International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 5 May 2023, pp: 956-965www.ijaem.net ISSN: 2395-5252

# Flood Monitoring System in Rwanda Using Internet of Things.

Mukamana Florentine[1], NyeshejaMuhire Enan[2],
Sibomana Louis[3]

[1]*African Center of Excellence in Internet of Things/ University of Rwanda /College of Science and Technology,Kigali, Rwanda*
[2]*Dean Faculty of Computing and Information Sciences, University of Lay Adventists of Kigali, Rwanda*
[3]*Head of Department, Science, Technology Development and Outreach at NCST, Rwanda, and Researcher at University of Rwanda*

---------------------------------------------------------------------------------------------------------------------------------------
Date of Submission: 15-05-2023 Date of Acceptance: 30-05-2023
---------------------------------------------------------------------------------------------------------------------------------------

**ABSTRACT**: A large amount of water overflowing is a natural disaster that occurs worldwide, causing significant harm to the environment. One issue that arises during flash floods is that residents may not have sufficient time to relocate their valuable possessions to safer areas. To address this problem, it is necessary to create technology that can monitor and alert individuals about potential flooding. This research presents an original flood monitoring system developed specifically for the Nyabugogo flood in Kigali, Rwanda. The system is designed to gather, monitor and analyze data from two different sources Mpazi and Rwampara Brooks. Each location is equipped with a flood detection system that includes an ultrasonic sensor for measuring water level changes, a raindrop sensor for precipitation intensity measurement, and a NodeMCU device for data collection. The water level is a crucial indicator of potential flooding, and the collected data is transmitted to ThingSpeak which is a web-based application, using the MQTT protocol. An alert message is then sent directed toward governmental institutions based on a predetermined limit value (threshold value). By using this flood monitoring system, public authorities may stay informed about rising water levels and potential floods.

**KEYWORDS:**NodeMCU, IoT, Flood, MQTT, ThingSpeak, IFTTT.

## I. INTRODUCTION

Flooding is a global issue resulting from poor water resource management that causes negative impacts such as erosion and pollution. Rwanda has been facing severe flooding due to heavy rainfall and steep slopes, resulting in loss of life and damage. A possible solution to mitigate the damage caused by floods is through the implementation of a flood monitoring system that uses advanced technologies such as the Internet of Things (IoT). The IoT technology has been utilized in various sectors, including smart cities, education, and healthcare. This research paper proposes the use of IoT in a flood monitoring system.

The flood monitoring systems using IOT have caught the attention of the research community. J. W. Simatupang and colleagues [1] employed Arduino and ultrasonic sensor components to detect floods and send alerts to end-users through GSM and GPRS technologies. However, the use of these technologies could negatively impact Arduino's functionality, leading to reduced module functionality and low responsiveness [2]. To enhance the stability and security of the system and improve signal reliability, an Ethernet shield could be added. Nonetheless, the absence of real-time monitoring of parameter changes is a limitation of this method and is associated with Costly charges due to the additional charges from GSM, GPRS and Ethernet shield expenses. Additionally, the system's responsiveness is poor.

Further, in a previous study [3], the authors developed a system utilizing Arduino and ESP8266 to create a WiFi network, which stores sensor information in a database. They also created an Android application for registered users to access real-time data from these sensors. However, this system has limitations. It could be improved by using the NodeMCU ESP8266 as both storage and WiFi, but currently, it only allows registered users with compatible devices to access the data. Additionally, it only considers one river as the source of flooding and may not be cost-effective.

Moreover, in [4], a novel solution has been implemented to help mitigate the consequences of disasters. The system comprises two sensors that are ultrasonic sensor and water flow sensor. It also uses a NodeMCU ESP8266 as a microcontroller as well as a Blynk application. To enhance its capabilities, the system can incorporate a that can keep track of variations in rainfall levels. A raindrop sensor can be integrated into the system. In [5], another system has been proposed to predict various parameters including water level, weather conditions, and water flow. The system communicates real-time data from sensors to a mobile application. However, it lacks an emergency alert feature.

Furthermore, the authors in [6] presented a system that enables the government's emergency cell management to receive timely risk alerts during emergencies. The system leverages temperature, humidity, and level sensors to collect relevant data, which is then processed using the ESP8266 microcontroller. During emergencies, the system displays the collected data and sends alert messages to citizens. One way to enhance the system could be to allow users to view sensor data in real-time, which would provide additional benefits.

