

## **PUBLIC TRANSPORTATION OPTIMIZATION**

### **PHASE 5 : Project documentation and submission**

#### **1.INTRODUCTION:**

Internet of Things, also known as IoT, is the term used when devices or objects are embedded with the ability to communicate through the internet. Here, the devices can collect data from the environment and send that data either to cloud or share it with other related devices in the network.

Nowadays, IoT is grabbing a lot of attention, with the terminologies like smart homes and smart cities. Everyday appliances like coffee makers, refrigerators and TVs are already equipped with the capability of connecting to the internet. People have the means to remotely access their homes. For instance, they can check to see if the doors are closed or get notified when there is an intruder. With the help of smart watches or smart bands people can track their sleeping pattern and even count the number of steps they walked in a day. All these devices can communicate wirelessly and send or share their data.

In IoT, devices are equipped with sensors, actuators, processors and hardware for wireless connection. Sensors are used to collect data from the surrounding physical environment. There are different kinds of sensors that have unique properties and are used in different circumstances, for example, temperature sensors, proximity sensors, IR sensors, accelerometer, pressure sensors and light sensors. Sensors are chosen depending on the application of IoT.

While sensors are used to detect any changes in the physical environment, actuators are used to interact with the environment, such as, the data from the temperature sensor is used to automatically control the thermostat. The sizes of IoT devices are small, and due to their size and power they have limited computing capabilities. The data collected from the sensors are, if possible, processed within the sensor or sent to a nearby device, or to a remote server for processing and storage. All IoT devices are equipped with the hardware that allows them to connect wirelessly. The type of communication that occurs in IoT is mainly machine to machine (M2M). There is little human interaction, for example, when installing the devices, giving instructions or accessing the data.

This project is going to talk about IoT and the possible application of IoT to improve the public transportation

1. Define IoT and how it can be used in the public transportation system
2. Defines different communication technologies that are available for IoT. Through these technologies IoT devices can send and receive data.
3. How data is managed in an IoT system.
4. Security issues of an IoT system are discussed

5. The practical part of this project shows how sensors can be used for collecting passenger data in a bus, and how one can remotely access this data.
6. Concludes the findings of this project.

## 2. A BRIEF DESCRIPTION OF INTERNET OF THINGS:

Internet of Things is coined from two words, 'Internet' and 'Things'. The 'Internet' refers to the globally interconnected computer networks that use the TCP/IP protocol to link devices worldwide. The 'Internet' uses different media types to transmit data like ethernet, wireless and optical fibers. On the other hand, the 'Things' are any objects found in the world. It is not only limited to electronic devices, but also to non-electronic devices like furniture, clothes and equipment.

IoT has been getting a lot of attention in the recent years. IoT's applications can be found in various sectors of the society, and each of these sectors define IoT in their own way. In addition, different authors have suggested their own definitions for IoT. Nonetheless, its characteristics can be divided into two main concepts. The first one is that objects can have sensing and communicating capabilities. Secondly, they can communicate through the internet without human interference also known as Machine to Machine (M2M) communication. For that reason, every object that connects to the internet requires a unique identifier .

Internet of Things (IoT) is an idea and paradigm where all the objects or things in the environment can communicate and interact with each other through wired or wireless connections along with unique addressing schemes, to create new applications and reach common goals. IoT enables objects to communicate with other objects, share data with each other, and make their own decisions based on that information. New applications are already developed using IoT, for instance, smart homes, smart cities, where everything is interconnected with one another.

The term 'IoT' was first coined by sensor researcher Kevin Ashton in his presentation in 1999. Since then the term has been widely accepted. The very first IoT enabled environment was created using RFID (Radio-Frequency Identification). RFID is a technology that automatically identifies an object using a tag. In the early stage's RFID was used to manage inventory by wirelessly reading the pre-embedded tags on the items. RFID uses radio-frequency electromagnetic fields i.e. when an object carrying the tag comes near the reader, it will automatically identify that objective.

### 2.1 COMMUNICATION IN INTERNET OF THINGS:

In an IoT system, sensors and actuators need to constantly communicate with one another in order to share information. The connection should be such that they will experience less interference and avoid data loss. The typical wired connection is not a good option for IoT devices, since sensors are sometimes installed in a difficult geographical location. In such cases, wireless connection provides the flexibility, scalability and cost-effective solutions. In addition, IoT devices have limited power supply.

They are built to last for years on a single battery. Due to such limitations IEEE 802.15.4 standard has specified a wireless link for low-power personal area networks (LoWPANs) which is also used by ZigBee devices. ZigBee connects a wide range of devices into one single control network and enhances the IEEE 802.15.4 standard by adding a network layer, security layer and an application framework.

Depending on the application, communication between IoT devices can be with short distances or longer distances. For short distance communication technologies such as Wi-Fi, Bluetooth, ZigBee and 6LoWPAN are used which are part of the Personal Area Network (PAN) and Local Area Network (LAN) class. In long distance communication the range can reach up to few kilometers, in such cases technologies from Wide Area Network (WAN) class are used, such as Cellular 2G/3G/4G or 5G and satellite.

Traditional Wi-Fi protocols are not suitable for sensor communications due to their large energy consumption. In other words, IEEE has a new standard 802.11ah which is a low-power Wi-Fi version that can be used for IoT. The concept of 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks) is to give an IP (Internet Protocol) address to all the small IoT devices. It also defines the frame and mechanisms for transmitting IPv6 packets over the IEEE 802.15.4 network.

In many situations the IoT system uses a gateway that is used to connect to the internet. These gateways or hubs support IP connectivity that allows them to connect to the outside world. IoT devices inside a local network are connected using a non-IP connectivity technology. The reason behind this is that the traditional TCP/IP protocol requires additional processing power, and to process them IoT devices have to consume more power. The life cycle of an IoT device depends upon the amount of power it uses. In such a case, more power equals reduced life cycle of the device. For that reason, sensors are connected to the internet via gateways.

Due to the popularity of IoT, IoT devices are used in different areas for various applications. These IoT devices are heterogeneous in nature, meaning that they have different communication methods, network connectivity and protocols. Unlike devices that are equipped with continuous power supply, communication in IoT devices involves various challenges, since they run solely on battery power. IoT devices are small with low processing power and memory, so they cannot process the typical TCP/IP protocol that all the other devices use in order to communicate through the internet.

Various communication challenges involved in IoT are the following:

- a) Identifying all the devices, as there will be millions of smart devices connecting to the internet. All these devices need a unique address to identify them and to communicate with other devices. This requires a large addressing space that can provide a unique address to every single smart device that is connected to the internet.
- b) Power consumption, since the process of sending and receiving data is a power consuming task, more so in a wireless connection, a solution for communication with low power consumption is required.
- c) Routing protocols that require low memory and for efficient communication pattern.

d) High speed and nonlossy communication.

e) Scalability and mobility.

The above list explains some of the challenges involved in the communication field of IoT that needs to be considered before deploying an IoT system.

## **2.2 Data Management in Internet of Things:**

In IoT, data management deals with collecting, storing and mining of data in order to get a valuable information out of them. With data management real time data can be fused with data that has already been collected and stored, from where new information can be discovered. Data management acts as a layer between devices that generate data and the applications that access these data.

According to Statista Research Department (2020), by the end of 2018 there were around 22 billion IoT devices connected worldwide and the number of devices connecting to the internet are increasing day by day. Taking that into consideration one can imagine the volume of data continuously generated by these devices. These large quantities of data need to be properly stored and processed within a reasonable amount of time. The term 'Big Data' is used to describe these large volumes of data. Big Data 9 deals with storing huge amount of data and uses an analytics software and hardware to interrogate, analyze and generate useful information.

Based on the application of IoT, some data requires real-time processing to interact with the sensors and actuators whereas other data can be directly transported for storage. Data can be stored in the cloud or in the local servers. Later, the stored data are processed and analyzed in depth using analytics tools. One example of real-time data processing can be seen in the use of smoke sensors. The data from the smoke sensors needs to be continuously processed for early fire detection such that appropriate actions can be carried out immediately.

In order to provide quick computing and processing power for IoT devices Fog computing is used. Fog computing is a virtualized platform that provides computing, networking, storage and management services for end devices and it typically lies between the end devices and cloud services.

## **2.3 Security in Internet of Things:**

IoT uses end devices that collect data from the physical world and uses different wireless communication technologies to send these data for processing or storage. They either directly connect to the internet using different network protocols or use a gateway that handles all the internet protocols. If the IoT devices are not properly secured they can give access to hackers, allowing them to steal the data or even re-program the device causing them to malfunction. Lack of security in the IoT devices could even allow hackers to invade one's privacy. There have been many cases of hackers finding vulnerabilities in remote monitoring devices, where they are able to change the devices settings and even let other users to access these devices. That is why security plays a major role in IoT to protect it against malicious attacks and malfunctions.

Security in IoT application can be divided into three main categories: authenticity, confidentiality and integrity. Authenticity enables devices to identify each other and make sure that the received message is from the right owner. Confidentiality means the information obtained must be kept secret and no third parties should be able to access them. Integrity of data implies that the received data has not been modified in any way or corrupted during transmission. Data integrity check detects all intentional or accidental changes in the message. To provide authenticity and integrity on a message a short piece of information is used called message authentication code (MAC).

Due to its diverse nature, IoT requires security measures at different levels. At the lower level there are sensors with limited computing capacity that cannot provide strong security. In the middle layer there is networking that can be affected with denial of service (DoS), eavesdropping and interception. On the other hand, privacy and security are the main concern of the upper application layer. The use of strong encryption and authentication measure ensures the protection of user's privacy.

## **2.4 Use of IoT in the Public Transportation :**

Public transportation is a transport service available for general public that operates on fixed routes and charges a certain fee for each trip. Public transportation helps to reduce air pollution and traffic congestion by providing the service to many people at the same time. Public transportation also influences the quality of life in urban city. In order to afford living in the urban city people have to work and they are required to arrive at work on time. Sometimes they might also have meetings that are related to work. If the public transportation they are using are unreliable, they will be in constant stress when heading out to reach their destination. This negatively influences the quality of life. Also, arriving late to work or in a meeting can negatively affect a person's reputation. A poor public transportation can also refrain families from going out together, for example to a movie or to a park. Enjoying time with family can positively influence one's quality of life. When people have access to a public transportation that they know they can rely on and feel safe when using it, they might not consider buying their own personal transportation device, for instance, a car or a motorcycle. This can result in less traffic on the road. In addition, people would not have to spend their income on fuel and maintenance of their vehicle. This can reduce the financial burden, especially for a family that has a low-income source. Public transportation can also provide employment opportunities to many people. Employment also plays a major role that affects the influence on the quality of a person's life. Public transportation service needs to be affordable, well-organized and it should use the urban space efficiently. However, due to rapid urbanization and poor management of public transport service, the transportation system in Nepal is in dire need of improvement. Currently, most of the operation of public transport is carried out by privately owned organizations. There can be more than two different organizations running the public transportation service in the same route. In addition, these organizations lack proper coordination and they do not run on schedules. Due to which the roads in Nepal are congested with public vehicles and the public transportation service has become unreliable, resulting in poor service. This has led to big traffic congestion, increased pollution and increased road accidents. Also, the unreliability and poor transportation service has forced the general public to invest in their own personal transportation device. This has further increased the number of vehicles on the road. It is also difficult to collect the

records of number of people that use the public transportation service as there are no proper ticket system in place.

There are adequate number of vehicles but what they lack is a proper system and regular vehicular frequency. He also agrees with the fact that in order to improve, the government needs to invest more towards modernizing the public transportation services.

