

Adaptive AIoT-Based Flood Early Warning and Community Risk Mapping System

Gaurav Ramesh Rao Savant¹, Harshal Narayan Rao Ranotkar², Milind V. Lande³

^{1,2} Lecturer in Computer Engg, Department of Computer Engineering, Government Polytechnic, Arvi, Maharashtra India, 442201.

³ Lecturer in Computer Engg, Department of Computer Engineering, Government Polytechnic, Gadchiroli, Maharashtra India, 442605.

Abstract - Flood monitoring systems commonly depend on measured water level and fixed threshold rules. The uploaded study describes an Arduino-based design using water-level sensing, a servo-operated gate, GSM alerts, LCD display, LEDs and a buzzer. This proposal reframes that foundation into a new research project: an Adaptive AIoT Flood Early Warning and Community Risk Mapping System. The proposed system combines water level, rainfall, flow tendency and soil-moisture observations with an edge-based machine-learning model. Instead of treating every threshold crossing as the same event, the system estimates a short-term flood-risk class and updates the warning level according to recent trends. A low-power long-range communication layer is proposed for distributed sensor nodes, while a gateway provides local storage and optional cloud synchronization [1]. A community-facing dashboard can display station status, risk level, recent trends and the affected zone. The research will evaluate prediction quality, warning lead time, communication reliability, energy use and false-alarm rate. The project is deliberately designed as a research prototype rather than a claim of a fully validated operational flood-control system.

Keywords- AIoT, flood early warning, edge computing, machine learning, LoRaWAN, risk classification.

I. INTRODUCTION

The uploaded paper establishes a practical IoT flood-management architecture in which sensor readings are processed by an Arduino Uno and predefined water-level thresholds trigger local and GSM-based warnings and automatic gate movement. The source explicitly identifies

sensor accuracy, communication/power dependence and large-scale deployment cost as important limitations. These points provide a useful starting point for a new research direction rather than a text-rewriting exercise.

Recent literature indicates that flood prediction is increasingly combining hydrological and meteorological variables with machine-learning and deep-learning methods, while data availability and generalization remain important challenges. A 2020 systematic review covering studies from 2015–2019 highlights rainfall and water level among important predictors and identifies limited historical datasets as a continuing problem. Other recent work is exploring AIoT architectures, edge processing, heterogeneous sensing and long-range communication. Therefore, the proposed research focuses on a practical gap: how to build a small, resilient, locally deployable prototype that converts heterogeneous sensor observations into an interpretable, adaptive risk signal without depending entirely on a remote cloud service.

II. PROPOSED RESEARCH IDEA

The central idea is to replace a simple fixed-threshold warning chain with an adaptive risk-estimation pipeline. Multiple sensor nodes periodically collect water level, rainfall, flow-related information and soil moisture. An edge gateway performs data cleaning, missing-value handling, feature extraction and short-term risk inference. The output is a four-level status such as Normal, Watch, Warning and Critical. The model can use a lightweight algorithm such as Random Forest or gradient-boosted trees

for the first prototype, followed by comparison with an LSTM model if sufficient time-series data are available [10-16].

A key research feature is sensor-fusion validation. The system will compare the prediction produced from water level alone against predictions obtained from multiple variables. The study will also test whether trend-based features—such as rate of water-level rise and accumulated rainfall—reduce false alarms. This creates measurable research questions rather than merely adding more hardware.

III-RESEARCH GAP AND PROPOSED CONTRIBUTION

Table 1- Proposed Contribution observation

Existing/Source Approach	Observed Issue	Proposed Research Extension
Fixed water-level thresholds	Same rule may not reflect changing rainfall or rise rate	Adaptive multi-variable risk classification
Single primary water-level input	Limited representation of flood dynamics	Sensor fusion using rainfall, level, trend and soil moisture
Central Arduino decision logic	Limited computational scope	Edge gateway for preprocessing and ML inference
GSM-centric notification	Coverage and delivery can vary	Long-range low-power sensor networking plus local fallback alerts
Prototype alert/control demonstration	Limited quantitative validation	Evaluation using accuracy, recall, false alarms, lead time, latency and energy
Site-specific deployment	Generalization is uncertain	Configuration layer for different sites and threshold/calibration profiles

IV- RESEARCH OBJECTIVES

- Design a low-power multi-sensor flood-monitoring node suitable for river, drainage-channel or reservoir-edge experiments.
- Develop an edge-based data-processing pipeline for filtering, synchronization, feature extraction and anomaly detection.
- Develop and compare water-level-only and multi-sensor machine-learning risk classifiers.
- Generate localized warning states using both current conditions and short-term trends.
- Provide a resilient alert mechanism with local audio/visual warning and a remote notification channel.
- Evaluate the prototype using measurable technical and operational performance indicators.

- Document limitations and identify the conditions under which the model should defer to a conservative rule-based warning.

