PROJECT TITTLE: FLOOD MONITORING SYSTEM

TEAM MEMBERS:

KRISHNA PRIYA.S

KALAIMATHI.S

PABITHA.J

MADHAVI.M

ABINAYA.A

PHASE-3: DEVELOPMENT PART-1

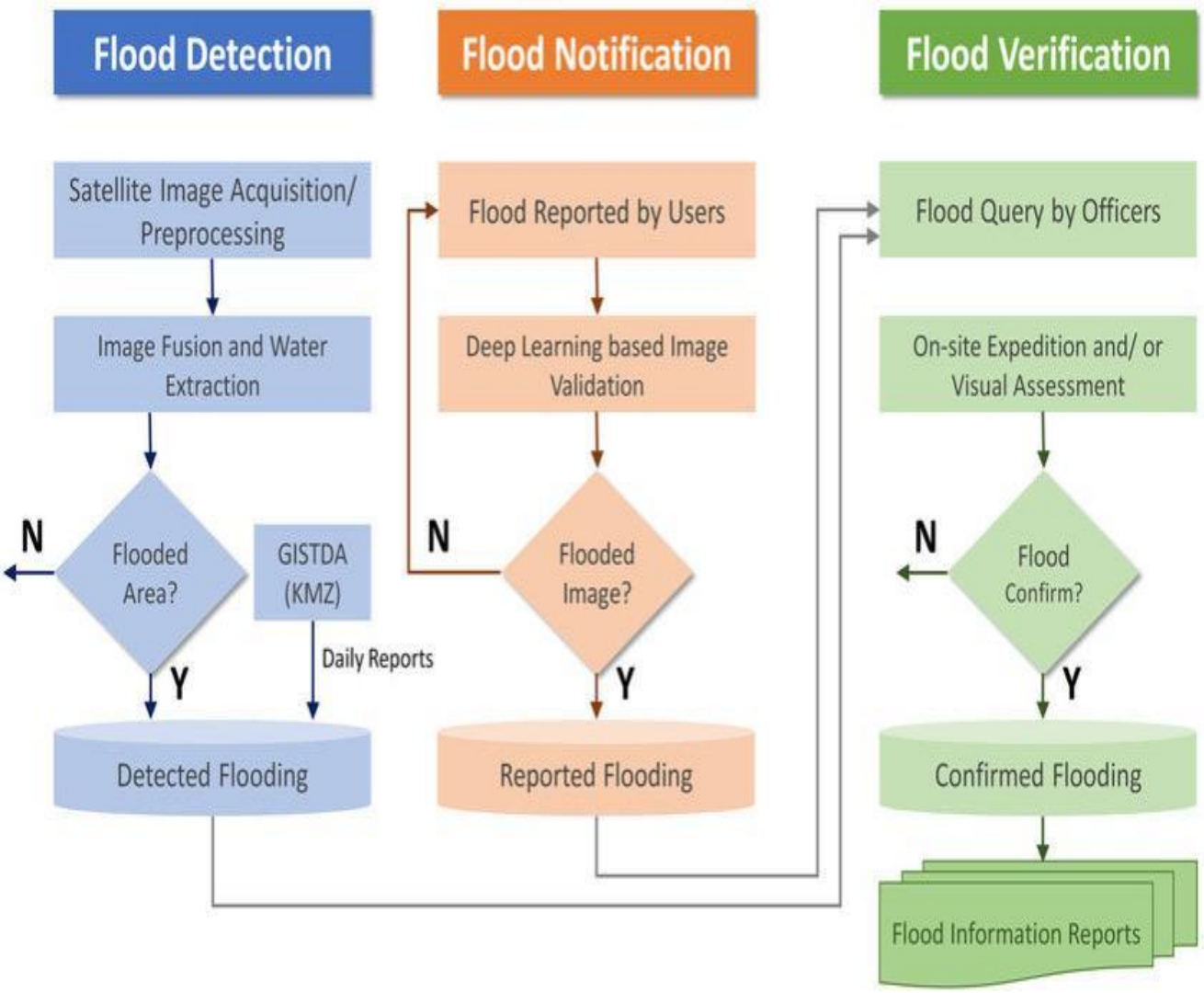
- **INTRODUCTION:**

  Floods are a major natural hazard that can cause significant damage to propertinfrastructure. Effective flood monitoring and early warning systems are essential for reducing the risks associated with floods.Flood monitoring systems rely on data to generate accurate forecasts and warnings. This data can come from a variety of sources, including water level sensors, rain gauges, satellite imagery, and aerial drone surveys.

