IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151118

Smart Flood Detection and Impact Mapping System Using GSM-Based Alerts and Satellite Image Heat Maps

Mrs. Vyshnavi V J[1], Amar[2], Divit V[3], Hari Narayana S[4], Kiran S[5]

Assistant Professor, Department of Computer Science and Engineering, K. S. School of Engineering and Management, Bengaluru, India[1]

B.E. Student, Department of Computer Science and Engineering, K. S. School of Engineering and Management, Bengaluru, India[2-5]

Abstract: Floods rank among the most damaging natural hazards, and the harm to people and property grows sharply when warnings come late or never reach the affected community. Most IoT flood monitors described in recent literature depend on internet links, cloud back-ends and smartphone applications, which are exactly the services that become congested or fail during a disaster; moreover, they seldom assist with damage assessment after the water recedes. This paper presents a low-cost, two-stage framework that addresses both phases. The first stage is an Arduino-based field node that reads a water level sensor and a rain sensor, grades the situation as Normal, Warning or Danger against fixed thresholds, drives local LED and buzzer indicators, and, at the Danger grade, sends SMS messages through a SIM800L GSM modem without needing the internet. A battery or solar supply allows deployment at remote sites. The second stage is a Python pipeline that compares pre- and post-flood satellite images, isolates newly inundated regions, assigns severity classes and renders red-orange-green heat maps together with the affected area in km². A bench prototype using a water container demonstrates the sensing and indication logic. The framework supports early warning for residents and farmers and provides authorities with a spatial view for planning relief.

Keywords: Flood detection, Internet of Things, Arduino, GSM SMS alert, satellite imagery, heat map, disaster management.

I. INTRODUCTION

Flooding repeatedly damages homes, farmland, roads and utilities, and in severe events it claims lives. How much harm a flood finally causes is determined not just by its size but also by how early residents are warned and how well relief is coordinated once it has passed. Communities living along rivers and in low-lying rural areas are particularly exposed, because they often have weak mobile data coverage, unreliable electricity and little access to costly monitoring infrastructure.

A large share of the IoT-based flood monitoring systems proposed in recent years sends sensor data to a cloud service and notifies users through a smartphone application or web dashboard [1]–[5]. Such designs work well when networks are healthy, but heavy rain and flooding are precisely the conditions under which internet access degrades. In addition, existing work concentrates on detection and notification; it rarely connects the warning stage to the later task of mapping which areas were hit hardest, even though authorities need that picture to prioritise rescue and relief.

This paper describes a combined solution with two cooperating parts. A sensor node built around an Arduino microcontroller watches water level and rainfall continuously and uses GSM text messages, which require only cellular voice-grade coverage, to warn registered mobile numbers. After the event, a software module analyses satellite imagery of the region, detects the flooded extent, ranks it by severity and presents the outcome as a heat map. The main contributions are: (i) an internet-independent alerting node with three-level flood grading and local audio-visual warning; (ii) a satellite-image workflow that turns pre- and post-flood scenes into severity heat maps with affected-area figures; and (iii) a working bench prototype that integrates the sensing, grading and indication stages.

II. RELATED WORK

Several recent studies on IoT-assisted flood monitoring were examined to understand their strengths and gaps. Manoharan et al. [1] built a low-cost real-time monitor that issues early alerts, but they report limited sensor accuracy, dependence

 This work is licensed under a Creative Commons Attribution 4.0 International License

DOI: 10.17148/IJARCCE.2026.151118

on mains power, restricted coverage and no forecasting. Zulkapli et al. [2] combined sensing with mobile-app notifications and automatic water discharge for urban flash floods; their design requires internet access, suffers notification delay and relies on valves that may not discharge efficiently. Ankit et al. [3] monitor flood conditions and actuate water-flow control, which adds mechanical complexity and still needs connectivity. Kalai Selvi et al. [4] log water levels, send automatic alerts and analyse historical readings, yet inherit internet dependence and sensor reliability issues. Myint and Thein [5] use cloud-hosted IoT monitoring in urban areas but offer no SMS channel and experience delays.

