INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION SECTOR

STUDY PERIOD 2022-2024

FOCUS GROUP ON AI NATIVE NETWORKS

AINN-I-xx

Original: English

Question(s): N/A | Virtual, TBD 2024

INPUT DOCUMENT

Source:	*Vajra IITB*
Title:	*Team Vajra IITB - Report on ITU WTSA Hackathon 2024 – Flood Monitoring and Alerting System due to Heavy Rainfall in Urban Areas*

Contact:	Manjesh Hanawal Kushal Pawaskar	E-mail: mhanawal@iitb.ac.in kushalpawaskar02@gmail.com

Abstract: This document contains the submission of a report for Team Vajra IITB towards ITU WTSA Hackathon 2024 for the use case *Flood Monitoring and Alerting System due to Heavy Rainfall in Urban Areas*

Use case introduction: "Flood Monitoring and Alerting System due to Heavy Rainfall in Urban Areas"

In urban areas, sudden, heavy rainfall can cause severe waterlogging and flooding, resulting in potential damage to property and loss of life. This situation requires immediate action and response from both authorities and residents to mitigate the damage. To this end, a rainfall forecasting model is required that can predict extreme rainfall with a granularity of 15 minutes to 1 hour in advance. Mobile systems play an important role to provide such services/features. Upon detecting an extreme weather event, the system triggers emergency alerts that are sent to the residents, emergency responders, and relevant authorities using the Public Warning Systems. In mobile networks, network resource allocation in the affected regions can be increased for better communication of emergency responders and residents when rescue operations are being carried out.

It is the season of monsoon. The city of Mumbai is buzzing with its activities like usual. Suddenly, a rainfall forecasting model, which predicts rainfall in several regions of Mumbai forecasts extreme rainfall in some regions in the coming minutes, which are thus likely to get waterlogged. An emergency warning system, based on the model's forecast, sends alerts to the residents to either avoid the region, if possible, or to be prepared for a possible waterlogging scenario. It also sends alerts to the relevant authorities and emergency responders so that they can prepare to tackle the harsh situation. The alerts are sent through the 5G network to ensure that they reach the destination as soon as possible. Network resource allocation in the region to be affected is increased to ensure that communication between relevant parties is not a bottleneck during the emergency.

Consider the scene map below:

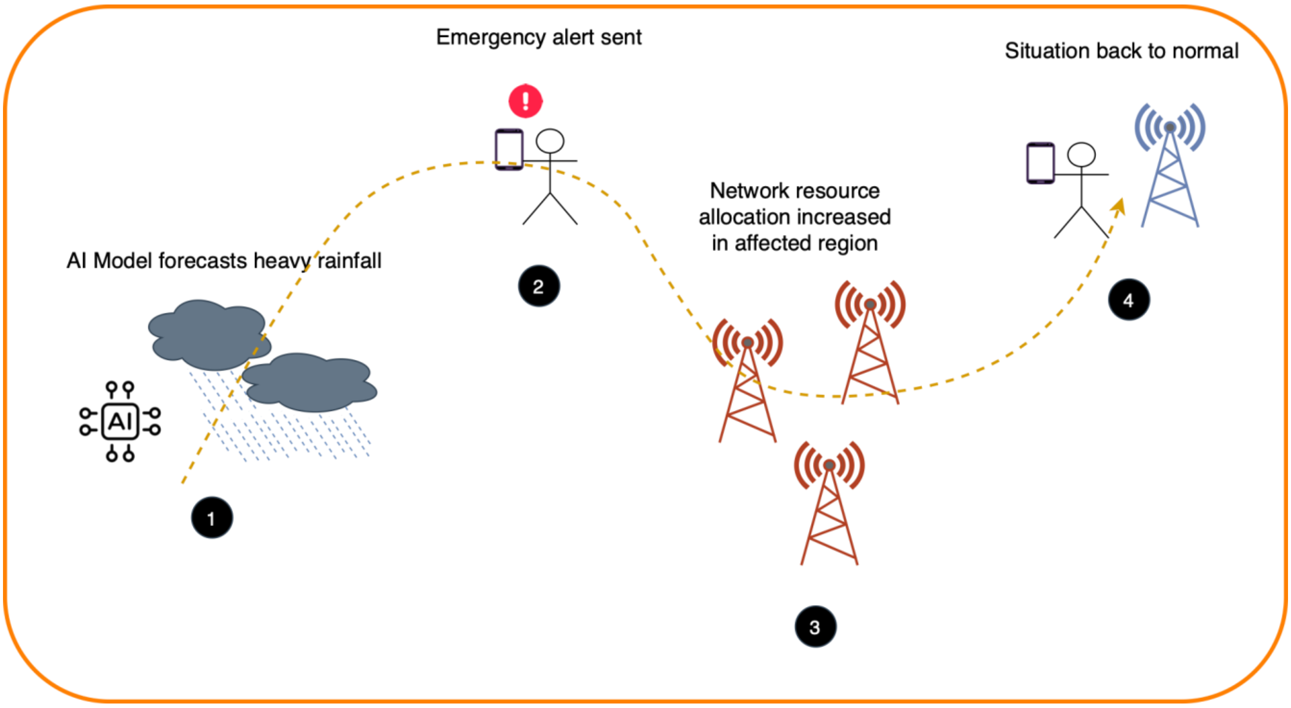


Figure 1

Phase 1: "*Forecasting heavy rainfall*": Based on the forecast data available from Global Forecasting System (GFS) over a huge region in coarse intervals, and the observed historical data from local weather stations, rainfall for next 15 minutes to next 1 hour predicted using machine learning models.

Phase 2: "*Sending emergency alerts*": System sends emergency alerts to residents in affected and surrounding areas via Public Warning System (PWS) support in the 5G networks. Emergency responders and other relevant authorities are also notified about the same.

Phase 3: "*Resource allocation increase in affected areas*": The affected regions allocate dedicated network resources to make sure that important communication regarding emergency response is not hindered and all the operations are carried out effectively in the duration of extreme rainfall

Phase 4: "*Emergency situation resolved*": Eventually, the situation resolves as rainfall slows down. The forecasting system stops sending alert and network resource allocation is brought back to normal.

UN SDG goals:

Our use case aligns with Goal 11: Make cities and human settlements inclusive, safe, resilient and sustainable under the topic Disaster risk reduction (DRR).

Justification: The UN Commission on Sustainable Development (UNCSD) addressed risk management and vulnerability in the context of its thematic issues of water, sanitation and human settlements in its 2004-2005 cycle and then in the context of drought and desertification in its 2006-2007 cycle. On the occasion of World Water Day 2004, Guidelines for Reducing Flood Losses was launched. This inter-agency publication, led by Department of Economic and Social Affairs (DESA) aimed at providing decision-makers with a range of options to consider for reducing losses associated with flooding.

Our project gives a mechanism for authorities and responders (decision makers) to act quickly in the event of flooding in urban areas. It also leverages the existing mechanisms provided by the mobile network systems through Public Warning System (PWS) to alert the people in the affected areas. This helps in making cities resilient and sustainable by reducing the losses due to disaster like flooding due to heavy rainfall.

Use case requirements

Requirement-1: It is critical that rainfall forecasting models are deployed on the edge or cloud, enabling low-latency processing of weather data and issuing accurate, near real-time predictions.

Requirement-2: It is critical that emergency alerts through Public Warning System are distributed to people in affected regions with minimal delay, ensuring timely response to the predicted rainfall.

Requirement-3: It is critical that dedicated communication resources are allocated to authorities and emergency responders based on geographical location to effectively manage rescue operations. This can be enabled through a dedicated network slice.

Requirement-4: It is critical that after the extreme weather event is over, the system stops sending alerts and high priority resource allocation is brought back to normal.

PS1: Pipeline design

- **AI /ML:** Concept used is time series forecasting, anomaly detection
- **Figure 2 is in Relation with ITU Y.3172**

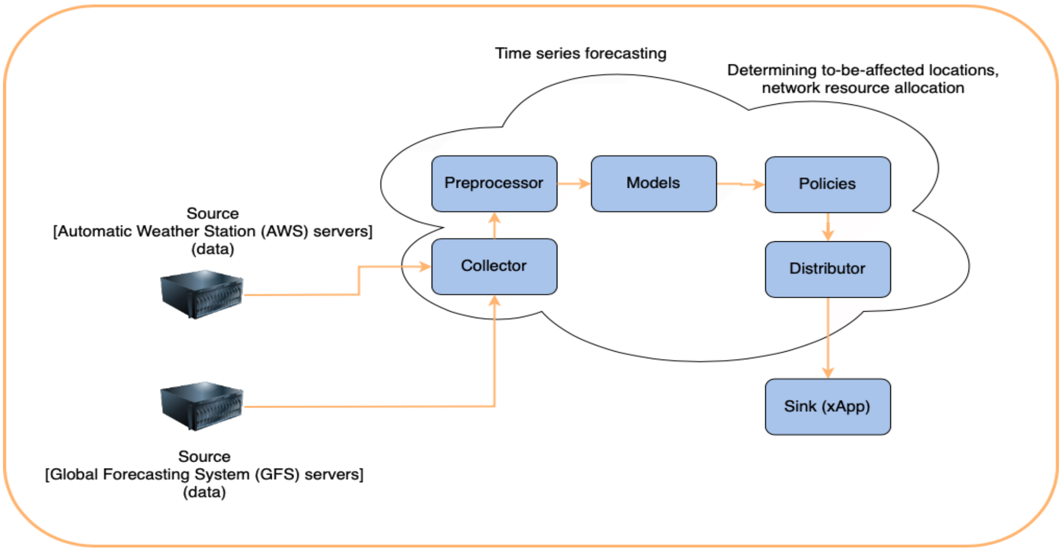


Figure 2

Requirements for this type of application?