V- RESEARCH QUESTIONS

- Does combining water level, rainfall and temporal trend features improve flood-risk classification compared with water level alone?

How early can the proposed system identify a transition from normal conditions to a warning state?

How does edge inference affect alert latency when compared with cloud-dependent processing?

How does sensor noise or temporary communication loss affect the reliability of the warning decision?

Can a lightweight model provide useful performance on an embedded/edge platform with limited computational resources?

VI- SYSTEM ARCHITECTURE

The proposed architecture has five logical layers: (1) sensing, (2) communication, (3) edge analytics, (4) application/visualization, and (5) alerting. Sensor nodes collect observations at a configurable interval. A low-power long-range link transports compact measurements to a gateway. The gateway validates packets, stores recent observations, calculates temporal features and executes the trained model. A dashboard presents the latest measurements, model output and sensor-health information. Local alarms remain available so that the system can continue to warn users when backhaul connectivity is unavailable.

- Sensing layer: waterproof/non-contact water-level sensor, rain gauge, soil-moisture sensor and optional flow/velocity sensor.
- Node controller: ESP32-class or equivalent low-power microcontroller[13].
- Communication layer: LoRa/LoRaWAN-class link for low-rate long-range telemetry; GSM/Internet can be retained as an optional backhaul.
- Edge layer: Raspberry Pi-class gateway or equivalent edge computer for preprocessing and model inference.

- Data layer: local time-series storage with optional cloud synchronization[11].
- Application layer: web dashboard showing water level, rainfall, trend, risk state, sensor health and event history.
- Alert layer: buzzer/LED at the local station plus configurable remote notification.

The system comprises four principal subsystems: sensing, control, actuation, and communication/notification. Water-level transducers provide continuous analog or digital readings of river, reservoir, or basin levels. An Arduino Uno serves as the central processing unit, acquiring sensor data, evaluating it against configurable thresholds, and coordinating downstream actions. Actuation is implemented via a servo motor that modulates a floodgate to regulate hydraulic discharge[3]. Local alerting employs an audible buzzer, an LCD for real-time numeric readouts, and a set of LEDs that signify discrete risk categories. Remote notifications are transmitted through a GSM module capable of dispatching automated SMS messages to designated stakeholders.

a) Block Diagram:

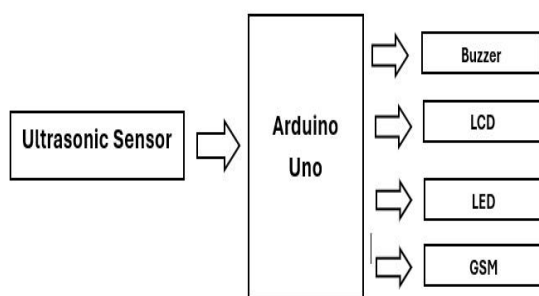


Fig.1: Block Diagram

The system continuously monitors water levels in real time. Upon detecting high water levels, it autonomously triggers alarms, visual signals, and SMS notifications, ensuring rapid and reliable flood warnings. This integration of IoT and automation enhances flood preparedness and minimizes potential damage and casualties [4-6].

b) Circuit Diagram:

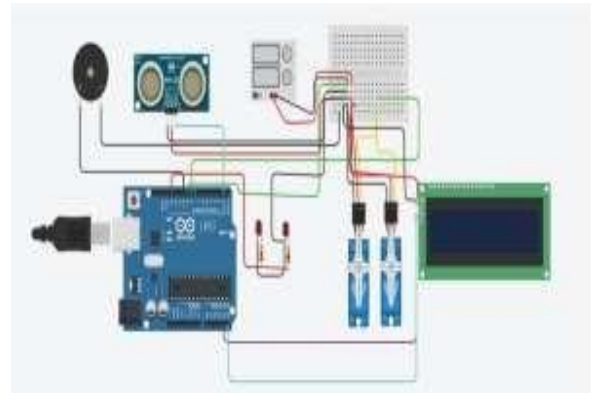


Fig.2: Circuit Diagram

Circuit diagram and each component connect each other. The ultrasonic sensor, GSM, LED, LCD, buzzer is connected to Arduino Uno. In the dam or river, the ultrasonic sensor sense the water it and display the water level on the LCD display. The servo motor opens and closes the gate automatically and GSM module sending SMS notification to a predefined number.

c) Software:

The Arduino IDE is a free, open-source software that allows users to write and upload code to Arduino boards. It is compatible with Windows, macOS, and Linux operating systems

