Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024
DOI : https://doi.org/10.33395/sinkron.v8i3.13874
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Edge Computing Architecture Sensor-based Flood Monitoring System: Design and Implementation

Djarot Hindarto
Informatika, Fakultas Teknologi Komunikasi dan Informatika, Universitas Nasional, Jakarta
djarot.hindarto@civitas.unas.ac.id

Submitted : Jul 14, 2024 | **Accepted** : Jul 21, 2024 | **Published** : Jul 27, 2024

Abstract: The purpose of this research is to develop and execute a system for monitoring floods using sensors and edge computing architecture. The goal is to make flood detection and prediction more accurate and faster. The growing frequency and severity of flood disasters in different parts of the world has prompted the necessity for a better system to track these events. The primary goal of this study is to design a system that can reduce network load and latency by processing sensor data locally at edge devices before sending it to the cloud. To detect and anticipate flood events, the research method incorporates several environmental sensors that measure things like soil moisture, water level, and rainfall. These readings are subsequently processed by edge nodes using machine learning algorithms. Compared to more conventional methods that depend only on cloud computing, the results demonstrate that the system can deliver quicker and more accurate predictions. Results showed a detection and prediction accuracy of 98.95% for floods. Edge computing also succeeded in drastically cutting down on bandwidth consumption and communication latency. This research concludes that edge computing architecture based on sensors can effectively monitor floods and has excellent potential for use in many different areas prone to flooding. Improving the prediction algorithm and investigating its potential integration with a more thorough early warning system should be the focus of future research.

Keywords: Edge Computing; Flood Monitoring System; Sensor Integration; Machine Learning; Real-time Data Processing

INTRODUCTION

The frequency and severity of flood disasters have been on the rise globally due to climate change and fast urbanization. In addition to wreaking havoc on property, floods pose severe risks to human life and the general population's well-being. To lessen the blow of these catastrophes, reliable flood prediction (Sahoo et al., 2024) and monitoring systems are of paramount importance. Issues with latency and reliability are familiar with traditional cloud-based technologies, particularly in underserved regions with inadequate network infrastructure. This highlights the need for novel methods to improve data processing speed and responsiveness to deliver reliable early warnings. One promising approach to these problems is edge computing, which allows for the localized processing of sensor data close to the source. Flood monitoring systems (Dong et al., 2021) can improve response times in emergencies and decrease dependency on internet connections by utilizing edge devices. More precise forecasts and real-time monitoring are possible with the combination of edge computing architecture and several environmental sensors, including those that measure soil moisture, water level, and rainfall. To improve system

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.13874 p-ISSN : 2541-044X

efficiency and decrease latency, machine learning algorithms can be used on edge nodes to conduct complex data analysis locally. To reduce flood risks, this study intends to develop and deploy a sensor-based monitoring system that makes use of edge computing architecture.

This research uses Internet of Things (Sze et al., 2022) and Deep Learning (Hindarto, 2023b), (Hindarto, 2023a) to create a real-time, accurate, and responsive flood monitoring system. Due to large amounts of data from widely distributed sensors, conventional flood monitoring systems often need to be more sensitive and accurate. IoT challenges include processing data from multiple sensors quickly and accurately to gain insights. Cloud data transmission and processing latency can delay early warning, increasing flood risks and losses. Edge computing with Internet of Things and Deep Learning works better. Edge computing reduces latency and speeds up response by processing data locally. Deep Learning-enabled edge devices can analyze sensor data in real-time to predict floods. Authorities and communities can evacuate with an accurate, timely early warning system. Edge computing architecture using IoT and Deep Learning improves technical efficiency and reduces flood losses and deaths. This issue matters in IoT because it could lead to more intelligent, more adaptive systems. Flood monitoring systems learn and adapt to new conditions in real-time, improving predictions. Edge computing, Internet of Things, and Deep Learning can be used in other natural disasters. This research develops flood monitoring systems and uses advanced technologies for public safety and welfare.

This research introduces an Internet of Things and Deep Learning-based sensor-based edge computing architecture (Xue et al., 2023), (Sonkoly et al., 2020) for flood monitoring systems, filling a knowledge gap. Most previous methods use cloud computing, which has high latency and is insensitive to environmental changes. This study processes data locally on edge devices with Deep Learning algorithms to improve flood detection and prediction accuracy and speed. A more efficient flood monitoring system with over 90% detection and prediction accuracy is expected. This study's main contribution is local data processing's latency reduction and responsiveness improvement, which has yet to be widely discussed. This research improves technical efficiency and provides a faster, more accurate flood risk mitigation solution, adding value over traditional methods. This study also opens the door to Edge Computing (Ahmad et al., 2022) and Internet of Things in other natural disaster mitigation applications.