- **SRC of data:** Global Forecasting System (GFS) forecast, analysis data; Automatic Weather Station (AWS) observed rainfall data. This data comes from webpages that make them available periodically. The screenshots of the web pages are in Figure 3 and Figure 4. GFS data provides weather data (Precipitation rate, precipitable water, relative humidity, total cloud cover) over grids of size 0.25º x 0.25º latitude, longitude and is made available every 6 hours with granularity of 3 hours. AWS data provides ground observations of actual rainfall and is available every 15 minutes.
- **Collector**: Edge or cloud. Collects data from the GFS web server and AWS web server.
- **Preprocessor**: Edge or cloud. Preprocesses collected data to make it suitable for the models.
- **Models:** Time series forecasting (cloud) [Linear Regression, Dense Neural Networks]. Predicts rainfall in next 15 minutes based on previous 1 hour of observed rainfall from AWS data and forecasts from GFS data.
- **Policies:** Determining to-be-affected locations for network resource allocation (cloud). Determines the locations based on threshold on predicted rainfall so that alerts can be sent to the residents of those regions and dedicated network resource allocation can be done.
- **Distributor :** Edge or cloud. Distributes the policy information to the xApp.
- **Sink:** xApp. Sends E2 messages to configure emergency alert broadcast and dedicated resource allocation

