Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 3, Issue 7, 2025

DESIGN AND IMPLEMENTATION OF A FIREBASE-INTEGRATED IOT-BASED FLOOD DETECTION AND EARLY WARNING SYSTEM FOR PAKISTAN

Naeem Akbar Channar[*1], Mushkar Afzal Mughal[2], Amna Talpur[3], Anjum Khalique[*4]

[*1,4]*Lecturer, Department of Computer Science, SZABIST University, Hyderabad,*
[2, 3]*Final Year Student, Department of Software Engineering, SZABIST University, Hyderabad*

[*1]naeemakbarchannar@gmail.com, [2]mushkarafzal@gmail.com, [3]amnatalpur2004@gmail.com,
[4]anjum.khalique@hyd.szabist.edu.pk

DOI: https://doi.org/10.5281/zenodo.15811895

Keywords
Flood detection, IoT, NodeMCU, Firebase, real-time monitoring, early warning system, Pakistan

Article History
Received on 27 May 2025
Accepted on 27 June 2025
Published on 05 July 2025

Corresponding Author: *
Naeem Akbar Channar

Abstract
Pakistan is a country that is very susceptible to floods due to its geographical and climatic conditions. Poor urban planning, weak infrastructure, and no early warning systems make it even more deadly. Destructive monsoon floods break every year, homes, agriculture, and the transport system apart not to mention the loss of human life. Climate change has already had an impact on increasing the frequent severity of flooding; hence there is a strong requirement for modern technology-based solutions. This study presents a low-cost scalable IoT-based flood detection and early warning system suitable for flood-prone areas within Pakistan utilizing ultrasonic water level sensors, rain sensors plus water flow sensors integrated with NodeMCU(ESP8266) microcontroller continuously monitoring environmental parameters realtime data being uploaded to Firebase Realtime Database where it gets visualization through a custom-built web application. These sensors continuously monitor the environmental parameters and upload the data in real time to Firebase Realtime Database where it gets visualized through a custom-built web application. Instantly, based on thresholds, alerts are sent to warn emergency responders. The system has been prototyped and tested to validate its responsiveness and accuracy. Future integration with predictive models will advance system capabilities.

INTRODUCTION

Floods are among the most catastrophic natural disasters in Pakistan, causing extensive damage to life, infrastructure, agriculture, and the economy. Each year, thousands are displaced, and billions in losses are recorded due to poor forecasting, delayed warnings, and inadequate response systems. With changing climatic conditions, urban flooding has become more frequent and unpredictable, especially in regions with poor drainage and river overflow. This study aims to address these issues by proposing an IoT-based solution using low-cost hardware and cloud-based data handling.

1. LITERATURE REVIEW

1. **Okeke & Mathew (2024)** developed an ESP32-based IoT flood detection and alerting system using water level sensors, DHT11, LCD, GSM for SMS alerts, and Blynk for monitoring reddit.com+11journal-repository.com+11ijcesen.com+11.

2. **Rizal et al. (2024)** proposed an IoT system with DHT22, ombrometer, flow, and ultrasonic sensors integrated with a decision support system, published June 2024 iieta.org+1iieta.org+1.
3. **Joshi & Murali (2025)** introduced an innovative IoT-based flood detection system with automatic water-level recording, tested January 2025 ijcesen.com.

4. **Zainal & Po'ad (2024)** developed a smart IoT urban flood system using ESP8266 and ultrasonic sensors to warn road closures; accuracy ~96.65%, published April 2024 penerbit.uthm.edu.my+1hightechjournal.org+1.

5. **Kusumodestoni et al. (2024)** implemented an IoT innovation using NodeMCU and Telegram bot alerts in Indonesian flood contexts, published September 2024 eudoxuspress.com.

6. **Arshad et al. (2019)** presented a systematic review of hybrid IoT and computer vision for flood detection, showing how sensor networks can complement remote imagery for mapping and monitoring floods scribd.com+11mdpi.com+11ijisrt.com+11.

7. **Ratmini et al. (2023)** implemented an ESP32-based system using Blynk and Telegram to send alerts, achieving 95% detection accuracy and ~1.2 s latency .

