Internet of Things and Cyber-Physical Systems 3 (2023) 249–261

Contents lists available at ScienceDirect

Internet of Things and Cyber-Physical Systems

journal homepage: www.keaipublishing.com/en/journals/internet-of-things-and-cyber-physical-systems

Development of a smart sensing unit for LoRaWAN-based IoT flood monitoring and warning system in catchment areas

Muhammad Izzat Zakaria [a], Waheb A. Jabbar [b,*], Noorazliza Sulaiman [a]

[a] Faculty of Electrical and Electronics Engineering Technology, Universiti Malaysia Pahang, 26600, Pekan, Pahang, Malaysia

[b] School of Engineering and the Built Environment, Birmingham City University, Birmingham, B4 7XG, UK

ARTICLE INFO

Keywords:
Flood monitoring
IoT
LoRa
LoRaWAN
Smart sensor
TTN

ABSTRACT

This study introduces a novel flood monitoring and warning system (FMWS) that leverages the capabilities of long-range wide area networks (LoRaWAN) to maintain extensive network connectivity, consume minimal power, and utilize low data transmission rates. We developed a new algorithm to measure and monitor flood levels and rate changes effectively. The innovative, cost-effective, and user-friendly FMWS employs an HC-SR04 ultrasonic sensor with an Arduino microcontroller to measure flood levels and determine their status. Real-time data regarding flood levels and associated risk levels (safe, alert, cautious, or dangerous) are updated on The Things Network and integrated into TagoIO and ThingSpeak IoT platforms through a custom-built LoRaWAN gateway. The solar-powered system functions as a stand-alone beacon, notifying individuals and authorities of changing conditions. Consequently, the proposed LoRaWAN-based FMWS gathers information from catchment areas according to water level risks, triggering early flood warnings and sending them to authorities and residents via the mobile application and multiple web-based dashboards for proactive measures. The system's effectiveness and functionality are demonstrated through real-life implementation. Additionally, we evaluated the performance of the LoRa/LoRaWAN communication interface in terms of RSSI, SNR, PDR, and delay for two spreading factors (SF7 and SF12). The system's design allows for future expansion, enabling simultaneous data reporting from multiple sensor monitoring units to a server via a central gateway as a network.

1. Introduction

Floods, which are primarily caused by heavy rainfall, are catastrophic natural events that can suddenly impact vast regions, leading to significant loss of life and infrastructure damage. These disasters typically occur when intense rain raises river water levels rapidly [1–4]. Malaysia frequently experiences such events, resulting in substantial losses of valuable resources [5,6]. Implementing flood monitoring systems (FMS) can help mitigate the impact of floods on human lives, and many such systems are extensively utilized by disaster management agencies to monitor flood levels [7,8]. However, most of these systems are expensive and too complex for easy use and maintenance. Additionally, many traditional floodgates in water canals are manually operated and lack real-time water level monitoring, increasing the risk of water overflow during flash floods [5].

Flood monitoring is a promising application for the emerging Internet of Things (IoT) technology. Long-range wide area networks (LoRaWAN) represent an advanced technology for connecting various devices to the internet and serve as a significant IoT enabler, featuring low-cost devices and extended battery life [9,10]. LoRaWAN can maintain extensive network connectivity, consume minimal power, and utilize low data transmission rates [11,12]. As a low-power wireless system with long-range coverage capabilities, LoRaWAN operates on the ISM radio band, which is freely available for industrial, scientific, and medical purposes [13]. By default, LoRaWAN operates in Class A, the most common application for a star-topology system requiring one-way communication between end devices and gateways [14,15].

Despite these advances in communication technology, disaster management systems, such as real-time flood monitoring, are often lacking, especially in developing countries that frequently experience natural disasters. Flood monitoring presents a complex IoT application due to the challenges of covering large areas, managing numerous sensors, and handling the communication medium. IoT enables all smart devices to connect to cloud platforms, allowing access to information from sensors and end devices without human intervention. IoT has been employed for monitoring applications in agriculture [16], air quality [14,17], and

* Corresponding author.
E-mail address: waheb.abdullah@bcu.ac.uk (W.A. Jabbar).

https://doi.org/10.1016/j.iotcps.2023.04.005
Received 25 December 2022; Received in revised form 10 April 2023; Accepted 24 April 2023
Available online 2 May 2023

water quality management [18].

In Malaysia, floodgates in water canals are traditionally operated manually and require operators to constantly monitor water levels [19]. When left unmanaged, pump failure or neglected equipment can cause a large flood that can claim numerous human lives and thousands worth of damages. In addition, when an operator becomes unavailable or unaware of the water level, the river flow is stopped, thereby causing overflow in the canal. These scenarios can be prevented with the adoption of a monitoring system. Related to this, a telemetry system has been introduced to shorten the response time in which authorized personnel can react [20]. Several developed countries apply similar systems, but the range of information that can be received by the public remains limited. Data are usually relayed through meteorology agencies or departments and then through the response teams involved in rescuing civilians. Therefore, there is an urgent need to develop an IoT-based smart sensing unit to update flood-related data in real-time, which can be accessed by the public using their smartphones.

The main driver of the implementation of IoT in flood monitoring is the lack of a real-time monitoring system. The available data have a period gap of around 1 h. The existing FMWS focus only on the main rivers and transfers the data to the authorized via telemetry services [20]; such data must then be manually updated to the official website. In comparison, the LoRaWAN can eliminate the intermediate step by connecting the sensing device to the cloud and then alerting the targeted person either by email or SMS. The data are automatically updated on the Things Network (TTN) website and by the Tago. io application and its corresponding webpage.

In this paper, we present the design of our innovative system along with preliminary results that demonstrate its functionality and effectiveness. The proposed system is designed to enable real-time and simultaneous reporting of data from multiple sensing units distributed across catchment areas, connected to an IoT platform via a central network gateway. This system offers significant benefits, including the reduction of losses caused by floods and the development of a strategic plan to manage future flood disasters, particularly in the East Coast states of Malaysia, where floods annually result in loss of lives and infrastructure damage. The novelty of our study comes firstly from the introduced new algorithm specifically designed for measuring and monitoring flood levels and their changing rates. Secondly, the integration of multiple IoT platforms in our system with both TagoIO and ThingSpeak IoT platforms to improve data analysis and visualization capabilities using PCs/Laptops/smartphones. This integration provides a more comprehensive and versatile solution for real-time flood monitoring and response. Thirdly, our proposed FMWS is designed to be solar-powered, making it a more sustainable and self-sufficient solution for flood monitoring in remote areas with limited power infrastructure. Fourthly, the customizable warning levels that are provided by our system (safe, alert, cautious, or dangerous) suit the specific requirements of different geographical areas and flood-prone regions. This adaptability is another aspect that differentiates our work from the existing literature. Finally, we have conducted a thorough evaluation of the LoRa/LoRaWAN communication interface in terms of RSSI, SNR, PDR, and delay for two spreading factors (SF7 and SF12). This comprehensive evaluation adds value to our research by providing insights into the real-world performance of our proposed system.

In summary, this work builds on previous research efforts by offering the following key contributions.

(i) Development of a novel algorithm to measure and monitor flood levels using an ultrasonic sensor and Arduino IDE.
(ii) Design and fabrication of a cutting-edge flood monitoring sensing unit powered by LoRa wireless technology.
(iii) Implementation of the newly designed LoRaWAN gateway and sensors, along with the establishment of a LoRa network for monitoring flood levels and updating flood data on an IoT platform.
(iv) Comprehensive analysis of the collected data and real-time evaluation of risks associated with rising flood levels, enabling the provision of early flood warnings to individuals and authorities.

The rest of this paper is structured as follows: Section II discusses relevant literature, while Section III outlines the methodology and materials used in the study. Section IV describes the system architecture and implementation of the proposed solution. Section V presents the results and discussions, and Section VI offers conclusions and suggests avenues for future research.

2. Related works

In recent years, there has been a growing body of research focusing on environmental monitoring sensor networks, particularly in the area of flood detection systems [21]. These studies have aimed to address diverse challenges and requirements, considering the specific nature and location of each application. Various sensors have been employed in these networks to collect data pertinent to different aspects of flood monitoring, such as rainfall tracking or river gauging. In each case, the objective is to accurately observe and analyze key parameters to improve flood prediction, early warning, and overall disaster management. This section presents an overview of the relevant literature, highlighting the key advancements and identifying areas for further exploration.

In [1], a system known as ShonaBondhu ("golden friend" in Bengali), was developed using gradient sensing middleware to manage flash floods. It uses low-cost sensors and intelligent long-term data processing to make local decisions in its surrounding area. Souza et al. [22] developed a low-cost flood warning system that can be used both by authorities and the public based on the *E*-noe6 project to reduce damages and losses caused by flood in urban centres. The system observes the water level of either a runway or a river and triggers emergency warnings to the authorities via telephone calls and/or SMS messages. A previous study [23] used a predictive environment of sensors in a deployment to monitor the activities of the Honduras River. This approach is based on two groups of nodes for cost minimization. The first and second groups support long- and short-range radio communication, respectively. A testbed of three nodes was then implemented to investigate the river.