With the help of IoT, it will be easier to collect data from the public transportation services. IoT sensors can be installed in the public vehicles that can count the number of passengers in the vehicle. The data obtained through these sensors can be used in many ways. The data can be analyzed to find out the peak hours of the day when a larger number of people use the public transportation system. Using that information more vehicles can be dispatched during the peak hours. The data can also be used such that if the number of passengers reaches the passenger limit of the vehicle it will warn the driver or notify the authorities. Public vehicles can also be installed with GPS that can constantly track their locations and display them to the right agency. By joining the IoT devices with GPS, the authorities can constantly monitor the public vehicles in real-time from one centralized location. The data collected through IoT becomes crucial when planning for the future of the public transportation system.

### **3. COMMUNICATION TECHNOLOGIES IN IOT:**

According to Centenaro et al. wireless communication is the only feasible solution for a large majority of the IoT applications and services. Different types of wireless technologies are available in IoT that can be used either in short range communication or in long range communication. For short range communication that occurs in Wireless Personal Area Network (WPAN), where technologies like NFC, RFID, Bluetooth, ZigBee 6LoWPAN and Wi-Fi can be used. Whereas, for long range communication in Wireless Wide Area Network (WWAN), technologies such as Cellular and Low-Power Wide Area Networks (LPWANs) are available. Cellular technologies lie under licensed frequency bands and they include 2G/3G/4G and future 5G. On the other hand, LPWANs operate in the unlicensed frequency bands. SIGFOX and LoRa is the prominent technology of LPWANs.

#### **3.1 SHORT RANGE COMMUNICATION:**

In short range communication, the communication between the devices can occur within the range of 0–100 meters, 0 meter meaning the devices need to be in close contact in order to communicate. For example, when using Near Field Communication (NFC), the data is only transmitted when the devices come in contact or if they are no more than few centimeters apart.

Some of the short range communication technologies are NFC, RFID, Bluetooth low energy (BLE), Zigbee and 6LoWPAN.

NFC also known as Near Field Communication is a technology used for short range communication. It works by bringing two devices close to each other. The devices either have to touch or the distance between them should be no more than few centimeters apart. NFC uses the same technology as in RFID.

RFID is mainly used for identification purposes whereas NFC can be used for two way communication. The tag in the NFC can be used to identify an object or it can be rewritten with new data. NFC has two modes: active and passive. In the active mode both the devices can produce magnetic field. In the passive mode only one device produces the magnetic field and the second device modulates that energy to send back the data. The passive mode is suitable for small devices that need to run on limited battery power. Due to its proximity feature, one of the area NFC is used is for secure payment transactions.

RFID (Radio Frequency Identification) is one of the wireless communication technologies used to identify, record and track an object. RFID consists of two parts: readers and tags. Here, the readers are connected to the power source. Readers send out radio waves which are then picked by the tags. The tag contains a microchip and an antenna. The tag is connected to an object and the microchip in the tag helps to identify that object. By connecting the reader to the internet, it is possible to track and identify an object automatically and even globally. It can even track an object in real time. One of the major characteristics of RFID is that it can automatically identify an item, without any human interactions. Unlike bar code readers it does not require any line of sight. The range of communication in RFID depends upon its frequency.

BLE is a wireless technology that is created for low-powered devices. BLE was introduced in the market as Bluetooth 4.0. The IEEE standard for BLE is IEEE 802.15.1. BLE is suitable for short-range communication where devices need to send small quantities of data over short distances. Its main features are that it consumes less power and it also supports star topology. It is designed to be used for IoT applications. BLE consists of master and slave devices. Here, the master is the central device and slave devices are the smart devices that run on battery power. The master device controls all the connected slave devices. In order to save their battery life, devices are in sleep mode by default. One of the main differences between classic Bluetooth and BLE is that, in classic Bluetooth the devices are always connect with each other, this can strain the battery life of IoT devices. However, in BLE the IoT devices only wake up to receive message from the other device. BLE is used in fitness monitoring devices, smart watches and other monitoring devices.

Zigbee is a wireless communication protocol that is efficient for low-cost, lowpower and low-data rate applications. It is used for controlling and monitoring of devices in a personal area network. Zigbee is designed to be used in short range wireless communication for low data rate transfer. It provides a secured network for the devices to communication. Zigbee is useful for application where devices are required to run on battery power for long period of time up to few years. The communication distance in Zigbee can range from 10–100 meters. Any obstacles between the devices can affect the range of communication. Zigbee is based on IEEE 802.15.4 standard. The IEEE 802.15.4 standard provides the PHY and MAC layers and Zigbee adds network layers, application layers and security layers. Zigbee's network supports star, tree and mesh networking. In Zigbee, data can hop through different nodes that are connected in the mesh network, to reach its final destination. Zigbee is used in smart home devices, traffic management systems and for data collection that requires short range wireless communication.

6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks) standard is developed with the idea of using an IPv6 version of Internet Protocol (IP) over low-power and low-data rate wireless

networks. Using different protocols in sensor networks requires a dedicated gateway that translates the protocol to 15 standard IP protocols in order to connect to the internet. 6LoWPAN eliminates this process by using the standard IP based protocols where sensors can simply communicate using the standard and readily available bridges and routers. Another advantage of using IPv6 in sensor networks is that it has more addresses available than IPv4 and it can give a unique address to every device connecting to the internet. Similar to Zigbee, 6LoWPAN is built over low-power IEEE 802.15.4 protocol. 6LoWPAN can be applicable to be used in home automation, industrial automation and in healthcare automation along with real-time monitoring

### **3.2 LONG RANGE COMMUNICATION:**

When dealing with long range communication in IoT, it is important to take a note of its requirements, which are low-power consumption and low-data rate transfer. Additionally, it is equally necessary to consider the cost when using long range wireless communication technologies, especially in a situation where there can be hundreds of devices connected. One of the solutions to long range connection is to use cellular networks, as it can provide the coverage required for long range communication. But the cellular technologies that are currently used are not designed for IoT applications. For instance, they consume a large amount of power. Therefore, a new wireless technology called Low-Power Wide Area Networks (LPWAN) is used to meet the requirements of IoT applications for long range wireless communication. It provides low-power, low-cost and long range connection solutions for IoT devices. By using LPWAN the range of communication can be extended from 10–40 km in rural areas. On the other hand, due to the physical barriers present in the urban areas, the range of connection can only reach up to 5 km. In LPWAN, three of the most popular technologies are Sigfox, LoRa and NB-IoT.

Sigfox is a radio based LPWAN network operator that provides a cost effective solution to connected IoT devices. Sigfox optimizes their hardware design for a cost effective network solution. They deploy their own dedicated base stations. The base station receives messages from the devices, amplifies those messages and sends them to the Sigfox cloud. From the Sigfox cloud the message is forwarded to the customer's backend platform. In a Sigfox network the devices only wake up to forward a message, thus extending the battery life of a low-powered device. Sigfox network allows devices to only send small messages, up to 12 bytes in size, and the messaging capacity of each device is up to 140 messages per day. Sigfox provides bidirectional communication service, meaning that the connected devices can both send and receive messages. Due to their low cost feature and ease of configuration, Sigfox can also be used as a secondary network solution to many of the popular network types such as Wi-Fi and GPRS.

LoRa (Long Range) is a LPWAN technology developed by Semtech Corporation which is also responsible for developing the chipset used in the physical layer of LoRa. LoRa is based on the proprietary spread spectrum modulation technique which is a derivative of Chirp Spread Spectrum (CSS). In LoRa the physical layer is proprietary, however the upper layers, also known as LoRaWAN, are open. LoRaWAN handles the communication protocols and network architecture, whereas the physical layer of LoRa is responsible for long range communication link. LoRaWAN network uses star-of-stars topology.



The components in a LoRaWAN network consists of end devices, gateways, a network server and an application server. End devices generate data and with the help of LoRa and LoRaWAN forward these data to the gateways. The gateways are connected to the network server using different communication networks, such as 3G/4G, Wi-Fi and Ethernet. The network server receives the data packet from the gateways and carries out its security check. The network server then sends the data to the application server. The network server is also responsible for transferring data back to the end devices.

LoRa uses less power than Wi-Fi and provides long-range communication than Bluetooth, which is why it is a perfect candidate for long range IoT applications. LoRa can be used in smart technologies used in smart cities, such as smart meters and different monitoring sensors, where devices are required to send only few bytes of data at a time.

NB-IoT (Narrowband Internet of Things) is a LPWAN technology developed by the 3rd Generation Partnership Project (3GPP). It was added into their release of Release-13 back in 2016. NB-IoT was developed to meet the requirements of IoT by focusing on features such as wide coverage, long battery life, low-cost and reduced complexity. NB-IoT is not backward compatible with its existing 3GPP devices, however it can co-exist with Global System for Mobile Communication (GSM), General Packet Radio Service (GPRS) and Long-Term Evolution (LTE) technologies . LTE only requires a software update to support NB-IoT on their existing infrastructures. IoT devices require lower bandwidth to communicate, in order to save cost and battery life. For that reason, NB-IoT has reduced its minimum bandwidth requirement to 180 kHz. By reducing the bandwidth, it is possible to deploy NB-IoT inside one GSM carrier of 200 kHz and inside the physical resource block of one LTE carrier of 180 kHz. According to the communication protocol of NB-IoT bases on the LTE protocol, but NB-IoT reduces many functionalities of that protocol in order to support various IoT applications.

### **3.3 APPLICATION PROTOCOL:**

In IoT, end devices are connected to the back-end servers using different communication technologies. Depending on their applications, IoT devices are either connected through a gateway, that forms its own LAN, or through cellular networks that covers wider range. After a secured connection is made, the devices can send their data to the server or to a cloud using the internet. The server has to update its data every time a new value is produced by the end devices. Application layer protocols carry these new values from the end devices and deliver them to the servers. Application layer protocols are also responsible for delivering messages from the servers to users' applications. In addition, users are able to access and control the end devices through these protocols.

Popular application layer protocols such as HTTP, SMTP and FTP requires higher computing power to process them. For IoT applications, it is important to use those protocols that can work with devices running on battery power with lower computing capacity.

Some of the popular application layer protocols used in IoT applications are: MQTT, CoAP, AMQP, XMPP and DDS.

#### **3.3.1 MQTT:**

MQTT (Message Queue Telemetry Transport) is a lightweight messaging protocol that runs on top of TCP stack and uses a publish/subscribe protocol. MQTT is a simple and open sourced protocol which makes it a perfect candidate for IoT application and M2M communications where devices run on low power, low computing capacity and low bandwidth. In a publish/subscribe architecture, the publisher and subscriber do not request for updates, which decreases the network bandwidth and extends the battery life of the devices. Also, the TCP used in MQTT is modified so that it has low overhead.

In MQTT, there is a publisher that sends messages, a subscriber that receives messages and a broker that lies in between the publisher and the subscriber. Here, the publisher and the subscriber are each connected to the broker and they communicate with each other through the broker. The publisher and the subscriber cannot communicate with each other directly. The publisher sends messages to the broker, broker filters those messages and subscriber receives messages they are subscribed to, from the broker. MQTT supports various devices and platforms but they have high latency, making them unsuitable for real-time applications. In MQTT, security is handled by the broker. Security measure such as username/password authentication can be activated in the broker. The data transfer for authentication is encrypted using TLS (Transport Layer Security) or SSL (Secure Sockets Layer).

