International Journal of Multidisciplinary and Current Educational Research (IJMCER)
ISSN: 2581-7027 ||Volume|| 6 ||Issue|| 4 ||Pages 30-40 ||2024||

Real-time Flood Surveillance and Early Warning System using IoT- AI Fusion for Flood Prediction: Smart Framework to Monitoring Solutions

[1,]Justin Louise R. Neypes, [2,] Maksuda Sultana, DIT, [3,]Andrew Jules R. Miyajima
College of Computer Studies, AMA University

ABSTRACT : The Philippines faces increasing vulnerability to climate change-induced floods and cyclones, particularly in Valenzuela City, Metro Manila. To address this, the researchers propose a Real-Time Flood Surveillance and Early Warning System using IoT and AI. This system integrates IoT sensors, machine learning models, and a web interface to accurately monitor flood levels, predict floods, and enable proactive response measures. Using hardware like NodeMCU and software tools like Firebase, the system demonstrates improved accuracy in flood prediction, with the XGBoost Regressor performing best. This solution empowers authorities to mitigate flood risks and safeguard communities from climate change impacts.

KEYWORDS : Artificial Intelligence (AI), Climate Change, Early Warning System, Flooding, Internet of Things (IoT), Machine Learning, Philippines, Valenzuela City, XGBoost Regressor

I . INTRODUCTION

The Philippines is particularly vulnerable to the effects of climate change, as seen by the rise in extreme weather events and the increasing frequency of cyclones. The effects of these natural catastrophes are severe, with an average of 20 cyclones striking the country each year in addition to the growing problem of flooding [1]. As the most common natural disaster, floods submerge formerly dry ground and cause extensive destruction that includes fatalities and significant harm to private property and vital public health infrastructure [2]. The Philippines is the country most affected by rainfall pattern variability brought on by climate change among Southeast Asian countries [3]. Flooding is a constant hazard, especially in Metro Manila, where it has caused significant damage and affected thousands of individuals. Massive floods cause multimillion- to multibillion-peso losses in agricultural, livestock, infrastructure, and productivity across multiple sectors, aggravating socioeconomic problems [4]. As a result, the cost goes beyond financial sums and includes the loss of human life, forced relocation, and the severe psychological effects on the populations affected.

Located inside the boundaries of Metro Manila, 14 kilometers north of Manila, Valenzuela City is a vital metropolitan hub distinguished by its industrial strength and lively residential neighborhood. Valenzuela, with all its wealth, is physically vulnerable to flooding because of its low-lying terrain and proximity to three interconnecting rivers: the Tullahan, Polo, and Meycauayan. Due to this convergence, the city is more susceptible to flash floods and tidal and surges, which are especially common during the monsoon season. The consequent flooding, which is made worse by poor solid waste management and drainage systems, can last for long periods, with floodwaters remaining for as long as four weeks [5]. Large portions of Valenzuela were drowned in 2023 as the city struggled with the flooding caused by the Southwest Monsoon, also referred to as "Habagat" locally. The significant impact of such climatic catastrophes was demonstrated by the approximately 1,000 families that sought shelter in 13 evacuation centers located across the city [6]. The Valenzuela City government tracks the growing floodwater levels manually and updates its Facebook page with the date, time, location, and status. The manual labor required by the current systems can obstruct prompt and effective disaster management operations. The following are some examples of limitations: (a) data collection delays when staff members use traditional tools to physically measure water levels; (b) lack of real-time information when manual data collection methods only produce sporadic snapshots of water levels that miss the dynamic nature of flood events; and (c) difficulty coordinating emergency responses when communication and collaboration with pertinent stakeholders are hampered by the lack of timely and accurate information. The researchers have suggested implementing an IoT-AI fusion system to tackle this issue. By developing a flood prediction and early warning detection system, the system seeks to reduce the damages and casualties associated with impending floods. The researchers specifically want to address the following goals:

1. Implement an IoT-based sensor network to accurately monitor precise water level estimation and analyze real-time flood levels in vulnerable areas.
2. Utilize machine learning to predict water levels in vulnerable areas, empowering local authorities to take proactive measures and initiate swift responses during potential flooding events.
3. Develop a web-based interface to enable easy access, visualization, and control of the flood monitoring and early detection system.