Xiuhong et al. [24] proposed a remote water-level monitoring system that was successfully applied in Poyanghu Lake. The system consists of a field sensor, a base station, a data centre, and a WEB-releasing module. It can realize real-time remote monitoring and provide early warning of abnormal events and protection under certain dangerous circumstances. Meanwhile, another study [2] offered useful answers for two core questions about several adaptive systems to natural disasters: "What are the key requirements to providing a reliable WSN-based system (e.g., a river monitoring system)?" and "How can an adaptable and reliable WSN-based system be developed?" The authors of that work devised a reliable WSN/IoT-based river monitoring system which was successfully deployed in the city of São Carlos, Brazil. The approach was eventually adopted for several systems.

Another study [6] proposed a flood detection and warning system called FLoWS to monitor and manage flood situations. The proposed system can also provide crucial information to the public and the authorities about affected areas through SMS and MMS using Arduino Uno combined with a GSM module. In addition, the system allows the public and local authorities to monitor live graph data of flood levels using an Android application. An IoT vision of flood monitoring problems in densely inhabited areas was discussed in Ref. [25]. The diversity of the territory (e.g., mountains and urban areas) is one of the usual problems considered in the flood monitoring process. Thus, the authors redesigned the intelligent IoT-based WSN to overcome diversity and scalability problems in order to improve reliability and efficiency. A system for the real-time monitoring of water conditions was developed in Ref. [26] to be used in monitoring floods in Nakhon Si Thammarat, Thailand. The developed system serves as an information channel for flooding between the involved authorities and experts and as a web-based information service for the public. The system consists of a sensor network, a processing/transmission unit (GPRS), and database/application server (VirtualCOM Middleware).

In [27], the mobile app called "Crowdsourcing" was developed to broadcast flood information to residents involved in preventing disasters in flood-prone areas. In this app, the sensors are integrated via SMS. The developed mobile application can also be used to exchange messages with servers, which interpret data for appropriate alert level determination prior to sending alerts to the users. The mobile app with sensors is then utilized as a flood risk management system. Islam et al. [28] incorporated a web- and belief rule-based expert system with sensors to predict floods according to rainfall and river flow on a real-time basis. The system also facilitates the monitoring of the intensifying factors of floods in a given area.

Innovative tools associated with the IoT provide a unique opportunity to anticipate hazards and monitor extreme events, such as flooding, in real-time. Their adaptability makes them ideal for gathering vast amounts of intricate data. When organized in networks, these "smart technology devices" can serve as both an early warning system and a tool for collecting flood data, ultimately reducing vulnerability. One practical example is the Calderdale Flood Sensor Network, established in the UK after the community experienced severe flooding in 2015 [29]. Leveraging LPWAN network connectivity, this IoT implementation set up multiple monitoring stations to oversee rising water levels in the area.

Another article presents a comprehensive analysis of LoRaWAN performance under various parameter settings [30]. The study aims to provide insights into the impact of different configuration choices on network performance and offer guidelines for optimizing LoRaWAN deployments. To achieve this, the researchers conducted extensive experiments with different spreading factors, coding rates, payload sizes, and transmission power levels. Their findings revealed the trade-offs between communication range, energy consumption, and network capacity associated with each parameter. By understanding these trade-offs, network operators can make informed decisions when configuring their LoRaWAN networks to achieve optimal performance in terms of reliability, coverage, and energy efficiency.

In [31], a retransmission-assisted resource management technique called R-ARM for LoRaWAN in the context of the IoT is presented. The authors address the challenges of limited resources and unpredictable network conditions in LoRaWAN networks, which can lead to performance degradation and inefficient resource utilization. R-ARM aims to improve resource management by using retransmission as a strategy to balance network load and ensure reliable communication. R-ARM employs adaptive transmission power control, spreading factor optimization, and efficient channel allocation techniques to enhance the performance of LoRaWAN networks. The proposed approach not only increases the reliability and capacity of the network but also reduces energy consumption and latency. Overall, the R-ARM technique provides a promising solution for resource management in IoT deployments using LoRaWAN.

Despite considerable research on IoT-based wireless sensor networks (WSNs) and their applications in monitoring various environmental parameters and providing updated information via the internet, the use of low-power wide-area network (LPWAN) technologies (LoRa, Sigfox, NB-IoT) remains a new and emerging field. In general, recent literature on proposed systems includes a traditional sensing unit (controller and sensor) with a communication interface, power supply, and IoT server. The on-site sensing unit is typically placed in catchment areas to monitor water levels. However, some of these systems do not utilize long-range wireless networks, relying instead on short-range communication such as ZigBee, Wi-Fi, and Bluetooth, which limits real-time data transmission over long distances.

Other systems have applied the WSN concept to flood monitoring but do not support real-time data updates via the Internet. As a result, most existing flood monitoring systems (FMS) operate locally, updating data to a local server that is only accessible to authorities and agencies. Although a few studies have applied the IoT concept to FMS, they have not addressed issues related to connectivity range, microcontroller processing capabilities, sensing accuracy, energy consumption, and practical implementation.

In our study, we focused on developing smart sensing nodes using LoRaWAN technology, which combined an ultrasonic sensor and Arduino microcontroller to ensure efficient real-time information transfer and increased network connectivity range in the targeted area. Smart and efficient sensing units would update information on the IoT cloud platform, making it accessible to the public and authorities in real-time. We fabricated a LoRaWAN gateway to collect data from multiple sensors simultaneously and send it to the TTN server, which supported a wide variety of IoT integration platforms and offered multiple options for user data accessibility. An IoT dashboard/GUI was developed to enable real-time flood data acquisition for both the public and authorities.

3. Methodology and materials

3.1. Conceptual framework

This section describes the conceptual framework and the methodology adopted in the current work, including the systematically organized stages of the research, in conjunction with the detailed implementation features of the proposed system. This section also presents the structural components of the proposed system and their integration to achieve the research objective. The flowchart in Fig. 1 illustrates the research stages of the present study and future works. The methodology adopted for this research was performed in the following phases.

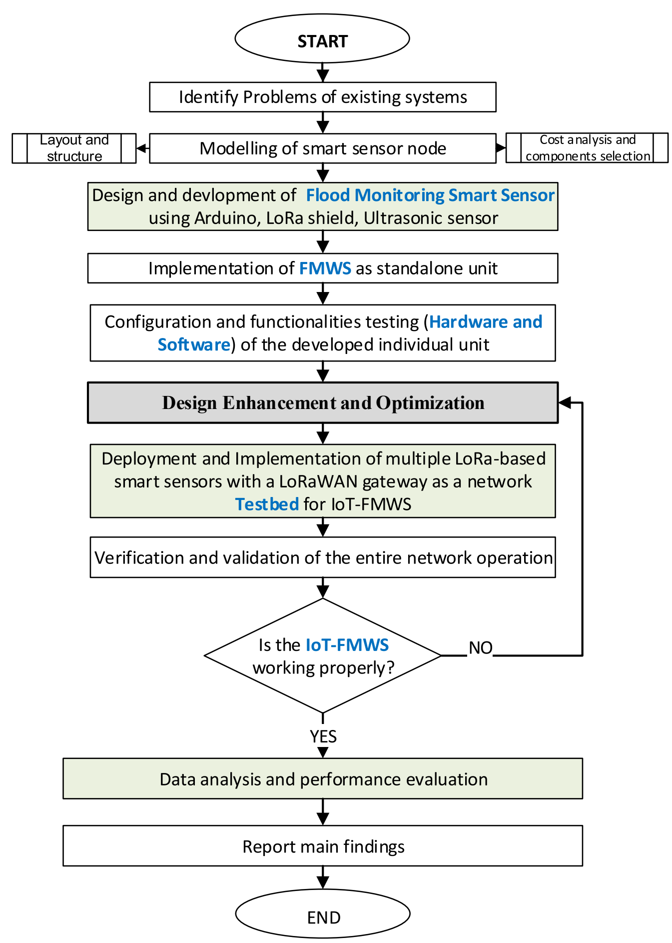


Fig. 1. Flowchart of research activities.

(i) Phase I: research was conducted to identify the problems encountered by existing flood monitoring systems and their limitations. A new design was then proposed to overcome these problems.
(ii) Phase II: smart sensors for flood level monitoring were designed and fabricated to measure water levels using Arduino, an ultrasonic sensor, and a LoRa shield.
(iii) Phase III: the developed individual unit was tested to confirm its functionalities and to evaluate the effectiveness of the proposed system.
(iv) Phase IV: the flood monitoring system units were connected to form a full wireless network.
(v) Phase V: a survey and geographical study of the considered areas were conducted in collaboration with the authorized agency.
(vi) In the final phase, the distribution of units and the implementation of the entire network testbed with testing of data updating to the IoT server were finalized.

The work reported in this paper covers the first three phases in the adopted methodology, whereas our future work will focus on the other phases to accomplish this research and implement the flood monitoring network.

Several steps were involved in the design and modelling phases, including wiring and hardware implementation, programming phase, and interfacing Arduino with LoRa and an HC-SR04 ultrasonic sensor, apart from the fabrication and configuration of the LoRaWAN gateway. For design enhancement and optimization, any problem that occurs while building the project was identified and solved during the testing phase. Some improvements were also performed to prevent error recurrence. Finally, the complete design of the system as a stand-alone unit is evaluated. In the following sections, details of the exploited components and system architecture are presented.

3.2. Main components of LoRaWAN-FMWS

3.2.1. LoRaWAN gateway

In this study, we utilized a Raspberry Pi 3.0, equipped with a LoRa Pi hat, as a LoRaWAN gateway. The Raspberry Pi is a compact, affordable, and versatile computer board designed for education, customization, and programming. It functions similarly to a standard PC, requiring a power supply, keyboard for command input, and display. This credit-card-sized computer is an ideal platform for interfacing with various devices and sensors. Raspberry Pi's primary components include central and graphics processing units, audio and communication interfaces, and RAM. Instead of a hard disk, it uses SD flash memory for storage. The device is powered through a micro USB connector and can connect to the internet via an Ethernet/LAN cable or a USB dongle for Wi-Fi and Bluetooth connectivity.