### **3.3.2 CoAP:**

CoAP (Constrained Application Protocol) is a request/response application protocol. CoAP is designed for constrained environment. Constrained environment consists of constrained nodes and constrained networks. Constrained nodes are devices that run on battery power and have low computing capacity. These nodes use constrained networks that have low bandwidth and low data transfer rate. In a machine to machine communication, CoAP provides low overhead, simplicity and reliable packet delivery. CoAP is designed by Internet Engineering Task Force (IETF). The main idea behind CoAP is that it can easily be translated to HTTP, so that the constrained devices can seamlessly communicate with the web.

In a CoAP, messages between the devices are exchanged over UDP (User Datagram Protocol). UDP reduces the overhead of TCP and the overall bandwidth of the network. It also supports multicast. UDP is considered to be an unreliable protocol, so in order to insure reliability CoAP has added its own mechanism to the UDP. CoAP adds two bits in its header when sending the packets. These new bits help identify the message type and its Quality of Service (QoS) level. In CoAP there are four different types of messages: Confirmable, Non-Confirmable, Acknowledgment and Reset.

1. Confirmable: It requires an Acknowledgment message to be send back in order to guarantee the reliability of the communication.
2. Non-Confirmable: It does not require any response.
3. Acknowledgment: This type of message is a response to the acknowledgment of the confirmable message.

4. Reset: This type of message informs that the received message could not be processed. Although CoAP is a simple and easy to use protocol in a constrained device, it has its own demerits. It lacks a built-in security features, has high latency and cannot be used with complex data type.

### **3.3.3 AMQP:**

AMQP (Advanced Message Queuing Protocol) is a messaging protocol that can use both a publish/subscribe and a request/response architecture. AMQP is mainly popular in the industrial messaging applications as it provides the required security, reliability, interoperability and scalability. For a reliable connection, AMQP uses TCP as a transport layer protocol. Here, the security in TCP is handled using TLS or SSL.

For reliability, AMQP uses message-delivery guarantees, which are: At most once (here a message is send at least once even if it is delivered or not), At least once (each message is certainly delivered once or more than once) and Exactly once (a message will be certainly delivered but only once)

In a publish/subscribe communication, AMQP can store messages in a queue before sending them to the subscriber. This process is handled using two components: Exchange queue and Message queue. Message queues can consist of more than one queue. Exchange queue is responsible for delivering the message to the right queue order. There is a set of rules that are pre-defined which helps Exchange queue to route the messages to its right queue order. The message queue can store these messages in the queue until it can send them to the receiver. In this way, even if there is a connection loss between the devices, the messages do not get lost and are stored in the queue. After the connection is up again, the messages in the queue are forwarded to the receiver.

### **3.3.4 XMPP:**

XMPP (Extensible Messaging and Presence Protocol) is an instant messaging protocol that was used for exchanging messages and video calls. XMPP was a well-known protocol that was widely used in the internet. It was developed in 1999 by an open source community for real-time messaging applications. XMPP is already two decades old. Since then, new types of data applications are constantly developed. Being an old protocol, XMPP could not support these new data types that are created by different data applications and because of this Google stopped supporting the XMPP standard. However, in the recent years it is used for IoT applications. XMPP supports exchanging small messages between devices, has low latency and can be used for real-time communication. Features like this has made XMPP a viable candidate for various IoT applications.

For text based communication, XMPP uses XML (Extensible Markup Language). XML adds a high network overhead that requires additional processing power, which means IoT devices will have to consume additional power in order to process them. This can be eliminated by compressing the XML using EXI (Efficient XML Interchange).

XMPP can support both a publish/share and a request/response messaging model. XMPP runs over TCP and gets all the security measures that are already built-in the TCP such as TLS/SSL. The reliability in

XMPP is ensured by the TCP. On the other hand, XMPP lacks QoS options which is an important aspect in the M2M communication. XMPP is a simple protocol that can be used in various applications. But when using XMPP for IoT, it is important to note that XMPP consumes high bandwidth and requires additional computing power, lacks QoS and can only be used for simple data types.

### **3.3.5 DDS:**

DDS (Data Distributed Service) is a protocol developed by Object Management Group (OMG). It follows the publish/subscribe model. It was developed to be used for real-time communication in the M2M architecture. DDS supports various levels of QoS policies that guarantee the reliability of the communication between the two devices. DDS uses standard such as security, reliability, priority, durability and many more. To achieve reliability and QoS, DDS uses broker-less architecture. This broker-less architecture helps DDS provide the real-time communication services for the IoT applications.

There are five entities that control the flow of data in DDS: the Topic, DataWriter, DataReader, Publisher and Subscriber. Before publishing the data, the Topic of DataWriter and the Topic of DataReader should be compatible. The topic should have the same name and the same data type. After that the application in the publisher side uses DataWriter to interact with the Publisher and writes the data value. The Publisher publishes the data to the Subscriber. The Subscriber sends the data to the DataReader that is associated with the DataWriter. Finally, the application in the subscriber side receives the data from the DataReader and the flow ends. Here, QoS is used to control the flow of data. Each of the entities have their own QoS policies. (OpenDDS.)

| Application layer protocols | Advantages   | Disadvantages   |
|-----------------------------|--|---|
| MQTT                        | Open sourced, less network bandwidth, uses TCP, data is encrypted using TLS/SSL  | High latency, not suitable for real-time application  |
| CoAP                        | Low overhead, simple and reliable packet delivery, easily translated to HTTP, supports multicast                         | Lacks built-in security, high latency, not suitable for complex data                                |
| AMQP                        | Uses TCP, provides security, scalable, store messages in queues, data is encrypted using TLS/SSL                         | Requires high processing power, extra memory, high bandwidth,                                       |
| XMPP                        | Low latency, suitable for real-time communication, uses TCP, data is encrypted using TLS/SSL                             | Lacks QoS, high bandwidth, requires additional computing power, only suitable for simple data types |
| DDS                         | Provides QoS policies, real-time communication, provides security, provides reliability through broker-less architecture | Not widely used, QoS policies requires strict DDS environment                                       |

#### Advantages and disadvantages of application layer protocols

The main advantages and disadvantages of each of the application layer protocols are highlighted that not all application layer protocols are suitable for applications that require real-time communication. It is equally important to study the disadvantages of the protocols to understand how that might affect the communication between two devices.

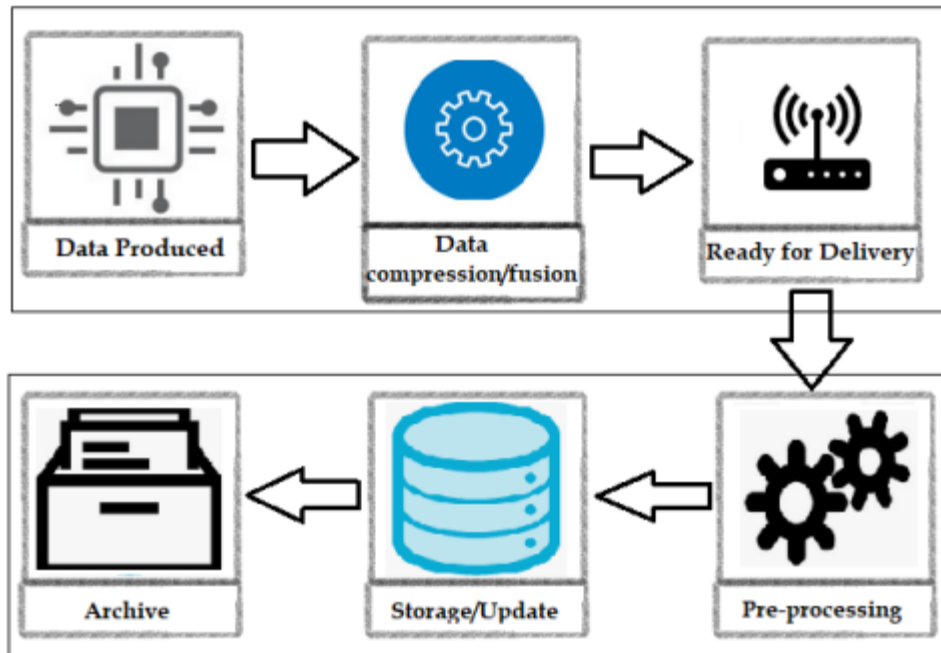
#### 4. DATA MANAGEMENT IN IOT:

Internet of Things consists of a network infrastructure that connects the physical objects to the internet. IoT uses various objects that are equipped with sensors to collect data from the environment and pass on that data for further processing. There can be more than one device connected in an IoT network and these devices can share information and data for making smart decisions. Based on that decision it is also possible to allow the objects to make real changes in the physical world. By doing this it can eliminate the need of human interaction to make new decisions. IoT acts as a platform where developers can develop new applications and services by harnessing the information collected using “Things”. Moreover, by fusing the already existing data with the new data that are collected using “Things”, it is possible to harness new information.

IoT has very unique needs when it comes to data management. When looking at the traditional data management systems, they mainly deal with the collection, storage, retrieval and the update of the data. In the case of IoT, it generates realtime data which can be heterogeneous in nature. These types of data need to be summarized first before they are sent to the network. The data that are collected online must be logged and audited before storing, so that they can later be retrieved whenever required. This process also helps to analyze the data offline. In IoT new data are constantly generated, due to which the offline data must be updated accordingly. Some applications in IoT require the access of offline data either to make real-time decisions or to look for trends to get new ideas. Unlike a traditional data management system that mostly deals with offline storage, data management in IoT has to deal with online-offline communication/storage dual operations.

#### **4. 1 DATA LIFECYCLE:**

For better understanding the data produced by IoT devices, it is important to look at the lifecycle of data in an IoT system. This starts with production, then moves to preprocessing, filtering and finally storing and archiving. Preprocessing can be done in a remote server which is far away from where the data is created. Or preprocessing can also be achieved at the edge of the network which is close to where the data is created. Not all data in an IoT system are required to be sent to a remote server for processing, especially on applications that require real-time access to the data. Sending these data to remote servers will only create latency and consume resources.



**Figure:1** Lifecycle of data in an IoT system

Lifecycle of a data in IoT starts from its production. Data is produced using sensors and devices that are used in an IoT network. IoT devices can be programmed to produce data periodically or they are programmed to produce data only when a query is sent. Some IoT devices are able to store the data for a short period of time before sending it to the gateway of the network. The devices use short range communication technologies to send these data to the gateways. There can be more than one device connected in an IoT network gateway and all the data from these devices are collected in that gateway. (Sb.)

After the data are collected in the gateway, the gateway performs filtration, data aggregation and compression of the raw data. This is done, because cost-wise it is very expensive to transmit a large amount of data through the network. They also have limited bandwidth, and constantly sending raw uncompressed data through the network creates latency. This can result in a life threatening situation in applications that require quick and real-time response and actions, for example in self-driving vehicles and health monitoring devices in a hospital. (Sb.)

The lifecycle of data in an autonomous IoT system working in real-time is different from that of other non-autonomous applications. An autonomous IoT system can refer to IoT devices used in factories to monitor the condition of machines and to check the air quality and humidity levels inside the buildings, sensors that are used in self-driving vehicles and smart devices used in hospitals that constantly monitor the condition of a patient. In such situations, after the data are produced they are instantly processed within the network. The device itself can also process the data without having to send them further up the network, provided that the device is equipped with enough computing power. Otherwise, the processing can take place at the edge of the network, in the gateways. This way the response time is

quicker and affective. The processed data are then stored within the network. One can access these data using end-devices that request the data from the network. This type of setup is suitable for applications where it is not necessary to send all the data further up the network for in-depth analysis and storage. The gateway can filter the data and only send the summarized version of the collected data. It is also important to note that the storage in these types of setups is limited, meaning that older data gets deleted when new data are updated. (Sb.)