I. METHODOLOGY

The Real-Time Flood Surveillance and Early Warning System using IoT-AI Fusion was envisioned as an interactive web-based information system that allows the users to (a) accurately monitor precise water level estimation and analyze real-time flood levels, (b) be notified when the flood levels have reached a certain threshold, (c) visualize and control the flood monitoring and early detection system, (d) collect rainfall data, and lastly (e) generate predictions for flood water levels. To offer full flood monitoring and prediction capabilities, the Real-Time Flood Surveillance and Early Warning System was created as an integrated platform consisting of an IoT sensor network, machine learning models, and a web-based interface.

Web Development: The web application's front end uses React.js to produce a smooth and engaging user interface. To make sure the program is not only functional but also aesthetically pleasing on all devices, the researchers integrated Bootstrap. The goal of the design philosophy is to create an interface that is easy to use and intuitive for users. Material UI was selected because of its extensive collection of design components, which enabled the researchers to incorporate an adjustable theme system with options for light and dark modes, improving user comfort and accessibility. Iterative developments were made during the implementation phase to make sure that every feature complied with the user experience objectives.The application's back end is driven by Firebase. To ensure a quick depiction of flood levels, the researchers processed and received data from IoT sensors in real-time using Firebase's real-time database. Firestore is used for storing structured data, such as user credentials and other important data. User profile photos and other assets are handled by Firebase's Storage service for media storage in an efficient manner, ensuring a smooth data flow across the application.

By using a test-driven development methodology, the researchers made sure that every component was completely evaluated before moving on to optimization. The promise, async, and await syntaxes in JavaScript helped the researchers improve the application's speed by reducing unnecessary calls and streamlining data retrieval. The goal of the refactoring stage was to make the code more readable and maintainable while laying the groundwork for further development.The front-end and back-end components have been thoroughly integrated to create a unified and engaging online application. The process makes it easier to create and deploy quickly, enabling real-time updates and prompt feedback on flood conditions. By taking a proactive approach, users might potentially save lives and property in flood-prone areas by making timely decisions.

Machine Learning : Machine learning (ML) is a process in which computing systems learn from data and use algorithms to execute tasks without being explicitly programmed [7]. Machine learning algorithms use past data as inputs to predict future values. A machine learning (ML) model ensures that, given a sample, the expected result matches the actual result each time. How an algorithm learns to improve its prediction accuracy is a common way to classify classical machine learning. Supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning are the four fundamental methods. Depending on the kind of data that needs to be predicted, data scientists select several types of algorithms. The data scientist must use both labeled inputs and desired outputs to train the algorithm in supervised machine learning. For the following tasks, supervised learning algorithms work well with (a) binary classification; (b) multi-class classification; (c) Regression modeling, and (d) Ensembling [8].

The researchers' main focus in this work is regression modeling or the prediction of continuous variable values. Predictive analytics uses machine learning and predictive modeling to examine historical data and forecast future patterns. Regression using machine learning (ML) techniques includes neural network regression, decision tree regression, random forests, support vector regression machines, and others. An ML prediction model called an artificial neural network (ANN) is made to function similarly to a human brain. Neurons in one layer of a neural network provide information to several neurons in the layer above it, and so on. Only input and output nodes are used in linear regression models' prediction processes. The hidden layer is another tool the ANN utilizes to improve prediction accuracy [8].

Performance metrics are used in machine learning regression models to compare the actual data from the testing data set with the predictions made by the trained model. Performance metrics for regression usually involve calculating an error score to summarize the predictive skill of a model. The (root) mean squared error, the mean absolute error, the Pearson correlation coefficient, and the coefficient of determination are the most often used performance metrics for assessing and reporting a regression model's performance [8].

Data Collection : Due to privacy concerns, the researchers built a new dataset that meets two key criteria: (1) closely simulating real weather patterns, and (2) mimicking the output format of Internet of Things (IoT) sensors. The simulated dataset was developed using Microsoft Excel 365 and Visual Basic Script to create a function that calculates weather data for the next 30 minutes.To generate the initial flood height (H_init), the researchers utilize the RAND function to produce values between 0.000 and 4.000 meters. Subsequently, the ROUND function is used to round these values to three decimal places.

H_init = ROUND(RAND()*4, 3)

To generate the rain intensity (R_intensity), the RANDBETWEEN function was used to produce values between 0% and 100%.