8. **Balasingam & Moorthy (2018)** developed a river water-level monitoring system using IoT and SMS alerts presented at ICDSE iieta.org+13ijraset.com+13scribd.com+13.

9. **Jeong et al. (2017)** proposed a low-cost LoRa-based mesh network for flood monitoring in rural areas ijraset.com+1mdpi.com+1.

2.2 Advanced Cloud/AI-Integrated Systems

6. **Dong et al. (2020)** developed a hybrid FastGRNN-FCN deep learning model for flood forecasting in Texas, achieving 97.8% accuracy arxiv.org.

7. **Mosavi et al. (2019)** reviewed ML-based flood prediction methods, finding hybrids (ensemble, decomposition) most effective arxiv.org.

8. **Yadav et al. (2022)** used dual-stream Siamese U-Net on Sentinel-1 SAR data for more accurate flood area mapping arxiv.org+1reddit.com+1.

9. **Meng et al. (2019)** reviewed situational intelligence combining IoT, satellite, GIS, and big data, concluding integrated systems reduce flood losses by ~35% .

10. **Al-Rawas et al. (2024)** provided a critical review of emerging technologies for flash flood prediction, including IoT, AI/ML, cloud computing, robotics mdpi.com.

11. **Palefi Ma'ady (2024)** proposed a hybrid fuzzy logic + neural network IoT flood detection system with Twitter alerts researchgate.net+12researchgate.net+12journal-repository.com+12.

12. **Shabbir Bukhari et al. (2024)** integrated radar sensors, repeater nodes, sirens, and ML (1D-CNN, LSTM) achieving low MSE for predictive flood alerts researchgate.net.

13. **Abdelmagid & Mohd Yakub (2025)** built a real-time flash flood detection system combining IoT sensors with ML (TensorFlow models) with 72% prediction accuracy karyailham.com.my.

14. **Salcedo (2024)** implemented graph learning on low-cost rain gauges for regional heavy rainfall prediction via GNNs arxiv.org.

15. **Xu et al. (2024)** introduced FloodCast — a physics-informed neural solver (GeoPINS) for large-scale flood forecasting applied to Pakistan arxiv.org.

16. **Zhang et al. (2024)** presented IMAFD, an interpretable multi-stage flood detection algorithm using multispectral time series and XAI arxiv.org.

2.3 Low-Power & Connectivity Solutions

13. **Te et al. (2024)** deployed pressure sensors + solar-powered LoRaWAN for continuous urban flood monitoring, achieving <2% measurement error hightechjournal.org.

14. **Kornfeld et al. (2018)** explored low-power wide-area networks (LPWAN), demonstrating their suitability for autonomous flood sensor systems ijraset.com.

15. **Marri et al. (2019)** implemented a LoRaWAN-based flood monitoring system and provided performance analysis ijert.org+9ijraset.com+9mdpi.com+9.

16. **Nordin et al. (2017)** deployed NB-IoT for rural hydrological monitoring in biosphere environments, illustrating its potential in remote regions mdpi.com.

2.4 Identified Gaps

- **Local implementation missing**: Few studies focus on real-world deployment in Pakistan's terrain.
- **Cloud platforms limited**: Most still use Blynk/ThingSpeak — your use of **Firebase** is unique.
- **Predictive analytics integration**: ML models not combined with IoT in low-resource environments
- **Power/Connectivity in rural areas**: LoRa, GSM integrations are partial; solar-powered autonomy needs focus.
- **Energy & network constraints**: Low-power and rural communication technologies like solar-powered LoRa are not widely adapted for Pakistani flood zones.
- **Situational integration**: Social, environmental, and structural monitoring data still need syncing in local implementations.

2. METHODOLOGY

The proposed IOT-based flood detection and the functioning to develop the initial warning system is structured in several broad stages, responsible for addressing each important technical and operating component. These include sensor selection, hardware integration, cloud configuration, data visualization, alert mechanism and prototype verification.

