Edge-Integrated IoT and Computer Vision Framework for Real-Time Urban Flood Monitoring and Prediction

Rupesh Mandal[1], Bobby Sharma[2], Dibyajyoti Chutia[3]
Department of Computer Science and Engineering-School of Technology,
Assam Don Bosco University, Guwahati, Assam, India[1, 2]
North Eastern Space Applications Centre-Department of Space, Government of India, Umiam, Meghalaya, India[3]

***Abstract*—Urban flash floods pose a critical threat to rapidly growing cities in India, where unplanned development, climate variability, and inadequate drainage amplify risks. Guwahati, in Northeast India, experiences recurrent inundation during monsoons, disrupting livelihoods and damaging infrastructure. This study presents an integrated IoT and AI-enabled framework for urban flood monitoring and prediction. A LoRa-based IoT sensor network was deployed to capture localized hydrological and meteorological parameters, overcoming the limitations of coarse weather APIs. Rainfall forecasting was implemented at the edge layer using Random Forest, XGBoost, CatBoost, and K-Nearest Neighbors, fused through a fuzzy logic model that achieved 92.4% accuracy, surpassing individual classifiers. In parallel, a computer vision pipeline detected drainage blockages from geotagged user images, with EfficientNetB0-U-Net achieving ~91% accuracy, outperforming ResNet50, InceptionV3, and MobileNetV2. By combining rainfall prediction, IoT sensing, and blockage detection, the proposed framework delivers a holistic, low-cost, and scalable early warning system, marking a novel contribution toward resilient urban flood management in resource-constrained settings.**

Keywords—Urban flood prediction; IoT sensor networks; edge computing; fuzzy logic fusion; hydraulic blockage detection; computer vision

I. Introduction

Natural hazards such as floods, tsunamis, earthquakes, and storms continue to pose significant threats worldwide, affecting human lives, resources, and infrastructure [1]. In the absence of effective monitoring and mitigation strategies, these hazards can escalate into full-scale disasters that disrupt urban life, damage infrastructure, and impede economic growth [2], [3]. Among natural hazards, floods are the most recurrent, accounting for nearly 41% of all reported disasters in the past decade [4]. Flash floods, in particular, have become increasingly destructive in urban settings. In India, such events now occur throughout the year, driven by excessive rainfall, rising water levels, and atmospheric dynamics that lead to intense precipitation [5].

Urban centers, while functioning as economic hubs, face heightened flood vulnerability due to rapid population growth, unsustainable land use, and inadequate waste and sewage management. For instance, Guwahati in Assam—located along the Brahmaputra and its tributary Bharalu—regularly experiences severe inundation in low-lying areas such as Anil Nagar and Zoo Road during the monsoon season, disrupting livelihoods and causing economic losses [6]. Land reclamation near water bodies has further aggravated seasonal flooding challenges, delaying urban development for extended periods. Although cities like Mumbai have relatively advanced flood warning and evacuation systems [7], most Indian cities, including Delhi, Bangalore, and Kolkata, lack such mechanisms.

Recent advances in the Internet of Things (IoT) have demonstrated potential for disaster monitoring and management. By integrating sensors, communication systems, and data analytics, IoT can enable real-time monitoring of hydrological parameters. However, large-scale deployment of IoT-based flood management systems remains constrained by high costs, integration complexities, limited adaptability to diverse flood scenarios, and challenges in decision-making. Despite these issues, IoT has already shown value in healthcare, defense, and disaster response. When coupled with artificial intelligence (AI), data collected from mobile devices, environmental sensors, and satellites can be transformed into actionable insights for flood prediction and mitigation.

A critical but often underexplored factor in urban flash flooding is drainage and culvert blockage. Even in well-designed drainage systems, the accumulation of debris such as boulders, vegetation, and solid waste can drastically reduce flow capacity, leading to localized inundation. Brooks [12], through scaled laboratory models and field observations, demonstrated that culvert inlets are the primary deposition points for boulders and emphasized the need for structural modifications to mitigate blockage. Building on this, Iqbal et al. [13] investigated blockage mechanisms under varying debris types, orientations, and flow conditions, highlighting how hydraulic blockage intensifies during dynamic flood hydrographs. More recently, Iqbal et al. [14] introduced a computer vision-based approach to automate culvert blockage detection, testing multiple CNN models on real-world datasets. While NASNet achieved the highest accuracy (85%) and MobileNet demonstrated faster inference, limitations such as background clutter and simplified labelling criteria constrained performance. These studies underscore the importance of integrating blockage detection into urban flood management frameworks. Manual inspection remains labor-intensive and reactive, whereas AI-driven image-based classification provides a scalable solution for real-time monitoring.

The North-Eastern region of India is particularly vulnerable, with Guwahati witnessing repeated instances of urban flooding in recent years [8]. While complete prevention of floods may not be feasible, adopting advanced technologies such as IoT and AI can significantly reduce their adverse impacts [9], [10]. IoT networks equipped with real-time sensing and analytics can provide both visual and sensor-based data, thereby enhancing early detection and timely warnings [4], [11].

The remainder of this paper is structured as follows. Section II presents a review of existing literature on IoT-based flood monitoring, machine learning for rainfall prediction, and computer vision approaches for blockage detection. Section III details the proposed methodology, including the study area, overall system architecture, IoT framework design, rainfall prediction framework, and blockage detection pipeline. Section IV discusses the results and experimental evaluations, highlighting both quantitative metrics and qualitative analyses. Section V concludes the paper with key findings and outlines potential directions for future research.

Unlike most existing studies that focus either on rainfall prediction or hydrological monitoring in isolation, this work proposes a holistic urban flood management framework that integrates three critical components: 1) a LoRa-based IoT network for real-time sensing of water levels and meteorological parameters, 2) an edge-deployed rainfall prediction system using machine learning with fuzzy fusion for improved accuracy, and 3) a computer vision–based deep learning approach for detecting blockages in drainage systems. To the best of our knowledge, this is the first research effort tailored to Guwahati city, which combines IoT and AI at both the sensing and infrastructure levels to provide a scalable, low-cost, and real-time flash flood early-warning system.

II. Literature Review

Floods are among the most destructive natural disasters worldwide, responsible for significant economic losses, ecological disruption, and fatalities [18], [19]. In India, their frequency has risen steadily due to climate change, unplanned urbanization, and population pressure [15]–[17]. According to the Global Climate Risk Index 2021, India ranks seventh among the most disaster-prone nations [16]. Between 1990 and 2020, floods accounted for more than half of the country's climate-related disasters [16], affecting millions of people each year.

Urban flooding, unlike traditional riverine floods, results primarily from excessive surface runoff in densely built environments where impermeable surfaces and inadequate drainage limit water absorption [20], [21]. This phenomenon is often described as a "hidden challenge" due to limited data availability and its localized, rapid-onset nature [4]. Unregulated settlement growth, inefficient waste management, and encroachment on natural drainage channels have intensified the vulnerability of Indian cities to flash floods [22]–[24]. Short-lived but high-intensity convective storms can overwhelm urban drainage capacity, leading to sudden waterlogging without prior warning [25]–[30]. Major Indian cities such as Mumbai, Delhi, Chennai, Hyderabad, and Surat have witnessed repeated urban flood events in recent decades, highlighting the growing severity of the problem [7].

The Northeastern Region (NER) is particularly vulnerable, with Guwahati city serving as a stark example. Positioned on the banks of the Brahmaputra River and its tributary, the Bharalu, Guwahati experiences frequent inundation in low-lying localities such as Anil Nagar and Zoo Road. These events are aggravated by rapid urban sprawl, poor drainage maintenance, and intense monsoonal rainfall [31], [8]. Reports describe floodwaters submerging key government establishments such as the State Assembly Secretariat, with some neighborhoods reporting waist-to-neck-deep water levels [8], [32]. Such recurring floods not only cause economic and infrastructural damage but also disrupt essential services and daily life.

Although complete prevention of urban floods is infeasible, their impacts can be mitigated through early-warning systems, efficient drainage monitoring, and integration of modern technologies [9]. Against this backdrop, researchers have explored IoT-based sensing, machine learning, and computer vision as emerging tools for real-time flood detection, prediction, and mitigation. The following sections review existing literature across three domains most relevant to this study: IoT-based flood monitoring frameworks, machine learning for rainfall and flood prediction, and computer vision for blockage detection in urban drainage systems.

A. IoT-Based Approaches for Urban Flood Monitoring

The Internet of Things (IoT) has been recognized as a transformative technology for environmental monitoring and disaster preparedness. By deploying water-level, rainfall, humidity, and flow sensors, IoT frameworks enable continuous, high-resolution data acquisition [4], [10], [11], [33]-[35]. LoRa/LoRaWAN has been widely adopted for its long-range, low-power communication capabilities, making it suitable for flood-prone or low-connectivity areas [36], [37]. Several studies have demonstrated IoT-based systems for flood monitoring and risk mitigation: Arshad et al. [38] emphasized the importance of IoT sensors for evacuation planning; Yang and Chang [39] integrated IoT and ML for regional inundation prediction; Li et al. [40] applied IoT with GIS for subway flood monitoring; Vitry et al. [41] introduced FloodX, combining alternative sensors and computer vision; and Soh et al. [42] developed an IoT-cloud system using image processing for severity assessment. More recently, Samikwa et al. [43] designed lightweight AI algorithms deployable on edge devices like Raspberry Pi.

Complementary to these, Bakhsh et al. [83] developed a flood forecasting model that integrates Wireless Sensor Networks (WSN), GIS, and Artificial Neural Networks (ANN). Their approach demonstrated that WSN-based solutions are not only cost-effective but also suitable for developing countries, though accuracy improvements remain necessary. Extending the scope from monitoring to prevention, Muniandy et al. [89] proposed an IoT-enabled flood prevention system capable of autonomously regulating river water flow and quality through real-time sensor networks. Their framework emphasizes preventive strategies, aligning with the Sustainable Development Goals (SDGs) on water management and climate resilience.

B. Machine Learning for Rainfall and Flood Prediction

Climate change has intensified rainfall variability, significantly increasing the frequency and severity of flash floods worldwide [44]–[58]. This challenge is particularly acute in South Asia, where the Northeast region of India experiences highly erratic precipitation patterns driven by its complex topography and monsoon dynamics. States such as Assam and Meghalaya are especially vulnerable, receiving some of the heaviest rainfall globally and experiencing recurrent flood events that disrupt infrastructure, livelihoods, and ecosystems [50]–[56]. Traditional forecasting approaches based on statistical correlations or hydrological models have made important contributions but often fail to capture the nonlinear, localized, and rapidly evolving dynamics of flash floods. Their reliance on historical datasets with coarse spatial and temporal resolution limits their ability to predict sudden high-intensity rainfall bursts, which are increasingly common under climate change.