R_intensity = RANDBETWEEN(0, 100)

To generate the flood height prediction (Pred_H), the researchers made a calculation with the following constraints:

Let $k = 0.01$, $\Delta H = 0.3$

$$PredH = Hinit + k * Rintensity \ \{-\Delta H \ if \ Rintensity < 40 \mid * 0 \ if \ PredH < 0\}$$

Where:
PredH = Flood Height Prediction
Hinit = Initial Flood Height
Rintensity = Rain Intensity
ΔH = Fixed height deduction when rain intensity is below 40%.
k = Proportionality constant

In order to make this a function, the researchers made a custom function as shown below.

```
Function CalculateFloodHeight(initialHeight As Double, rainIntensity As Double) As Double
        Dim k As Double
        k = 0.01

        Dim deltaH As Double
        deltaH = 0.3

        Dim futureHeight As Double
        futureHeight = initialHeight + k * rainIntensity

        If rainIntensity < 40 Then
                futureHeight = futureHeight - deltaH
        End If

        If futureHeight < 0 Then
                futureHeight = futureHeight * 0
        End If

        CalculateFloodHeight = futureHeight
End Function
```

The resulting dataset comprised over 15,000 records, containing features such as Initial Flood Height (H_init), Rain Intensity (R_intensity), and the Predicted Height of Floodwater (Pred_H) for the label.

Data Preparation : The dataset was split into two features, H_init and R_intensity, which represent initial height and rain intensity, respectively, and a label called Pred_H, which represents the predicted height of the rainfall levels. After being converted into Numpy arrays, the dataset is divided into training and testing subsets, with the training subset comprising 80% of the entire population and the testing subset being 20% of it.

Machine Learning Algorithms : Floodwater level prediction was done using a range of machine learning (ML) algorithms, such as Gradient Boosting Regression, XGBoost Regressor, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Linear Regression, and Artificial Neural Network (ANN).

Model Training and Evaluation : Google Colaboratory was used to train the model, with the T4 GPU acting as the hardware accelerator and Python 3 as the runtime type. TensorFlow was used for neural networks, and scikit-learn was used to implement traditional machine learning algorithms. Once the baseline performance was determined, it could be used as a standard measure of how well future model improvements worked. Each algorithm was first trained and assessed using default parameters and configurations to produce this baseline result. A systematic approach was utilized to tune the parameters of the machine learning models to improve their performance. To find the ideal set of parameters that minimize prediction errors and increase model performance, an exhaustive search through a specified parameter grid was conducted using GridSearchCV. The machine learning models were retrained using the GridSearchCV-identified ideal parameter configurations after the parameter tuning phase was finished. Test data was utilized to evaluate the trained models, and accuracy and dependability were measured using performance metrics such as Mean Absolute Error (MAE) and Mean Squared Error (MSE). The closer the values are to zero, the more accurate the model is [9].

Model Selection: The best-performing model is chosen based on MAE over MSE because it is computationally cheap due to its simplicity and it provides an even measure of how well the model performs. It is also less sensitive towards outliers as opposed to MSE [10].

Model Deployment: The model is saved using the Joblib library, which allows for efficient serialization of Python objects. Using the Flask framework, an API is developed to deploy the model. The web-based application could communicate with the trained machine learning models via the API, which acted as an interface. The application was able to display and analyze the expected floodwater levels after the data was processed by the prediction models using the endpoints it supplied.

Internet of Things (IoT) : The Internet of Things (IoT) refers to the wireless connection of ordinary objects to the Internet [11]. It is a fundamental technology utilized in early warning systems for flooding. IoT features offer a reliable assurance for proactive awareness and readiness, contributing to the reduction of catastrophic effects. IoT technologies are a very useful tool for the transmission of disaster preparation data, even though they cannot prevent disasters from happening. Such information can be utilized to create geographic flood simulation models, which support the development of flood catastrophe risk management policies [12].

