Prototype Design and Development of IoT and Machine Learning Based Flood Monitoring, Alerting and Forecasting System

[1]Ali Hamed Ali Hamed Al Maqbali
[2]Muhammad Nauman Bashir, https://orcid.org/0000-0002-2721-3490
[3]Sania Nauman
[1,2]Department of Computing and Electronic Engineering, Middle East College, Muscat Oman
[3]Department of Electrical and Computing Engineering, Comsats University Islamabad Lahore Campus, Pakistan

Corresponding Author: *Ali Hamed Ali Hamed Al Maqbali (email: 20F20169@mec.edu.om)

***Abstract*—Floods are among the most common natural disasters worldwide, affecting every aspect of society. In recent years, research has focused on developing models to predict floods. Few models have proposed an integration of alert aid mechanisms for risk mitigation, policy recommendations, and reducing fatalities and property damage caused by floods. However, predicting and mitigating floods is complex, requiring thorough research on the contributing factors. This research work has designed and developed an Internet of Things based flood status evaluation prototype with alert mechanism along with using a prediction model to predict floods using machine learning so that the authorities could take necessary steps to avoid the damage due to flood and do necessary managements. The machine learning algorithm streamlines the data collection and data process. The implementation results show that the objectives are well accomplished, and the proposed model effectively completes data collection and prediction tasks.**

Keywords*— *Flood prediction, Internet of Things, Risk mitigation, Data collection, Machine Learning Model Implementation

I. INTRODUCTION

Significant casualties can be caused by the flood disasters that naturally occur worldwide [1]. As temperatures are rising due to global warming, it is leading to increased evaporation from both land and oceans, resulting in alterations in the frequency and intensity of heavy precipitation events that could subsequently impact the occurrence and scale of river flooding. Developing an accurate flood forecasting and prediction model is crucial for reducing damage and minimizing the number of victims, especially in this era of global warming. Rain forecasting aids in water resource allocation, proper management, planning, flood warnings, and flood damage mitigation [2]. The goal of this project was to create a flood warning system to notify authorities and reduce risks for the general public, contributing to the development of smart cities and supporting Oman's Vision 2040 for greater disaster resilience [3]. This experimental research work aims to develop a Flood Monitoring, Alerting and Forecasting System using

Machine Learning (ML) algorithms. This initiative holds substantial potential as Oman continues to grow and compete globally. Currently, Oman lacks a systematic deployment of such systems, which will drive high demand. The system is a prototype which is designed with cost-effective components such as Arduino boards, ultrasonic sensors, water level sensors, and a Wi-Fi module to simulate the flood warning systems. The system operates by utilizing sensors to monitor rising water levels due to rainfall. The Arduino board gathers data from these sensors and transmits it to the Cloud via the Wi-Fi module. When the water levels surpass a predetermined threshold, the system sends a signal to the authorities which can enable them to promptly alert the public about the impending flood, as illustrated in the accompanying figure 1. This approach not only ensures timely warnings but also promotes broader implementation due to its affordability [4].

Fig. 1. Flood Affecting Omani Village [4]

Machine learning, a branch of artificial intelligence (AI), is dedicated to developing data-driven computer systems that improve their performance over time. ML techniques allow software to enhance its capabilities by collecting and processing data. Initially, data is converted from analog to digital format and then analyzed by ML algorithms to identify critical conditions. In the final stage, this data is transmitted to the core controller, which selects the most accurate classifier. Once high-resolution data is obtained, it can be monitored and managed remotely from any location with GSM service [5]. Further, the goal of this project was to develop a system for monitoring, alerting, and forecasting floods by leveraging the Internet of Things (IoT) to provide advance warnings [6]. The project objectives include: collecting data using microcontrollers and water level sensors in flood-prone areas, transmitting water level and location information to authorities via GSM, displaying alarms on an LCD in the control room, using motors to open reservoir gates based on the received information, and processing sensor data over time with machine learning algorithms to determine flood intensity and patterns. Section II discusses the related theory and methodology adopted for this research. Section III covers the project designs, Section IV presents the results of the prototype development, and Section V concludes the discussion.

II. LITERATURE REVIEW

The development of the project's embedded system follows a typical process [7]. The V-methodology is selected for its straightforward and sequential approach, which aligns well with the project requirements [8]. The stages of the V-methodology include requirements, design, implementation, verification, and maintenance. This model is well-organized and progresses linearly, similar to the waterfall model, with each phase commencing only after the previous one is completed. What sets the V-methodology apart is that testing activities, such as planning and design, occur before coding begins. The model illustrates connections between phases, starting with verification stages (requirements collection, system design, architecture, and form design) and followed by validation stages (unit testing, integration testing, system testing, and acceptance testing). This approach enables early testing, helping to identify any project flaws or errors at an early stage.