Machine learning (ML) has emerged as a promising alternative because of its ability to model complex nonlinear relationships between environmental variables. Early applications in flood prediction applied supervised models such as Random Forests (RF), Support Vector Machines (SVM), and Artificial Neural Networks (ANN), which demonstrated stronger predictive accuracy compared to conventional regression and statistical models [61]–[64]. Ensemble methods have further improved robustness, enabling the integration of multiple weak learners to achieve higher accuracy across diverse geographies. Yang et al. [59] employed ensemble models for summer rainfall prediction in China, while Basha et al. [60] conducted a comparative analysis of deep learning and conventional ML, showing that deep models such as Long Short-Term Memory (LSTM) networks consistently outperformed classical algorithms. Similarly, Anochi et al. [65] applied ML techniques to South America, and Zhang and Ye [66] benchmarked 21 algorithms globally, both studies confirming ML's superiority over general circulation and climatological models in precipitation forecasting.

Recent works have expanded this field by exploring hybrid architectures and intelligent decision-support systems. Ali et al. [86] applied deep learning models—LSTM, Bi-LSTM, and Deep Recurrent Neural Networks (DRNN)—to predict floods in Malaysia. Their results showed that DRNN, enhanced with layer normalization and Leaky ReLU activation, provided the highest predictive accuracy and required less training time compared to LSTM and Bi-LSTM, making it more suitable for real-time applications. Complementing this purely data-driven approach, Marouane [80] introduced a distributed flood early warning framework using Multi-Agent Systems (MAS) and the Anytime Algorithm. By integrating GIS, hydrodynamic, and remote sensing data, this system reduced forecasting response times nearly fivefold, demonstrating the importance of intelligent architectures for real-time disaster management. Similarly, Md Rashid et al. [82] extended MAS-based frameworks by combining them with Case-Based Reasoning (CBR) to develop a Flood Early Warning and Response System (FEWRS). This framework not only improved technical accuracy but also explicitly considered socio-economic, environmental, and governance aspects of flood management, highlighting the need for community-centered resilience strategies.

Taken together, these studies show a clear evolution from statistical and rule-based approaches toward deep learning, hybrid, and agent-based frameworks for rainfall and flood prediction. However, key limitations remain: most ML models still rely on historical or localized datasets that do not fully capture sudden extreme events, computationally intensive deep learning approaches often hinder real-time deployment in resource-limited settings, and socio-technical integration into governance structures is still in its infancy. These gaps point toward the necessity of developing lightweight, real-time AI–IoT systems that combine efficient ML models with sensor-based data acquisition, ultimately ensuring scalable, accurate, and actionable flood prediction for vulnerable regions.

TABLE I. IOT AND ML-BASED APPROACHES FOR FLOOD/RAINFALL PREDICTION (SELECTED WORKS)

Reference #	Methods Used	IoT Device	Accuracy/Results	Key Insights / Limitations
[78]	Neuro-Fuzzy + LSTM + GA	DPS310, DHT22, Rain sensor	92.9% (Hybrid)	Advanced hybrid ML; no field IoT validation.
[77]	BiLSTM, LSTM, ARIMA	Weather sensors	BiLSTM: 92%	Accurate but resource-intensive.
[69]	Logistic Regression	DHT-11	92%	Lightweight, but scalability limited.
[68]	DT, RF, Naïve Bayes	DHT-11, BMP-180	DT: 95%, NB: 83%	Focused accuracy; lacks real-world validation.
[67]	LR, SVM, RF, DT	DHT-11, BMP-180	Up to 95% (DT)	Small-scale IoT testbed.
[71]	LSTM + PSO	NA	94%	Deep learning, computationally heavy.
[70]	MLR, Logistic Regression	Arduino + LM35	~81%	Low-cost but limited accuracy.
[74]	SVM, KNN, ANN, DNN	GPS-enabled IoT devices	ANN: 89%	Integrated sensors; energy concerns.
[73]	KNN, DT, RF, LR	IoT weather station	RF error: 0.083	IoT–ML integration; localized only.
[72]	ANN variants (FFNN, RNN, ENN)	NA	MAE: 0.54 (ENN best)	Neural nets effective; no IoT link.

The works presented in Table I demonstrate the steady evolution of IoT and machine learning applications for flood and rainfall prediction. Early studies primarily relied on simple statistical and regression models [75], but more recent works have shifted towards neural networks, deep learning, and hybrid architectures [77], [78], [76]. Distributed and agent-based frameworks [80], [92] have further advanced the field by improving decision-making and integrating governance perspectives, while IoT-driven monitoring and prevention solutions [99], [112] highlight the move toward practical, cost-effective deployment. Despite these advances, most frameworks remain constrained to localized datasets, simulation-based validation, or resource-intensive models, underscoring the need for lightweight, field-deployable AI-IoT systems that integrate hydrological sensing with infrastructure monitoring.

C. Computer Vision for Blockage Detection in Urban Drainage Systems

Blockages in culverts and drainage channels are a critical yet underexplored factor in urban flooding, as even well-designed drainage systems can fail when debris accumulation reduces their flow capacity. Unlike hydrological or rainfall-based causes of flooding, blockage issues often arise from localized debris such as vegetation, sediments, plastics, and boulders, which can obstruct culvert inlets or narrow drainage passages. Early investigations by Balkham et al. [79] addressed this problem through a risk-based framework in the United Kingdom, providing local-scale guidelines for blockage management, while Kramer et al. [81] extended this line of work by introducing mathematical models and laboratory-scale experiments to quantify the effects of different debris types and alignments. Although valuable, these approaches were often limited by oversimplification, with restricted applicability to real-world conditions. Field-supported laboratory evidence by Brooks [12] later confirmed that culvert inlets are the primary points of debris deposition, particularly boulders, thereby validating the vulnerability of structural entry points during intense flows.

With the advent of advanced computing, researchers shifted toward AI-driven solutions for blockage assessment. Iqbal et al. [13] proposed computer vision methods using convolutional neural networks (CNNs) to automate blockage classification, testing architectures such as NASNet, MobileNet, and others on real-world and laboratory datasets. Their results demonstrated that NASNet achieved the highest accuracy of around 85%, while MobileNet provided faster inference speeds, making it suitable for near-real-time applications. In a follow-up study, Iqbal et al. [14] extended this work by integrating CNN-extracted features into regression models such as Artificial Neural Networks (ANNs) to predict hydraulic blockage, thereby attempting to link visual blockage data with quantitative hydrological behavior. This represented a significant step forward, as it bridged the gap between visual inspection and hydraulic modeling.

Despite these advances, challenges remain. Computer vision models often struggle with background clutter, varying illumination, occlusion of culvert openings, and the scarcity of high-quality, annotated datasets. Furthermore, most existing approaches have been validated only in laboratory conditions or using limited synthetic datasets, raising concerns about their robustness in complex real-world environments. Another critical gap is the lack of integration between image-based blockage detection systems and IoT sensor networks, which could enable holistic urban flood monitoring by combining hydraulic, meteorological, and infrastructural indicators. Addressing these challenges through the development of scalable, data-rich, and integrated AI–IoT frameworks is essential for achieving reliable and proactive blockage detection in urban flood management.

As shown in Table II, research on blockage detection in culverts and drainage systems has advanced from conceptual risk frameworks [79], [81] to AI-driven visual recognition methods [13], [14]. More recent contributions employ deep learning pipelines, including segmentation-classification approaches and semantic segmentation architectures such as SHARP-Net. Despite notable improvements in accuracy, most of these studies remain limited to controlled laboratory experiments or narrowly scoped datasets (e.g. ICOB, VHD, S-BIRD). Real-world deployment challenges—including background clutter, occlusions, lighting variability, and lack of integration with IoT sensor networks—remain largely unresolved. These limitations highlight a critical research gap: the need for a scalable and integrated system that combines IoT-based sensing with image-based blockage detection for comprehensive urban flood monitoring.

TABLE II. COMPUTER VISION AND ML APPROACHES FOR BLOCKAGE DETECTION (SELECTED WORKS)

Reference #	Methods Used	Dataset	Accuracy/Results	Key Insights / Limitations
[86]	SHARP-Net (semantic segmentation)	Culvert-Sewer, DeepGlobe	IoU: 94.7%	Struggles under occlusion.
[87]	Siamese NN + Binary classifier	80k UK trash screen images	Acc: 91%, AUC: 0.98	Needs hydro model integration.
[88]	Logistic Regression	Trash screen CCTV	88%	Dataset bias (80% blocked).
[85]	YOLOv5	S-BIRD sewer dataset	mAP 96.3%	Sewer-focused, not culverts.
[84]	KNN, ANN, SVR, 1D-CNN	HBD dataset	ANN: $R^2 = 0.95$	Strong prediction; lab-scale only.
[14]	ANN + MobileNet features	Lab culvert dataset	$R^2 = 0.7855$	Weak link visual ↔ hydraulic.
[13]	CNNs (NASNet, MobileNet, etc.)	ICOB, VHD, SIC	NASNet: 85%, MobileNet: 78%	Clutter & labeling issues.
[12]	Lab + field blockage study	Culvert inlet models	Inlets = primary deposition	Focused only on boulders.
[81]	Mathematical model + experiments	Lab culvert debris tests	Showed debris type/alignment effects	Simplified; lacked real-world validation.
[79]	Risk-based blockage framework	Field data (UK culverts)	Guidelines for blockage mgmt.	Early risk framework; no automation.

III. Proposed Methodology

A. Study Area

The study was conducted in Anil Nagar and Zoo Road localities of Guwahati City, Assam, India, one of the most flood-prone urban centers in Northeast India. Guwahati lies along the Bharalu River, a tributary of the Brahmaputra, and experiences a humid subtropical climate with intense monsoonal rainfall (June–September). Anil Nagar, a densely populated residential and commercial zone with low-lying terrain, frequently suffers from flash floods due to inadequate drainage and rapid urbanization. Heavy rainfall causes sharp rises in water levels, leading to recurrent inundation, waterlogging, and economic losses. For this study, a pilot area in Kamrup Metropolitan District was selected to test the proposed IoT- and vision-based flood monitoring system. The selected sites are shown in Fig. 1.

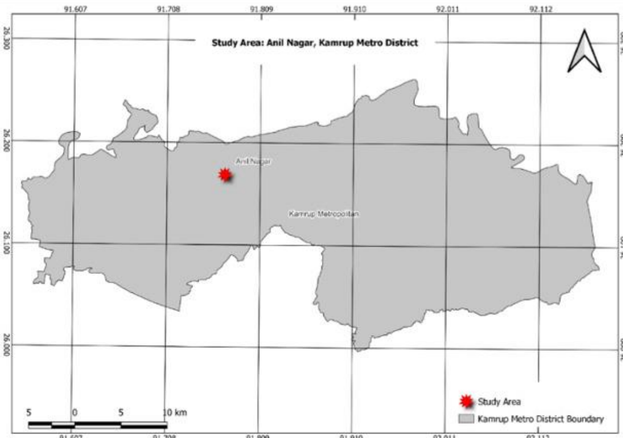


Fig. 1. Study area map: Anil Nagar in Kamrup Metro District.

B. Overall Architecture

The overall process flow of the proposed methodology is illustrated in Fig. 2. The framework integrates IoT-based sensing, machine learning-driven rainfall prediction, and computer vision-based blockage detection into a unified flood monitoring and early warning system.

A LoRa-based IoT network was deployed across the study area using two types of devices—water sensing nodes and base stations. Water sensing nodes monitor drainage water levels, while base stations collect meteorological parameters such as rainfall, temperature, and air pressure. These devices transmit real-time data wirelessly for further processing.