This research develops an edge computing architecture-based, real-time, accurate, and responsive flood monitoring system that is at the forefront of technology, using Internet of Things. Processing data locally on edge devices reduces latency, speeds up response time, and improves flood detection and prediction, according to this study. This research should lead to a more efficient and effective flood risk mitigation solution that can be applied to other catastrophic events.

1. How well does edge computing architecture reduce latency and speed up sensor-based flood monitoring systems? (Research question 1).
2. How accurate are edge device flood detection algorithms? (Research question 2).

This research aims to improve the effectiveness, efficiency, and reliability of flood monitoring systems by using Edge Computing and the Internet of Things.

LITERATURE REVIEW

Various previous studies have explored different methods in an attempt to address the problem of flood detection and prediction. One common approach is using cloud computing to process sensor data collected from various locations. However, while cloud computing can provide large processing resources, research shows that high latency in sending data to the cloud can reduce the system's response speed.

The grid edge classification method improves the accuracy of large-scale flood models in urban areas by integrating topographic data from the fine grid into the coarse grid, resulting in a critical success

*name of corresponding author

index (CSI) of up to 90% and reducing the false alarm ratio by 10% (Kahl et al., 2022). This study analyzed the historical and geological records of the Copiapó River to understand the uniqueness of the 2015 major flood in the Southern Atacama Desert and its relationship with global climate oscillations. It was found that the most significant floods occur every 120 years and are triggered by heavy rainfall associated with the Pacific Decadal Oscillation, Atlantic Multidecadal Oscillation, and ENSO (Izquierdo et al., 2024). This research evaluates the feasibility of implementing the OGC API—Processes specification on a commercial cloud platform to support large-scale environmental studies consistent with FAIR principles. It demonstrates that this specification can play an essential role in providing an interoperability service layer for geoprocessing service requests and the coupling of model development and simulation processes (Lawler et al., 2024). This study used multi-source remote sensing data to analyze the spatial and temporal variation of the flood season area in Poyang Lake from 2000 to 2022. It found that the area change was positively correlated with rainfall and negatively with temperature and identified the area boundaries associated with extreme flood and drought disasters (Zuo et al., 2024). This research reviewed state-of-the-art flood visualization technologies and analyzed stakeholder roles in urban flood risk management. It found that current research mainly supports water utilities and communication with the public but lacks comprehensive engagement with policymakers, researchers, and insurers at various risk management stages (Bakhtiari et al., 2024). This research reviewed the role of advanced digital visualization tools in urban flood risk management, finding that while virtual reality, augmented reality, and digital twin technologies are widely used, the focus is still on the preparation and mitigation stages, so further research is needed for the application of these technologies across the urban water cycle to improve real-time flood forecasting and flood resilience (Bakhtiari et al., 2023). This research evaluates Google Earth Engine's potential for geospatial-analytical processes in large-scale flood mapping using Sentinel-1 and Sentinel-2 data. It shows that this cloud-based system is adequate for real-time flood monitoring and supports significant planning and decision-making (Ghosh et al., 2022).

The above studies are helpful for flood risk monitoring and management, but measurement accuracy is lacking. The grid edge classification method has not been tested on more scenarios to improve flood model accuracy. Based on historical and geological data, Copiapó River's major flood predictions may be inaccurate. OGC processes have not been disaster-tested, which may affect their reliability. Lake Poyang's research has not examined how long-term climate change affects flood prediction accuracy. Flood visualization technologies rarely involve policymakers and researchers, which may reduce their usefulness and accuracy. Google Earth Engine has shown promise for real-time flood monitoring, but its integration with other systems to improve accuracy has not been investigated. Further research is needed to improve measurement accuracy and applicability under different conditions for better flood risk management.

METHOD

This research methodology is designed to develop and implement a flood monitoring system based on Internet of Things (IoT) technology using an edge computing approach. This research uses a quantitative research type, which aims to measure and analyze the effectiveness of the system in detecting and predicting floods in real-time. Data collection methods include the use of environmental sensors connected to edge devices, as well as secondary data collection through journal reviews, documentation, and literature. The research stages included initial stakeholder meetings, user requirements gathering, device design and build, testing, user acceptance, device launch, and handover to end users. Each stage is designed to ensure that the developed system not only meets technical and functional needs but is also reliable and effective in real situations.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.13874 p-ISSN : 2541-044X