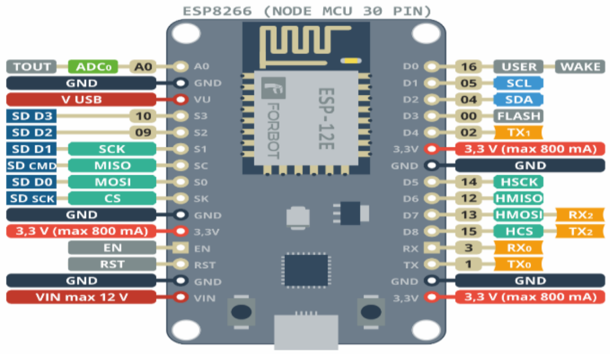

Fig. 1: ESP8266 NodeMCU 30 PIN

The ESP8266, as shown in Figure 1, is known for its low-cost and integrated Wi-Fi capabilities, is widely utilized in IoT projects. In this study, the researchers investigate the specific pins on the ESP8266 that facilitate communication with these sensors. The ultrasonic sensor utilizes pins D1, D2, G, and 3V, while the rain sensor employs pins AO, G, and VU for communication. The ESP8266 microcontroller uses its GPIO (General Purpose Input/Output) ports to interface with sensors, such as rain and ultrasonic sensors. These pins can be set up to receive data (input) or send signals (output). For example, when using an ultrasonic sensor, the ESP8266 triggers the sensor to create ultrasonic pulses by sending signals through particular GPIO pins (D1, D2). The ESP8266 receives these pulses through its input pins after they bounce off objects and return to the sensor as electrical signals. Similarly, The ESP8266 receives an analog signal from the rain sensor proportionate to the intensity of the rainfall. By reading this signal through its analog input pin, the ESP8266 can identify whether rain is occurring or not.

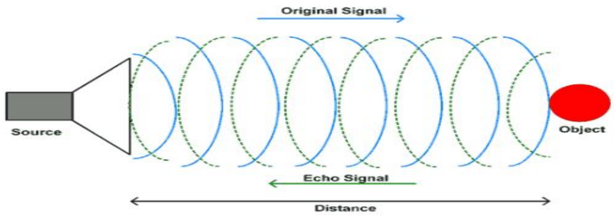

Fig. 2:Ultrasonic Sensor Operation

A transmitter that produces ultrasonic waves is how an ultrasonic sensor works as depicted in Figure 2. These waves move through the atmosphere until they come into a barrier. The waves are then reflected to the sensor's receiver. The sensor determines the distance to the object by measuring how long it takes for a wave to make a full circle. The airspeed at which sound travels must be taken into account when measuring distance. Typically, the sensor outputs an electrical or digital value that indicates the calculated distance. This output can be used for several purposes, including determining whether or not objects are present.

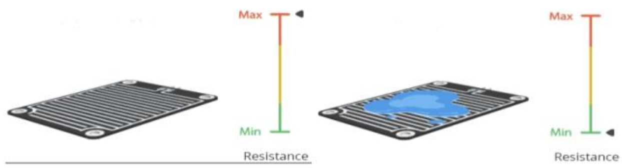

Fig. 3: Rain Sensor Operation

The rain sensor works by using a sensing pad that has exposed copper traces, which act as a potentiometer or variable resistor. The amount of water on the sensor pad's surface affects how resistant it is. More water leads to better conductivity and lower resistance, while less water results in worse conductivity and higher resistance. This is an inverse connection. As a result, resistance fluctuations based on the amount of water sensed on the detecting pad essentially dictate the sensor's output.

Hardware Requirements: The widely used ESP8266 WiFi module serves as the foundation for the NodeMCU open-source development board and firmware. It allows you to program the ESP8266 WiFi module using the Arduino IDE. This study used this module as depicted in Figure 4.

Fig. 4: Rain Sensor Operation

Ultrasonic sensors work via wave ultrasonics. Ultrasonic waves are defined as waves that have frequencies higher than sound waves. The ultrasonic sensor employed in this study was the JSN-SR04T module, which is shown in Figure 5.

Fig. 5: JSN-SR04T Ultrasonic Sensor

Rain-detecting devices are made of rain sensor modules. A raindrop can function as a switch when it strikes the rainboard. It can also be used to measure the strength of the rain. The rain sensor used in this investigation is the HL-83 module, which is shown in Figure 6.

Fig. 6: HL-83 Rain Sensor

Additional hardware were also utilized for the completion of the prototype. To connect each part of the system to the others, jump wires or dupont wires were used. With a 30000mAh capacity and a 5-volt output, the BAVIN PC089 Power Bank served as the nodeMCU's power supply. It is important to note that the minimal specifications call for a battery with a 5-volt output and a minimum capacity of 1000mAh.

Fig. 7: Dupont Wires (Left) and BAVIN PC089 Power Bank (Right)

These elements were carefully chosen and set up to allow for the real-time capture of rainfall and water level data, which is then stored in Firebase for additional analysis and model improvement.