The acquired IoT data, along with historical datasets from external APIs, are pre-processed and analyzed using multiple machine learning algorithms for rainfall prediction. Their outputs are refined through a fuzzy logic-based fusion framework, which demonstrated superior accuracy compared to individual classifiers. By executing these tasks at the edge, the system achieves low latency, improved reliability, and reduced dependence on cloud-only computation.

At the cloud layer, images of strategic canal locations where blockages are most likely to occur are captured and uploaded by users through the mobile application. The system incorporates geofencing, which triggers an alert when users enter blockage-prone zones, prompting them to capture and submit images. Importantly, this process is designed to occur before flood events so that blockages can be identified proactively when rainfall is predicted, thereby minimizing reliance on user activity during emergencies. Once uploaded, these images are processed on the server using computer vision-based deep learning models to detect and classify blockages. By integrating blockage detection results with rainfall forecasts and IoT-based monitoring data, the cloud layer enables a comprehensive flash flood prediction and early warning system for the selected localities.

Finally, the mobile application and web dashboard provide stakeholders with real-time visualizations, alerts, and decision-support tools. The app displays live water levels, rainfall forecasts, and blockage alerts, while also offering geofencing-based warnings and safe-route suggestions. The dashboard aggregates all information, ensuring that authorities and citizens have timely access to actionable flood intelligence.

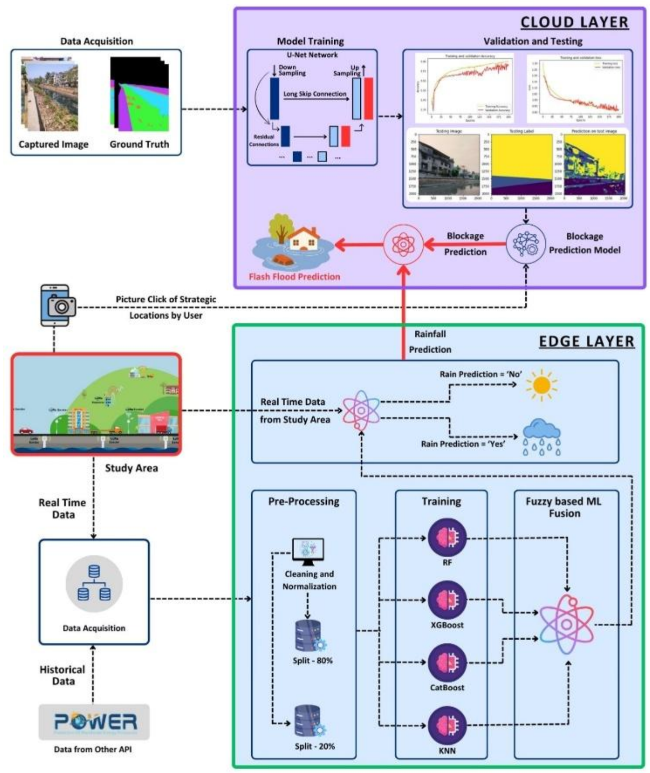


Fig. 2. Overall architecture of the proposed work.

C. IoT Framework and Hardware Design

The IoT framework was designed to provide real-time monitoring of drainage water levels and meteorological parameters within the study area. The system comprises two main devices: water sensing nodes and base stations.

The water sensing nodes are compact instruments deployed along roadside drains and low-lying urban locations to detect potential flash flood events. Each node is equipped with a contact-type water level sensor, enclosed within PVC pipes positioned 200 cm apart for accurate depth measurements. To enable reliable communication in flood-prone environments, the nodes employ LoRa transceivers, which facilitate real-time data transmission across long ranges. Unlike conventional IoT devices dependent on internet access, this system remains operational even during disasters when internet connectivity is disrupted, thereby ensuring uninterrupted flood monitoring. Designed for autonomous operation, the nodes are powered by rechargeable battery packs integrated with solar panels, eliminating dependence on conventional power supplies. This ensures continuous functionality even during prolonged power outages, often associated with flood scenarios.

The base stations, developed using Raspberry Pi 4 boards, serve as gateways for aggregating data from multiple sensing nodes. In addition to LoRa receivers, they are equipped with BMP180 pressure sensors, tipping rain gauges, anemometers, and temperature–humidity sensors, enabling on-site weather monitoring. These parameters were purposely included because relying solely on weather APIs can be misleading, as such services typically provide averaged values over large geographic areas. In reality, micro-climatic variations in parameters such as rainfall, wind speed, and humidity can occur within ranges as small as 4–5 km. For flash flood predictions, localized rainfall data is particularly critical, since heavy rainfall may occur in one neighborhood while an adjacent area just 2 km away remains unaffected. The incorporation of direct measurements at the base stations, therefore, ensures high-resolution, site-specific meteorological data, significantly improving the accuracy of the overall flood monitoring and prediction system.

This hardware architecture, integrating low-power LoRa communication with solar-powered autonomy and localized weather sensing, establishes a scalable, resilient, and energy-efficient flood monitoring system suitable for deployment in vulnerable urban environments. Fig. 3 illustrates the deployment of IoT sensors in the study area, showing the placement of water sensing nodes and base stations across critical flood-prone locations.

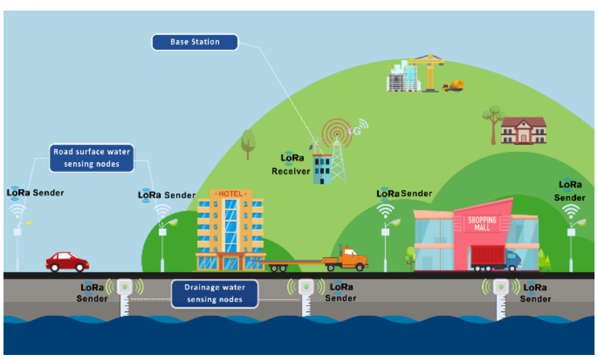


Fig. 3. Deployment of nodes and base stations.

D. Edge-Based Rainfall Prediction Using Machine Learning Fusion

Accurate rainfall prediction is essential for flash flood forecasting, as localized precipitation patterns often determine the severity of inundation. In this study, rainfall prediction was developed by integrating historical meteorological datasets with real-time IoT sensor data collected from the deployed framework in the study area. Ten years of historical data were obtained from the Indian Meteorological Department (IMD) and the NASA POWER database, covering rainfall, temperature, humidity, and pressure parameters. These datasets were merged with on-site measurements from IoT base stations to provide both temporal depth and spatial specificity, addressing limitations of weather APIs that only provide coarse, area-wide averages. As the IoT base stations continue to record location-specific data, the dataset will grow continuously, enabling future retraining of the models. This evolving dataset is expected to improve predictive accuracy and adaptability, including unprecedented rainfall patterns that may arise due to climate change.

The complete dataset comprised 91,991 instances with seven features (six independent variables and one dependent variable). Pre-processing involved three major steps: 1) cleaning, where missing values were handled using mean imputation; 2) normalization, which standardized feature ranges; and 3) splitting, where the dataset was divided into training and testing subsets in an 80:20 ratio. These steps enhanced the classifiers' ability to learn effectively and minimized bias.

Four machine learning algorithms were evaluated for rainfall prediction: Random Forest (RF), Extreme Gradient Boosting (XGBoost), Categorical Boosting (CatBoost), and K-Nearest Neighbors (KNN). These models were iteratively optimized during both training and testing to capture the nonlinear and multivariate patterns of rainfall events. To further improve reliability, the outputs from individual classifiers were passed through a fuzzy logic-based fusion framework. Rather than assigning rainfall into discrete categories, the fuzzy layer refines the combined outputs of multiple models to increase prediction probability and confidence. A decision matrix (Table III) was designed to integrate model outputs into a unified outcome.

TABLE III. INPUT INTO THE FUZZY LAYER FOR THE PROPOSED MODEL

Individual Model Prediction				**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**

A key feature of this work is that the machine learning models and the fuzzy fusion framework were deployed on Raspberry Pi devices at the edge layer. This deployment enables on-site rainfall prediction with low latency, reducing dependence on cloud servers and ensuring functionality even in disaster scenarios where internet connectivity may be disrupted. Edge-level prediction thus allows the system to provide faster and more reliable early warnings, an essential requirement for flash flood-prone urban environments. Moreover, the architecture has been designed in a modular manner, allowing additional nodes, base stations, or processing units to be integrated seamlessly. This modularity supports future scalability to larger urban areas or more complex drainage networks without fundamentally altering the core framework, thereby making the system adaptable for broader deployment.

The experimental results demonstrated that the fuzzy fusion model consistently outperformed standalone classifiers, producing more stable and accurate rainfall forecasts while reducing false positives. As illustrated in Fig. 4, by combining IoT-based observations, historical meteorological records, and intelligent fusion of multiple classifiers at the edge, this rainfall prediction module forms a critical component of the proposed framework, directly supporting an integrated flash flood early warning system for Guwahati's vulnerable localities.

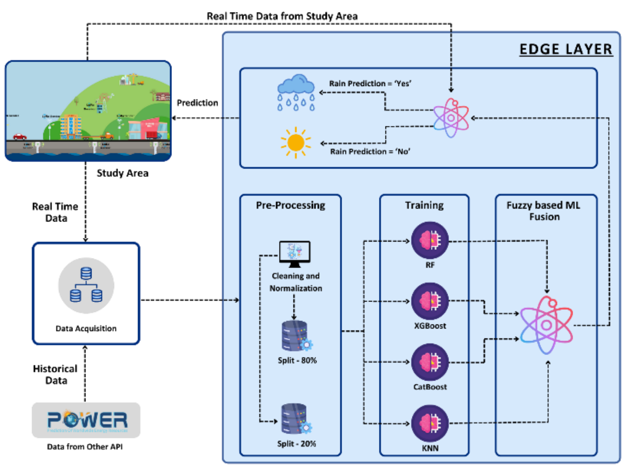


Fig. 4. Workflow of the rainfall prediction framework integrating IoT data and historical meteorological datasets. The figure illustrates data pre-processing, machine learning classification (RF, XGBoost, CatBoost, KNN), and fuzzy logic-based fusion, all deployed on Raspberry Pi devices at the edge layer to enable real-time, low-latency rainfall prediction and early warning.

E. Computer Vision–Based Blockage Detection

Blockages in urban drainage canals significantly reduce hydraulic capacity and are among the leading causes of flash flooding in Guwahati. To address this, a computer vision-based blockage detection module was developed and integrated into the proposed framework. A custom dataset comprising 1,130 high-resolution images of drainage canals and outlets was collected from strategic locations within the Anil Nagar and Zoo Road localities. Images were captured using smartphone cameras (Realme 9 Pro 5G) under varying lighting and environmental conditions to represent real-world variability. Each image, originally sized at 3468 × 4624 pixels, was resized to 1920 × 2560 pixels and then divided into non-overlapping patches of 512 × 512 pixels, yielding a total of 16,950 samples for training and evaluation. The dataset was annotated using the CVAT (Computer Vision Annotation Tool), where each patch was manually segmented into four classes (Table IV):

TABLE IV. SEGMENTATION CLASS LABLES

Sl. No.	Class	Color Labels
1	Water	#33DDFF
2	Boundary	#B83DF5
3	Blockage	#FF6A4D
4	Background	#000000

The careful assignment of these classes ensured that canal regions, potential obstruction zones, and their surroundings were distinctly represented, enabling the model to effectively learn spatial differences between clear flow areas and blocked sections. Fig. 5 shows sample annotated images, where the original canal image is placed on the left and the corresponding segmentation mask on the right, highlighting the four defined classes.

