Available online at www.HighTechJournal.org

ISSN: 2723-9535

HighTech and Innovation Journal

Vol. 7, No. 2, June, 2026

A Modular LoRaWAN IoT Flood Monitoring with Support Vector Regression-Based Prediction System

Nathan Cir Francis C. Navarro [1*], Juan Benito S. Reblando [1], John Ronald B. Pandato [1], Mikaela Riana U. Villanueva [1], Eiji Johann S. Dee [1], Alvin Y. Chua [1]

[1] *Department of Mechanical Engineering, De La Salle University, 2401 Taft Ave, Malate, Manila 1004, Philippines*.

Received 01 February 2026; Revised 20 May 2026; Accepted 27 May 2026; Published 01 June 2026

Abstract

The challenge in developing a functional flood monitoring system is addressing the tradeoffs between cost-effectiveness, ease of implementation, and overall system functionality. This paper presents a low-cost LoRaWAN-based flood monitoring and prediction system (FMPS), utilizing modular and commercially available components with an integrated Support Vector Regression (SVR) flood prediction model. Calibration of the DPS310 pressure sensor was optimized using an iterative MATLAB program, which reduced the average sensor error to 0.5772 kPa. The enclosure bias test and indoor versus outdoor deployment tests revealed a negligible pressure reading difference ranging from 0.1 to 0.2 kPa, while data transmission testing showcased a 0.04% server-side decoding error. Finally, the SVR model achieved a high correlation coefficient (R^2 = 0.9477), a low Mean Absolute Error (MAE = 0.0512), and a prediction validity of 1.5 hours in advance, showcasing its effectiveness. These results highlight the system's reliability and prediction accuracy compared to other monitoring systems, therefore providing a scalable and cost-effective solution for accelerating emergency response times during intense flooding scenarios.

Keywords: Flooding; Flood Height; Prediction; LoRaWAN; Support Vector Regression; Pressure Sensor; Internet of Things.

1. Introduction

Climate change in the Philippines has become one of the country's most economically damaging phenomena, frequently causing hydrological hazards and floods, and affecting millions across the archipelago. In the Greater Metro Manila Area (GMMA), in particular, the effects of floods are aggravated by the accumulation of garbage and poor urban planning. The region produces 9,500 tons of garbage per day, with 40% of this being uncollected and accumulating in difficult-to-reach areas [1]. This was evident during Typhoon Carina in 2024, when the Metro Manila Development Authority (MMDA) collected 387.65 tons of garbage following the typhoon [2]. Garbage build-up leads to the congestion of urban drainage systems and, in turn, higher flood levels and damage. This pattern of garbage accumulation and flooding has intensified in recent years.

As a result, the Philippine government utilizes various technologies to predict and notify the public about incoming tropical storms while also mapping vulnerable areas. Collecting accurate information on these susceptible areas requires a comprehensive, large-scale system that measures pressure, temperature, and humidity against historical data. For developing countries such as the Philippines, these systems are prohibitively expensive and difficult to maintain, but

* Corresponding author: nathan_navarro@dlsu.edu.ph

https://doi.org/10.28991/HIJ-2026-07-02-01

they remain vital for flood monitoring and response. In practice, implementing such infrastructure requires specialized personnel and proprietary parts for replacement or repair. Hence, there is a critical need for cost-effective and relatively simple systems to perform their functions with minimal operational drawbacks.

Currently, most flood monitoring systems are limited in their ability to read the water level during flood scenarios. Ultrasonic sensors [3-5] calculate flood height by transmitting waves and measuring the time it takes for the waves to return, but are prone to reading fluctuations due to environmental interference. Float switches [6] were utilized as measurement levels to determine flood height when water flows through them, but do not have the ability to perform real-time precise flood readings. Imaging systems [7] combine sensors with a camera to determine the water level based on captured images; however, they are reliant on ample lighting, lens clarity, and their ability to properly process images. In light of these limitations, recent studies utilized pressure sensor systems [8, 9] to calculate flood height using a derived inverse manometer equation, but require a hermetically sealed system and precise pressure measurements to perform flood height readings. Regardless of which sensor system is chosen, the challenge now shifts to addressing the trade-offs between system accuracy and functionality, ease of implementation, and cost-effectiveness.

Aside from the aforementioned, there are also radar-based Systems and Narrowband Internet of Things (NB-IoT) systems used in flood monitoring scenarios. The former [10-12] utilizes imaging radars by transmitting radio waves and analyzing its reflections to measure various flood parameters such as water height, rainfall accumulation, and detection of water surfaces. While these systems perform well under harsh weather conditions, they suffer from high implementation and operational costs. Conversely, NB-IoT systems [13-15] utilize a sensor network connected to a central microcontroller to aggregate and transmit telemetry data via cellular networks. While this setup is advantageous for remote area deployments due to its low latency and good signal penetration, it is reliant on the presence of a cellular network, and has a higher carrier cost in the long run as opposed to its LoRaWAN counterpart [16, 17].

Along with telemetry advancements, flood monitoring systems have integrated numerical prediction technologies to accelerate emergency response times. For example, artificial neural networks (ANNs) [18-20] establish complex flood data patterns during training, but lack adaptability to abrupt data fluctuations for real-time data. Alternatively, fuzzy logic systems (FLSs) [21] use rule-based inference framework in generating flood height predictions; however, they struggle with identifying flood parameter relationships. The Adaptive Neuro-Fuzzy Inference System (ANFIS) [19, 20, 22] leverages the strengths of ANNs and FLSs to yield a higher prediction accuracy albeit with slower solution times and higher computational resource usage. As a suitable middle-ground option, Support Vector Regression (SVR) models [19, 23-26] have lower solution times than complex deep learning models, but use more computational resources than basic linear regression models.

To address the tradeoffs on sensor reliability, communication load, and computational load and accuracy, this study presents a low-cost, modular urban Flood Monitoring and Prediction System (FMPS). To minimize environmental interference while maintaining precise water level measurements as opposed to other sensor system options, the proposed device utilizes pressure sensors to create accurate real-time flood height readings. Long Range (LoRa) technology ensures no subscription costs while graphically and numerically visualizing pertinent flood data on the PacketView website. Moreover, the device implements an SVR model to account for non-linear flood data, achieving high accuracy without computational drawbacks from more complex machine learning algorithms. Lastly, this work validates the FMPS based on its operational functionality, overall system performance and accuracy, and deployment complexity.

2. Theoretical Considerations

2.1. Conceptual Framework

Figure 1 shows the conceptual framework illustrating the workflow of the fully functional FMPS. The device consists of four key components: a rain gage for measuring precipitation rate, an AHT10 sensor for measuring relative humidity and ambient air temperature, a DPS310 sensor for obtaining internal pipe pressure, and a BMP280 sensor for reading atmospheric pressure. Telemetry from these four sensors is transmitted to the Adafruit Feather M0 microcontroller, which converts the data into a packet for transmission to the Pico Next Indoor Gateway. The gateway then sends the data to the PacketView website for end-user data visualization. Simultaneously, the SVR model is trained with historical flood data from the 2025 study by Arante et al. [27]. The real-time telemetry then functions as the testing dataset for the SVR model, generating a flood height prediction that is also displayed on the PacketView website.

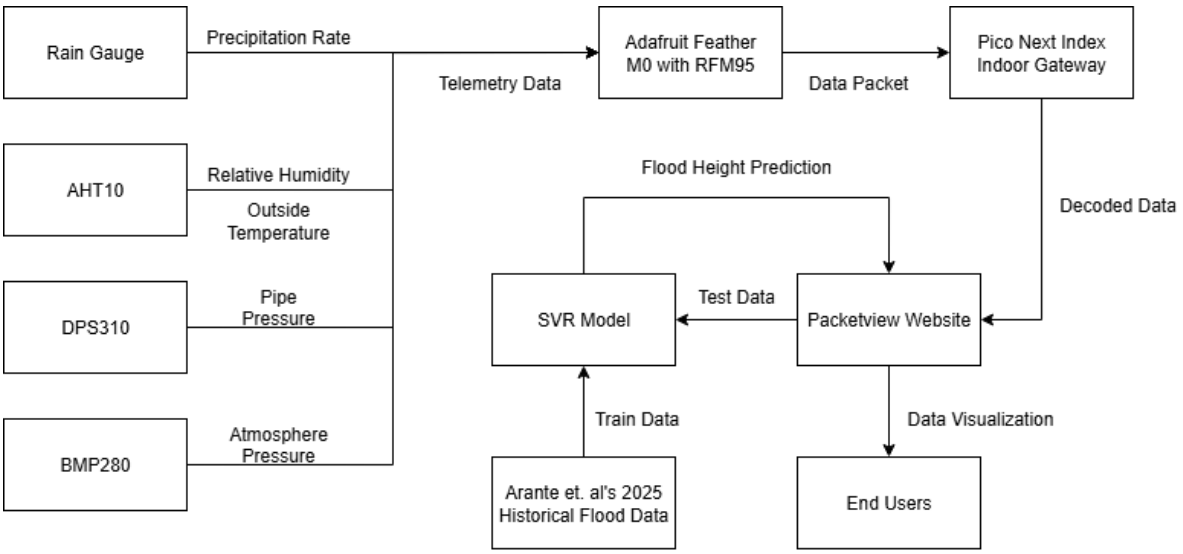


Figure 1. Conceptual framework for meteorological and hydrological data acquisition with flood monitoring and prediction

Figure 2 shows the conceptual framework for the flood height prediction model. There are two general inputs for the SVR algorithm. The first is the training dataset obtained from the study by Arante et al. [27], which enables the model to establish relationships between the input and output parameters. The second is the testing dataset, which originates from the FMPS's telemetry for generating the flood height prediction.

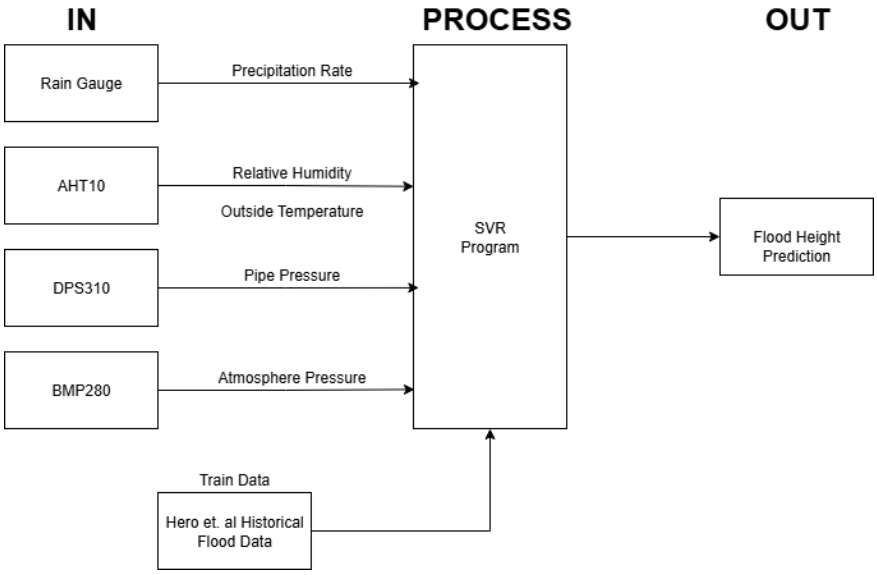


Figure 2. Flood height prediction conceptual framework

The SVR framework employs the linear, polynomial, and radial basis function (RBF) kernels as regression functions to predict the resulting flood height given the input parameters: precipitation, ambient temperature, internal pressure, ambient atmospheric pressure, and ambient relative humidity. It uses the concept of support vectors, which are the data points closest to the hyperplane, to define the model. The linear kernel in Equation 1 is typically used for parameters that exhibit a linear relationship. Due to the proportionality of the data points, this is the fastest kernel to train. Meanwhile, the polynomial kernel in Equation 2 captures nonlinear polynomial relationships by mapping the datapoints into a feature-space of a higher degree. This allows more complex functions to fit in the model; however, it also increases the risk of overfitting and computational time. Lastly, the RBF kernel in Equation 3 is the most flexible kernel, allowing complex, nonlinear relationships between input parameters to be processed by the algorithm. It uses the Gaussian function to map the relationship between parameters. The RBF kernel prevents overfitting and has a lower computation time, since it generalizes better than the polynomial kernel. The equations for the three kernels are shown below [28]:

$$K(i,j) = x_i \cdot x_j \tag{1}$$

$$K(i,j) = \left(\gamma x_i \cdot x_j + c\right)^d \tag{2}$$

$$K(i,j) = e^{(-\gamma|x_i - x_j|^2)} \tag{3}$$

where, K(i, j) represents the kernel function, x_i and x_j are the data input vectors of the i-th and j-th data points, c is a free constant, d is the polynomial degree, and γ is the scaling hyperparameter. Other parameters, including the regularization parameter C, and the error threshold parameter ε were accounted for in SVR model tuning discussed in the following sections.