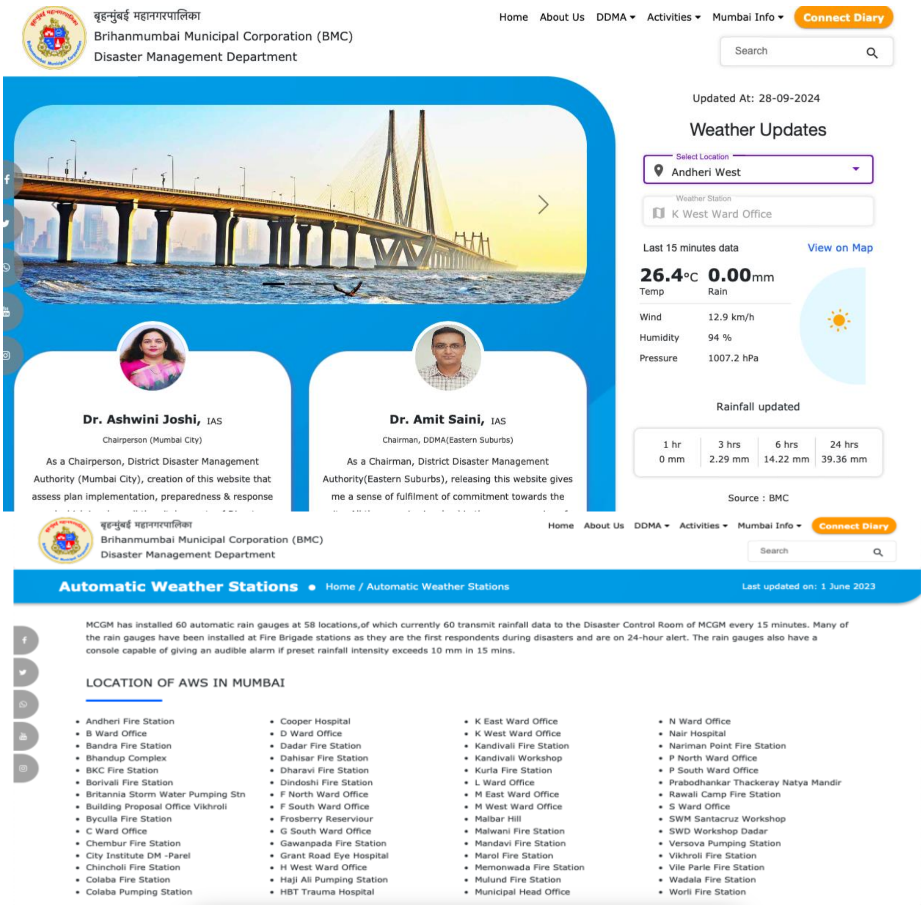


Figure 3: Automatic Weather Stations (AWS) website

https://dm.mcgm.gov.in/auto-weather-station

https://dm.mcgm.gov.in/

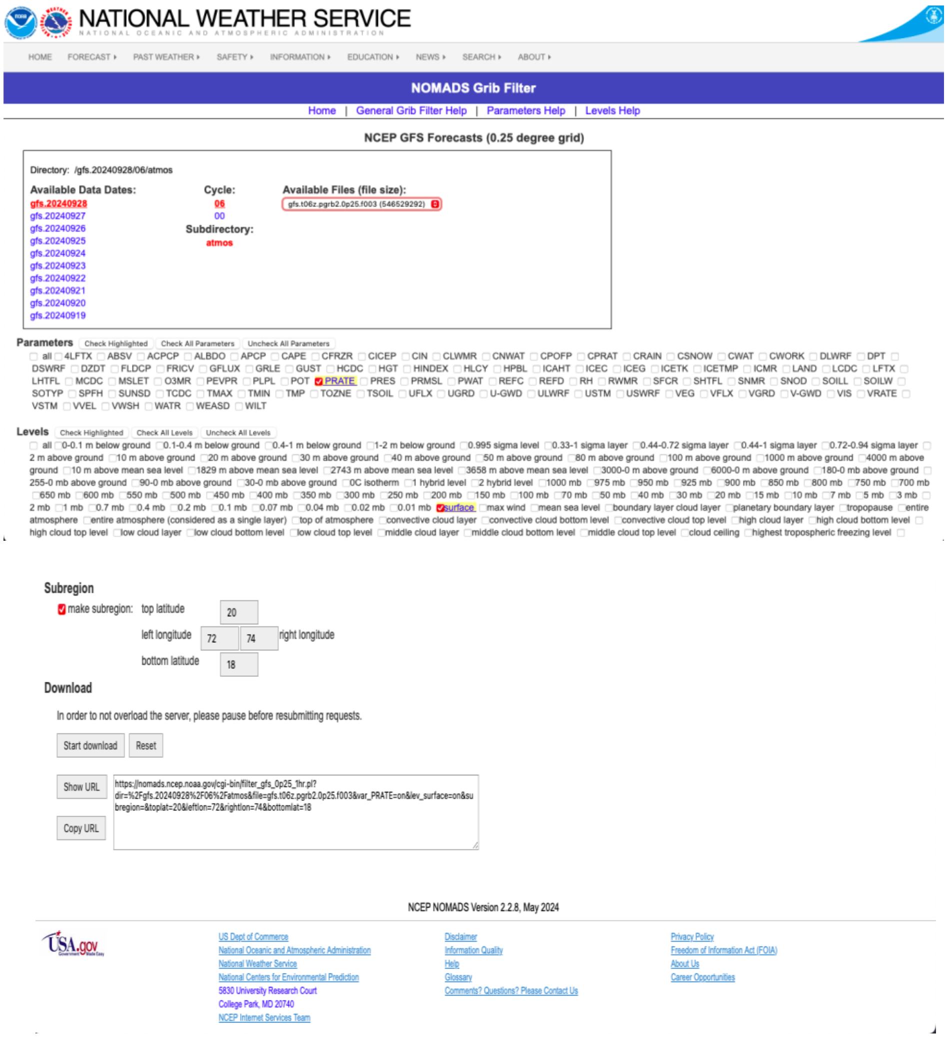


Figure 4: Global Forecasting System(GFS) website

https://nomads.ncep.noaa.gov/gribfilter.php?ds=gfs_0p25_1hr

https://rda.ucar.edu/datasets/d084001/dataaccess/# (for historical data)

PS2: xApp design

- Open RAN concept used is dynamic network resource allocation based on priority

- **What is the role of xApp?**
- The xApp provides a dynamic resource allocation function in the RAN. It enables allocation of dedicated resources through network slicing
- The xApp also configures the base stations of the to-be-affected regions to broadcast emergency messages using Public Warning System
- The resource allocation and broadcasting mechanism needs to be validated before deploying in the field.

- **What is the role of Sandbox?**
- To verify and optimize the xApp based resource allocation, operators may use AI/ML and experiment with several strategies before deploying in the field.