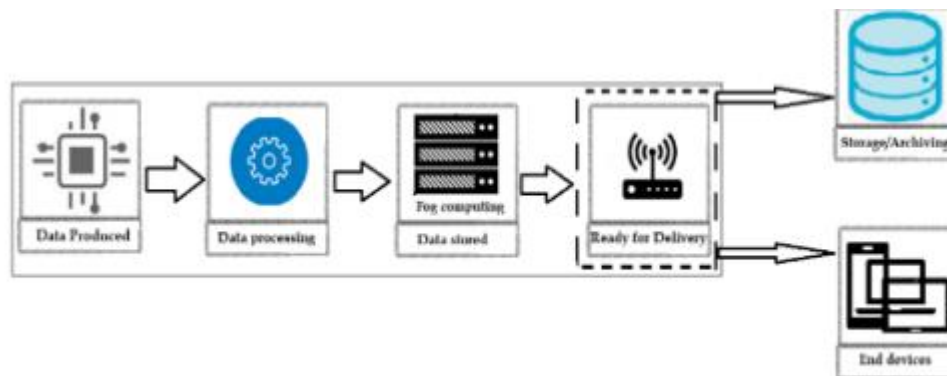


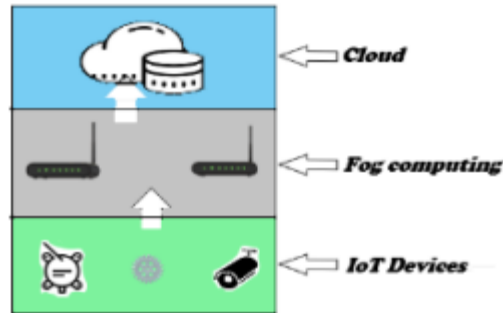
Figure 2. Lifecycle of an IoT data in an autonomous IoT system

The gateway of the IoT network uses wireless or wired broadband communication technology to send its data for permanent storage. Before storing the data, they are pre-processed to remove any redundancies and to see if any information is missing. The collected data might come from different sources and they can be heterogenous in nature which is why the data are integrated into a unified schema before they are stored. There are two ways of handling the final data in an IoT system. One is to store the data that can be easily accessed and updated when needed by the system. Another way is to archive the data. The data that do not need to be accessed by the system immediately for their current operation are sent for archiving. The process of storing the data in an IoT system can be an online/offline operation, whereas archiving is an offline, long term storage operation. The archival data are important for more in-depth analysis, and they can be combined with new data to gain insights and to determine future trends.

## 4.2 Fog Computing :

Fog computing is the process of bringing the computing power closer to the IoT devices rather than having them sent the data to the cloud for processing. Fog computing receives real-time data produced by the IoT devices and takes actions immediately. This type of setup is important in cases where a decision has to be made within seconds. Without fog computing, all the data created by the IoT devices has to do a round trip to the cloud for processing. In critical situations, the time it takes for the data to make a round trip to the cloud, can cause a catastrophic failure. (Cisco 2015.) Figure 3 shows how fog computing acts as a middle layer between IoT devices and the cloud.





**Figure 3:**Fog computing

IoT is used in many applications. They are used in manufacturing to increase productivity, for securities, to control traffic in big cities and in self-driving vehicles. In factories, machines are equipped with IoT sensors. The sensors study the condition of the machines, and if the sensor finds some error the sensor immediately sends a message warning about that error. Technicians can then work on that error before they have to completely shut down the machines, which would otherwise cost production loss of the entire company. IoT is also used for security purposes. IoT is used in security cameras to detect movements. If the sensor detects movements from the intruders, it can quickly notify the security personnel before the intruders can do any serious damage. (Sb.)

According to Statista Research Department (2020) the number of IoT devices connected worldwide will reach 38.6 billion by the year 2025. Having all these IoT devices send their data through the internet will cause a huge network traffic and increase latency. IoT devices can generate data rapidly, and in many situations that data needs to be processed within seconds. (Sb.)

A large factory can easily house more than a hundred IoT sensors. These sensors generate data that can be crucial to run the entire factory. They can also contain sensitive data. The amount of data coming from these IoT sensors can effortlessly consume the network bandwidth, causing latency and at the same time slowing down the response time. On the other hand, cloud is not a safe place to store sensitive data because clouds are prone to online attacks and they lack privacy. This is where fog computing becomes useful.

Fog computing is placed as near to the IoT devices as possible. They are usually situated inside the company network where all the IoT device are connected. (Sb.) Fog computing is created using fog nodes. Fog nodes can be any device that has enough computing power, enough storage and that can connect to the internet, which is why fog nodes can be routers, switches and even security cameras. Fog nodes can be placed anywhere in the building giving them the same physical security as any other IT devices present in a company or in a factory. Having the nodes inside the company network means they have better cyber security that protects their private data. IoT devices can easily communicate with the nodes and a decision can be made within milliseconds. This saves the company's network bandwidth. Holding the data closer to the company allows technicians to access them whenever required, in order to get a better insight. The nodes can be programmed to only send non sensitive and summarized data to the cloud for future analysis and for long term storage. (Sb.)

Fog computing helps companies to save costs by eliminating the need to transfer large volumes of data to the cloud. Fog computing brings the computing power closer to the IoT devices, which increases the response time and the productivity. With the help of fog computing the private data gets more secured from both cyber-attacks and physical attacks. (Sb.)

#### 4.3 Big Data:

As the name implies, Big Data is simply known as large volume of data. The internet produces petabytes of data every single day. That number is only expanding due to the advancement of IoT. IoT is already playing a huge role in different domains of the society such as healthcare, transportation, industries and agriculture. IoT uses sensors to interact with the environment and collect data. IoT then connects these sensors to the internet so that they can transfer and exchange the data with other devices. By deploying IoT in different domains of the society it is generating data in huge volumes. This will eventually create difficulties when dealing with storage, retrieval, security and analysis of the collected data. These problems can be tackled by Big Data. In an IoT system sensors, clouds, databases and machine-to-machine communication are the major sources for Big Data, as shown in Figure 4

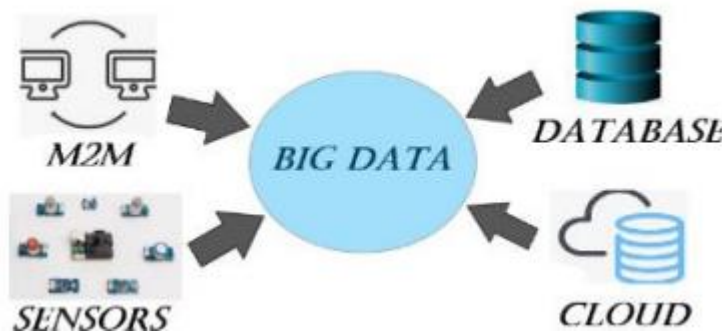


Figure 4: Big Data sources in IoT

In healthcare, IoT is incorporated with medical equipment's. Sensors placed in the equipment collect data which can be analyzed to provide better services. IoT can collect valuable information about different diseases and it can help find patterns that can be crucial when diagnosing the disease. IoT has also made it easier to monitor the patients. Patients are fitted with small sensors that continuously monitor their condition and these sensors help the medical team to respond quicker in case of an emergency. By merging IoT with healthcare system it has now been possible to provide healthcare services over long distances. The patients' data is stored in the cloud and can be accessed from anywhere by the medical team.

Another IoT domain that produces large volumes of data is transportation. Transportation is an important aspect of civilization. People use transportation for various purposes and the quality of transportation directly impacts the lifestyle of a city. The data collected from the sensors are used for controlling traffic, finding better routes for emergency vehicles and for surveillance. Furthermore, the collected data can also be shared with vehicles to minimize accidents. These data can later be analyzed when planning to add new routes.

Other IoT domains such as agriculture and industries are using IoT to increase their production. Farmers are able to remotely monitor their crops and they can control the different components of the greenhouse depending on the needs. In industries, the IoT sensors monitor different equipment's and their conditions. They are also used to check the quality of the products. The data collected from the IoT sensors are analyzed for research and to gain competitive advantage in the market (SAS 2020a).

According to SAS (2016b), Doug Laney defined Big Data as three Vs, which are: Volume (amount of data), Velocity (speed of data processing) and Variety (types of data). Big Data solves the problem of storing unlimited amount of data in a secured and organized manner. Many IoT domains use cloud computing to store their data. For example, healthcare systems store their data in the cloud which can be accessed from any location.

Traditionally data were stored using Relational database. Relational database use SQL to store its data in a table format. But with IoT, the collected data are heterogeneous in nature and cannot be stored in Relational database. In such situation NoSQL database is used to store the data. Big Data can be equipped with both NoSQL and Relational database, depending on the application. For instance, agriculture and healthcare system still use Relational database system to store their data, whereas smart cities use NoSQL database.

After storing the data, data analytic software is used to handle the data. Data analytic involves data mining to extract insights and data visualization. Devices can store and share data more efficiently, which allows them to make quick decisions. (SAS 2016b.) In IoT, Big Data is an inseparable field. Everyday IoT is producing data in huge amounts. Without Big Data IoT will face the problems relating to storage, data retrieval and data analyzing.

#### **4.4 Data Mining:**

Data mining is the process of extracting new information and patterns from the large volumes of data collected in a database. Data mining searches through the large volumes of data and analyzes them to find patterns and trends. This newly found patterns and trends are used for making future decisions. The process of mining valuable information from the data collected using machines are adopted by many business, governments and scientists. IoT has made it much easier to collect the data. But IoT is collecting data in huge volumes with high velocity. Data mining uses artificial intelligence and machine learning to find the hidden gems of knowledge quicker, which can then be implemented into action, resulting in better outcomes. (SAS 2020a.)

Organizations are using data mining for various purposes. They use data mining algorithms to detect any fraudulent credit card activities, calculate credit risks, manage resources and increase response time. Marketing departments are using data mining to expand their sales. Companies are collecting customers data and their buying patterns to suggest new products. Companies look at the trends to bring certain products on sale. With the help of data mining, organizations are able to reach the right customers for their marketing campaigns. Data mining is aiding organizations to improve their customer service, attract new customers and retain existing customers. Data mining is also applicable in

healthcare and in science. In healthcare data mining is used for identifying adverse side effects of drugs during clinical trial. (SAS 2020a, 1– 3).

As shown in Figure 5, there are seven phases in data mining one must follow to discover new information: Data Integration, Data Selection, Data Cleaning, Data Transformation, Data Mining, Presentation and Deployment. In data integration data are collected from different sources. From the collection, only the useful data are selected. The selected data are then checked for errors, redundancy and for missing values. The raw data are then transformed into data that are acceptable for data mining. After that, data mining algorithms and techniques are applied to the data. The obtained information and patterns are visualized using graphs which makes them easier for users to study and interact for finding meaningful patterns. In the last phase, the discovered patterns are used when making decisions. (Sumiran 2018.)

