The Scientific Temper (2024) Vol. 15 (4): 3136-3148
E-ISSN: 2231-6396, ISSN: 0976-8653
Doi: 10.58414/SCIENTIFICTEMPER.2024.15.4.22
https://scientifictemper.com/

RESEARCH ARTICLE

Smart flood monitoring in Guwahati city: A LoRa-based AIoT and edge computing sensor framework

Rupesh Mandal[1*], Bobby Sharma[1], Dibyajyoti Chutia[2]

Abstract
In today's context, urban flooding has emerged as a pervasive and significant global challenge, resulting in substantial economic losses spanning both human lives and property damage. With a concerning rise in urban flood-related fatalities and financial impacts, there's an urgent call for enhanced flood risk management strategies. Although floods, as natural disasters, cannot be entirely prevented or eliminated, their catastrophic effects can be significantly reduced or mitigated. Cutting-edge technologies like the internet of things (IoT) and artificial Intelligence offer promising solutions for flood prediction. These advancements facilitate early warning systems, enabling pre-emptive evacuation measures to safeguard lives and minimize economic repercussions. This work aims to develop and implement a system leveraging IoT-derived data with an edge computing framework. It also uses machine learning techniques for fuzzy-based fused framework to provide rainfall prediction early flood warnings, focusing on risk mitigation as a proactive approach to address this pressing issue.

Keywords: Urban flood, Internet of things, Artificial intelligence, LoRa, Edge computing, Rainfall prediction.

Introduction
Natural disasters like cyclones, thunderstorms, tornadoes, cloudbursts, floods, earthquakes, avalanches, and landslides are primary causes of both human casualties and property damage (Zhou, Xu and Fujita,2018). Among these, floods stand out, responsible for a significant 32% of damages and nearly 26% of global deaths (Sivaramakrishnan and Singh, 2003 and Dhar and Nandargi, 2019). Unlike riverine floods, urban floods arise from excessive runoff in densely populated areas, lacking adequate drainage (Gupta and Nair, 2011; Hung, Babel, Weesakul and Tripathi, 2008).

[1]Department of Computer Science and Engineering, School of Technology, Assam Don Bosco University, Guwahati, Assam, India.

[2]North Eastern Space Applications Centre, Department of Space, Government of India, Umiam, Meghalaya, India.

***Corresponding Author:** Rupesh Mandal, Department of Computer Science and Engineering, School of Technology, Assam Don Bosco University, Guwahati, Assam, India., E-Mail: rupesh.mandal@dbuniversity.ac.in

How to cite this article: Mandal, R., Sharma, B., Chutia, D. (2024). Smart flood monitoring in Guwahati city: A LoRa-based AIoT and edge computing sensor framework. The Scientific Temper, **15**(4):3136-3148.

Doi: 10.58414/SCIENTIFICTEMPER.2024.15.4.22

Source of support: Nil

Conflict of interest: None.

Urban flooding, unlike its traditional counterparts, remains a widely discussed yet hidden challenge due to data deficiencies. Rapid urbanization, coupled with climate change, has made waterlogging a pressing issue, notably in India over the past few decades. Urbanization rates have soared from 18% in 1960 to 35% in 2019, expected to surpass 50% by 2050 due to employment opportunities and smart city development (Sukhwani, 2020). However, this swift urban growth leads to various challenges—lack of preparedness, poverty, inadequate infrastructure planning, uncontrolled settlements, and industrial expansion (Machado and Diliey, 2009 and Jarah *et al*., 2019).

The unchecked surge in population and settlements results in drain blockages and overloading, causing waterlogging and subsequent urban floods (Vorobevskii *et al*., 2020). Urbanization alters natural catchment areas, rendering land surfaces impermeable (Awakimjan, 2015 and; Zameer *et al*., 2013). The primary causes of urban floods lie in the incapacity of natural and drainage systems to manage runoff discharge and precipitation volume in urban zones. High rainfall and impermeable surfaces, alongside dense construction, contribute to escalated urban flood instances (Gaitan *et al*., 2016 and Yao *et al*., 2016). Additionally, sustained moderate rainfall and intense, localized convective storms within narrow watersheds can swiftly elevate water levels, leading to urban flash flooding (Abdullah *et al*., 2014, Abdullah *et al*., 2018, Coulthard *et al*., 2007 and Mohtar *et al.,* 2020.)

Received: 10/11/2024 **Accepted:** 26/11/2024 **Published** : 20/12/2024

India's diverse topography and meteorological conditions contribute to an escalating frequency of urban floods (Dhiman *et al.*, 2019). Numerous Indian cities have experienced devastating urban floods due to unplanned urbanization and climate change, including Kerala (2018), Hyderabad (2016, 2000), Chennai (2015, 2004), Srinagar (2014), Guwahati (2010), Delhi (2009, 2002, 2003), Surat (2006), and Mumbai (2005) (Rafiq *et al.*,2016). The Northeastern region faces heightened risks, particularly Guwahati, which has suffered recurrent urban flooding crises (Davis, 2014 and Borah, 2019). While urban flooding cannot be entirely prevented, employing advanced technology to mitigate its impacts becomes imperative. Technological interventions, especially leveraging IoT, offer promise in reducing the impact of climate change on urban flooding, crucial for sustaining a city's economic growth (Miller and Hutchins,2017). Recent years have emphasized the necessity of urban flood control due to their increasing frequency and severity, urging a focus on technological interventions to minimize property damages (Chen *et al.*, 2014, Zanella *et al.*, 2014 and Hu and Ni, 2017)

IoT, with its real-time sensors, significantly enhances flood monitoring and data capture, surpassing traditional systems (Amaxilatis *et al.*, 2019; Dong and Yang, 2019). Its ability to generate images and sensor data improves overall supervision, enabling early flood detection by monitoring regular features for disaster warnings, ultimately preventing damage (Mosquera and Dilley,2009).

Major contributions of this work include:

- Development of an IoT based LoRa framework for tracking road and drainage water level and other weather parameters from the study Area.
- Integration of Artificial Intelligence with IoT for efficient prediction of rainfall at the Edge devices.
- Development of an efficient fuzzy based machine learning fusion approach for an optimized and improvised prediction.