We developed our LoRaWAN gateway using the Raspberry Pi, connected to a HAT-LRGW-915. The gateway was configured for 915 MHz operation and integrated with the TTN platform to receive and display data in real-time. Fig. 2 illustrates the interior and exterior of the fabricated LoRaWAN gateway. Capable of receiving data from thousands of smart nodes simultaneously, the gateway's performance depends on the received signal strength and configured LoRa parameters. Our fabricated LoRaWAN gateway features eight channels. For the sensing node, it is essential to ensure that the LoRa module operates at a frequency similar to the one established by the LoRa Alliance.

3.2.2. Arduino Uno

Arduino is used as MCU and it provides a 3.3 V power supply to the LoRa shield and 5 V to the sensor. The usage of this MCU has several advantages. First, it has the ability to swap and change the microcontroller chip or the Microchip ATmega328 P. Second, it makes maintenance more flexible and easier to execute. The power consumption of the Arduino Uno is lower than that of Raspberry Pi. Hence, the former is suitable for off-grid implementation. Furthermore, in terms of data computation, the Arduino Uno is good enough to process and transmit data from multiple sensors simultaneously.

3.2.3. LoRa shield

For the communication system between the sensing node and the gateway, having a long-range and a low-power transceiver are some of the criteria considered for battery-powered devices. In the current study, the LoRa Shield and LoRaWAN were chosen for the physical layer and the MAC layer, respectively. The LoRaWAN focuses on the range and low power consumption in transporting data from one end to another. The LoRa Shield is based on 915 MHz—a frequency that is pre-configured to work with the Malaysia LoRa frequency spectrum. This variance is suitable for wireless network applications in different regions with varying frequency setups by Semtech. Therefore, making the right module with the right operation frequency is crucial in preventing the constant interruption and disconnection of the wireless connection. One of the features of the LoRa shield is its compatibility with 3.3 v or 5v I/O Arduino boards, such as Leonardo, Uno, Mega, and DUE. Thus, the DIOs can be directly placed and connected to the Arduino IO. Furthermore, the shield has a built-in temperature sensor and low-battery indicator. This can help developers monitor the condition of the board and prevent it from overheating if this parameter is included in the algorithm.

LoRa uses sub-GHz frequency (below 1 GHz) to communicate, which is regulated by each country's regulatory body. In Malaysia, the LoRa frequency is 920 MHz–923 MHz, also known as923-S1. For the LoRa

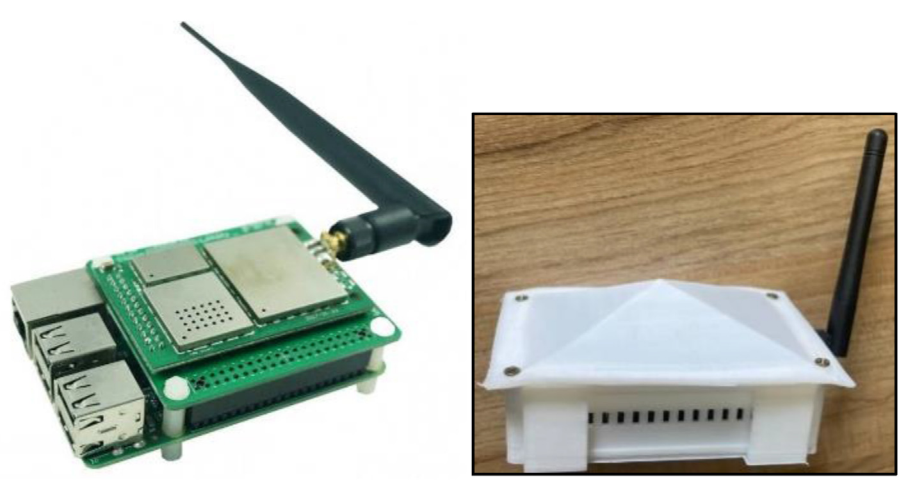


Fig. 2. Raspberry Pi 3-based LoRaWAN gateway.

antenna used in the system, we utilized a standard dipole antenna with a gain of 2 dBi, which is commonly used in LoRaWAN applications. The output power of the node was set to the maximum permissible limit according to the regional regulations, which in our case was 20 dBm (100 mW) for the 923 MHz frequency band, which is the designated frequency for LoRaWAN applications in Malaysia.

3.2.4. LoRaWAN protocol

LoRaWAN is a low-power, long-range communication protocol specifically designed for IoT applications. It enables devices to communicate with one another over a large area while consuming minimal power. LoRaWAN is based on the LoRa modulation technique, which provides an extended range and low data rates, making it an ideal choice for various IoT applications, including smart agriculture, smart cities, and industrial monitoring. LoRaWAN operates on an unlicensed frequency spectrum, and its network architecture is typically laid out in a star topology. The network comprises end devices (sensors and actuators), gateways, and a network server. The gateways relay messages between the end devices and the network server, which manages the entire network. LoRaWAN employs adaptive data rate (ADR) and various SF to optimize power consumption, network capacity, and device communication range.

LoRaWAN provides a robust and secure communication channel for IoT devices. The security of LoRaWAN is based on two main layers: network-level security and application-level security. Both layers rely on cryptographic methods to ensure the confidentiality, integrity, and authenticity of the data transmitted.

3.2.4.1. Network-level security. This layer of security protects the communication between IoT devices (end nodes) and the network server. LoRaWAN uses a unique 128-bit network session key (NwkSKey) to secure network-level communication. The NwkSKey is used to encrypt the payload and validate the integrity of the transmitted data (using Message Integrity Code or MIC). This layer of security ensures that the data is securely transmitted from the device to the network server, preventing unauthorized access and tampering.

3.2.4.2. Application-level security. This layer of security is responsible for protecting the data transmitted between IoT devices and the application server. LoRaWAN uses a unique 128-bit application session key (AppSKey) to encrypt and decrypt the payload at the application level. This ensures that only authorized applications can access and interpret the data sent by the IoT devices.

LoRaWAN also uses a secure key exchange mechanism during the device activation process. There are two main methods for device activation in LoRaWAN: (i) *Over-the-Air Activation (OTAA)*: In OTAA, devices generate a dynamic set of session keys (NwkSKey and AppSKey) during the activation process by communicating with the network server. The device and the network server both have a pre-shared application key (AppKey) that is used to derive the session keys securely. This method is considered more secure as the session keys are generated dynamically and can be updated periodically. (ii) *Activation by Personalization (ABP)*: In ABP, devices are pre-configured with static session keys (NwkSKey and AppSKey) before deployment. This method is simpler but less secure, as the keys are static and can be vulnerable if not managed properly.

In summary, LoRaWAN provides a secure communication channel for IoT devices by implementing network-level and application-level security using strong cryptographic methods. By using unique session keys and secure key exchange mechanisms, LoRaWAN ensures the confidentiality, integrity, and authenticity of the transmitted data, making it suitable for various IoT applications. In this paper, we have used Activation by Personalization (ABP) for the authentication and security of the LoRaWAN system as stated clearly in Algorithm 1. ABP offers a straightforward and efficient approach to device authentication, making it suitable for our application.

3.2.5. Ultrasonic sensor

Sensor selection is a significant part of system design because it has a huge effect on a system's performance during its entire lifetime. Hence, a system is designed and built to measure the liquid level and provide accurate data using ultrasonic sensors. The HC-SR04 is an affordable sensor, which makes it an attractive choice for large-scale deployment in flood monitoring systems. It provides non-contact measurement, which is crucial for flood monitoring systems to avoid contamination or damage due to direct contact with water. The module's measurement range is from 2 cm to 400 cm, and its ranging accuracy can reach up to 3 mm. The sensor's module includes a receiver, an ultrasonic transmitter, and a control circuit. Ease of integration is another advantage of HC-SR04 which is compatible with Arduino microcontrollers, simplifying the integration process with our system's hardware.

We did consider other sensors during the design phase, such as infrared range sensors. However, we ultimately chose the HC-SR04 due to the reasons mentioned above, as well as its widespread use and community support, which facilitates troubleshooting and future enhancements. The HC-SR04 sensor has a greater advantage over infrared range sensors because the former can detect any type of obstacle by sending ultrasound pings. In comparison, infrared sensors cannot provide the same detection for all surfaces, because information about the surface must be initially known.

3.2.6. Power supply

A solar panel and rechargeable batteries are utilized to provide a continuous power supply for the sensor nodes, thus ensuring their sustainable operation. An Arduino-based solar charger shield is attached to the node to charge the battery from the solar panel during the daytime and to turn on the node. During the nighttime, the node is powered directly by a rechargeable battery.

3.2.7. IoT platform

The TTN platform is a perfect match for implementing the IoT in a system that uses LoRaWAN as a wireless communication network. TTN is an open-source and open-network platform for building an IoT application at a low cost while providing a good security feature. Given that TTN can support the LoRa system, the developer can have real-time online monitoring features that enhance the usage of the devices. Furthermore, online information can be accessed through smart devices, such as laptops, computers, and smartphones. Web browsers are used on a range of devices, including desktops, laptops, tablets, and smartphones. The purpose of a web browser is to fetch information resources from the Web and display them on a user's device. Overall, the reason for choosing TTN is based on several factors, including its open-source nature, active community support, global coverage, ease of integration with many IoT platforms, and cost-effectiveness. These factors make it suitable for our application, which aims to be affordable, scalable, and easily deployable.