II. RESULTS AND DISCUSSION

Web Development

Fig. 8: Dashboard

Figure 8 shows the dashboard wherein the users will see which areas are in normal, warning, and dangerous conditions based on the flood water level. When the water level reaches a certain threshold, the user can send an SMS alert notification to the residents for evacuation purposes. This also shows the visualization of the water prediction and the rain analog.

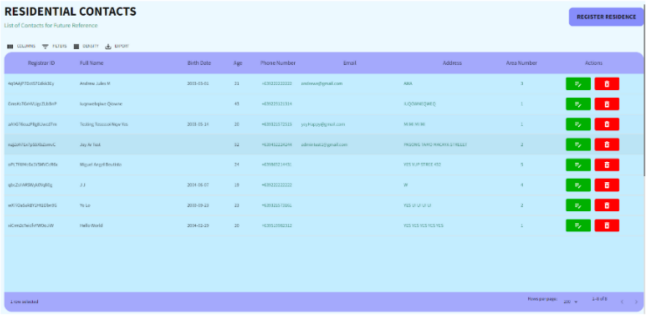

Fig. 9: Residential Contacts

Figure 9 shows the residential contact information such as their name, birth date, age, phone number, email, address, and area number.

Fig. 10: Manage Team

Figure 10 shows the team members and their access roles.

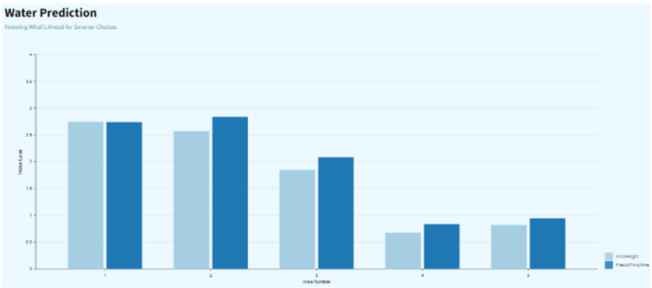

Fig. 11: Water Prediction Visualization

Figure 11 shows the bar graph of water levels at the initial height and the predicted water level at 30-minute intervals.

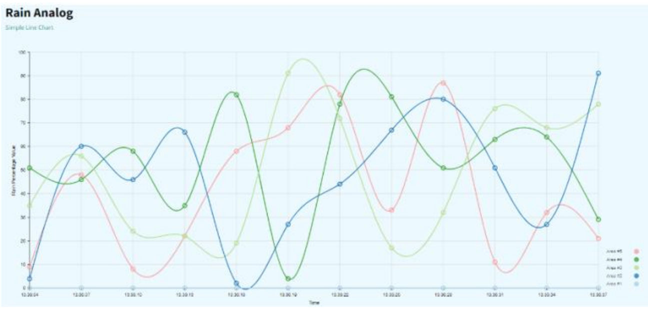

Fig. 12: Rain Analog Visualization

Figure 12 shows the estimated volume of rainwater obtained from the sensor through a line chart.

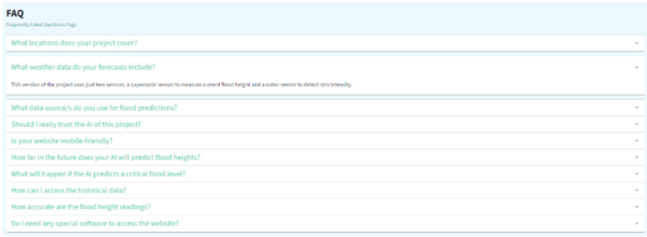

Fig. 13: Frequently Asked Questions (FAQs) Page

Figure 13 shows the frequently asked questions about the system for the users' reference.

Machine Learning

Table 1

Baseline Performance of Different Machine Learning Algorithms

ML ALGORITHM	MAE	MSE
Linear Regression	0.006753632766555791	0.0007935518959850468
Gradient Boosting Regression	0.007506282542510411	9.996147120212344e-05
XGBoost Regressor	0.005934867885087615	8.643463286613198e-05
K-Nearest Neighbors	0.03385379999999999	0.0024829657199999995
Support Vector Machine	0.1154198189878486	0.019929332060531785
Artificial Neural Network	0.975300133228302	1.3359211683273315

Table 1 shows the results of training the baseline model showing that different machine learning algorithms perform differently when it comes to forecasting floodwater levels. With the lowest Mean Absolute Error (MAE) of roughly 0.0059 and Mean Squared Error (MSE) of roughly 8.643e-05, XGBoost Regressor stood out for its exceptional performance. This suggests that it can forecast floodwater levels with accuracy.Additionally performing well and obtaining relatively low MAE and MSE values, Linear Regression and Gradient Boosting Regression demonstrated their efficacy in precisely predicting floodwater levels. K-Nearest Neighbors, on the other hand, showed greater error metrics, suggesting that their applicability to the dataset was limited.

