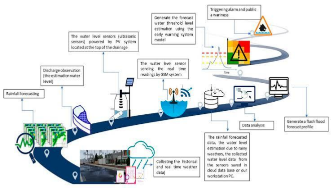
Project Objectives:

1. Flood Monitoring: The primary objective of the project is to develop a real-time flood monitoring and early warning system to detect and predict flooding events in a specific geographic area.

2. Public Safety: Enhance public safety by providing timely flood alerts and evacuation recommendations to residents in flood-prone regions.

3.Emergency Response Coordination : Improve coordination among emergency response agencies by providing them with real-time flood data, enabling quicker and more effective response.

Diagram


Triggering alarm and public a wariness
Generate the forecast water threshold level estimation using the early warning system model
The water level sensors (ultrasonic sensors) powered by PV system located at the top of the drainage
The water level sensor sending the real time readings by GSM system
Discharge observation (the estimation water level)
Rainfall forecasting
Time
Data analysis
Generate a flash flood forecast profile
The rainfall forecasted data, the water level estimation due to rainy weathers, the collected water level data from the sensors saved in cloud data base or our workstation PC.
Collecting the historical and real time weather data)

IoT Sensor Deployment :

1. Site Selection :
- Identify flood-prone areas based on historical data and local knowledge.

2. Sensor Selection:
- Choose sensors for water level monitoring, rainfall measurement, weather conditions, and any other relevant data.

3. Sensor Placement:
- Install sensors at strategic locations, such as near water bodies, floodplains, open areas, and within the soil.

4. Power Supply:
- Ensure a stable power supply for the sensors, which could include batteries, solar panels, or wired connections.

5. Data Transmission:
- Use appropriate communication protocols to transmit data from the sensors to a central server.

6. Data Storage:
- Set up a central server or cloud-based platform for data reception and storage.

7. Data Analysis and Processing:
- Develop algorithms for real-time data analysis and detection of flood-related conditions.

8. Early Warning System:
- Implement an alert system to notify relevant authorities and the public in case of potential flood risks.

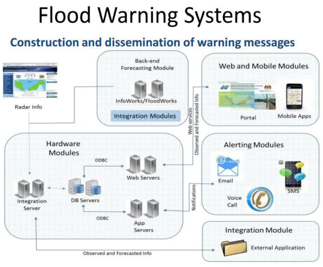
9. User Interface:
- Create a user-friendly interface for data visualization and alert access.

10. Testing and Calibration:
- Regularly test and calibrate the sensors to maintain their accuracy.

11. Community Awareness:
- Educate local communities about the deployment and alert system.

12. Emergency Response Coordination :
- Establish protocols for emergency response teams based on the sensor data.

ARCHITECTURE :



Communication Protocols:

The choice of communication protocols depends on factors such as sensor type, data volume, and range. Here are some commonly used communication protocols:

1. Wi-Fi: Sensors equipped with Wi-Fi modules can transmit data over local Wi-Fi networks, suitable for relatively short distances, like within urban areas or near Wi-Fi hotspots.

2**. Cellular Networks:** Sensors with cellular modules can transmit data over 2G, 3G, 4G, or 5G networks. Cellular communication provides broad coverage and is suitable for remote areas but may require a subscription plan.

3. LoRaWAN : LoRaWAN (Long Range Wide Area Network) is a low-power, long-range wireless protocol suitable for IoT applications. It's cost-effective and can transmit data over several kilometers.

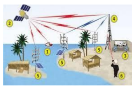
4. Satellite Communication : In remote areas with limited terrestrial network coverage, sensors equipped with satellite communication modules can transmit data via satellite. This ensures wide coverage but can be more expensive.

5. Mesh Networks : In some cases, a mesh network can be created where sensors communicate with one another, and the data is relayed to a central server. This approach can extend the range of the network.

6. Bluetooth : For short-range communication between sensors and a gateway device, Bluetooth can be used. It's suitable for applications with sensors in close proximity.

The choice of protocol should take into account factors like range, power consumption, cost, and available infrastructure. In a flood monitoring system, reliability and real-time data transmission are critical, as timely alerts are essential for public safety and emergency response coordination.

Diagram



.

Platform Development:

1. Data Ingestion :
- Implement data ingestion mechanisms to receive real-time data from IoT sensors. This can be achieved through APIs, data feeds, or message queues.
- Ensure data validation and error handling to maintain data integrity.

2. Database Storage :
- Create a database system to store incoming sensor data. Use a relational or NoSQL database, depending on the data structure and volume.
- Design a schema that accommodates different sensor types and data parameters.

3. Data Processing :
- Develop backend services that process incoming data. This includes data analysis, normalization, and quality checks.
- Implement algorithms to detect anomalies, such as sudden changes in water levels or heavy rainfall.

4. Real-Time Monitoring :
- Build a real-time monitoring dashboard using web technologies like HTML, CSS, and JavaScript. Use frameworks like React, Angular, or Vue.js for interactivity.
- Visualize data using charts, maps, and other graphical representations for easy comprehension.