The authors in [9] proposed a flood monitoring system that leverages IoT and AI technologies to predict floods before they occur, using the BMP180 sensor. This innovative system employs an Arduino Uno microcontroller to detect flooding by monitoring changes in water level, humidity, air pressure, and temperature. A Raspberry Pi, combined with the YOLO object detection algorithm, conducts rescue missions via drones, identifying trapped individuals and alerting authorities through a web portal and mobile application. The design is cost-effective, easy to replicate, and operates fully autonomously without human intervention. It provides accurate readings with the SRF-05 ultrasonic sensor and uses an LCD display for local alerts. However, this design has limitations in remote status monitoring due to the need for closely positioned components, which restricts connectivity. Additionally, the servo motor used for gate control allows only 180-degree rotation. The project could be improved by implementing long-range communication between components, using a Wi-Fi-enabled Arduino for better connectivity, and enhancing the servo motor's rotation to 360 degrees by integrating TIP parts and diodes.

The literature includes a design by [10] which developed a project aimed at mitigating flooding impacts in Chennai, India, by alerting authorities and the public through a government-controlled webpage. This battery-operated system comprises a relay driver circuit, sensor circuit, controller circuit, and SMS circuit. Ultrasonic sensors placed at various locations monitor water levels, sending data to a microcontroller, which then transmits the information to the cloud via a Wi-Fi module. The system functions as an alert mechanism and water level monitor, featuring affordable construction, easy maintenance, and accurate circuit operation, as confirmed by sample results. However, the design has some limitations: it does not prevent vehicles from entering flooded areas, the specific components used are not disclosed, and the relays may not be optimal for circuit control. Improvements could include mounting water level sensors on light poles to operate lever gates, adding warning lights and sirens to alert drivers, and using an Arduino for simpler circuit programming.

Another design by [11] is the "Arduino Flood Detector System," created to assist commuters and vehicles during floods. The system activates when an ultrasonic sensor detects water on the roads and sends SMS alerts to residents, continuously updating the water level on the user interface. Its main benefits include displaying the water level on the user interface, automatically sending SMS alerts, and providing continuous updates until water levels normalize. However, it has some shortcomings: the ultrasonic sensor can be affected by non-water elements during floods, and it requires a reliable server and internet connection, which may be disrupted during floods. Improvements could be made by integrating a Wi-Fi module with the Arduino for better connectivity and adding advanced features like automatic gate control based on water levels to notify authorities more effectively

The project designed by [12] proposed a flood monitoring system using IoT and Machine Learning to predict floods before they occur. This system uses an Arduino Uno microcontroller to detect flooding based on sensor readings of water level, humidity, atmospheric pressure, and temperature. An object detection algorithm notifies authorities via web and mobile applications. The project emphasizes the use of Machine Learning and AI for reliable flood prediction and employs neural networks for rainfall prediction. It offers advanced monitoring of abnormal and normal machine behaviors and provides a comprehensive survey and analysis of flood-related issues.

III. SYSTEM DESIGN

The system is initiated by the water sensor, activation button, control unit, and other components. Figure 2 presents a block diagram of the system, offering a high-level overview of its primary elements and their connections. It illustrates the interactions within the Flood Monitoring, Alerting, and Forecasting System. At the core of this system is the Arduino

Uno, which manages operations. The setup is divided into inputs and outputs. Inputs include a 7.4V power supply that energizes the entire system, including the Arduino and sensors; a water level sensor that measures water heights to inform alerts and forecasts; and a GPS module that provides locational data, potentially tracking flood locations. Outputs consist of six electronic components controlled by the Arduino, including a database that stores environmental and water level data to support historical analysis and machine learning model training.

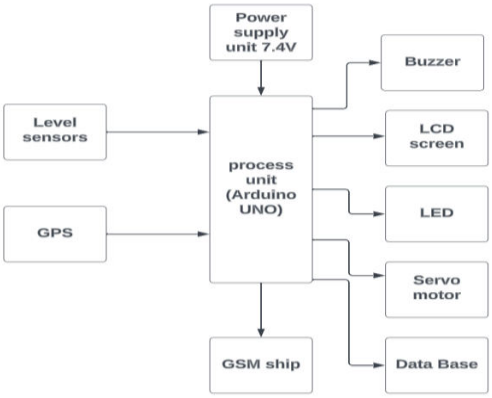


Fig. 2. System Block Diagram

Figure 3 shows the system flowchart, detailing the sequential processes and steps involved in the operation. It begins with the level sensor reading water levels and using a specific algorithm to analyze the data. Alerts are displayed on an LCD screen in the monitoring room, and data is periodically recorded to help predict potential floods. When the sensor detects a certain water level, it sends a signal to the microcontroller on the Arduino board. This activates the control unit, which commands a motor to light up a danger lamp (LED) and, if necessary, orders the opening of the dam gate.