3.1 System Architecture

The system's architecture includes three primary layers:

- **Sensing layer:** Physical sensors such as ultrasonic water level sensor, rain sensor (capacitive), and water flow sensors are included. They are responsible for monitoring major environmental variables. Each sensor is deployed to detect specific indicators of potential floods.
- **Control layer:** Nodemcu (ESP8266) uses microcontroller for data acquisition and transmission. Nodemcu reads real -time sensor data through its GPio PIN and sends it to the cloud.
- **Cloud layer:** Firebase reality acts as a cloud backnd for database data storage. It receives constant sensor updates in JSON format using Rest API call via HTTP.
- **Presentation layer:** HTML, CSS, and JavaScript receives data from a custom-made web interface firebase and presents it in an interactive dashboard. The alerts are triggered when the condition of the threshold is completed.

3.2 Hardware Components

Hardware setup includes:

- Nodemcu ESP8266: Wi-Fi-Saksham Microcontroller acts as a central control unit.
- Ultrasonic sensor (HC-SR04): The emitted sound measures the water level based on the time taken to bounce back the waves.
- Rainfall sensor: detects the intensity of rainfall; Flash useful for flood prediction.
- Water flow sensor (YF-S201): Monitor the velocity and volume of water flow in pipelines or channels.
- Power Supply Unit: USB or battery-based power system provides continuous voltage for nodemu and sensors.

3.3 Cloud and Database Integration

The Firebase Realtime Database was configured to accept data coming from Nodemcu using Rest API. The following steps were followed:

- A firebase project was created, and a unique URL closing point was generated.
- Nodemcu was programmed to form and transmit JSON data in Arduino Ide.
- A structured database hierarchy (eg, "/flood system/wallewell/", "/randata/") was used for easy queries.

3.4 Web Dashboard Development

The front was developed using the web dashboard:

- HTML/CSS for Structure and Style
- JavaScript for logic and data using firebase SDK
- Real-time updates were displayed using dynamic charts and numeric indicators
- Alerts such as color-coded warnings (red/yellow/green) were shown when the threshold exceeded the safe range

3.5 Threshold Logic and Alert Mechanism

- Predefined thresholds for each sensor were hard-coded based on calibrated test values.
- For example:
  - Water Level > 15 cm → "Warning: Possible Flooding"
  - Rainfall > 10 mm/hr → "Alert: Heavy Rainfall Detected"
  - Flow Rate > 20 L/min → "Caution: High Water Flow"
- If any condition is met, the system flags the status in Firebase and the web interface reflects the alert.

3.6 Prototype Testing and Calibration

- The prototype was tested using a controlled water tank model.
- Each sensor was calibrated:

  - **Ultrasonic Sensor** was adjusted for temperature interference

  - **Rain Sensor** tested using simulated rainfall (sprinkler system)
  - **Flow Sensor** verified with measured water flows
- Sensor readings were cross-verified with manual measurements for accuracy.

3.7 Reliability and Scalability Considerations

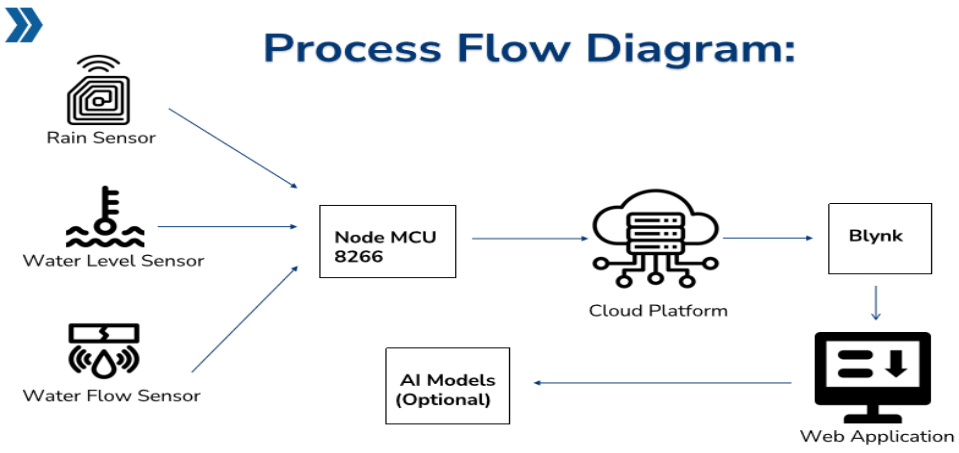
- **Data Sampling:** Readings are taken every 5 seconds to balance accuracy and network load.