Literature Review

Recently, IoT applications have seen substantial use in managing and monitoring entire environments (Jiao and Liu, 2018). One primary application focuses on urban flood monitoring, aiming to address challenges, exploit advantages, and enhance effectiveness. To ensure accurate analysis for predicting and preventing urban floods, researchers are actively exploring IoT applications in disaster prediction, prevention, and flood impact analysis (Orozco and Caballero, 2018). Several studies delve into implementing IoT-based applications. A comprehensive assessment of using computer vision and IoT-based sensors for flood monitoring and mapping has been conducted and the study has also highlighted the significance of IoT sensors in gathering intelligence for early warnings and evacuations (Arshad *et al.*, 2019). In another work, prediction models have been developed using IoT sensor data and machine learning for forecasting regional inundation (Yang and Chang, 2020). Li *et al.* (2022) devised an intelligent system leveraging GIS technology and rainwater simulation models to prevent waterlogging and control floods. Similarly, the FloodX project has been launched, which utilizes alternative and conventional sensors for urban flood monitoring (Vitry *et al.*, 2017). In another approach, Artificial Intelligence was integrated with third-party weather forecasts and local sensor data to create a prediction system for urban floods. The work specifically used artificial neural networks and machine learning alongside IoT for flood prediction, aiming to enhance prediction accuracy (Wang and Abdelrahman, 2023). Likewise, an early monitoring system was proposed employing Computer Vision and IoT cloud for identifying flood severity levels and issuing community alerts (Soh *et al.*, 2022). Additionally, hardware-based sensor prototypes have been developed, implementing lightweight AI algorithms on edge devices like Raspberry Pi for flood monitoring (Samikwa *et al.*, 2020). Another study has explored IoT applications in monitoring river water levels, developing specialized systems for this purpose (Moreno *et al.*, 2019). Various communication technologies, including LPWAN like LoRa, have been incorporated into IoT-based flood monitoring for cost-effective and energy-efficient systems (Ragnoli *et al.*, 2020). Solar energy, microcontrollers, and power management techniques have been suggested for IoT-based flood early warning systems (Uranus *et al.*, 2022). The necessity of early warning systems and real-time monitoring powered by IoT for disaster prediction and flood impact analysis has been emphasized by Kitagami *et al.* (2016).

Utilizing wireless sensors driven by IoT is another avenue for flood monitoring and forecasting, coupled with computational models like ANNs (Bande and Shete, 2017). Mane *et al.* highlighted IoT's potential in precise flood prediction through data mining and wireless sensor networks (Mane and Mokashi, 2015). The role of IoT in real-time monitoring and managing urban public safety and emergency data was one in a study (Du and Zhu, 2012). A literature study was conducted on IoT-based flood data handling, proposing an architecture for IoT infrastructure in flood monitoring (Ghapar *et al.*, 2018). IoT-based sensors play a vital role in urban flood monitoring, installed in flood-prone areas like drainage networks, lagoons, and lakes to gather water levels and relevant data (Keung *et al.*,2018 and Liu *et al.*, 2022). These sensors transmit data through advanced technologies such as LoRa, which consumes minimal power and operates on a Long-Range basis, beneficial in low-connectivity or infrastructurally inadequate regions.

Incorporating edge computing into urban flood monitoring systems proves crucial, enabling local data processing at edge devices, reducing latency, and enhancing response times (Liu *et al.*,2022). This is particularly valuable in time-sensitive applications like flood monitoring, ensuring real-time data analysis for swift alerts and evacuation orders. Edge computing supports decentralized flood monitoring systems with on-site sensors and predictive algorithms. Summing up, the literature on IoT-based flood monitoring underscores the role of IoT sensors, computer vision, AI algorithms, and communication tech in early alerts, severity measurement, and flood prediction. Integrating diverse data sources like local sensors, meteorological forecasts, and remote sensing elevates precision and efficiency in flood monitoring and emergency response. Developing hardware prototypes and leveraging cost-effective communication technologies remains critical in advancing this field.

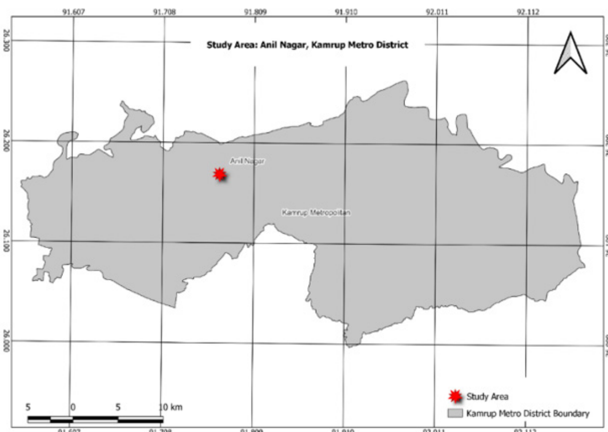


Figure 1: Study area map; Anil Nagar in Kamrup Metro District

Methodology

The primary aim of this work was to deploy IoT devices within the specified study area, detailed in the subsequent section. These devices would utilize advanced communication protocols like long range (LoRa) to establish an independent network, enabling real-time data transmission without reliance on satellite uplinks or mobile networks. Moreover, these devices would conduct live monitoring of drainage water levels, overlaying this information on base maps, including roads and settlements. Additionally, the project includes the development of an android app designed to track flash floods in the area. It will showcase real-time videos and photos captured by cameras placed in various zones, aiding in the monitoring and analysis of flood events. The app will also provide alternate routes based on flash flood alerts, ensuring user safety. Furthermore, the project involves crafting a prototype unified user interface (UUI) for control room operations during flood scenarios. This interface will receive incident alarms and dispatch alerts via an SMS gateway to the emergency response team. This comprehensive approach aims to efficiently manage rescue and relief activities. The subsequent sections elaborate on the different components and tasks accomplished within this project.

Study Area

The study zone designated for investigation was pinpointed as the Anil Nagar and Zoo Road localities within Guwahati City, situated in Assam, India (represented in Figure 1). Guwahati, the principal city in the Northeastern region, serves as the gateway to the seven sister states. Recognized for its vibrant cultural heritage, lush landscapes, and historical significance, this urban hub embodies diverse attractions. Anil Nagar, nestled in the southern precincts of Guwahati City, aligns along the banks of the Bharalu River, a tributary of the formidable Brahmaputra River. This area features low-lying terrain encompassing a blend of residential quarters, commercial hubs, and open expanses. Under this study's purview, a humid subtropical climate prevails, characterized by distinct wet and dry seasons. The monsoon, spanning from June to September, brings considerable rainfall to the region. Anil Nagar, in particular, contends with vulnerability to sudden floods during this period, attributed to its geographical layout and inadequate drainage infrastructure.

Anil Nagar, in the city of Guwahati, faces severe flood risks, leading to significant economic losses and disruptions in daily life. The escalating vulnerability to flash floods stems from rapid urbanization, inadequate drainage, and intense monsoon rains. Heavy rainfall swiftly raises water levels in the Bharalu River and its tributaries, resulting in waterlogging and inundation of residential areas and streets. With a mix of commercial and residential properties and a dense population, the area's low-lying residences, roads, critical infrastructure, and utilities are highly susceptible to submersion during flash floods. Initially, a small portion from the Kamrup Metro District has been selected for the pilot study. The selection of this area for study is due to its extreme susceptibility to urban flooding, necessitating effective monitoring and a warning system. The solution developed to address this issue could potentially be extended to other flood-prone regions within Guwahati and across the nation.

Materials Used

The system is structured around two device categories: water-sensing nodes and base stations. These components, detailed in Table 1, have been specifically selected and tailored to ensure optimal functionality, reliability, and durability for the system's development.

Architecture

The design of this study follows a layered structure consisting of four distinct layers. This structured approach allows for an iterative and systematic method to address challenges, refine processes, and ensure smooth functionality. Figure 2 visually demonstrates the layered architecture of the proposed system.