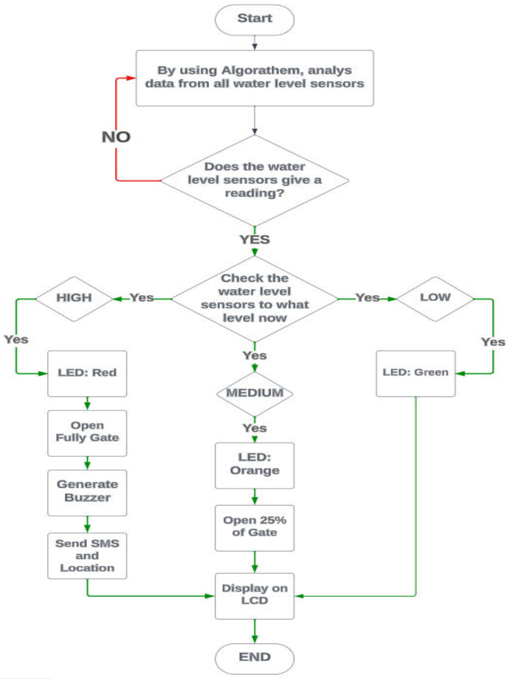


Fig. 3. System Flow Chart

The Arduino Uno excels in hobbyist projects and educational settings, allowing users to explore the realms of electronics and programming with ease. Its accessibility and extensive support community have made it a favorite among enthusiasts and professionals alike, fostering an environment where creators can develop and share a plethora of digital projects and applications. Additionally, the integration with various sensors and modules amplifies its utility in complex systems, from simple LED displays to more sophisticated robotic controls, underscoring its role as a fundamental component in the toolkit of modern developers as shown by the figure 4.

The Arduino Uno, renowned in microcontroller-based applications, is prized for its versatility and open-source nature, powered by the ATmega328P. This board is indispensable for prototyping, boasting 14 digital I/O pins and 6 analog inputs. It supports PWM output and features a USB interface for seamless programming. In the proposed project, the Arduino IDE, an open-source integrated development environment, will be employed to program Arduino boards equipped with water level sensors, utilizing version 1.8.19. Tinkercad, a user-friendly and free 3D modeling software from Autodesk, will assist in creating and simulating circuit diagrams. It provides access to pre-designed circuits and essential code snippets for straightforward manipulation.

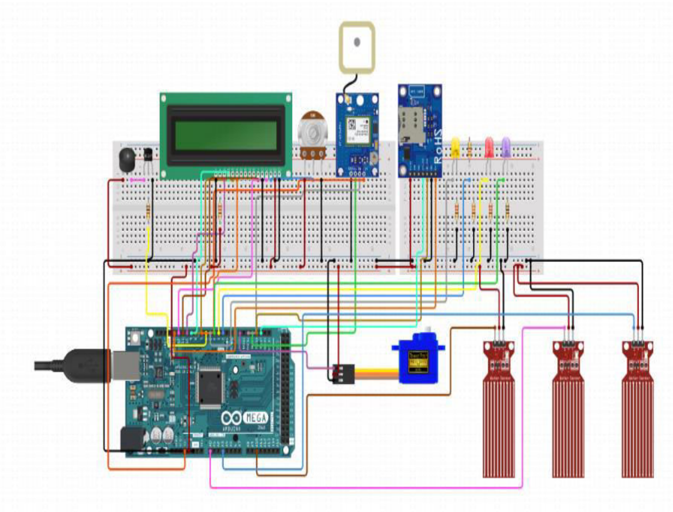


Fig. 4. System Schematic Diagram

Fritzing adheres to Arduino principles, enabling users to design and document Arduino-based prototypes. It facilitates the creation of manufacturing-ready circuit diagrams and supports code development. In the context of machine learning with Python, data from Excel files will undergo preprocessing using pandas to structure and prepare it for machine learning tasks. Features will be selected and standardized to train models such as the Random Forest Classifier, with performance evaluation aimed at effectively forecasting flood risks. This integrated approach combines software tools and machine learning techniques to predict flood events, enhancing the Flood Monitoring, Alerting, and Forecasting System with real-time data analysis and predictive alerts. It leverages established methods for data collection and preprocessing, alongside Python for computational logic.

IV. SYSTEM TESTING

This section details the circuit setup featuring an Arduino UNO R3, a water level sensor, an LED module, and a servo motor. The design was created using Proteus 8 professional simulation software, which supports IoT technology simulations. Three tests were conducted on the system components, and simulations were performed using the Tinkercad program. These tests involved an Arduino UNO, a water level sensor, an LED module, and a servo motor. The system was powered via a USB connection and an external source, such as a PC or a 5-volt battery. Additionally, ultrasonic sensors were tested with three LEDs to assess their capability in measuring water distances. The LEDs provided visual feedback based on sensor readings, following the methodology described in [13].

Value testing of the design ensures that the device operates within specified maximum voltage limits outlined in the datasheet. Simulation testing involves integrating a DC voltmeter in Proteus software to measure voltage and validate system functionality. The system test integrates an Arduino, water level sensor, and LED module. The water level sensor monitors water levels, prompting the Arduino to control LED and servo motor responses based on predefined water level thresholds. In software simulations, an Arduino UNO R3, water level sensor, LED module, and servo motor are utilized, powered either by USB or an external battery, to simulate IoT applications using Proteus 8 and Tinkercad programs, as depicted in the accompanying figures.