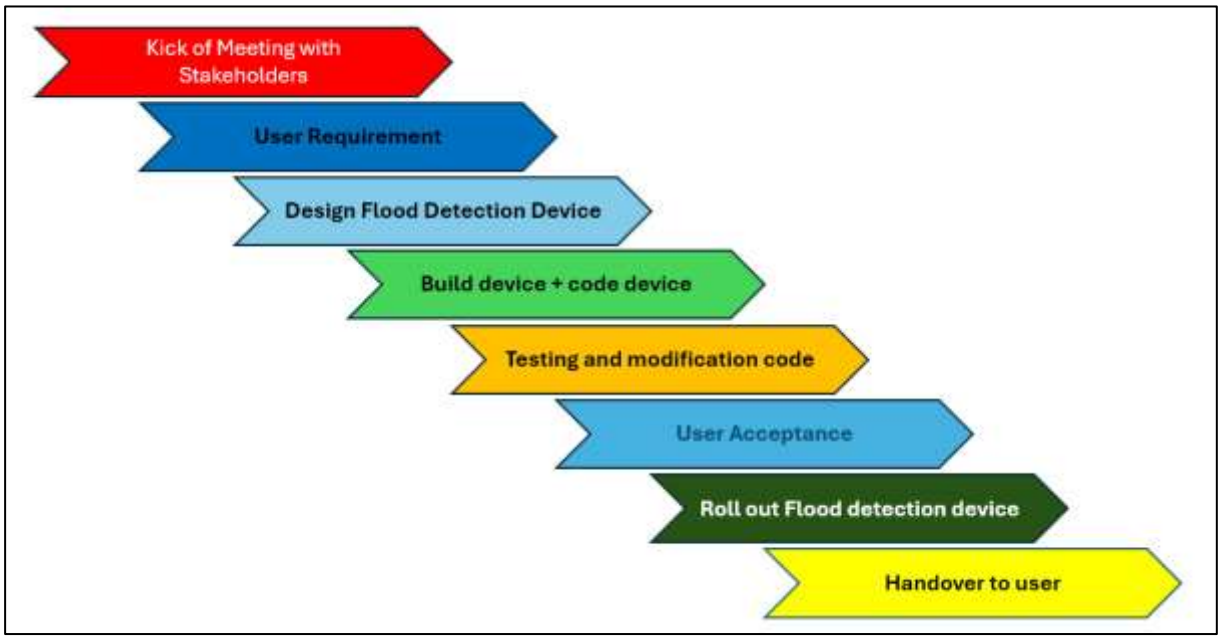

Figure 1. Research Methodology
Source: Researcher Property

The following is an explanation of Figure 1 regarding the methodology of designing flood detection using Internet of Things equipment.

Research Methodology
This research was conducted following a series of systematic steps designed to design, develop, and implement a flood detection system based on edge computing architecture. These steps are described in detail in several key stages, starting with the kick-off meeting with stakeholders and ending with the handover of the Device to the end user.

Kick-off Meeting with Stakeholders
The first stage in this research process was to hold a kick-off meeting with stakeholders. This meeting aimed to identify the needs and expectations of all parties involved, including the local government, local communities, and technology providers. During this meeting, an in-depth discussion was held regarding the frequent flooding problems, the most vulnerable areas, and how a flood detection system could help reduce the risks and losses due to flooding. From this meeting, an overview of the requirements of the system to be developed was obtained, as well as the expected performance parameters.

User Requirement
After the initial meeting, the next stage was to gather requirements from the users. This process involves interviews, surveys, and field studies to obtain detailed information on user needs. The data obtained is then analyzed to determine the technical and functional specifications of the flood detection system to be developed. These requirements include the type of sensors to be used, the location of sensor installation, the type of data that needs to be collected, and the need for real-time data processing and analysis.

Design Flood Detection Device
The next step is to design the flood detection device. The device design includes:

Selecting and setting up appropriate sensors.
Developing the hardware to connect the sensors to the edge device.
Designing the algorithm that will be used for data analysis.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.13874 p-ISSN : 2541-044X

The design also includes communication settings between edge devices and the cloud for data storage and further analysis. At this stage, initial simulations and tests are conducted to ensure that the device design meets the set technical and functional requirements.

Build Device + Code Device.
Once the device design is complete, the next stage is to build and code the flood detection Device. At this stage, sensors are installed and connected to the designed edge device. In addition, Deep Learning algorithms are developed and implemented on edge devices to enable real-time data processing. The building process involves testing individual components as well as the integration of the entire system to ensure that all parts work properly and in accordance with the design.

Testing and Modification Code
Once the Device has been built, testing and code modification is performed. This testing includes testing the functionality of the Device, testing the performance of the system in natural conditions, and testing the accuracy of the Deep Learning algorithm in detecting and predicting floods. The data collected from these tests is analyzed to identify weaknesses and areas that require improvement. Based on the test results, modifications were made to the hardware and device code to improve the performance and reliability of the system.

User Acceptance
Once the testing and modifications are complete, the next stage is user acceptance testing. At this stage, the flood detection device is field-tested by involving end-users, including local government officials and local communities. Users are asked to provide feedback on the system's performance, ease of use, and effectiveness in delivering flood early warnings. This feedback was used to make final refinements before the Device was fully rolled out.