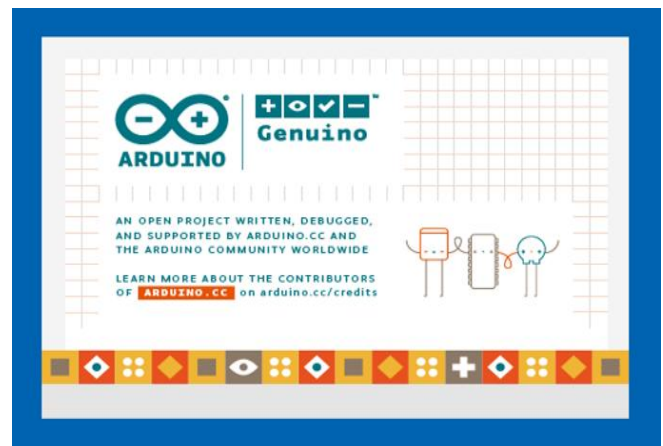


Fig.3 Arduino Software

d) Hardware Connection:

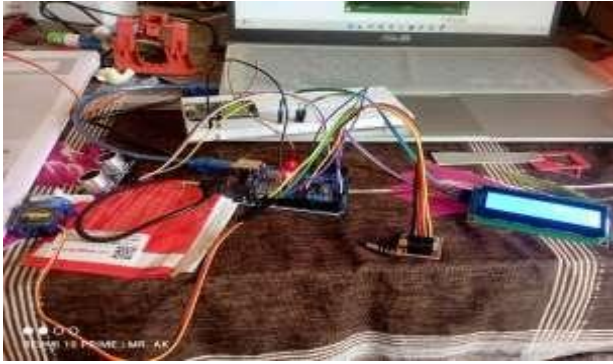


Fig.4: Hardware Connection

e) Actual Model:

Fig.5: Actual Model

f) Live Message:

Fig.6: Live Message

VII- DATA AND FEATURE DESIGN

The research dataset should contain timestamped observations collected during normal and elevated-water conditions. Where field flood events are insufficient, historical records may be used for model development, while the physical prototype is used for sensing, communication and end-to-end validation. The study must clearly separate training, validation and test periods to reduce leakage from adjacent time samples.

- Direct variables: water level, rainfall intensity/accumulation, soil moisture and optional flow measurement.
- Trend variables: water-level change over 5, 10, 15 and 30-minute windows; rainfall accumulation over selected windows.
- Context variables: sensor ID/location, time of observation and sensor-health indicators.
- Quality variables: missing-data flag, signal quality and outlier status.
- Target variable: a documented risk class or forecast horizon, defined before model evaluation.

VIII - MACHINE-LEARNING METHODOLOGY

The first model should be intentionally lightweight. Random Forest can provide a strong baseline for mixed features and is comparatively interpretable. A second model, such as an LSTM, can be evaluated when a sufficiently large and continuous time-series dataset is available. The research should not claim that one algorithm is superior before testing it on the project's own dataset.

- Step 1: synchronize and clean sensor streams.
- Step 2: detect impossible readings, sensor dropouts and obvious outliers.
- Step 3: construct rolling-window and rate-of-change features.
- Step 4: label observations using a predefined, reproducible flood-risk rule or validated event record.
- Step 5: train baseline and candidate models using time-aware data splitting.
- Step 6: evaluate precision, recall, F1-score, confusion matrix and false-alarm rate.
- Step 7: measure warning lead time and inference latency.

- Step 8: deploy the selected lightweight model on the edge gateway and compare laboratory/offline results with live operation.

IX-EXPERIMENTAL PLAN

Table 2- Proposed Phase wise Working plan

Phase	Activity	Output	Evaluation
Phase 1	Requirements, site profile and sensor calibration	System specification	Calibration error
Phase 2	Build multi-sensor node and gateway	Working prototype	Packet delivery, power use
Phase 3	Collect/prepare time-series data	Curated dataset	Completeness, quality
Phase 4	Train baseline and adaptive ML models	Candidate models	F1, recall, false alarms
Phase 5	Edge deployment and live tests	Integrated system	Latency, lead time
Phase 6	Stress testing	Reliability report	Packet loss, sensor faults

X- EXPECTED OUTCOMES

- A working prototype for distributed flood-condition sensing.
- A reproducible multi-sensor dataset and documented preprocessing pipeline.
- A comparison between fixed-threshold and adaptive ML-based warning logic.
- An edge-deployed risk classifier with measured inference latency.
- A dashboard and local alert mechanism for demonstration and testing.
- A quantitative report covering accuracy, false alarms, lead time, communication reliability and energy behavior.
- A documented pathway for future field validation with local disaster-management stakeholders.

XI-IMPORTANT SCOPE AND SAFETY BOUNDARIES

The prototype should be treated as a research and demonstration system unless it has undergone appropriate engineering validation and authorization. Automatic physical control of real dams, floodgates or other critical hydraulic infrastructure should not be inferred from a laboratory prototype. For the proposed research, the actuation interface can be represented by a low-risk laboratory model or simulation. The warning system should also include a conservative fail-safe state and a clearly documented manual override.