During this phase of the project, the focus is on executing the project plan and finalizing deliverables, including essential items such as equipment, resources, and materials. The primary objective is to position a water level sensor atop a miniature hill above a tank to monitor rising water levels. The hill structure is constructed from clay and polystyrene mold, simulating a mountainous cliff-side. A polystyrene mold is placed within a plastic container at the tank's center to prevent overflow. These components rest on a flat polystyrene sheet, supported by a cylindrical column made from sturdy cardboard, creating two levels: a surface with a miniature terrain hill and a hollow, mountain-shaped lower portion with dimensions of 100mm height and an 80mm opening box. Water is poured into this box, and when it reaches 80mm, the sensor detects the change, triggering water flow beneath the mountain structure.

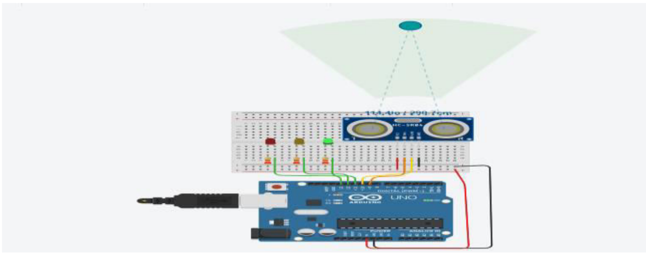


Fig. 5. Low Level

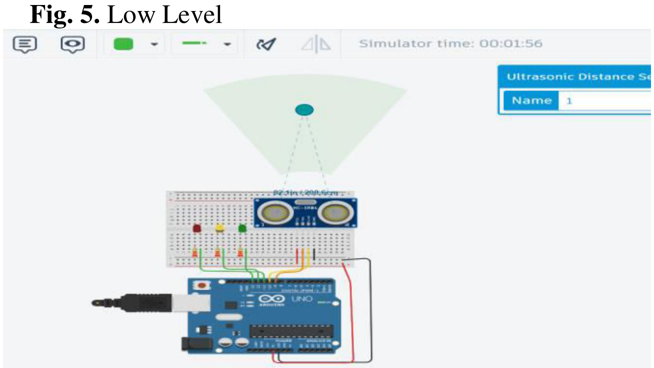


Fig. 6. Medium Level

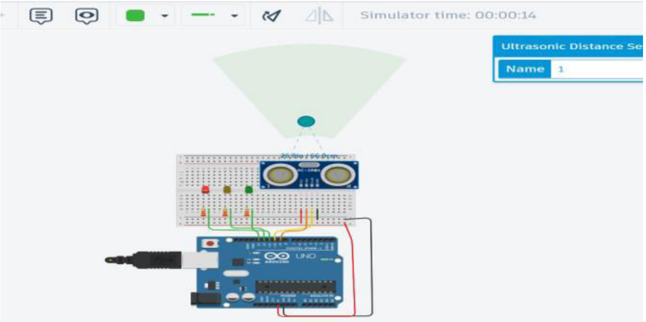


Fig. 7. High Level

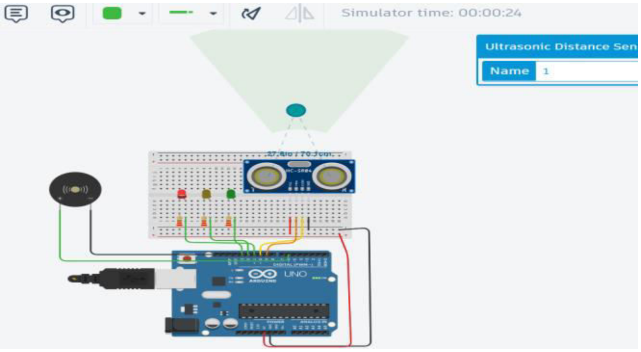


Fig. 8. Buzzer Test

Fig. 9. Ultra Sonic Sensor with Motor

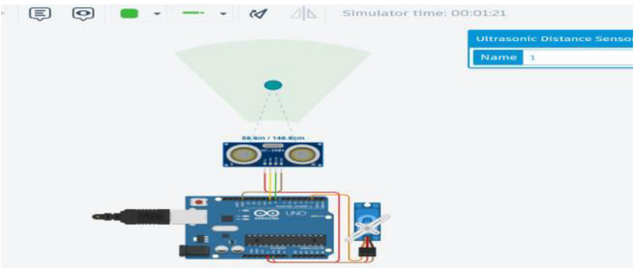


Fig. 10. 25% Gate Opening

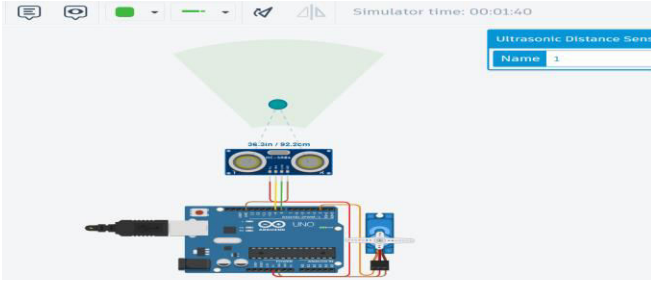


Fig. 11. Full Gate Open

While real-time implementing using discrete components, the project is segmented into two parts: the first focuses on integrating the water level sensor, while the second involves adding aesthetic elements such as miniature palm trees, rocks, and sand to replicate the Omani environment [14]. A plastic film shields the central path from water, ensuring the sensor's operation as water moves from the first to the second segment. A crucial aspect of this setup is a network of black foam plastic pathways with miniature light poles, where the sensor activates upon water crossing. To ensure all components, including the Arduino Uno, LED module, and buzzer, function correctly before final assembly, a breadboard is used for initial testing and validation. This step is essential before proceeding to soldering and integrating these components into the project.