Additionally, a group of researchers A. Diriyana et al. [7] utilized various technologies to create a system for displaying data and providing flood early warning information. Their system involved the use of an Arduino Uno microcontroller board to present information, a NodeMCU for data management and storage, an ultrasonic sensor for distance detection, the ThingSpeak web application for displaying and notifying real-time information by using the Telegram platform. While their system was effective, it was not cost-efficient. To improve its cost-effectiveness, the researchers suggest using NodeMCU to both display and manage data. They recommend incorporating a device to detect rainfall intensity in the vicinity.

This project aims to generate a system for detecting floods that can gather information on the factors that contribute to flooding in a particular region by detecting changes in those variables. The system will also determine the severity of the flood, enabling timely action to be taken.

It is worth noting that previous research studies [1-7] have focused solely on a single river or location as the main contributor to flooding. In contrast, this paper introduces several key contributions, including the use of two distinct locations (Rwampara brook and Mpazi brook) to accurately predict incoming floods. The study also leverages NodeMCU esp8266, which is equipped with an Ethernet shield, to effectively connect and transfer data. Additionally, public authorities are notified about upcoming floods, enabling them to inform citizens on time. Finally, the study presents detailed channel statistics based on real-time data entries over time.

In this article, the issue of flooding in Kigali City, Rwanda, is addressed through the utilization of IoT technology. A system for tracking floods has been developed, which utilizes ultrasonic and raindrop sensors to detect input parameters. If flooding is detected, A notification is issued and conveyed to the responsible decision-makers through the use of the ThingSpeak application and IFTTT. A prototype of this system has been developed and tested to ensure its effectiveness.

The system being suggested is deployed in two brooks Mpazi and Rwampara, both of which play a role in causing the Nyabugogo flood. The intention is to provide real-time information about flooding, aiding authorities in the efficient protection of the environment. This output serves as a crucial aid for decision-makers in addressing the impact of flooding not just in Rwanda, but also in other mountainous areas around the globe.

The organization of this paper is as follows. In Section 2, we present an overview of the classification model and description. The used materials and the system analysis are presented in section 3 and section 4 respectively. The results and interpretations of the system are presented in Section 5. It summarizes the findings and proposes avenues for future research and recommendations. Finally, there are acknowledgments and references.

## II. CLASSIFICATION MODEL AND DESCRIPTION

This section presents an original classification model for an IoT_based flood monitoring system, which comprises both tangible and intangible components.

### II.1 The overall classification model

The flood monitoring classification model which illustrates two locations of a flood. Each location has a channel flood detector consisting of a microcontroller and sensors. The sensors measure the height of the water in the river and the intensity of the rain around the river. The information collected by the sensors is transmitted to the ThingSpeak web server via the built-in WiFi of NodeMCU. ThingSpeak serves as a flood monitoring system, where the data received from both flood detector systems is saved, examined, and displayed graphically. The IFTTT which is a web server is used to send message notifications to users, alerting them about the current state of affairs.

Fig -1depicts the flood monitoring classification model.