With an MAE of about 0.1154, Support Vector Machine (SVM) performed moderately, although Artificial Neural Network (ANN) displayed the greatest error metrics, suggesting difficulties in identifying underlying patterns in the dataset.

Table 2

Improved Model Performance using GridSearchCV

ML ALGORITHM	MAE	MSE
Linear Regression	-	-
Gradient Boosting Regression	0.002055154071020467	2.4194904201522686e-05

XGBoost Regressor	0.002796881855495274 6	2.3963883134386347e-0 5
K-Nearest Neighbors	0.016328094462903047	0.0006718154349423552
Support Vector Machine	0.05064085490569372	0.003548020947470483
Artificial Neural Network	0.03896820545196533	0.01142108254134655

Table 2 shows the results following parameter optimization with GridSearchCV. The improved model performance results show significant improvements in predicted accuracy for several machine learning techniques.

When compared to their baseline performance, the Mean Absolute Error (MAE) and Mean Squared Error (MSE) values of the Gradient Boosting Regression and XGBoost Regressor showed noticeable improvements. These improvements suggest improved accuracy and dependability in floodwater level prediction, which makes them attractive options for integration into the Real-Time Flood Surveillance and Early Warning System.K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Artificial Neural Network (ANN) likewise demonstrated a discernible improvement, with a significant drop in both MAE and MSE values. This suggests that the models have increased in both generalization and predictive accuracy.

In the case of Linear Regression, since it is a straightforward technique without tunable parameters, there are no hyperparameters to improve so there is no need to apply GridSearchCV. The study found significant differences in the predictive performance of different models after thoroughly evaluating various machine learning algorithms to forecast floodwater levels in the Real-Time Flood Surveillance and Early Warning System. The results of training the baseline model showed that the XGBoost Regressor was the most accurate approach, with the lowest Mean Absolute Error (MAE) and Mean Squared Error (MSE) values. Additionally, K-Nearest Neighbors exhibited limitations in its applicability to the dataset, but Linear Regression and Gradient Boosting Regression provided promising results with relatively low error metrics. Support Vector Machine (SVM) had moderate performance, whilst Artificial Neural Network (ANN) demonstrated the greatest error metrics.Additional optimization using GridSearchCV led to notable improvements in the predicted accuracy of multiple machine-learning methods. Significant gains in MAE and MSE values were demonstrated by both Gradient Boosting Regression and XGBoost Regressor, demonstrating improved accuracy and dependability. Given these findings, the final regression model to be implemented in the system is determined to be the XGBoost Regressor as it performed exceptionally well in the baseline and enhanced model performance assessments, showing notable increases in predictive accuracy following GridSearchCV parameter modification. This model would be appropriate for incorporation into the Real-Time Flood Surveillance and Early Warning System.

Internet of Things (IoT)

Fig. 14: IoT Prototype

The prototype for flood monitoring, which combines physical components and Internet of Things technologies, is depicted in Figure 14. The NodeMCU serves as the central hub of the system, facilitating WiFi-enabled connections with external devices and organizing the collection of data from various sensors. With the help of a well-positioned ultrasonic sensor module, the JSN-SR04T accurately measures water levels and recognizes any variations that might point to rising floodwaters. The HL-83 rain sensor module simultaneously detects rainfall and measures its intensity to offer real-time precipitation data. The system uses programmed algorithms to process the data it gathers from various sensors and examine trends and patterns. These insights are sent via the NodeMCU to the Firebase web application, where machine learning algorithms can use the data for additional analysis and flood event prediction.