3.2.8. TagoIO IoT platform

This is an IoT integration to manage our device, store data, run analytics, and integrate services. TagoIO is an easy-to-use IoT platform and user management system because it combines several features. It creates a complete and smart platform with IoT and allows users to quickly store, visualize, and act on sensor data. TagoIO requires a sign-up to create a dashboard and use the application.

3.2.9. ThingSpeak IoT platform

ThingSpeak [32] is an open-source IoT application and API that stores and retrieves data from objects using the HTTP protocol over the Internet or via a LAN. ThingSpeak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates. The primary reason for using ThingSpeak is to showcase the versatility of the proposed flood monitoring and warning system in terms of compatibility with multiple IoT platforms. Our main objective is to be used later on for data acquisition and analysis when implementing multiple sensing nodes as a network since it is integrated with MATLAB. While TagoIO and The Things Network (TTN) have been used extensively in the paper, incorporating ThingSpeak demonstrates the system's ability to work with a variety of platforms for data visualization and analysis.

3.3. System design and fabrication

The ultrasonic sensor was chosen in this work due to its ability to measure the level by up to 4 m. This allows the device to be placed on a lamppost or similar support at a suitable location that is highest from the ground. Thus, it can detect the water level as it increases during a flood. In Malaysia, a flood in 2007 reached 2.75 m—the highest level observed since 1950 [33]. The sensor mainly measures the distance (*D(t)* at time *t*) from the object's surface to the sensor. To measure the flood level, two formulas were used. The first is given by

$$D(t) = \frac{t \times C_s}{2}, \tag{1}$$

where t is the time taken to receive the echo wave, and C_s is the speed of sound and 344 m/s. The second formula is expressed as

$$Flood\ Level = D_{max} - D(t), \tag{2}$$

where $D_{\max}$ is the max distance from the sensor to the ground and is equal to 3 m (the height of the device support), and *D(t)* is the instantaneous distance that is estimated using Eq. (1). The flood changing rate is measured in cm/min and is calculated based on the averaged value of water levels during a flood. The current measured water level is compared with the previous level to find the rate of change in 1 min. This is computed by taking the difference in the current and previous readings over a certain period.

In the design phase of the proposed unit, we used Matters Control software, which is a 3D drawing software that has low-power processor requirements. The design objectives were to make the system compactable and to include the microcontroller, LoRa shield, sensor, battery, and solar panel with all the required wiring. The system enclosure should be waterproof, small in size, and easily attached to any lamppost or similar standing support. Fig. 3 shows the 3D design of the finished product for the proposed sensing unit.

The exterior view of the fabricated smart sensing unit is shown in Fig. 4. The solar panel is placed on a suitable angle to achieve the best performance, ensure optimal battery charging, and supply the required power for the system. The electronic components are presented in the schematic diagram, and the circuit was tested on a virtual breadboard. The schematic drawing is shown in Fig. 5 with the connections between the microcontroller, the LoRa module, and the ultrasonic sensor.

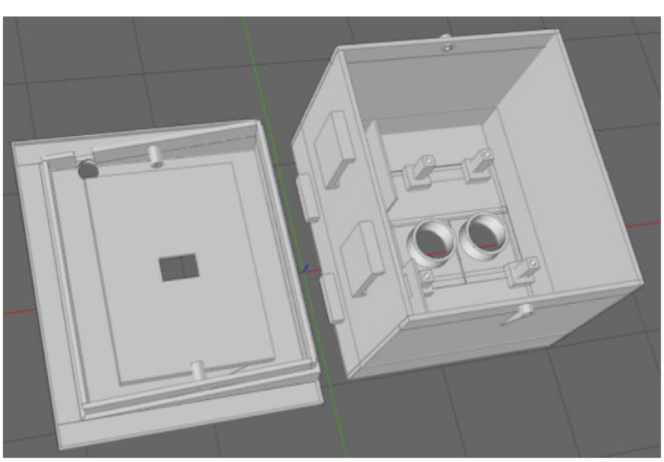


Fig. 3. 3D drawing for system casing.

3.4. Performance metrics

3.4.1. Signal-to-noise ratio

We are looking for higher values of SNR to ensure reliable communication links. SNR refers to the ratio of transmitted signal power to noise power. Ideally, SNR should be maximised to ensure that the demodulation at the receiver end is straightforward and that the signal can be decoded correctly at the receiver end. To improve the LoRa performance, the spreading factor and the forward error correction (FEC) techniques are used, thus ensuring significant SNR enhancements. Specifically, the SNR ranges between -20 and $+ 10$ dB. The received signal has less distortion if the range is around $+10$ dB. LoRa has an SNR range of between -7.5 and -20 dB [34].

3.4.2. Spreading factor

SF refers to the parameter that determines the duration of a symbol being transmitted, which directly affects the data rate and the range of communication. Higher spreading factors result in longer symbol durations, leading to lower data rates and increased communication range, while lower spreading factors result in shorter symbol durations, yielding higher data rates and reduced communication range.

3.4.3. Received signal strength indicator

RSSI is measured in dBm and reflects the power of the received signal in milliwatts. The aim is to maximize the RSSI value, as a higher value denotes a more robust received signal, leading to improved communication quality between the transmitter and receiver. In LoRaWAN systems, the acceptable RSSI value can vary based on the specific application and environmental factors. Generally, an RSSI value of -120 dBm is considered the minimum acceptable value for establishing a connection in LoRaWAN. However, it is preferable to have a higher RSSI value for enhanced communication quality and reliability. Typically, RSSI values ranging from -120 dBm to -100 dBm are considered weak signals, while values between -100 dBm and -80 dBm are seen as moderate signals. Strong signal strength is usually characterized by RSSI values between -80 dBm and -60 dBm. To ensure dependable communication and reduce packet loss, it is desirable to have an RSSI value higher than -100 dBm, preferably within the moderate to the strong signal range. It is essential to note that these ranges can vary depending on the specific LoRaWAN deployment, and acceptable RSSI values may be influenced by factors such as gateway density, environmental conditions, and the desired communication range.

3.4.4. Packet delivery ratio

PDR is an essential metric for evaluating the performance of wireless communication systems, as it indicates the ratio of successfully delivered packets to the total number of packets transmitted. We appreciate the opportunity to clarify our methodology for computing the PDR. In our study, the PDR is calculated using the following formula:

$$PDR = \frac{Number\ of\ successfully\ received\ packets}{Total\ number\ of\ transmitted\ packets} \times 100\% \tag{3}$$

To compute the PDR, we first count the total number of packets transmitted by the IoT-FMWS devices over a given time period. We then count the number of packets that are successfully received by the network server during the same time period. The PDR is obtained by dividing the number of successfully received packets by the total number of transmitted packets and multiplying the result by 100 to express it as a percentage. The PDR serves as an indicator of the reliability of the communication link between the IoT-FMWS devices and the network server. A higher PDR implies a more reliable communication channel, with a lower likelihood of data loss during transmission. In our analysis, we evaluate the PDR under different parameter settings and environmental conditions to assess the performance of our LoRaWAN-based IoT-FMWS.

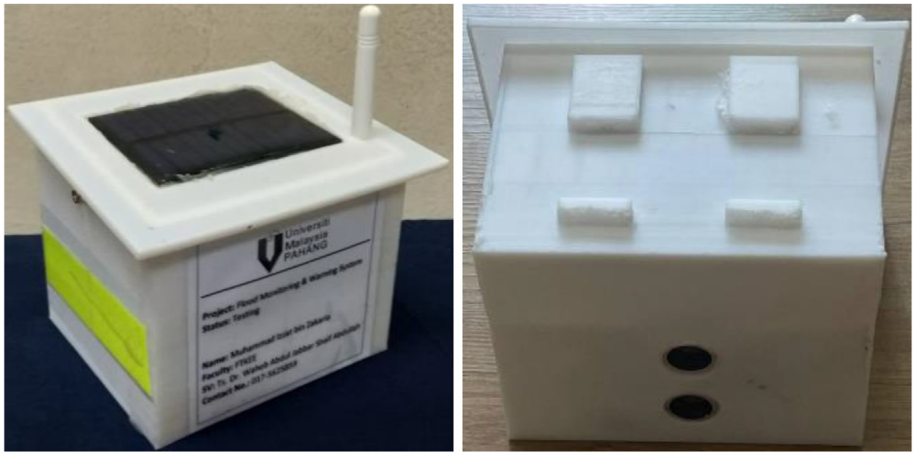


Fig. 4. LoRaWAN FMWS exterior view.

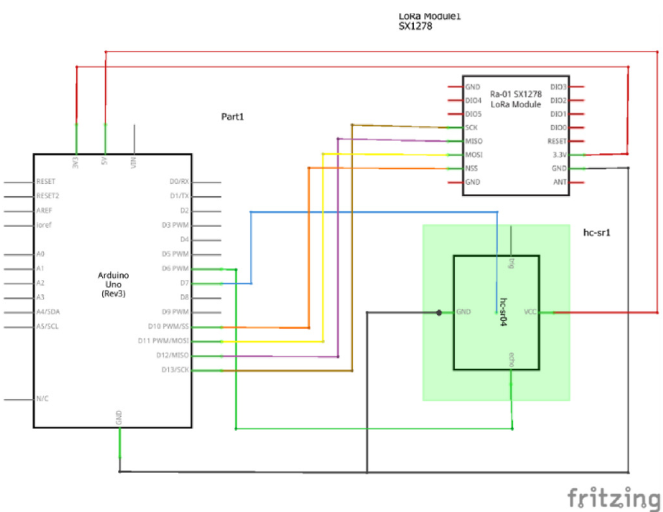


Fig. 5. The schematic diagram for the first design.