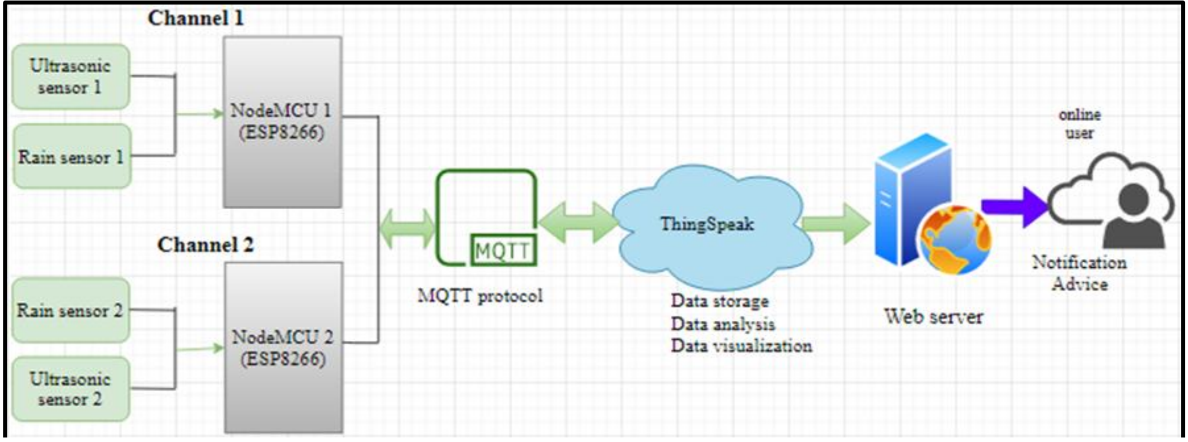


Fig -1: FLOOD MONITORING CLASSIFICATION MODEL

### II.2 The process of connecting the hardware components

In the workshop, an IoT-based flood monitoring system was built and set up on a circuit board. The hardware connection of the system, as shown in Fig -2 involved connecting two flood detector system to a computer for testing and monitoring purposes. The software used included Arduino IDE, ThingSpeak and the If This Then That (IFTTT) application.Both detectors share a common hardware setup consisting of a breadboard, nodeMCU, HC-SR04 ultrasonic sensor, 5 LEDs and buzzer. The first flood detector is designated for the Mpazi river location while the second one is designated for the Rwampara brook location.

The flood detector system monitors the water levels in each stream and the intensity of rainfall. The increase in water is measured using HC-SR04 ultrasonic device. A rain drop instrument is employed to detect rain on its surface, displaying analog values for rain intensity. The nodeMCU facilitates the transmission of information to web applications, and Light-emitting diodes offer extensive status signals for flooding conditions. Data in ThingSpeak is both saved and evaluated with the MQTT protocol. For authorized users, ThingSpeak issues warning messages with the assistance of IFTTT.

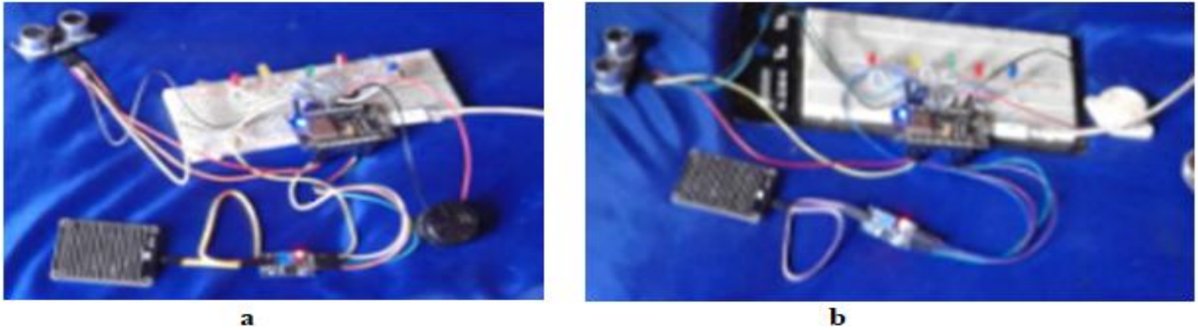

a
b

Fig -2: a. FLOOD DETECTOR IN MPAZI b. FLOOD DETECTOR IN RWAMPARA

The flood detector system monitors the water levels in each stream and the intensity of rainfall. The increase in water is measured using HC-SR04 ultrasonic device. A rain drop instrument is employed to detect rain on its surface, displaying analog values for rain intensity. The nodeMCU facilitates the transmission of information to web applications, and Light-emitting diodes offer extensive status signals for flooding conditions. Data in ThingSpeak is both saved and evaluated with the MQTT protocol. For authorized users, ThingSpeakissues warning messages with the assistance of IFTTT.

### II.3 System Flowchart

A visual representation outlining the various processes and steps within the system is depicted in the flowchart that is illustrated in Fig -3. After initialization, the NodeMCU establishes a connection to the Wi-Fi and MQTT protocol. It then proceeds to receive data from both sensors. The measurements from the ultrasonic sensor are evaluated against pre-programmed threshold values. The range of these readings typically falls between 0 to 400 centimeters, as reported in prior studies [8].