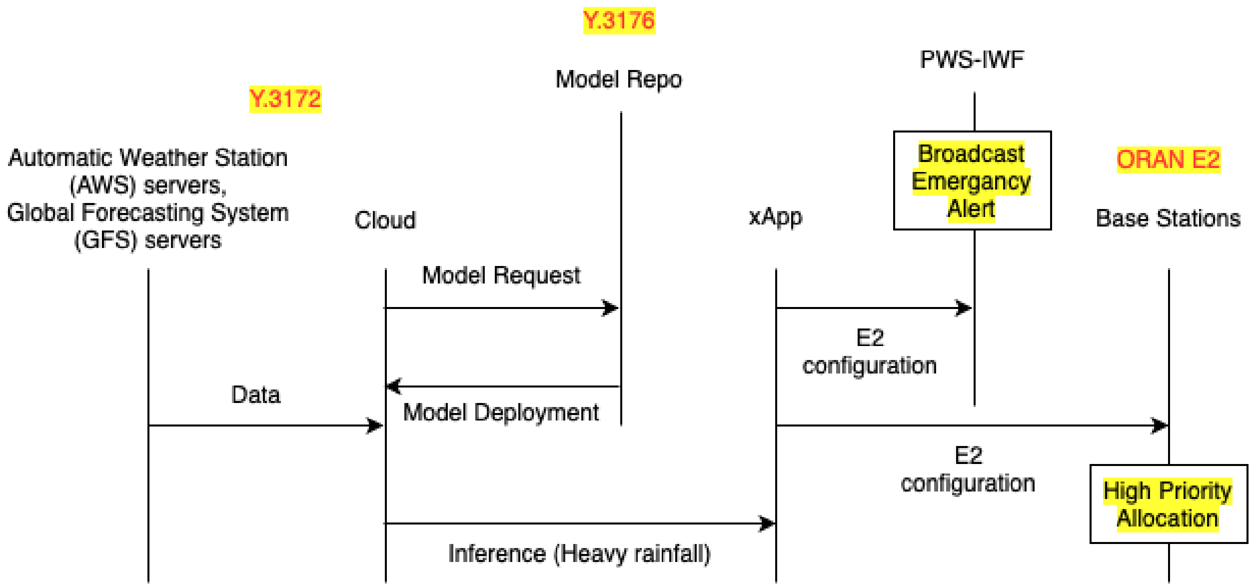
xApp design:

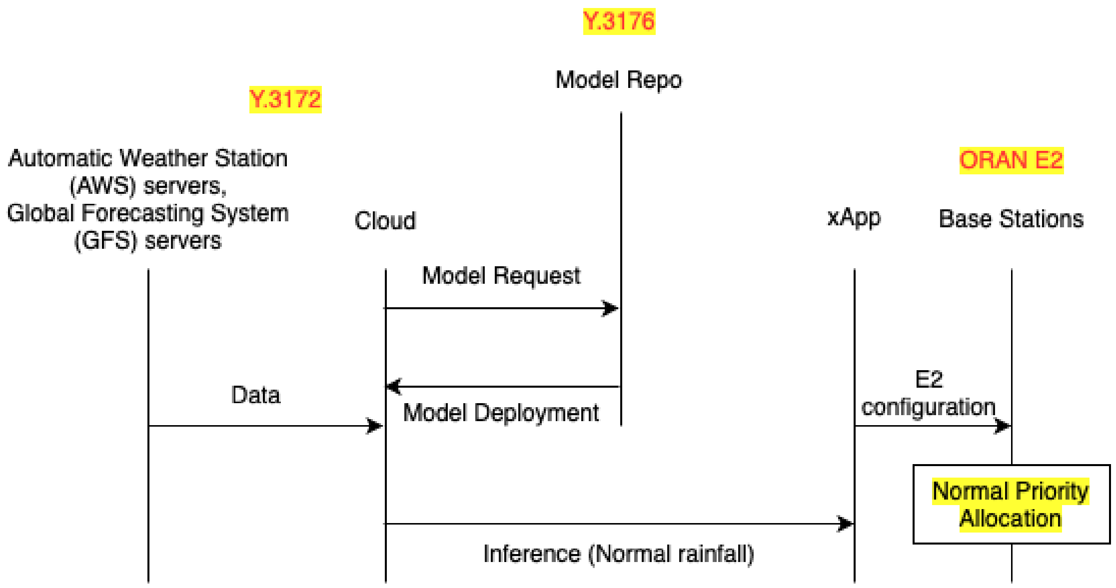
code snippet:

```
# Receive input from distributor node about the to-be-affected regions
# We are observing 36 regions, so input is a 36-dimensional vector. An entry of 1 in the vector denotes that
#  the region is likely to be flooded, whereas 0 denotes that the region is unlikely to be flooded
while True:
    try:
        distributor_conn, distributor_addr = distributor_sock.accept()
        print(f"Connected to distributor node: {distributor_addr}")
        while True:
            data = distributor_conn.recv(1024).decode()
            if not data:
                break
            print("Received from distributor node:", data)
            print("Each number (0/1) corresponds to a particular region")
            print("0 denotes normal rainfall forecasted in next 15 minutes")
            print("1 denotes heavy rainfall forecasted in next 15 minutes")
            status = data.split(" ")
            print()
            print("Decoding and sending commands to respective base stations")
            print()
            assert(len(status) == len(regions))
            for i in range(len(regions)):
                region = regions[i]
                base_stations = get_BS_by_region(region)
                pws_servers = get_PWS_IWF_servers(region)
                if status[i] == "1":
                    send_e2(region, base_stations, 1, cmds[“HIGH_PRIORITY_RESOURCE_ALLOCATION"])
                    send_e2_pws(region, pws_servers, 1, cmds["BROADCAST_MESSAGE"])
                else:
                    send_e2(region, base_stations, 0, [cmds["NORMAL_RESOURCE_ALLOCATION"]])
    except:
        distributor_sock.close()
        break
```

Relation to Standards

Relations to various ITU standards are shown in the following diagrams.

