FLOWS: A PREDICTIVE FLUVIAL FLOOD MAPPING WEB APPLICATION FOR THE MUNICIPALITY OF CAMALIGAN, CAMARINES SUR USING MACHINE LEARNING

An Undergraduate Thesis
Presented to the Faculty of
Computer Science Department
BICOL UNIVERSITY COLLEGE OF SCIENCE
Legazpi City

In Partial Fulfillment of the
Requirements for the Degree of
BACHELOR OF SCIENCE IN COMPUTER SCIENCE

Alexandra Nicole D. Eclarinal
Yna Gabrielle P. Foronda
Francis Maurice B. Miranda

MAY 2025

Republic of the Philippines
Bicol University
College of Science
Legazpi City

RECOMMENDATION FOR THE ORAL DEFENSE

The undergraduate thesis entitled, "**FLOWS: A PREDICTIVE FLUVIAL FLOOD MAPPING WEB APPLICATION FOR THE MUNICIPALITY OF CAMALIGAN, CAMARINES SUR USING MACHINE LEARNING,"** prepared and submitted by **ECLARINAL, ALEXANDRA NICOLE D., FORONDA, YNA GABRIELLE P., AND MIRANDA, FRANCIS MAURICE B.,** in partial fulfillment of the requirements for the degree of BACHELOR OF SCIENCE IN COMPUTER SCIENCE, is hereby submitted to the thesis committee for oral examination.

ARIS J. ORDOÑEZ, DIT
Programming Adviser

In partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science, this undergraduate thesis entitled, "**FLOWS: A PREDICTIVE FLUVIAL FLOOD MAPPING WEB APPLICATION FOR THE MUNICIPALITY OF CAMALIGAN, CAMARINES SUR USING MACHINE LEARNING**", prepared and submitted by **ECLARINAL, ALEXANDRA NICOLE D., FORONDA, YNA GABRIELLE P., AND MIRANDA, FRANCIS MAURICE B.**, is hereby recommended for Oral Examination.

THESIS COMMITTEE

MARY JANE B. BURCE, MIT
Member

MARY JOY P. CANON, DIT
Member

RYAN A. RODRIGUEZ, MSCS, MIT
Chairman

Republic of the Philippines
Bicol University
College of Science
Legazpi City

RESULTS OF FINAL DEFENSE

Researchers	:	ALEXANDRA NICOLE D. ECLARINAL YNA GABRIELLE P. FORONDA FRANCIS MAURICE B. MIRANDA
Title	:	FLOWS: A PREDICTIVE FLUVIAL FLOOD MAPPING WEB APPLICATION FOR THE MUNICIPALITY OF CAMALIGAN, CAMARINES SUR USING MACHINE LEARNING
Place	:	Bicol University College of Science, Legazpi City
Date	:	May 4, 2025
Time	:	1:00 - 2:30 P.M.

PANEL OF EXAMINERS ACTION

RYAN A. RODRIGUEZ, MSCS, MIT ____________
Chairman

MARY JANE B. BURCE, MIT ____________
Member

MARY JOY P. CANON, DIT ____________
Member

Republic of the Philippines
Bicol University
College of Science
Legazpi City

APPROVAL SHEET

Upon recommendation of the Oral Examination Committee, this undergraduate thesis entitled, "**FLOWS: A PREDICTIVE FLUVIAL FLOOD MAPPING WEB APPLICATION FOR THE MUNICIPALITY OF CAMALIGAN, CAMARINES SUR USING MACHINE LEARNING",** prepared and submitted by **ECLARINAL, ALEXANDRA NICOLE D., FORONDA, YNA GABRIELLE P., AND MIRANDA, FRANCIS MAURICE B.**, is hereby approved in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

LEA D. AUSTERO, DIT
Department Chair

JOCELYN E. SERRANO, M.SC.
Dean, BUCS

Republic of the Philippines
Bicol University
College of Science
COMPUTER SCIENCE DEPARTMENT
Legazpi City

CONTENT ADVISER'S CERTIFICATION

This is to certify that this undergraduate thesis entitled, "**FLOWS: A PREDICTIVE FLUVIAL FLOOD MAPPING WEB APPLICATION FOR THE MUNICIPALITY OF CAMALIGAN, CAMARINES SUR USING MACHINE LEARNING"**, prepared and submitted by **ECLARINAL, ALEXANDRA NICOLE D., FORONDA, YNA GABRIELLE P., AND MIRANDA, FRANCIS MAURICE B.,** in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science, has been evaluated by the undersigned.