Figure 3 shows the input parameters used for the WRF-Hydro model, typically employed for hydrometeorological forecasting and flood prediction. This model requires atmospheric forcing inputs, including near-surface air temperature, specific humidity, surface pressure, shortwave radiation, longwave radiation, surface wind, and precipitation. For this study, the WRF-Hydro model served as a basis for selecting the input parameters for the proposed flood height prediction model: temperature, pressure, humidity, and precipitation. Meanwhile, Support Vector Regression (SVR) was used as the primary machine learning algorithm model. To evaluate and assess the performance of the trained model, the following metrics were calculated: the root mean square error (RMSE) and the coefficient of determination (R^2).

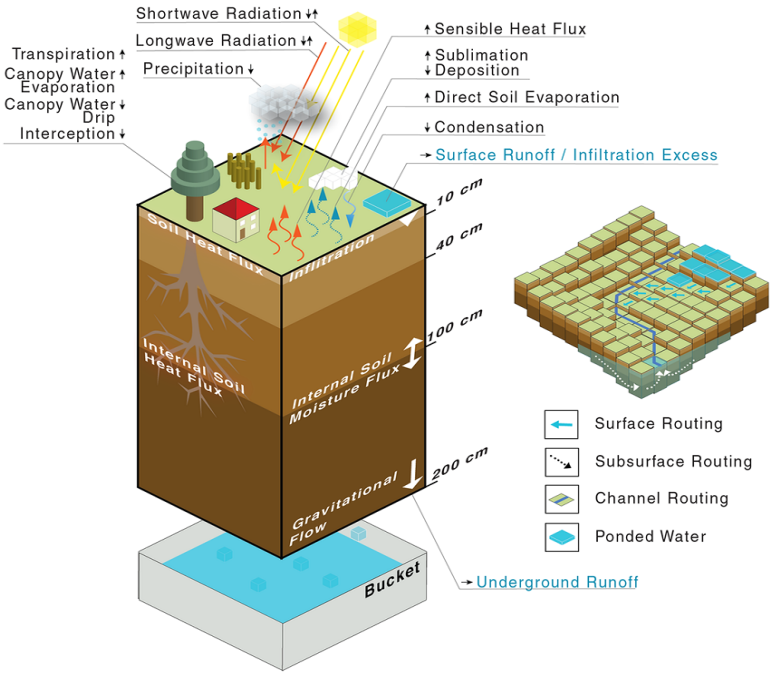


Figure 3. Weather Research and Forecasting (WRF) - Hydro Model [29]

2.2. Theoretical Framework

Figure 4 shows a visualization of the inverse barometer effect which the FMPS is based upon [8, 9]. The open-ended part of the tube that is submerged in floodwater increases the pressure inside the tube. Since the atmospheric and inside pipe pressure can be obtained through climate data and pressure sensor data, respectively, the height of the air inside the column reflects the flood height, which can be calculated using Equation 4:

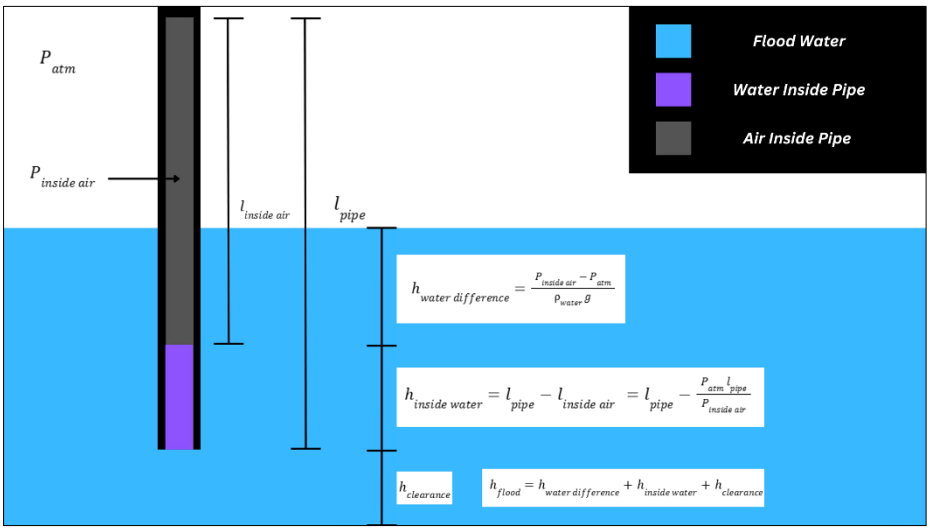


Figure 4. U-tube manometer concept of the flood monitoring system [9]

$$h_{water\ difference} = \frac{P_{inside\ air} - P_{atm}}{\rho_{water}\, g} \quad (4)$$

where, $h_{water\ difference}$ is the flood height difference between the actual flood level outside the tube and the water level inside the tube, $P_{inside\ air}$ is the pressure caused by the air inside the pipe which is obtained from the pressure sensor, P_{atm} is the atmospheric pressure which can be obtained from environmental data, ρ_{water} is the density of the water that can be calculated experimentally, and g is the acceleration due to gravity which is assumed to be 9.8066 m/s^2.

However, this does not account for the water height that originates from the inflow of floodwater into the pipe. This can be calculated by relating the two pipes and atmospheric pressures through Boyle's Law, assuming that the dimensions of the pipe remain the same and that the pressures are different. The derived equation for the water inside the pipe is shown in Equation 5:

$$h_{inside\ water} = l_{pipe} - l_{inside\ air} = l_{pipe} - \frac{P_{atm}\, l_{pipe}}{P_{inside\ air}} \quad (5)$$

where, l_{pipe} is the total length of the pipe, $l_{inside\ air}$ is the pipe length corresponding to the volume occupied by air inside the pipe, and $h_{inside\ water}$ is the height of water inside the pipe. Finally, the total flood height can be calculated by adding the previous quantities plus the clearance of the pipe from the ground ($h_{clearance}$), which is set during installation, and can be expressed as follows:

$$h_{flood} = h_{water\ difference} + h_{inside\ water} + h_{clearance} \quad (6)$$

Alongside the flood level calculations is the flood prediction algorithm, which will involve the use of several parameters in order to predict the flood level in any particular area.

The following equations are used for fitting the SVR model:

$$MSE = \frac{1}{n}\sum_{k=1}^{n}\left(y_{pred} - y_{obs}\right)^2 \quad (7)$$

where n is the total number of data points or observations, y_{pred} is the predicted value for the determined parameter at a certain point, y_{obs} is the observed value of the determined parameter at a certain point, and the error is obtained from the difference between the observed and predicted values. This is the primary model used to generate the prediction model. If the square root of the MSE is obtained, then the root mean square error (RMSE) is obtained as shown below [19]:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{k=1}^{n}\left(y_{pred} - y_{obs}\right)^2} \quad (8)$$

Another parameter for fitting the SVR model is the mean absolute error (MAE), which is given as [30]:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_{obs} + Y_{pred}\right| \quad (9)$$

After training and testing the data, the performance of the model is evaluated via the correlation coefficient of determination R^2, which measures how well the outcomes were replicated by the model [30].

$$R^2 = \frac{\sum_{i=1}^{n}\left(y_{pred} - \underline{y_{pred}}\right)\left(y_{obs} - \underline{y_{obs}}\right)}{\sqrt{\sum_{i=1}^{n}\left(y_{pred} - \underline{y_{pred}}\right)^2\left(y_{obs} - \underline{y_{obs}}\right)^2}} \quad (10)$$

where, $\overline{y}_{pred}$ is the mean of the predicted values, and $\overline{y}_{obs}$ is the mean of the observed values.

Finally, the Long Range (LoRa) network is a communication technology that is used to transmit information over long distances. With its select frequency range and noise reduction technology, it is considered a reliable technology for transmitting information securely and efficiently. This is different from Long Range Wide Area Network (LoRaWAN), which refers to the network of devices utilizing LoRa to communicate with each other. Figure 5 shows a summary of the LoRaWAN system.

In summary, LoRaWAN features end nodes, which are devices that gather and transmit information. These devices then transmit the data to a gateway, which is connected to the LoRaWAN servers. From the servers, the data is compiled and sent to application servers, which manipulate the information and output it to its recipients. This technology will be utilized by the group in their study since it has a secure communication line and a wider range than other IoT communication protocols, thus ensuring that the correct data is received from the probe to the network and application servers.

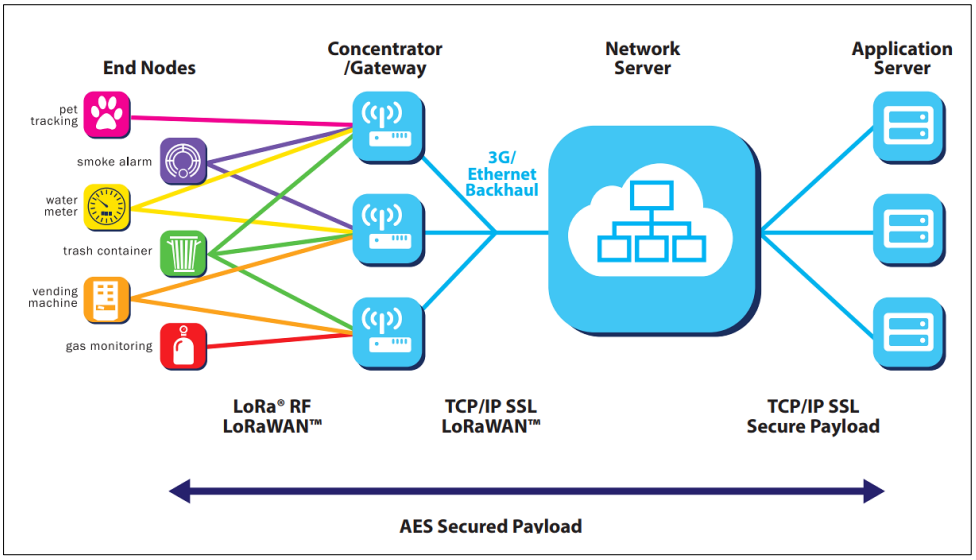


Figure 5. LoRaWAN network structure [31]

3. Methodology

This section covers the design and fabrication procedure for the creation of the FMPS, along with the procedure for testing the accuracy and effectiveness of the FMPS.

Figure 6 illustrates the design methodology flowchart for this study. The design of the FMPS was split into three phases: the physical flood probe development, LoRaWAN data transmission configuration, and the implementation of the SVR-based prediction model. For the first phase, the physical parts of the probe were first constructed, followed by the sensor selection and PCB design. For the LoRaWAN data acquisition system, firmware for data transmission and receiving was developed, and the PacketView website user interface (UI) was designed to visualize the transmitted data. Finally, for the SVR prediction model, the SVR algorithm was coded, and training data was acquired from the study by Arante et al. [27] to train the model. Once all steps were completed, the FMPS was assembled, and the sensors were calibrated prior to performance evaluation.