Table 1: Components used for design of water sensing nodes (left) and base station (right)

Water sensing nodes	*Base station*
Arduino Nano	Raspberry Pi 4
868 Mhz Lora	868 Mhz Lora
Lora antenna	Lora Antenna
Water level sensor (contact Type sensor)	BMP180
2S 3P Battery pack with BMS	Air quality measurement
Enclosure	Encloser
PCB with components	Base station assembly
Connectors	Anemometer and tripling rain gauge
Solar panel and charge controller	Web-Cam
Voltage regulator	PCB with components

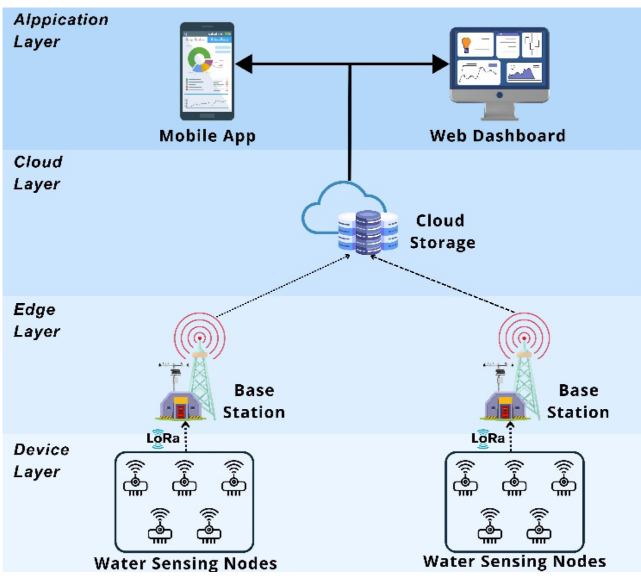


Figure 2: Layered architecture

Device layer

This layer encompasses the water sensing nodes, pivotal devices designed to measure water levels in specific zones such as roads and drainage systems, crucial for identifying potential flash flood scenarios. These nodes are engineered with versatility to function across diverse situations and effectively operate in multiple flood-prone areas. To facilitate seamless data transmission, these devices integrate LoRa communication modules, enabling real-time flood monitoring even without internet connectivity. Operating autonomously, these nodes are powered by rechargeable battery packs equipped with solar panels, eliminating reliance on conventional electricity sources. The nodes employ contact-type water level sensors installed within PVC pipes, positioned 200 cm apart, to detect water levels. Additionally, their battery packs are solar-charged, ensuring independent operation regardless of electric supply disruptions during disasters.

Edge layer

This layer is the base station, a comprehensive assembly of diverse sensors capable of capturing an extensive range of weather parameters, including wind speed, rainfall, wind direction, atmospheric pressure, humidity, altitude, and air quality metrics. Serving as the central hub, it receives transmitted data from water sensing nodes and LoRa receivers. It then merges information gathered from weather sensors and water sensing nodes. This amalgamated data can be uploaded to a cloud server via internet connectivity, encompassing weather data and various weather parameters. Additionally, the base station can locally process data, ensuring data retention in scenarios where network connectivity for cloud uploads is unavailable. However, a consistent power supply is essential to sustain base station operations. Equipped with a camera module, the base station can capture real-time flood situations, enhancing its monitoring capabilities during flood incidents.

Cloud layer

Within this layer, the information gathered from the device layer (Water Sensing Nodes) and the edge layer (Base Stations) is consolidated and stored, forming a vital resource for future urban flood analyses. Google Firebase serves as the storage platform for all data, encompassing both current and historical datasets. This layer ensures scalability robust storage, and offers security measures alongside remote monitoring capabilities.

Application layer

This segment manages the interaction between the application and the real-time database. To facilitate this, two distinct applications have been developed: a) An Android application catering to the general public and b) a dashboard designed for stakeholders involved in Disaster Management operations.

Figure 3 illustrates the design of both the water sensing nodes and base station components. The figure shows how the water sensing nodes were installed on roads with the existing electric poles of streetlights. These nodes were powered by a battery pack and recharged by solar panels.

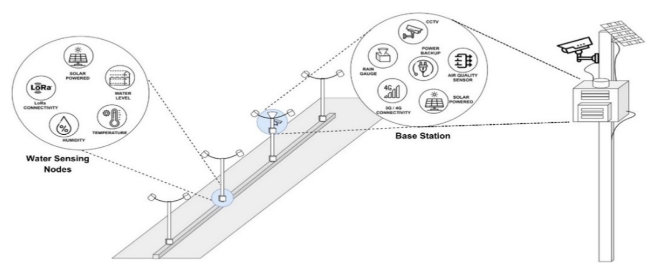


Figure 3: Design of base station and water sensing nodes

The base station is also supposed to be installed within the range of 5 km and it requires an uninterrupted power supply. The components used for the water sensing nodes, as well as the base station, are given in Table 1.

Prediction Models

A rainfall prediction system was developed using time series data of weather parameters. Various machine learning models were trained and tested using 10 years' worth of weather data from Guwahati city. The data was sourced from the Indian Meteorological Department (2 years) and NASA POWER (10 years). Initially collected data from the devices were insufficient for machine learning models, but ongoing data collection will enhance future training and improve predictions. Different classifiers including Random Forest, XGBoost, CatBoost, and KNN, were tested.

Random forest

Breiman introduced random forest (RF) in 2001 specifically for classification tasks. This ensemble machine learning algorithm utilizes multiple classification trees, hence the name "random forest." When computing regression, it combines diverse decision trees for both regression and classification (Bowles, 2019). Despite its inclination toward variables with higher levels among categorical variables with different levels, the RF algorithm is regarded as an exceptionally robust learning algorithm in modern times (Cutler and Stevens,2012).

Extreme gradient boosting

Extreme Gradient Boosting (XGBoost) represents an advanced machine learning technique rooted in the gradient boosting algorithm developed by (Chen and Guestrin, 2016). It excels in handling overfitting through model regularization. This specific algorithm was chosen for the current study due to its exceptional speed in execution. XGBoost was applied across all three training and testing ratios.

Regarding categorical boosting

The categorical boosting model (CatBoost) operates through a four-part technique:

- Data pre-processing involves gathering pollutants and alternative meteorological data, addressing missing values.
- Analyzing the relationship between meteorological variables and pollutants.
- Determining feature importance by assessing meteorological parameters and their association with air pollutants before model implementation.
- Utilizing models like ARIMAANN, ARIMA-SVM, PCR, DT, and CatBoost (Shahriar, 2021).

K-nearest neighbors

K-nearest neighbor (KNN) stands as a non-parametric learning algorithm employing the Euclidean, Manhattan, and Minkowski distance approaches for classification (Bowles,2019). Studies suggest KNN excels when working with a minimal number of features (Oswal,2019). The calculation of the Euclidean distance involves equation 4, demonstrated below, where x_{ij} and x_{io} represent the ith data point in the jth predictor and predictand.

$$d_j = \sqrt{\sum_{i=1}^{n}(x_{ij} - x_{io})^2} \tag{1}$$

The following equation is utilised to calculate the KNN value

$$Z_r = \sum_{K=1}^{K} f_k(d_j * Z_k) \tag{2}$$

$\boldsymbol{Z_r}$ and $\boldsymbol{Z_k}$ denote the predicted and neighboring data, respectively, while $\boldsymbol{f_k(d_j)}$ represents the kernel function utilized. This study leverages seven meteorological features, making KNN a preferred choice. As per (Zhang *et al.*, 2017), the efficacy of KNN in modeling hinges on the number of neighbors (K) used, set to 5 after initial evaluation in this study. KNN with K=5 was applied across all three training and testing ratios.