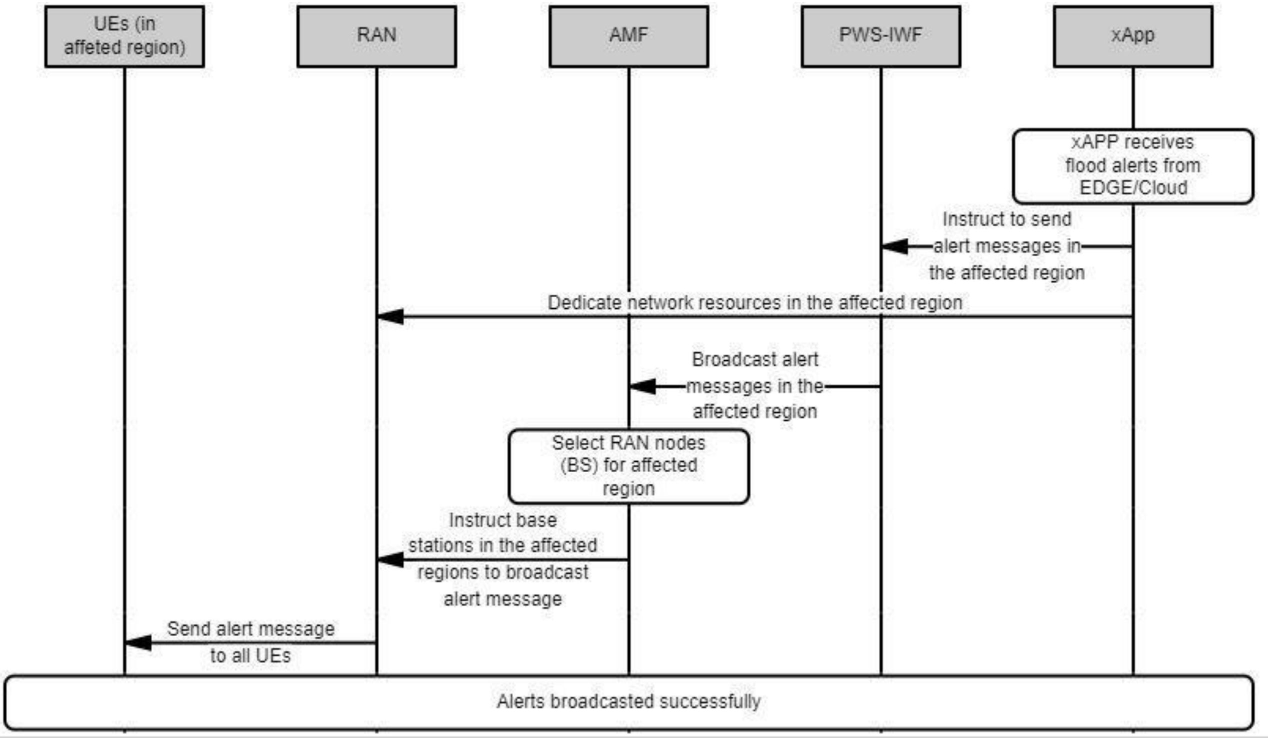




Description: Data is collected from two **Sources**. 1) Automatic Weather Station (AWS) website and 2) Global Forecasting System (GFS) website. This task is performed by the **Collector** node which is located in the Edge/Cloud. The collected data is then sent to the **Preprocessor** node, also located in theEdge/Cloud, which preprocesses the data in a way which would be suitable for model training and inference. The preprocessed data is then sent to the **Model** node (in the Edge/Cloud) which performs forecasting of the rainfall for the next 1 hours at a granularity of 15 minutes. The model for forecasting can be imported from a Model Repository (marketplace), as per **ITU Y.3176**, which stores multiple models. This is done by sending a request for the required model. The marketplace will then return the requested model. The training, if required, and inference will then occur at the Model node.

The forecasted rainfall for multiple regions is then sent to the **Policy** node, which will determine whether a region is likely to be flooded or not, based on some thresholding logic.
This information about which regions are likely to be flooded and which aren't is then sent to the **Distributor** node, which will then send it to the xApp, which is the **Sink** node.
The above pipeline design is as per **ITU Y.3172** standard.
The xApp will then send E2 messages to base stations so that the dedicated network resources can be allocated to the affected regions. The xApp will also manage the broadcasting of emergency alerts to the residents of the affected regions by sending E2 messages to PWS-IWF servers (more information below).