Issued this ____ day of May 2025 at BICOL UNIVERSITY COLLEGE OF SCIENCE, Legazpi City.

ARLENE A. SATUITO
Content Adviser

Republic of the Philippines
Bicol University
College of Science
COMPUTER SCIENCE DEPARTMENT
Legazpi City

PROGRAMMING ADVISER'S CERTIFICATION

This is to certify that this undergraduate thesis entitled, "**FLOWS: A PREDICTIVE FLUVIAL FLOOD MAPPING WEB APPLICATION FOR THE MUNICIPALITY OF CAMALIGAN, CAMARINES SUR USING MACHINE LEARNING"**, prepared and submitted by **ECLARINAL, ALEXANDRA NICOLE D., FORONDA, YNA GABRIELLE P., AND MIRANDA, FRANCIS MAURICE B.,** in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science, has been evaluated by the undersigned.

Issued this ____ day of May 2025 at BICOL UNIVERSITY COLLEGE OF SCIENCE, Legazpi City.

ARIS J. ORDOÑEZ, DIT
Programming Adviser

Republic of the Philippines
Bicol University
College of Science
COMPUTER SCIENCE DEPARTMENT
Legazpi City

EDITOR'S CERTIFICATION

This is to certify that this undergraduate thesis entitled, "**FLOWS: A PREDICTIVE FLUVIAL FLOOD MAPPING WEB APPLICATION FOR THE MUNICIPALITY OF CAMALIGAN, CAMARINES SUR USING MACHINE LEARNING"**, prepared and submitted by **ECLARINAL, ALEXANDRA NICOLE D., FORONDA, YNA GABRIELLE P., AND MIRANDA, FRANCIS MAURICE B.,** in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science, has been evaluated and edited by the undersigned.

Issued this 19th day of May 2025 at BICOL UNIVERSITY COLLEGE OF SCIENCE, Legazpi City.

LAIKA B. CAMPOSANO
Editor

ACKNOWLEDGEMENT

This research would not have been possible without the guidance and assistance of several individuals who, in one way or another, contributed their valuable help in the preparation and completion of this study.

We are heartily thankful to our advisers, Dr. Aris J. Ordonez and Prof. Arlene A. Satuito, whose invaluable guidance, expertise, and unwavering support shaped the direction of this research and greatly contributed to its successful completion. We would also like to extend our thanks to the members of the panel, Prof. Ryan A. Rodriguez, as our chair, and Prof. Mary Jane B. Burce and Dr. Mary Joy P. Canon as our panel members, for their constructive feedback and academic guidance throughout the development of this study.

We would like to thank Dr. Vladimir R. Foronda from Central Bicol State University of Agriculture (CBSUA) and Chief Meteorologist Officer, Engr. Josephine G. Albaño, Engr. Dan R. Ragodon, Ma'am Niva Jean Zamudio, and Ma'am Allen Buendia, from DOST-PAGASA's Bicol River Basin Flood Forecasting and Warning Center (BRBFFWC), for providing us with the data for training the model and also for providing us with valuable insights during our consultations and user testing. Their participation and insights enriched the quality and relevance of this work.

We are also thankful for Sir Roel C. De Loyola and Sir Fulgencio A. Agna from the Camaligan Disaster Risk Reduction Environment and Safety Management Office (CADRRESMO) for accommodating us for our consultations and user testing, and Sir Patrick De Leon from the Camaligan Planning Office for providing supplementary data for our study. We would like to mention UP Resilience Institute, as their Project NOAH served as a valuable reference for our paper.

We are also thankful for the inspiration and moral support provided by our classmates, peers, and technical mentors, who lent their time and knowledge during the system design and development phase.

We would like to extend our sincerest gratitude to the Department of Science and Technology – Science Education Institute (DOST-SEI) for their support throughout our academic journey. As scholars, we are deeply thankful not only for the educational opportunities but also for the financial assistance, which helped us focus on our research and academic responsibilities without additional burden.

We dedicate this paper and achievement to our family, whose unconditional love, sacrifices, and belief in us have been our greatest sources of strength. Their unwavering support throughout this endeavor made all the difference.

This study would not have been possible without the contributions of each individual and institution mentioned above.

A.N.D.E.

Y.G.P.F.

F.M.B.M.