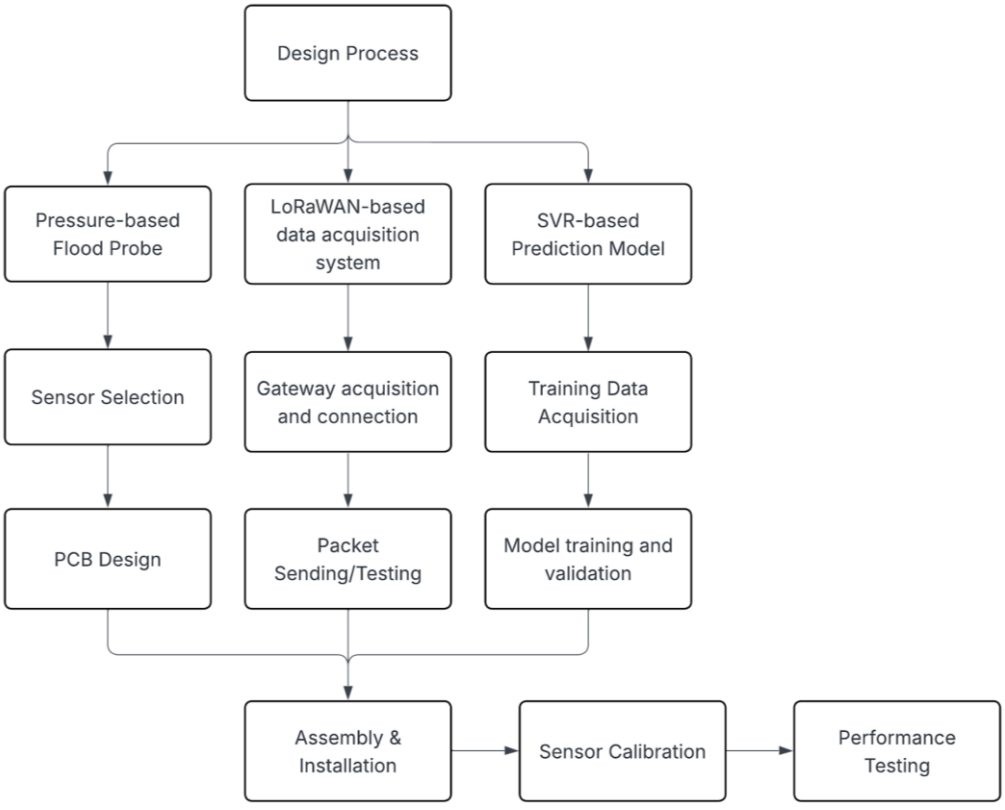
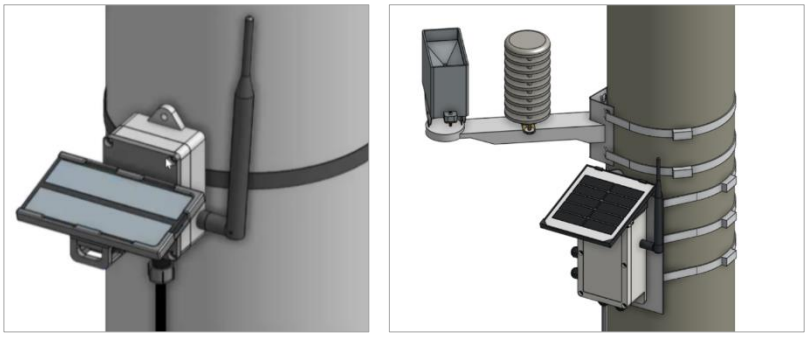


Figure 6. Design methodology flowchart

3.1. Flood Monitoring System Design

As shown in Figures 7a and 7b, the design of the final probe was derived from the 2024 design by Te et al. [9]. One key improvement over the previous iteration is the elimination of the use of 3D-printed parts for securing the electronics inside the enclosure. Instead, PG7 gland nuts prevent water ingress at the cable inlets. Moreover, the work of Te et al. [9] was utilized to design an airtight endcap, ensuring that the internal pressure readings of the DPS310 sensor remain highly accurate. The system's design was evaluated based on three key metrics: overall cost, modularity, and system performance. The developed FMPS cost Php 9,665.02 to fabricate, making it significantly more cost-friendly than existing weather stations and viable for mass deployment by local government units (LGUs). In terms of modularity, the device utilizes commercially available components that can be easily substituted, therefore simplifying maintenance and repairs. Finally, the device retains its accuracy despite its modularity and lower manufacturing cost, as discussed in the following section.



Figures 7. Final probe designs for Te et al. (2024) [9] (left) and the current system design in 2025 (right)

3.2. Printed Circuit Board (PCB) Design

Figure 8 shows the PCB design for the FMPS electronic enclosure. The board was designed using KiCAD, which incorporates the Institute for Interconnecting and Packaging Electronic Circuits (IPC) 2221 and National Electrical Manufacturers Association (NEMA) standards for PCB design. To ensure manufacturing quality, PCB fabrication was outsourced to JLCPCB by uploading the necessary Gerber files to their website and specifying the board size limitations.

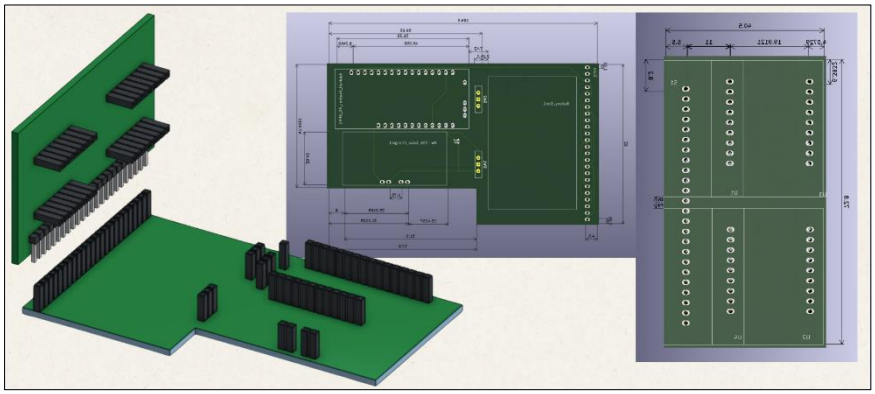


Figure 8. FMPS electronics box PCB design

The main PCB containing the battery, HW736 solar charger module, and Adafruit Feather M0 has a width of 62mm and a length of 104.5mm with a thickness of 1.6mm. Meanwhile, the sensor PCB containing the four RJ45 Adapter Connector Modules has a width of 40.5mm and a length of 72.8mm, with a thickness of 1.6mm. For both boards, the indicators for each specific function and names of the components were silkscreened in white text to provide better contrast to the green PCB. Before delivery, the company provided a Flying Probe Test to ensure that all components would have electrical contact as intended. The board material is a FR4-Standard TG 135-140 following the IPC Class 2 Standard with Hot Air Solder Leveling (HASL) with lead surface finish. The vias were tented to protect against electrical shorts.

3.3. LoRaWAN Uplink and Visualization

Figure 9 shows the LoRAWAN gateway setup, which comprises a Wi-Fi router and the Pico Next Indoor Gateway. When the gateway is powered on and connected to the router via an Ethernet cable, it automatically finds and connects to the nearest LoRaWAN network server. Once this connection is established, the FMPS scans for the gateway to send

its sensor data. The gateway then forwards the data to the Actility LoRaWAN server, which routes it to the custom PacketView dashboard in real-time. The web interface was developed using PacketView's built-in dashboard editor with the following design considerations: data completeness, ease of use, and visualization effectiveness. The actual user interface (UI) is presented and analyzed in the following sections of this article.

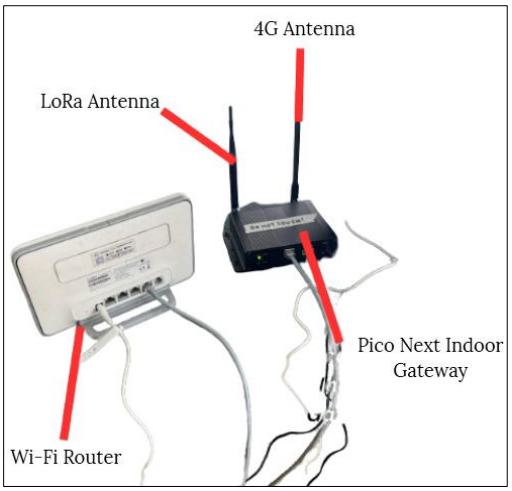


Figure 9. Pico next indoor gateway setup

3.4. SVR Program Creation

The SVR algorithm was implemented within a Jupyter Notebook environment using Python. The baseline code for the program was initially generated via Anaconda's built-in AI assistant and subsequently evaluated and modified by the researchers to fit the constraints of this flood prediction use case. The notebook is divided into several parts to seamlessly track the program execution and troubleshoot as necessary. The first stage involves importing the necessary data science libraries and preprocessing the historical training data. Then, the program loads and pre-processes the real-time telemetry data gathered by the FMPS.

Once the data is prepared, the SVR hyperparameters are defined through a parameter grid containing the gamma (γ), epsilon (ε), and regularization (C) values based on the best machine learning practices according to JupyterLab's Anaconda AI assistant, as seen in Table 1. A GridSearchCV function then iterates model configurations utilizing the stored SVR parameters to determine the optimal parameters for each kernel type. After optimizing the hyperparameters, the SVR model generates a flood height prediction using the testing dataset, and the model is evaluated based on the metrics discussed in the following sections. Figure 10 also summarizes the SVR program flowchart.

Table 1. Selected SVR parameters for hypertuning

Kernel	Regularization (C)	Gamma (γ)	Epsilon (ε)
Linear	0.1, 1, 10, 100	N/A	0.01, 0.1, 0.2
RBF	0.1, 1, 10, 100	0.001, 0.01, 0.1, 1	0.01, 0.1, 0.2

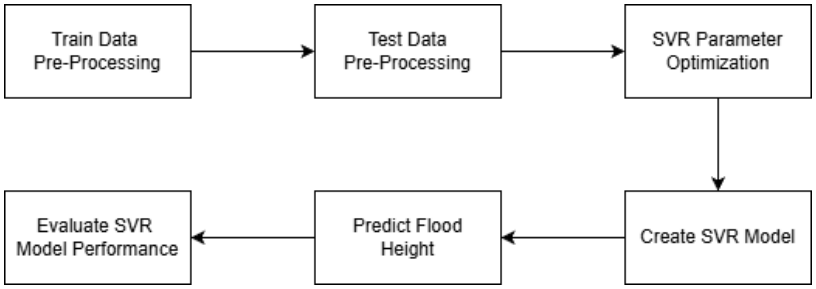


Figure 10. SVR program flowchart

3.5. Battery Testing

Table 2 shows estimated daily power consumption figures for all components. By summing up the daily power consumption of all sensors, the theoretical yield was determined to be 0.441672 Wh. However, to consider discrepancies between the actual power draw of the sensors and their datasheet specifications, the daily power consumption was rounded up to 0.5 Wh. It should be noted that the rain gauge sensor had a negligible power draw and contributed minimally to the total power consumption. Additionally, a factor of safety of 1.85 was applied to the battery's capacity,

reducing the nominal 3.7 Wh capacity to an effective 2 Wh. By dividing the allowable battery capacity by the power consumption, the system is projected to operate for four (4) days without sunlight. This duration accounts for the lack of charging from the solar panel during cloudy weather conditions. To validate the actual capacity, the device was allowed to run a full discharge cycle, and the current drain of the battery was measured using the multimeter.

Table 2. Power draw calculations of the FMPS electronics

Device	Voltage Input	Current Draw	Highest Power Consumption	Use Duration	Daily Power Consumption
Feather M0	4.2 V	2.7 mA	11.55 mW	24 hrs.	0.2772 Wh
AHT10	-	-	0.5 mW	24 hrs.	0.0168 Wh
BMP280	4.2 V	1120 μA	4.704 mW	24 hrs.	0.112896 Wh
DPS310	4.2 V	345 μA	1.449 mW	24 hrs.	0.034776 Wh

3.6. LoRaWAN Location Testing

Assuming that the device was installed at a fixed location, the gateway was moved to various locations to determine the effective range of the LoRa data transmission. Five separate test sites were chosen on the De La Salle University Manila campus: Yuchengco Hall, St. Joseph Hall, St. Miguel Building, Velasco Building, and Henry Sy Sr. Building grounds. The uplink of the device was then checked via the PacketView website and the Actility Network Server (LNS) wireless logger to determine if the device was able to successfully transmit and receive packets during the deployment.