On the remote-sensing side, radar-based and optical satellite methods for flood extent and impact mapping have been reported [6]–[8], along with machine-learning susceptibility mapping [9], GIS-based multi-criteria risk assessment [10] and LoRa-based detection [11]. These studies, however, treat ground sensing and satellite analysis separately. Table I condenses the limitations of the monitoring systems and the corresponding emphasis of the present work.

TABLE I LIMITATIONS OF REVIEWED MONITORING SYSTEMS

Ref.	Main limitation observed	Emphasis of this work
[1]	Sensor accuracy, power dependency, limited coverage, no prediction	Calibrated thresholds, battery/solar supply
[2]	Internet dependency, alert delay, valve inefficiency	GSM SMS alerts, simpler design
[3]	Internet dependency, complexity, no prediction	Simple, low-cost SMS-based node
[4]	Internet dependency, sensor issues, complexity	Reduced complexity, real-time SMS
[5]	Internet dependency, delay, no SMS support	Fast GSM-based notification

Across these works the recurring weaknesses are reliance on internet connectivity, system complexity and slow alerts, and none of them joins real-time warning with post-flood impact mapping. This gap motivates the framework proposed here.

III. PROBLEM DEFINITION AND OBJECTIVES

The need for this work was established through discussions with local residents and with farmers in flood-prone areas. Residents pointed to the absence of a dependable real-time monitoring facility, which raises the danger to life and property. Farmers highlighted the lack of any early warning for agricultural land, which directly threatens crops and income and leaves little time to move livestock and equipment. Both groups wanted prompt text-message alerts that do not depend on a data connection.

The problem can therefore be stated as follows: existing flood monitoring solutions are expensive, complicated and tied to internet connectivity, which delays alerts; a low-cost, real-time system able to warn people without internet access, and to support post-flood assessment, is required. The objectives are to (1) detect rising water in real time with sensors, (2) deliver alerts over GSM without internet, (3) visualise affected regions as heat maps, (4) help authorities in rescue planning and decision-making, and (5) compare flood impact across different areas. Flood forecasting and automatic control of gates or valves are deliberately left outside the present scope.

IV. SYSTEM ARCHITECTURE

The framework is divided into two subsystems, depicted in Fig. 1: a real-time detection and alert subsystem, and a post-flood impact detection and heat-mapping subsystem. Information flows from the former to residents and village officials through the mobile network, and from the latter to rescue teams, disaster managers and government authorities.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151118