ABSTRACT

ECLARINAL, ALEXANDRA NICOLE D., FORONDA, YNA GABRIELLE P., AND MIRANDA, FRANCIS MAURICE B., "FLOWS: A PREDICTIVE FLUVIAL FLOOD MAPPING WEB APPLICATION FOR THE MUNICIPALITY OF CAMALIGAN, CAMARINES SUR USING MACHINE LEARNING," (Unpublished Undergraduate Thesis, Bicol University College of Science, Legazpi City, May 2025)

In the Philippines, recurrent typhoons and heavy rainfall frequently inundate low-lying communities such as the Municipality of Camaligan, where both urban and river flooding pose serious risks. To improve preparedness, this study developed a predictive fluvial flood mapping web application. The study's objectives included data preprocessing (historical rainfall and water level, and other relevant data), predictive model training, model evaluation, web application implementation, and system user evaluation. The modeling pipeline integrated a NARX neural network for time-series prediction, and QGIS and HEC-RAS for hydrodynamic simulation, all orchestrated by automation scripts. After iterative fine-tuning, the final model achieved an R^2 of 0.87, Nash–Sutcliffe Efficiency of 0.87, RMSE of 0.178, and a prediction latency of 2.93 seconds. The web application was developed to showcase the flood extent maps, nearby critical facilities, and reports. User surveys indicated strong satisfaction, with 86.4% of respondents rating the system 4 or 5 out of 5 for usability and effectiveness. These results demonstrate the accuracy of the NARX algorithm while proving the feasibility of giving machine learning predictions spatial context through simulations for localized flood forecasting, offering an accurate and efficient tool for communities and responders to visualize and respond to flood threats.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

Background of the Study

The Philippines has ranked at the top of the World Risk Index (WRI) for three consecutive years (2022-2024), according to the World Risk Report of the Institute for International Law of Peace and Armed Conflict of the Ruhr-University Bochum, having an average of 46.86 score based on the interaction between the country's exposure and vulnerability (Atwii et al., 2022; Frege et al., 2023, 2024). Its high exposure to natural hazards is due to a combination of factors: an average of 20 typhoons annually, its location along the Pacific typhoon belt and the Ring of Fire, widespread poverty, and underdevelopment (Asian Disaster Reduction Center, n.d.; Climate Adaptation Platform, n.d.).

With a well-defined rainy season and the risks mentioned, excessive precipitation is common. Under some conditions, a surplus of rainfall can result in a disastrous flood (PAGASA, n.d.). Floods have caused direct and indirect impacts on people's lives, which include fatalities, health concerns, and destruction of shelters, livelihood, food sources, services, and assets (Alcantara & Christopher, 2019). A local study by Pascual et al. (2024) determined that knowledge and experience on flash floods, perceived severity, and perceived vulnerability significantly affect the intention for flash flood mitigation. This highlights how access to critical information is crucial in flood preparedness and mitigation.

Flood mapping is an existing method for providing knowledge regarding flood risk. This method shows the depth and extent of flooding in specific locations for certain periods. With this, individuals would be able to assess the risk of floods under specific scenarios, which can be used in risk mitigation and preparations (Smith, n.d.).

In addition, modern flood forecasting now involves the standard procedure of monitoring and analyzing hydrological and meteorological conditions in a river basin (PAGASA, n.d.). To build climate resilience, the national government of the Philippines deployed flood monitoring stations nationwide. Local Flood Early Warning Systems have also been developed in coordination with international institutions, combining the capacities of PAGASA, local government units (LGUs), and communities, with the first GIZ-LFEWS pilot launched in 2008 in the Binahaan Watershed of Leyte (Deutsche Gesellschaft für Internationale Zusammenarbeit, n.d.). Additionally, flood hazard maps developed by local institutions, such as the Project NOAH of the University of the Philippines Resilience Institute, are accessible through their web portal and mobile application (Cadiz, 2018).

In small and medium-sized river basins, water levels may rise rapidly, often without sufficient time to issue warnings. Limited data registration and coping strategies of such recurring events pose major challenges to local government units and their communities (Deutsche Gesellschaft für Internationale Zusammenarbeit, n.d.).

Meanwhile, Artificial intelligence (AI), the science and engineering of creating intelligent computer programs (Mccarthy, 2007), has made significant strides in Disaster Risk Reduction and Management (DRRM). AI has been applied to quantify climate change risks, which consists of several challenges: system complexity, uncertainty quantification, localization, and computational and practical constraints. Jones et al. (2023) explained how AI and machine learning (ML) can address these challenges by using data-derived models to fill gaps where scientific knowledge is lacking. Aside from the fact that they are cheaper to run than simulations, the models are also particularly suited for adapting physically based models to specific local conditions. The constraint of lacking relevant data can be alleviated through transfer learning that also addresses the computational cost of training complex models (Cheong et al., 2022).