In this study, we used the confirmed mode of the LoRaWAN network. This option was made to ensure that data packets transmitted by IoT-FMWS devices were recognized by the network server and confirmed that they had been successfully received. Given the critical nature of flood monitoring and warning, reliable communication channels are crucial to ensure that important data is not lost during transmission. The confirmation mode offers additional reliability layers, including a confirmation mechanism that confirms the network server receives data packets. If the IoT-FMWS device does not receive the acknowledgement, the data package can be retransmitted to ensure its delivery. This method could lead to higher energy consumption and latency, but the priority of flood monitoring and warning systems should be given to data reliability.

3.4.5. *Time delay*

Time delay is one of the common criteria for communication analysis. The node is set to send packets every 60 s using the Arduino-based algorithm. The lower the delay, the higher the reliability of the communication link. To calculate the transmission delay in a LoRaWAN network, we need to consider several factors, including the payload size (32 bits), spreading factor (SF7 or SF12), bandwidth (125 kHz, 250 kHz, and 500 kHz), and coding rate (4/5, 4/6, 4/7, or 4/8). Accordingly, the symbol duration and the number of symbols per payload can be calculated. The time-on-air (ToA) is determined based on these parameters and it can be easily extracted from TTN and TagoIO. ToA represents the transmission delay in seconds for a single LoRaWAN packet. Keep in mind that additional delays may be introduced by factors such as network latency and processing time at the gateway and network server.

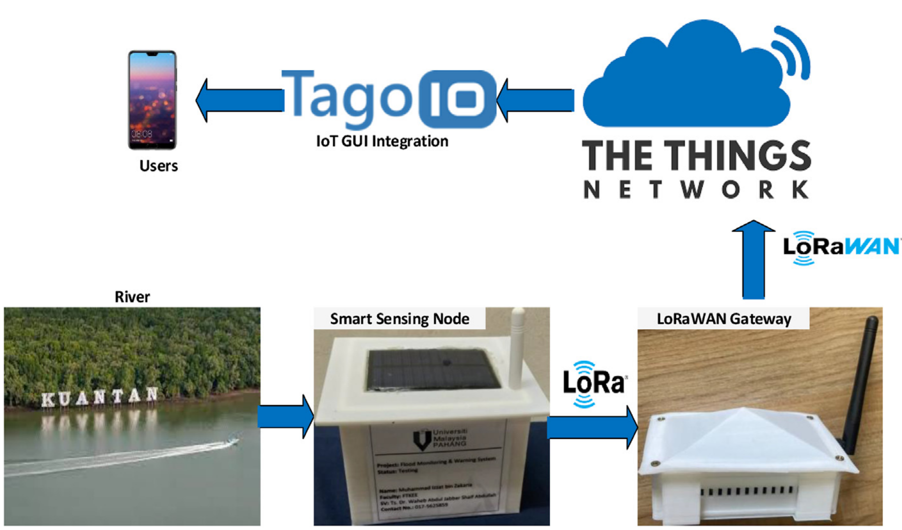


Fig. 6. Overall system architecture.

4. Proposed system architecture and implementation criteria

This study focuses on the development of a new, unified, efficient, and user-friendly FMWS based on LoRa/LoRaWAN technology for real-time data acquisition. The overall structure of the proposed FMWS system is presented in Fig. 6, which also illustrates the data flow and explains the working principle of the proposed system. In the beginning, the ultrasonic sensor, which is connected to the Arduino, measures the flood level by sending and receiving ultrasonic pulses to the water surface. The ultrasonic sensor emits the ultrasonic signal from the "Trigger" pin and receives the pulse through the "Echo" pin. The "Echo" and "Trigger" pins then send a digital reading to the microcontroller to determine the flood level according to some predefined parameters in the software program written in Arduino IDE. The values of the risk degrees are also defined in the program. These values can be altered based on the required circumstances in the actual implementation. Algorithm 1 was proposed to measure the distance between the developed system and the water surface and return the flood level.

Algorithm 1. Flood Level Monitoring

```
Assume the FMWS is placed at a 3-meter height.
Require: Low Flood Level
Ensure the water level is lower than a predefined threshold
1. Install the LoRaWAN gateway and connect to the Internet
2. Register the gateway in the TTN console (GW ID, frequency plan, router,
   and GW key)
3. Create LoRaWAN-FMWS application in the TTN console (app ID, app
   EUI, and TTN-handler)
4. Register LoRa Node under LoRaWAN-FMWS application (Dev. ID, Dev.
   EUI, app key, and app EUI)
5. Define Device activation method (ABP)
6. Get Network and app session keys and device address
7. Define Libraries for smart LoRa node and TTN
8. Define LoRa-node pin mapping ➲ for ultrasonic and LoRa connection
9. Set LoRa configuration parameters
10. Let D_MAX = 300.0 ➲ The maximum distance in cm to the water surface
11. D(t) ← Distance measured by the ultrasonic sensor at a time, t
12. F_Level(t) ← Level of flood at time, t
13. F_Rate(t) ← Flood Rate of at the time, t
14. Initialize the sensing node ➲ at t = 0
15. Acquire the state (Trigger, Echo) of HC-SR04 ➲ Start and Stop state
16. TimeElapsed = stopTime – startTime ➲ Time difference between Trigg.
   and Echo
17. D(t) = (TimeElapsed * 34300) / 2 ➲ Where 34300 is sonic speed in cm/s
18. F_level= (D_MAX - D(t) ➲ Flood level in cm
19. Estimate F_Rate(t) (increment/decrement) based on previous readings
20. for each round, do
21.   Get D(t), F_level, F_Rate(t)
22.   Multiply each reading by 100 ➲ represent each sensor by two words
23.   Split both words (16 bits) into two payloads of 8 bits
24.   Encode all payloads into ONE packet of 4 bytes
25.   Establish a connection between LoRa Node and LoRaWAN GW
26.   Update the status of sensors in the TTN server (online)
27.   Send data to LoRaWAN GW
28.   Upload data to the TTN server over the Internet
29.   Decode the received Payloads to retrieve the original sensor readings
30.   Integrate the data into the TagoIO dashboard
31.     if F_level < 50 cm then
32.       Switch ON Green LED
33.       Print ("FMWS NORMAL STATUS")
34.     else if
35.       F_level >= 50 cm && F_level < 100 cm then
36.       Switch ON YELLOW LED
37.       Print ("FMWS ALERT STATUS")
38.     else if
39.       F_level >= 100 cm && F_level < 150 cm then
40.       Switch ON ORANG LED
41.       Print ("FMWS CAUTIOUS STATUS")
42.     else
43.       F_level >= 150 cm then
44.       Switch ON RED LED
45.       Print ("FMWS DANGEROUS STATUS")
46. end for
47. User remotely monitors flood status in real-time via TagoIO web-based
   and mobile App
48. END
```

Next, we define four water levels (L) for test purposes, as follows: NORMAL ($L = < 50$ cm), ALERT (50 cm $< L = <$ 100 cm), CAUTION (100 cm $< L = <$ 150 cm), and DANGEROUS ($L >$ 150 cm). Each condition has a unique reaction upon receiving the system. The LoRa sensing node sends the collected data to the LoRaWAN gateway, which is connected to the Internet. The web-based TTN IoT server receives these data, displays them in real-time, and stores the data in the cloud. The information on the flood level is also integrated with the TagoIO server, which is monitored using the developed web-based TagoIO dashboard and as a graphical user interface (GUI) on mobile smartphones accessible to users. The TagoIO is used here because it has a user-friendly interface and can work efficiently with the TTN server so that it can be easily controlled and monitored via smartphone. The TagoIO also allows the sending of different types of alerts to users, including sound alarms, LED blinking, SMS messages, and phone calls. Through this procedure, flood level monitoring and warning can be accessed via the webserver (PC and laptop) and mobile phones.

5. Experimental results and validation

This section presents the obtained results to verify and validate the implementation of the proposed LoRaWAN-based FMWS. For validation purposes, Arduino IDE coding was developed according to the proposed algorithm in order to receive a signal from the HC-SR04 sensor. This sensor measured the water level and incremental rates in a virtual river that we fabricated using a water container and a water pump. We then designed three main scenarios to verify the functionality and performance of the proposed sensing node. Two scenarios focused on the data updating to the cloud at different water levels of the virtual river to check the system capabilities and update data in real-time to the IoT servers and mobile applications. The third scenario focused on the LoRa communication interface performance in terms of distance from the sensing node to the gateway. This was to ensure the system's ability to reliably transmit data from different sensors to the gateway. In the first scenario, the GUI based on TagoIO—in addition to the TTN—were utilized to observe the updated data about flood level, flood rate, transmission frequency, SNR, and received signal strength indicator (RSSI). The second scenario used the integration of the ThingSpeak IoT platform with TTN to visualize the collected data in real-time via the public and private ThingSpeak channels. In the third scenario, the distance between the node and the gateway was changed from 100 to 600 m. Two spreading factors (SF7 and SF12) were then tested. Ten collected readings were averaged for every 100 m to obtain a general idea about the system's behaviour before the practical implementation.