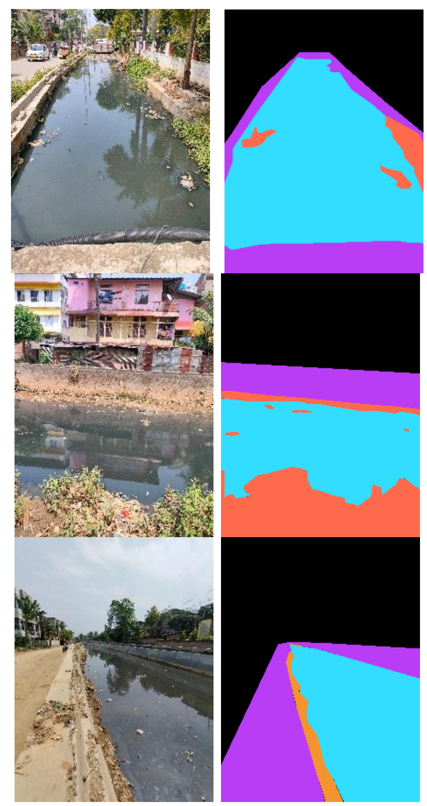


Fig. 5. Sample dataset images with annotations. Left: original canal images. Right: annotated segmentation masks showing four classes (Water, Boundary, Blockage, Background).

The original images were captured at a high resolution of 3468 × 4624 pixels. Due to computational resource limitations, they were first resized to 1920 × 2560 pixels and subsequently divided into patches of 512 × 512 pixels, producing a total of 16,950 samples. This approach preserved greater global context compared to lower-resolution alternatives, while ensuring manageable input sizes for the models. The resulting patches were normalized and augmented using random rotation, brightness adjustment, flipping, and noise injection. The dataset was then split into 70% training, 15% validation, and 15% testing subsets.

For segmentation, four deep learning models were trained and evaluated to identify blockages: a U-Net with EfficientNetB0 backbone, a U-Net with ResNet50 backbone, a U-Net with InceptionV3 backbone, and a U-Net with MobileNetV2 backbone. The U-Net framework was selected for its strong performance in semantic segmentation, while the different backbones were used to investigate trade-offs in feature representation, computational efficiency, and accuracy. All models were trained under identical conditions to ensure fair comparison.

As shown in Fig. 6, the encoder backbone incorporates MBConv blocks, where each block first expands the number of channels using 1×1 convolutions, denoted as Ex(.) in Eq. (3). This is followed by a Depthwise Convolution D(.) and refinement using a Squeeze-and-Excitation attention module Se(.). Finally, another set of 1×1 convolutions, represented as Cn(.), compresses the channel count back, forming the well-known "inverted bottleneck" structure. Residual connections are integrated to improve gradient propagation and training stability.

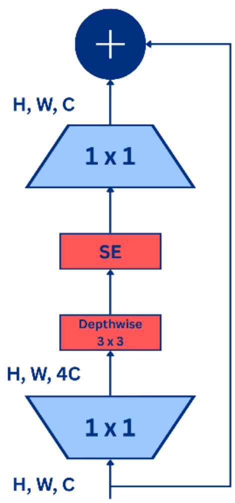


Fig. 6. Illustration of the MBConv block and skip connection mechanism used in the encoder–decoder of U-Net backbones.

The decoder mirrors this design, with upsampling operations that reconnect to the encoder via skip connections. Eq. (3) and (4) describe the feature propagation in the encoder and decoder stages. Here, X_{En}^{i} denotes the feature map produced by the i^{th} MBConv encoder block, while X_{De}^{j} and X_{En}^{j} represent the feature maps from the j^{th} decoder and encoder blocks, respectively. Skip connections are combined using the CBAR(.) function, which merges the encoder's feature map with the upsampled feature map from the preceding decoder block. The merged output is then processed through a 3×3 convolution (C3×3(.)), batch normalization (BN(.)), an activation function δ(.), and a residual connection, as expressed in Eq. (5). This design ensures better gradient flow, higher feature reuse, and robust segmentation performance.

All models were implemented in TensorFlow 2.10 and trained using an Nvidia RTX 3080 GPU (10,496 CUDA cores, 10GB GDDR6X memory). Training was performed for 100 epochs with a batch size of 16, using the Nadam optimizer (learning rate = 0.001). A hybrid loss function combining Dice Loss and Focal Loss was applied to maximize segmentation overlap and address class imbalance.

Once deployed on the cloud server, the trained models process geotagged images uploaded via the mobile application. When a user enters a geofenced high-risk drainage zone, the application prompts them to capture and submit canal images. The models then classify the canal as "blocked" or "clear". Hosting the computer vision models on the cloud ensures not only centralized processing but also inherent scalability, as elastic computing resources can accommodate larger datasets and concurrent image submissions from wider geographical areas. The comparative performance of the four segmentation approaches is presented in the Results and Discussion section.

$$\mathrm{X}_{\mathrm{En}}^{\mathrm{i}} = \mathrm{Cn}\left(\mathrm{Se}\left(\mathrm{D}\left(\mathrm{Ex}\left(\mathrm{X}_{\mathrm{En}}^{\mathrm{i-1}}\right)\right)\right)\right) + \mathrm{X}_{\mathrm{En}}^{\mathrm{i-1}} \quad (1)$$

$$\mathrm{X}_{\mathrm{De}}^{\mathrm{j}} = \mathrm{CBAR}\left(\mathrm{concat}\left(\mathrm{X}_{\mathrm{En}}^{\mathrm{j}}, \mathrm{U}\left(\mathrm{X}_{\mathrm{De}}^{\mathrm{j-1}}\right)\right)\right) \quad (2)$$

$$\mathrm{CBAR}(\mathrm{X}) = \delta(\mathrm{BN}\left(\mathrm{C}_{3\mathrm{X}3}(\mathrm{X})\right) + \mathrm{X} \quad (3)$$

IV. Results and Discussion

A. IoT Deployment and Field Validation

The proposed IoT framework was deployed in Anil Nagar and Zoo Road localities of Guwahati, with 10 water sensing nodes and 2 base stations strategically placed along drainage channels and flood-prone areas. Each water sensing node was equipped with LoRa transceivers and solar-powered batteries, while the base stations collected supplementary meteorological parameters, including rainfall, temperature, humidity, and wind speed.

To ensure real-world practicality, the proposed framework employs solar-powered sensing nodes equipped with 7500 mAh lithium-ion battery packs that are continuously recharged during the day by integrated solar panels. This design ensures uninterrupted operation even during prolonged cloudy or rainy conditions, thereby addressing energy constraints often faced by IoT systems. Coupled with low-power LoRa communication modules, the nodes achieve efficient energy usage and reliable data transmission. The hardware requirements remain modest, with commodity sensors and Raspberry Pi-based edge nodes keeping costs affordable for larger deployments. The modular architecture further supports integration of heterogeneous devices, enabling scalability in dense urban environments. Fault tolerance and data reliability are enhanced through redundant data logging at both node and base-station levels, ensuring continuity of operations even in cases of sensor malfunction or communication disruptions.

To validate the field data, real-time sensor outputs were compared with the NASA POWER API and Indian Meteorological Department datasets. The results showed a high degree of consistency, with local rainfall measurements aligning closely with API estimates but offering finer spatial granularity. This confirmed that localized sensors captured micro-climatic variations more effectively than coarse-resolution APIs, a critical factor for flash flood prediction in Guwahati, where rainfall can vary significantly within 2–3 km.

A representative comparison between IoT-based rainfall measurements and NASA POWER API values for a sample week in July 2023 is presented in Table V. The results highlight that IoT sensors captured short-duration rainfall bursts and localized variations which were not fully reflected in API data, demonstrating the added value of in-situ measurements for urban flood forecasting.

TABLE V. VALIDATION OF SENSOR DATA FROM THE ZOO ROAD BASE STATION AGAINST NASA POWER API VALUES (10–16 JULY 2023)

Date	IoT Sensor Rainfall (mm)	NASA POWER API (mm)	Difference (mm)
10-Jul-23	54.2	52.8	1.4
11-Jul-23	12.6	11.9	0.7
12-Jul-23	0	0	0
13-Jul-23	38.7	40.1	–1.4
14-Jul-23	22.1	23.3	–1.2
15-Jul-23	5.8	6.4	–0.6
16-Jul-23	47.5	46.2	1.3

During the 2023 monsoon season, the deployed network successfully detected rapid increases in water levels along the Bharalu tributary, providing early indications of waterlogging in Anil Nagar. As illustrated in Fig. 7, the water sensing nodes captured a clear rise in drainage levels, validating the effectiveness of the deployed sensors. These results demonstrate the reliability of the IoT network for continuous monitoring, even under adverse weather conditions, and its capacity to generate high-resolution hydrological datasets essential for predictive modeling.

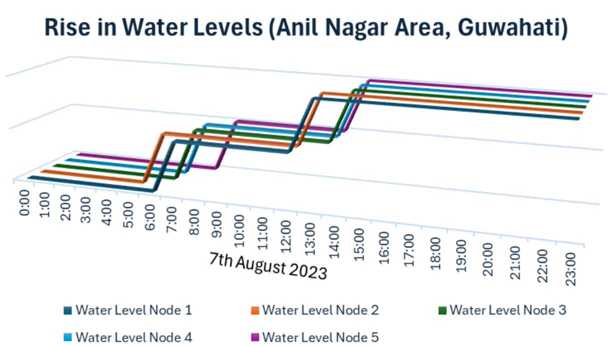
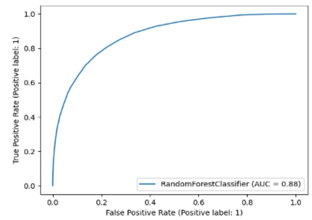


Fig. 7. Rise in water levels as depicted by the water sensing nodes

B. Rainfall Prediction Performance

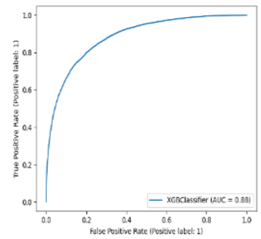
The time series data of weather parameters were analyzed for the development of the rainfall prediction system. Datasets were obtained from the Indian Meteorological Department (two years) and NASA POWER (ten years) for Guwahati city. Several classification algorithms, including Random Forest (RF), XGBoost, CatBoost, and K-Nearest Neighbors (KNN), were tested to evaluate their predictive performance.