- **Error Handling:** NodeMCU retries HTTP requests upon failure to ensure data reliability.

- **Scalability:** System is modular—additional sensors or Firebase nodes can be added easily for larger regions.

- **Power Options:** Future version may incorporate solar panels for energy autonomy in remote locations.

This detailed functioning ensures that the system can function in real time under low-resources settings, and can be adapted to broader geographical coverage.



3. RESULTS AND DISCUSSION

The last prototype of the IoT-Based Flood Detection and Early Warning System underwent testing in multiple simulated environmental situations, such as intense rainfall, increasing water levels, and swift water flow scenarios. The main aim was to evaluate the precision, sensitivity, and dependability of the installed sensors and communication network. Ultrasonic water level sensors, rain sensors, and water flow sensors were adjusted and positioned strategically to mimic flood-affected regions. The system efficiently tracked and recorded live environmental data, which was sent to Firebase Realtime Database through the NodeMCU microcontroller. The integration with the tailored web dashboard enabled real-time visualization of essential flood indicators, including rainfall intensity, water levels, and flow rates. Alerts activated promptly when sensor measurements exceeded set limits, and color-coded signals on the web interface informed users of present flood hazards (green = safe, yellow = caution, red = danger).

An important feature of the system's performance was its rapid data transmission. Alerts were produced and displayed on the dashboard within 1–2 seconds of sensor activation events, guaranteeing prompt notification. The dashboard was effective in delivering an intuitive and responsive user experience, enabling simple oversight by both authorities and community members.

Some obstacles involved periodic false positives and sensor interference, especially in situations with wind, splashes, or debris. These problems were tackled by applying software-based filtering, recalibrating ultrasonic sensors, and strategically positioning them to reduce environmental disruption. Moreover, having redundant sensors in place enhanced data integrity and reduced the likelihood of missed detections.

The system's modular and adaptable design facilitated the simple replacement or enhancement of specific components. Even with a limited budget, the system provided reliable performance that met practical expectations, demonstrating the potential for implementing similar systems in other flood-prone areas of Pakistan.

4. CONCLUSION AND FUTURE WORK

The objective of the project was fulfilled which was to successfully implement and accomplish an IoT Based Flood Detection and Early Warning System customized to Pakistan scenario. The system employed NodeMCU microcontroller, cheap environmental sensors, and Firebase cloud storage to monitor rainfall, water flow, and water level 24/7. Real-time monitoring via the web dashboard and alert system did help inform timely response, particularly in areas without more contemporary infrastructure.

During simulation test, the effectiveness of the system was validated. The results demonstrated sensitivity in diagnosis, data transmission efficiency, and alerting reliability. Affordability, accessibility and ease of upkeep were the focus of the design – 3 things that make it an ideal system for application in both urban, and rural areas. Moreover, the modular design made it easier to upgrade, and the interface made it possible for even non-technical people to run it over without fears.

For future studies, several improvements should be considered in order to improve the robustness, intelligence, and scalability of the system:

1. Predictive Analytics: Use machine learning algorithms (LSTM, Random Forest) to process historical and current data for flood prediction.
2. Enhanced Connectivity: Add NarrowBand IoT (NB-IoT), GSM or LoRaWAN protocols to extend the Opro9's edge to poor internet coverage rural areas.
3. Renewable Energy Integration: Use solar panels and rechargeable batteries install the device in any off-grid location and enjoy long-lasting operation.
4. Mobile Applications: Create Android - iOS applications for public and field participation.
5. Disaster Agency Link-up: Work together with local and national disaster management authorities to make the system fit well with emergency plans.
6. AI-Based Decision Making: Use edge computing and AI models for onsite data handling and wiser alerts.
7. Public Awareness Drives: Hold training programs to make sure the community knows and reacts well to flood warnings.