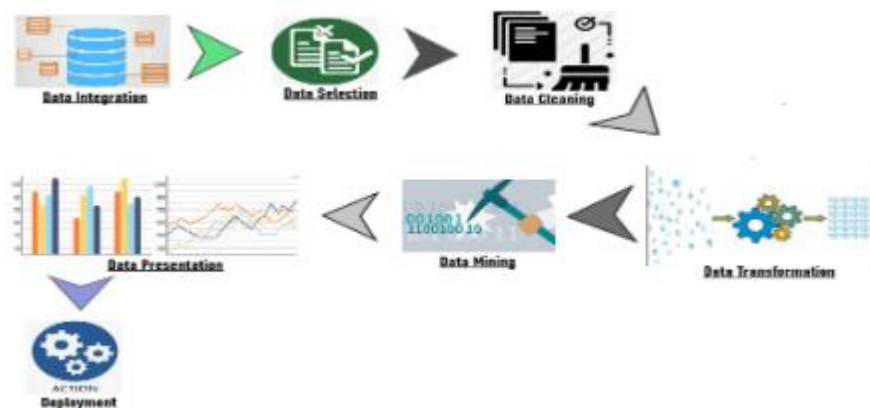


Figure 5: 5. Phases of data mining

Data mining uses various techniques and algorithms to identify a model or patterns from the available data. They are: Classification, Clustering, Regression, Artificial Intelligence, Neural Networks, Association Rules, Decision Trees and Genetic Algorithm. Different algorithms are suitable for different applications. In order to detect fraudulent credit applications, the algorithms are familiarized with both valid records and fraudulent records. Based on these, the algorithms determine whether the application is valid or fraud. For this type of setups Classification Techniques are applied. Regression Technique predicts the outcomes based on the known target values. This technique is used for predicting stock prices, sales and product failure rates. Another important data mining technique is Neural Network. Neural Network can adopt to the changes in the input without having to change the criteria of the output. Neural network is able to extract complex patterns that are too difficult for humans and other computer techniques. They are used for training computers to pronounce English texts and recognize handwritten characters. Neural network is suitable for predicting and forecasting needs and are already deployed in many industries.

Data mining has a high importance in IoT. The data collected from IoT are very valuable. But due to the velocity and volume of data it generates, humans by themselves are not able to take advantage of these data. Data mining is assisting many industries and businesses to grow. They are able to make

advantageous decisions while still maintaining good customer service. Any industry that generates data can take advantage of data mining.

## 5 SECURITY IN IOT:

The idea behind IoT technology is to create a network where devices can seamlessly connect with other devices to positively influence people's lives. IoT has already built its root in the society. Its application ranges from smart homes, to smart cities, smart healthcare and smart industries. The purpose of deploying IoT in these fields is to make people's life easier and comfortable while saving costs and energy. IoT collects huge amounts of data from these fields. The collected data can consist of personal and private information about users and companies. Without proper security hackers can exploit this information with the intention of harming an individual or a company. For instance, in smart homes, if the wireless security cameras have poor security features, attackers can use that camera to spy on people's lives. In healthcare systems smart devices are used to monitor the patients' health. These devices collect critical information of the patient and help track their health condition. Hackers can hack into these devices to give false reading which can danger the patient's life. IoT needs to provide strong security features in of its devices and the advancement of IoT will depend on its ability to gain users' trust by providing strong security and privacy features.

Security is a challenging topic in IoT. This is because IoT is heterogenous in nature. IoT consists of different technologies that are connected together to form an IoT network. IoT lacks a uniform standard and governance, which is why its security and privacy issue is a major concern. A simple IoT architecture consists of three layers: Sensing layer, Network layer and Application layer, as shown in Figure 6.

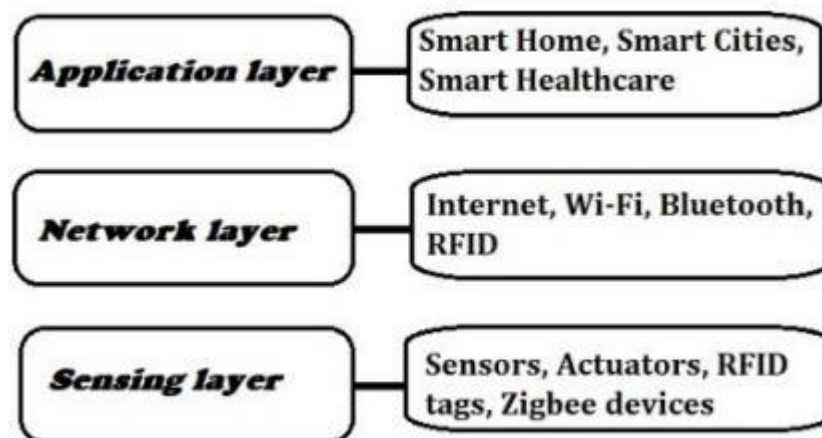


Figure 6. A simple IoT architecture

Each of these layers consists of different technologies and each of these technologies has their own security issues. From the security point of view connecting different technologies creates a wider area for network related attacks. If one of these devices is infected with malware it can become the starting point of infection and can be used to infiltrate the network, causing further damage.

## **5.1 Security Issues in Different Layers of IoT:**

Unlike traditional IT systems, securing an IoT system brings a whole new challenge. Due to its heterogeneous nature IoT systems consist of different devices with different platforms and each of these systems require their own security measures. Also, it is not possible to use traditional security solutions as IoT devices are resource constrained and cannot compute heavy processing. As mentioned earlier, a simple IoT architecture consists of three layers. These layers have to communicate with each other in order for the IoT system to function. If one of these layers is compromised it can bring the whole system to a halt. The security issues of each of the three layers are discussed below.

### **5.1.1 Sensing Layer:**

Sensing layer is made up of sensors and actuators that sense changes in the physical phenomenon of the world and makes changes in the physical world respectively. RFID tags and Zigbee devices are present in this layer. These devices are resource constrained and they are usually deployed in the open environment, making them more vulnerable to both cyber-attacks and physical attacks.

Being out in the open means that attackers can physically access and damage the device. They can perform node tampering where they can connect to the device and steal its sensitive information such as cryptographic keys and routing table. Attackers can also replace the node with a fake node and use that node to send malicious data. Since devices used in this layer are powered using battery, attackers try to drain its power by running infinite loops or by increasing its power consumption. Such types of attacks will lead to denial of service due to a dead battery. In the sensing layer devices are connected using wireless communication protocols. Because of this they are more susceptible to attacks such as eavesdropping and interference. Attackers are also known to use SideChannel Attacks where attackers can study the power consumption, electromagnetic radiation and time consumption of the sensor nodes to reveal its sensitive information.

Sensing layer is responsible for collecting data and based on these data certain action is performed. The collected data can be used for making important decisions. If an attacker interferes and continuously feed false data, for instance by creating a false node, it will compromise the integrity of the whole system.

### **5.1.2 Network Layer :**

Network layer deals with the communication between different IoT devices and IoT layers. It includes both short-range and long-range communication technologies. Network layer establishes a connection through which devices are able to exchange or share information. In IoT different types of network technologies has to communicate with each other. The interoperability among these networks can create security issues related to privacy and data loss.

Just like any other networks, IoT network can also be affected by attacks such as man-in-the-middle, distributed denial of service (DDoS), eavesdropping, jamming attacks and routing attacks. In the man-in-the-middle attacks, attackers listen to the communication between two devices. The attacker can

control the whole communication channel through which they can either modify the data passing through or steal valuable information that might give the access to the network. Another type of attack used for stealing information is eavesdropping attack. In this type of attack, attackers use sniffing tools to intercept wireless traffic between the two nodes. Attackers record data packets obtained through sniffing which they can later feed to a cryptographic tool and read the data. IoT networks consists of many individual nodes and attackers can use those nodes to overflow the network resulting in the DDoS attack. In other types of attacks, attackers use radio interference that blocks or corrupts the radio signals of the IoT devices. This type of attacks impacts the availability of the IoT systems resulting in denial of service.

Attacks in Network layer can result in loss of the quality of service and privacy concern. Without proper security mechanisms attackers can deplete the battery life of an IoT device using malicious jamming attacks. It is easier for attackers to target sensor nodes, because sensor nodes can operate without dedicated centralized server, otherwise the server could be used to monitor and control the network traffic .

### **5.1.3 Application Layer:**

Application layer is the most diverse layer of the IoT system. It consists of devices from different manufactures. Also, there are no universal standards present for constructing the application layer. Every application has a unique security requirement. The main purpose of the application layer is to provide services to the end users. An application such as smart home, smart cities and smart meters are present in this layer. Users use this layer to access data. The security challenges in the application layer are related to authentication, access control, data protection and data recovery.

There can be many users accessing the application in the application layer. Every user should have an access privilege assigned to their account that determines how much data they can access. Attackers can use phishing attacks or sniffing attacks to steal user credentials. Without proper authentication and access control measures attackers can compromise the entire IoT application. Application layer can also be affected by malware attacks. Attackers might use cross-site scripting (XSS) to inject an application with malware such as trojan horse, viruses and worms. In an XSS attack, attackers inject malicious script to a trusted website and when a user runs that script attackers can damage the whole system. Just like in the network layer, attackers can use DDoS attack to stop legitimate users from accessing the service. They can attack the servers and make it too busy to respond any request from the users.

By attacking the application layer attackers can get a hold of all the private data of the users. Once inside the application attackers can slowly work their way through the system and ultimately compromise the whole IoT application. They can leak the private data of the users in that system or use it to blackmail them.

| IoT Layers:   | Sensing Layer  | Network Layer  | Application Layer  |
|---------------|--|--|--|
| Attack types: | physical<br>damage, node<br>tampering, fake<br>node, Side-<br>Channel Attack | radio<br>interference,<br>eavesdropping,<br>DDoS, man-in-<br>the-middle, | data access and<br>authentication,<br>malware attacks,<br>cross-site<br>scripting,<br>phishing attacks |

|  |  |                                     |  |
|--|--|-------------------------------------|--|
|  |  | jamming attacks,<br>routing attacks |  |
|--|--|-------------------------------------|--|

IoT layers and its Attack types

## 5.2 Steps to Secure the IoT System:

Most of the IoT devices are deployed out in the open which makes them an easier target for the attackers. For that reason, the devices should be placed in an area where people cannot easily access them. The devices should also have proper access control and node authentication to block any illegal access. Besides that, the data between the nodes should be encrypted using non-linear key algorithms and cryptography techniques to insure confidentiality and integrity. To ensure that only legitimate users are able to access the data identity management systems should be implemented. Data transport layer security (DTLS) can be used for communicating between IoT devices. DTLS prevents threats such as eavesdropping and data tampering. The communication through the internet protocol (TCP/IP) can be secured using SSL or TLS. In the application layer strong authentication and access control mechanisms should be implemented. It is also important to make sure that the software used in the application layer is secure. Finally, users should be educated about security practices such as using strong passwords, being aware of spam emails and phishing websites.

When it comes to security in IoT there are no unified standard to define it, which is why it is necessary to develop a well-defined security and privacy policy that ensures confidentiality, access control and privacy of users' data. Before deploying the IoT devices they should be thoroughly tested against known attacks to make sure they can be used for that particular application. Since more and more users will be using IoT enabled applications in the near future it is important to protect their data and privacy.

## 6 PRACTICAL PART:

For the practical part the Raspberry Pi was used. The Raspberry Pi is a creditcard sized computer (Raspberry Pi Foundation 2020). The Raspberry Pi is used for many electronic projects. In the Raspberry Pi Raspbian was used as an operating system. Along with the Raspberry Pi two IR obstacle



sensors, three different colored LEDs (red, blue and red) and resistors were used. The IR obstacle sensor sends the input when a person passes through the sensor. This setup is suitable for a bus with two separate doors where one door is used for boarding the bus and another one for exiting. The IR obstacle sensor would be deployed in each of these doors. When a passenger enters or exits the bus, the sensor deployed at the door will trigger and update the information about the number of passengers in the bus. The LEDs were used as a visual representation; green LED represents a passenger entering the bus, blue LED represents a passenger exiting the bus and red LED will warn when the bus has reached its passenger limit.