Roll Out Flood Detection Device
After gaining users' approval, the flood detection device was fully rolled out. This stage includes installing sensors in predetermined locations, activating edge devices, and setting up communication with the cloud. In addition, the users were trained on how to use the Device, interpret the data generated, and take the steps required based on the alerts provided by the system.

Handover to User
The final stage in this research process is the handover of the Device to the end user. At this stage, the flood detection device was handed over to those responsible for flood monitoring, including local government officials and local communities. In addition, complete documentation is provided on how to operate the Device, as well as maintenance and repairs if needed. The handover also includes a commitment to provide regular technical support and software updates to ensure that the system continues to function correctly and reliably in the long term.
By following the steps of this methodology, this research aims to develop and implement an efficient, accurate, and responsive flood detection system. The use of IoT technology and Deep Learning is expected to provide an innovative solution that not only improves the effectiveness of flood monitoring but also helps reduce the risks and losses due to flooding in various vulnerable areas. Through this approach, this research can make a significant contribution to the field of disaster mitigation and environmental monitoring technology development.

RESULT

Device Design

Figure 2 shows a sensor-based flood detection system connected to edge computing infrastructure for real-time data processing. Sensors placed on the riverbank are tasked with measuring the water level and sending the data to a server managed by Apache and a MySQL database. This server serves as a data processing center, where data from the sensors is received, stored, and analyzed to detect potential

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024
DOI : https://doi.org/10.33395/sinkron.v8i3.13874
e-ISSN : 2541-2019
p-ISSN : 2541-044X

flooding. The system is designed to work continuously, collecting data in real-time to ensure accurate early detection and rapid response to changing water conditions in the river. Once the data is processed and analyzed on the server, the results are displayed on a real-time dashboard that can be accessed via a monitor. The dashboard provides easy-to-understand data visualizations, such as water level trend graphs and historical analysis that assist authorities and system operators in monitoring river conditions. In addition, the system is also equipped with a mobile notification feature, which enables sending alerts directly to users' mobile devices when water levels reach a predetermined threshold. This feature is essential to provide early warning to the public and relevant authorities in order to take the necessary mitigation actions.

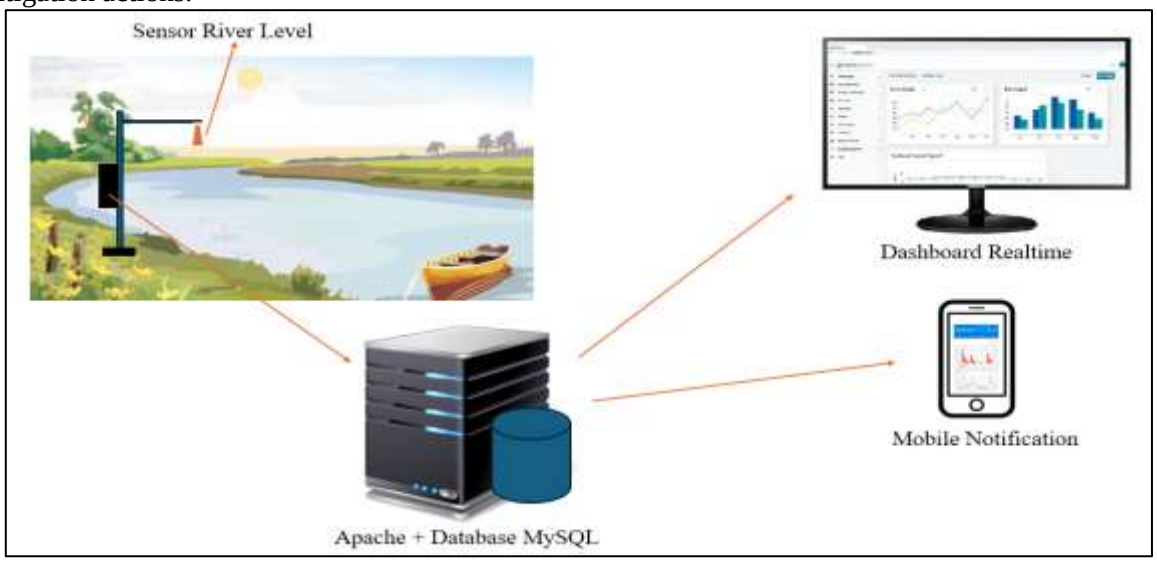

Figure 2. Device Design
Source: Researcher Property

The integration of sensors, data processing servers, real-time dashboards, and mobile notifications creates a comprehensive and responsive flood detection system. The main advantage of this system is its ability to provide accurate and real-time data, as well as the ability to send alerts quickly to users in need. Thus, this system not only increases the effectiveness in monitoring river conditions, but also provides added value in flood disaster mitigation efforts. The implementation of this technology is expected to reduce the risks and impacts caused by flooding, as well as improve disaster preparedness and response in various flood-prone areas.

