


Figure 5.8 Layered architecture of AmbuSens

Layer 2: In the AmbuSens system, cloud computing has an important role. Layer 2 is responsible for handling the cloud-related functions. From Layer 1, WBANs attached to the different patients deliver data to the cloud end. The cloud is used for the long-term analysis and storage of data in the AmbuSens system. Moreover, the previous health records of the patients are stored in the cloud in order to perform patient-specific analysis. A huge volume of health data is produced by the WBANs, which are handled by the cloud with the help of big data analytics for providing real-time analysis.

Layer 3: In the AmbuSens system, the identity of the patients remains anonymous. An algorithm is designed to generate a dynamic hash value for each patient in order to keep the patient's identity anonymous. Moreover, in the AmbuSens system, at different time instants, a new hash value is generated for the patients. The entire hashing mechanism of the AmbuSens is performed in this layer.

Layer 4: The users simply register into the system and use it as per requirement.

Hardware

In the AmbuSens system, a variety of hardware components are used such as sensors, communication units, and other computing devices.

Sensors: The sensors used in the AmbuSens system are non-invasive. The description of the sensors used for forming the WBAN in the AmbuSens system are as follows:

Optical Pulse Sensing Probe: It senses the photoplethysmogram (PPG) signal and transmits it to a GSR expansion module. Typically, PPG signals are sensed from the ear lobe, fingers, or other location of the human body. Further, the GSR expansion module transfers the sensed data to a device in real-time.

Electrocardiogram (ECG) unit and sensor: The ECG module used in AmbuSens is in the form of a kit, which contains ECG electrodes, biophysical 9” leads, biophysical 18” leads, alcohol swabs, and wrist strap. Typically, the ECG sensor measures the pathway of electrical impulses through the heart to sense the heart’s responses to physical exertion and other factors affecting cardiac health.

Electromyogram (EMG) sensor: This sensor is used to analyze and measure the biomechanics of the human body. Particularly, the EMG sensor is used to measure different electrical activity related to muscle contractions; it also assesses nerve conduction, and muscle response in injured tissue.

Temperature sensor: The body temperature of patients changes with the condition of the body. Therefore, a temperature sensor is included in the AmbuSens system, which can easily be placed on the body of the patient.

Galvanic Skin Response (GSR) sensor: The GSR sensor is used for measuring the change in electrical characteristics of the skin.

Local Data Processing Unit (LDPU): In AmbuSens, all the sensors attached to the human body sense and transmit the sensed data to a centralized device, which is called an LDPU. An LDPU is a small processing board with limited computation capabilities. The connectivity between the sensors and the LDPU follows a single-hop star topology. The LDPU is programmed in such a way that it can receive the physiological data from multiple sensor nodes, simultaneously. Further, it transmits the data to the cloud for long-term storage and heavy processing.

Communication Module: Each sensor node consists of a Bluetooth (IEEE 802.15.1 standard) module. The communication between the sensor nodes and the LDPU takes place with the help of Bluetooth, which supports a maximum communication range of 10 meters in line-of-sight. The LDPU delivers the data to the cloud with 3G/4G communication.

Front End

In the AmbuSens system, three actors—doctor, paramedic/nurse, and patient—are able to participate and use the services. The web interface is designed as per the requirements of the actors of the system. Each of the actors has an option to log in and access the system. The confidentiality of a patient and their physiological data is important in a healthcare system. Therefore, the system provides different scopes for data accessibility based on the category of an actor. For example, the detailed health data of a patient is accessible only to the assigned doctor.

These data may not be required for the nurse; therefore, a nurse is unable to access the same set of data a doctor can access. The system provides the flexibility to a patient to log in to his/her account and download the details of his/her previous medical/treatment details. Therefore, in AmbuSens, the database is designed in an efficient way such that it can deliver the customized data to the respective actor. Each of the users has to register with the system to avail of the service of the AmbuSens.

Therefore, in this system, the registration process is also designed in a customized fashion, that is, the details of a user to be entered into the registration form is different for different actors. For example, a doctor must enter his/her registration number in the registration form.

III) IoT Analytics

5.4 Introduction

The sensors collect data from the environment and serve different IoT-based applications. The raw data from a sensor require processing to draw inferences. An IoT based system generates data with complex structures; therefore, conventional data processing on these data is not sufficient. Sophisticated data analytics are necessary to identify hidden patterns. In this chapter, we discuss a few traditional data analytics tools that are popular in the context of IoT applications. These tools include k-means, decision tree (DT), random forest (RF), k-nearest neighbor (KNN), and density-based spatial clustering of applications with noise (DBSCAN) algorithms.

