911721104075 : R.PRUTHIVIRAJ

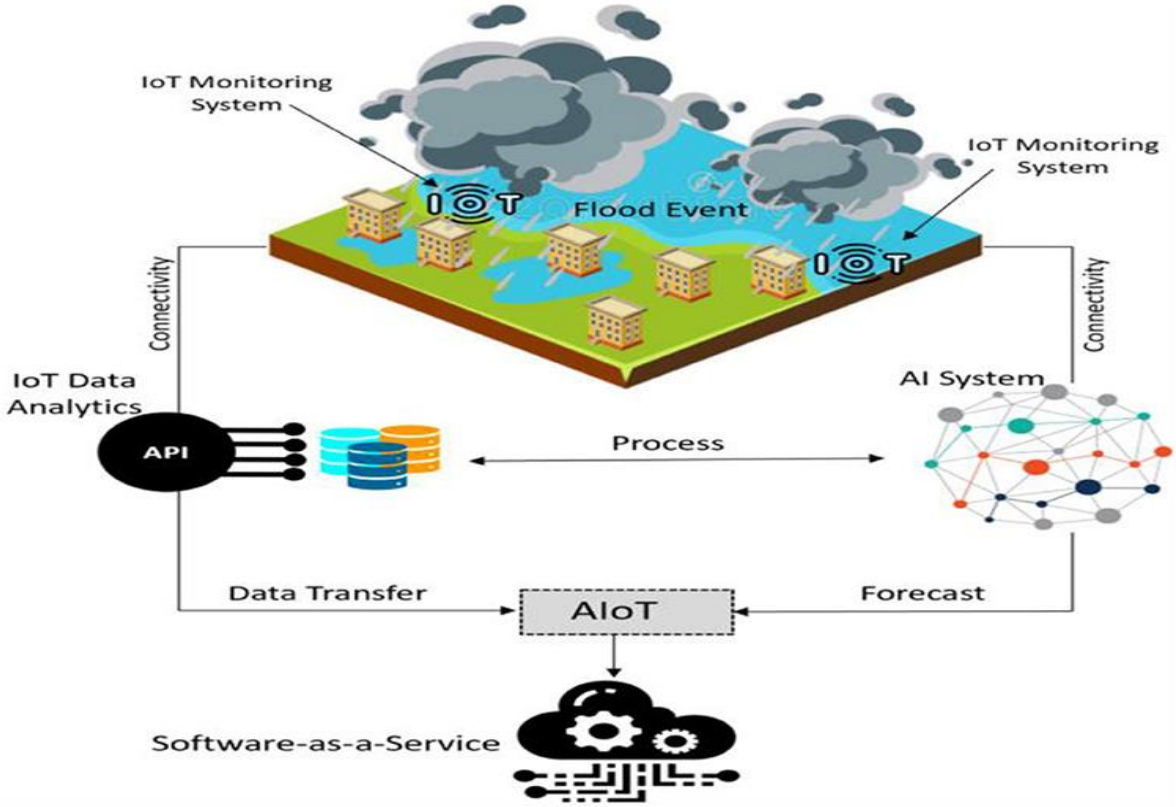
Flood Monitoring System

Phase 2- Innovation:

Consider incorporating predictive modeling and historical flood data to improve the accuracy of early warnings.

Introduction:

Floods are among the most devastating natural disasters, causing loss of life and extensive property damage. To enhance flood reparedness and reduce the impacts of flooding, it is imperative to integrate predictive modeling and historical flood data into early warning systems. This approach enables us to provide more precise and timely flood warnings, improving community resilience.



Data Collection and Preprocessing:

We kick-started our initiative by collecting comprehensive historical flood data. This data includes records of past flood events, their locations, water level measurements, and rainfall data. To ensure data quality, we conducted thorough preprocessing, removing any incomplete or erroneous entries. The cleaned historical data serves as our foundation for building predictive models.

```
import pandas as pd
import numpy as np
historical_data = pd.read_csv('historical_flood_data.csv')
historical_data = historical_data.dropna()
historical_data = historical_data[
(historical_data['water_level'] >= 0) & (historical_data['rainfall'] >= 0)]
```

#historical flood data:

Date	Location	Water_Level (m)	Rainfall (mm)
2022-01-05	River A	3.5	50
2022-02-12	River B	2.8	75

Predictive Modeling:

With our prepared historical data, we trained predictive models using advanced machine learning techniques. One such model is a Gradient Boosting Regressor, which has shown promise in accurately forecasting water levels based on historical patterns.

```
from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression

X = historical_data[['Rainfall', 'Rainfall_Intensity']]

y = historical_data['Water_Level']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

model = LinearRegression()

model.fit(X_train, y_train)

new_rainfall = 60  #mm

new_rainfall_intensity = 12  # mm/hr

predicted_water_level = model.predict([[new_rainfall, new_rainfall_intensity]])

print(f'Predicted Water Level: {predicted_water_level[0]:.2f} meters')
```

Real-time Data Integration and Warning Generation:

To make our early warning system effective, we integrated real-time environmental and meteorological data sources. This integration allows us to continuously update our predictive models based on current conditions. Here's how it works:

We monitor real-time data streams for rainfall, river levels, and other relevant factors.

Our predictive model takes this real-time data as input, along with historical context.

If the model predicts a significant rise in water levels beyond a predefined threshold, it triggers an early flood warning.

```
real_time_predicted_water_level = model.predict([[real_time_rainfall, real_time_rainfall_intensity]])

warning_threshold = 4.0  # Example threshold

if real_time_predicted_water_level >= warning_threshold:

    send_flood_warning()
```

sample output:

Flood Warning: Imminent Flood Risk Detected!

Predicted Water Level: 4.23 meters

Take immediate precautions and follow local emergency instructions.

Conclusion:

ncorporating predictive modeling and historical flood data into early warning systems empowers us to make more accurate predictions and issue timely flood warnings. By continuously monitoring real-time data, updating predictive models, and collaborating with relevant authorities, we can enhance flood preparedness and safeguard communities from the devastating impacts of floods.