Due to the difficulties to create a real flood scenario for testing purposes, we first conducted several tests in a simulated environment to obtain the preliminary results. First, we adopted a prototype for our system to suit the virtual river. The water level was changed by utilizing a double container with valves and a water pump. The sensing node sent the gathered data to the Internet over the LoRaWAN gateway and also sent alerts, when necessary, to individuals with mobile phones via the TagoIO dashboard. In addition, the information was made available through the TTN end device to instantaneously track and analyze changes in flood level and store the data history in the cloud, which indicated the situation of the virtual river level. The data were then sent to the cloud, which recorded them continuously according to water level, as shown in Fig. 7. Usually, the received data payloads in the TTN dashboard are unreadable characters that must be decoded using the Java decoder function. These data are usually displayed in the TTN with node ID and some LoRa-related information about SNR, RSSI, frequency, and SF.

On our TagoIO channel, we added a chart and a gauge to visualize the instantaneous and recorded values of flood level (data collected) in real-time and over a defined interval, as illustrated in Fig. 8. All data were collected from the system, pushed into the TTN, and integrated with TagoIO, which allowed for various calculations, such as sum, variance, and counting, thus satisfying the security requirements. As indicated in the figures, the system shows different readings of water level as a function of time in the virtual river prototype. The normal zone is represented by green colour in Fig. 8(a), whereas the alert zone is yellow, as shown in Fig. 8(b). In contrast, the caution zone (Fig. 8(c)) and the danger zone (Fig. 8(d)) in the gauge range are represented by the colours orange and red, respectively.

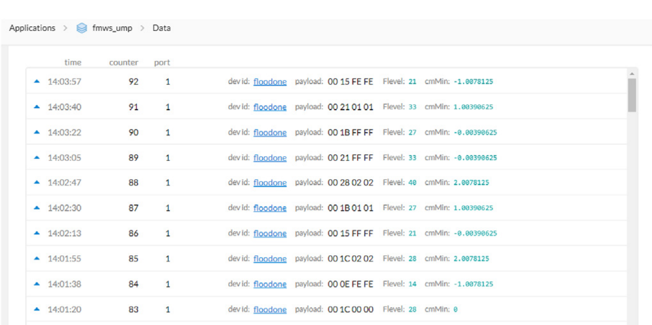
Fig. 8(a) shows the variation in water level within the considered period. During this interval, we changed the amount of water in the virtual river to test all situations (high, medium, and low levels). For each situation, we observed the received values in the field chart and gauge of flood level and then compared them with the actual level in the prototype. The values were accurate and changed in real-time. Similarly, the alert, caution, and dangerous water levels are displayed in a short interval of time, as shown in Fig. 8(b), (c), and (d), respectively. These values can be observed from a web server using PC/laptops/smartphones. The data of the particular period can be further analyzed at the main station for flood monitoring, and appropriate action can be implemented based on the flood level.

We designed a user-friendly interface using TagoIO. In addition to the flood level, several parameters are also displayed simultaneously in the dashboard, including flood rate, node transmission frequency, SNR, and RSSI. Changes in all these values can be monitored in real-time via the dashboard, indicating the transmission link's stability and reliability. According to the obtained results, we can confirm the validity of the proposed sensing node and its ability to measure the flood level accurately.

The second scenario integrates the TTN with the ThingSpeak IoT platform and presents an example to verify and validate the implementation of the proposed LoRaWAN-based FMWS. The developed smart sensing unit measures the water level in a virtual river which was fabricated using a water container and a water pump. The LoRaWAN gateway posts data to the Internet and sends alerts to individuals with mobile phones via the ThingSpeak channel. In addition, the information becomes available through a public channel on ThingSpeak to track and analyze changes in flood level instantaneously and store the data history in the cloud, subsequently indicating the situation of the virtual river level. The data are sent to ThingSpeak, which records them continuously according to water level, as shown in Fig. 9.

On our public channel on ThingSpeak, we add a chart and a gauge to visualize the instantaneous and recorded values of flood level (data collected) in real-time and over a defined interval. All data are collected from the system and pushed into ThingSpeak, which allows for various calculations such as sum, variance, and counting; it satisfies security requirements. As shown in the figures, the system shows different readings of water level in the virtual river prototype as a function of time. The dangerous zone is represented by red colour in the gauge range, the cautious zone by blue colour, and the normal zone by green colour.

Fig. 9 (a) shows the variation in water level within 30 min starting at 20:30 and ending at 21:00. During this interval, we change the amount of water in the virtual river to test all situations (high, medium, and low levels). For each situation, we observe the received values in the field chart and gauge the flood level and compare them with the actual level in the prototype. The values are accurate and change in real-time. Similarly, the medium water level is displayed in a short interval of time, as shown in Fig. 9 (b). The flood situation is also depicted in the flood level gauge, as shown in Fig. 9 (c, d, e). These values can be observed from a web server using PCs or laptops. The data of the particular period can be



Applications > fmws_ump > Data

time	counter	port	
14:03:57	92	1	dev id: floodone payload: 00 15 FE FE Flevel: 21 cmMin: -1.0078125
14:03:40	91	1	dev id: floodone payload: 00 21 01 01 Flevel: 33 cmMin: 1.00390625
14:03:22	90	1	dev id: floodone payload: 00 1B FF FF Flevel: 27 cmMin: -0.00390625
14:03:05	89	1	dev id: floodone payload: 00 21 FF FF Flevel: 33 cmMin: -0.00390625
14:02:47	88	1	dev id: floodone payload: 00 28 02 02 Flevel: 40 cmMin: 2.0078125
14:02:30	87	1	dev id: floodone payload: 00 1B 01 01 Flevel: 27 cmMin: 1.00390625
14:02:13	86	1	dev id: floodone payload: 00 15 FF FF Flevel: 21 cmMin: -0.00390625
14:01:55	85	1	dev id: floodone payload: 00 1C 02 02 Flevel: 28 cmMin: 2.0078125
14:01:38	84	1	dev id: floodone payload: 00 0E FE FE Flevel: 14 cmMin: -1.0078125
14:01:20	83	1	dev id: floodone payload: 00 1C 00 00 Flevel: 28 cmMin: 0
14:01:03	82	1	dev id: floodone payload: 00 1B 01 01 Flevel: 27 cmMin: 1.00390625

Fig. 7. Data on flood level and rate, as shown in TTN.

further analyzed using MATLAB at the main station for flood monitoring, and appropriate action can be implemented based on the flood level.

In the third scenario, we investigated the LoRa-related metrics with respect to the distance between the LoRa sensing node and the fabricated multi-channel LoRaWAN gateway for two SFs, SF7 and SF12. The primary reason for selecting only two SFs, SF12 and SF7, was to illustrate the trade-offs between communication range, data rate, and energy consumption for the IoT-FMWS in different scenarios. SF12 provides longer communication range and better signal penetration at the cost of lower data rates and higher energy consumption, while SF7 offers higher data rates and lower energy consumption at the expense of reduced communication range and signal penetration. By comparing these two extreme cases, we aimed to demonstrate the impact of different SFs on the performance of our system. This scenario aimed to recognize the LoRaWAN technology capability in providing long and reliable wireless coverage in the study area. For the sake of system validation, an area within a one km radius around a catchment area was considered in this experiment, as shown in Fig. 10. This scenario will be further extended in future experiments with a real implementation of the flood sensing node near the river. To determine the testing ground, a map within a certain radius was set beforehand. The file is saved in. kml format and combined with the Tracklia android app, which can read a. kml file and have real-time tracking on the map. This feature is of great help compared to the traditional technique of simply eyeballing the position. Using the app provides a real-time distance between the sensing node and the gateway while also helping trace the changes in the considered performance metrics, including SNR, RSSI, delay, and PDR.

Accordingly, several SNR and RSSI readings were measured in a practical scenario. The setup of the measurement campaign kept the gateway fixed and connected to the Internet outside the office at 1 m above the ground level, and the sensing node was mounted on a motorcycle that was moving away from the gateway. The sensing node was moved from the centre at the gateway through the measurement path and sent the data packets to the gateway every 100 m. The averaged values of the considered metrics were recorded at the LoRaWAN gateway side every 100 m up to a distance of 1 km.

In Malaysia, The frequency band allocated for LoRaWAN is 920 MHz–923 MHz, which falls under the AS923-S1 frequency plan. This frequency plan typically has 8 channels, and the maximum allowed transmission power is 20 dBm (100 mW). Both the sensing node and the gateway were operated at the 915–923 MHz band with two different spreading factors, SF7 and SF12, and a 20 dBm transmission power with 125 kHz bandwidth. The duty cycle regulations for LoRaWAN are defined by the Malaysian Communications and Multimedia Commission (MCMC). For the 923 MHz frequency band, which is used in our case, the maximum allowed duty cycle is 1%, as per MCMC guidelines. This means that for every 1% of time spent transmitting, the device must remain inactive for 99% of the time. Adhering to these regulations ensures that the radio spectrum is shared efficiently among multiple devices, minimizing interference and promoting optimal network performance.