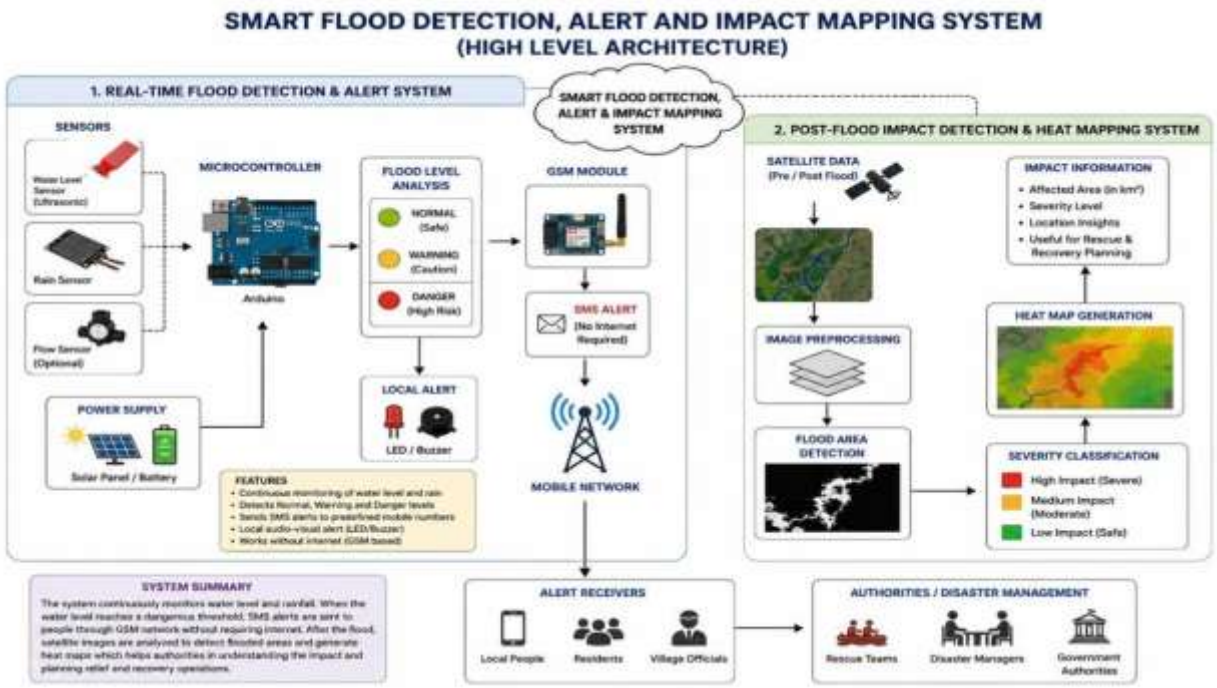


Fig. 1 High-level architecture of the proposed flood detection, alert and impact mapping system

A. Real-Time Detection and Alert Subsystem

Water level and rain sensors supply environmental data to an Arduino microcontroller, which compares the readings with predefined thresholds and classifies the condition into one of three levels, summarised in Table II. When the Danger level is reached, the GSM module sends an SMS to every registered number while a red LED and buzzer warn people near the node. Because messages travel over the cellular network rather than the internet, alerts can still be delivered when data services are unavailable. A rechargeable battery, optionally topped up by a solar panel, keeps the node running in remote locations.

TABLE II FLOOD LEVEL CLASSIFICATION AND SYSTEM RESPONSE

Level	Meaning	System response
Normal (green)	Safe condition	Continue monitoring
Warning (yellow)	Water level is rising	Caution; local indication
Danger (red)	Critical flood level detected	SMS to registered numbers; LED and buzzer on

B. Post-Flood Impact Mapping Subsystem

After an event, satellite scenes of the region from before and after the flood are collected and pre-processed to improve quality and discard irrelevant content. Water pixels are identified in both scenes, and the difference between them yields the newly flooded area. The flooded regions are then divided into high, medium and low impact classes (Table III) and drawn as a colour-coded heat map. The output also reports the affected area in km², the severity level and location insights useful for rescue and recovery planning, giving a geographical view that complements ground-level observation.

TABLE III SEVERITY CLASSES USED IN THE HEAT MAP

Class	Colour	Interpretation
High impact	Red	Severe; needs immediate rescue and relief
Medium impact	Orange	Moderate
Low impact	Green	Low impact or safe

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151118

V. REQUIREMENTS

The monitoring node uses inexpensive, widely available parts, and the analysis module relies on open-source Python libraries. Table IV lists the components and their roles.

TABLE IV HARDWARE AND SOFTWARE COMPONENTS

Component	Role
Arduino UNO / Nano	Central microcontroller; reads sensors and controls outputs
Water level sensor	Measures water height in the monitored area
Rain sensor	Detects rainfall
Flow sensor (optional)	Measures water flow rate
SIM800L GSM module	Sends SMS alerts without internet
LEDs and buzzer	Local audio-visual warning
Battery / solar panel	Power supply for remote operation
Arduino IDE	Firmware development and upload
Python, Matplotlib, Seaborn	Satellite image processing, plotting and heat map generation

VI. IMPLEMENTATION

A. Hardware Setup

An Arduino UNO forms the core of the prototype. The water level sensor is immersed in a plastic container that stands in for a flood-prone area, and a rain sensor and optional flow sensor can be attached to further input pins. Green, yellow and red LEDs and a buzzer are wired through a breadboard to digital output pins, so that each flood level has a distinct local indication. The SIM800L communicates with the Arduino over a serial (TX/RX) link and needs a suitably rated supply, since it draws high current while transmitting. In the field, a battery charged by a solar panel powers the unit.

B. Firmware Logic

The firmware, written in the Arduino IDE, runs a continuous loop: it initialises the sensors, indicators and GSM modem; samples the water level and rainfall; compares the values with the thresholds; sets the status; and acts on it. A Warning triggers a local caution indication, whereas a Danger switches on the red LED and buzzer and sends an SMS to each registered number using AT commands. A status flag prevents the same event from generating repeated messages before the loop resumes. In the prototype, the analogue water-level reading from pin A0 (0–1023) is mapped to levels using thresholds of 600, 630 and 650, which were chosen from earlier readings of the sensor (Table V). The loop repeats every 500 ms and prints each reading and status on the serial monitor.