Machine learning is a growing field of computational algorithms to replicate human intelligence by learning from data, regarded as the working horse in the new era of big data (El Naqa & Murphy, 2015). ML has been used in predicting flood susceptibility maps (El-Haddad et al., 2021), classifying regions as flooded or not (Madhuri et al., 2021), simulating and producing urban flood inundation data for training models (Hou et al., 2021), and real-time forecasting of urban floods (Berkhahn et al., 2019). While machine learning models have shown high accuracy in flood inundation modeling, reducing the computational time compared to simulating flood forecasting, challenges remain: low generalizability across regions, lack of public datasets, and difficulty embedding expert knowledge (Karim et al., 2023).

Recent year have seen the launch of several flood mapping initiatives through web applications like Google's Flood Hub (Nearing et al., 2024); the United Nations University Institute for Water, Environment, and Health's Flood Mapping Tool, which uses historical satellite imagery from Landsat to reveal inundation patterns (Mehmood et al., 2021); a flood-prone areas portrayal (Tavares da Costa et al., 2019); Copernicus' Emergency Management Service' Global Flood Awareness System (GloFAS) fully operational since April 2018; and the Dartmouth Flood Observatory (DFO), founded in 1993 based on optical satellite imagery (United Nations-Spider, n.d.). Local developments include the University of the Philippines' Project NOAH, which provides services beyond flood modeling (Mahar & Lagmay, 2014); Mapakalamidad.ph, which provides crowdsourced and up-to-date flood information (Mejico, 2020); and the Automated Flood Warning Online-based system of the Province of Leyte, which monitors the flooding situation through the system's graphical representation (Bentoso et al., 2021).

Spatial analysis software, tools that allow users to analyze and interpret geographic information, is essential for flood modeling. Dano et al. (2019) integrated a Geographic Information System with an Analytic Network Process and remote sensing-derived variables to map and assess the flood susceptibility in Perlis, Malaysia. In a flood susceptibility mapping in Bihar, India, it was integrated with the Analytical Hierarchy Process (AHP) and Remote Sensing (RS) with a cloud computing API such as the Google Earth Engine (GEE) (Swain et al., 2020). HEC-RAS and ArcGIS were also used in assessing the risk of hydrological hazard of the Turcu River in Romania (Trif et al., 2023).

With the wide variety of technologies and existing tools present, along with the multitude of potential algorithms for flood prediction, there is a need to improve the current scope of flood mapping and forecasting in the Philippines, especially in LGUs. The use of hybrid algorithms, with mapping software revolving around GIS-based flood risk assessment, has already been attempted in the country, specifically in the province of Leyte and Metro Manila (Cabrera & Lee, 2019). However, an attempt in the Bicol Region has yet to be made and is limited to assessments rather than an updated real-time source. A reliable and accessible flood forecasting and risk monitoring system is a technology that the region would heavily benefit from.

With this, the study proposed a predictive flood mapping system for the Municipality of Camaligan, Camarines Sur. The province of Camarines Sur hosts the Bicol River, one of the largest and longest river systems in the region. This river spreads across Camarines Sur, from Lake Bato to San Miguel Bay. The river system passes through multiple municipalities in the province, such as Naga, Nabua, Iriga, and Camaligan. In particular, Camaligan experiences heavy flood activity not just due to rainfall and typhoons, but also from the river's overflow. In 2020, Camaligan experienced three consecutive typhoons, which led to the flooding of the Bicol River, affecting the numerous rice paddies and nearby coastal residences (Garcia, 2022).

A more accurate method for predicting these potential overflows would help improve the disaster risk management of the municipality for better flood preparedness. In addition, Camaligan also has a working weather station, which helps gather accurate weather information in the area, making predictions much more accurate and reliable.

The proposed system seeks to deliver a dependable and user-friendly way for public users to access prediction flood maps of Camaligan, Camarines Sur, through a web application. By incorporating the use of machine learning algorithms, the system was able to forecast potential flooding accurately, pinpoint vulnerable areas, and issue early flood warnings. A map will visualize Camaligan's geographical features and essential facilities—like hospitals, barangay centers, schools, fire stations, and police stations. This system will be optimized for both mobile and desktop users. This innovative approach to public flood forecasting in the Municipality of Camaligan could significantly diminish climate-related risks and casualties while serving as a stepping stone for improving flood forecasting technology in the country.