During the testing phase, the Random Forest (RF) model correctly classified 8094 out of 9261 negative instances and 8213 out of 9138 positive instances, yielding an accuracy of 88.6% with a miss rate of 11.4%. The comparison between expected and actual outcomes is shown in Fig. 8. The XGBoost model produced similar results, with 8103 negatives and 8134 positives classified correctly, corresponding to an accuracy of 88.2% and an 11.8% miss rate, as illustrated in Fig 9. In comparison, CatBoost achieved the best standalone performance, predicting 8241 negatives and 8231 positives, resulting in an accuracy of 89.5% and a 10.5% miss rate, as shown in Fig 10. By contrast, the KNN model exhibited weaker performance, with 7213 negatives and 7311 positives correctly classified, yielding 78.9% accuracy and a 21.1% miss rate, as depicted in Fig. 11.



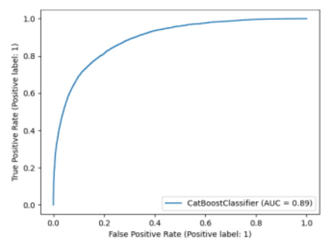
RF Testing Results			
No. of samples		N= 18399	
Input	Expected Result (E)	Output Result (O)	
		O-Negative	O-Positive
	E-Negative 9261	8094	1167
	E-Positive 9138	925	8213
	Accuracy=	0.886298168	
	Miss Rate=	0.113701832	

Fig. 8. Statistical analysis for the random forest model



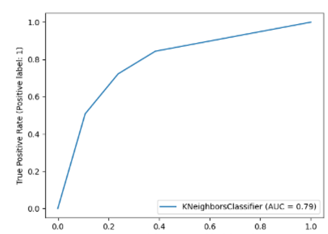
XGBoost Testing Results			
No. of samples		N= 18399	
INPUT	Expected Result (E)	Output Result (O)	
		O-Negative	O-Positive
	E-Negative 9261	8103	1158
	E-Positive 9138	1004	8134
	Accuracy=	0.882493614	
	Miss Rate=	0.117506386	

Fig. 9. Statistical analysis for a random XGBoost model



CatBoost 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	

Fig. 10. Statistical analysis for the 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	

Fig. 11. Statistical analysis for the K-Nearest Neighbor model

Proposed Fusion model 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	

Fig. 12. Statistical analysis for the proposed fusion model

Subsequently, the outputs of all classifiers were integrated using the proposed fuzzy fusion framework. This system processed the test data by considering actual class labels alongside predictions from individual classifiers. The fuzzy model identified 8347 negatives (out of 9216) and 8652 positives (out of 9138), achieving an overall accuracy of 92.4% with a miss rate of 7.6%, as illustrated in Fig. 12. This demonstrates the advantage of the fusion approach in reducing false classifications and improving robustness compared to standalone models.

A detailed summary of training and testing results across all models, along with the fuzzy fusion framework, is presented in Table VI. Notably, the fused system consistently outperformed the four individual classifiers, confirming its effectiveness in rainfall prediction. Furthermore, Table VI also includes a comparative evaluation against previously reported rainfall prediction approaches, highlighting the superior accuracy and reduced miss rate of the proposed methodology.

TABLE VI. COMPARISON OF MACHINE LEARNING MODELS AND FUZZY FUSION FRAMEWORK FOR RAINFALL PREDICTION

Model	Accuracy (%)	Miss Rate	Remarks
Random Forest (RF)	88.6	0.14	Strong generalization, moderate error rate
XGBoost	88.2	0.12	Effective but prone to overfitting with small samples
CatBoost	89.5	0.1	Best standalone classifier
KNN	78.9	0.21	Sensitive to noise, weakest performance
Fuzzy Fusion	92.4	0.08	Highest accuracy, reduced false predictions

C. Blockage Detection via Deep Learning

The efficacy of various deep learning architectures was evaluated for the task of blockage detection in canal networks. Four models, like U-Net with EfficientNetB0 backbone, U-Net with ResNet50 backbone, U-Net with InceptionV3 backbone, and U-Net with MobileNetV2 backbone were trained and tested on a dataset of 1,130 high-resolution images (3468 × 4624 pixels) annotated into blockage and non-blockage classes. To address computational constraints and increase the effective dataset size, the images were resized to 1920 × 2560 pixels and divided into non-overlapping 512 × 512 patches, resulting in 16,950 samples. These patches were normalized and augmented to enhance robustness and variability. Model training was conducted in a high-performance GPU environment (NVIDIA RTX 3080), enabling efficient parallel computation and accelerated convergence. The performance of the models was evaluated using standard metrics, including Jaccard's coefficient (IoU), mean Intersection-over-Union (mIoU), Dice coefficient, pixel accuracy, precision, and error rate. The U-Net with EfficientNetB0 backbone achieved the best results, with a pixel-wise accuracy of 91%, Dice score of 86%, and mIoU of 78%. This superior performance can be attributed to EfficientNet's inverted bottleneck blocks and squeeze-and-excitation modules, which provided effective feature extraction while retaining spatial details through U-Net's skip connections. The combination allowed the model to segment small or irregular blockages with higher precision than other backbones.

$$\text{IoU} = \frac{TP}{TP+FP+FN} \tag{4}$$

$$\text{mean IoU} = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i+FP_i+FN_i} \tag{5}$$

$$\text{Dice} = \frac{2 \cdot TP}{2 \cdot TP+FP+FN} \tag{6}$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{7}$$

$$\text{Error Rate} = \frac{FP+FN}{TP+TN+FP+FN} \tag{8}$$

The terms TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative, respectively, and are fundamental in understanding the performance of classification models.

The ResNet50 backbone ranked second, with an accuracy of 85%. Its deep residual connections supported robust feature extraction and strong generalization, but the absence of symmetric skip connections in its native architecture limited precise boundary reconstruction. InceptionV3 achieved 72% accuracy, leveraging multi-scale convolutional kernels for contextual understanding but failing to produce sharp pixel-wise predictions due to the lack of a dense decoder path. At the lower end, MobileNetV2 achieved only 65% accuracy, reflecting its limited representational power as a lightweight architecture optimized for efficiency rather than dense segmentation.

Dataset characteristics also influenced performance. The modest dataset size (1130 images) and class imbalance, where blockage pixels formed only a small fraction of each image, caused models to be biased toward background predictions. Manual annotations introduced additional variability, particularly in ambiguous regions such as shadows or partially submerged debris. While EfficientNetB0-U-Net was relatively robust to such noise, MobileNetV2 and InceptionV3 were more affected. Resizing high-resolution images to 512 × 512 patches also led to loss of fine details, reducing the ability of weaker backbones to detect small debris.

TABLE VII. COMPARATIVE ANALYSIS OF EXISTING CLASSIFICATION MODELS WITH THE SEGMENTATION-BASED MODELS

Model	Pixel Accuracy	Mean IOU	Dice Coefficient	Precision	Recall
MobileNetV2	65%	40%	55%	58%	52%
InceptionV3	72%	50%	63%	65%	60%
ResNet-50	85%	68%	78%	80%	75%
EffecientNetB0	**91%**	**78%**	**86%**	**88%**	**85%**

Training and validation accuracy curves over 100 epochs (Fig. 13) show that the EfficientNetB0-U-Net achieved smooth convergence, with training accuracy reaching ~93% and validation accuracy ~90%, indicating minimal overfitting. ResNet50 also demonstrated stable learning, converging around 87% training accuracy and 84% validation accuracy. InceptionV3 showed slower convergence, plateauing at ~73% training accuracy and ~70% validation accuracy, while MobileNetV2 displayed severe overfitting, with training accuracy climbing to ~66% but validation accuracy stagnating at ~30%. These trends are further supported by the comparative results summarized in Table VII, which highlight the superior performance of EfficientNetB0-U-Net over the other backbones.

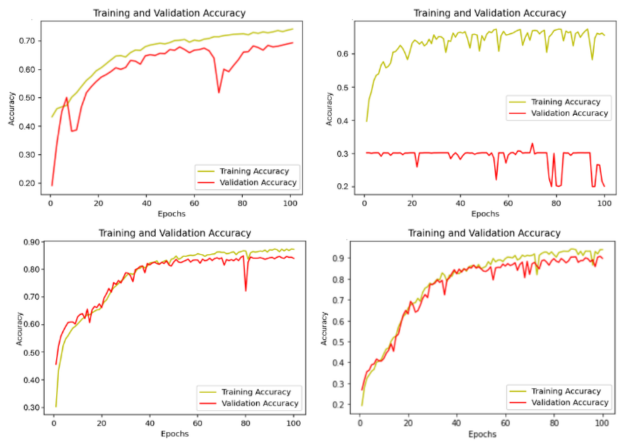


Fig. 13. Training and validation accuracy curves over 100 epochs for the four models: Top Left – MobileNetV2-U-Net; Top Right – InceptionV3-U-Net; Bottom Left – ResNet50-U-Net; Bottom Right – EfficientNetB0-U-Net

Loss curves: Fig. 14 further confirms these observations. EfficientNetB0-U-Net and ResNet50 showed steadily declining training and validation loss with minimal divergence, reflecting robust learning. InceptionV3's validation loss exhibited early instability with sharp spikes before stabilizing, while MobileNetV2 showed highly erratic validation loss throughout, indicating poor generalization.

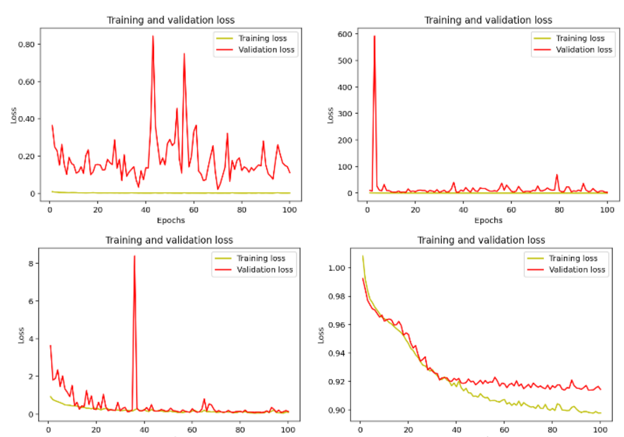


Fig. 14. Training and validation Loss curves over 100 epochs for the four models: Top Left – MobileNetV2-U-Net; Top Right – InceptionV3-U-Net; Bottom Left – ResNet50-U-Net; Bottom Right – EfficientNetB0-U-Net

Sample qualitative outputs (Fig. 15–18) further reinforce the quantitative findings. To examine model performance under different conditions, three representative test images were chosen, corresponding to low, moderate, and high blockage levels. Predictions were then generated using four models—MobileNetV2-U-Net (Fig. 15), InceptionV3-U-Net (Fig. 16), ResNet50-U-Net (Fig. 17), and EfficientNetB0-U-Net (Fig. 18)—and compared against their respective ground truth masks.