3.7. Device Assembly and Setup

Figure 11 shows the physical setup of the installed device on the De La Salle University Manila campus, while Figure 12 shows the electronic assembly of the device. The assembly process began by verifying that all internal circuit connections were secured and that the sensors remained functional. Then, the PVC endcap, housing, and pole arm mount were secured onto the wall by tightening the worm gear clamps. Finally, all screws and bolts were tightened before testing commenced.

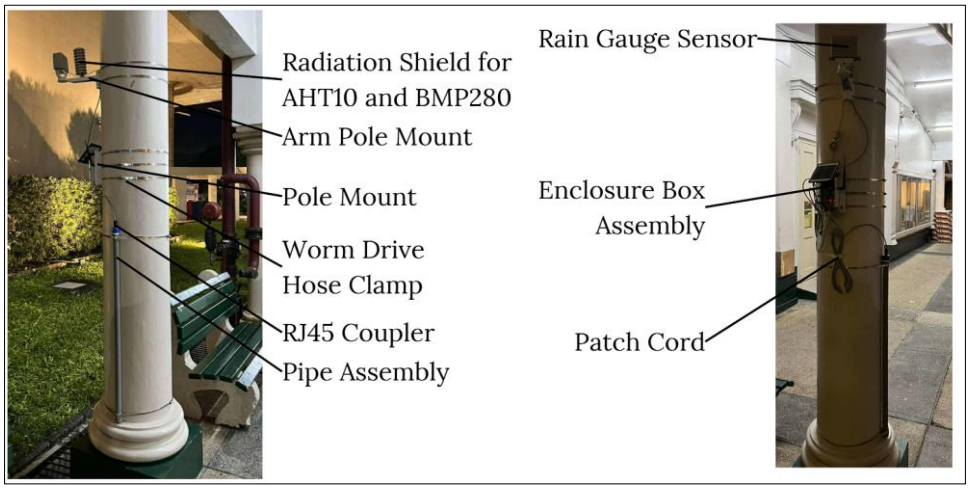


Figure 11. Device installation and assembly

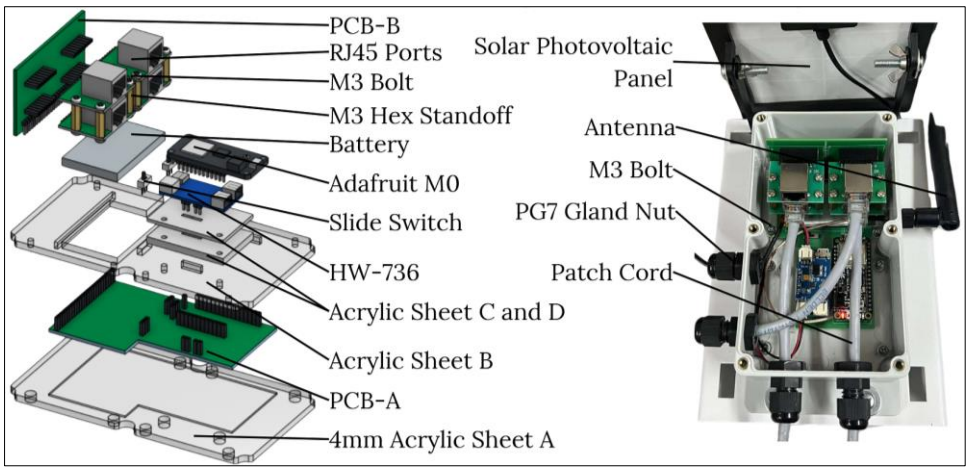


Figure 12. Electronics box assembly

3.8. Performance Testing Procedures

3.8.1. Sealing Test

Three tests were conducted to evaluate the hermetic sealing and Teflon tape applied to the PVC male adapter, as seen in Figure 13. The first test involved submerging the probe in a one meter water column to determine the level of water infiltration over time. The second test was performed to assess the hermetic integrity of the endcap. While the device was functioning, water was poured into a graduated cylinder in increments of 2 cm every 8 minutes. The pressure data from the DPS310 sensor were then extracted to determine if the pressure readings were maintained at each increment. Lastly, the third test setup utilized a modified endcap with a tire valve installed at the other end of the PVC pipe. In this setup, the internal pressure was increased, and a bubble test was conducted to detect any pressure leakages from the sealed endcap.

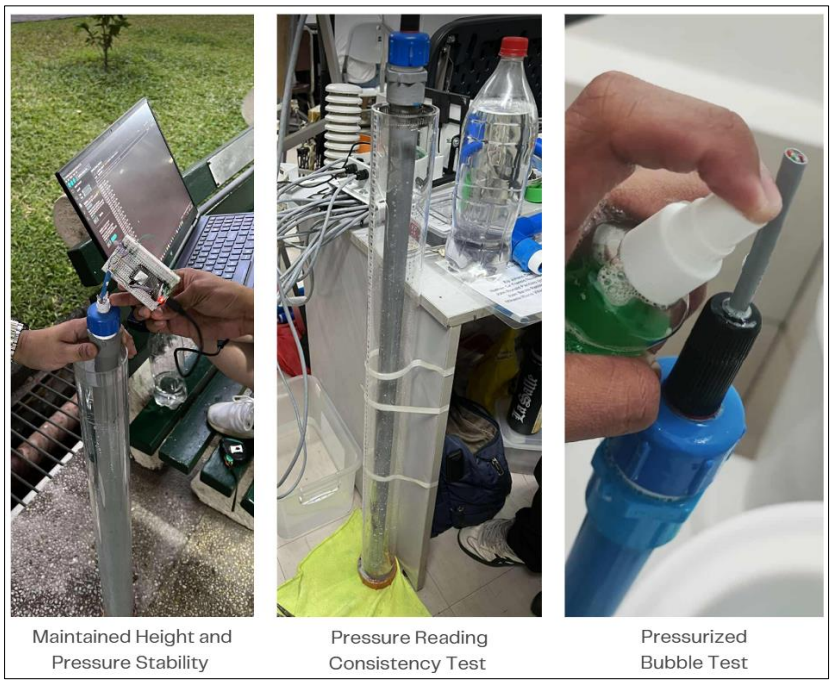


Figure 13. Sealing test setups

3.8.2. Sensor Calibration

The FMPS utilized the BMP280 and DPS310 pressure sensors for measuring the ambient atmospheric and internal pipe pressure, respectively. According to their datasheets, both sensors exhibit a similar accuracy of ±1 hPa. However, the BMP280 is less susceptible to sensor drift, making it suitable as the reference for DPS310 calibration. Consequently, the calibration process involved adjusting the DPS310 firmware with an optimal scaling factor to match the BMP280's baseline readings. The factor was determined through an experimental setup, in which water was poured into a graduated cylinder. The pipe housing containing the DPS310 was fixed inside the cylinder to prevent reading fluctuations. The pressure readings for increasing water heights were obtained while the actual and calculated water heights were recorded. The percentage error between the actual and calculated readings were then calculated, and a MATLAB script was developed to evaluate multipliers ranging from 0.95 to 1.05 in increments of 0.001. For every multiplier, the relative errors were calculated to generate a dataset relating each multiplier to the set of errors. The script also calculated the average percentage error for each multiplier by computing the mean of the relative errors. This produced two line plots showing the change in both the average error and the highest relative errors across the tested multiplier spectrum.

3.8.3. Enclosure Bias Test, and Indoor vs. Outdoor Deployment Test

Figure 14 shows the system layout for the enclosure bias test. The setup differed from the one described in the previous section by incorporating an extension arm to offset the pipe from the base post. This allowed the enclosure to be easily adjusted, making the data gathering process more efficient. For the first test, the water height inside the graduated cylindrical enclosure was increased by 0.02 m every 8 minutes over a total duration of 56 minutes. For every one-minute interval, the readings from the FMPS were recorded via PuTTY.

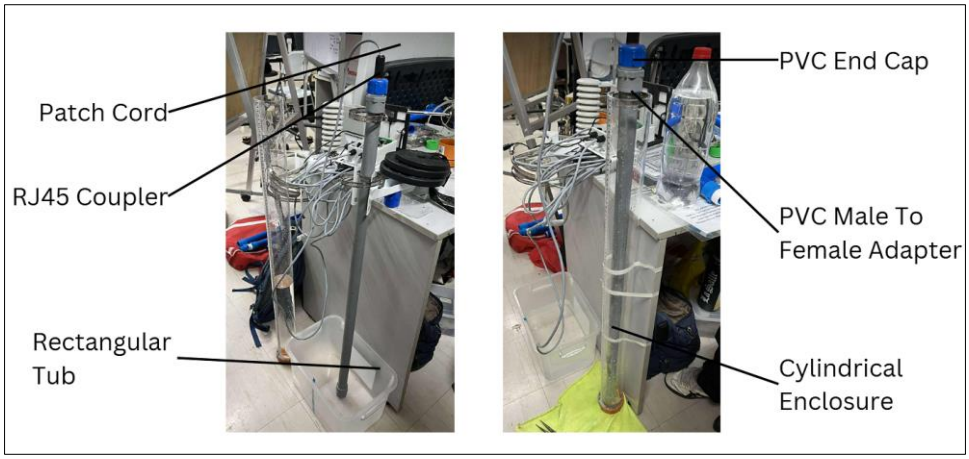


Figure 14. Enclosure bias test setup

Subsequently, the process was repeated for the rectangular tub enclosure. Similarly, for the indoor versus outdoor test illustrated in Figure 15, only the deployment location was varied. The sensor readings were monitored, and the percent difference between both deployments was calculated to determine the influence of the location and the enclosure shape on the function of the FMPS.

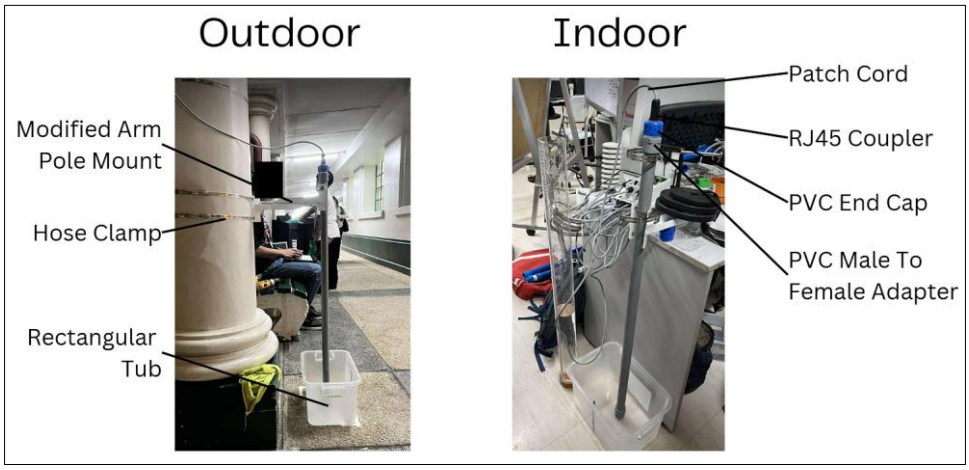


Figure 15. Outdoor versus indoor deployment test setup

3.8.4. SVR Prediction Test

The SVR model's accuracy was determined by calculating the R^2 and RMSE values using Python's standard scientific computing and machine learning libraries. This was executed for both the RBF and linear kernels for comparison with each other. The kernel exhibiting superior metrics was then selected for the flood prediction model. To further visualize model performance, the experimental values were plotted against the predicted values to determine how far the actual prediction is from the ideal prediction line ($y = x$). Time validity was then determined based on the time range of the experimental testing dataset and its corresponding prediction data points.

3.8.5. LoRa Connection, and Data Visualization Test

LoRa connectivity testing was done by outputting telemetry to two separate platforms: the Arduino IDE's serial monitor running at a 9600 baud rate, and the PacketView website. Both platforms displayed the data in real time. However, the serial monitor provided the data on a localized platform, whereas PacketView allowed for remote, real-time viewing via a web browser.

In both scenarios, the data points were collected, and their percent differences were calculated to determine post-transmission data discrepancies. This enabled the researchers to quantify any packet losses across the LoRaWAN server. For the data visualization test, telemetry was sent from the device to the PacketView website. The test was considered a success when the data captured in the serial monitor matched the values displayed on their corresponding website widgets (e.g., internal pipe pressure was outputted on the pipe pressure widget).