5.4.1 Machine learning

The term "machine learning" was coined by Arthur Lee Samuel, in 1959. He defined machine learning as a "field of study that gives computers the ability to learn without being explicitly programmed". ML is a powerful tool that allows a computer to learn from past experiences and its mistakes and improve itself without user intervention. Typically, researchers envision IoT-based systems to be autonomous and self-adaptive, which enhNHHances services and user experience. To this end, different ML models play a crucial role in designing intelligent systems in IoT by leveraging the massive amount of generated data and increasing the accuracy in their operations. The main components of ML are statistics, mathematics, and computer science for drawing inferences, constructing ML models, and implementation, respectively

Key Points:

- ML is an important tool, which is used by different social networking websites such as facebook and twitter.
- Autonomous vehicles use ML to determine their paths and speeds.

5.4.2 Advantages of ML

Applications fueled by ML open a plethora of opportunities in IoT-based systems, from triggering actuators to identifying chronic diseases from images of an eye. ML also enables a system to identify changes and to take intelligent actions that relatively imitates that of a human. As ML demonstrates a myriad of advantages, its popularity in IoT applications is increasing rapidly. In this section, we discuss the different advantages of ML, as depicted in Figure 5.9.

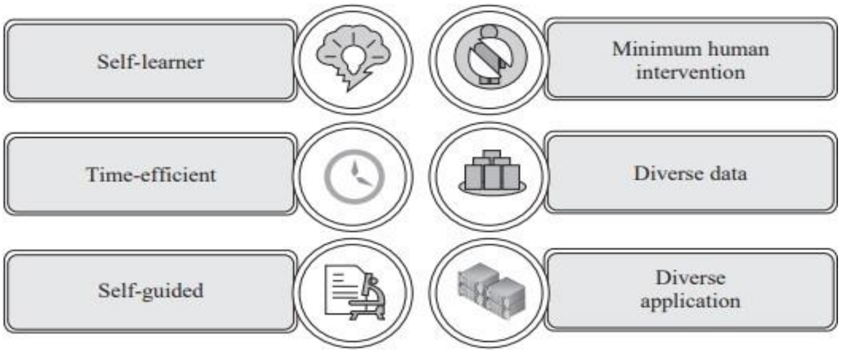


Figure 5.9 Advantages of ML

Self-learner: An ML-empowered system is capable of learning from its prior and run-time experiences, which helps in improving its performance continuously. For example, an ML-assisted weather monitoring system predicts the weather report of the next seven days with high accuracy from data collected in the last six months. The system offers even better accuracy when it analyzes weather data that extends back to three more months.

Time-efficient: ML tools are capable of producing faster results as compared to human interpretation. For example, the weather monitoring system generates a weather prediction report for the upcoming seven days, using data that goes back to 6–9 months. A manual analysis of such sizeable data for predicting the weather is difficult and time-consuming. Moreover, the manual process of data analysis also affects accuracy. In such a situation, ML is beneficial in predicting the weather with less delay and accuracy as compared to humans.

Self-guided: An ML tool uses a huge amount of data for producing its results. These tools have the capability of analyzing the huge amount of data for identifying trends autonomously. As an example, when we search for a particular item on an online e-commerce website, an ML tool analyzes our search trends. As a result, it shows a range of products similar to the original item that we searched for initially.

Minimum Human Interaction Required: In an ML algorithm, the human does not need to participate in every step of its execution. The ML algorithm trains itself automatically, based on available data inputs. For instance, let us consider a healthcare system that predicts diseases. In traditional systems, humans need to determine the disease by analyzing different symptoms using standard "if– else" observations. However, the ML algorithm determines the same disease, based on the health data available in the system and matching the same with the symptoms of the patient.

Diverse Data Handling: Typically, IoT systems consist of different sensors and produce diverse and multi-dimensional data, which are easily analyzed by ML algorithms. For example, consider the profit of an industry in a financial year. Profits in such industries depend on the attendance of laborers, consumption of raw materials, and performance of heavy machineries. The attendance of laborers is associated with an RFID (radio frequency identification)-based system. On the other hand, industrial sensors help in the detection of machiney failures, and a scanner helps in tracking the consumption of raw materials. ML algorithms use these diverse and multi-dimensional data to determine the profit of the industry in the financial year.

Diverse Applications: ML is flexible and can be applied to different application domains such as healthcare, industry, smart traffic, smart home, and many others. Two similar ML algorithms may serve two different applications.

5.4.3 Challenges in ML

An ML algorithm utilizes a model and its corresponding input data to produce an output. A few major challenges in ML are listed as follows:

Data Description: The data acquired from different sensors are required to be informative and meaningful. Description of data is a challenging part of ML.