Objectives of the Study

The main objective of this study was to develop a predictive flood mapping web application for the Municipality of Camaligan to improve flood prediction and facilitate the timely dissemination of flood information. Specifically, the study aimed to achieve the following objectives:

1. To gather and process historical water level and rainfall data for the model, and critical facilities and Digital Elevation Model for the web application;
2. To train a machine learning model capable of predicting water levels in Camaligan using relevant features such as historical water levels and rainfall data;
3. To evaluate the performance of the flood prediction model using evaluation metrics for regression models;
4. To develop a user-friendly web application that visualizes the flood extent map in Camaligan, Camarines Sur; and
5. To assess the web application's usability and effectiveness through testing and user surveys.

Significance of the Research

The significance of the study outlines the impact of the research on various stakeholders. Below are the key beneficiaries who will gain from this research.

Community. This especially entails the people living in areas highly prone to flooding, with a limited scope of flood forecasting. This is valuable in determining areas that are passable or not, and can help citizens make informed decisions regarding their safety, like planning evacuation routes.

Government. Having this system could help government officials and agencies provide better interventions for local communities in terms of disaster preparedness and resilience.

Local Officials and Emergency Responders. Having a more up-to-date source would help local officials anticipate flood events and administer early suspension notifications to the public, improve disaster response plans, allocate resources more effectively, and potentially reduce economic losses due to flooding.

Environmental Organizations. This study can be leveraged to advocate for climate action, engage communities, monitor flood mitigation efforts, and build partnerships for sustainable development. The web application provides a valuable tool for evidence-based decision-making and community engagement.

Researchers. The researchers involved in the development of this project will gain valuable experience in applying Artificial Intelligence technologies to real-world problems. This study may also serve as a stepping stone for further exploration of AI-driven solutions in Disaster Risk Reduction and Management.

Future Researchers. This study can also serve as a reference for any potential researchers who are also interested in this field. They can either utilize it as a method to improve the application itself or create a similar application for other regions.

Scope and Limitations of the Project

This study focused on the development of a system tailored for the Municipality of Camaligan, capable of predicting potential flooding areas six hours in advance and updating these predictions every six hours. The results were visualized on a map, which users can access through a web application.

This study was heavily dependent on third-party data as opposed to primary source gathering, which can lead to some data constraints. For the training and evaluation of the machine learning model, the provided data from DOST-PAGASA's Bicol River Flood Forecasting and Warning Center (Camaligan station) and the Central Bicol State University of Agriculture (CBSUA), such as the historical water level data and rainfall, were utilized. The data for the web application, such as critical facilities, were sourced from the UP Resilience Institute and the Camaligan DRRMO.

In addition, the considered independent variables were the historical water level and rainfall data, in meters and millimeters, respectively. The dependent variable pertains to the model's water level prediction, measured in meters. The model is specifically designed to predict fluvial flooding, which refers to flooding caused by the overflow of rivers, particularly as a result of sustained heavy rainfall that raises water levels beyond the river's capacity. It did not account for pluvial flooding, which occurs when intense rainfall overwhelms the local drainage systems or absorption capacity of the ground, leading to surface flooding even in the absence of river overflow. The study did not account for non-rainfall-related factors such as dam releases or upstream land-use changes that may influence river flow, but may incorporate tide changes. To gauge the overall model performance of the system, the following metrics were used together with their target criteria: coefficient of determination (R^2), Nash-Sutcliffe efficiency (NSE), root mean square error (RMSE), and prediction latency.

The website can be accessed by any device that has internet connectivity. The automation, particularly the script for exporting the flood extent maps from HEC-RAS, is limited to the Windows Operating System. Lastly, the researchers assessed the system usability through user surveys. Due to the unpredictability of real-time flood events, they tested the system using historical scenarios such as Typhoon Kristine. This ensured consistent and timely evaluation within the study period. By setting clear boundaries, the study ensured realistic and attainable outcomes given the available resources and timeline.

Definition of Terms

The following terms related to the research are defined operationally for better understanding:

NARX Algorithm – Nonlinear Autoregressive with eXogenous Input is a computational method, a recurrent dynamic network with feedback connections enclosing several layers of the network. This was the algorithm used to help the model analyze historical data and make water level predictions.

Artificial Intelligence (AI) – A discipline that enables a system to learn and perform water level predictions without the need for expert interference.

Camaligan, Camarines Sur – A municipality in the province of Camarines Sur, where the study focused due to its susceptibility to flooding and the presence of a river water level monitoring station.