### **6.1 Hardware Setup:**

The components used in this project are:

- One Raspberry Pi 3
- SD card
- Two IR Obstacle sensors
- Three LEDs
- Three resistors
- Wires
- Breadboard

The Raspberry Pi was mounted with a SD card which had Raspbian OS installed. Using the breadboard, the LEDs were connected to GPIO pins with protecting resistors. In this case GPIO17 was connected to the green LED, GPIO22 to the blue LED and GPIO20 to the red LED. The IR obstacle sensor has three pins: SIG, VCC and GND, as shown in figure 7.

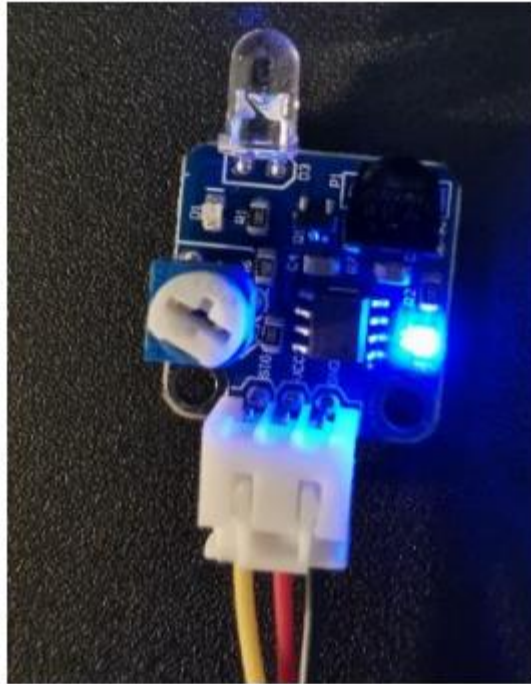


Figure 7: IR Obstacle Sensor

In the sensor SIG is for the signal, VCC and GND are for power and ground respectively. The SIG pin of the first IR obstacle sensor was connected to GPIO21 which would detect a passenger boarding the bus. The SIG pin of the second IR obstacle sensor was connected to GPIO26. This will detect a passenger exiting the bus. The power and ground pins were connected to their appropriate pins in the Raspberry Pi.

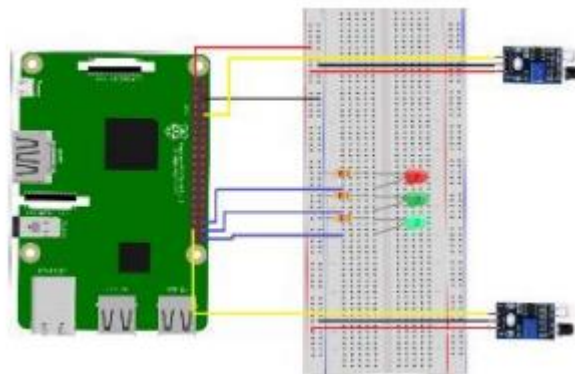


Figure 8: Raspberry Pi circuit diagram

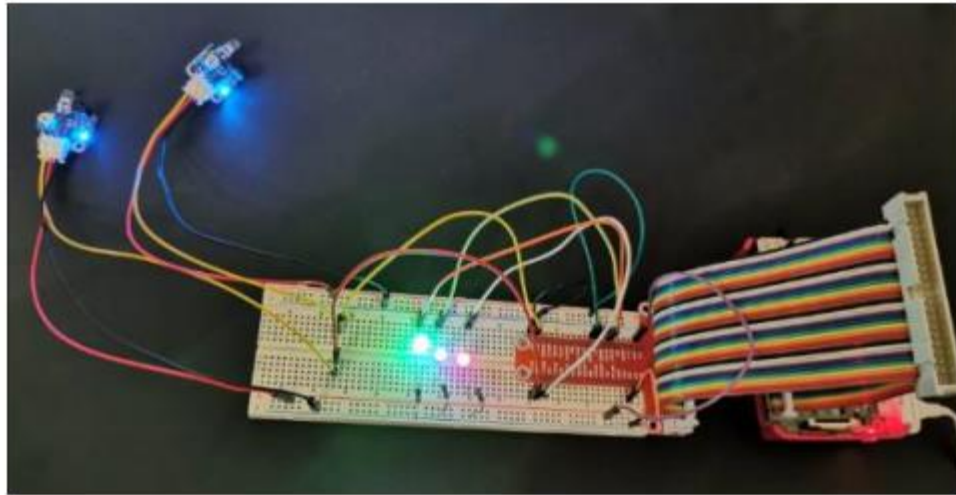


Figure 9: Raspberry Pi circuit

Figure 8 shows the circuit diagram of the project and Figure 9 shows how all of the devices are connected with the Raspberry Pi.

## 6.2 Python Code :

In the first part of the Python code different modules were imported. To control the LEDs and GPIO channels of the Raspberry Pi, gpiozero and RPi.GPIO modules were imported. To manipulate time and date, according to the needs, time and datetime modules were imported. For connecting to the MariaDB server a mysql.connector module was used. Finally, a csv module was imported to write the data obtained through the sensor to a csv file in the local directory. The Python code used for importing these modules are as following:

```
import RPi.GPIO as GPIO
from gpiozero import LED
import time
from datetime import datetime
import mysql.connector
import csv
```

After importing the necessary modules, the Python code goes on to assign the GPIO pins to its respective LEDs and sensors. For the numbering of the pins the BCM numbering system was used. The Python codes are as following:

```

passengerCountIn = 21 # detects when passenger enter the bus
LedIn = LED(17) # lights up to indicate passenger in
passengerCountOut = 26 # detects when passenger leaves the bus
LedOut = LED(22) # lights up to indicate passenger exit
LedMax = LED(20) # lights up to indicate the bus is full

GPIO.setmode(GPIO.BCM)
GPIO.setup(passengerCountIn, GPIO.IN)
GPIO.setup(passengerCountOut, GPIO.IN)

```

The next part of the code is used for connecting to the MariaDB server. This code contains the name of the server host, username, password and the database, as shown below:

```

mydb = mysql.connector.connect(host="localhost", user="gaurab",passwd="gaurav",
db="CSVFile") # connect to MariaDB server
cur = mydb.cursor()

```

The rest of the codes defines the function which are as following:

```

count = 0
dbMsg = "0"

def date_now():
    now = datetime.now()
    dateString = now.strftime("%Y-%m-%d") # shows date
    return(dateString)

```

```

def time_now():
    now = datetime.now()
    timeString = now.strftime("%H:%M") # shows time
    return(timeString)

def one_passenger():
    LedIn.on()
    time.sleep(0.1)
    LedIn.off()
    return("{} Passenger In!".format(count))

def more_passenger():
    LedIn.on()
    time.sleep(0.1)
    LedIn.off()
    return("There are {} Passengers in the bus!".format(count))

def max_passenger():
    LedIn.on()
    LedMax.on()
    time.sleep(0.1)
    LedIn.off()
    return("PASSENGER LIMIT REACHED!")

def one_passenger_leave():
    LedOut.on()
    LedMax.off()
    time.sleep(0.1)
    LedOut.off()
    return("Passenger Out! {} Passengers remaining!".format(count))

```

```

def zero_passenger():
    LedOut.on()
    LedMax.off()
    time.sleep(0.1)
    LedOut.off()
    return("No Passengers remaining!")

def overload_message():
    LedOut.on()
    time.sleep(0.1)
    LedOut.off()
    return("Passenger Out! {} Passengers remaining! PASSENGER LIMIT
REACHED!".format(count))

def write_csv(bb): # create csv file to save the data
    with open('/home/pi/Documents/Server/passenger_data1_2.csv', mode='a') as
passenger_data:
        sensor_data = csv.writer(passenger_data, delimiter=',',
quotechar='"', quoting=csv.QUOTE_MINIMAL)
        write_data = sensor_data.writerow([date_now(), time_now(), count, bb])
        return(write_data)

def databaseConnection(): # connect to sql database
    sql = "INSERT INTO Passenger_Data (DATE, TIME, STATUS, MESSAGE)
VALUES (%s, %s, %s, %s)"
    val = (date_now(), time_now(), count, dbMsg)
    cur.execute(sql, val)
    mydb.commit()
    print(cur.rowcount, "record inserted.")

while True:

```

```
if 0 == GPIO.input(passengerCountIn): # when passenger enters
```

```
    count += 1
```

```
    if count == 1:
```

```
        print(one_passenger())
```

```
        dbMsg = one_passenger()
```

```
        write_csv(one_passenger()) # writes to csv file
```

```
        databaseConnection() # writes to sql database
```

```
    elif count >= 15:
```

```
        print(max_passenger())
```

```
        dbMsg = max_passenger()
```

```
        write_csv(max_passenger())
```

```
        databaseConnection()
```

```
    else:
```

```
        print(more_passenger())
```

```
        dbMsg = more_passenger()
```

```
        write_csv(more_passenger())
```

```
        databaseConnection()
```

```
else:
```

```
    if 0 == GPIO.input(passengerCountOut): # when passenger exits
```

```
        if count <= 0:
```

```
            print(zero_passenger())
```

```
            dbMsg = zero_passenger()
```

```
            write_csv(zero_passenger())
```

```
            databaseConnection()
```

```
        elif 0 < count <= 15:
```

```
            count -= 1
```

```
            print(one_passenger_leave())
```

```
            dbMsg = one_passenger_leave()
```

```
            write_csv(one_passenger_leave())
```

```
            databaseConnection()
```

```
else:
    count -= 1
    print(overload_message())
    dbMsg = overload_message()
    write_csv(overload_message())
    databaseConnection()
```

### 6.3 Software Setup:

The software downloaded in the Raspberry Pi would allow a user to remotely access the database where the data collected through the sensors are stored. For this purpose, three different pieces of software were downloaded: NGINX, MariaDB and PHPMyAdmin. Before downloading these software's, the Raspberry Pi was updated using the following codes:

```
$sudo apt-get update
$sudo apt-get upgrade
```

First NGINX software was installed. NGINX is open source web server software. With NGINX, users will be able to access the database through a web browser using Raspberry Pi's IP address. NGINX was installed in the Raspberry Pi by running the following command in the terminal:

```
$sudo apt-get install nginx
```

After installing NGINX, php was also configured so that it will work with NGINX. For that, PHP 7.3 was installed using the following code:

```
$sudo apt-get install php7.3-fpm php7.3-mbstring php7.3-mysql php7.3-curl php7.3-gd
php7.3-curl php7.3-zip php7.3-xml -y
```

Once done, the configuration file of NGINX was also edited. The code used for opening the configuration file was:

```
$sudo nano /etc/nginx/sites-enabled/default
```

Inside the editor, the index line was edited to add 'index.php' as follows:

```
index index.php index.html index.htm;
```

In addition, inside the same editor, the following line was uncommented:



```
location ~ \.php$ {
    include snippets/fastcgi-php.conf;
    fastcgi_pass unix:/var/run/php/php7.3-fpm.sock;
}
```