Amount of Data: In order to provide an accurate output, a model must have sufficient amount of data. The availability of a huge amount of data is a challenge in ML.

Erroneous Data: A dataset may contain noisy or erroneous data. On the other hand, the learning of a model is heavily dependent on the quality of data. Since erroneous data misleads the ML model, its identification is crucial.

Selection of Model: Multiple models may be suitable for serving a particular purpose. However, one model may perform better than others. In such cases, the proper selection of the model is pertinent for ML.

Quality of Model: After the selection of a model, it is difficult to determine the quality of the selected model. However, the quality of the model is essential in an ML-based system.

5.4.4 Types of ML

Typically, ML algorithms consist of four categories:

(i) Supervised (ii) Unsupervised (iii) Semi-supervised (iv) Reinforcement Learning (Figure 5.10).

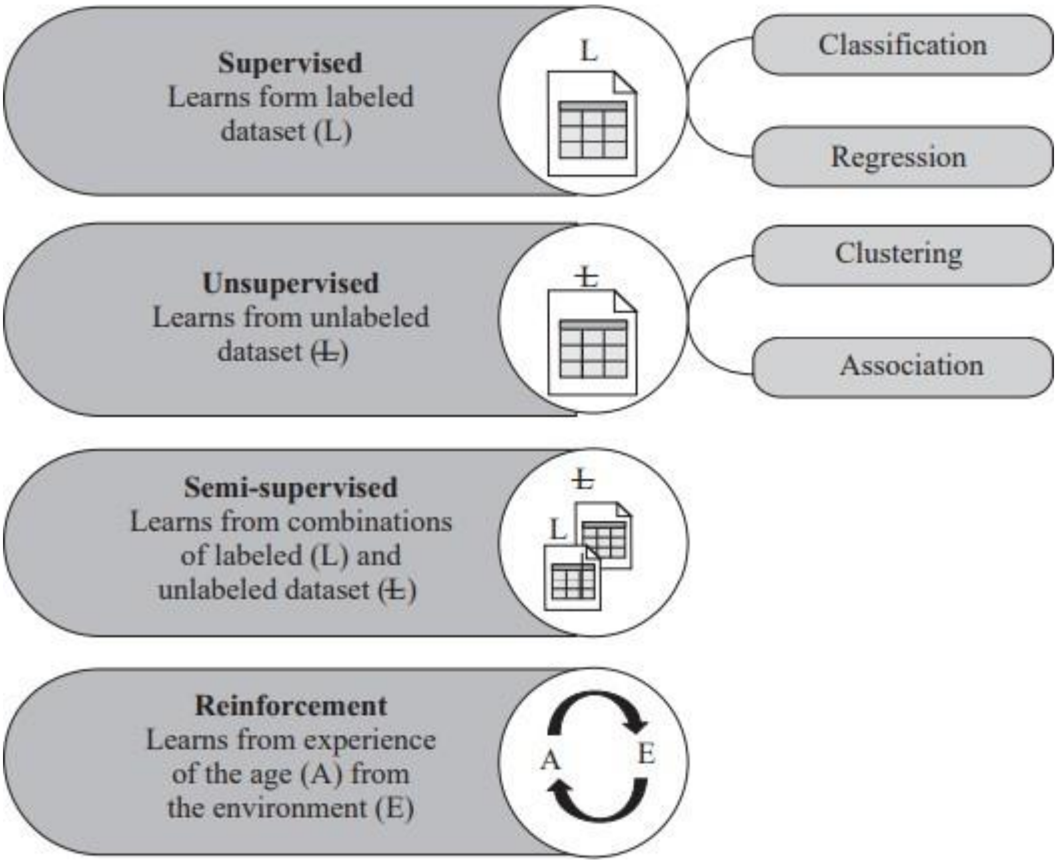

Supervised
Learns form labeled dataset (L)
L
Classification
Regression
Unsupervised
Learns from unlabeled dataset (Ł)
Ł
Clustering
Association
Semi-supervised
Learns from combinations of labeled (L) and unlabeled dataset (Ł)
Ł
L
Reinforcement
Learns from experience of the age (A) from the environment (E)
A
E

Figure 5.1 Types of ML

The different categories of ML are labeled- and unlabeled-data. As the name suggests, labeled data contain certain meaningful tags, known as labels. Typically, the labels correspond to the characteristics or properties of the objects. For example, in a dataset containing the images of two birds, a particular sample is tagged as a crow or a pigeon. On the other hand, the unlabeled dataset does not have any tags associated with them. For example, a dataset containing the images of a bird without mentioning its name.