Flood monitoring system datasets are used to train AI models to predict flood conditions and to develop early warning systems. These datasets are essential for developing and improving flood monitoring systems.

FLOOD DETECTION:

Flood detection is the process of identifying and monitoring flood events. It is important for early warning and evacuation, as



well as for damage assessment and recovery.

Satellite imagery:

- Satellite imagery can be used to identify areas that have been flooded. This information can be used to assess the extent of flooding and to guide rescue and relief efforts.

Image fusion and water extration:

- Image fusion is used in a variety of applications, including flood monitoring. By fusing images from different sensors, such as optical and radar satellites, we can create images that have a higher resolution and that can better distinguish between water and other land features. This can be used to more accurately detect and map flood extents.
- Water extraction is the process of identifying and separating water pixels from other pixels in an image. It is a key step in flood monitoring, as it allows us to measure the extent of flooding and to track changes over time.

NECESSARY STEPS TO FLOW:

- **Collect data:**

The first step is to collect data from a variety of sources, such as rain gauges, water level sensors, and satellite imagery. This data can be collected in real time or at regular intervals.

- **Preprocess the data:**

Once the data is collected, it needs to be preprocessed to ensure that it is in a format that can be used by the flood monitoring system. This may involve cleaning the data, removing outliers, and converting the data to a consistent format.

- **Develop a flood detection model:**

The next step is to develop a model to detect floods. This can be done using a variety of machine learning algorithms, such as support vector machines, random forests, and neural networks.

- **Train the model:**

The flood detection model needs to be trained on a dataset of historical flood events. This will allow the model to learn how to identify floods in new data.

- **Evaluate the model**:

Once the model is trained, it needs to be evaluated on a held-out dataset to assess its performance. This will help to identify any areas where the model needs to be improved.

- **Deploy the model**:

Once the model is trained and evaluated, it can be deployed to production. This means making the model available to users so that they can use it to detect floods.

Here is a simple example of a flood detection system in Python:

Python

```
Import numpy as np

Import pandas as pd

From sklearn.svm import SVC


# Load the data

Data = pd.read_csv('flood_data.csv')


# Preprocess the data

X = data.drop('flood', axis=1)

Y = data['flood']


# Split the data into training and test sets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)


# Train the model

Model = SVC()

Model.fit(X_train, y_train)
```

```
# Evaluate the model

Y_pred = model.predict(X_test)

Accuracy = np.mean(y_pred == y_test)

Print('Accuracy:', accuracy)


# Deploy the model

Def detect_flood(data):

  Prediction = model.predict(data)

  Return prediction[0]


# Example usage:


# Get the current data

Current_data = get_current_data()


# Detect a flood

Flood_detected = detect_flood(current_data)


# If a flood is detected, take appropriate action

If flood_detected:

  Send_flood_warning()
```

This is a very simple example, and it can be extended to incorporate more complex features and models.

Benefits of loading and preprocessing the dataset for a flood monitoring system:

- **Improved accuracy and reliability:**
  By ensuring that the dataset is consistent and accurate, we can improve the accuracy and reliability of the flood monitoring system.
- **Reduced risk of errors:**
  By cleaning and preprocessing the data, we can reduce the risk of errors in the flood monitoring system. This is because the system will not be able to learn from inaccurate or inconsistent data.
- **Improved performance:**
  By pre-processing the data, we can improve the performance of the flood monitoring system.
- **Reduced costs:**
  By pre-processing the data, we can reduce the costs associated with the development and maintenance of the flood monitoring systems.

Challenges Involved in Loading and Preprocessing a Flood Monitoring System Dataset:

- **Data availability and quality**:

Flood monitoring system datasets can be large and complex, and it may be difficult to find datasets that are complete, accurate, and up-to-date. Additionally, the quality of the data can vary widely, and it may be necessary to clean and preprocess the data before it can be used.