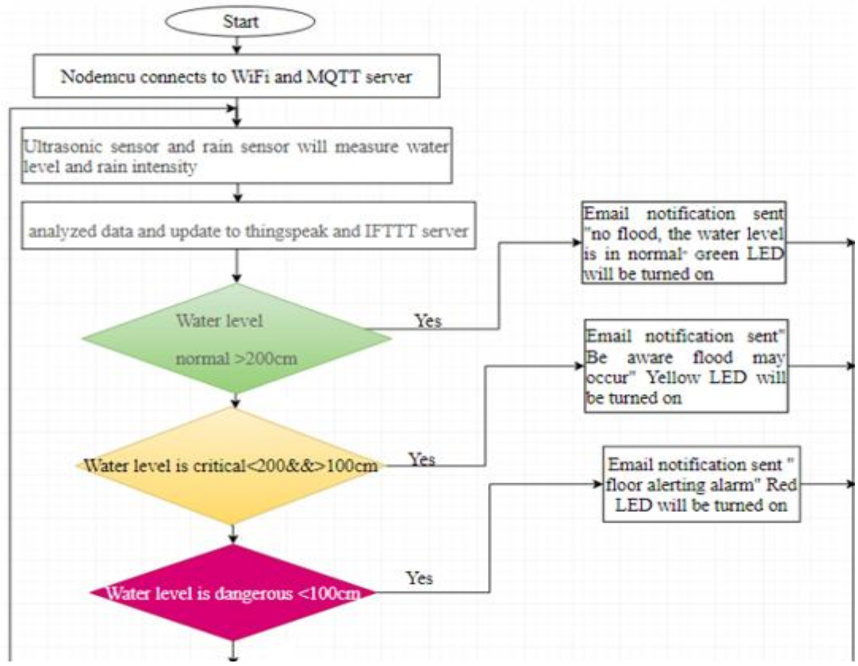


Fig -3: SYSTEM FLOWCHART

The readings recorded by the ultrasonic sensor are classified into three categories based on the risk of flooding. To capture readings, the flood detection system incorporates an ultrasonic sensor that is mounted onto the surface of the water.When the water is in normal condition, a notification is sent indicating that the level of water is ordinary and there is no risk of overflowing. If the water level is critical, a notification is sent to warn that there may be a risk of flooding and caution is advised. If the water level is unsafe, an alert message is sent and a buzzer is activated to sound a flood alert. The water level and rainfall intensity in the area around the stream are monitored using NODEMCU where ThingSpeak receives the data and uses it to refresh the web application with up-to-date information. If the predefined threshold value is satisfied, ThingSpeakwill automatically activate an external application called IFTTT.

## III. USED MATERIALS

### III.1 HC-SR04 ultrasonic sensor

The HC-SR04 Ultrasonic sensor is commonly used distance measuring sensor in electronics and robotics projects. It is specifically designed to measure the distance between the surface of water and the sensor and can accurately measure distances between 2 centimeters and 4 meters. The HC-SR04 sensor consists of four pins that are Vcc, GND, Trig, and Echo and operates on a 5V power supply[9]. It is a relatively inexpensive and easy-to-use sensor.

Fig -4: HC-SR04 ULTRASONIC SENSOR

### III.2 Raindrop sensor

The raindrop sensor is an instrument that measures the intensity of rainfall and is useful in determining the amount of precipitation that could potentially cause a rise in water levels. The sensor detects raindrops using a conductive surface that changes its conductivity when a rain droplet falls on it. The sensor provides critical data on rainfall intensity and duration that is used in an IoT flood monitoring system to predict potential flood situations and mitigate their impact. When there is no rainfall, the intensity of precipitation ranges between 500 and 1023, while moderate rainfall falls within the range of 300 to 500, and heavy rainfall causes the sensor to vary beyond 300. Fig -5 shows the image of raindrop sensor.

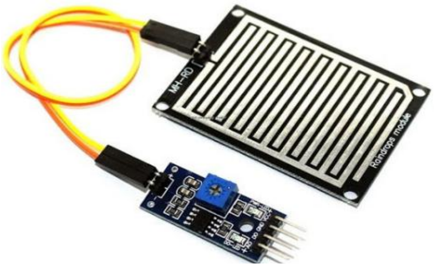


**Fig -5: RAINDROP SENSOR**

### III.3 NodeMCU

NodeMCU is similar to the Arduino IDE in that it has all of the pin functionality of NodeMCU boards. The actual GPIO pin can be found behind the board, making it easy to program with any compiler. NodeMCU has a significant advantage with its Wi-Fi API feature, which enables it to connect to various devices and retrieve information from the internet. This feature enables users to control their devices from anywhere in the world. NodeMCU has a limitation in that it only supports a single ADC, resulting in the device possessing just one analog pin. The resolution of the ADC pin is 10 bits allowing it to read values between 0 and 1024. The maximum voltage that can be read by the ADC pin is 1V, and any reading above this, up to the Vcc of 3.3V, is set to the maximum value of 1024[10].

