# Flood Monitoring and Early Warning System

**Problem definition :**

The aim of this project is to address these issues and develop a “Flood monitoring and early warning system “ solution that leverages technology to enhance the accessibility ,warning and monitoring of flood.

**Malfunctions :**

* Data Inaccuracies
* False Alarms
* Missed Warnings
* Communication Failures
* Community Engagement Issues
* System Downtime
* Resource Constraints
* Obsolete Technology
* Resistance to Change

**Algorithm: Flood Monitoring and Early Warning System**

**Inputs:**

* Real-time weather data (rainfall, temperature, wind speed, etc.).
* Real-time river and water level data.
* Geographic information (maps, terrain data, and land use).
* Historical flood data.

**Outputs:**

* Early flood warnings and alerts.

**Algorithm Steps:**

1. **Data Acquisition:**
   * Gather real-time weather data from weather stations or sources like the National Weather Service.
   * Collect real-time river and water level data from monitoring stations.
   * Acquire geographic information and land use data to understand local terrain.
   * Retrieve and update historical flood data.
2. **Preprocessing:**
   * Clean and preprocess the acquired data to remove noise and errors.
   * Convert data units (e.g., inches of rainfall to millimeters) for consistency.
   * Aggregate data over relevant time intervals (e.g., hourly or daily).
3. **Hydrological Modeling:**
   * Calculate rainfall runoff using rainfall data and land characteristics.
   * Predict river and water levels using hydrological models.
   * Analyze the potential flood extent and impact based on these predictions.
4. **Threshold Determination:**
   * Set thresholds for river levels and rainfall intensity that trigger warnings based on historical data and local conditions.
5. **Real-Time Monitoring:**
   * Continuously monitor real-time weather and river data.
   * Compare current data with established thresholds.
6. **Early Warning Generation:**
   * If data exceeds defined thresholds, generate flood alerts.
   * Determine the severity of the alert based on the deviation from thresholds.
7. **Alert Dissemination:**
   * Disseminate alerts to relevant authorities, emergency services, and the community through multiple channels (e.g., SMS, sirens, mobile apps, social media, and local radio).
   * Prioritize alerting based on the severity of the alert.
8. **Community Engagement:**
   * Engage with the community by providing information and educational material about flood risks.
   * Encourage the community to report flood-related incidents and concerns.
9. **Response Planning:**
   * Collaborate with local authorities to develop evacuation plans and flood response strategies.
   * Ensure the availability of evacuation routes and shelters.
10. **Monitoring and Evaluation:**
    * Continuously assess the system's performance and accuracy.
    * Collect feedback from the community and relevant stakeholders.
    * Make necessary adjustments to improve the system.
11. **Public Outreach:**
    * Regularly inform the public about the status of flood monitoring and early warning efforts.
    * Promote a culture of preparedness and provide guidance on what to do in case of a flood warning.

**CODING:**

import random

import time

# Simulated river level monitoring function

def monitor\_river\_level():

return random.uniform(0, 10) # Simulate river level between 0 and 10 meters

# Function to issue a flood warning

def issue\_flood\_warning(severity):

print(f"Flood warning - Severity: {severity}")

# Threshold values for flood warnings (adjust as needed)

threshold\_low = 2.0 # Low flood risk

threshold\_medium = 5.0 # Moderate flood risk

threshold\_high = 7.5 # High flood risk

while True:

# Monitor the river level

current\_river\_level = monitor\_river\_level()

# Check if river level exceeds the thresholds

if current\_river\_level >= threshold\_high:

issue\_flood\_warning("High")

elif current\_river\_level >= threshold\_medium:

issue\_flood\_warning("Moderate")

elif current\_river\_level >= threshold\_low:

issue\_flood\_warning("Low")

# Pause for a set interval (simulating real-time monitoring)

time.sleep(60) # Wait for 60 seconds before the next reading.

**Flow chart:**

Ultrasonic sensor and sends sonic pulse reflected from water in a place that require observation

Ultra sonic sensor receives sonic pulse reflected from water level

Water level at dangerous level

Water level at safe level

Liquid level?

HIGH LOW

MEDIUM

System sends alert to every local at every minute

System sends to every local every five minutes.

Alarm sounded

System sends alert to locals that request alerts

Indicator turns yellow

Indicator turns green

Water level cautious level

Indicator turns red