- **Data heterogeneity**:

  Flood monitoring system datasets can be heterogeneous, meaning that the data may come from a variety of sources and in a variety of formats. This can make it difficult to load and integrate the data into a single dataset. Use

- **Scalability**:

  Flood monitoring system datasets can be very large, and it can be challenging to load and preprocess these datasets on a large scale.

- **Time constraints:**

Flood monitoring systems need to be able to process data and generate predictions in real time. This can be challenging, especially if the dataset is large and complex.

Dataset of Flood Monitoring System:

To calculate the flood monitoring system of a dataset, we can use the following steps:

- Calculate the average water level.
- Calculate the standard deviation of the water level.
- Calculate the flood warning level and flood alert level.
- Create a new DataFrame containing the flood monitoring system data

```
Import pandas as pd
Import numpy as np
df=pd.Data Frame({
    'Date':pd.to_date time (['2023-08-01','2023-08-02','2023-08-03','2023-08-04','2023-08-05'])
   'Water_level':[10,11,12,13,14]
})
#Calculate the average water level
avg_water_level=df['water_level'].mean()
#Calculate the standard deviation of the water level
Std_water_level=df['water_level'].std()
#Calculate the flood warning level
Flood _warning_level=avg_water_level+2*std_water_level
#Create a new Data frame containing the flood monitoring system data flood _monitoring_system_df=pd.data frame({
     'Date':df['Date'],
      'water_level':df[water_level'],
'flood_warning_level': flood_ monitoring_level,
'flood_alter_level': flood _alter_level
})
#print the flood monitoring system data
```

Print(flood_monitoring_system_df)

OUTPUT:

Date	Water_level	Flood _warning level	Flood _alter_level
2023-08-01	10	15.162278	16.743416
2023-08-02	11	15.162278	16.743416
2023-08-03	12	15.162278	16.743416
2023-08-04	13	15.162278	16.743416
2023-08-05	14	15.162278	16.743416

The flood warning level is the water level at which a flood warning is issued. The flood alert level is the water level at which a flood alert is issued. A flood warning is issued when the water level is expected to reach or exceed the flood warning level. A flood alert is issued when the water level is expected to reach or exceed the flood alert level.

How to overcome the challenges involved in loading and preprocessing a flood monitoring system datas

- **Choose the right tools and technologies**:

There are a variety of tools and technologies available for loading and preprocessing large and complex datasets. Choose the tools and technologies that are best suited for your specific needs and requirements.

- **Use a consistent data format:**

Choose a consistent data format for your flood monitoring system dataset. This will make it easier to load and process the data in a consistent manner.

- **Handle missing data carefully:**

There are a variety of ways to handle missing data in a flood monitoring system dataset. Choose the approach that is best suited for your specific needs and requirements.

- **Clean and filter the data:**

It is important to clean and filter the data to remove noise and inconsistencies. This will help to improve the accuracy and reliability of the flood monitoring system.

Loading the dataset of a flood monitoring system

- **Identify the data sources:**

The first step is to identify the data sources that you need to load. This may include historical rainfall data, river levels, flood maps, satellite imagery, and other relevant data.