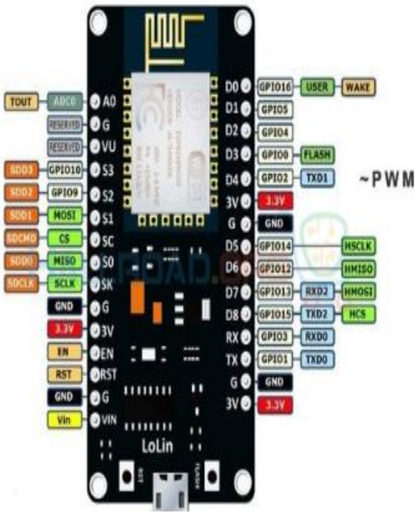


**Fig -6: NodeMCU**

### III.4 ThingSpeak

ThingSpeak is a freely accessible platform that has been developed specifically for the analysis of Internet of Things (IoT) data. It allows internet-connected devices to send data using the MQTT protocol [11][12]. It offers data storage, analysis, and visualization capabilities. The platform can be used in three main steps: first, creating a channel and collecting data, secondly analyzing and visualizing the data online, and lastly acting based on the data[13]. In this study, the sensor data was collected by connecting the sensors to a NodeMCU device, which then uploaded the data to ThingSpeak using the channel number, write API, and author details. To connect NodeMCU to the network, a network SSID and password were used. We then employed the ThingSpeak channel to keep track of the water level and the intensity of the rainfall.

### III.5 IFTTT

IFTTT is a free online service that enables users to create applets which perform an action when a trigger occurs. In our study, we developed an applet consisting of a single app web hook[14]. This applet can be created by following a series of steps using any internet-connected computer [15]. Once created, an URL is generated which can be used to send alerting messages when a particular trigger condition is fulfilled. By using IFTTT, we are able to send push notifications. To trigger notifications through IFTTT, we utilize the ThingHTTP app.

## IV. SYSTEM ANALYSIS

In Fig -7, we can see the results of the system analysis. The flood detector system is capable of detecting both water levels and rain intensity. Afterward, the flood monitoring data is transmitted to the server of the information system through Wi-Fi. This system for monitoring floods processes data in real-time and makes it available for visualization. Users can access and view the data online.

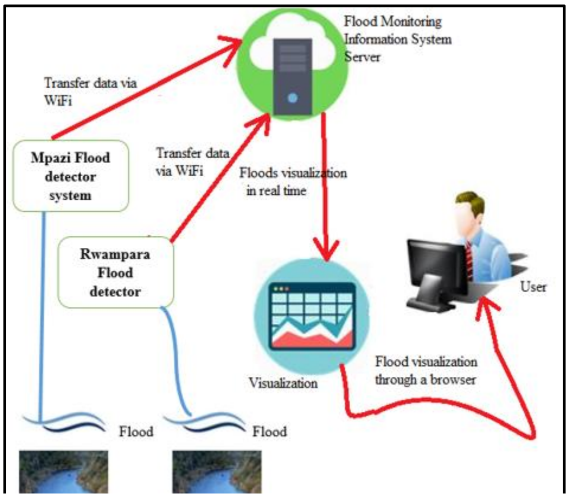


**Fig -7: FLOOD MONITORING SYSTEM ANALYSIS**

There are multiple software tools available to support the development of a flood detection system. Xampp and web programming components are two well-known examples. Xampp is a free and open-source cross-platform software package that provides a local web server environment for building and testing web applications.

In this system, we used xampp to create and test a web-based user interface that displays real-time data from sensors. It serves as a secure platform for storing data gathered by the flood detector system in a database that is accessible to users. Additionally, the web programming aspect of Xampp plays a crucial role in facilitating web development and presenting flood-related information in a visually appealing manner.

## V. RESULTS AND INTERPRETATIONS

In this chapter, we present our findings and provide a detailed explanation of the found results. To test the prototype of our flood monitoring system, we assembled the breadboard with nodeMCU and sensors, as shown in Fig-2. The system was then tested At Rwampara Brook and Mpazi Brook, which are two separate locations. By establishing a remote connection between NodeMCU and the web services (ThingSpeak and IFTTT). We were able to update the fields of the ThingSpeak channel with live data and trigger notifications based on the readings. Specifically, public authorities were alerted via email by utilizing the combined functionalities of ThingSpeak and IFTTT.

The results of the classification model show that nodeMCU was used to read the values of two flood detectors. Each detector sent its readings to ThingSpeak, which in turn the fields were refreshed with the latest data. By visualizing and analyzing the live information regarding the Mpazi and Rwampara streams. ThingSpeak was able to provide valuable insights into flood patterns and potential risks.

To evaluate the flood monitoring system in Rwanda, a connection test is performed to confirm its functionality. The system utilizes HC-SR04 ultrasonic sensors and raindrop sensors to detect water levels and rain intensity at two brooks. After gathering analog data, it is sent to the ThingSpeak web application in the form of graphics using NodeMCU. The system is designed to automatically send the data to the ThingSpeak application once collected. However, the speed of data transmission is dependent on factors such as the signal from the cellular network, the time at which it was sent, and the application programming interface (API) provided by ThingSpeak.