Supervised Learning: This type of learning supervises or directs a machine to learn certain activities using labeled datasets. The labeled data are used as a supervisor to make the machine understand the relation of the labels with the properties of the corresponding input data. Consider an example of a student who tries to learn to solve equations using a set of labeled formulas. The labels indicate the formulae necessary for solving an equation. The student learns to solve the equation using suitable formulae from the set. In the case of a new equation, the student tries to identify the set of formulae necessary for solving it. Similarly, ML algorithms train themselves for selecting efficient formulae for solving equations.

The selection of these formulae depends primarily on the nature of the equations to be solved. Supervised ML algorithms are popular in solving classification and regression problems. Typically, the classification deals with predictive models that are capable of approximating a mapping function from input data to categorical output. On the other hand, regression provides the mapping function from input data to numerical output. There are different classification algorithms in ML. However, in this chapter, we discuss three popular classification algorithms: (i) k-nearest neighbor (KNN), (ii) decision tree (DT), and (iii) random forest (RF).

We use regression to estimate the relationship among a set of dependent variables with independent variables, as shown in Figure 5.11. The dependent variables are the primary factors that we want to predict. However, these dependent variables are affected by the independent variables. Let x and y be the independent and dependent variables, respectively. Mathematically, a simple regression model is represented as:

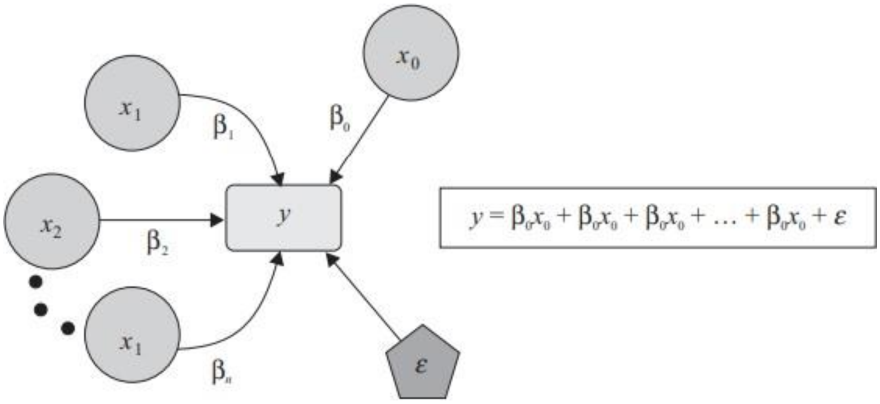

x_1
β_1
β_0
x_0
x_2
y
β_2
x_1
β_n
ε
$y = \beta_0 x_0 + \beta_0 x_0 + \beta_0 x_0 + \ldots + \beta_0 x_0 + \varepsilon$

Figure 5.11 Regression model

y = β0 x0 + βx + (17.1) where β represents the amount of impact of variable x on y and denotes an error. In the given equation, x0 creates β0 impact on y, which indicates that the value of y can never be 0. Similarly, for multiple variables, say n, the regression model is represented as:

$$y = \sum_{i=0}^{n} \beta_i x_i + \epsilon$$

Unsupervised Learning: Unsupervised learning algorithms use unlabeled datasets to find scientific trends. Let us consider an example of the student similar to that described in the case of supervised learning, and illustrate how it differs in case of unsupervised learning. As already mentioned, unsupervised learning does not use any labels in its operations. Instead, the ML algorithms in this category try to identify the nature and properties of the input equation and the nature of the formulae responsible for solving it. Unsupervised learning algorithms try to create different clusters based on the features of the formulae and relate it with the input equations. Unsupervised learning is usually applied to solve two types of problems: clustering and association. Clustering divides the data into multiple groups. In contrast, association discovers the relationship or association among the data in a dataset.

Semi-Supervised Learning: Semi-supervised learning belongs to a category between supervised and unsupervised learning. Algorithms under this category use a combination of both labeled and unlabeled datasets for training. Labeled data are typically expensive and are relatively difficult to label correctly. Unlabeled data is less expensive than labeled data. Therefore, semi-supervised learning includes both labeled and unlabeled dataset to design the learning model. Traditionally, semi-supervised learning uses mostly unlabeled data, which makes it efficient to use, and capable of overcoming samples with missing labels.

Reinforcement Learning: Reinforcement learning establishes a pattern with the help of its experiences by interacting with the environment. Consequently, the agent performs a crucial role in reinforcement learning models. It aims to achieve a particular goal in an uncertain environment. Typically, the model starts with an initial state of a problem, for which different solutions are available. Based on the output, the model receives either a reward or a penalty from the environment. The output and reward act as inputs for proceeding to the next state. Thus, reinforcement learning models continue learning iteratively from their experiences while inducing correctness to the output.