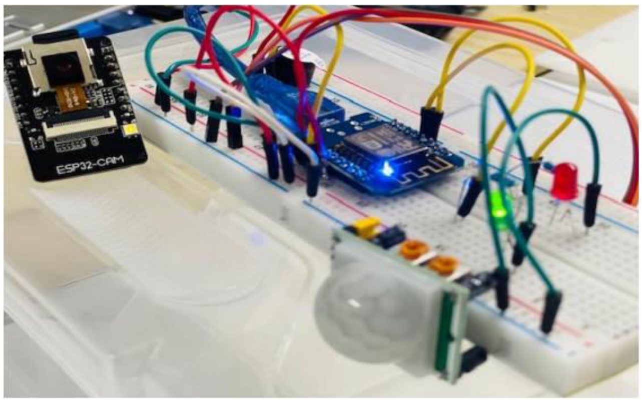


Fig. 12. Implementation using breadboard

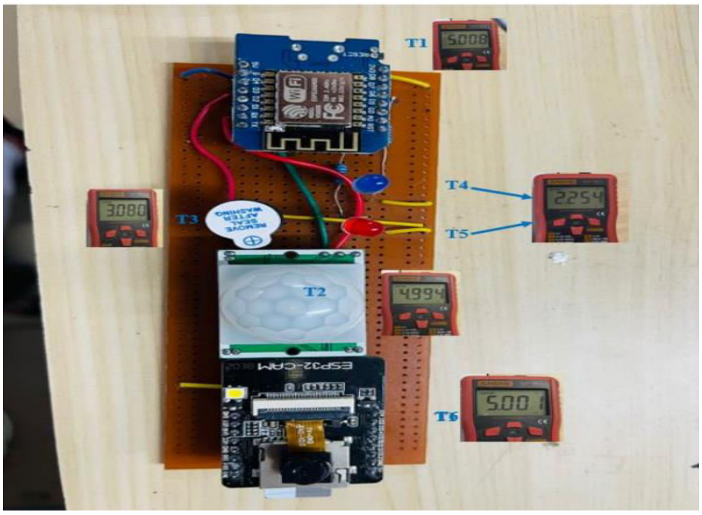


Fig. 13. Circuit Design

Once the functionality of the system is validated with the prototype board as per the design plan, the system components are soldered onto the printed circuit board and connected to their respective ports or ground. Hardware simulation can be conducted using simulation software or hardware simulation platforms, which emulate physical components within a virtual environment. This process typically involves the following steps:

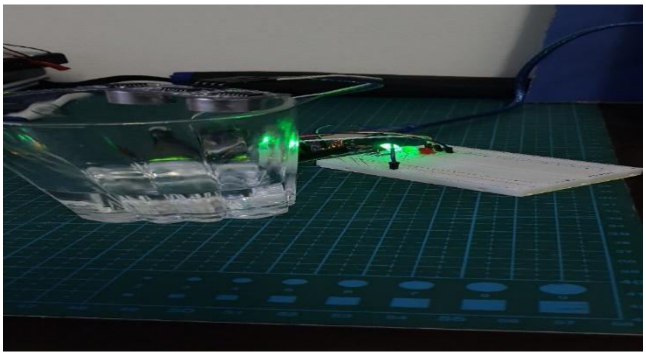


Fig. 14. Green LED for Low Level

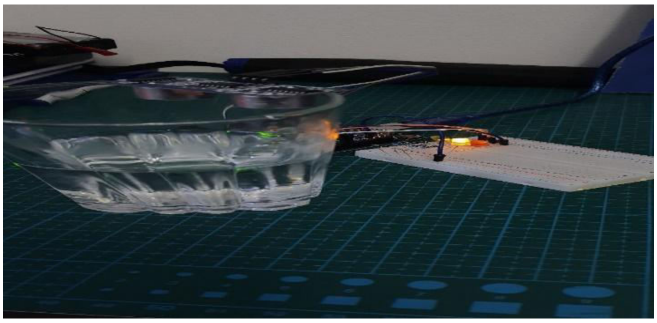


Fig. 15. Amber LED for Medium Level

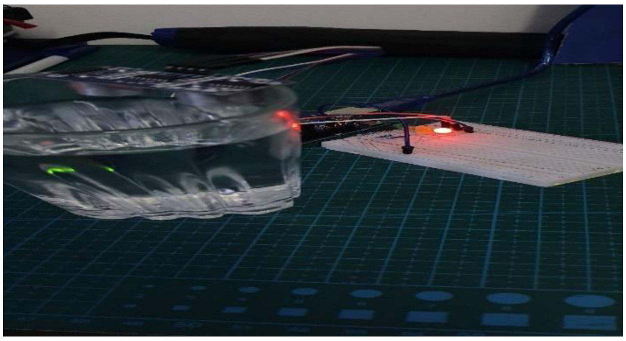


Fig. 16. Red LED for High Level

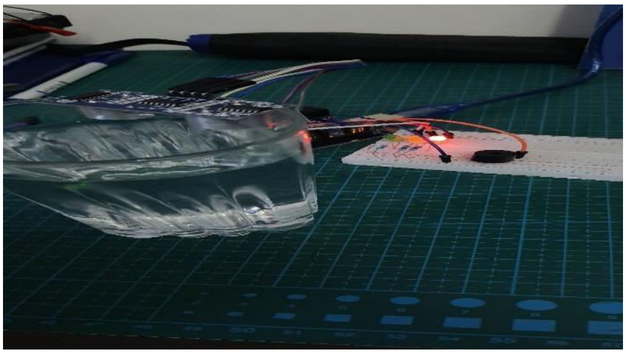


Fig. 17. Red LED for High Level and Buzzer

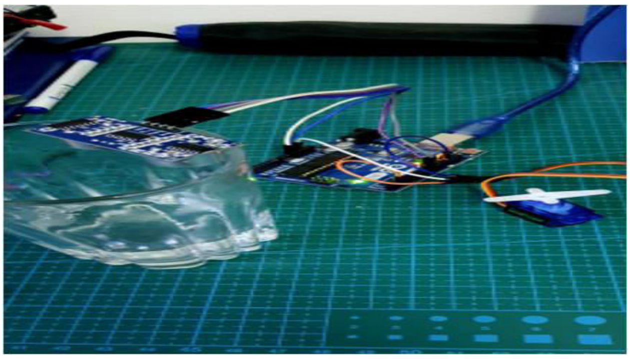


Fig. 18. Servo Motor Working for Gate Operation for 25%

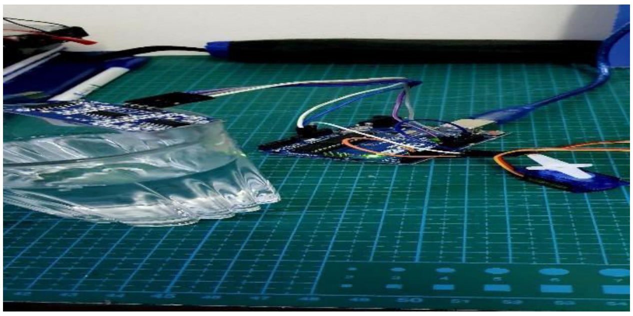


Fig. 19. Servo Motor Working for Gate Operation for 100%

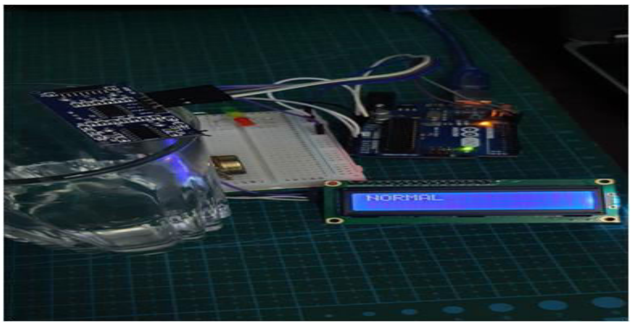


Fig. 20. LCD status display

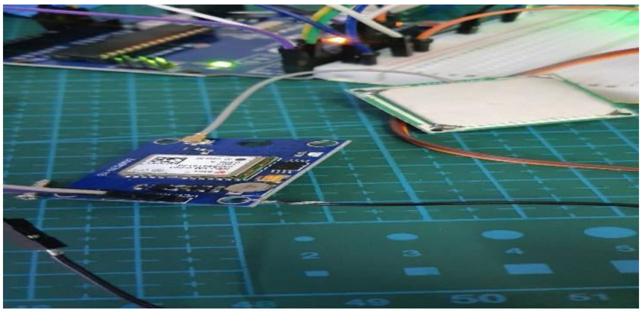


Fig. 21. GPS Operation

A notification was delivered from the control room to the phone indicating that the simulation had successfully concluded.