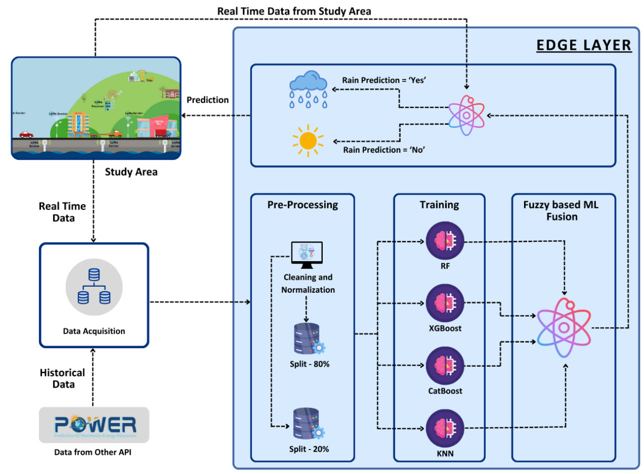


Figure 4: Architecture of rainfall prediction framework

This research also introduces a rainfall prediction framework (Figure 4) utilizing a fusion technique in machine learning for smart cities. The framework primarily comprises two layers: training and testing, each comprising multiple stages. The initial stage of the training layer involves extracting weather attributes from advanced sensors in the smart city. However, for this research, a real-time pre-labeled dataset for rainfall prediction from a weather forecasting website (https://power.larc.nasa.gov/data-access-viewer) for Guwahati City was used, encompassing 91991 instances and 7 features (6 independent and 1 dependent). The data pre-processing phase involves three activities: cleaning, normalization, and splitting. Cleaning aims to address missing values via mean imputation, while normalization standardizes attribute values. These activities facilitate classifiers in achieving maximum accuracy. In the third pre-processing activity, cleaned and normalized data is divided

Table 2: Input into the fuzzy layer for the proposed model

Individual model prediction				*Final rainfall prediction*
RF	*XGBoost*	*CatBoost*	*KNN*	
Yes	Yes	Yes	Yes	Yes
Yes	Yes	Yes	No	Yes
Yes	Yes	No	Yes	Yes
Yes	Yes	No	No	Yes
Yes	No	Yes	Yes	Yes
Yes	No	Yes	No	Yes
Yes	No	No	Yes	Yes
Yes	No	No	No	No
No	Yes	Yes	Yes	Yes
No	Yes	Yes	No	No
No	Yes	No	Yes	No
No	Yes	No	No	No
No	No	Yes	Yes	No
No	No	Yes	No	No
No	No	No	Yes	No
No	No	No	No	No

into training and test subsets using an 80:20 split ratio. Post-pre-processing, the dataset undergoes classification, with both training and test data fed into four techniques (RF, XGBoost, CatBoost and KNN. These algorithms are iteratively optimized during training and testing to enhance accuracy. Following classification, the trained models are input into the fuzzy layer, responsible for developing and implementing fuzzy logic for final prediction. The input into the fuzzy layer for the proposed model is presented in Table 2.

Prediction Models

IoT framework development

The devices underwent several stages of development, progressing through various prototype versions that built upon the successes and insights gained from earlier iterations. The final model represents an innovative solution for flood monitoring, integrating advanced technologies to bolster disaster resilience and preparedness. Figures 5, 6, and 7 visually illustrate the evolution of these devices across their developmental phases.

Mobile application development frameworks

Utilized open-source mobile app development frameworks, enabling cross-platform compatibility with Android and iOS devices. Additionally, an android-based mobile application and a dashboard were developed to provide real-time urban flood information with user interaction capabilities. The application was structured into three layers using a service-oriented architecture. These layers consist of the client layer,

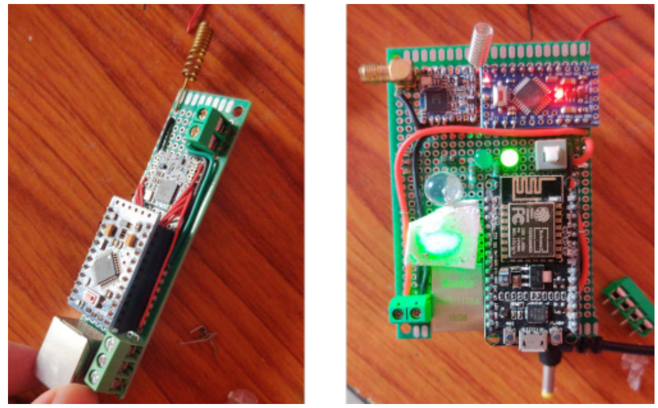


Figure 5: Initial prototype showcases water sensing node (Left) and base station (Right)

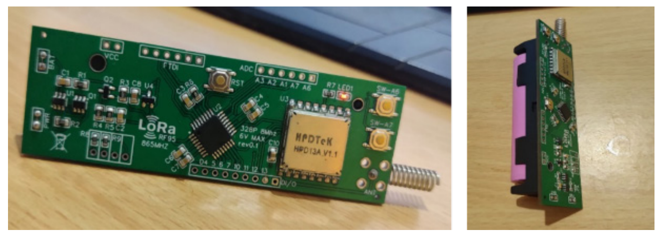


Figure 6: An upgraded and miniaturized water sensing node

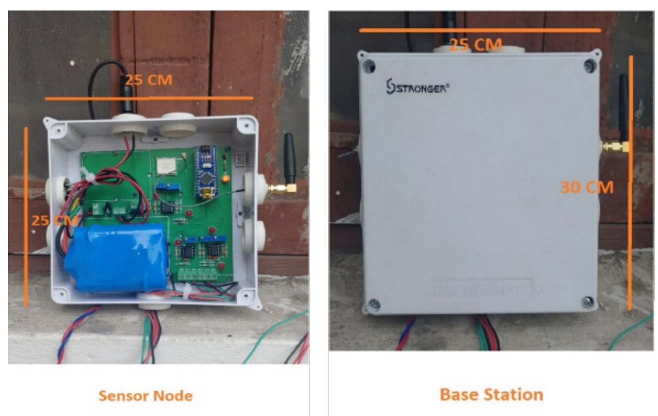


Figure 7: Final prototype featuring water sensing node (Left) and base station (Right)

business layer, and database layer, each serving specific purposes. Technical specifications include.