4. Results and Discussion

4.1. Battery Duration Test

Table 3 shows the battery test results from the theoretical calculation and the actual deployment test. When fully deployed, the FMPS was able to last for only 3 days and 53 minutes, which is 1 day, 18 hours, and 19 minutes shorter than the theoretical draw. This can be attributed to the solar panel not fully charging due to the cloudy weather, which minimized sunlight exposure to the photovoltaic cell. Given the earlier assumptions of rounding up the daily consumption and applying a safety factor of 1.85 to the battery capacity, the results verify the functionality of the FMPS battery during deployment scenarios.

Table 3. FMPS battery test results

Calculated Daily Power Consumption	Rounded Up Daily Power Consumption	Battery Capacity (from specifications sheet)	Assumed Usable Battery Capacity	Theoretical Battery Duration
0.4417 Wh	0.5 Wh	4.4 Wh	2.4 Wh	4d 19h 12m
Voltage Output	**Measured Capacity**	**Converted Capacity**	**Actual Battery Duration**	**Difference**
3.69 V	1200 mAh	4.44 Wh	3d 0h 53m	1d 18h 19m

4.2. Sealing Test

Table 4 shows the FMPS probe submersion testing data. Water infiltration, in this instance, refers to how much water enters the pipe after being submerged into water. If the water infiltration increases with a longer submersion time, then it means that there is a leakage present in the end cap assembly. The end cap connector seal's first iteration was able to sustain a water infiltration of 25.4 mm for the first minute and increased to 38.1 mm after 15 minutes of submersion, therefore indicating a leakage. For the second iteration, the 25.4 mm of water infiltration was maintained for both 1-minute and 15-minute submersion times, which led to this iteration being used for the probe.

Table 4. Submersion test data

Submersion Duration	Sealed End Cap	
	1st Iteration Infiltration	2nd Iteration Infiltration
1 min	25.4 mm	25.4 mm
15 min	38.1 mm	25.4 mm

To further test the seal, the inside pipe pressure was measured at increasing water heights over time, as seen in Figure 16. Since pressure sensors have sensitive readings, having a graph that flatlines at 101325 kPa or 1 atm of pressure would indicate the presence of an air leak in the system. However, since the pipe pressure does not drop or return to atmospheric pressure, this proves that the hermetic sealing of the end cap connector was successful. It is important to note that the water height was increased every 8 minutes to test if the pressure readings would also vary at different water heights. Because the readings were relatively stable prior to increasing the water height, it can be said that the sensors are functioning as intended.

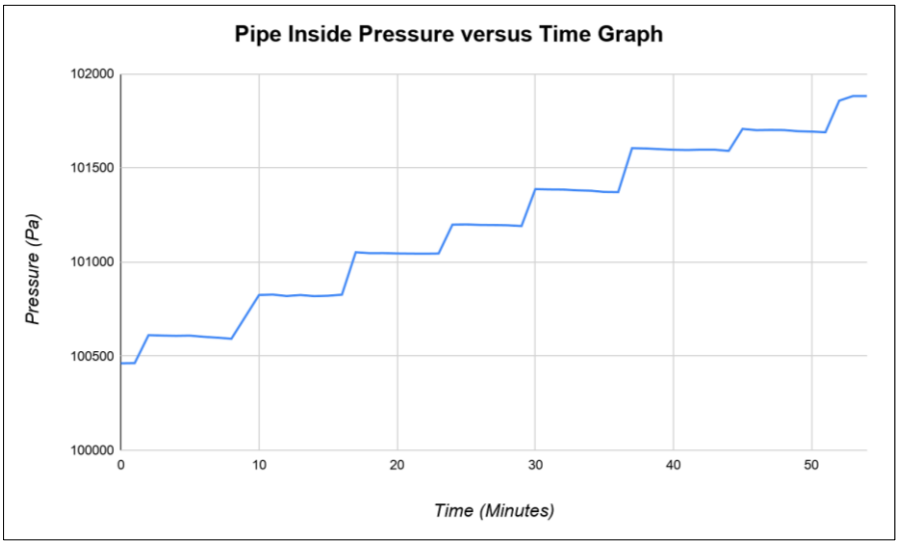


Figure 16. Line graph of pipe inside pressure versus time

Finally, Table 5 shows the end cap bubble test results. The pressure values chosen for the test were based on the limitations of the DPS310 pressure sensor, which was 125 kPa. This corresponded to 2 meters of flood height, but the seal was tested at higher pressures to ensure the seal's durability. The test revealed that for all pressure values, no bubbles were observed in the end cap assembly. The presence of bubbles would mean that a section of the end cap assembly did not have a proper application of epoxy resin and Teflon tape, therefore allowing air bubbles to form as a result of a leakage.

Table 5. End cap bubble test

Pressure	Presence of Bubbles
100 kPa	No
150 kPa	No
200 kPa	No
300 kPa	No

4.3. Sensor Calibration Test

Figure 17 shows the plot of the pipe inside pressure and the actual flood height. The graph shows that when the flood height is increased, the pipe's internal pressure also increases at a nonlinear rate. Observing the graph, the trend for both the uncalibrated and calibrated DPS310 sensors was the same. However, if the sensor is calibrated, then the negative pressure readings are canceled, which allows for more accurate readings.

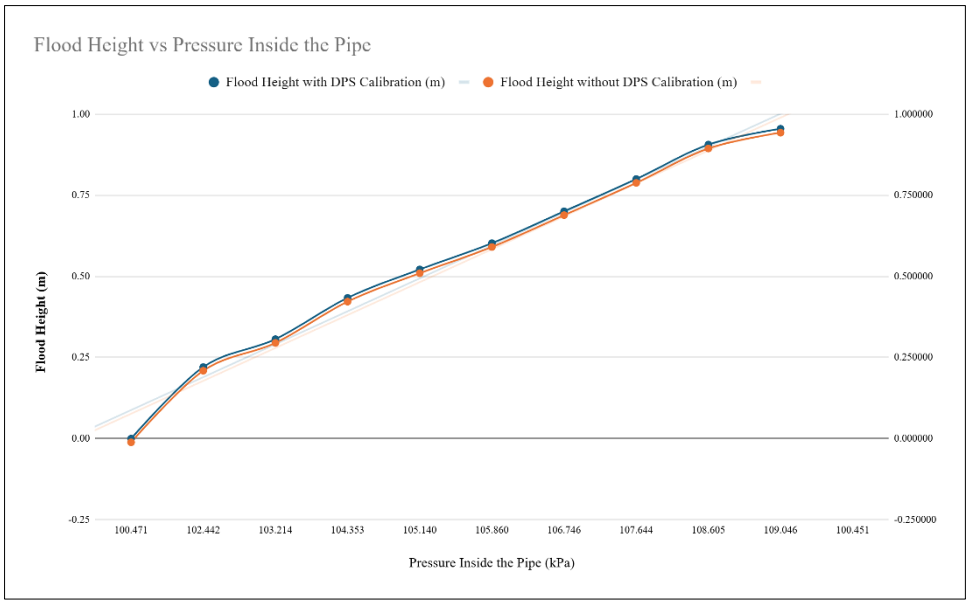


Figure 17. Flood height versus inside pipe pressure plot

Figure 18 shows the plot of the average relative errors with respect to the applied DPS310 multiplier, which shows the difference between the actual flood height reading and the calculated height. The graph shows that a multiplier of 1.001 to the DPS310 sensor leads the calculated height to approach the actual height reading. This multiplier is applicable to all data points for this test because the device computes for the flood height based on the reading provided by the pressure sensors at a particular instance. This implies that only the internal pipe pressure and ambient atmospheric pressure affect the flood height calculation. If the device were to be deployed in different environmental conditions, then a slight recalibration might be required due to barometric and ambient temperature values which affect the system's accuracy.

Figure 19 shows the plot of the maximum relative errors with respect to the applied DPS310 multiplier, which defines the largest deviation of the data points. The graph shows that the correction factor to be applied remains the same regardless of which error basis was selected, with the lowest average relative error being 0.577% while the lowest value of the highest relative error was 1.705%.

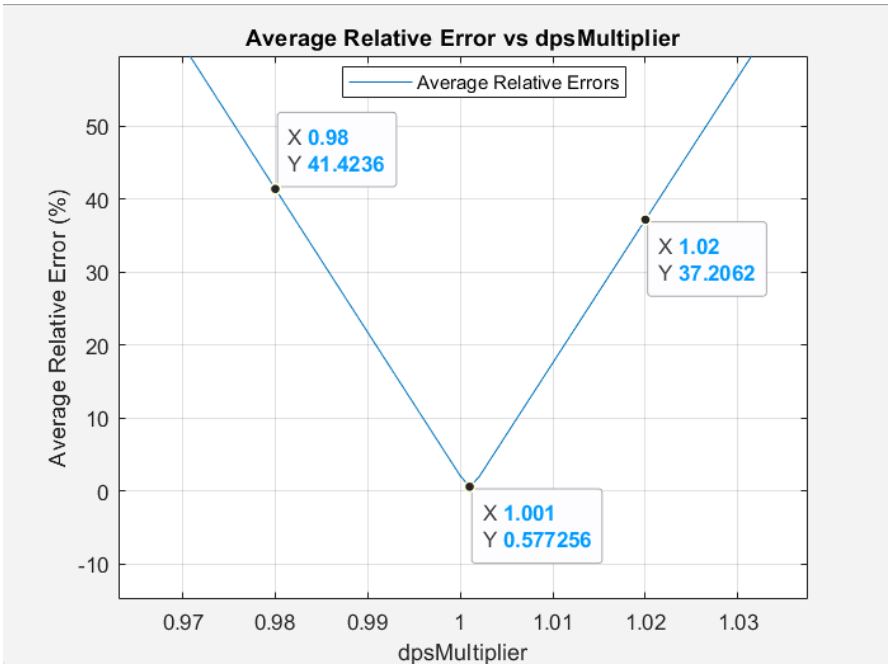


Figure 18. DPS310 multiplier versus average relative error plot

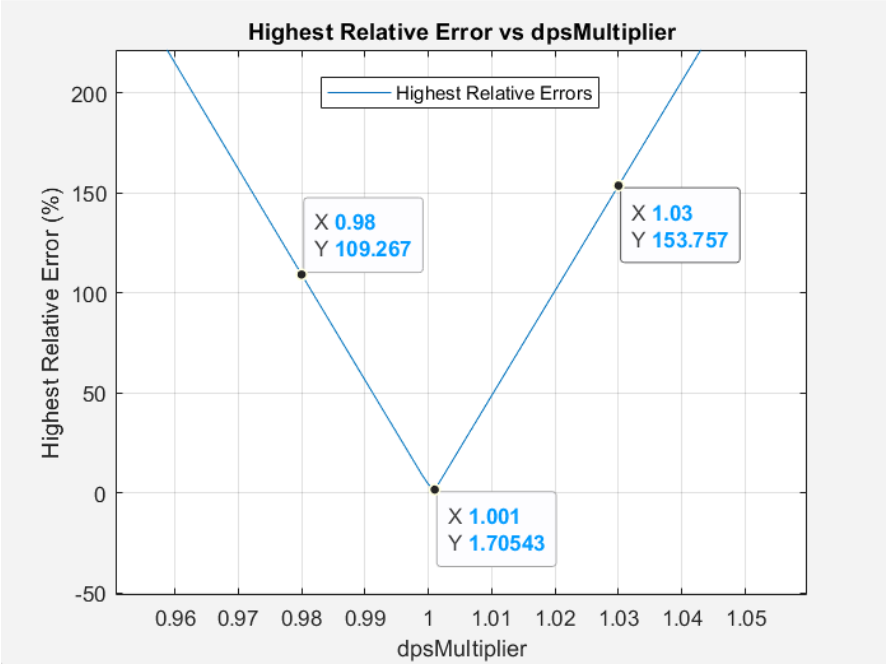


Figure 19. DPS310 multiplier versus the highest relative error plot

4.4. Enclosure Bias and Indoor versus Outdoor Deployment Test

Figure 20 shows the comparison of the pipe inside pressure for the cylindrical versus the rectangular enclosure. Based on the graph, the difference between the two enclosures was roughly 0.1 kPa, which is considered negligible in the flood height calculation. This indicates that the enclosure size and geometry did not have a significant effect on the pipe pressure readings, meaning that the device should function as normal regardless of whether the setup is experimental or a real-world deployment.