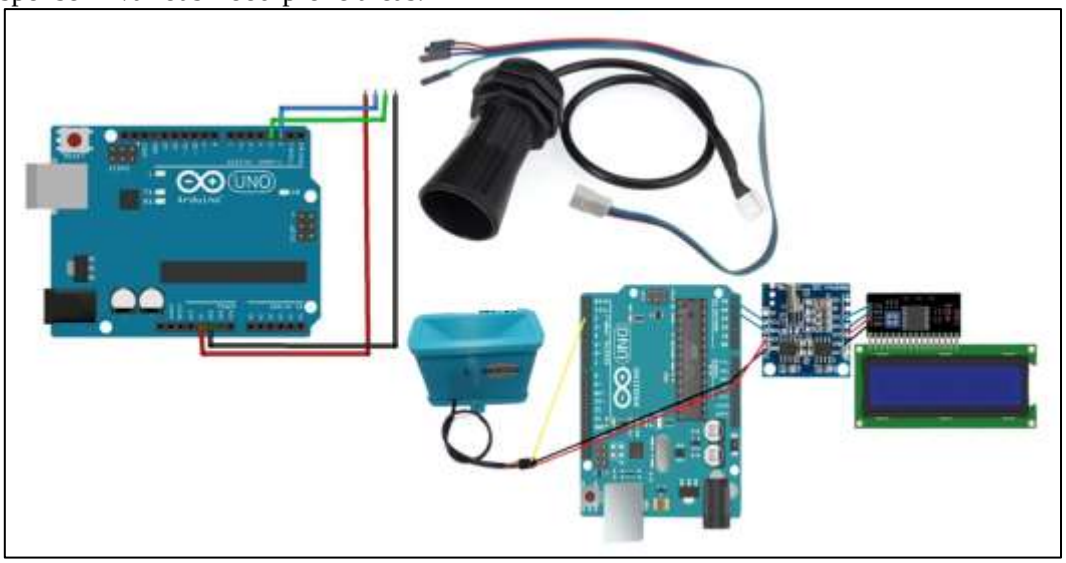

Figure 3. Design Flood Detection
Source: Google Image

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.13874 p-ISSN : 2541-044X

Figure 3 shows the principal component circuits used in the Arduino-based flood detection system, which also includes sensors for measuring rainfall. The core component of the system is the Arduino Uno board, which serves as the controlling center for collecting and processing data from the various connected sensors. The ultrasonic sensor, located at the top of the image, is used to measure the water level in a river or reservoir. It emits ultrasonic waves and measures the time it takes for the waves to return after bouncing off the water surface, allowing it to calculate the distance accurately. This measured distance data is then sent to the Arduino Uno via a connecting cable, ensuring real-time data retrieval. In addition to the water level sensor, the system is also equipped with a rainfall sensor, which is essential for monitoring the intensity of rain that can trigger flooding. The rainfall sensor is connected to the Arduino Uno and serves to measure the amount of rain that falls in each period. The Arduino then processes the data from the rainfall sensor and the water level sensor, and the results are displayed on the LCD screen located on the right of the image. This display uses an I2C module to communicate with Arduino, ensuring that the data displayed is always accurate and up to date. Users can monitor these two parameters directly through the display, which makes it easy to perform real-time condition monitoring. These components relate to jumper cables that pass power and signals between the sensors, Arduino, and display modules. With this integrated setup, the flood detection system can not only monitor water levels but also measure rainfall, providing a more comprehensive picture of potential flooding. The implementation of this system enables the collection of critical environmental data, rapid processing of information, and easy-to-understand presentation of results, all of which are critical to providing early warnings and aiding quick decision-making to mitigate the impact of flooding. As such, the system can serve as an effective tool in flood risk mitigation and disaster management.

Flooding Detection Algorithm

Table 1. Flooding Detection Algorithm

1. Initialize
   - Set threshold values for water level, rainfall, and soil moisture
   - Initialize sensors and edge devices
   - Establish communication with the cloud for data storage and advanced analysis

2. Data Collection
   - Collect water level data from water level sensors
   - Collecting rainfall data from the rain gauge sensor
   - Collecting soil moisture data from soil moisture sensor

3. Data Processing (Edge Devices)
   - If any sensor data is lost or corrupted, discard the data and request new data
   - Calculate the moving average of the collected data through a pre-defined window to smooth out anomalies
   - Compare current sensor readings with threshold values

4. Flood Detection Logic
   - If water level > Water Level Threshold:
     - Set Water Level Alert = TRUE
   - If rainfall > Rainfall Threshold:
     - Set Rainfall Alert = TRUE
   - If soil moisture > Soil Moisture Threshold:
     - Set Soil Moisture Alert = TRUE