XII- NOVELTY STATEMENT FOR THE PROPOSED PROJECT

The proposed novelty is not the use of IoT sensors or SMS alerts by themselves; those elements are already common in the source paper and recent literature. The research contribution is the experimentally testable integration of adaptive multi-sensor risk inference, edge-side processing, sensor-health awareness and resilient local/long-range communication, followed by a quantitative comparison against a fixed-threshold baseline. The project therefore turns the original monitoring prototype into a research question about whether adaptive sensor fusion can improve warning reliability and lead time under constrained connectivity.

CONCLUSION

The Adaptive AIoT-Based Flood Early Warning and Community Risk Mapping System provides a smart and proactive solution for flood management. By combining IoT sensors, AI-based prediction, real-time monitoring, and risk mapping, the system can provide timely warnings and identify vulnerable areas. It can help authorities improve emergency response, reduce damage, protect communities, and build a more resilient flood-management framework.

REFERENCES

- [1] Azid, S. I., Sharma, B. N., Raghuwaiya, K., Chand, A., Prasad, S., & Jacquier, A. (2015). SMS Based Flood Monitoring and Early Warning System. *Proceedings of the International Conference on Communication, Computing and Virtualization*.
- [2] Marsono, H. R., & Nugroho, A. S. (2016). Internet of Things-Based Flood Early Warning System in Java Island. *2016 4th International Conference on Cyber and IT Service Management (CITSM)*, 71–75.
- [3] Alfieri, L., Pappenberger, F., Wetterhall, F., Haiden, T., Cloke, H. L., & Thielen, J. (2016). Operational early warning systems for water-related hazards. *Environmental Science & Policy*, 58, 117–122.
- [4] Qiu, Z., Zhang, C., Cao, H., Wang, Y., & Xia, Y. (2017). Research on Flood Monitoring and Alarm System Based on Internet of Things. *Procedia Engineering*, 174, 1538–1545.
- [5] Hasby, S. B. M., & Abd El-Aal, A. M. (2017). An IoT-Based Flood Monitoring and Warning System Using Wireless Sensor Networks. *International Journal of Distributed Sensor Networks*, 13(4).
- [6] Priya, S. Jana, et al. (2017). Flood Monitoring and Alerting System. *International Journal of Computer Engineering & Technology*, 8(2), 15.
- [7] A Flood Warning System to Critical Region. (2017). *Procedia Computer Science*, 109, 1104–1109. DOI: 10.1016/j.procs.2017.05.453.

- [8] Sood, S. K., Sandhu, R., Singla, K., & Chang, V. (2018). IoT, big data and HPC based smart flood management framework. *Sustainable Computing: Informatics and Systems*, 20, 102–117. DOI: 10.1016/j.suscom.2017.12.001.
- [9] Supani, A., Andriani, Y., Taqwa, A., & Ciksadan. (2018). Study on implementation of flood early warning system with Internet of Things in peri-urban settlement of Palembang for sustainability. *IOP Conference Series: Earth and Environmental Science*, 202, 012001. DOI: 10.1088/1755-1315/202/1/012001.
- [10] Participatory early warning and monitoring systems: A Nordic framework for web-based flood risk management. (2018). *International Journal of Disaster Risk Reduction*, 31, 1295–1306. DOI: 10.1016/j.ijdr.2018.01.038.
- [11] Moreno, C., et al. (2019). Rivercore: IoT device for river water level monitoring over cellular communications. *Sensors*, 19(1).
- [12] Zahir, S. B., et al. (2019). Smart IoT Flood Monitoring System. *Journal of Physics: Conference Series*, 1339(1).
- [13] Alaydrus, S. M. A. E., Fathurrahman, F. H., & Suksmono, A. B. (2019). An IoT-Based Early Warning System for Flood Detection. 2019 *International Conference on Informatics and Computational Sciences (ICICoS)*, 92–96. DOI: 10.1109/ICICOS.2019.8857127.
- [14] Chen, X., Guo, P., & Ma, X. (2019). The Application of IoT in Flood Monitoring and Warning System. 2019 4th *International Conference on Control and Robotics Engineering (ICCRE)*, 273–277. DOI: 10.1109/ICCRE.2019.8863913.
- [15] Computer Vision and IoT-Based Sensors in Flood Monitoring and Mapping: A Systematic Review. (2019). *Sensors*, 19(22), 5012. DOI: 10.3390/s19225012. This is particularly relevant to your topic because it reviews IoT sensors, computer vision, flood monitoring, mapping, modelling, and early-warning applications.
- [16] IoT Based Flood Detection and Notification System using Decision Tree Algorithm. (2019). 2019 *International Conference on Intelligent Computing and Control Systems (ICCS)*. DOI: 10.1109/ICCS45141.2019.9065799. This is useful for the AI/ML component of an adaptive flood-warning system.