8. Scalable Cloud Migration: While Firebase suits the initial development, moving to more scalable platforms such as AWS IoT or Google Cloud IoT can improve performance and analytics at the national level.

To conclude, the project provides a hands-on, low-cost, and able-to-grow fix for Pakistan's flood watch problems. With right help from government and groups this setup can turn into an important tool in the country's disaster readiness and answer system which will save lives and cut down on money loss.

REFERENCES

Obinna, O. R., & Mathew, E. (2024). Design Analysis of an IoT-based Early Flood Detection and Alerting System, IJAERS.

Al Rawas, G., Nikoo, M. R., Al Wardy, M., & Etri, T. (2024). A Critical Review of Emerging Technologies for Flash Flood Prediction, Water, MDPI.

Rizal, M. H. et al. (2024). Design of Flood Early Detection Based on IoT and Decision Support System, IIETA.

Joshi, M. & Murali, S. (2025). Smart Flood Detection & Alert System using Automatic Recorder, IJCESEN.

Zainal, Z. A. & Po'ad, F. A. (2024). Smart Flood Monitoring System for Urban Road Closures, EEEE.

Kusumodestoni, R. H. et al. (2024). IoT Innovation for Flood Detection with Telegram Bot, JoCAAA.

Palefi Ma'ady, M. N. P. (2024). IoT-Based Flood Detection with Hybrid Fuzzy Logic and Neural Networks, TELKOMNIKA.

Shabbir Bukhari, S. A. et al. (2024). Enhancing Flood Monitoring Using ML and IoT Integration, Natural Hazards.

Abdelmagid, K. M. & Mohd Yakub, M. F. (2025). Early Flash Flood Detection Using Machine Learning, CTDS.

Salcedo, E. (2024). Graph Learning-Based Heavy Rainfall Prediction, arXiv.

Xu, Q. et al. (2024). FloodCast: Physics-Informed Neural Solver for Large-Scale Flood Forecasting, arXiv.

Zhang, Z. et al. (2024). IMAFD: Interpretable Multi-Stage Flood Detection from Multispectral Time Series, arXiv.

Te, M. C. L. et al. (2024). Smart IoT Urban Flood Monitoring via Pressure Sensor and LoRaWAN, HTIJ.

Balasingam, B. & Moorthy, A. (2018). IoT-Based River Water Level Monitoring Using SMS Alerts, ICDSE.

Mosavi, A. et al. (2019). Flood Prediction Using Machine Learning: A Review, arXiv.

Arshad, B. et al. (2019). Computer Vision and IoT-Based Sensors in Flood Monitoring: A Systematic Review, Sensors.

Ratmini et al. (2023). ESP32-Ultrasonic System with Blynk and Telegram Alert Mechanism.

Jeong, D. W. et al. (2017). Low-Cost IoT Flood Monitoring via LoRa Mesh, Sustainability.

Kornfeld, A. et al. (2018). LPWAN Solution for Autonomous Flood Monitoring, Sensors.

Marri, H. B. et al. (2019). LoRaWAN-Based Flood Monitoring System, IEEE Access.

Nordin, R. et al. (2017). NB-IoT for Rural Hydrological Monitoring, ICSIMA.

Dong, S. et al. (2020). Hybrid FastGRNN-FCN for Flood Prediction, arXiv.

Yadav, R. et al. (2022). Attentive Dual-Stream Siamese U-Net for Flood Detection, arXiv.

Meng, X. et al. (2019). Internet of Floods: Near Real-Time Detection, Water, MDPI.

Purkovic, D. et al. (2019). Smart River Monitoring in Japan, SMAGRIMET.

Al Qundus, J. et al. (2020). WSN for AI-Based Flood Detection, Annals of Operations Research.

Ubaidah, M. S. S. et al. (2022). IoT-Based Flood Detection with Forecasting, MJSAT.

Iqbal, F. R. et al. (2023). IoT Flood Early Warning: A Review, JoCPES.

Tibin M. Thekkil & N. Prabakaran (2017). WSN Early Flood Detection.

Science Publishing Group (2024). SentryLeaf IoT Flood Monitoring System.

PMCID (2023). Review on Sensor Technologies from 2000–2023