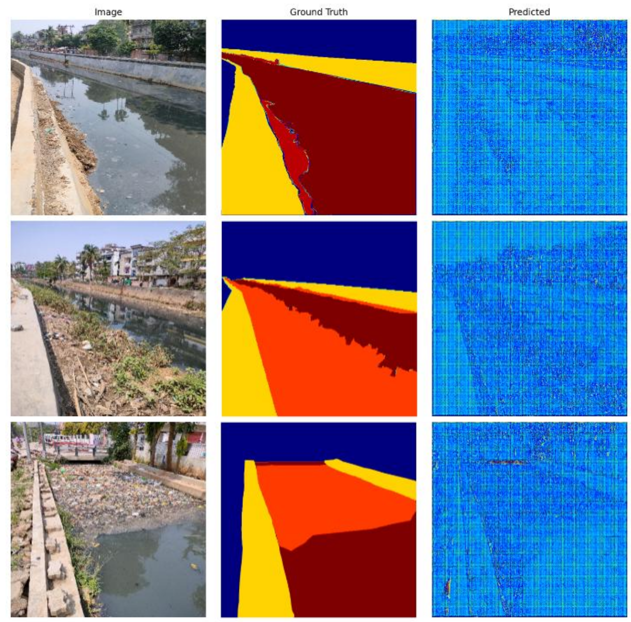


Fig. 15. Predicted segmentation outputs using MobileNetV2-U-Net. Top: low blockage; Middle: moderate blockage; Bottom: high blockage, shown alongside ground truth masks.

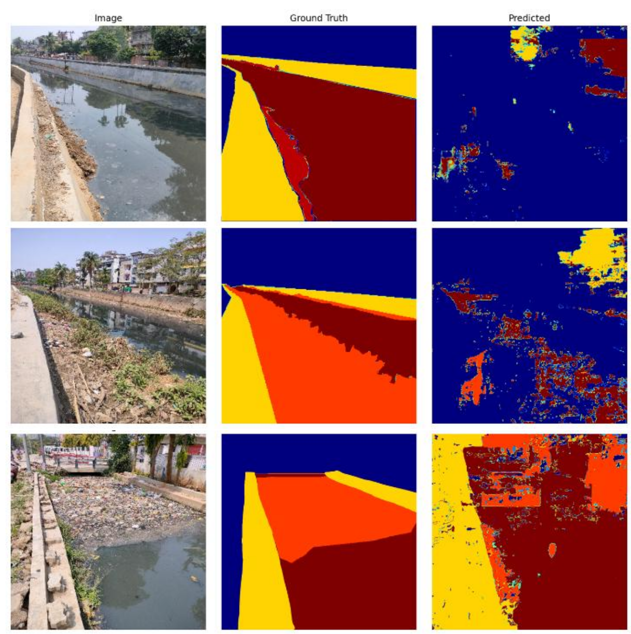


Fig. 16. Predicted segmentation outputs using InceptionV3-U-Net. Top: low blockage; Middle: moderate blockage; Bottom: high blockage, shown alongside ground truth masks.

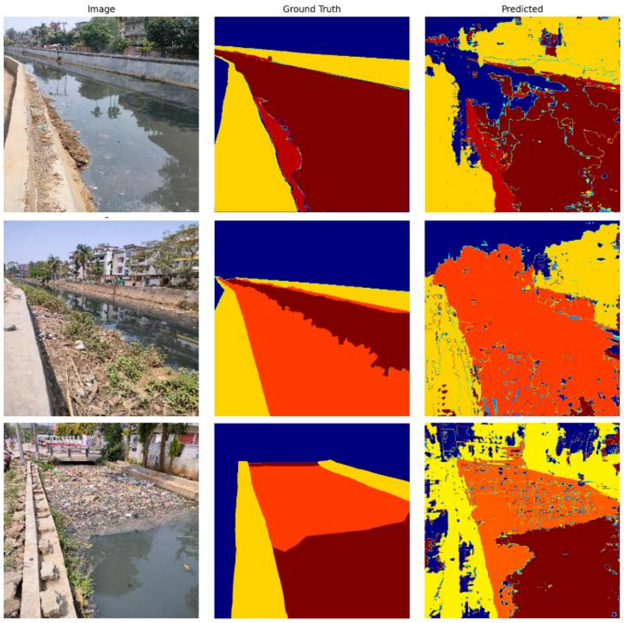


Fig. 17. Predicted segmentation outputs using ResNet50-U-Net. Top: low blockage; Middle: moderate blockage; Bottom: high blockage, shown alongside ground truth masks.

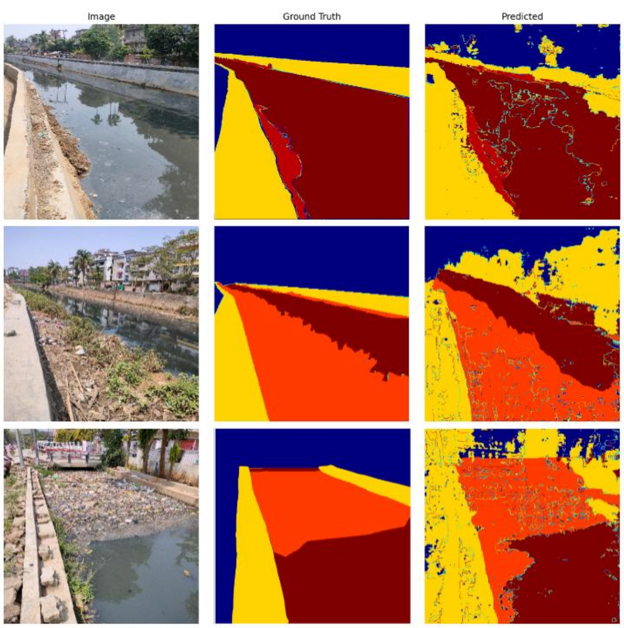


Fig. 18. Predicted segmentation outputs using EfficientNetB0-U-Net. Top: low blockage; Middle: moderate blockage; Bottom: high blockage, shown alongside ground truth masks.

Among these, EfficientNetB0-U-Net consistently produced segmentation maps that most closely resembled the ground truth. It excelled at detecting small debris, partial blockages, and irregular boundary regions, which are often the most challenging to segment in real-world canal imagery. This effectiveness can be attributed to the synergy of EfficientNet's inverted bottleneck blocks with U-Net's skip connections, which preserved fine spatial details while also capturing high-level contextual features. The ResNet50-U-Net also performed strongly, especially in delineating larger blockage regions and textured surfaces. However, its predictions often showed less precise boundaries, with minor leakage into canal walls or surrounding vegetation. InceptionV3-U-Net, by contrast, generated noticeably coarser maps, often blurring blockage boundaries and underestimating severity when debris overlapped with shadows or reflections. These limitations highlight the advantage of EfficientNetB0-U-Net for this task.

The weakest results were observed with MobileNetV2-U-Net, which frequently misclassified debris and garbage as background. While the architecture is computationally efficient and well-suited for edge devices, its reduced representational power significantly limited its performance in this task. Segmentation maps were often incomplete, failing to highlight subtle obstructions, and in many cases, the model ignored partially submerged or fine-structured debris entirely. This outcome underscores the trade-off between efficiency and accuracy: while MobileNetV2 is attractive for deployment in low-resource environments, it is not well-suited for applications requiring precise spatial resolution.

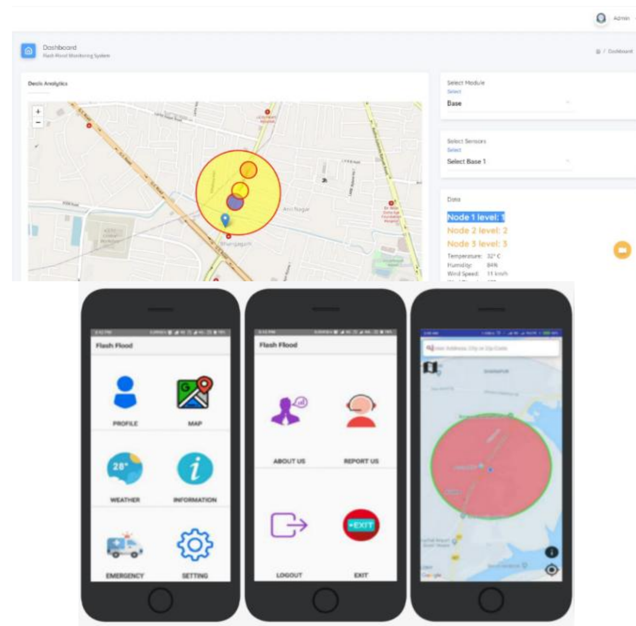


Fig. 19. Centralized dashboard (top) displaying sensor data, rainfall predictions, and visualizations, and mobile application (bottom) showing flooded areas, and emergency alerts.

A centralized dashboard and companion mobile application were developed to strengthen integration with existing flood management practices, as illustrated in Fig. 19 (dashboard at the top and mobile app at the bottom). The dashboard provides two modules: Node, showing individual sensor data, and Base Station, displaying aggregated meteorological parameters and water levels along with insights into potential flash flood events. Interactive visualizations, including a meteogram and a daily water level graph, further support decision-making. Complementing this, the mobile app integrates Google Maps to mark flooded zones, suggest alternate safe routes, and deliver emergency alerts, weather updates, and user reporting services. Together, these tools ensure that the predictive outputs of the framework can be translated into actionable information for both authorities and the community, thereby addressing the challenge of integrating novel technologies with real-world flood management infrastructure.

V. Conclusion and Future Works

This research presented a comprehensive framework for urban flood monitoring and prediction, integrating IoT sensing, edge-based machine learning, and computer vision techniques. The system was deployed and validated in flood-prone localities of Guwahati, Assam (Anil Nagar and Zoo Road), demonstrating its ability to provide real-time rainfall forecasting, drainage monitoring, and blockage detection. The LoRa-based IoT network ensured reliable data transmission even in disaster scenarios, while the edge-deployed machine learning models, enhanced with fuzzy fusion, achieved higher prediction accuracy compared to standalone classifiers. Additionally, the use of deep learning–based semantic segmentation models enabled effective identification of blockages in drainage systems, a critical but often overlooked factor in urban flooding.

The results confirmed that the EfficientNetB0-U-Net outperformed other backbones for blockage detection, while the fuzzy fusion framework consistently improved rainfall prediction reliability. Together, these contributions advance the state of real-time, low-cost, and scalable flood monitoring systems, offering a practical tool for urban disaster preparedness in data-scarce and resource-constrained environments.

Despite these advances, several limitations remain. The dataset size for blockage detection was modest, and environmental variability (lighting, occlusion, and reflections) posed challenges to generalization. Rainfall prediction relied partly on historical datasets, which may not fully capture extreme short-duration rainfall events. Furthermore, integration of IoT networks with governance and community-level decision-making remains limited.

Although the system showed promising results, there is scope for further improvement and extension in future research.

- Expanding datasets through larger-scale image collection campaigns and crowdsourced contributions to improve the robustness of blockage detection models.
- Enhancing model generalization using advanced data augmentation, transfer learning, and multimodal fusion (combining IoT, imagery, and satellite data).
- Integration with Decision-Making Systems: Extend the framework to cover multiple urban localities with varied topographies and drainage infrastructures, alongside developing stronger links with municipal authorities and community-based early warning systems. This will ensure that model outputs not only generalize across different contexts but also translate into actionable flood preparedness measures.
- Evaluation of IoT system performance under extreme weather conditions, such as heavy rainfall and strong winds, to ensure reliability during severe flood events.