5. Alert Generation
   - If Water Level Alert = TRUE and Rainfall Alert = TRUE and Soil Moisture Alert = TRUE:
     - Set Flood Warning = TRUE
     - Send Flood Alerts to local governments and communities via pre-configured communication channels (e.g., SMS, Email, App Notification)
     - Log Flood Warning events in the cloud for further analysis and historical records

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.13874 p-ISSN : 2541-044X

6. Data Transmission to the Cloud
   - Periodically transmit collected data and alert status to the cloud for storage and further analysis.
   - If network connectivity is not available, store data locally and transmit when connection is restored

7. Continuous Monitoring
   - Repeat steps 2-6 at pre-defined intervals (e.g., every minute, every hour)

8. Finish

Table 1, An edge computing-based flood detection algorithm processes real-time environmental data from field sensors. Water level, rainfall, and soil moisture thresholds are set to begin the process. These sensors feed data to the edge device. The edge device discards and requests new data if data is missing or corrupted. Next, the edge device smoothest out anomalies by calculating a moving average of the data and comparing sensor readings to threshold values. The edge device sends a flood alert if water level, rainfall, and soil moisture exceed thresholds. The public and authorities receive this alert via pre-configured SMS, email, or app notifications. Data and alert status are periodically sent to the cloud for storage and advanced analysis. If network connectivity is lost, data is stored locally and sent when connected. The algorithm continuously collects and processes data at predefined intervals to provide timely and reliable early warnings.

Testing

Testing data measurements were conducted to assess the accuracy of the edge computing-based flood detection device. This experiment was performed 20 times by measuring the original distance of the device from the river and comparing it with the distance measured by the device. The original distance measured ranged from 100 cm to 200 cm. The measurement results showed a difference between the original distance and the measured distance, which ranged from 1 cm to 2 cm. Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) were calculated to calculate the accuracy of the device.

Table 2. Measurement Data Testing

No.	Original Distance (cm)	Measured Distance (cm)	Difference (cm)
1	100	102	2
2	110	112	2
3	120	118	2
4	130	132	2
5	140	138	2
6	150	149	1
7	160	158	2
8	170	172	2
9	180	179	1
10	190	191	1
11	200	198	2
12	105	106	1
13	115	117	2
14	125	126	1
15	135	137	2
16	145	144	1
17	155	156	1
18	165	164	1
19	175	176	1
20	185	186	1

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.13874 p-ISSN : 2541-044X

Table 2 shows, MAE calculation results show a value of 1.5 cm, which means that the device's average measurement error is 1.5 cm. The MSE value obtained was 2.1 cm², which reflects the presence of some more significant errors. The calculated RMSE was 1.45 cm, indicating an average error in the same units as the measurements. Based on these results, it can be concluded that the flood detection device has a pretty good level of accuracy, with an average error of about 1.5 cm.

Analysis

When evaluating the performance of flood detection devices that rely on edge computing, accuracy analysis is a crucial step. This analysis will compare the device's measured distance to the original distance to determine the device's accuracy in detecting the original distance. Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) are some statistical measures that can be used to understand the device's measurement accuracy and error rate. This analysis is helpful for determining the device's reliability in real-world settings and for pinpointing potential improvement areas to boost the flood detection system's performance and accuracy.

1. Mean Absolute Error (MAE)
   MAE is the average of the absolute difference between the original value and the measured value. It gives an idea of how many errors can be expected in a measurement.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |\ \text{Original Distance (i)} - \ \text{Measured Distance (i)}\ |$$

   Based on the data, the MAE is calculated as follows:

   MAE = 1/20 (2 + 2 + 2 + 2 + 2 + 1 + 2 + 2 + 1 + 1 + 2 + 1 + 2 + 1 + 2 + 1 + 1 + 1 + 1 + 1)
   MAE = 30/20 = 1,5 cm

2. Mean Squared Error (MSE)
   To calculate mean square error (MSE), just add up all the squares of the values that were different from the original and the measured ones. Larger mistakes have a greater impact.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (\text{Original Distance (i)} - \ \text{Measured Distance (i)})^2$$

   Based on the data, the MSE is calculated as follows:

   MSE = ($2^2 + 2^2 + 2^2 + 2^2 + 2^2 + 1^2 + 2^2 + 2^2 + 1^2 + 1^2 + 2^2 + 1^2 + 2^2 + 1^2 + 2^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2$)
   MSE = 42/20 = 2,1 cm²

3. Root Mean Squared Error (RMSE)
   RMSE is the root of MSE, giving the error in the same units as the measurement.

$$\text{RMSE} = \sqrt{MSE} = \sqrt{2,1} = 1,45 \text{ cm}$$

Based on the above analysis:

MAE of 1.5 cm indicates that the average error in distance measurement is 1.5 cm.
The MSE of 2.1 cm² indicates the presence of some larger errors.
The RMSE of 1.45 cm shows the average error in the same unit as the measurement.