Message Flow in Network



The flow of messages within the networks are as shown in the figure above. Once the xApp receives a flood warning from the distributor, it contacts the Public Warning System Interworking Function (PWS-IWF) or and passes on the information about the warning and the information about the affected regions. It also instructs the RAN to dedicate resources for the BS in the affected region. The PWS-IWF, passes on the information received from the xApp to AMF. AMF prepares a list of RAN nodes in the affected regions and instructs them to broadcast the warning messages. The RAN then sends the alert messages to all the UEs in the affected area. Note that this class is different from Short Message Services which involves other network functions.

Code submission details

1. All the program-related files and dataset information are submitted in the github repository mentioned below
2. Github Repo link: (https://github.com/VajraSecurity/VajraIITB.git)

Self-Testing results

Model Testing:

```
Andheri training score: 0.41152566691120107 validation score: 0.4347087981414317
B ward training score: 0.40314127243359854 validation score: 0.5022416324033716
Bandra training score: 0.41732792907884964 validation score: 0.4741932004211249
Byculla training score: 0.3879058487156294 validation score: 0.5134037518206107
C ward training score: 0.38445528260624295 validation score: 0.5123139955007963
Chembur training score: 0.3761361468698051 validation score: 0.457152215269935
Chincholi training score: 0.42905905330812577 validation score: 0.46487843923527483
Colaba training score: 0.4034061680081563 validation score: 0.49810835462054703
D Ward training score: 0.3733717867382622 validation score: 0.474379231704524
Dahisar training score: 0.43335999227311517 validation score: 0.4575127960597488
Dindoshi training score: 0.4214397406263416 validation score: 0.4524241156828591
F North training score: 0.4096942974252509 validation score: 0.4680941504841687
F South training score: 0.3978542065184908 validation score: 0.4801070712899449
G South training score: 0.39849363961325535 validation score: 0.4832974443286834
Gowanpada training score: 0.39956192408712865 validation score: 0.420040926121073
H West ward training score: 0.4218665056441906 validation score: 0.4937470084483826
K East ward training score: 0.4297376498169233 validation score: 0.47151137431094114
K West ward training score: 0.43720964854383737 validation score: 0.47338362966321645
```

```
Andheri Test R2-score: 0.34037245391983617
B ward Test R2-score: 0.31262233782613646
Bandra Test R2-score: 0.3200387696456767
Byculla Test R2-score: 0.32811606378566327
C ward Test R2-score: 0.3053474503876432
Chembur Test R2-score: 0.31410293969179026
Chincholi Test R2-score: 0.3045634100136856
Colaba Test R2-score: 0.31098260594692817
D Ward Test R2-score: 0.26155577208370673
Dahisar Test R2-score: 0.3666454083909243
Dindoshi Test R2-score: 0.31867780963555203
F North Test R2-score: 0.3267544795993418
F South Test R2-score: 0.29389090853826105
G South Test R2-score: 0.2988581937899367
Gowanpada Test R2-score: 0.24344197291140957
H West ward Test R2-score: 0.34895473969624224
K East ward Test R2-score: 0.3240759444789555
K West ward Test R2-score: 0.3124699184832427
```

R^2 scores for different regions of Linear Regression model on Training (2016-2019), Validation (2020-2021) and Test (2022-2023) data

```
Training score:0.4350 Validation score:0.4219  Andheri
Training score:0.4061 Validation score:0.5046  B ward
Training score:0.4455 Validation score:0.4794  Bandra
Training score:0.3911 Validation score:0.5090  Byculla
Training score:0.3908 Validation score:0.5127  C ward
Training score:0.4017 Validation score:0.4454  Chembur
Training score:0.4404 Validation score:0.4633  Chincholi
Training score:0.3846 Validation score:0.4766  D Ward
Training score:0.4384 Validation score:0.4577  Dahisar
Training score:0.4389 Validation score:0.4537  Dindoshi
Training score:0.4250 Validation score:0.4522  F North
Training score:0.4247 Validation score:0.4757  F South
Training score:0.4367 Validation score:0.4536  G South
```