Moreover, Figure 21 shows the comparison of the pipe inside pressure for the indoor versus the outdoor deployment. The pressure difference was also shown to be roughly 0.1 kPa, which indicates that the deployment location does not affect the pipe pressure readings. However, other environmental parameters, such as relative humidity and temperature, fluctuated greatly, which indicates that only these two parameters vary significantly with respect to where the FMPS is deployed.

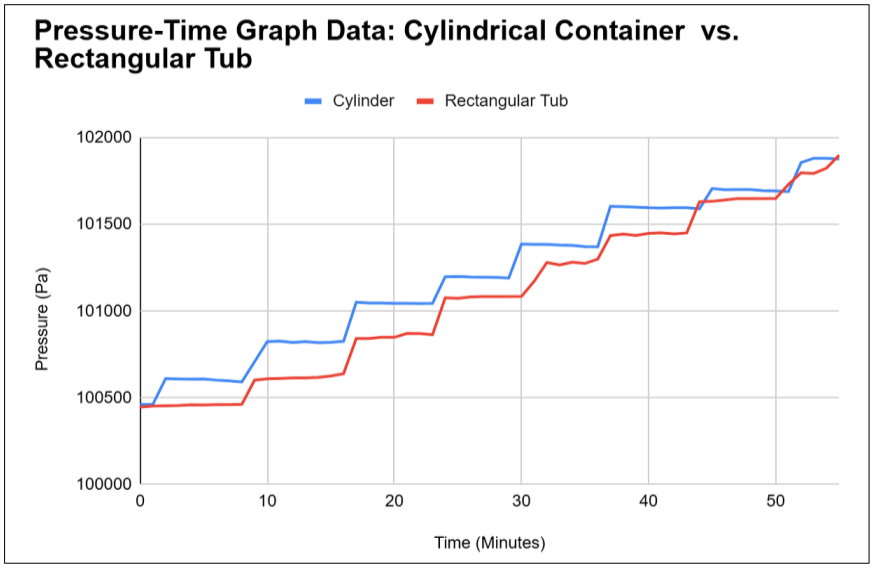


Figure 20. Pipe inside pressure comparison of the cylindrical versus the rectangular enclosure

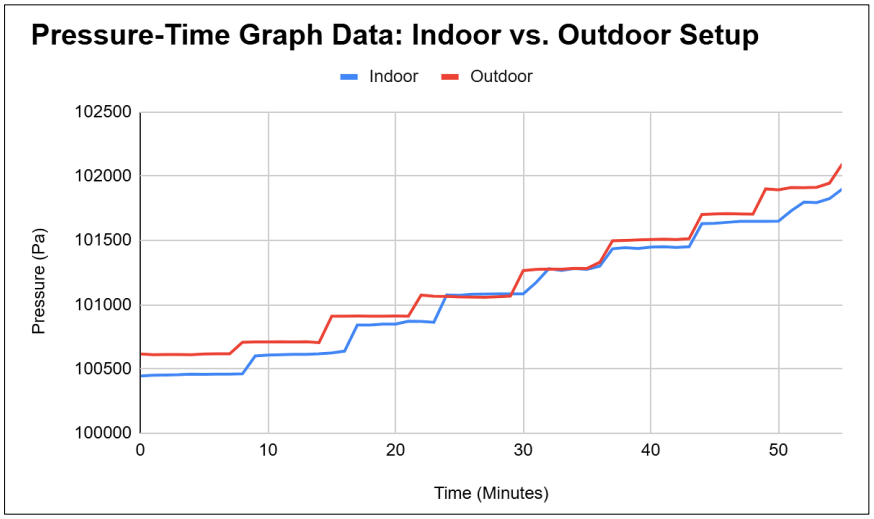


Figure 21. Pipe inside pressure comparison of the indoor versus the outdoor deployment

4.5. SVR Prediction Test

Using the RBF and linear kernels with optimized parameters, Figures 22a and 22b show the actual versus predicted values plot for both kernels, where the red dotted line is the ideal prediction line (actual equals predicted). The R^2 coefficient of the linear kernel was observed to be 0.903, while that of the RBF kernel was 0.948, indicating that the RBF kernel produces more accurate predictions than the linear kernel.

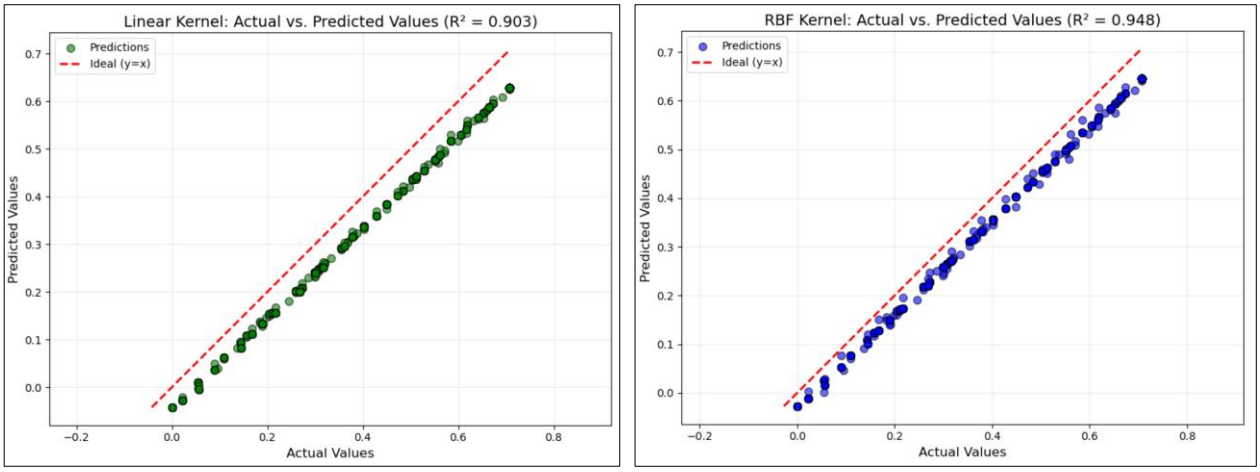


Figure 22. Actual versus predicted values plot of the linear kernel (left) and the RBF kernel (right)

Figure 23 shows the time series plot of the actual and predicted values. As observed in the graph, the trendline for the predicted values follows that of the actual values with a small error, where the predicted values were slightly lower than the actual values. The generated SVR model had an MAE of 0.0512 and an RMSE of 0.0523 for one hour and 30 minutes of simulated data, indicating the accuracy of the prediction model over that time frame.

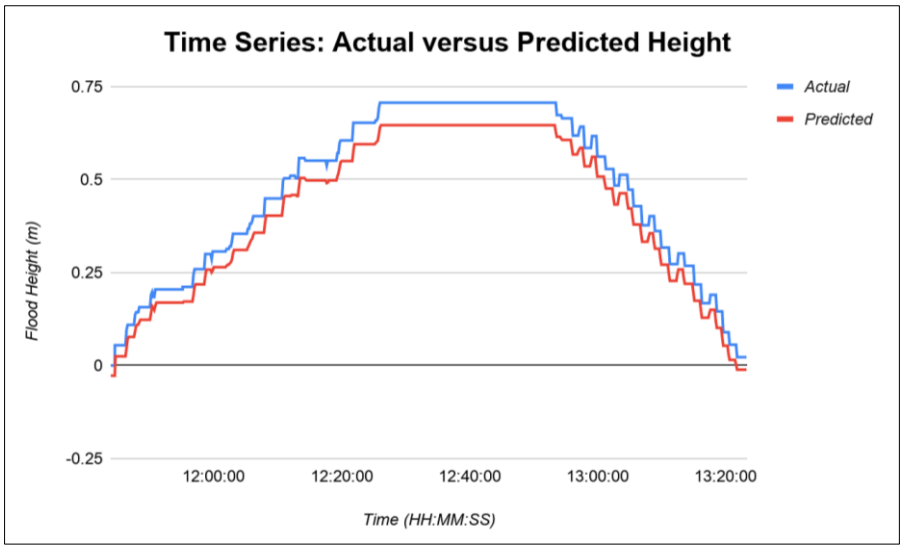


Figure 23. Time series plot of actual and predicted values

Table 6 shows a comparison of this study's flood prediction model with other studies. Compared to Tolentino et al.'s study on MLR [7], their model is based on river flow rate, precipitation rate, and present flood level with an RMSE of 0.3536. With their R^2 values stated in the Table 6, the generated SVR model accounts for more environmental variables with a higher R^2 coefficient and RMSE, indicating that a more complex machine learning algorithm is better suited for a nonlinear dataset.

Looking at the neuro-fuzzy network created by Arante et al. [27], their hydrological model utilizes the same parameters as this study with the exception of precipitation, which is expressed as a 0 or 1 boolean value indicating the presence of rain as opposed to an actual numerical value like in this study. For a more complex prediction model, their RMSE is lower which shows that a more complex prediction model has a better ability to relate variables with each other. This, however, comes at the price of a more complicated implementation to the actual system.

Finally, Yan et al.'s [32] study on a combined SVC and SVR model utilizes a more complex hydrological model featuring a 1D pipe network coupled with a 2D surface model using the MIKE Flood software. While their model was able to generate a higher correlation coefficient than this study, their RMSE value was higher, indicating that a more complex hydrological model has more factors that can affect prediction accuracy, but can better represent the flooding situation as opposed to a simpler model like in this study.

Table 6. Comparison of this study to other flood prediction models

Author	Machine Learning Model	Hydrological Model Variables	R^2 Coefficient	RMSE
Present study	SVR	Pipe and Barometric Pressure, Ambient Temperature and Humidity, Present Flood Level, Precipitation Rate	0.948	0.0523
Tolentino et al. (2022) [7]	Multiple Linear Regression (MLR)	Precipitation Rate	0.9258 for flood level and precipitation rate	0.3536
		River Flow Rate	0.9606 for river flow rate and precipitation rate	
		Present Flood Level	0.8884 for flood level and river flow rate	
Arante et al. (2025) [27]	Neuro-fuzzy Network with Genetic-Based Algorithm	Pipe and Barometric Pressure, Ambient Temperature and Humidity, Present Flood Level, Presence of Rain	N/A	0.036986
Yan et al. (2018) [32]	Combined SVC and SVR	1D Pipe Network with 2D MIKE Flood Software Surface Model	0.972	0.038

4.6. LoRa Distance Test

Table 7 shows the LoRa distance test results, which utilized Google Maps for approximating the distance between the gateway and the FMPS. For Yuchengco and St. Joseph Hall, the FMPS was able to connect to the LoRaWAN server and transmit data. However, for St. Miguel Hall and Velasco Building, the heavy obstruction level and long distance led

to a failure in connecting to the server and sending data. Finally, for the Corazon Aquino Democratic Space (CADS), despite the short distance, it was only able to attempt a connection to the server. These findings indicate that obstruction level and distance affect data transmission, with the former having a stronger influence on the data transmission process. This implies that the optimal setup for the gateway is that it should be placed at a high location to increase its visibility to the FMPS, and the FMPS can be placed along the electrical posts near roads for it to be securely mounted.