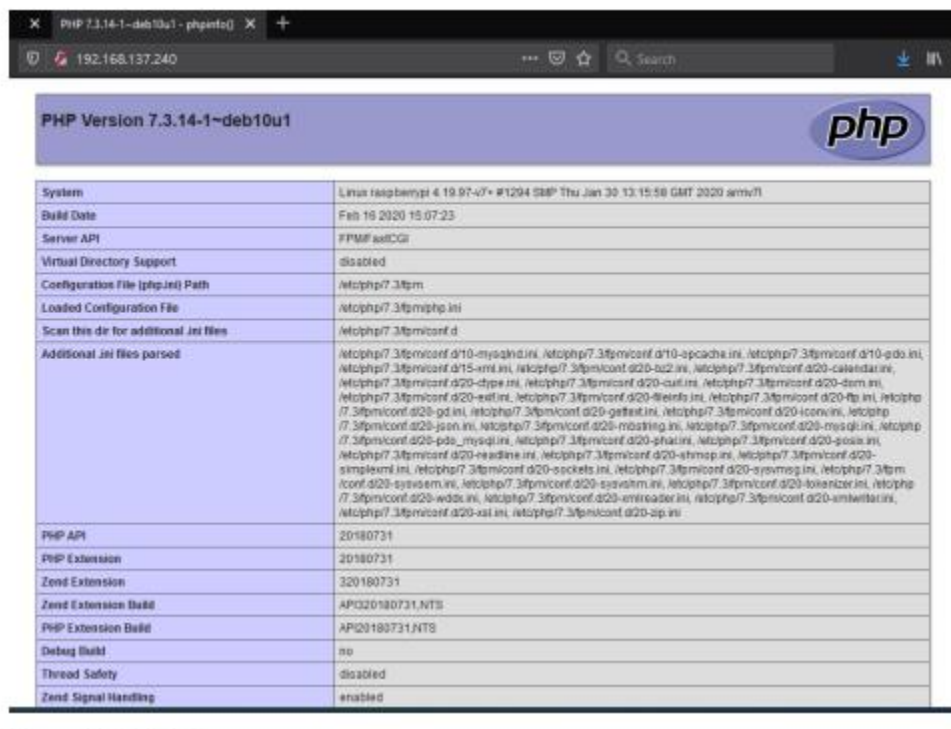
To test the PHP setup, inside the file `/var/www/html/index.php`, the following code was added:

```
$<?php phpinfo(); ?>
```

To start up the NGINX software, the following command was used:

```
$sudo systemctl start nginx
```

After finishing the NGINX web server setup with PHP, it could be accessed using the Raspberry Pi's IP address in the web browser. In this case the IP address was: `http://192.168.137.240/`, as shown in Figure 10.



|   |  |
|---|--|
| System                                  | Linux raspberrypi 4.19.97-v7+ #1294 SMP Thu Jan 30 13:15:58 GMT 2020 armv7l  |
| Build Date                              | Feb 19 2020 15:07:23   |
| Server API                              | FFPMF/fastCGI  |
| Virtual Directory Support               | disabled   |
| Configuration File (php.ini) Path       | /etc/php/7.3/fpm   |
| Loaded Configuration File               | /etc/php/7.3/fpm/php.ini   |
| Scan this dir for additional .ini files | /etc/php/7.3/fpm/conf.d  |
| Additional .ini files parsed            | /etc/php/7.3/fpm/conf.d/10-mysqlnd.ini, /etc/php/7.3/fpm/conf.d/10-openssl.ini, /etc/php/7.3/fpm/conf.d/10-pdo.ini, /etc/php/7.3/fpm/conf.d/15-xml.ini, /etc/php/7.3/fpm/conf.d/20-bcmath.ini, /etc/php/7.3/fpm/conf.d/20-calendar.ini, /etc/php/7.3/fpm/conf.d/20-curl.ini, /etc/php/7.3/fpm/conf.d/20-dom.ini, /etc/php/7.3/fpm/conf.d/20-ffi.ini, /etc/php/7.3/fpm/conf.d/20-fileinfo.ini, /etc/php/7.3/fpm/conf.d/20-ftp.ini, /etc/php/7.3/fpm/conf.d/20-gd.ini, /etc/php/7.3/fpm/conf.d/20-gettext.ini, /etc/php/7.3/fpm/conf.d/20-iconv.ini, /etc/php/7.3/fpm/conf.d/20-jpeg.ini, /etc/php/7.3/fpm/conf.d/20-mbstring.ini, /etc/php/7.3/fpm/conf.d/20-mysqlnd.ini, /etc/php/7.3/fpm/conf.d/20-pdo_mysql.ini, /etc/php/7.3/fpm/conf.d/20-phar.ini, /etc/php/7.3/fpm/conf.d/20-posix.ini, /etc/php/7.3/fpm/conf.d/20-readline.ini, /etc/php/7.3/fpm/conf.d/20-shmop.ini, /etc/php/7.3/fpm/conf.d/20-simplexml.ini, /etc/php/7.3/fpm/conf.d/20-sockets.ini, /etc/php/7.3/fpm/conf.d/20-sysvmsg.ini, /etc/php/7.3/fpm/conf.d/20-sysvsem.ini, /etc/php/7.3/fpm/conf.d/20-sysvshm.ini, /etc/php/7.3/fpm/conf.d/20-tokenizer.ini, /etc/php/7.3/fpm/conf.d/20-xmlrpc.ini, /etc/php/7.3/fpm/conf.d/20-xsl.ini, /etc/php/7.3/fpm/conf.d/20-zip.ini |
| PHP API                                 | 20180731   |
| PHP Extension                           | 20180731   |
| Zend Extension                          | 320180731  |
| Zend Extension Build                    | API320180731.NTS   |
| PHP Extension Build                     | API20180731.NTS  |
| Debug Build                             | no   |
| Thread Safety                           | disabled   |
| Zend Signal Handling                    | enabled  |

Figure 10: PHP info page

After that MariaDB software was installed. MariaDB is a relational database management system that is used for storing huge amounts of data. In the project all the sensor data are stored in the MariaDB database. To install MariaDB in the Raspberry Pi, the following command was used:

```
$sudo apt install mariadb-server
```

To stop unwanted users from accessing the database, MariaDB can be secured using passwords for the root users. To do that, the following command was used:

```
$sudo mysql_secure_installation
```

Finally, PHPMyAdmin software was installed. PHPMyAdmin is a graphical user interface for the MariaDB database. PHPMyAdmin makes it easier to edit the database using GUI instead of the command line. To install PHPMyAdmin the following command was used:

```
$sudo apt install phpmyadmin
```

PHPMyAdmin requires few inputs during the installation process. During the installation it asks to connect to the MariaDB database. For that, the password used while securely installing MariaDB is required. It will then ask for a new password for root users to access PHPMyAdmin. After completing the installation process a new user was created to access the tables within PHPMyAdmin. The new user was created inside the MariaDB command line. The following command is used for accessing the MariaDB command line:

```
$sudo mysql -u root -p
```

In the command line, a new user was created as follows:

```
$GRANT ALL PRIVILEGES ON *.* TO 'gaurav'@'localhost' IDENTIFIED BY 'gaurav' WITH  
GRANT OPTION;
```

Finally, PHPMyAdmin was linked with NGINX by running the following command in the terminal:

```
$sudo ln -s /usr/share/phpmyadmin /var/www/html
```

To access PHPMyAdmin through the browser, Raspberry Pi's web address was used. In this case it was: <http://192.168.137.240/phpmyadmin/>.

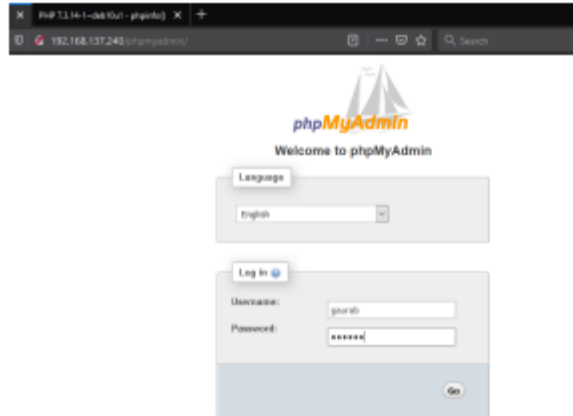


Figure 11: PHPMyAdmin login page

Figure 11 shows the login page of PHPMyAdmin that was accessed from a remote web browser.

#### 6.4 Data Collection:

For data collection in the MariaDB database I first had to create a database and a table. First, through the web browser I connected to PHPMyAdmin. Using the username and password that I created I was able log in to PHPMyAdmin as shown in Figure 12.

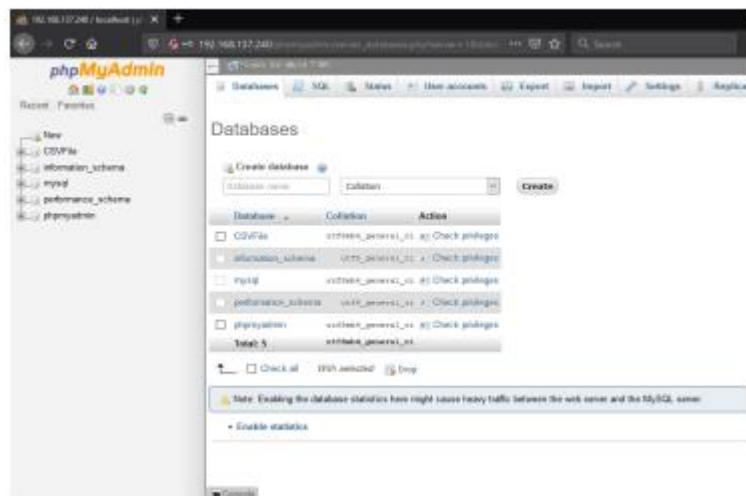


Figure 12: PHPMyAdmin server

Inside the server I created a new database called CSVFile using Create database as shown in Figure 13.

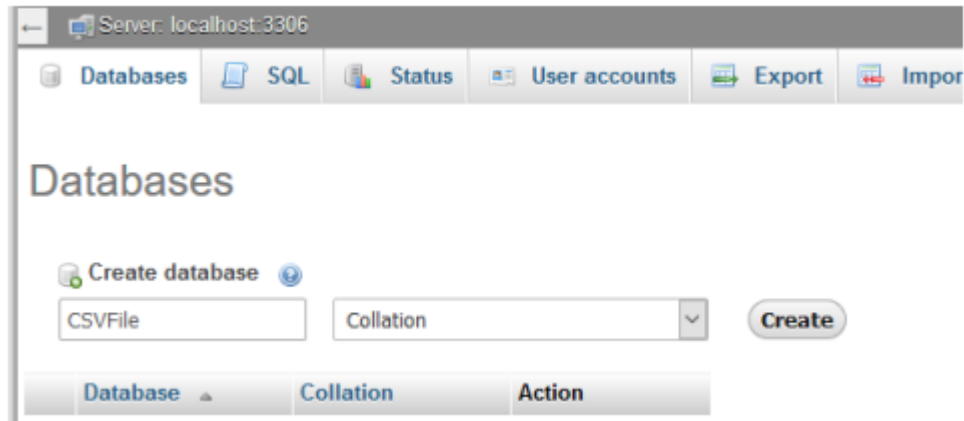


Figure 13: Creating a new database

Inside the CSVFile database I created a new table called Passenger\_Data. This is introduced in Figure 14.

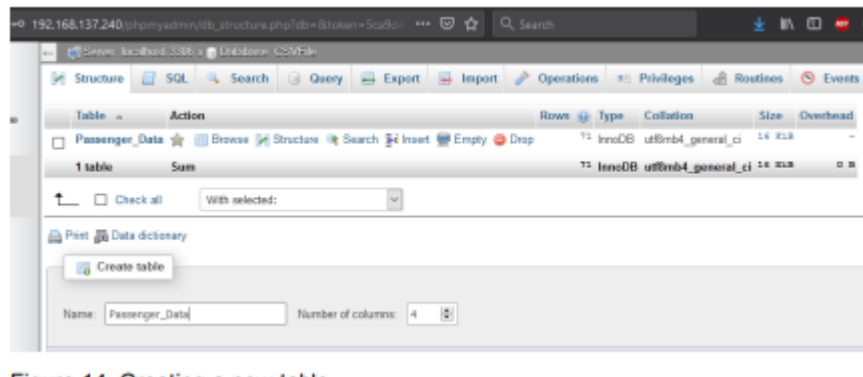


Figure 14: Creating a new table

Inside the table I created four columns: DATE, TIME, STATUS, MESSAGE. The value for all of these columns will be filled using python code which is why I put TEXT as a Type for DATE, TIME and MESSAGE, which is shown in Figure 15. The status will show how many passengers are in the bus.