5. Alert Generation:
- Implement an alert system that continuously monitors sensor data. When specific conditions, like abnormal water level increases, are detected, trigger alerts.
- Send alerts through various communication channels, such as SMS, email, or automated phone calls. Ensure the system prioritizes and logs alerts.

6. User Interface:

- Create user-friendly interfaces accessible via web browsers or mobile apps. These interfaces provide a real-time view of sensor data and alerts.
- Allow users to customize their dashboard, set alert preferences, and access historical data.

7. Historical Data Storage :

- Archive sensor data in a separate database or data warehouse for historical analysis and reporting.
- Implement data retention policies to manage the storage of historical data efficiently.

8. Data Analysis Tools :

- Develop tools for historical data analysis, which could include generating reports, identifying trends, and running simulations.
- Use data analytics and visualization libraries like D3.js for detailed insights.

9. Security and Access Control:

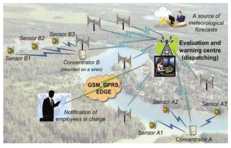
- Implement robust security measures to protect the platform from unauthorized access and data breaches.
- Control user access and permissions to ensure data privacy and security.

Platform Operation :

The web-based platform continually receives data from IoT sensors, processes it in real-time, and visualizes it for users. When critical conditions indicative of potential floods are detected, the system generates alerts and notifies relevant authorities and the public. Historical data is stored for later analysis and reporting, contributing to a better understanding of flood patterns and risk mitigation.

Overall, this platform ensures that all stakeholders, including emergency response teams and the general public, have access to real-time and historical data, enabling informed decision-making and timely response to flood events. It enhances public safety and response coordination by providing accurate and up-to-date information.

DIAGRAM :


A source of meteorological forecasts
Sensor B3
Sensor B2
Sensor B1
Evaluation and warning centre (dispatching)
Concentrator B (mounted on a siren)
GSM, GPRS, EDGE
Notification of employees in charge
Sensor A2
Sensor A3
Sensor A1
Concentrator A

Code Implementation :

The implementation of an IoT-based flood monitoring and early warning system involves several programming languages and technologies across different components of the system. Here's an overview of the key technologies used for each aspect of the project:

1. IoT Sensor Communication:

Programming Languages: C/C++, Python, or JavaScript (for sensor firmware)

Technologies: MQTT, HTTP, LoRaWAN, Cellular, Wi-Fi, or Bluetooth for data transmission

MQTT

python

```
    import paho.mqtt.client as mqtt

def on_connect(client, userdata, flags, rc):
    print("Connected with result code "+str(rc))
    client.subscribe("sensors/flood_monitoring_data")

def on_message(client, userdata, msg):
    # Process and store incoming sensor data
    process_sensor_data(msg.payload)

client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message
client.connect("mqtt.broker.com", 1883, 60)
client.loop_forever()
```

2. Data Processing:

Programming Languages: Python, JavaScript, or Java.

Technologies: Pandas, NumPy, TensorFlow, or scikit-learn for data analysis and machine learning

(Simple Data Analysis in Python)

``

python

```
    import pandas as pd

    # Load sensor data into a DataFrame
    data = pd.read_csv("sensor_data.csv")

    # Calculate average water level
    avg_water_level = data['water_level'].mean()
```

3. Web-Based Platform:

- **Programming Languages:** HTML, CSS, JavaScript (front-end), Python, Node.js, or Java (back-end)

- **Technologies**: React, Angular, or Vue.js (for front-end), Flask, Express.js, or Django (for back-end)

(**Node.js Server in JavaScript)**

javascript

~~~
    const express = require('express');
    const app = express();
    const port = 3000;

    app.get('/', (req, res) => {
        res.send('Welcome to the Flood Monitoring Platform');
    });

    app.listen(port, () => {
        console.log(`Server is running on port ${port}`);
    });
    ```
~~~

4. Flood Prediction Models:

Programming Languages: Python, R

Technologies:Machine learning libraries like scikit-learn or TensorFlow

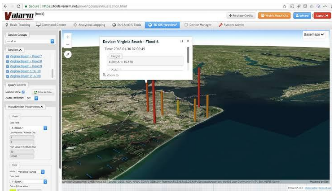
(**Random Forest Classifier in Python**):

Python

```
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.model_selection import train_test_split
    from sklearn.metrics import accuracy_score

    # Split the dataset into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # Create a random forest classifier
    clf = RandomForestClassifier(n_estimators=100, random_state=42)

    # Fit the model to the training data
    clf.fit(X_train, y_train)

    # Make predictions on the testing data
    predictions = clf.predict(X_test)
```

```
    # Calculate accuracy
    accuracy = accuracy_score(y_test, predictions)
```

    ``



Device: Virginia Beach - Flood 6

Time: 2018-01-30 07:00:49

Height

4-20mA 1: 15.678

Latest only: ☑

Auto-Refresh: Off

Data field: 4-20mA 1

Low Value In / Altitude Out: 4, 4

High Value In / Altitude Out: 20, 10000

Mode: Variable Range

Data field: 4-20mA 1

Enhancing Public Safety and Emergency Response Coordination :

Providing Early Warnings to Residents:

Real-time flood monitoring offers early warnings to residents through various communication channels, ensuring that they receive timely and potentially life-saving information. Here's how these alerts are distributed:

1. Data Collection and Analysis:
IoT sensors continuously collect data on parameters like water levels, rainfall, and weather conditions. Real-time analysis is conducted to detect potential flood conditions.