Service-oriented architecture

The application architecture operates in three layers, with the database layer housing project-related information, including geospatial data and project details, ensuring secure and structured storage. The client layer, embodied by the dashboard and mobile app, serves as the user interface for accessing and visualizing data. Users interact with these interfaces to view project status, input field data, and retrieve pertinent information. The business

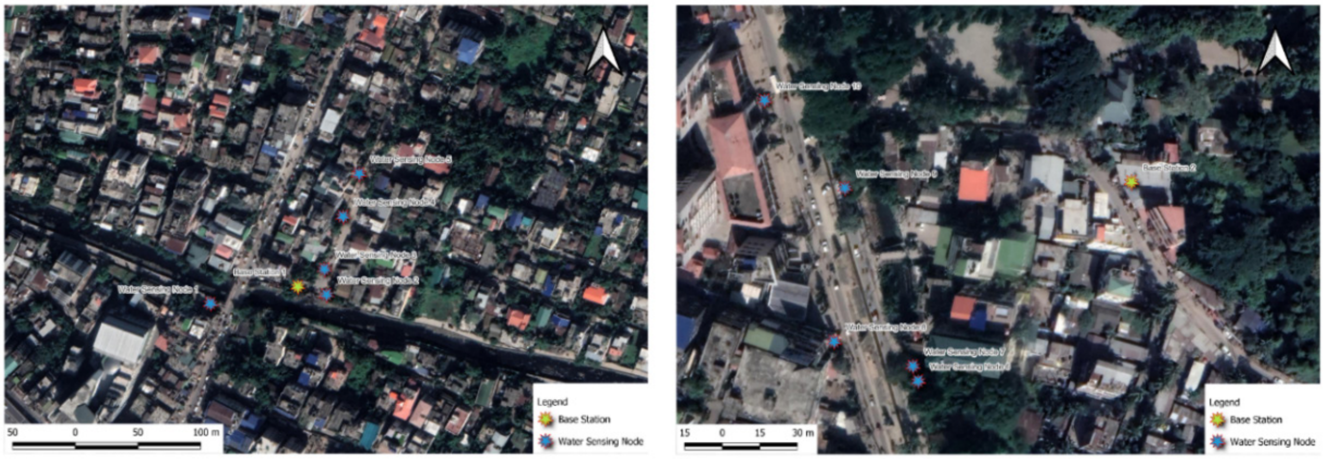


Figure 8: Placement of water sensing nodes and base station 1 in Anil Nagar area (left) and Zoo Road area (right), Guwahati, Assam, India

layer functions as an intermediary between the client and database layers, managing data logic and processing. To facilitate seamless data access, RESTful APIs are implemented, allowing the client layer to communicate with the business layer, accessing or updating project data as required. Adherence to open standards through RESTful APIs ensures interoperability and facilitates integration with external applications and stakeholders. This service-oriented architecture enhances modularity, flexibility, and efficiency in accessing and managing project information for effective monitoring and governance.

Results and Discussion

This section outlines the deployment and configuration process of the nodes, establishing connectivity among them. An IoT network equipped with sensors and configured with LoRa-based infrastructure was installed at selected sites within the Anil Nagar Area, detailed in Table 3. Testing was conducted to ensure seamless connectivity between the IoT devices and the central server using web protocols and services.

Table 3: Locations of base stations and their corresponding water sensing nodes

S. No.	*IoT Nodes*	*Latitude*	*Longitude*
1	Base Station 1	26.167094	91.76964
2	Water Sensing Node 1	26.166966	91.768945
3	Water Sensing Node 2	26.167037	91.76987
4	Water Sensing Node 3	26.167221	91.769851
5	Water Sensing Node 4	26.167602	91.769996
6	Water Sensing Node 5	26.16791	91.770121
7	Base Station 2	26.162917	91.781961
8	Water Sensing Node 6	26.162194	91.781118
9	Water Sensing Node 7	26.162247	91.781096
10	Water Sensing Node 8	26.162333	91.780781
11	Water Sensing Node 9	26.162885	91.780818
12	Water Sensing Node 10	26.163198	91.780494

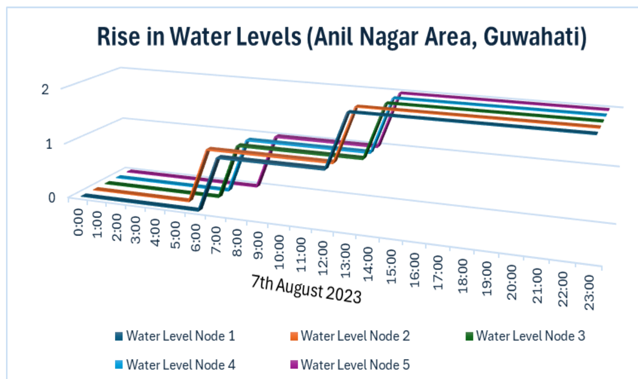


Figure 9: Collected data from water sensing nodes (Anil Nagar Area)

This study involved deploying base stations and water sensing nodes strategically across the study area. As part of this initiative, 2 base stations and 10 water sensing nodes were developed and installed. The framework was designed with each base station communicating with 5 water sensing nodes. Accordingly, 1 base station and 5 water sensing nodes were positioned in the Anil Nagar Area, while the other base station and the remaining water sensing nodes were placed in the vicinity of Zoo Road, near the Assam State Zoo. This extension aimed to broaden the scope of data collection. The coordinates of these installed devices are listed in Table 3.

The following figure (Figure 8) illustrates the geographical arrangement of the water sensing nodes and base stations positioned in the Anil Nagar and Zoo Road areas of Guwahati, respectively.

Figure 9 illustrates the data gathered from various water sensing nodes installed in the Anil Nagar area on August 7th, 2023. The data records indicate a noticeable increase in

Time	Base Station-Wind Direction	Weather API-Wind Direction
00:00:00	205.76	205.53
01:00:00	207.16	207.19
02:00:00	214.38	214.4
03:00:00	223.25	223.32
04:00:00	224.06	223.83
05:00:00	219.38	219.4
06:00:00	214.58	214.72
07:00:00	202.63	202.7
08:00:00	195.21	195.29
09:00:00	204.93	204.93
10:00:00	241.41	241.53
11:00:00	269.33	269.34
12:00:00	287.19	287.43
13:00:00	311.75	312.03
14:00:00	338.39	338.43
15:00:00	9.35	9.09
16:00:00	25.49	25.44
17:00:00	28.75	28.57
18:00:00	45.54	45.76
19:00:00	120.51	120.31
20:00:00	159.27	159.3
21:00:00	168.99	168.82
22:00:00	194.76	194.78

Figure 10: Data was collected through base station vs weather API and their comparison for August 7th 2023

water levels within the locality. Numeric values denote the water levels: 0 signifies no water, 1 indicates low water level (200 cm), 2 represents moderate water level (400 cm), and 3 signifies high water level (600 cm).

Figure 10 presents a snapshot of the data collected at base station on August 7th, 2023. Additionally, it includes a comparison between the data collected via sensors, particularly from the Base Station installed in Zoo Road, and the data acquired from a weather API.

Consistency

The data from the base station and the weather API (https://power.larc.nasa.gov/data-access-viewer/) are highly consistent across all metrics, with very close mean, median, and standard deviation values.

Temperature and Humidity

Both data sources show almost identical statistics, indicating reliable temperature and humidity readings.