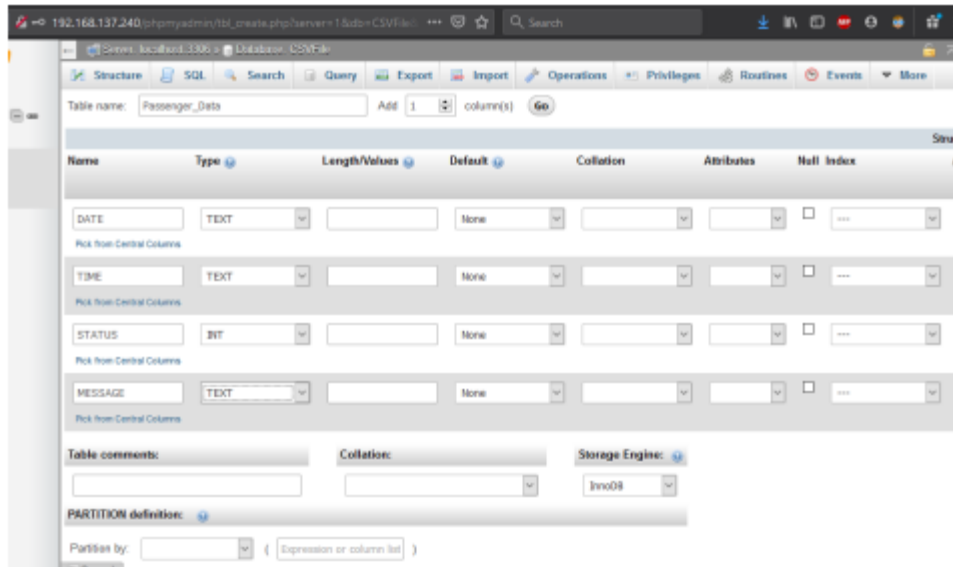


Figure 15: Creating columns and their types

After creating the database and a table inside the database, I added that information in my Python code. Through that code the data from the sensor was copied to the sql database table. After successfully uploading the data to the database it prints out a message "1, 'record inserted.'", as shown in Figure 16.

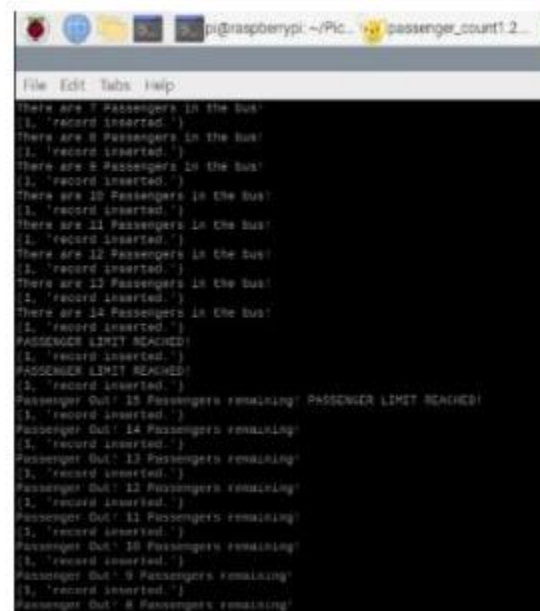


Figure 16: Sensor sending data to the database

In the PHPMYAdmin server, inside the Passenger\_Data we can click refresh to see the new data that has been copied to the database as Figure 17 shows

The screenshot shows the phpMyAdmin web interface. On the left, the database structure is visible, including 'information\_schema', 'mysql', 'performance\_schema', and 'phpmyadmin'. The 'Passenger\_Data' table is selected. The main area displays a table with columns: DATE, TIME, STATUS, and MESSAGE. The data shows a sequence of 18 entries from 2020-05-04 01:03 to 01:15, with messages indicating the number of passengers in the bus (1 to 15) and then 'PASSENGER LIMIT REACHED'.

| DATE       | TIME  | STATUS | MESSAGE   |
|------------|-------|--------|---|
| 2020-05-04 | 01:03 | 1      | 1 Passenger in the bus                          |
| 2020-05-04 | 01:03 | 2      | There are 2 Passengers in the bus               |
| 2020-05-04 | 01:03 | 3      | There are 3 Passengers in the bus               |
| 2020-05-04 | 01:03 | 4      | There are 4 Passengers in the bus               |
| 2020-05-04 | 01:03 | 5      | There are 5 Passengers in the bus               |
| 2020-05-04 | 01:03 | 6      | There are 6 Passengers in the bus               |
| 2020-05-04 | 01:03 | 7      | There are 7 Passengers in the bus               |
| 2020-05-04 | 01:03 | 8      | There are 8 Passengers in the bus               |
| 2020-05-04 | 01:03 | 9      | There are 9 Passengers in the bus               |
| 2020-05-04 | 01:03 | 10     | There are 10 Passengers in the bus              |
| 2020-05-04 | 01:03 | 11     | There are 11 Passengers in the bus              |
| 2020-05-04 | 01:03 | 12     | There are 12 Passengers in the bus              |
| 2020-05-04 | 01:03 | 13     | There are 13 Passengers in the bus              |
| 2020-05-04 | 01:03 | 14     | There are 14 Passengers in the bus              |
| 2020-05-04 | 01:03 | 15     | PASSENGER LIMIT REACHED                         |
| 2020-05-04 | 01:03 | 16     | PASSENGER LIMIT REACHED                         |
| 2020-05-04 | 01:03 | 17     | PASSENGER LIMIT REACHED                         |
| 2020-05-04 | 01:03 | 18     | Passenger Out 15 Passengers remaining PASSENGER |

Figure 17:New data in the database

We can also access these data from the Raspberry Pi terminal. To do that we can use the following commands:

```
$sudo mysql -u root -p
$show databases;
$use CSVFile;
$show tables;
$SELECT*FROM Passenger_Data;
```

The output of these commands is shown in Figure 18

The screenshot shows a terminal window on a Raspberry Pi. The user has executed the following commands: `sudo mysql -u root -p`, `show databases;`, `use CSVFile;`, `show tables;`, and `SELECT*FROM Passenger_Data;`. The output shows the 'CSVFile' database selected and the 'Passenger\_Data' table listed. The final command displays the same 18 rows of data as seen in Figure 17.

```
mysql> show databases;
+-----+
| Database |
+-----+
| CSVFile |
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
+-----+

mysql> use CSVFile;
Database changed

mysql> show tables;
+-----+
| Tables_in_CSVFile |
+-----+
| Passenger_Data |
+-----+

mysql> SELECT*FROM Passenger_Data;
+-----+
| DATE | TIME | STATUS | MESSAGE |
+-----+
| 2020-05-04 | 01:03 | 1 | 1 Passenger in the bus |
| 2020-05-04 | 01:03 | 2 | There are 2 Passengers in the bus |
| 2020-05-04 | 01:03 | 3 | There are 3 Passengers in the bus |
| 2020-05-04 | 01:03 | 4 | There are 4 Passengers in the bus |
| 2020-05-04 | 01:03 | 5 | There are 5 Passengers in the bus |
| 2020-05-04 | 01:03 | 6 | There are 6 Passengers in the bus |
| 2020-05-04 | 01:03 | 7 | There are 7 Passengers in the bus |
| 2020-05-04 | 01:03 | 8 | There are 8 Passengers in the bus |
| 2020-05-04 | 01:03 | 9 | There are 9 Passengers in the bus |
| 2020-05-04 | 01:03 | 10 | There are 10 Passengers in the bus |
| 2020-05-04 | 01:03 | 11 | There are 11 Passengers in the bus |
| 2020-05-04 | 01:03 | 12 | There are 12 Passengers in the bus |
| 2020-05-04 | 01:03 | 13 | There are 13 Passengers in the bus |
| 2020-05-04 | 01:03 | 14 | There are 14 Passengers in the bus |
| 2020-05-04 | 01:03 | 15 | PASSENGER LIMIT REACHED |
| 2020-05-04 | 01:03 | 16 | PASSENGER LIMIT REACHED |
| 2020-05-04 | 01:03 | 17 | PASSENGER LIMIT REACHED |
| 2020-05-04 | 01:03 | 18 | Passenger Out 15 Passengers remaining PASSENGER |
+-----+
```

Figure 18:Accessing the table through command line

Another advantage of using a PHPMyAdmin server is that we can easily convert the sql data into visual representation.

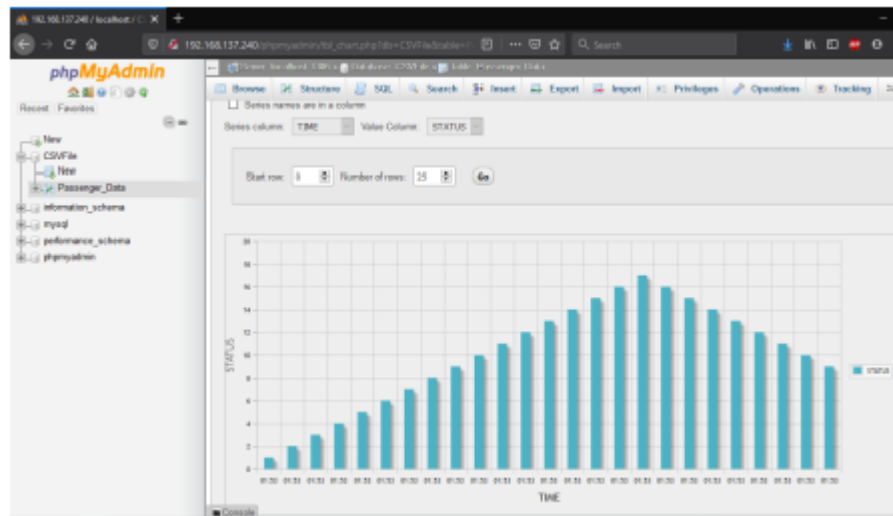


Figure 19: Column graph

Figure 19 shows the visual representation of the data that was collected earlier using the Raspberry Pi.

## 7 CONCLUSION:

The aim of the project was to use IoT in the public transportation system of Nepal through which the Department of Transport Management could efficiently monitor, control and improve the public transportation services. In the practical part of the project IoT sensors were used to monitor the passenger data of a bus. It could also provide a visual warning for the driver, if the number of passengers in the bus exceeded its limit. Furthermore, the project goes on to show the possibility of monitoring this data through remote access.

The project also introduced IoT and its technologies. It briefly discussed the applications of IoT and how they were used in different sectors of the society. It explained how the data collected from IoT have become more crucial when making important decisions and when planning for the future. Businesses are using IoT to stay ahead in the market, cities are using IoT to save energy and resources. At the same time IoT is used in agriculture, transportation and healthcare. Some of the security features of IoT were also discussed.

This project was only a prototype. It showed how one can collect, store and access the data using IoT. The use of Big Data to analyze and process the collected data and how to use them in the real working environment could be done in future work. The project can be improved by adding new features through further research. To build this on all the buses in any city requires official decisions and funding.