The real data collected from devices placed in two distinct areas, specifically Rwampara and Mpazi Brooks, have been presented using the ThingSpeak application. The results and relevant information can be observed in Fig -8.

Created: 2 months ago
Last entry: less than a minute ago
Entries: 5437

**Fig -8: FLOOD MONITORING SYSTEM ANALYSIS**

The data collected from the flood detector systems at Mpazi Brook and Rwampara Brook locations are stored in a single channel named IoT flood Monitoring system. The channel has four fields, with field charts1 and 2 containing data from Mpazi location, and field charts 3 and 4 containing data from the Rwampara Brook location. This information is illustrated in Fig -8.

### V.1 Information gathered by an ultrasonic sensor positioned in Mpazi location

The information presented in Chart 1 depicts the readings obtained from the ultrasonic sensor situated in the Mpazi flood detector. The purpose of the sensor is to gauge the level of flooding in Mpazi Brook, measured in centimeters. The label X denotes the timeline, while the label Y displays the values recorded by the ultrasonic sensor. The label Y also highlights that the distance between the ultrasonic sensor and the water level of Mpazi brook changes within the range of 8cm (the value indicated by the serial monitor) to 200 centimeters. Between 06:16 and 06:20, the water level reached 8cm indicating the occurrence of a flood. The heavy rainfall depicted in Chart 2 is the likely cause of this flood. However, the water level receded after 06:20, as the difference between the sensor reading and the water level significantly increased.

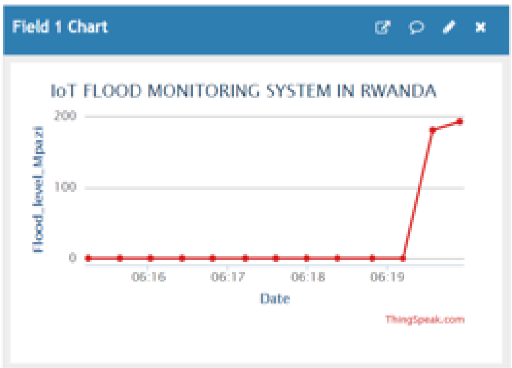


**Chart -1: The information in regards to the data collected by ultrasonic sensor at Mpazi**

### V.2 The measurements gathered by a raindrop sensor positioned in Mpazi flood detector

The data presented in field 2 Chart represents the measurements collected by a raindrop sensor located in the Mpazi flood detector. This sensor is used to measure the intensity of rain at the Mpazi brook location, with the label Y representing the sensor readings and the label X showing the corresponding time. The readings of the sensor range from 0 to 1024, and they indicate that heavy rain occurred between 06:15 and 06:17, followed by a period of moderate rain between 06:17 and 06:19. There was no rain detected after 06:19, and as a result, the ultrasonic sensor readings depicted in Chart 1 did not detect any flooding.

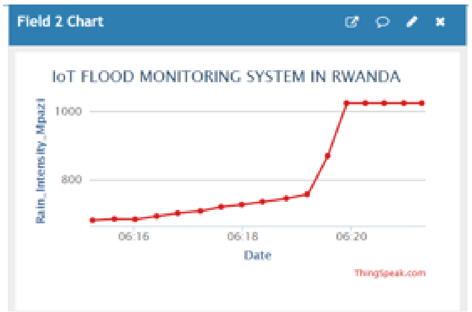


**Chart -2: The measurements gathered by a raindrop sensor located in the Mpazi flood detector**

### V.3 The measurements taken by an ultrasonic sensor installed in the Rwampara brook region

Chart -3 showcases the readings gathered by an ultrasonic sensor that has been positioned at the Rwampara flood detector site. Its key role is to measure the water level of the Rwampara brook over a period of time, which is depicted on the Label X axis. The Label Y axis denotes the readings obtained by the ultrasonic sensor. It can be observed from the recorded values that the distance between the level of the river and its indicator fluctuates between the reading displayed on the serial monitor that are 8 and 200cm. The data indicates that there was a flood in Rwampara brook before 05:56, as the water level was high enough to reach the water level indicator. However, from 06:04 onwards, there was no indication of flooding.