Based on the default setting and features of the fabricated multi-channel LoRaWAN gateway, the communication coverage can reach up to 10 km. In the conducted tests, a distance of 1 km was considered, and this was increased gradually by 100 m from the centre of the circle at the gateway. Fig. 11 shows the received signal strength at the gateway concerning the distance. It was observed that the RSSI decreased with increasing distance regardless of the utilized SF. The developed LoRa sensing node had a high receiver sensitivity at −146 dBm. Thus, the smallest value of RSSI was −102 dBm at 600 m, and this was still much higher than the minimum value of the receiver sensitivity. Furthermore, this was the worst recorded value during this experiment for both SFs. The RSSI was influenced by the distance between the node and the gateway for both SF7 and SF12. The SF12 had a higher RSSI compared to the SF7 for the first 400 m, but both had almost the same value on the 600 m. Despite the differences, all readings were in the acceptable range. Obviously, a higher connectivity range can be reached according to the

(a) Normal level

(b) Alert level

(c) Caution level

(d) Dangerous level

Fig. 8. Water level measurements with virtual river prototype as extracted from the TagoIO Channel.

obtained results. Consequently, we proved that the selection of a LoRaWAN-based sensing node for a long-range wireless communication flood monitoring system was an appropriate and precise option for the intended system.

The performance of the LoRa node communication quality was also evaluated in terms of SNR, as shown in Fig. 12. The same scenario was used to extract the values of SNR at different distances between the node and the gateway. SNR was measured in dB, and the values were either positive or negative. An SNR with a negative value means that the noise power is higher than the signal power. This is what we can observe at the 500 and 600 m distances with SF12. With SF12, we only had a positive value for the first 400 m, while at 500 m, the value was flipped into a negative region. On the contrary, the SNR was positive with values between 0 and 9 dB for all cases of SF7. This indicates that good communication is achieved when signal power is much higher than noise power.

The main advantage of the fabricated multi-channel LoRaWAN gateway is its capability to support a large number of end nodes, thus achieving better network scalability. In addition, higher data

(a) Long time interval

(b) Short time interval

(c) Normal flood level

(d) Alert/cautious flood level

(e) Dangerous flood level

Fig. 9. Flood level measurements extracted from ThingSpeak IoT Channel.

transmission reliability and lower time delay are among the multi-channel gateway features. In this experiment, we monitored the PDR and time delay as performance metrics of the proposed system. Although only one node and one gateway were utilized, the results provide good insights into network behaviours while implementing multi-node and multi-gateway networks. In this scenario, the impact of distance variation on the PDR and delay of data transmission were further investigated by comparing the performance under the considered SFs (7 and 12). Depending on the obtained results shown in Fig. 13, it can be noted that the SF had a limited impact on the PDR at a 500 m distance, while the PDR was 100% in all cases regardless of distance or SF values. In contrast, the impact of the changing SF on the time delay can be seen in Fig. 14. The higher the SF value, the longer the delay, regardless of the distance between the gateway and the sensing node. The measured values for the considered distance can confirm the reliability of the wireless communication link between the node and the gateway. The results showed that the packet loss had a clear impact on the delay. Overall, it is preferable to use the SF7 setting to eliminate the delay and packet loss during transmission for longer distances. The highest packet loss, which was 20%, occurred at 500 m and SF12. However, for the other cases, there was no packet loss observed even at 600 m. Based on the observation, SF7 achieved a less than 2.5-s delay compared to 3.75 s, which was the highest delay with SF12 at 600 m.

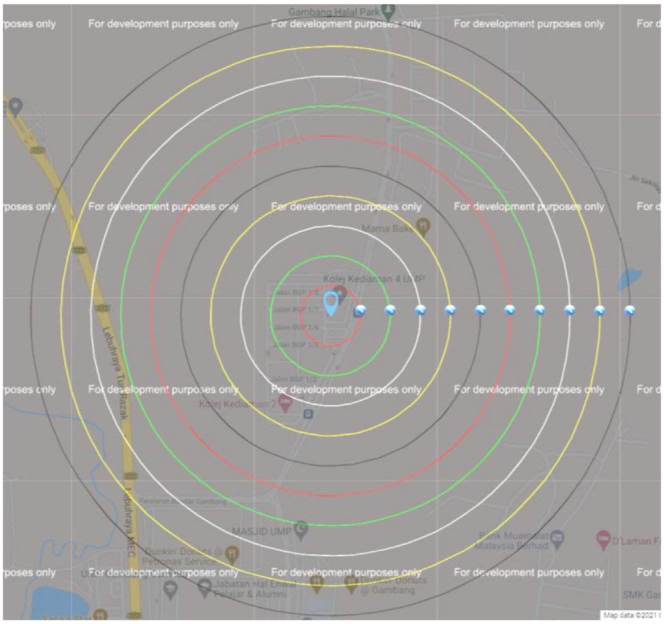


Fig. 10. One km radius from the LoRa gateway.

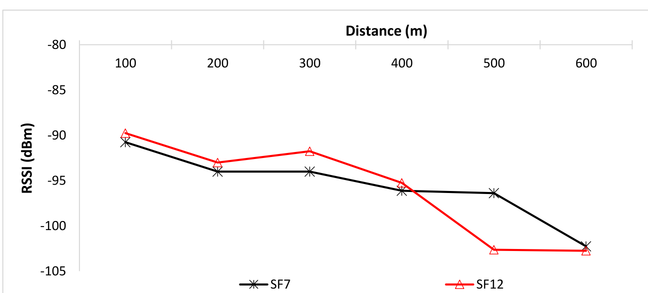


Fig. 11. RSSI vs. distances for SF7 and SF12.

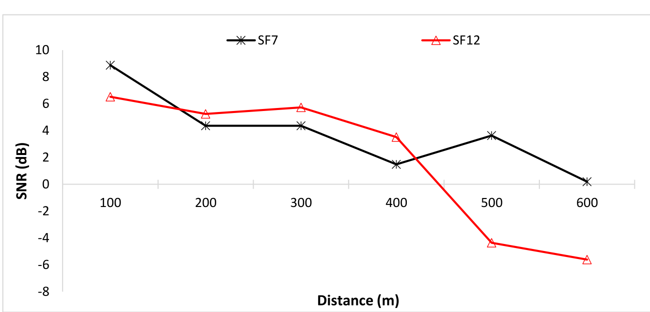


Fig. 12. SNR vs. distances for SF7 and SF12.

6. Conclusion and future works

In this study, we presented the development and implementation of a novel LoRaWAN-based smart sensing unit for a flood monitoring and warning system, specifically designed to mitigate and potentially eliminate human risks during flood seasons in Malaysia. Our system offers a cost-effective and user-friendly solution that can be utilized by both individuals and authorities for timely and accurate flood monitoring. The primary goal of this flood monitoring and warning system is to provide real-time, precise information about flood levels in catchment areas to civilians and relevant agencies. To achieve this, the system generates alerts when flood conditions reach "caution" or "dangerous" levels,

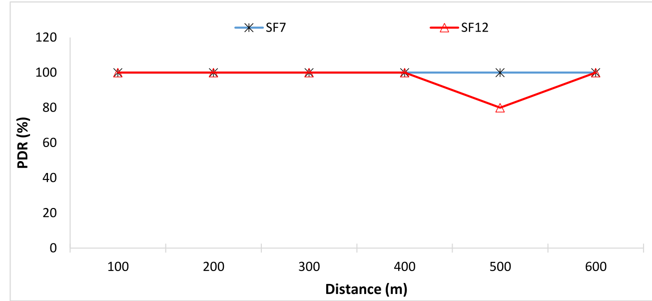


Fig. 13. PDR vs. distances for SF7 and SF12.

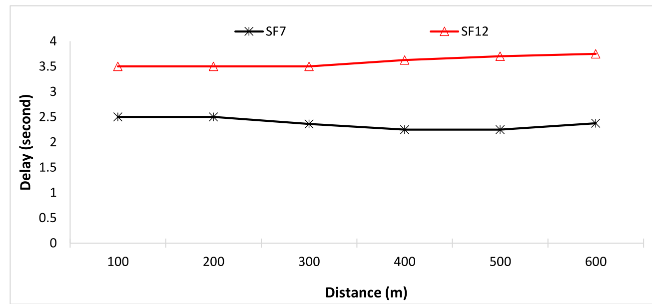


Fig. 14. Delay vs. distances for SF7 and SF12.

ensuring that appropriate actions can be taken to safeguard lives and property. Looking forward, our research will focus on expanding the capabilities of this system by enabling it to operate as a network. This will involve collecting data from multiple smart sensing units distributed across catchment areas and transmitting this information to a central station server. By aggregating and analyzing data from a broader geographic area, the system will be better equipped to support comprehensive flood management plans and help mitigate the impacts of future flood disasters. In conclusion, our LoRaWAN-based smart sensing unit for flood monitoring and warning holds significant promise for reducing human risks during flood seasons in Malaysia. By further refining its capabilities and expanding its reach through network integration, we aim to create a more robust and efficient system that can contribute to a safer and better-prepared future for communities affected by floods.

Future work could explore methods to extend the communication range, such as using more advanced network planning techniques, optimizing the placement of gateways, or employing mesh networking. As mentioned earlier, we used fixed SFs in our study, which may not be optimal for all deployment scenarios. In the future, the selection of SF that can be adapted to different deployment scenarios, taking into account factors such as network density, communication range requirements, and energy constraints. Thus, we can investigate the use of ADR to dynamically adjust SFs, data rates, and transmission power based on the individual communication requirements and channel conditions of the IoT-FMWS devices. Investigation and analysis that covers all available SFs (SF7 to SF12) in the LoRaWAN standard is also recommended. Such analysis will provide a more comprehensive understanding of the trade-offs between various SFs and their impact on the performance of the proposed system. We can explore more advanced data processing and filtering techniques to enhance the accuracy and reliability of the flood monitoring data. Techniques such as machine learning algorithms or statistical models could be employed to better analyze the data and generate more accurate flood predictions and warnings. The current implementation of IoT-FMWS relies solely on LoRaWAN for communication. Future work could explore integrating the system with other communication technologies, such as cellular IoT (e.g., NB-IoT, LTE-M) or satellite communication, to improve the system's resilience and coverage.