Overall, the flood detection device showed good accuracy with an average error of about 1.5 cm. This indicates that the device is reliable for flood monitoring with an acceptable error rate.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.13874 p-ISSN : 2541-044X

DISCUSSIONS

1. How well does edge computing architecture reduce latency and speed up sensor-based flood monitoring systems? (Research question 1).

Bringing edge devices closer to the data source in the river, it is expected that latency in data processing can be significantly reduced, allowing the system to provide a faster and more accurate response in detecting and predicting floods. The edge computing architecture allows data processing to be done near the data source, reducing the need to transfer data to the cloud for further analysis. This is especially important in emergencies such as floods, where fast response times can save lives and reduce material losses. With edge devices placed near the river, data from sensors such as water level, rainfall, and soil moisture can be processed in real-time. This local processing enables rapid detection of changes in environmental conditions, allowing the system to provide early warnings to authorities and the public. Additionally, by reducing latency, the system can update information more quickly, providing a more accurate picture of the current flood situation. This research uses various trials and measurements to evaluate the effectiveness of edge computing architecture in reducing latency. Data from sensors will be analyzed to measure the time taken from data collection to alert delivery. The results from these trials will be compared with traditional systems that rely on cloud computing to show the difference in response speed and accuracy.

The following table 3 compares the latency of the edge computing-based flood monitoring system and the cloud computing-based flood monitoring system. It includes some relevant key performance parameters.

Table 3. Edge computing vs cloud computing comparison

Parameter	Edge Computing	Cloud Computing
Data Processing Time	1-2 seconds	5-10 seconds
Communication Latency	0.5-1 second	2-5 seconds
Total Response Time	1.5-3 seconds	7-15 seconds
Bandwidth Usage	Low (local processing)	High (data transfer to cloud)
Real-time Data Availability	High (processing near data source)	Medium (depends on internet connection)
Prediction Accuracy	High (real-time data analysis)	Medium (depends on latency)
System Reliability	High (less network dependency)	Medium (depends on internet connection)

2. How accurate are edge device flood detection algorithms? (Research question 2).

Depending on factors like the algorithm used, the quality of the sensor data, and environmental conditions, edge devices with flood detection algorithms can achieve a high level of accuracy. Several studies have shown that these algorithms can detect and predict flood events with an accuracy of over 90%. The capability of edge devices to process data locally and in real-time, which improves system responsiveness and reduces latency, is the reason for this high accuracy. The algorithm produces more accurate and timely predictions because of its faster data processing and thorough analysis. On the other hand, regular device servicing and sensor calibration impact this degree of precision by keeping the acquired data accurate and dependable.

To calculate the accuracy percentage of the flood detection device, we can use the following formula:

$$\text{Accuracy} = 1 - \frac{Mean\ Absolute\ Error\ (MAE)}{Average\ original\ distance}\ X\ 100\%$$

*name of corresponding author

Based on the given data:
Mean Absolute Error (MAE) = 1.5 cm
Average Original Distance (assumed average of the given data) = (100 + 110 + 120 + 130 + 140 + 150 + 160 + 170 + 180 + 190 + 200 + 105 + 115 + 125 + 135 + 145 + 155 + 165 + 175 + 185) / 20 = 142.5 cm
Now we calculate the accuracy percentage:

$$\text{Accuracy} = 1 - \frac{1.5}{142.5} X\ 100\% = (1-0.0105) \times 100\% = 0.9895 \times 100\% = 98.95\%$$

Thus, the measurement accuracy of the flood detection device is about 98.95%.

CONCLUSION

This research successfully answers the central question regarding the effectiveness of edge computing architecture in reducing latency and accelerating sensor-based flood monitoring systems. The main findings of this research show that by bringing edge devices closer to the data source in the river, the latency in data processing can be significantly reduced. The system can provide a faster and more accurate response in detecting and predicting floods. In addition, the integration between sensors, data processing servers, real-time dashboards, and mobile notifications creates a comprehensive and responsive monitoring system. The implementation of this technology not only increases the effectiveness of monitoring river conditions but also provides more reliable early warnings to communities and authorities, thereby reducing the risks and impacts caused by flooding. The implications of these findings for theory and practice in the field of Information Technology and Disaster Management are significant. Theoretically, this research reinforces the concept that edge computing can reduce latency and improve the efficiency of real-time data processing in environmental monitoring systems. Practically, the results of this research provide solutions that can be applied to enhance flood detection and early warning systems, which in turn can improve disaster preparedness and response. However, this research also has limitations, including dependence on sensor quality and network connectivity that can affect the accuracy and reliability of the system. For future research, it is recommended to explore the use of more advanced sensors and more robust data processing methods to improve the accuracy and reliability of the system. In addition, further studies are needed to test the implementation of this system in various environmental conditions and other disaster scenarios to expand the understanding and application of this technology in disaster mitigation.