III. CONCLUSION

In conclusion, the Philippines is confronted with significant issues as a result of an increasing number of cyclones and extreme weather events, which worsen the issue of floods, the most common natural disaster in the country. In addition to seriously damaging property and infrastructure, floods pose a threat to human life and public health, particularly in densely populated areas like Metro Manila, particularly Valenzuela City. This study's Real-Time Flood Surveillance and Early Warning System using IoT and AI meets the growing demand for more efficient flood monitoring and response systems. By integrating IoT technology and machine learning algorithms, the system offers accurate and timely predictions of flood events, enabling authorities to take preventive measures and mitigate the effects of disasters. The web-based interface helps with coordination and decision-making during emergencies by enabling real-time display of water levels and facilitating contact with stakeholders. Following a thorough testing and evaluation process, the findings demonstrated that the XGBoost Regressor outperformed the baseline and upgraded model assessments. This model demonstrated significant improvements in projected accuracy after parameter modification, making it the preferred model for the flood monitoring system. Moreover, the system's ability to dependably record rainfall and water level data in real-time was made possible by the NodeMCU, rain sensor module (HL-83), ultrasonic sensor module (JSN-SR04T), and other IoT hardware components that played a key role in data transmission and collection.

ACKNOWLEDGEMENTS

The researchers would like to express their sincere gratitude to the university's engineering professors for their helpful advice and assistance during this project. In particular, the researchers would like to thank Engr. Jeffrey Leonen and Engr. Delaney Ofrecio for their significant contributions and expertise, which greatly enhanced the development and implementation of the Real-Time Flood Surveillance and Early Warning System. Their advice and support were crucial to this project's successful conclusion. The researchers appreciate your hard work and devotion to our project.

REFERENCES

[1] United Nations Climate Change, "Urban Flood Monitoring - Philippines," United Nations Climate Change, https://unfccc.int/climate-action/momentum-for-change/activity-database/urban-flood-monitoring (accessed Mar. 29, 2024).

[2] World Health Organization, "Floods," World Health Organization, https://www.who.int/health-topics/floods#tab=tab_1 (accessed Mar. 29, 2024).

[3] R. S. Bacero and A. M. Fillone, "Impact of flooding on truck movement in Metro Manila, Philippines," Asian Transport Studies, vol. 9, p. 100113, Jun. 2023. doi:10.1016/j.eastsj.2023.100113

[4] PAGASA, "Floods," PAGASA, https://www.pagasa.dost.gov.ph/learning-tools/floods (accessed Mar. 29, 2024).

[5] Climate & Development Knowledge Network, "Building resilience to climate change locally: The case of Valenzuela City, Metro Manila," INSIDE STORIES on climate compatible development , https://cdkn.org/sites/default/files/files/Valenzuela-Inside-Story.pdf (accessed Mar. 29, 2024).

[6] GMA Integrated. News, "Parts of Valenzuela remain flooded," GMA News Online, https://www.gmanetwork.com/news/topstories/metro/877562/parts-of-valenzuela-remain-flooded/story/ (accessed Mar. 29, 2024).

[7] ScienceDirect, "Machine learning," Machine Learning - an overview | ScienceDirect Topics, https://sciencedirect.com/topics/computer-science/machine-learning (accessed Mar. 29, 2024).

[8] V. Plevris, G. Solorzano, N. Bakas, and M. Ben Seghier, "Investigation of performance metrics in regression analysis and machine learning-based prediction models," 8th European Congress on Computational Methods in Applied Sciences and Engineering, Jun. 2022. doi:10.23967/eccomas.2022.155

[9] M. Kinoshita and Y. Shiomi, "Data-driven modeling of car-following behavior on freeways considering spatio-time effects: A comparison of different neural network structures," International Journal of Intelligent Transportation Systems Research, vol. 21, no. 1, pp. 86–98, Mar. 2023. doi:10.1007/s13177-022-00339-9

[10] A. Jadon, A. Patil, and S. Jadon, "A Comprehensive Survey of Regression Based Loss Functions for Time Series Forecasting," Arxiv, https://arxiv.org/pdf/2211.02989.pdf (accessed Mar. 29, 2024).

[11] J. Edwards, "Signal Processing opens the internet of things to a new world of possibilities: Research leads to new internet of things technologies and applications [special reports]," IEEE Signal Processing Magazine, vol. 35, no. 5, pp. 9–12, Sep. 2018. doi:10.1109/msp.2018.2847567

[12] E. Samikwa, T. Voigt, and J. Eriksson, "Flood prediction using IOT and artificial neural networks with Edge Computing," 2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics), Nov. 2020. doi:10.1109/ithings-greencom-cpscom-smartdata-cybermatics50389.2020.00053