Table 7. LoRa distance test

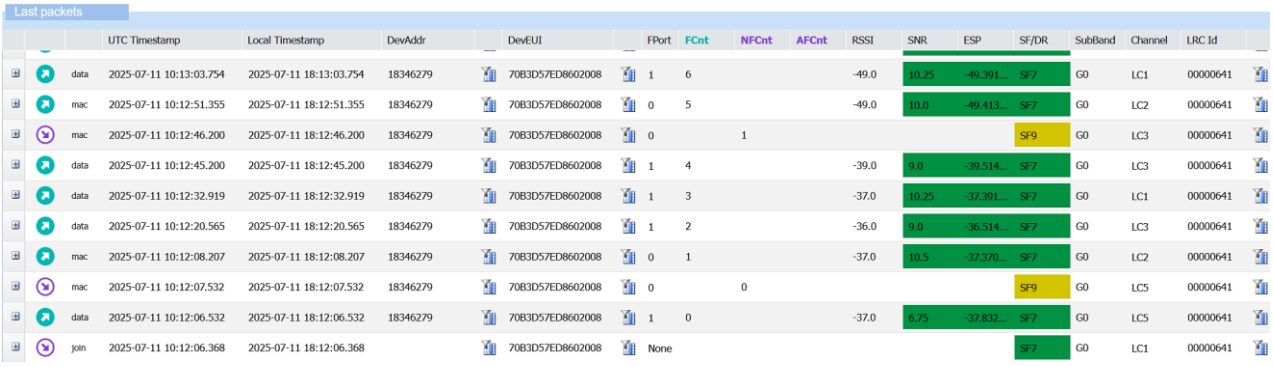
Device	Location	Distance from FMPS	Obstruction Level	Signal Quality	Connection Status
FMPS	LS Walk (near Central Plaza)	N/A	N/A	N/A	N/A
Gateway	Yuchengco Hall	37.64 m	Light	Good	Connected
	St. Joseph Hall	70.26 m	Light	Good	Connected
	St. Miguel Hall	108.56 m	Heavy	Poor	Failed
	Velasco Building	101.1 m	Heavy	Poor	Failed
	CADS	37.44 m	Heavy	Poor	Attempted

To discuss the parameters above, obstruction level refers to the degree to which the FMPS's line of sight to the gateway was blocked. Signal quality refers to the quality of the connection between the FMPS and the LoRaWAN server. Finally, connection status refers to whether the FMPS was able to connect to the LoRaWAN server and send data to it. The main limitation in this test was the optimization of the network coverage, since the gateway needed to be connected to a power source in order to activate its connection, therefore limiting its possible deployment locations.

If FMPS implementation is scaled to multiple nodes over a specific area, there are a few factors that affect system performance. Firstly, the topology of Metro Manila limits the visibility of the gateway from the FMPS, which would lead to data transmission errors. The gateway is further limited by the requirement of a stable internet connection, which reduces the number of locations where it can be installed. Finally, communication between multiple FMPS nodes and one gateway may lead to a data transmission bottleneck, which leads to delays in visualizing the pertinent flood data.

4.7. LoRa Connection and Data Visualization Test

Figure 24 shows the Actility LNS data transmission log, which records and verifies any communication between the LoRaWAN server and the FMPS. There are two main connections in LoRaWAN, which are Over-The-Air Activation (OTAA) or Activation by Personalization (ABP). The former works by the end-device sending a Join Request with its unique identifiers to the server, which responds with a Join Accept message, which includes the device address, the app session key, and the network session key that are randomly generated for each new join session, therefore making it more secure and resilient against replay attacks and key reuse. The latter, on the contrary, skips the join procedures and already provides the keys and device address, which automatically connects the server and the device. This provides a faster implementation at the cost of less network security. For this application, OTAA was utilized in order to ensure the security of the data being transmitted from the device to the LoRaWAN server.


Last packets

	UTC Timestamp	Local Timestamp	DevAddr	DevEUI	FPort	FCnt	NFCnt	AFCnt	RSSI	SNR	ESP	SF/DR	SubBand	Channel	LRC Id
data	2025-07-11 10:13:03.754	2025-07-11 18:13:03.754	18346279	70B3D57ED8602008	1	6			-49.0	10.25	-49.391...	SF7	G0	LC1	00000641
mac	2025-07-11 10:12:51.355	2025-07-11 18:12:51.355	18346279	70B3D57ED8602008	0	5			-49.0	10.0	-49.413...	SF7	G0	LC2	00000641
mac	2025-07-11 10:12:46.200	2025-07-11 18:12:46.200	18346279	70B3D57ED8602008	0		1					SF9	G0	LC3	00000641
data	2025-07-11 10:12:45.200	2025-07-11 18:12:45.200	18346279	70B3D57ED8602008	1	4			-39.0	9.0	-39.514...	SF7	G0	LC3	00000641
data	2025-07-11 10:12:32.919	2025-07-11 18:12:32.919	18346279	70B3D57ED8602008	1	3			-37.0	10.25	-37.391...	SF7	G0	LC1	00000641
data	2025-07-11 10:12:20.565	2025-07-11 18:12:20.565	18346279	70B3D57ED8602008	1	2			-36.0	9.0	-36.514...	SF7	G0	LC3	00000641
mac	2025-07-11 10:12:08.207	2025-07-11 18:12:08.207	18346279	70B3D57ED8602008	0	1			-37.0	10.5	-37.370...	SF7	G0	LC2	00000641
mac	2025-07-11 10:12:07.532	2025-07-11 18:12:07.532	18346279	70B3D57ED8602008	0		0					SF9	G0	LC5	00000641
data	2025-07-11 10:12:06.532	2025-07-11 18:12:06.532	18346279	70B3D57ED8602008	1	0			-37.0	6.75	-37.832...	SF7	G0	LC5	00000641
join	2025-07-11 10:12:06.368	2025-07-11 18:12:06.368		70B3D57ED8602008	None							SF7	G0	LC1	00000641

Figure 24. Time series plot of actual and predicted values

To verify the accuracy of the transmitted data, the data before and after transmission were compared to obtain their percent difference. This was done through the Serial Monitor of the Arduino IDE for pre-transmission data, and the PacketView website for post-transmission data. The measured difference between the two was about 25.68 Pascals on average, translating to about a 0.025% difference, which indicates that there are sustained packet losses during the data transmission process due to decoding and encoding errors from the device and the LoRaWAN server. However, the magnitude of this difference is not significant enough to alter flood height calculations. Aside from packet loss, the

difference may increase due to long-term sensor drift from the deterioration of the pressure sensors, and environmental interference such as vibrations and harsh weather conditions which may affect the accuracy of the device's readings.

Figure 25 shows the PacketView website layout, which was utilized to visualize key FMPS information. It features four main widgets: the probe location and status widget (upper left), the sensor reading widget (upper right), the flood monitoring and prediction widget (bottom left), and the alarm widget (bottom right). This layout provides a direct way to numerically visualize the data. If the three dots on the upper right of each widget is clicked, then the data can be displayed as a time series graph of that parameter as seen in Figure 26.

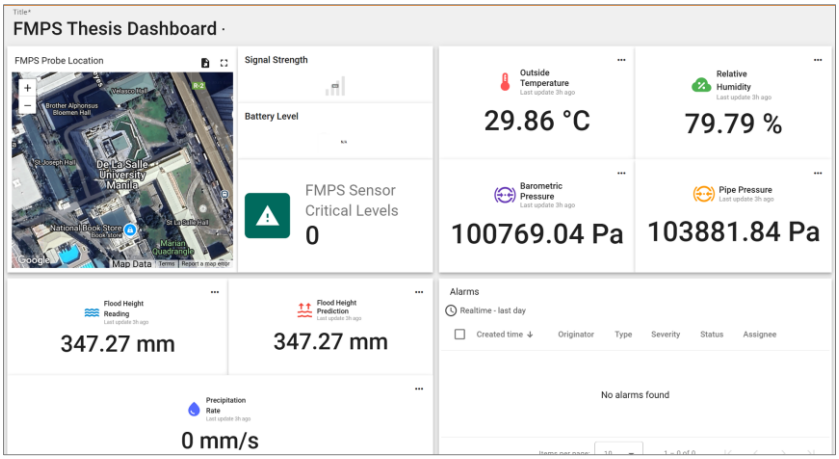


Figure 25. PacketView website general layout

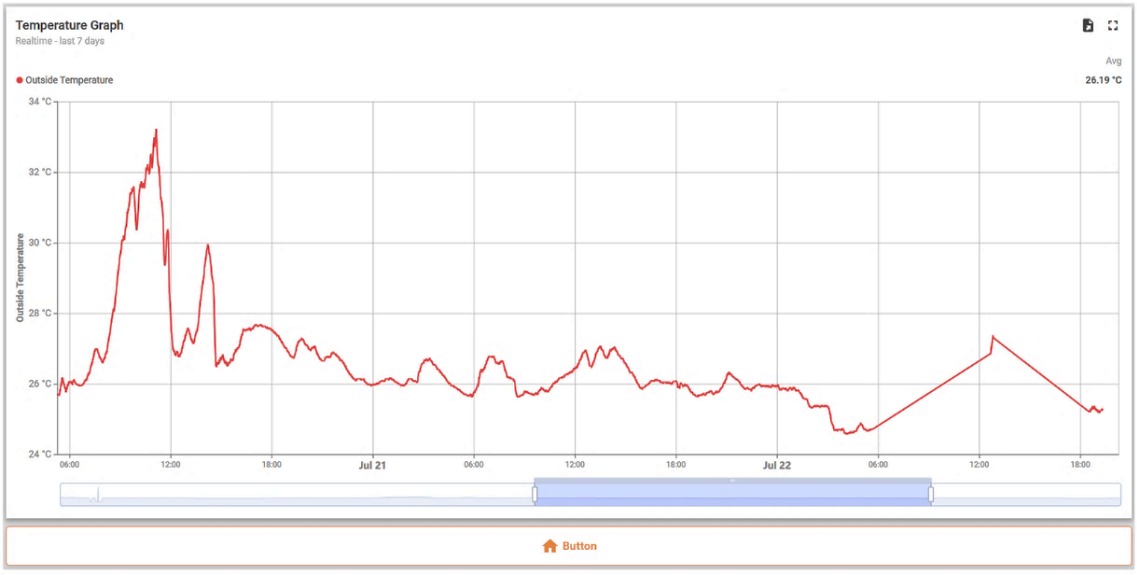


Figure 26. PacketView website telemetry graph sample

In real time, the FMPS transmits data to the LoRaWAN server when its sensors generate readings. Then, the values and graphs update every few seconds when the FMPS gathers a new reading. The alarm system, though not fully implemented in the study, is then triggered when a 0.5-meter flood height is reached until 2 meters is reached. This can be integrated into existing systems, such as PAGASA's rainfall warning systems, or by utilizing the MMDA flood height color scheme.

Figure 27 shows the FloodCast website created by Seva et al. [33]. This interface visualizes the MMDA flood height color scheme along with its corresponding interpretations. The page also features a time-series graph plotting flood height predictions throughout the day, along with a street camera photo of the selected location at a particular time. This enables users to quickly understand the implications of each flood level, thereby allowing them to respond effectively in the event of an emergency. Conversely, the PacketView interface showcases a complete set of numerical flood and environmental telemetry which can also be converted into graphical formats. While the text-heavy interface is less intuitive than the FloodCast website, the completeness of the data enables users to generate deeper insights regarding the various factors contributing to flood height fluctuations. Therefore, this allows stakeholders to make informed decisions regarding flood mitigation.

Figure 27. FloodCast website information page [33]

5. Conclusion

The FMPS leverages replicable and cost-effective components to achieve system modularity and functionality. The PVC endcaps achieved hermetic sealing through the proper application of epoxy resin and Teflon tape, validated by submersion and bubble tests. Moreover, other evaluations such as location and data transmission tests, indoor versus outdoor deployment, and enclosure bias tests revealed minimal data packet losses and sensor readings variances under diverse environmental conditions, proving its capability for real-world deployment. The implemented SVR model generates flood predictions with a correlation coefficient of 0.9477, RMSE of 0.0523, and a look-ahead prediction validity of 1.5 hours, highlighting its capability for improving emergency response times. Finally, the PacketView website showcases versatility in data visualization, ensuring intuitive graphical and numerical telemetry data formats for non-technical community stakeholders.

To further improve the study, future work should focus on five major areas. Firstly, weather chamber simulations can test the system's endurance against adverse weather conditions, therefore strengthening the FMPS' environmental durability. Next, extending the deployment of the FMPS can generate more historical data to further refine the system's SVR model. Scaling FMPS implementation to multiple nodes leads to a more dynamic system that can accurately map flood data for a particular area. Testing the telemetry against dynamic disturbances such as forced vibrations, wind loads, and turbulent fluid flows can isolate sensor instability. Finally, a comparison of the SVR model against more computationally intensive machine learning algorithms aids in the identification of the best hydrological and mathematical model for non-linear flooding scenarios.