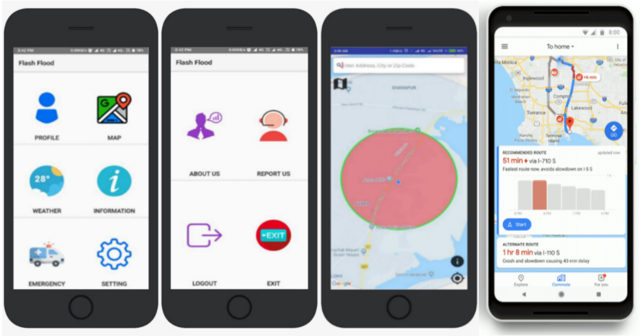
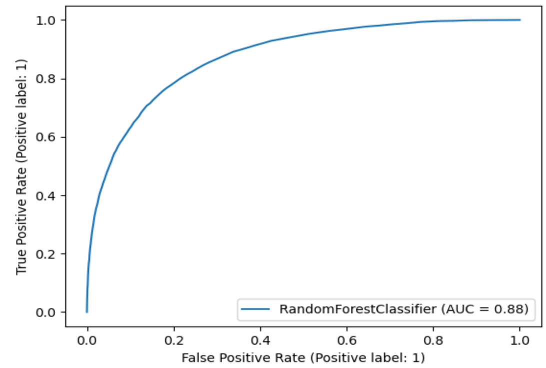


Figure 11: Screenshots of the android application developed

Figure 12: Dashboard of the project work with live streaming facility



RF Testing Results			
No. of samples		N=	18399
Input	Expected Result (E)	Output Result (O)	
		O-Negative	O-Positive
	E-Negative 9261	8094	1167
	E-Positive 9138	925	8213
	Accuracy=	***0.886298168***	
	Miss Rate=	***0.113701832***	

Figure 13: Statistical analysis for random forest model

Rainfall and Pressure

Slight variations exist in rainfall and pressure data, but overall, the values are very close.

Wind Metrics

Wind speed and direction show minor differences, but the overall patterns are similar.

The mobile application developed (Figure 11) offers user-friendly interfaces. It displays flooded areas marked by red circles on Google Maps and provides alternative routes. Users can interact with the app using multiple services, including reporting, emergency alerts, and accessing weather reports. The developed dashboard is depicted in the following figure (Figure 12), showcasing various statistics and live streaming feeds of the respective areas.

Following is the discussion of the statistical methods used to evaluate the predicted performance of the suggested framework, as well as other popular classification models like random forest (RF), extreme gradient boosting (XGBoost), categorical boosting (CAtBoost), and K-nearest neighbour (KNN). Below are several formulae where O-negative and O-positive stand for projected negatives and positives, respectively, and E-negative and E-positive denote expected negatives and expected positives, respectively.

$$Accuracy = \frac{\left(\frac{O_{Negative}}{E_{Negative}} + \frac{O_{Positive}}{E_{Positive}}\right)}{(E_{Negative} + E_{Positive})} \quad (3)$$

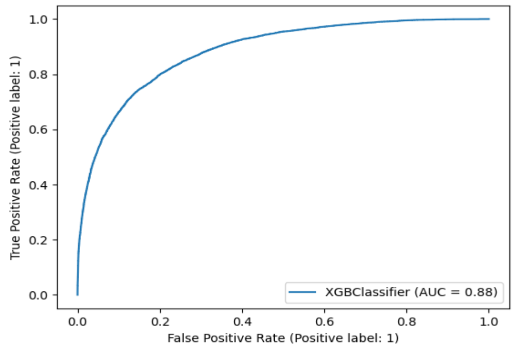
$$Miss\ Rate = \frac{\left(\frac{O_{Positive}}{E_{Negative}} + \frac{O_{Negative}}{E_{Positive}}\right)}{(E_{Negative} + E_{Positive})} \quad (4)$$

During the testing phase using random forest (RF), out of 9261 instances, 8094 were classified as negative, while 8213 instances were identified as positive out of 9138 (as detailed in Figure 13). Comparing the expected versus achieved output, the testing accuracy stood at 88.6%, with an 11.4% miss rate

In the XGBoost testing phase, among 9261 instances, 8103 were classified as negative, and 8134 instances were marked as positive (as depicted in Figure 14). The analysis of expected versus achieved output during XGBoost testing demonstrated an accuracy of 88.2 and an 11.8% miss rate.

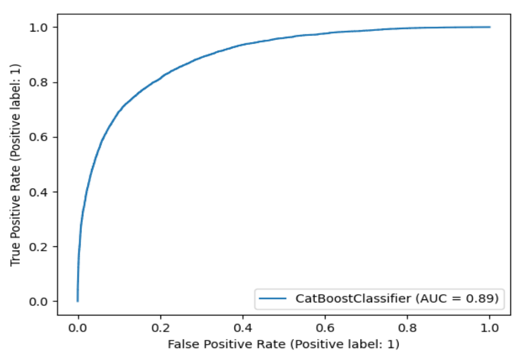
In the testing phase using CatBoost, among 9261 instances, 8241 were classified as negative, and 8231 instances were identified as positive (detailed in Figure 15). The testing analysis revealed an accuracy of 89.5% and a miss rate of 10.5% when comparing the expected *versus* achieved results.

In KNN's testing phase, out of 9261 instances, 7213 were classified as negative, and 7311 instances were categorized



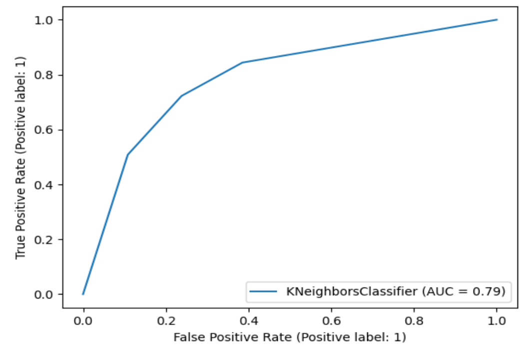
XGBoost Testing Results			
No. of samples		N=	18399
INPUT	Expected Result (E)	Output Result (O)	
		O-Negative	O-Positive
	E-Negative 9261	8103	1158
	E-Positive 9138	1004	8134
	Accuracy=	***0.882493614***	
	Miss Rate=	***0.117506386***	

Figure 14: Statistical analysis for XGBoost model



XGBoost Testing Results			
No. of samples		N=	18399
INPUT	Expected Result (E)	Output Result (O)	
		O-Negative	O-Positive
	E-Negative 9261	8241	1020
	E-Positive 9138	907	8231
	Accuracy=	***0.895266047***	
	Miss Rate=	***0.104733953***	

Figure 15: Statistical analysis for CatBoost model



KNN Testing Results			
No. of samples		N=	18399
INPUT	Expected Result (E)	Output Result (O)	
		O-Negative	O-Positive
	E-Negative 9261	7213	2048
	E-Positive 9138	1827	7311
	Accuracy=	***0.789390728***	
	Miss Rate=	***0.210609272***	

Figure 16: Statistical analysis for K-nearest neighbor

KNN Testing Results			
No. of samples		N=	18399
INPUT	Expected Result (E)	Output Result (O)	
		O-Negative	O-Positive
	E-Negative 9261	8347	914
	E-Positive 9138	486	8652
	Accuracy=	***0.923908908***	
	Miss Rate=	***0.076091092***	

Figure 17: Statistical analysis for proposed model

as positive (referenced in Figure 16). This testing analysis exhibited an accuracy of 78.9 and a 21.1% miss rate when comparing expected versus achieved results with KNN.