- **Collect the data:**

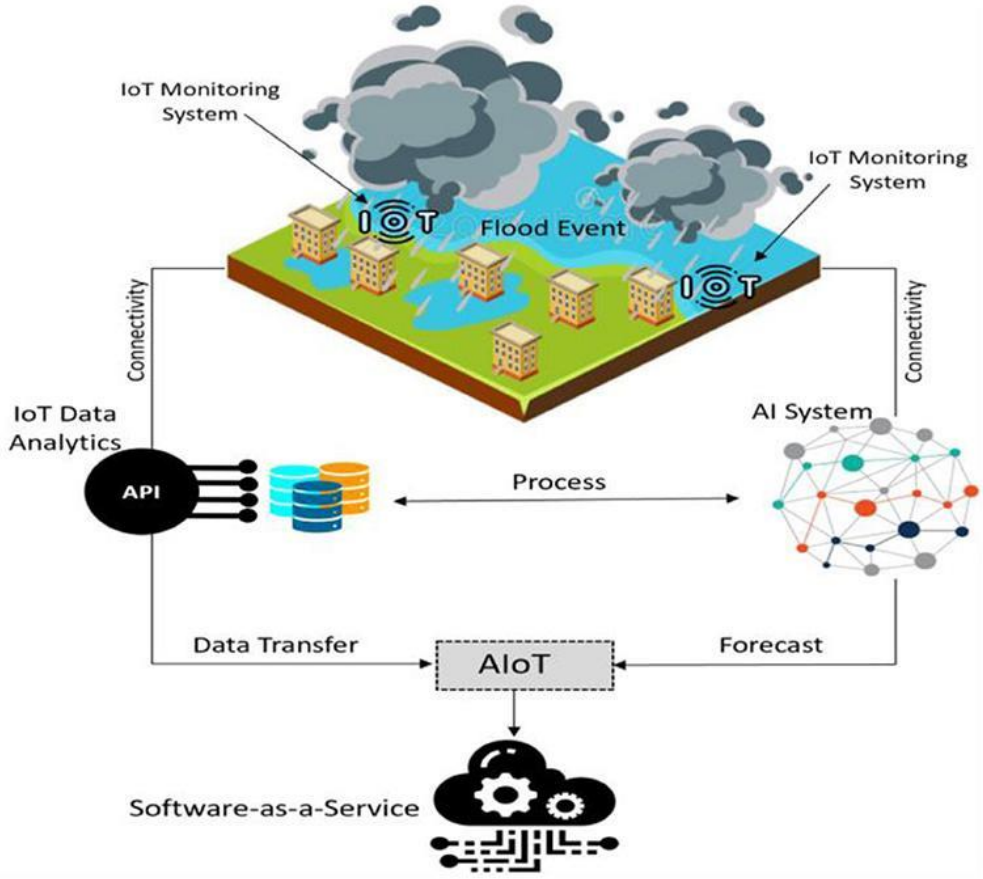
Once you have identified the data sources, you need to collect the data. This may involve downloading data from websites, scraping data from web pages, or accessing data from databases.

- **Clean and pre-process the data**:

Once you have collected the data, you need to clean and pre-process it. This may involve removing noise and inconsistencies, converting the data to a consistent format, and handling missing data.

- **Load the data into a data store**:

Once the data is clean and pre-processed, you can load it into a data store. This may involve a relational database, a NoSQL database, or a cloud-based data store.



Water level of dataset:

Name of the Dam	Max water level	Threshold water level	Current status water level	% of Dam filled
Selaulim	**41.15m**	**20.42m**	**37.14m**	**67**
Aujunem	**93.20m**	**61.50m**	**88.82m**	**77**
Chapoli	**38.75m**	**22.00m**	**36.21m**	**99**
Amthane	**50.25m**	**29.00m**	**48.05m**	**73**
Panchuvadi	**26.60m**	**14.70m**	**22.30m**	**43**

Preprocessing tha Dataset:

- **Data cleaning:**

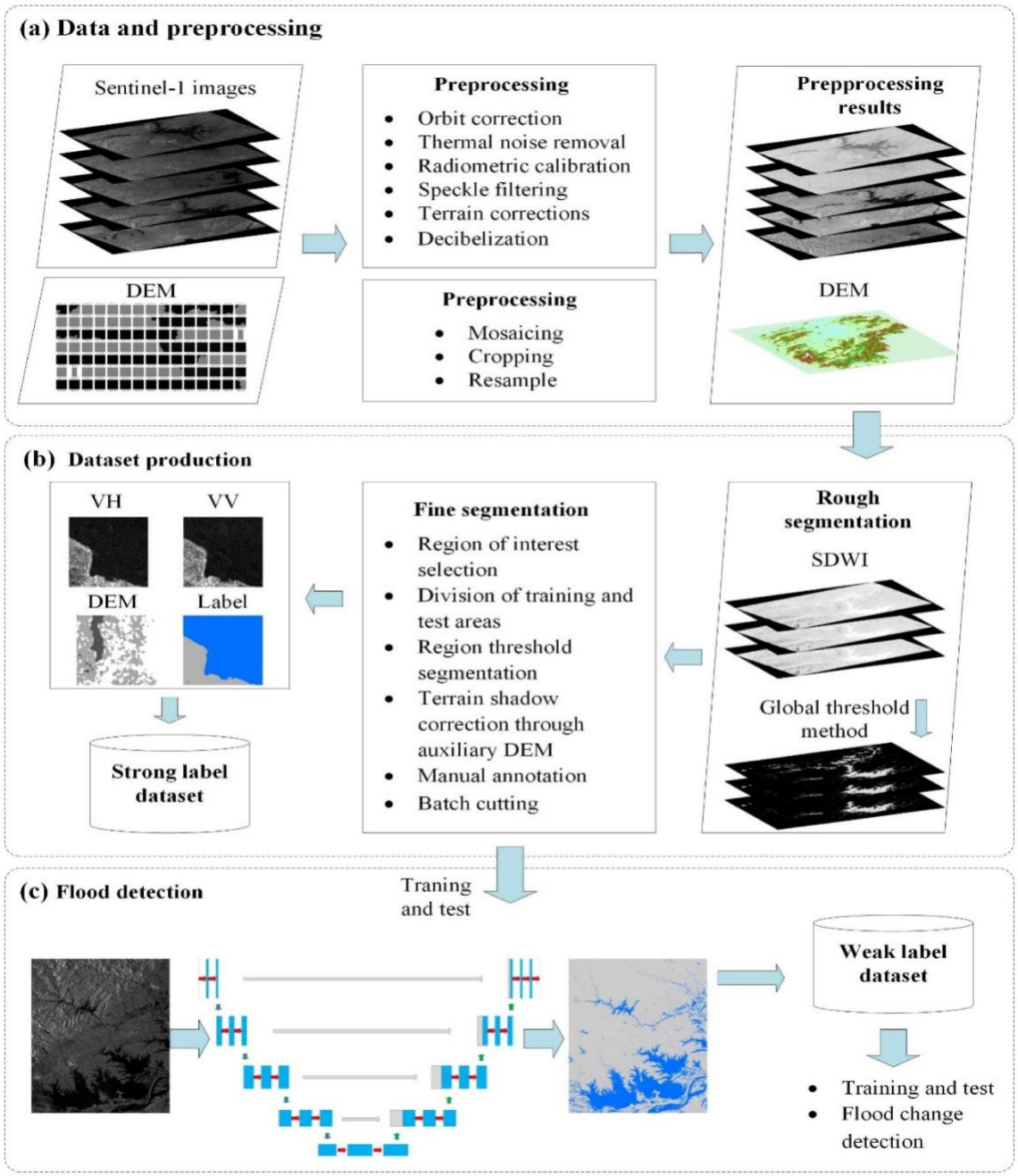
This involves removing any errors or inconsistencies from the data. For example, this may involve correcting typos in the data, removing duplicate records, and filling in missing values.

- **Feature engineering**:

This involves creating new features from the existing data that may be more informative for the machine learning algorithms. For example, this may involve creating features that represent the rate of change of different variables or the difference between variables at different times or locations.

- **Data normalization:**

This involves transforming the data so that all of the features are on the same scale. This is important to ensure



that the machine learning algorithms do not give more weight to features that are on a larger scale

Program:

```
Import time

Import board

    Import digitalio
    Import paho.mqtt.client as mqtt

    # Define the MQTT broker address
    BROKER_ADDRESS = “localhost”
    # Define the MQTT topic
    TOPIC = “flood/alert”

    # Set up the digital pin for the water sensor
    Sensor_pin = digitalio.DigitalInOut(board.D18)
    Sensor_pin.direction = digitalio.Direction.INPUT

    # Create an MQTT client
    Client = mqtt.Client()
    Client.connect(BROKER_ADDRESS)

    # Start a loop to monitor the water sensor
    While True:


    # Read the value of the water sensor
       Sensor_value = sensor_pin.value

       # If the sensor value is high, then the water level is high and there is a risk of flooding
       If sensor_value == digitalio.HIGH:

          # Publish a flood alert message to the MQTT topic
          Client.publish(TOPIC, “Flood alert!”)

          # Print a message to the console
          Print(“Flood alert!”)

       # Wait for 1 second before checking the sensor again
       Time.sleep(1)

    # Disconnect from the MQTT broker
    Client.disconnect()
```

Output:

Flood alert**!**

- **CONCLUSION:**

Flood monitoring systems are an important tool for managing flood risk and reducing the impacts of flooding. As technology continues to advance, flood monitoring systems are becoming more sophisticated and affordable, making them more accessible to a wider range of communities

THANK YOU