R^2 scores for different regions of Dense Neural Network model on Training (2016-2019), Validation (2020-2021)

xApp Testing:

```
(itu) [55ms][~/Documents/IITB/DS593/code/xapp]$ python app.py
```

```
(itu) [58ms][~/Documents/IITB/DS593/code/xapp]$ python distributor.py
```

```
(itu) [4m34.020s][~/Documents/IITB/DS593/code/xapp]$ python app.py
Connected to distributor node: ('127.0.0.1', 50453)
Received from distributor node: 0 1 0 1 0 1 1 0 1 1 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 1 0 0 1
Each number (0/1) corresponds to a particular region
0 denotes normal rainfall forecasted in next 15 minutes
1 denotes heavy rainfall forecasted in next 15 minutes

Decoding and sending commands to respective base stations

Andheri base stations: Normal Rain ['NORMAL_PRIORITY']
B ward base stations: Heavy Rain HIGH_PRIORITY
B ward PWS-IWF servers: Heavy Rain BROADCAST_ALERT
Bandra base stations: Normal Rain ['NORMAL_PRIORITY']
Byculla base stations: Heavy Rain HIGH_PRIORITY
Byculla PWS-IWF servers: Heavy Rain BROADCAST_ALERT
C ward base stations: Normal Rain ['NORMAL_PRIORITY']
Chembur base stations: Heavy Rain HIGH_PRIORITY
Chembur PWS-IWF servers: Heavy Rain BROADCAST_ALERT
Chincholi base stations: Heavy Rain HIGH_PRIORITY
Chincholi PWS-IWF servers: Heavy Rain BROADCAST_ALERT
Colaba base stations: Normal Rain ['NORMAL_PRIORITY']
D Ward base stations: Heavy Rain HIGH_PRIORITY
D Ward PWS-IWF servers: Heavy Rain BROADCAST_ALERT
Dahisar base stations: Heavy Rain HIGH_PRIORITY
Dahisar PWS-IWF servers: Heavy Rain BROADCAST_ALERT
Dindoshi base stations: Normal Rain ['NORMAL_PRIORITY']
F North base stations: Normal Rain ['NORMAL_PRIORITY']
F South base stations: Heavy Rain HIGH_PRIORITY
F South PWS-IWF servers: Heavy Rain BROADCAST_ALERT
G South base stations: Heavy Rain HIGH_PRIORITY
G South PWS-IWF servers: Heavy Rain BROADCAST_ALERT
Gowanpada base stations: Normal Rain ['NORMAL_PRIORITY']
H West ward base stations: Heavy Rain HIGH_PRIORITY
H West ward PWS-IWF servers: Heavy Rain BROADCAST_ALERT
K East ward base stations: Normal Rain ['NORMAL_PRIORITY']
K West ward base stations: Normal Rain ['NORMAL_PRIORITY']
Kandivali base stations: Normal Rain ['NORMAL_PRIORITY']
Kurla base stations: Normal Rain ['NORMAL_PRIORITY']
L ward base stations: Normal Rain ['NORMAL_PRIORITY']
M West ward base stations: Normal Rain ['NORMAL_PRIORITY']
MCGM 1 base stations: Normal Rain ['NORMAL_PRIORITY']
Malvani base stations: Normal Rain ['NORMAL_PRIORITY']
Marol base stations: Normal Rain ['NORMAL_PRIORITY']
Memonwada base stations: Normal Rain ['NORMAL_PRIORITY']
Mulund base stations: Heavy Rain HIGH_PRIORITY
Mulund PWS-IWF servers: Heavy Rain BROADCAST_ALERT
N ward base stations: Heavy Rain HIGH_PRIORITY
N ward PWS-IWF servers: Heavy Rain BROADCAST_ALERT
```

```
Nariman Fire base stations: Normal Rain ['NORMAL_PRIORITY']
Rawali camp base stations: Normal Rain ['NORMAL_PRIORITY']
S ward base stations: Heavy Rain HIGH_PRIORITY
S ward PWS-IWF servers: Heavy Rain BROADCAST_ALERT
SWD Workshop dadar base stations: Normal Rain ['NORMAL_PRIORITY']
Thakare natya base stations: Heavy Rain HIGH_PRIORITY
Thakare natya PWS-IWF servers: Heavy Rain BROADCAST_ALERT
Vikhroli base stations: Normal Rain ['NORMAL_PRIORITY']
Worli base stations: Normal Rain ['NORMAL_PRIORITY']
vileparle W base stations: Heavy Rain HIGH_PRIORITY
vileparle W PWS-IWF servers: Heavy Rain BROADCAST_ALERT
```

Comments:

1. app.py is the xApp program, distributor.py is the program run by Distributor node.
2. xApp connects to the Distributor node and the Distributor node sends 0/1 data denoting normal or heavy rainfall respectively, regionwise, to the xApp when it gets the information from the Model.
3. xApp then determines the base stations corresponding to each region and sends E2 configuration messages to the base stations as required for dynamic resource allocation and to the PWS_IWF servers for emergency alert broadcasting.
4. The base stations of the regions where heavy rainfall is forecasted will be sent E2 messages for "HIGH_PRIORITY" allocation and "BROADCAST_ALERT" (from PWS-IWF and AMF), whereas the base stations where normal or zero rainfall is forecasted will be sent E2 messages for "NORMAL_PRIORITY" allocation.