Eventually, the entire testing dataset is fed into the fuzzy system for the ultimate prediction. This input includes the test data, output class, and predictions from the employed classifiers. The proposed fused machine learning-based fuzzy system identified 8347 instances as negative out of 9216 and 8652 instances as positive out of 9138 (outlined in Figure 17). When juxtaposing the fuzzy system's output

Table 4: Comparative analysis of existing algorithm with the proposed framework

ML Algorithm	*Accuracy*	*Miss Rate*
Random forest	0.86	0.14
XGBoost	0.88	0.12
Cat boost	0.9	0.1
KNN	0.79	0.21
Proposed algorithm	0.92	0.08

with the expected result, we achieved an accuracy of 92.4% and a miss rate of 7.6%. A comprehensive view of the results from training and test data using RF, XGBoost, CatBoost, KNN, and the proposed fused machine learning technique has been provided in Table 4. Notably, the proposed fused technique exhibited superior performance compared to the four individual machine learning techniques. Additionally, Table 4 offers a comparative analysis between the proposed fused machine learning algorithms and previously published methods for rainfall prediction, considering accuracy and miss rates.

Conclusion and Future Work

This article details the development and deployment of an IoT framework tailored for monitoring urban flood situations, specifically focusing on Anil Nagar and Zoo Road areas in Guwahati City, India. The water sensing nodes are engineered to monitor water levels on roads and detect nearby potential water logging, while the base stations collect this data from the sensing nodes. These data sets are then uploaded to a cloud server, enabling the dissemination of real-time water level information to the general public *via* an Android application. Adopting an edge computing strategy, the bass station device has been deployed with a rainfall prediction model that can use real-time meteorological data from many areas to forecast when it will rain in the next 24 hours. Guwahati City's weather data from the last 10 years was used to train this model. Four separate prediction models—Random forest, extreme gradient boosting, categorical boosting, and K-nearest neighbor—are combined in the proposed fuzzy-based fused framework to forecast rainfall. Additionally, disaster management authorities have access to this data, aiding in swift decision-making during search and rescue operations. This initiative significantly benefits the general public, empowering them to view current flood situations in their vicinity through the Android application. This visibility allows them to choose alternative routes when navigating flood-affected areas. Looking ahead, the future scope of this work involves advancements that aim to create a flood prediction model leveraging water sensing nodes, a location-specific rainfall prediction model and a computer vision-based approach.

Acknowledgments

The funding and assistance from the "North Eastern Space Applications Centre (NESAC), Department of Space, Government of India, Umiam, Meghalaya" were crucial in allowing us to carry out this study, and we are very thankful to them.

References

Abdullah, J., & Julien, P. Y. (2014). Distributed flood simulations on a small tropical watershed with the TREX model. Journal of Flood Engineering, 5(1-2), 17–37.

Abdullah, J., Muhammad, N. S., Julien, P. Y., Ariffin, J., & Shafie, A. (2018). Flood flow simulations and return period calculation for Kota Tinggi watershed, Malaysia. Journal of Flood Risk Management, 11, 766–782.

Jarah, S. H. A., Zhou, B., Abdullah, R. J., Lu, Y., & Yu, W. (2019). Urbanization and urban sprawl issues in city structure: A case of the Sulaymaniah Iraqi Kurdistan Region. Sustainability, 11(2), 485.

Amaxilatis, D., Boldt, D., Choque, J., Diez, L., Gandrille, E., Kartakis, S., ... & Vestergaard, L. S. (2018). Advancing experimentation-as-a-service through urban IoT experiments. IEEE Internet of Things Journal, 6(2), 2563-2572.

Arshad, B., Ogie, R., Barthelemy, J., Pradhan, B., Verstaevel, N., & Perez, P. (2019). Computer vision and IoT-based sensors in flood monitoring and mapping: A systematic review. *Sensors*, 19(22), 5012.

Awakimjan, I. (2015). Urban flood modelling recommendations for Ciudad Del Plata (Bachelor's thesis). University of Twente, Netherlands.

Bande, S., & Shete, V. V. (2017). Smart flood disaster prediction system using IoT & neural networks. *2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)*, 189-194. IEEE.

Barthélemy, J., Verstaevel, N., Forehead, H., & Perez, P. (2019). Edge-computing video analytics for real-time traffic monitoring in a smart city. *Sensors*, 19(9), 2048.

Borah, S. (2019). Assam: Guwahati gets a big boost in its fight against water-logging. EastMojo. Retrieved from https://www.eastmojo.com/assam/2019/10/16/assam-guwahati-gets-a-big-boost-in-its-fight-against-waterlogging (Accessed December 10th, 2019).

Bowles, M. (2019). *Machine Learning with Spark and Python: Essential Techniques for Predictive Analytics*. John Wiley & Sons.

Chen, S., Xu, H., Liu, D., Hu, B., & Wang, H. (2014). A vision of IoT: Applications, challenges, and opportunities with China perspective. IEEE Internet of Things Journal, 1(4), 349–359.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794).

Coulthard, T., Frostick, L., Hardcastle, H., Jones, K., Rogers, D., Scott, M., & Bankoff, G. (2007). The June 2007 floods in Hull. Final Report by the Independent Review Body, 21st November.

Cutler, D. R., Cutler, A., & Stevens, J. R. (2012). Random forests. In C. Zhang & Y. Ma (Eds.), *Ensemble machine learning* (pp. 157–175). Springer.

Davies, R. (2014). 8 Dead, 1 Missing after Flash Floods in Guwahati, Assam. Flood List. Retrieved from http://floodlist.com/asia/flash-floods-guwahati-assam

Dhar, O. N., & Nandargi, S. (2003). Hydrometeorological aspects of floods in India. Natural Hazards, 28, 1–33.

Dhiman, R., VishnuRadhan, R., Eldho, T. I., & Inamdar, A. (2019). Flood risk and adaptation in Indian coastal cities: recent scenarios. Applied Water Science, 9(1), 5.

Dong, W., & Yang, Q. (2019). Data-driven solution for optimal pumping units scheduling of smart water conservancy. *IEEE Internet of Things Journal*, 6(5), 1.

Du, C., & Zhu, S. (2012). Research on urban public safety emergency management early warning system based on technologies for the internet of things. Procedia Engineering, 45, 748-754.

Gaitan, S., van de Giesen, N. C., & ten Veldhuis, J. A. E. (2016). Can urban pluvial flooding be predicted by open spatial data and weather data? Environmental Modelling & Software, 85, 156–171.

Ghapar, A. A., Yussof, S., & Bakar, A. A. (2018). Internet of Things (IoT) architecture for flood data management. *International Journal of Future Generation Communication and Networking*, 11(1), 55-62.

Gupta, A. K., & Nair, S. S. (2011). Urban floods in Bangalore and Chennai: Risk management challenges and lessons for sustainable urban ecology. Current Science, 100(11), 1638-1645.