TABLE V PROTOTYPE THRESHOLD MAPPING (ANALOGUE READING)

Reading	Status	Indication
Below 600	Tank empty	All outputs off
600 to 629	Low water level	Green LED
630 to 649	Medium water level	Yellow LED
650 and above	High water level – alert	Red LED and buzzer

C. Post-Flood Image Analysis

The analysis module is written in Python and follows six steps: (1) load the pre- and post-flood satellite images; (2) pre-process them through noise removal, resizing and alignment, and enhancement; (3) detect water pixels and compare the pre-flood extent with the post-flood extent to obtain a binary flood mask in which newly flooded pixels are white; (4) group flooded regions into high, medium and low impact classes; (5) render the classified result with Matplotlib and Seaborn as a red, orange and green heat map; and (6) report the affected area in km² (the flooded pixel count scaled by the ground area represented by one pixel), the severity level and location insights.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151118

D. Integration

The two subsystems complement each other in time. The sensor node issues the early warning while the flood is developing, and the analysis module is applied after the event to assess damage. Their outputs reach the community and the authorities, respectively, as indicated in Fig. 1.

VII. RESULTS AND DISCUSSION

Fig. 2 shows the assembled prototype, with the Arduino UNO, breadboard carrying the LEDs and buzzer, and the water level sensor placed in the container. Raising the water level in the container moves the sensor reading across the thresholds, and the green, yellow and red LEDs light in turn to show the Normal, Warning and Danger conditions; at the highest level the buzzer also sounds. This confirms that the sensing, threshold comparison and local indication stages operate together as intended.

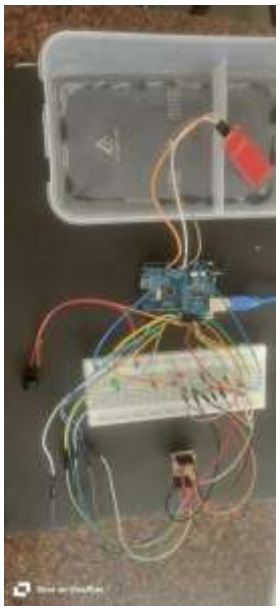


Fig. 2 Hardware prototype: Arduino UNO, breadboard with LEDs and buzzer, and water level sensor in the container

The impact map generated by the analysis module is illustrated in Fig. 3. High-impact locations appear as red zones that call for immediate rescue and relief, moderate zones are orange, and low-impact or safe locations are green. Presenting the information in this form allows authorities to rank areas by urgency and to allocate rescue teams and supplies accordingly, instead of depending on scattered ground reports.