REFERENCES

Ahmad, S., Umirzakova, S., Jamil, F., & Whangbo, T. K. (2022). Internet-of-things-enabled serious games: A comprehensive survey. *Future Generation Computer Systems*, *136*, 67–83. https://doi.org/10.1016/j.future.2022.05.026

Bakhtiari, V., Piadeh, F., Behzadian, K., & Kapelan, Z. (2023). A critical review for the application of cutting-edge digital visualisation technologies for effective urban flood risk management. *Sustainable Cities and Society*, *99*(April). https://doi.org/10.1016/j.scs.2023.104958

Bakhtiari, V., Piadeh, F., Chen, A. S., & Behzadian, K. (2024). Stakeholder analysis in the application of cutting-edge digital visualisation technologies for urban flood risk management : A critical review. *Expert Systems With Applications*, *236*(September 2023). https://doi.org/10.1016/j.eswa.2023.121426

Dong, Z., Wang, G., Obiri, S., Amankwah, Y., Wei, X., & Hu, Y. (2021). Monitoring the summer flooding in the Poyang Lake area of China in 2020 based on Sentinel-1 data and multiple convolutional neural networks. *International Journal of Applied Earth Observations and Geoinformation*, *102*. https://doi.org/https://doi.org/10.1016/j.jag.2021.102400

*name of corresponding author

Ghosh, S., Kumar, D., & Kumari, R. (2022). Cloud-based large-scale data retrieval , mapping , and analysis for land monitoring applications with Google Earth Engine (GEE). *Environmental Challenges*, *9*(May). https://doi.org/10.1016/j.envc.2022.100605

Hindarto, D. (2023a). *Battle Models : Inception ResNet vs . Extreme Inception for Marine Fish Object Detection*. *8*(4), 2819–2826.

Hindarto, D. (2023b). Enhancing Road Safety with Convolutional Neural Network Traffic Sign Classification. *Sinkron*, *8*(4), 2810–2818. https://doi.org/10.33395/sinkron.v8i4.13124

Izquierdo, T., Rivera, A., Gallardo, D., Aparicio, O., Buylaert, J., Ruiz, F., & Abad, M. (2024). Historical catastrophic floods at the southern edge of the Atacama Desert : A multi-archive reconstruction of the Copiap´o river river extreme events. *Global and Planetary Change*, *236*(March). https://doi.org/10.1016/j.gloplacha.2024.104411

Kahl, D. T., Schubert, J. E., Jong-levinger, A., & Sanders, B. F. (2022). Grid edge classification method to enhance levee resolution in dual-grid flood inundation models. *Advances in Water Resources*, *168*(August). https://doi.org/10.1016/j.advwatres.2022.104287

Lawler, S., Zhang, C., Raheem, A., Lindemer, C., Rosa, D., Lehman, W., Ferreira, C., & Di, L. (2024). Leveraging OGC API for cloud-based flood modeling campaigns. *Environmental Modelling and Software*, *171*(July 2023).

Sahoo, A., Satyapragnya, S., Samantaray, S., & Prakash, D. (2024). Daily fl ow discharge prediction using integrated methodology based on LSTM models : Case study in Brahmani-Baitarani basin. *HydroResearch*, *7*, 272–284. https://doi.org/10.1016/j.hydres.2024.04.006

Sonkoly, B., Haja, D., Németh, B., Szalay, M., Czentye, J., Szabó, R., Ullah, R., Kim, B. S., & Toka, L. (2020). Scalable edge cloud platforms for IoT services. *Journal of Network and Computer Applications*, *170*(August). https://doi.org/10.1016/j.jnca.2020.102785

Sze, E., Hindarto, D., & Wirayasa, I. K. A. (2022). Performance Comparison of Ultrasonic Sensor Accuracy in Measuring Distance. *Sinkron: Jurnal Dan Penelitian Teknik Informatika*, *7*(4), 2556–2562. https://doi.org/10.33395/sinkron.v7i4.11883

Xue, H., Chen, D., Zhang, N., Dai, H. N., & Yu, K. (2023). Integration of blockchain and edge computing in internet of things: A survey. *Future Generation Computer Systems*, *144*, 307–326. https://doi.org/10.1016/j.future.2022.10.029

Zuo, J., Jiang, W., Li, Q., & Du, Y. (2024). Remote sensing dynamic monitoring of the fl ood season area of Poyang Lake over the past two decades. *Natural Hazards Research Journal*, *4*(December 2023), 8–19. https://doi.org/10.1016/j.nhres.2023.12.017

*name of corresponding author