Acknowledgment

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

[1] A. Gregory Jonathan N. ,Arnold,Margaret,Buys,Piet,Chen, Robert S. ,Deichmann,Uwe Klaus,Dilley, Maxx,Kjevstad, Oddvar,Lerner-Lam, Arthur L. ,Lyon, Bradfield,Yetman, "Natural disaster hotspots: A global risk analysis," World Bank. Accessed: Jul. 02, 2025. [Online]. Available: https://documents.worldbank.org/en/publication/documents-reports/documentdetail/en/621711468175150317

[2] CRED, "EM-DAT - The international disaster database." Accessed: Jul. 02, 2025. [Online]. Available: https://www.emdat.be/

[3] "Flood mapping and flood dynamics of the Mekong Delta: ENVISAT-ASAR-WSM based time series analyses." Accessed: Jul. 02, 2025. [Online]. Available: https://www.researchgate.net/publication/236029131_Flood_mapping_and_flood_dynamics_of_the_Mekong_Delta_ENVISAT-ASAR-WSM_based_time_series_analyses

[4] "A comparison of selected global disaster risk assessment results | Natural Hazards." Accessed: Jul. 02, 2025. [Online]. Available: https://link.springer.com/article/10.1007/s11069-008-9272-0

[5] K. Hansson, M. Danielson, and L. Ekenberg, "A framework for evaluation of flood management strategies," J. Environ. Manage., vol. 86, no. 3, pp. 465–480, Feb. 2008, doi: 10.1016/j.jenvman.2006.12.037.

[6] T. Tingsanchali, "Urban flood disaster management," Procedia Eng., vol. 32, pp. 25–37, Jan. 2012, doi: 10.1016/j.proeng.2012.01.1233.

[7] "Urban Floods in India," ResearchGate. Accessed: Jul. 02, 2025. [Online]. Available: https://www.researchgate.net/publication/326441140_Urban_Floods_in_India

[8] "8 Dead, 1 Missing After Flash Floods in Guwahati, Assam – FloodList." Accessed: Jul. 04, 2025. [Online]. Available: https://floodlist.com/asia/flash-floods-guwahati-assam

[9] J. D. Miller and M. Hutchins, "The impacts of urbanisation and climate change on urban flooding and urban water quality: A review of the evidence concerning the United Kingdom," J. Hydrol. Reg. Stud., vol. 12, pp. 345–362, Aug. 2017, doi: 10.1016/j.ejrh.2017.06.006.

[10] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, "A Vision of IoT: Applications, Challenges, and Opportunities With China Perspective," IEEE Internet Things J., vol. 1, no. 4, pp. 349–359, Aug. 2014, doi: 10.1109/JIOT.2014.2337336.

[11] D. Amaxilatis et al., "Advancing Experimentation-as-a-Service Through Urban IoT Experiments," IEEE Internet Things J., vol. 6, no. 2, pp. 2563–2572, Apr. 2019, doi: 10.1109/JIOT.2018.2871766.

[12] Johannes Andreas Brooks, "Culvert Blockage Caused by Boulders in the Western Cape and the Development of Mitigation Measures: Physical Model Study", [Online]. Available: https://scholar.sun.ac.za/bitstream/10019.1/108286/2/brooks_culvert_2020.pdf

[13] "A Scaled Physical Model Study of Culvert Blockage Exploring Complex Relationships Between Influential Factors: Australasian Journal of Water Resources: Vol 27 , No 1 - Get Access." Accessed: Aug. 14, 2025. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/13241583.2021.1996679

[14] "Automating Visual Blockage Classification of Culverts with Deep Learning." Accessed: Aug. 14, 2025. [Online]. Available: https://www.mdpi.com/2076-3417/11/16/7561

[15] D. Eckstein, M.-L. Hutfils, and M. Winges, Global Climate Risk Index 2019. Bonn: Germanwatch, 2019.

[16] A. Patankar, "Impacts of natural disasters on households and small businesses in India," Social Science Research Network, Rochester, NY, Dec. 23, 2019. doi: 10.2139/ssrn.3590902.

[17] H. Ali, P. Modi, and V. Mishra, "Increased flood risk in Indian sub-continent under the warming climate," Weather Clim. Extrem., vol. 25, p. 100212, Sep. 2019, doi: 10.1016/j.wace.2019.100212.

[18] "India's Water Crisis: Challenges, Solutions and Barriers," ResearchGate. Accessed: Jul. 7, 2025. [Online]. Available:

https://www.researchgate.net/publication/368363874_India's_Water_Crisis_Challenges_Solutions_and_Barriers

[19] "Hydrometeorological aspects of floods in India," Natural Hazards. Accessed: Jul. 7, 2025. [Online]. Available: https://link.springer.com/article/10.1023/A:1021199714487

[20] A. K. Gupta and S. S. Nair, "Urban floods in Bangalore and Chennai: Risk management challenges and lessons for sustainable urban ecology," Curr. Sci., vol. 100, no. 11, pp. 1638–1645, 2011.

[21] H. Nguyen, M. Babel, S. Weesakul, and N. Tripathi, "An artificial neural network model for rainfall forecasting in Bangkok, Thailand," Hydrol. Earth Syst. Sci., vol. 13, pp. 1413–1425, Aug. 2009, doi: 10.5194/hess-13-1413-2009.

[22] S. H. A. Jarah, B. Zhou, R. J. Abdullah, Y. Lu, and W. Yu, "Urbanization and urban sprawl issues in city structure: A case of the Sulaymaniah Iraqi Kurdistan Region," Sustainability, vol. 11, no. 2, Art. 485, Jan. 2019, doi: 10.3390/su11020485.

[23] I. Awakimjan, "Urban flood modelling: Recommendations for Ciudad Del Plata." Accessed: Jul. 7, 2025. [Online]. Available: https://essay.utwente.nl/68990/

[24] Z. Ahmed and D. R. M. Rao, "Urban flooding – Case study of Hyderabad," Int. J. Eng. Res. Appl., vol. 2, pp. 1–6, 2013.

[25] S. G. Sabogal, N. van de Giesen, and M. ten Veldhuis, "Can urban pluvial flooding be predicted by open spatial data and weather data?," Environ. Model. Softw., vol. 85, pp. 156–171, 2016, doi: 10.1016/j.envsoft.2016.08.007.

[26] "How does imperviousness impact the urban rainfall-runoff process under various storm cases?," ResearchGate. doi: 10.1016/j.ecolind.2015.08.041.

[27] J. Abdullah and P. Y. Julien, "Distributed flood simulations on a small tropical watershed with the TREX model," Proc. Int. Conf. Hydroinformatics, 2014.

[28] J. Abdullah, N. S. Muhammad, P. Y. Julien, J. Ariffin, and A. Shafie, "Flood flow simulations and return period calculation for the Kota Tinggi watershed, Malaysia," J. Flood Risk Manag., vol. 11, no. S2, Feb. 2018, doi: 10.1111/jfr3.12256.

[29] P. T. Coulthard, R. D. Cutler, R. J. Hardcastle, and J. Hunter, "The June 2007 floods in Hull," Proc. Inst. Civ. Eng. Water Manag., vol. 161, no. 4, pp. 163–170, 2008.

[30] W. H. M. Wan Mohtar, J. Abdullah, K. N. Abdul Maulud, and N. S. Muhammad, "Urban flash flood index based on historical rainfall events," Sustain. Cities Soc., vol. 56, p. 102088, May 2020, doi: 10.1016/j.scs.2020.102088.

[31] "Flood risk and adaptation in Indian coastal cities: Recent scenarios," Appl. Water Sci.. Accessed: Jul. 8, 2025. [Online]. Available: https://link.springer.com/article/10.1007/s13201-018-0881-9

[32] S. Borah, "Assam: Guwahati gets a big boost in its fight against water-logging," EastMojo, Oct. 16, 2019. Accessed: Dec. 10, 2019. [Online]. Available: https://www.eastmojo.com/assam/2019/10/16/assam-guwahati-gets-a-big-boost-in-its-fight-against-waterlogging

[33] "Internet of Things for smart cities," ResearchGate. doi: 10.1109/JIOT.2014.2306328.

[34] "IoT driven automated object detection algorithm for urban surveillance system in smart city," ResearchGate. doi: 10.35940/ijeat.F1317.0986S319.

[35] "Data-driven solution for optimal pumping units scheduling of smart water conservancy," ResearchGate. doi: 10.1109/JIOT.2019.2963250.

[36] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," IEEE Commun. Surv. Tutor., vol. 19, no. 2, pp. 855–873, Apr. 2017, doi: 10.1109/COMST.2017.2652320.

[37] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," ICT Express, vol. 5, no. 1, pp. 1–7, Mar. 2019, doi: 10.1016/j.icte.2017.12.005.

[38] B. Arshad, R. Ogie, J. Barthelemy, B. Pradhan, N. Verstaevel, and P. Perez, "Computer vision and IoT-based sensors in flood monitoring and mapping: A systematic review," Sensors, vol. 19, no. 22, Art. 5012, Nov. 2019, doi: 10.3390/s19225012.

[39] "Regional inundation forecasting using machine learning techniques with the Internet of Things," MDPI Water. Accessed: Jul. 9, 2025. [Online]. Available: https://www.mdpi.com/2073-4441/12/6/1578

[40] "Development and application of flood control and waterlogging prevention intelligent monitoring system based on subway 'one map'," ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., vol. X-3/W1-2022, pp. 93–100, 2022. Accessed: Jul. 9, 2025. [Online]. Available: https://isprs-annals.copernicus.org/articles/X-3-W1-2022/93/2022/

[41] "floodX: Urban flash flood experiments monitored with conventional and alternative sensors," Earth Syst. Sci. Data, vol. 9, no. 2, pp. 657–666, 2017. Accessed: Jul. 9, 2025. [Online]. Available: https://essd.copernicus.org/articles/9/657/2017/

[42] Z. H. C. Soh, H. S. Gan, K. Y. Wong, W. H. Yeo, and Y. C. Lai, "Riverbank monitoring using image processing for early flood warning system via IoT," Int. J. Integr. Eng., vol. 14, no. 3, Art. 3, Jun. 2022.

[43] E. Samikwa, T. Voigt, and J. Eriksson, "Flood prediction using IoT and artificial neural networks with edge computing," in Proc. IEEE Int. Conf. Internet Things (iThings) / GreenCom / CPSCom / SmartData / Cybermatics, Nov. 2020, pp. 234–240, doi: 10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics50389.2020.00053.

[44] A. Kumar, S. Nagar, and S. Anand, "Climate change and existential threats," in Global Climate Change, S. Singh, P. Singh, S. Rangabhashiyam, and K. K. Srivastava, Eds. Elsevier, 2021, pp. 1–31. doi: 10.1016/B978-0-12-822928-6.00005-8.

[45] "A review of the global climate change impacts, adaptation, and sustainable mitigation measures," Environ. Sci. Pollut. Res. Accessed: Jul. 12, 2025. [Online]. Available: https://link.springer.com/article/10.1007/s11356-022-19718-6

[46] K. Sami, B. A. Mohsen, K. Afef, and Z. Fouad, "Hydrological Modeling Using GIS for Mapping Flood Zones and Degree Flood Risk in Zeuss-Koutine Basin (South of Tunisia)," J. Environ. Prot., vol. 4, no. 12, Art. no. 12, Dec. 2013, doi: 10.4236/jep.2013.412161.