6. Declarations

6.1. Author Contributions

Conceptualization, A.C.; methodology, J.B.R., N.C.F.N., M.R.V., and J.R.P.; software, J.B.R., M.R.V., and N.C.F.N.; validation, J.B.R, N.C.F.N., and M.R.V.; formal analysis, N.C.F.N., J.B.R., and J.R.P.; investigation, N.C.F.N., M.R.V., J.B.R., J.R.P., and E.J.D.; resources, N.C.F.N., M.R.V., J.B.R., J.R.P., E.J.D., and A.C.; data curation, J.B.R., M.R.V., J.R.P., and N.C.F.N.; writing—original draft preparation, E.J.D., N.C.F.N, J.B.R., J.R.P., and M.R.V.; writing—review and editing, N.C.F.N., M.R.V., J.R.P., J.B.R., E.J.D., and A.C.; visualization, J.R., M.R.V., J.R.P., N.C.F.N., and E.J.D.; supervision, A.C.; project administration, N.C.F.N. and A.C.; funding acquisition, N.C.F.N., J.B.R., J.R.P., M.R.V., E.J.D., and A.C. All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The training data was obtained from Arante et al. [27] and available at https://github.com/Dragonex68/-sensors-3656218/tree/main/_data (accessed on May 2026) with the permission of Dr. Alvin Chua and Hero Rafael Arante.

The rest of the FMPS telemetry data presented in this study are available on request from the corresponding author. The data are not publicly available due to it being transmitted directly to the created PacketView website.

6.3. Funding and Acknowledgments

The authors would like to show their appreciation to the DLSU Department of Mechanical Engineering, and Packetworx IoT Hub for the resources and insights that they have provided.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. 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.

7. References

[1] Reyes, M. A. L. L. (2023). Worsening garbage problem. Pilipino Star Printing Co., Metro Manila, Philippines. Available online: https://www.philstar.com/business/2023/02/04/2242354/worsening-garbage-problem (accessed on May 2026).

[2] Ong, G. (2024). 90 trucks of trash hauled in Metro after floods. Pilipino Star Printing Co., Metro Manila, Philippines. Available online: https://qa.philstar.com/nation/2024/07/30/2373969/90-trucks-trash-hauled-metro-after-floods (accessed on May 2026).

[3] Natividad, J. G., & Mendez, J. M. (2018). Flood Monitoring and Early Warning System Using Ultrasonic Sensor. IOP Conference Series: Materials Science and Engineering, 325(1), 12020. doi:10.1088/1757-899X/325/1/012020.

[4] Fontillas, K. A. P., Lu, A. J. C., Panandigan, Y. L. A., & Seacor, A. L. S. (2023). Initial Testing and Design of a Portable Ultrasonic Flood Level Monitoring System. 5th DLSU Senior High School Research Congress, 1-8.

[5] Yuwono, H., Limantara, L. M., Sholichin, M., & Siswoyo, H. (2025). A Model for the Reduction of Flood Peak Discharge (ΔQp) Due to the Retarding Basin. Civil Engineering Journal, 11(12), 5132–5143. doi:10.28991/CEJ-2025-011-12-012.

[6] Wan Hassan, W. H., Jidin, A. Z., Aziz, S. A. C., & Rahim, N. (2019). Flood disaster indicator of water level monitoring system. International Journal of Electrical and Computer Engineering, 9(3), 1694–1699. doi:10.11591/ijece.v9i3.pp1694-1699.

[7] Tolentino, L. K. S., Baron, R. E., Blacer, C. A. C., Aliswag, J. M. D., De Guzman, D. C. E., Fronda, J. B. A., ... & Fernandez, E. O. (2022). Real time flood detection, alarm and monitoring system using image processing and multiple linear regression. Journal of Computational Innovations and Engineering Applications, 7(1), 12-23.

[8] Dublin, A. C., Arce, M., Ortiz, E., Wong, L. M., Villaruel, K. P., Arante, H. R., Co, D., Chua, A., Sybingco, E., & Roque, M. A. (2024). a Novel Cost-Effective Pressure Sensor Based Flood Monitoring System With IoT. ASEAN Engineering Journal, 14(3), 53–61. doi:10.11113/aej.V14.20668.

[9] Te, M. C. L., Bautista, J. A. T., Dimacali, S. M. E. V., Lood, A. V. M., Pangan, M. G. M., & Chua, A. Y. (2024). A Smart IoT Urban Flood Monitoring System Using a High-Performance Pressure Sensor with LoRaWAN. HighTech and Innovation Journal, 5(4), 918–936. doi:10.28991/HIJ-2024-05-04-04.

[10] Amitrano, D., Di Martino, G., Di Simone, A., & Imperatore, P. (2024). Flood Detection with SAR: A Review of Techniques and Datasets. Remote Sensing, 16(4), 656. doi:10.3390/rs16040656.

[11] Tamašauskas, R., Grendaitė, D., & Stonevičius, E. (2026). Water level estimation from radar altimetry data in lowland temperate climate rivers. International Journal of Remote Sensing, 1–22. doi:10.1080/01431161.2026.2660252.

[12] Dzwonkowski, K., Winnicki, I., Pietrek, S., & Kroszczyński, K. (2026). Evaluation of radar-based precipitation estimates during a flood event using rain gauge validation. Scientific Reports, 16(1), 11174. doi:10.1038/s41598-026-40456-z.

[13] Sanz, E., Trincado, J., Martínez, J., Payno, J., Morante, O., Almeida-Ñaulay, A. F., Berlanga, A., Molina, J. M., Zubelzu, S., & Patricio, M. A. (2024). Cloud-based system for monitoring event-based hydrological processes based on dense sensor network and NB-IoT connectivity. Environmental Modelling and Software, 182, 106186. doi:10.1016/j.envsoft.2024.106186.

[14] Pahuriray, A. V., & Cerna, P. D. (2025). IoT-Enabled Flood Monitoring and Early Warning Systems: A Systematic Review. International Journal of Computer Science and Mobile Computing, 14(4), 50–67. doi:10.47760/ijcsmc.2025.v14i04.005.

[15] Singh, J., V. Swaroopa, & Bharathwaj, K. (2024). Comparative Analysis of Communication Technologies for Cloud-Based Flood Risk Predictive Modeling in Urban Areas. International Journal of Trendy Research in Engineering and Technology, 8(2), 11–19. doi:10.54473/ijtret.2024.8203.

[16] Tadrist, N., Debauche, O., Saïd, M., & Guttadauria, A. (2022). Towards Low-Cost IoT and LPWAN-Based Flood Forecast and Monitoring System. Journal of Ubiquitous Systems and Pervasive Networks, 17(1), 43–49. doi:10.5383/juspn.17.01.006.

[17] Rahman, H. U., Ahmad, M., Ahmad, H., & Habib, M. A. (2020). LoRaWAN: State of the Art, Challenges, Protocols and Research Issues. Proceedings - 2020 23rd IEEE International Multi-Topic Conference, INMIC 2020, 1–6. doi:10.1109/INMIC50486.2020.9318170.

[18] Abbot, J., & Marohasy, J. (2014). Input selection and optimisation for monthly rainfall forecasting in Queensland, Australia, using artificial neural networks. Atmospheric Research, 138, 166–178. doi:10.1016/j.atmosres.2013.11.002.

[19] Mosavi, A., Ozturk, P., & Chau, K. W. (2018). Flood prediction using machine learning models: Literature review. Water (Switzerland), 10(11), 1536. doi:10.3390/w10111536.

[20] Jimeno-Sáez, P., Senent-Aparicio, J., Pérez-Sánchez, J., Pulido-Velazquez, D., & María Cecilia, J. (2017). Estimation of instantaneous peak flow using machine-learning models and empirical formula in Peninsular Spain. Water (Switzerland), 9(5), 347. doi:10.3390/w9050347.

[21] Manocha, A., Sood, S. K., & Bhatia, M. (2025). Digital Twin-Assisted Fuzzy Logic-Inspired Intelligent Approach for Flood Prediction. IEEE Sensors Journal, 25(15), 27800–27807. doi:10.1109/JSEN.2023.3322535.

[22] Vakili, S., & Mousavi, S. M. (2022). Rainfall–runoff modeling using adaptive neuro-fuzzy inference system (ANFIS) and genetic algorithm (GA). Water Supply, 22(10), 7460–7475. doi:10.2166/ws.2022.318.

[23] Hakim, D. K., Gernowo, R., & Nirwansyah, A. W. (2024). Flood prediction with time series data mining: Systematic review. Natural Hazards Research, 4(2), 194–220. doi:10.1016/j.nhres.2023.10.001.

[24] Rahmati, O., Darabi, H., Panahi, M., Kalantari, Z., Naghibi, S. A., Ferreira, C. S. S., Kornejady, A., Karimidastenaei, Z., Mohammadi, F., Stefanidis, S., Tien Bui, D., & Haghighi, A. T. (2020). Development of novel hybridized models for urban flood susceptibility mapping. Scientific Reports, 10(1). doi:10.1038/s41598-020-69703-7.

[25] Liu, S., Wang, J., Wang, H., & Wu, Y. (2022). Post-processing of hydrological model simulations using the convolutional neural network and support vector regression. Hydrology Research, 53(4), 605–621. doi:10.2166/nh.2022.004.

[26] Quenum, G. M. L. D., Arnault, J., Klutse, N. A. B., Zhang, Z., Kunstmann, H., & Oguntunde, P. G. (2022). Potential of the Coupled WRF/WRF-Hydro Modeling System for Flood Forecasting in the Ouémé River (West Africa). Water (Switzerland), 14(8), 1192. doi:10.3390/w14081192.

[27] Arante, H. R. C., Sybingco, E., Roque, M. A., Ambata, L., Chua, A., & Gutierrez, A. N. (2025). Development of a Secured IoT-Based Flood Monitoring and Forecasting System Using Genetic-Algorithm-Based Neuro-Fuzzy Network. Sensors, 25(13), 3885. doi:10.3390/s25133885.

[28] Haddad, K., & Rahman, A. (2020). Regional flood frequency analysis: evaluation of regions in cluster space using support vector regression. Natural Hazards, 102(1), 489–517. doi:10.1007/s11069-020-03935-8.

[29] Rummler, T., Arnault, J., Gochis, D., & Kunstmann, H. (2019). Role of Lateral Terrestrial Water Flow on the Regional Water Cycle in a Complex Terrain Region: Investigation With a Fully Coupled Model System. Journal of Geophysical Research: Atmospheres, 124(2), 507–529. doi:10.1029/2018JD029004.

[30] Sharma, S., & Kumari, S. (2024). Comparison of machine learning models for flood forecasting in the Mahanadi River Basin, India. Journal of Water and Climate Change, 15(4), 1629–1652. doi:10.2166/wcc.2024.517.

[31] LoRa Alliance. (2026). What is LoRaWAN®. LoRa Alliance, Fremont, United States. Available online: https://lora-alliance.org/resource_hub/what-is-lorawan/ (accessed on May 2026).

[32] Yan, J., Jin, J., Chen, F., Yu, G., Yin, H., & Wang, W. (2018). Urban flash flood forecast using support vector machine and numerical simulation. Journal of Hydroinformatics, 20(1), 232–245. doi:10.2166/hydro.2017.175.

[33] Seva, R. R., Gutierrez, A. N. A., Arante, H., Chua, A., Roque, M. A. C., & Sybingco, E. (2025). FloodCast: A user-centered flood monitoring system to enhance user engagement. Progress in Disaster Science, 100492. doi:10.1016/j.pdisas.2025.100492.

[34] APEC. (2026). What is Hermetic Sealing. Automated Processing Equipment Corporation, & Ford Manufacturing. APEC, Michigan, United States. Available online: https://www.apecusa.com/blog/what-is-hermetic-sealing/ (accessed on May 2026).

[35] Byaruhanga, N., Kibirige, D., Gokool, S., & Mkhonta, G. (2024). Evolution of Flood Prediction and Forecasting Models for Flood Early Warning Systems: A Scoping Review. Water (Switzerland), 16(13), 1763. doi:10.3390/w16131763.