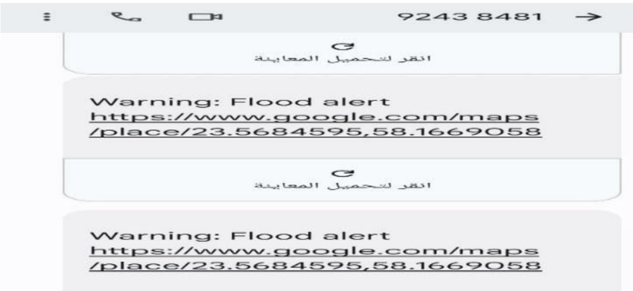
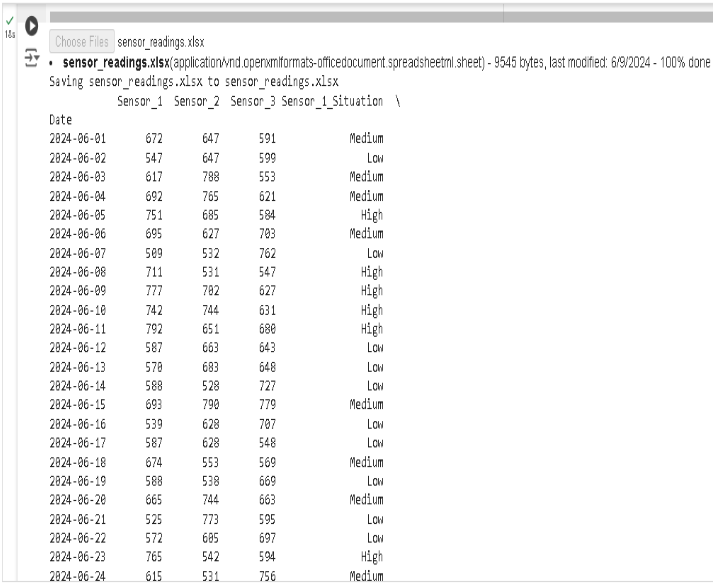


Fig. 22. IoT Blynk App

Fig. 23. Model for the Designed System

The Python script processes sensor data from 'Generated_Ali_projects_data.xlsx' by loading it into a pandas DataFrame and displaying initial rows. It includes functions to categorize sensor status ('Sensor1', 'Sensor2', 'Sensor3') as 'Normal', 'Average', 'Dangerous', or 'Mixed' and assigns corresponding actions such as adjusting lamps and alarms. The modified data is saved to 'Analyzed_Ali_projects_data.xlsx' and provides a download link.



```
Choose Files  sensor_readings.xlsx
• sensor_readings.xlsx(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 9545 bytes, last modified: 6/9/2024 - 100% done
Saving sensor_readings.xlsx to sensor_readings.xlsx
            Sensor_1  Sensor_2  Sensor_3 Sensor_1_Situation  \
Date
2024-06-01       672       647       591             Medium
2024-06-02       547       647       599                Low
2024-06-03       617       788       553             Medium
2024-06-04       692       765       621             Medium
2024-06-05       751       685       584               High
2024-06-06       695       627       703             Medium
2024-06-07       509       532       762                Low
2024-06-08       711       531       547               High
2024-06-09       777       702       627               High
2024-06-10       742       744       631               High
2024-06-11       792       651       680               High
2024-06-12       587       663       643                Low
2024-06-13       570       683       648                Low
2024-06-14       588       528       727                Low
2024-06-15       693       790       779             Medium
2024-06-16       539       628       707                Low
2024-06-17       587       628       548                Low
2024-06-18       674       553       569             Medium
2024-06-19       588       538       669                Low
2024-06-20       665       744       663             Medium
2024-06-21       525       773       595                Low
2024-06-22       572       605       697                Low
2024-06-23       765       542       594               High
2024-06-24       615       531       756             Medium
```

Fig. 23. Machine Learning Model Output

V. RESULTS & DISCUSSION

This section critically addresses system issues such as component overheating and malfunctions, underscoring the importance of following project design protocols for ensuring successful operation. Verification through simulation software confirms project integrity prior to soldering. The water level and ultrasonic sensors perform as expected, consistent with findings from Al-Saffar & Ercelebi's study [15], which anticipated SMS alerts and screen displays. However, the Arduino water level sensor's performance was inadequate despite efforts to integrate email notifications. Enhancing sensor accuracy is critical for reliable water level measurements, aligning closely with project goals [16]. Discrepancies in budgeting from local component purchases highlight the advantages of online sourcing, which allows for easier returns of defective items and minimizes financial risks. Future iterations aim to enhance traffic safety using innovative microcontroller-based solutions. Thorough exploration of traffic scenarios and creative utilization of components will drive project refinement, enhancing functionality and cost-effectiveness.

VI. CONCLUSION

The design and development of this IoT and Machine Learning-based Flood Monitoring, Alerting, and Forecasting System shows great potential for enhancing real-time flood management and ensuring public safety. Its modular design improves cost-effectiveness and predictive accuracy. The system efficiently monitors water levels and enables swift responses by seamlessly integrating IoT devices, machine learning algorithms, and real-time alert mechanisms. Machine learning plays a pivotal role in data analysis and flood prediction, contributing to precise real-time forecasts. Post-implementation, considerations such as energy consumption, data security, and seamless integration of machine learning models are crucial which will worked on in our upcoming research work. Regular updates and testing during the implementation phase ensure user-friendly interfaces and enhanced system reliability. Embracing advanced technologies for flood prediction and mitigation underscores a commitment to innovative solutions for environmental monitoring and disaster management. Future enhancements could include additional alerting methods, redundant sensors, and improved utilization of environmental data for more accurate predictive capabilities and these advancements aim to strengthen the system's effectiveness in flood management and mitigation.