[47] Most. R. M. Zinat, R. Salam, M. A. Badhan, and A. R. Md. T. Islam, "Appraising drought hazard during Boro rice growing period in western Bangladesh," Int. J. Biometeorol., vol. 64, no. 10, pp. 1687–1697, Oct. 2020, doi: 10.1007/s00484-020-01949-2.

[48] Q. Yu, Y. Wang, and N. Li, "Extreme Flood Disasters: Comprehensive Impact and Assessment," Water, vol. 14, no. 8, Art. no. 8, Jan. 2022, doi: 10.3390/w14081211.

[49] B. Manandhar, S. Cui, L. Wang, and S. Shrestha, "Urban Flood Hazard Assessment and Management Practices in South Asia: A Review," Land, vol. 12, no. 3, Art. no. 3, Mar. 2023, doi: 10.3390/land12030627.

[50] F. M. Underwood, "Describing long-term trends in precipitation using generalized additive models," J. Hydrol., vol. 364, no. 3–4, Art. no. 3–4, 2009.

[51] "Distributions of Annual Maximum Rainfall Series of North-East India | Request PDF." Accessed: Jul. 23, 2025. [Online]. Available: https://www.researchgate.net/publication/279195087_Distributions_of_Annual_Maximum_Rainfall_Series_of_North-East_India

[52] "Generalized Additive Models | 7 | Statistical Models in S | Trevor J." Accessed: Jul. 23, 2025. [Online]. Available: https://www.taylorfrancis.com/chapters/edit/10.1201/9780203738535-7/generalized-additive-models-trevor-hastie

[53] "The Unquiet River: A Biography of the Brahmaputra | Oxford Academic." Accessed: Jul. 23, 2025. [Online]. Available: https://academic.oup.com/book/36968

[54] "A Historical Understanding of Assam's Floods | Economic and Political Weekly." Accessed: Jul. 23, 2025. [Online]. Available: https://www.epw.in/engage/article/historical-understanding-assams-floods

[55] N. Jamwal, "Examining Assam's disaster readiness after the 2022 floods," Dialogue Earth. Accessed: Jul. 23, 2025. [Online]. Available: https://dialogue.earth/en/climate/examining-assams-disaster-readiness-after-the-2022-floods/

[56] "Majuli Island and Assam's Rivers Face Unpredictable Flooding in 2024." Accessed: Jul. 23, 2025. [Online]. Available: https://www.downtoearth.org.in/natural-disasters/assam-floods-2024-unprecedented-timing-and-fury-grips-state

[57] "What's really behind Assam's worsening floods? | PreventionWeb." Accessed: Jul. 23, 2025. [Online]. Available: https://www.preventionweb.net/news/india-whats-really-behind-assams-worsening-floods

[58] "Sixth Assessment Report — IPCC." Accessed: Jul. 23, 2025. [Online]. Available: https://www.ipcc.ch/assessment-report/ar6/

[59] "Multi-Model Ensemble Prediction of Summer Precipitation in China Based on Machine Learning Algorithms." Accessed: Jul. 25, 2025. [Online]. Available: https://www.mdpi.com/2073-4433/13/9/1424

[60] C. Z. Basha, N. Bhavana, P. Bhavya, and S. V, "Rainfall Prediction using Machine Learning & Deep Learning Techniques," in 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Jul. 2020, pp. 92–97. doi: 10.1109/ICESC48915.2020.9155896.

[61] "Study of short term rain forecasting using machine learning based approach | Request PDF." Accessed: Jul. 25, 2025. [Online]. Available: https://www.researchgate.net/publication/336831403_Study_of_short_term_rain_forecasting_using_machine_learning_based_approach

[62] "A novel approach for precipitation forecast via improved K -nearest neighbor algorithm | Request PDF," ResearchGate, doi: 10.1016/j.aei.2017.05.003.

[63] "Precipitation Forecasting in Northern Bangladesh Using a Hybrid Machine Learning Model." Accessed: Jul. 25, 2025. [Online]. Available: https://www.mdpi.com/2071-1050/14/5/2663

[64] K. Ahmed, D. A. Sachindra, S. Shahid, Z. Iqbal, N. Nawaz, and N. Khan, "Multi-model ensemble predictions of precipitation and temperature using machine learning algorithms," Atmospheric Res., vol. 236, p. 104806, May 2020, doi: 10.1016/j.atmosres.2019.104806.

[65] "Machine Learning for Climate Precipitation Prediction Modeling over South America." Accessed: Jul. 25, 2025. [Online]. Available: https://www.mdpi.com/2072-4292/13/13/2468

[66] Y. Zhang and A. Ye, "Machine Learning for Precipitation Forecasts Postprocessing: Multimodel Comparison and Experimental Investigation," Nov. 2021, doi: 10.1175/JHM-D-21-0096.1.

[67] "Improved Numerical Weather Prediction Using IoT and Machine Learning | Request PDF," in ResearchGate. doi: 10.1007/978-981-19-8086-2_109.

[68] P. Shah, Y. A, U. Khaitan, and S. Kayalvizhi, "Weather Management System Using Machine Learning Algorithm And IOT," in 2023 International Conference on Recent Advances in Electrical, Electronics, Ubiquitous Communication, and Computational Intelligence (RAEEUCCI), Apr. 2023, pp. 1–4. doi: 10.1109/RAEEUCCI57140.2023.10134454.

[69] "Rainfall Forecasting System Using Machine Learning Technique and IoT Technology for a Localized Region." Accessed: Jul. 26, 2025. [Online]. Available: https://www.researchgate.net/publication/366777278_Rainfall_Forecasting_System_Using_Machine_Learning_Technique_and_IoT_Technology_for_a_Localized_Region

[70] "Real-Time Rainfall Prediction System Using IoT and Machine Learning | SpringerLink." Accessed: Jul. 26, 2025. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-031-23973-1_10

[71] Y. Xu et al., "Research on particle swarm optimization in LSTM neural networks for rainfall-runoff simulation," J. Hydrol., vol. 608, p. 127553, May 2022, doi: 10.1016/j.jhydrol.2022.127553.

[72] "Artificial Neural Network Models for Rainfall Prediction | European Journal of Electrical Engineering and Computer Science." Accessed: Jul. 26, 2025. [Online]. Available: https://ejece.org/index.php/ejece/article/view/313

[73] "An IoT-Based Predictive Analytics for Estimation of Rainfall for Irrigation." Accessed: Jul. 26, 2025. [Online]. Available: https://www.researchgate.net/publication/343656881_An_IoT-Based_Predictive_Analytics_for_Estimation_of_Rainfall_for_Irrigation

[74] "An Intelligent Weather Prediction System Based on IOT | IEEE Conference Publication | IEEE Xplore." Accessed: Jul. 26, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/9455883

[75] "Low Cost IoT based Flood Monitoring System Using Machine Learning and Neural Networks: Flood Alerting and Rainfall Prediction | IEEE Conference Publication | IEEE Xplore." Accessed: Jul. 26, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/9074928

[76] Ali, M. H. M., Asmai, S. A., Abidin, Z. Z., Abas, Z. A., & Emran, N. A. (2022). Flood prediction using deep learning models. International Journal of Advanced Computer Science and Applications (IJACSA), 13(9), 972–981.

[77] "Real-Time Rain Prediction in Agriculture using AI and IoT: A Bi-Directional LSTM Approach - Google Search." Accessed: Jul. 26, 2025. [Online]. Available: https://www.google.com/search?q=Real-Time+Rain+Prediction+in+Agriculture+using+AI+and+IoT%3A+A+Bi-Directional+LSTM+Approach&oq=Real-Time+Rain+Prediction+in+Agriculture+using+AI+and+IoT%3A+A+Bi-Directional+LSTM+Approach&gs_lcrp=EgZjaHJvbWUyBggAEEUYOTIGCAEQRRg8MgYIAhBFGDzSAQkxNDA0ajBqMTWoAgiwAgHxBT2T-laKvkmb&sourceid=chrome&ie=UTF-8

[78] "Development of a Secured IoT-Based Flood Monitoring and Forecasting System Using Genetic-Algorithm-Based Neuro-Fuzzy Network." Accessed: Jul. 26, 2025. [Online]. Available: https://www.mdpi.com/1424-8220/25/13/3885

[79] "CIRIA C689 Culvert design and operation guide," ResearchGate. Accessed: Aug. 14, 2025. [Online]. Available: https://www.researchgate.net/publication/370033204_CIRIA_C689_Culvert_design_and_operation_guide

[80] Marouane, E. M. (2021). Towards a Real Time Distributed Flood Early Warning System. International Journal of Advanced Computer Science and Applications (IJACSA), 12(1), 34–41.

[81] "A physical model study of culvert blockage by large urban debris," ResearchGate. Accessed: Aug. 14, 2025. [Online]. Available: https://www.researchgate.net/publication/293637298_A_physical_model_study_of_culvert_blockage_by_large_urban_debris

[82] Md Rashid, N. A., Abidin, Z. Z., & Abas, Z. A. (2024). Integrating Multi-Agent System and Case-Based Reasoning for Flood Early Warning and Response System. International Journal of Advanced Computer Science and Applications (IJACSA), 15(12), 112–120.

[83] Bakhsh, S. T., Basheri, M., Ahmed, N., & Shahzad, B. (2020). A flood forecasting model based on wireless sensor and actor networks. International Journal of Advanced Computer Science and Applications (IJACSA), 11(5), 438–446.

[84] "Regression on Deep Visual Features using Artificial Neural Networks (ANNs) to Predict Hydraulic Blockage at Culverts," ResearchGate. Accessed: Aug. 24, 2025. [Online]. Available: https://www.researchgate.net/publication/351448845_Regression_on_Deep_Visual_Features_using_Artificial_Neural_Networks_ANNs_to_Predict_Hydraulic_Blockage_at_Culverts

[85] R. R. Patil, R. K. Calay, M. Y. Mustafa, and S. M. Ansari, "AI-Driven High-Precision Model for Blockage Detection in Urban Wastewater Systems," Electronics, vol. 12, no. 17, p. 3606, Jan. 2023, doi: 10.3390/electronics12173606.

[86] R. Alshawi et al., "SHARP-Net: A Refined Pyramid Network for Deficiency Segmentation in Culverts and Sewer Pipes," Aug. 02, 2024, arXiv: arXiv:2408.08879. doi: 10.48550/arXiv.2408.08879.

[87] R. Vandaele, S. L. Dance, and V. Ojha, "Deep learning for automated trash screen blockage detection using cameras: Actionable information for flood risk management," J. Hydroinformatics, vol. 26, no. 4, pp. 889–903, Apr. 2024, doi: 10.2166/hydro.2024.013.

[88] "CCTV image‐based classification of blocked trash screens - Smith - 2025 - Journal of Flood Risk Management - Wiley Online Library." Accessed: Aug. 24, 2025. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1111/jfr3.13038

[89] Muniandy, B., Maidin, S. S., Batumalay, M., & Dhandapani, L. (2025). Flood Prevention System Using IoT. International Journal of Advanced Computer Science and Applications (IJACSA), 16(3), 77–85.