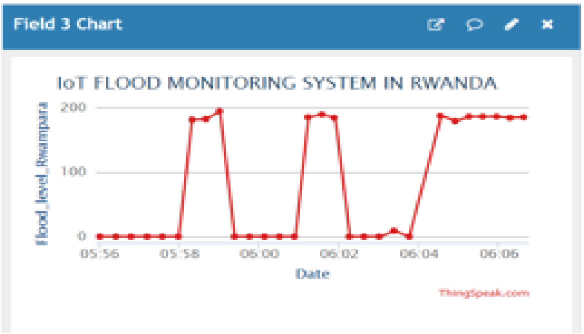


**Chart -3: The information obtained by the ultrasonic detector situated in Rwampara**

### V.4 The information obtained from the rain drop detector situated in the Rwampara brook area

Chart 4 displays the measurements obtained by a positioned in the Rwampara river, which determines the quantity of rain in the flood detector area. The time is represented on the Label X, while the raindrop sensor readings are displayed on the label Y. The range of the sensor readings is from 0 to 1024. Based on the graph, it can be inferred that there was heavy rainfall from 05:56 to 06:00, and this continued after 06:00.

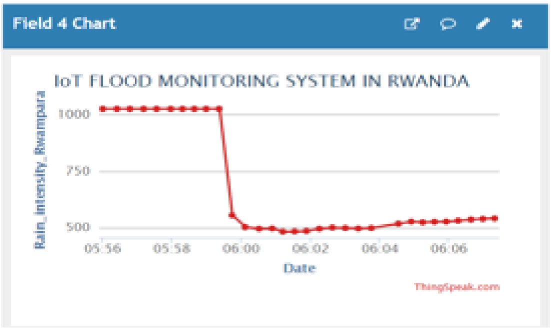


**Chart -4: Information collected by raindrop sensor located at Rwampara site**

Theinformation shown in chart 4 demonstrate that the flood level in a given location varies according to the intensity of rain. When the rain is heavy the flood level rises. Conversely, as observed in the charts for fields 3 and 4, floods can occur even in the absence of heavy rain, indicating that water from surrounding areas may be a contributing factor, as seen between 05:56 and 06:04.

To monitor floods, a web application is available through a locally hosted Xampp server. Accessing the application via any browser using the address localhost/iotfms/ yields the results as depicted in fig -9.

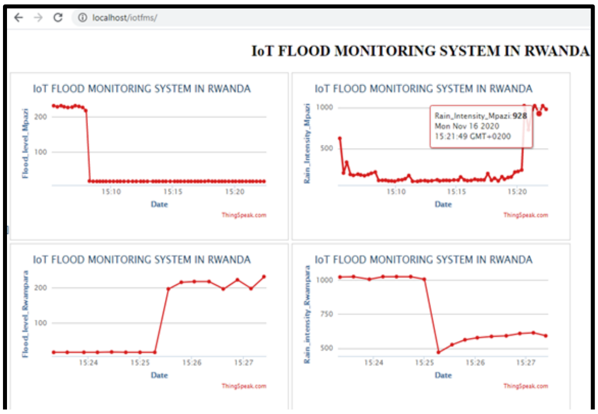


**Fig -9: Monitored data on ThingSpeak web application**

The comparison between flood levels and rainfall intensity is carried out by ThingSpeak. If there is a flood, the end-users are notified via an alert message, as shown in fig -10, notifying them to be cautious of the situation.

The event named "be_aware_flood_may_occur" occurred on the Maker Webhooks service

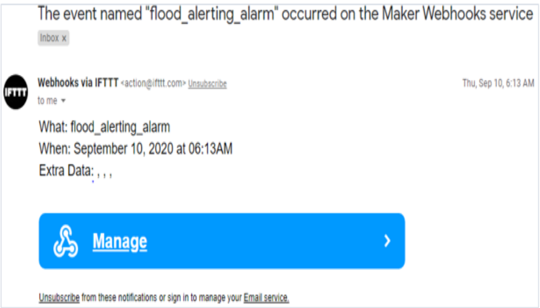
Webhooks via IFTTT <action@ifttt.com> Unsubscribe
to me
Tue, Sep 29, 6:41 PM (1 day ago)

What: be_aware_flood_may_occur
When: September 29, 2020 at 06:41PM
Extra Data: , , ,

Unsubscribe from these notifications or sign in to manage your Email service.

Fig -10: A flood may occur message

Fig -11 illustrates the flood alert message that is sent during a flood event, containing information about the alert type and the time it was sent. When the water levels are normal, a message is sent indicating the current state of the water levels.