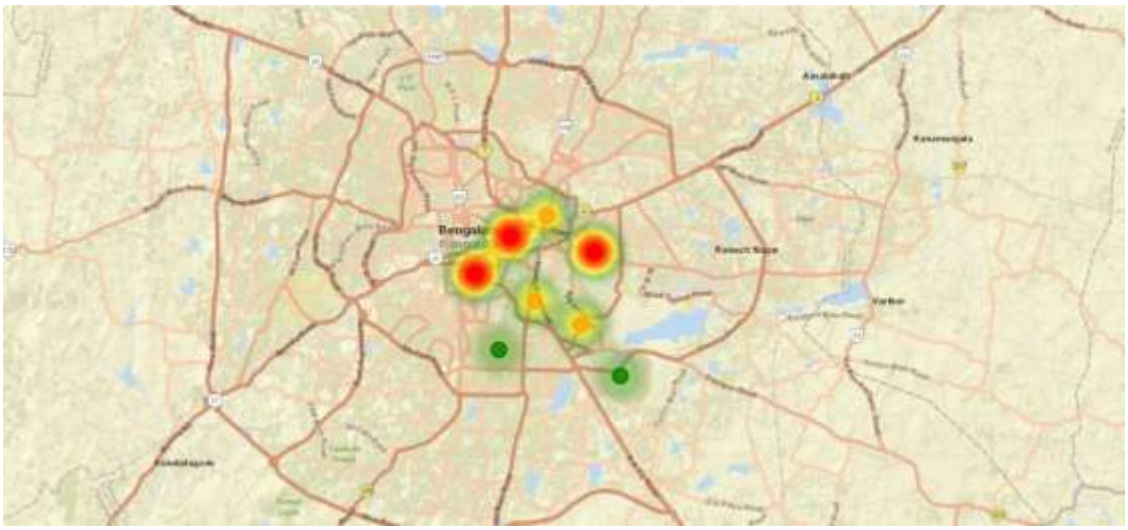


Fig. 3 Flood impact severity heat map (red: high, orange: moderate, green: low impact)

 This work is licensed under a Creative Commons Attribution 4.0 International License

DOI: 10.17148/IJARCCE.2026.151118

Some limitations should be noted. The present prototype is a single node with fixed thresholds that need recalibration for each installation site, and its water-level sensing was demonstrated on a bench set-up rather than on a real river. SMS delivery and the satellite pipeline have not yet been evaluated quantitatively, and SMS delivery depends on local cellular coverage, and optical satellite imagery can be obscured by cloud, which motivates the use of radar imagery in future work.

VIII. APPLICATIONS

The framework can serve riverside villages and residential areas that need early warning, and farmland where advance notice allows crops, livestock and equipment to be secured. Village officials, rescue teams and disaster managers can use the alerts during evacuation and the heat maps for damage assessment, relief prioritisation and long-term planning, including the identification of zones that are affected repeatedly. The node may also be placed at reservoirs, canals, drains and low-lying urban areas, and its GSM and solar/battery operation suits regions with poor internet coverage and unreliable power.

IX. CONCLUSION AND FUTURE WORK

This paper presented a low-cost framework that tackles two shortcomings of current flood monitoring: dependence on internet connectivity and the lack of post-flood impact analysis. By combining Arduino-based sensing, GSM SMS alerting and satellite-based heat mapping, it offers a practical aid for both early warning and disaster management, and a prototype confirmed the sensing and graded indication behaviour.

Future work includes flood prediction from historical sensor data using machine learning; multiple sensor nodes linked by LoRa for wider coverage [11]; better sensor calibration with fault detection; a web or mobile dashboard for authorities alongside SMS; use of SAR imagery to map floods through cloud cover [6], [8]; GPS-tagged alerts with automatic notification of rescue agencies; and integration with automatic water-control mechanisms.

ACKNOWLEDGMENT

The authors thank the management, the Principal, the Head of the Department of Computer Science and Engineering and the project coordinators of K. S. School of Engineering and Management, Bengaluru, for the facilities and support provided during this work.

REFERENCES

[1]. L. Manoharan, K. S. Vignesh, M. Saravanapandian, B. Ajay, R. K. Dwibedi and K. U. Kalyani, “IoT integrated flood detection and avoidance,” 2025.
[2]. Z. bin Zulkapli, M. Zolkapli, Z. Janin, N. Burham, M. Masrie and S. Shahbudin, “IoT-based flash flood monitoring with mobile notification alerts for urban areas,” 2025.
[3]. T. Ankit, K. Kumar, G. Pal, A. Kumar, M. Pal, A. Singh and M. Tyagi, “IoT based flood monitoring and control system,” 2025.
[4]. D. M. Kalai Selvi, V. Sharmila, P. Ezhumalai, M. Pranavi, P. J. Penubolu and Sadhvika, “Internet of Things based automated water level monitoring and alert system,” 2025.
[5]. M. M. Myint and T. L. L. Thein, “Real-time flood monitoring system using Internet of Things in urban areas,” 2025.
[6]. “Flood mapping using satellite radar,” *Nature Communications*, 2025.
[7]. “AI-based flood impact mapping using remote sensing,” *ISPRS*, 2025.
[8]. “Satellite-based flood detection using SAR data,” *Remote Sensing* (MDPI), 2025.
[9]. “Flood susceptibility mapping using machine learning,” *Scientific Reports*, 2025.
[10]. “Flood risk mapping using GIS and multi-criteria analysis,” *Frontiers in Water*, 2025.
[11]. “LoRa-based flood detection system,” 2025.

 This work is licensed under a Creative Commons Attribution 4.0 International License