VII. REFERENCES

[1] C. Prakash, D. Acharya and A. Barthwal, "Hydro-Meteorological Flood Data Sensing, Prediction and Classification using Internet of Things," in *2023 IEEE Region 10 Symposium (TENSYMP), pp.1-6, 2023.*, Canberra Australia, 2023.

[2] I. M.Hayder, T. A. Al-Amiedy, W. Ghaban, F. Saeed, M. Nasser, G. A. Al-Ali and H. A. Younis, "An Intelligent Early Flood Forecasting and Prediction Leveraging Machine and Deep Learning Algorithms with Advanced Alert System," *Processes,* vol. 11, no. 2, 2023.

[3] i. i. l. t. t. O. g. w. b. f. w. y. a. h. p. i. r. b. Muscat: Judging from footage shared on social media of the devastation unleashed by flash floods across many parts of the Sultanate of Oman, "Managing Flood Risks a New Perative for Oman," Oman Observer, 24 4 2024. [Online]. Available: https://www.omanobserver.om/article/1152322/business/markets/managing-flood-risks-a-new-imperative-for-oman. [Accessed 5 5 2024].

[4] Telegrafia, "Early Flood Warning Systems," Telegrafia, [Online]. Available: https://www.telegrafia.eu/en/solution/mass-public-warning/flood-warning-system/?gad_source=1&gclid=CjwKCAjw-O6zBhASEiwAOHeGxaUv-gohUIKoUYrDO03h-8DI9a6lHPbfvYBt_vtVsKPCrUe_C8bFTxoCLfEQAvD_BwE. [Accessed 12 10 2022].

[5] D. A. Bishop, A. P. Williams, R. Seager, A. M. Fiore, B. I. Cook, J. S. Mankin, D. Singh, J. E. Smerdon and M. P. Rao, "Investigating the Causes of Increased Twentieth-Century Fall Precipitation over the Southeastern United States. J. Climate," *AMS Journals,* vol. 32, p. 575–590, 2019.

[6] Z. M. Mahdi, G. M. Aditya, I. W. Putro, B. A. Pramudita and D. P. Setiawan, "Internet-of-Things-Based Rain Detection Device Using Machine Learning Classification for Rain Prediction at Telkom University," in *2023 International Conference on Data Science and Its Applications (ICoDSA)*, Bandung Indonesia, 2023.

[7] I. Y. S. H. A. Habsi, M. N. Bashir and R. Asuncion, "Cooling Treatment After Cardiopulmonary Resuscitation," *Journal of Big Data and Smart Cities ISSN: 2706-7912 ,* vol. 3, no. 3, pp. 22-27, 2023.

[8] ProgPtr, "The V-Cycle Model," ProgPtr, [Online]. Available: https://progptr.com/site/?id=V-Cycle%20Model&ppid=v-cycle-model. [Accessed 14 11 2022].

[9] A. I. Pathan, A. P. Kulkarni, N. L. Gaikwad, P. M. Powar and A. R. Surve, "An IoT and AI based Flood Monitoring and Rescue System," *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT),* vol. 9, no. 9, 2020.

[10] M. S. S. Sayyad, P. Surve, N. Shaikh, M. Gharat and P. Tambe, "IOT Based Early Flood Detection and Avoidance," *Iconic Research And Engineering Journals Publisher: IRE Journals ,* vol. 3, no. 12, pp. 50-55, 2020.

[11] E. D. Guzman, V. S. Cuadra, A. G. D. Luna and C. Villanueva, "Flood Detector System Using Arduino," *International Journal of Management and Applied Science (IJMAS),* vol. 2, no. 7, pp. 286-290, 2016.

[12] D. S. Rani, G. N. Jayalakshmi and V. P. Baligar, "Low Cost IoT based Flood Monitoring System Using Machine Learning and Neural Networks: Flood Alerting and Rainfall Prediction," in *2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, Banglore India, 2020.

[13] Z. S. O. Al-Naamani and M. N. Bashir, "Smart Beach Cleaning Robot System," *Journal of Big Data and Smart Cities ISSN: 2706-7912 2 ,* vol. 2, no. 2, pp. 5-10, 2023.

[14] O. M. J. Al-Amri, M. N. Bashir and S. Iqbal, "Smart Monitoring System for Sea Diver's Health and Location Conditions," *Journal of Big Data and Smart Cities ISSN: 2706- 7912 ,* vol. 2, no. 1, pp. 6-11, 2022.

[15] A. Djalilov, E. Sobirov, O. Nazarov, S. Urolov and I. Gayipov, "Study on automatic water level detection process using ultrasonic sensorc," in *3rd International Conference on Energetics, Civil and Agricultural Engineering, IOP Conference Series: Earth and Environmental Science, Volume 1142*, 2020.

[16] A. Asif, M. N. Bashir and S. Iqbal, "Smart Digester System for Biogas Production," *Journal of Big Data and Smart Cities ISSN: 2706-7912 1 ,* vol. 1, no. 2, pp. 1-6, 2021.