The event named "flood_alerting_alarm" occurred on the Maker Webhooks service

Webhooks via IFTTT <action@ifttt.com> Unsubscribe
to me
Thu, Sep 10, 6:13 AM

What: flood_alerting_alarm
When: September 10, 2020 at 06:13AM
Extra Data: , , ,

Unsubscribe from these notifications or sign in to manage your Email service.

Fig _11: A notification message that alerts people about flooding through an alarm.

## VI. CONCLUSION

A flood monitoring system using Internet of Things was designed, which collects information using sensors such as a raindrop sensor, an ultrasonic sensor, and a microcontroller named NodeMCU esp8266. The data collected by microcontroller was processed and transmitted to the ThingSpeak flood monitoring system via the MQTT protocol. Through this system, flood data from two different locations could be visualized in real-time on the ThingSpeak server. Additionally, any inaccuracies detected by the flood detector system could be corrected. The prototype was successfully tested, and the flood monitoring system's technology can aid in managing floods and protecting populations, as well as helping organizations manage the environment. To enhance the system's ability to detect and provide up-to-date information on ongoing activities, it is possible to integrate a camera and drone. By alerting users, precautions can be taken to prevent flooding.

## REFERENCES

[1]. S. M. S. S. Musa, M. S. M. Noorani, F. A. Razak, M. Ismail, M. A. Alias, and S. I. Hussain, “An early warning system for flood detection using critical slowing down,” Int. J. Environ. Res. Public Health, vol. 17, no. 17, pp. 1–13, 2020.

[2]. A. Akumar, “How to Interface the GSM Module with Arduino-Send and Receive SMS,” circuitstoday.com, 2022, [Online]. Available: https://www.circuitstoday.com/interface-gsm-module-with-arduino.

[3]. S. Patil, J. Pisal, A. Patil, S. Ingavale, P. Ayarekar, and P. S. Mulla, “A Real Time Solution to Flood Monitoring System using IoT and Wireless Sensor Networks,” pp. 1807–1811, 2019.

[4]. M. R. Uma, “Smart Flood Monitoring and Alert System using IoT,” vol. 11, no. 05, pp. 357–359, 2022.

[5]. A. Radhika, G. Manochitra, K. Keerthanen, K. Mahendran, R. M. Kumar, and N. M. Sarath, “Low cost iot based early detection of flood monitoring and alert system using GPS and GSM,” AIP Conf. Proc., vol. 2527, no. April 2018, 2022.

[6]. B. M. Shankar, T. J. John, S. Karthick, B. Pattanaik, M. Pattnaik, and S. Karthikeyan, “Internet of Things based Smart Flood forecasting and Early Warning System,” Proc. - 5th Int. Conf. Comput. Methodol. Commun. ICCMC 2021, no. Iccmc, pp. 443–447, 2021.

[7]. A. Diriyana et al., “Water Level Monitoring and Flood Early Warning Using Microcontroller With IoT Based Ultrasonic Sensor,” J. Tek. Inform. C.I.T, vol. 11, no. 1, pp. 22–28, 2019, [Online].

[8]. Roderick Burnett, “Understanding How Ultrasonic Sensors Work,” 24-03-2020, 2020. https://www.maxbotix.com/articles/how-ultrasonic-sensors-work.htm (accessed Oct. 26, 2020).

[9]. R. Rittenberry, “Hands-on technology.,” Occup. Health Saf., vol. 74, no. 2, p. 24, 2005.

[10]. A. Al Dahoud and M. Fezari, “NodeMCU V3 For Fast IoT Application Development,” Notes, no. October, p. 5, 2018.

[11]. V. Viegas, J. M. D. Pereira, P. Girão, and O. Postolache, “Study of latencies in ThingSpeak,” Adv. Sci. Technol. Eng.

Syst., vol. 6, no. 1, pp. 342–348, 2021.
[12]. P. Jebane, P. Anusuya, M. Suganya, S. Meena, and M. Diana, “IoT Based Health Monitoring and Analysing System using Thingspeak Cloud & Arduino,” Int. J. Trendy Res. Eng. Technol., vol. 5, no. 4, pp. 1–6, 2021, [Online].
[13]. T. Channel, G. S. Now, and S. Up, “What is ThingSpeak,” pp. 2–4, 2016.
[14]. M. Jugran, “ThingSpeak-IFTTT-ESP32-Predictive-Machine-Monitoring,” pp. 1–11.
[15]. F. Paci, D. Bianchin, E. Quintarelli, and N. Zannone, “IFTTT Privacy Checker,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 12515 LNCS, no. December, pp. 90–107, 2020.