Funding

This study was supported by the Universiti Malaysia Pahang (www.ump.edu.my), Malaysia, under UMP Post Graduate Research Scheme PGRS200352.

Author contributions

Conceptualization, investigation, methodology, formal analysis, M.I.Z and W.A. J.; software development, W.A.J.; hardware implementation, visualization and validation, M.I.Z.; resources, W.A.J.; writing—Original draft preparation, M.I.Z. and W.A. J..; writing—Review and editing, W.A. J. and N. S.; supervision, project administration, funding acquisition, W.A.J and N. S. All authors regularly discussed the progress during the entire work. All authors have read and agreed to the published version of the manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] N. Ahmed, S. Ghosh, R.A. Hassan, S.I. Galib, A. Azad, M.A. Syrus, A gradient sensing middleware to handle flash flood, Comput. Electr. Eng. 62 (2017) 44–52.

[2] J. Ueyama, B.S. Faiçal, L.Y. Mano, G. Bayer, G. Pessin, P.H. Gomes, Enhancing reliability in Wireless Sensor Networks for adaptive river monitoring systems: reflections on their long-term deployment in Brazil, Comput. Environ. Urban Syst. 65 (2017) 41–52.

[3] G. Galuppini, et al., A unified framework for the assessment of multiple source urban flash flood hazard: the case study of Monza, Italy, Urban Water J. 17 (1) (2020) 65–77.

[4] V. Abolghasemi, M.H. Anisi, Compressive sensing for remote flood monitoring, IEEE Sensors Letters 5 (4) (2021) 1–4.

[5] Y. Hashim, A.H.M. Idzha, W.A. Jabbar, The design and implementation of a wireless flood monitoring system, J. Telecommun. Electron. Comput. Eng. 10 (3–2) (2018) 7–11.

[6] S. Marzukhi, M.A.S.M. Sidik, H.M. Nasir, Z. Zainol, M.N. Ismail, Flood detection and warning system (FLoWS), in: Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication, ACM, 2018, p. 36.

[7] A. Domeneghetti, G.J.-P. Schumann, A. Tarpanelli, Preface: Remote Sensing for Flood Mapping and Monitoring of Flood Dynamics, Multidisciplinary Digital Publishing Institute, 2019.

[8] S.I. Khan, Z. Flamig, Y. Hong, Flood monitoring system using distributed hydrologic modeling for indus river basin, in: Indus River Basin, Elsevier, 2019, pp. 335–355.

[9] W.A. Jabbar, T. Subramaniam, A.E. Ong, M.I. Shu'Ib, W. Wu, M.A. de Oliveira, LoRaWAN-based IoT system implementation for long-range outdoor air quality monitoring, Internet of Things 19 (2022), 100540.

[10] M.A. Almuhaya, W.A. Jabbar, N. Sulaiman, S. Abdulmalek, A survey on LoRaWAN technology: recent trends, opportunities, simulation tools and future directions, Electronics 11 (1) (2022) 164.

[11] S. Wang, J. Zou, Y. Chen, C. Hsu, Y. Cheng, C. Chang, Long-term performance studies of a LoRaWAN-based PM2.5 application on campus, in: 2018 IEEE 87th Vehicular Technology Conference (VTC Spring), 2018, pp. 1–5, https://doi.org/10.1109/VTCSpring.2018.8417489, 3-6 June 2018.

[12] Y. Jeon, Y. Kang, Implementation of a LoRaWAN protocol processing module on an embedded device using Secure Element, in: 2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), 2019, pp. 1–3, https://doi.org/10.1109/ITC-CSCC.2019.8793333, 23-26 June 2019.

[13] Q. Liu, Y. Mu, J. Zhao, J. Feng, B. Wang, Characterizing packet loss in city-scale LoRaWAN deployment: analysis and implications, in: 2020 IFIP Networking Conference (Networking), 2020, pp. 704–712, 22-26 June 2020.

[14] L. Maziero, et al., Monitoring of electric parameters in the federal university of santa maria using LoRaWAN technology, in: 2019 IEEE PES Innovative Smart Grid Technologies Conference - Latin America (ISGT Latin America), 2019, pp. 1–6, https://doi.org/10.1109/ISGT-LA.2019.8895425, 15-18 Sept. 2019.

[15] X. Chen, J. Wang, L. Wang, A fast session key generation Scheme for LoRaWAN, in: 2019 Australian & New Zealand Control Conference (ANZCC), 2019, pp. 63–66, https://doi.org/10.1109/ANZCC47194.2019.8945521, 27-29 Nov. 2019.

[16] D. Davcev, K. Mitreski, S. Trajkovic, V. Nikolovski, N. Koteli, IoT agriculture system based on LoRaWAN, in: 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS), 2018, pp. 1–4, https://doi.org/10.1109/WFCS.2018.8402368, 13-15 June 2018.

[17] N.H.N. Ibrahim, A.R. Ibrahim, I. Mat, A.N. Harun, G. Witjaksono, LoRaWAN in climate monitoring in advance precision agriculture system, in: 2018 International Conference on Intelligent and Advanced System (ICIAS), 2018, pp. 1–6, https://doi.org/10.1109/ICIAS.2018.8540598, 13-14 Aug. 2018.

[18] H. Fathoni, H.Y. Miao, C.Y. Chen, C.T. Yang, A monitoring system of water quality tunghai lake using LoRaWAN, in: 2020 International Conference on Pervasive Artificial Intelligence (ICPAI), 2020, pp. 281–283, https://doi.org/10.1109/ICPAI51961.2020.00060, 3-5 Dec. 2020.

[19] P.-S. Liu, N.W. Chan, The Malaysian flood hazard management program, Int. J. Emerg. Manag. 1 (2003), https://doi.org/10.1504/IJEM.2003.003303, 01/01.

[20] N. Bopi, A. Ismail, S. Ahmad Zabidi, M. Hasan, Development of a river basin monitoring system for Malaysia, International Journal of u-and e-Service, Science and Technology 9 (2016) 349–360, https://doi.org/10.14257/ijunesst.2016.9.7.35, 07/31.

[21] R.M. Ramli, W.A. Jabbar, Design and implementation of solar-powered with IoT-Enabled portable irrigation system, Internet of Things and Cyber-Physical Systems 2 (2022) 212–225, https://doi.org/10.1016/j.iotcps.2022.12.002.

[22] A.S. Souza, A.M. de Lima Curvello, F.L.d.S. de Souza, H.J. da Silva, A flood warning system to critical region, Proc. Comput. Sci. 109 (2017) 1104–1109.

[23] E.A. Basha, S. Ravela, D. Rus, Model-based monitoring for early warning flood detection, in: Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems, ACM, 2008, pp. 295–308.

[24] X. Li, X. Cheng, P. Gong, K. Yan, Design and implementation of a wireless sensor network-based remote water-level monitoring system, Sensors 11 (2) (2011) 1706–1720.

[25] M. Ancona, A. Dellacasa, G. Delzanno, A. Camera, I. Rellini, An “internet of things” vision of the flood monitoring problem, in: Proc. 5th Int. Conf. Ambient Comput., Appl., Services Technol., 2015, pp. 26–29.

[26] J. Sunkpho, C. Ootamakorn, Real-time flood monitoring and warning system, Songklanakarin J. Sci. Technol. 33 (2) (2011).

[27] M.J.C. Samonte, R.M. Rozario, N.C.B. Aranjuez, C.A. Maling, Crowdsourced mobile app for flood risk management, in: Proceedings of the 3rd International Conference on Communication and Information Processing, ACM, 2017, pp. 65–71.

[28] R. Ul Islam, K. Andersson, M.S. Hossain, A web based belief rule based expert system to predict flood, in: Proceedings of the 17th International Conference on Information Integration and Web-Based Applications & Services, ACM, 2015, p. 3.

[29] i-SCOOP, Internet of things case: flood sensor network in Calderdale. https://www.i-scoop.eu/internet-of-things-iot/internet-things-case-flood-sensor/. (Accessed 10 April 2023).

[30] D. Magrin, M. Capuzzo, A. Zanella, A thorough study of LoRaWAN performance under different parameter settings, IEEE Internet Things J. 7 (1) (2019) 116–127.

[31] A. Farhad, D.-H. Kim, J.-Y. Pyun, R-ARM: retransmission-assisted resource management in LoRaWAN for the Internet of Things, IEEE Internet Things J. 9 (10) (2021) 7347–7361.

[32] M.A.G. Maureira, D. Oldenhof, L. Teernstra, ThingSpeak–an API and web service for the internet of things, World Wide Web 25 (2011) 1–4.

[33] S.M.H. Shah, Z. Mustaffa, K.W. Yusof, Disasters worldwide and floods in the Malaysian Region: a brief review, Indian Journal of Science and Technology 10 (2) (2017).

[34] M. Swain, D. Zimon, R. Singh, M.F. Hashmi, M. Rashid, S. Hakak, LoRa-LBO: an experimental analysis of LoRa link budget optimization in custom build IoT test bed for agriculture 4.0, Agronomy 11 (5) (2021) 820.