2. Alert Generation:
When the system's algorithms identify conditions indicative of a flood event (e.g., rapid water level rise), an alert is generated.

3. Communication Channels:

a. SMS (Short Message Service): Residents in the affected area receive text messages warning them of the potential flood.

b. Mobile Apps: Dedicated mobile apps send push notifications to users with real-time updates and alerts.

c. Sirens and Public Address Systems:
In areas with loudspeakers or sirens, alerts are broadcast, and residents are urged to take immediate action.

d. Email: Registered users receive email alerts with instructions and safety information.

e. Automated Phone Calls (Robocalls): Automated phone calls are made to landline and mobile numbers in the affected area, delivering recorded messages with flood warnings and safety guidance.

4. User Response: Residents who receive alerts are encouraged to follow safety protocols, which may include evacuating to safer areas, staying informed about flood conditions, and being prepared for potential flooding.

Emergency Response Agencies' Access to the Platform:

Emergency response agencies play a vital role in managing flood events. They can access the flood monitoring platform to receive real-time data, helping them make informed decisions and allocate resources effectively:

1. Access Credentials: Authorized personnel from emergency response agencies are provided with access credentials to log into the platform.

2. Real-Time Data: The platform provides a live feed of sensor data, alert status, and real-time monitoring visuals. Agencies can view water levels, rainfall, weather conditions, and any alerts that have been triggered.

3. Map Visualization: The platform often includes map visualizations that display flood-prone areas and the status of sensors. This aids agencies in identifying critical zones.

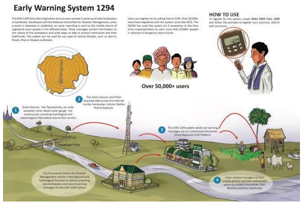
4. Data Analytics Tools: Agencies can access tools for data analysis and historical data retrieval. This helps them assess the severity of the flood event and understand past patterns.

5. Alert Management: Emergency response agencies can manage and prioritize alerts, ensuring that they respond promptly to critical conditions. They can also disseminate information to the public through various channels.

6. Resource Allocation: The real-time data enables agencies to allocate resources effectively. For example, they can deploy personnel, rescue teams, and equipment to areas at high risk of flooding.

7. Coordination: Multiple agencies, including police, fire departments, and disaster management agencies, can coordinate their efforts based on the data provided by the platform.

8. Public Communication: Agencies can use the platform to communicate with the public, issue evacuation orders, and provide safety information.



Early Warning System 1294

The EWS 1294 Early Warning System aims to warn people in advance of natural disasters in Cambodia. Developed with the National Committee for Disaster Management, when a event is detected or predicted, an voice recording is sent to the mobile phone of registered users people in the affected areas. These messages contain information on the nature of the emergency and what steps to take to protect themselves and their livelihoods. The system can be used for any type of natural disaster, such as storms, floods, fires or disease outbreaks.

Users can register for by calling free to 1294. Over 50,000+ users have registered with the system since late 2015. The NCDM has used the system on 3 occasions in the time since implementation to warn more that 20,000+ people in advance of dangerous storm fronts.

HOW TO USE

To register for the system, simply **DIAL FREE CALL 1294** and follow the prompts to register your province, district and commune.

1 Smart Sensors - like Tepmachcha, our solar powered, sonar based water gauge - are continuously recording hydrological and meterological information around the country

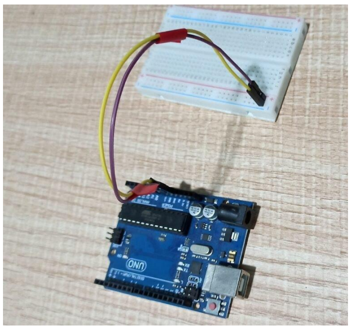
2 The smart sensors send their acquired data across the Internet via the Cambodian Cellular Mobile Phone Network.

3 This EWS 1294 system sends out warning messages via our customised Interactive Voice Repsonse (IVR) Platform

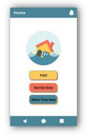
4 Users receive messages via their mobile phones and take appropriate action to protect themselves, their families and their livelihoods

The Provincial Centres for Disaster Management monitor meterological and hydrological forecasts to detect oncoming natural disasters and record warning messages for the EWS 1294 System

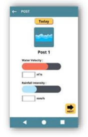
SENSOR :



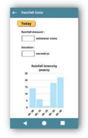
PLATFORM UI :



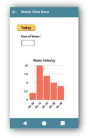
(a)



(b)



(c)



(d)