Hu, L., & Ni, Q. (2017). IoT-driven automated object detection algorithm for urban surveillance systems in smart cities. IEEE Internet of Things Journal, 5(2), 747-754.

Hung, N. Q., Babel, M. S., Weesakul, S., & Tripathi, N. K. (2008). An artificial neural network model for rainfall forecasting in Bangkok, Thailand. Hydrology and Earth System Sciences, 13(8), 1413–1425.

Jiao, D. L., & Luo, X. (2018). Water quality monitoring system based on LoRa. DEStech Trans. Comput. Sci. Eng, 10, 1107-1116.

Keung, K. L., Lee, C. K. M., Ng, K. K. H., & Yeung, C. K. (2018, December). Smart city application and analysis: Real-time urban drainage monitoring by iot sensors: A case study of Hong Kong. In 2018 IEEE international conference on industrial engineering and engineering management (IEEM) (pp. 521-525). IEEE.

Kitagami, S., Thanh, V. T., Bac, D. H., Urano, Y., Miyanishi, Y., & Shiratori, N. (2016). Proposal of a distributed cooperative IoT system for flood disaster prevention and its field trial evaluation. International Journal of Internet of Things, 5(1), 9-16.

Li, L., Yu, M., Ma, H., Meng, L., & Cui, Z. (2022). Development and Application of Flood Control and Waterlogging Prevention Intelligent Monitoring System Based on Subway" ONE MAP". ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 10, 93-98.

Liu, C. H., Yang, T. H., & Wijaya, O. T. (2022). Development of an artificial neural network algorithm embedded in an on-site sensor for water level forecasting. Sensors, 22(21), 8532.

Mane, S. S., & Mokashi, M. K. (2015, April). Real-time flash-flood monitoring, alerting and forecasting system using data mining and wireless sensor network. In 2015 International Conference on Communications and Signal Processing (ICCSP) (pp. 1881-1886). IEEE.

Miller, J. D., & Hutchins, M. (2017). The impacts of urbanisation and climate change on urban flooding and urban water quality: A review of the evidence concerning the United Kingdom. Journal of Hydrology: Regional Studies, 12, 345–362.

Mohtar, W. H. M. W., Abdullah, J., Maulud, K. N. A., & Muhammad, N. S. (2020). Urban flash flood index based on historical rainfall events. Sustainable Cities and Society, 56, 102088.

Moreno, C., Aquino, R., Ibarreche, J., Pérez, I., Castellanos, E., Álvarez, E., ... & Clark, B. (2019). RiverCore: IoT device for river water level monitoring over cellular communications. Sensors, 19(1), 127.

Mosquera-Machado, S., & Dilley, M. (2009). A comparison of selected global disaster risk assessment results. Natural Hazards, 48, 439–456.

NASA POWER Project. (2024). *Data Access Viewer*. https://power.larc.nasa.gov/data-access-viewer/ [Last Access Date : 10-August-2024]

Orozco, M. M., & Caballero, J. M. (2018). Smart disaster prediction application using flood risk analytics towards sustainable climate action. In MATEC Web of Conferences (Vol. 189, p. 10006). EDP Sciences.

Oswal, N. (2019). Predicting rainfall using machine learning techniques. arXiv preprint arXiv:1910.13827.

Poon, L. (2018). Urban flooding is worryingly widespread in the U.S., but under-studied. City Lab. Retrieved from https://www.citylab.com/environment/2018/12/urban-flooding-report-climate-change-data-disaster/577327/

Rafiq, F., Ahmed, S., Ahmad, S., & Khan, A. A. (2016). Urban floods in India. International Journal of Scientific & Engineering Research, 7(1), 721-734.

Ragnoli, M., Barile, G., Leoni, A., Ferri, G., & Stornelli, V. (2020). An autonomous low-power LoRa-based flood-monitoring system. Journal of Low Power Electronics and Applications, 10(2), 15.

Samikwa, E., Voigt, T., & Eriksson, J. (2020, November). Flood prediction using IoT and artificial neural networks with edge computing. In 2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics) (pp. 234-240). IEEE.

Shahriar, S. A., Kayes, I., Hasan, K., Hasan, M., Islam, R., Awang, N. R., ... & Salam, M. A. (2021). Potential of ARIMA-ANN, ARIMA-SVM, DT and CatBoost for atmospheric PM2. 5 forecasting in Bangladesh. Atmosphere, 12(1), 100.

Sivaramakrishnan, K. C., & Singh, B. N. (2003). Urbanisation. New Delhi: Planning Commission. Retrieved from http://planningcommission.nic.in/reports/sereport/ser/vision2025/urban.pdf (Accessed June 1st, 2019).

Soh, Z. H. C., Abd Razak, M. S., Hamzah, I. H., Zainol, M. N., Sulaiman, S. N., Yahaya, S. Z., & Abdullah, S. A. C. (2022). Riverbank Monitoring using Image Processing for Early Flood Warning System via IoT. International Journal of Integrated Engineering, 14(3), 166-174.

Sukhwani, V., Shaw, R., Deshkar, S., Mitra, B. K., & Yan, W. (2020). Role of smart cities in optimizing water-energy-food nexus: Opportunities in Nagpur, India. Smart Cities, 3(4), 1266–1292.

Uranus, H. P., Adhinugroho, N. R., Yulian, D. H., & Mangunsong, R. (2022). Design and realization of solar-powered IoT-based flood early warning system with telegram messaging, auto-restart watchdog, and power management. GCISTEM Proceeding, 1, 96-108.

Vitry, M., Dicht, S., & Leitão, J. (2017). FloodX: Urban flash flood experiments monitored with conventional and alternative sensors. Earth System Science Data, 9(2), 657-666.

Vorobevskii, I., Al Janabi, F., & Schneebeck, F. (2020). Urban floods: Linking the overloading of a stormwater sewer system to precipitation parameters. Hydrology, 7(2), 35.

Wang, Q., & Abdelrahman, W. (2023). High-precision AI-enabled flood prediction integrating local sensor data and 3rd party weather forecast. Sensors, 23(6), 3065.

Yang, S., & Chang, L. (2020). Regional inundation forecasting using machine learning techniques with the Internet of Things. Water, 12(6), 1578.

Yao, L., Wei, W., & Chen, L. (2016). How does imperviousness impact the urban rainfall runoff process under various storm cases? Ecological Indicators, 60, 893–905.

Zameer, R. M., & Rao, K. R. (2013). Urban flooding- case study of Hyderabad. Global Journal of Engineering Design and Technology, 2(4), 63-66.

Zanella, N., Bui, N., Castellani, A., Vangelista, L., & Zorzi, M. (2014). Internet of things for smart cities. IEEE Internet of Things Journal, 1(1), 22–32.

Zhang, S., Li, X., Zong, M., Zhu, X., & Wang, R. (2017). Efficient kNN classification with different numbers of nearest neighbors. *IEEE Transactions on Neural Networks and Learning Systems, 29*(5), 1774–1785.

Zhou, L., Wu, X., Xu, Z., & Fujita, H. (2018). Emergency decision making for natural disasters: An overview. International Journal of Disaster Risk Reduction, 27, 567–576.