Critical Facilities – Important infrastructure that is essential during natural calamities and disasters, such as schools, hospitals, fire stations, and police stations.

Digital Elevation Model (DEM) – A digital representation of a surface, showing elevation of different points. In this study, they are used to represent the area of Camaligan.

Features – Variables that were used as input to the machine learning model to predict flood levels, such as rainfall and flood level.

Flood level – The height of water above ground level in different areas of Camaligan, which will be visualized in the flood map using color indicators.

HEC-RAS – Stands for Hydrologic Engineering Center - River Analysis System, a hydraulic modeling software developed by the U.S. Army Corps of Engineers. Used for simulating the unsteady flow of the Bicol River in the vicinity of Camaligan.

Machine Learning (ML) – Techniques and models that benefit from large amounts of data in identifying patterns that contribute to the accuracy of their water level predictions.

Fluvial Flooding – Refers to the overflow of water from rivers, specifically the Bicol River, into adjacent areas, leading to inundation. This type of flooding occurs when the river's water level exceeds its capacity, often due to heavy rainfall or upstream water flow.

Flood Prediction – Flood level predictions are processed and delivered ahead of time, allowing for timely decision making.

Rainfall – The amount of rainfall over a given period, typically in millimeters per hour.

Predictive Flood Map – A map that shows the predicted extent and depth of the flood in an area, based on the model's prediction.

QGIS – Stands for Quantum Geographic Information System – an open-source spatial analysis software. Used to pre-process the digital elevation model and critical facilities of Camaligan.

Spatial Analysis Software – Software, such as a Geographic Information System, is used to process and visualize the flood prediction results.

Web Application – A web program that users can use to view the flood map of Camaligan and use its interface for accessing the flood prediction results.

CHAPTER 2

REVIEW OF RELATED LITERATURE AND STUDIES

This chapter examines previous research and studies pertinent to the development of a predictive flood mapping web application utilizing machine learning algorithms and spatial analysis software. This review explores key themes, including predictive flood mapping, artificial intelligence, machine learning, spatial analysis software, and algorithm evaluation, to establish a strong foundation for this study.

Predictive Flood Mapping

Floods, as natural disasters, cannot be entirely prevented. Therefore, it is essential for governments, relevant organizations, and communities to implement precautionary measures to mitigate their devastating impacts. Flood events cause thousands of casualties and considerable economic damage; as such, it is crucial to identify and minimize these losses (Bates, 2022). To minimize risks and ensure effective emergency responses, disaster management authorities must adopt proactive measures well in advance of a flood event. This includes leveraging state-of-the-art technologies that can predict disasters as early as possible, enabling timely and efficient response strategies. Flash floods must be predicted over a broad area with extended forecasting lead times to enable effective evacuation efforts. (Sayama et al., 2020). However, due to their unpredictable nature and numerous environmental variables, forecasting floods accurately remains difficult.

Thus, it is essential to improve the adoption of advanced technologies in developing automated systems for enhancing disaster prediction and forecasting (Munawar et al., 2022). However, until recently, such innovations have been limited to a few extensively researched locations.

In recent years, significant progress has occurred in weather forecasting and flood prediction, largely benefiting from the advancement of EWSs (Silvestro et al., 2019). Despite continuous innovation efforts by researchers and end-users, advancements in flood management have been limited, particularly in developing and least developed countries where population growth and urbanization are steadily increasing (Lim et al., 2019). According to Mohanty et al. (2019), disaster research is increasingly moving from risk and susceptibility assessments to developing frameworks to directly address local disaster problems. Additionally, there is a rising acknowledgment, in disaster science, of the importance of developing frameworks to strengthen the resilience of areas vulnerable to disasters, as emphasized by the Sendai Framework for DRR 2015–2030.

One way to enhance community resilience is a method called flood mapping. Maps that showcase flood probability are crucial for various use cases, including urban planning, risk reduction measures, and establishing advanced alert systems. (Panahi et al., 2021). Conducting flood susceptibility mapping (FSM) or flood mapping is critical to identify vulnerable zones, thereby helping to mitigate and reduce potential damage.

For instance, a study by Shah & Ai (2024) evaluated the damage from floods in Sindh, Pakistan, during 2022 and mapped areas at risk of flooding using FR and AHP methods. In the case of Natarajan et al. (2021), both control variables and historical flood data were integrated to develop a frequency ratio dataset for flood risk mapping. On the other hand, Sayama et al. (2020) examined the forecasting of flood occurrence by analyzing two extreme events using a nationwide rainfall-runoff model with high resolution (~150 m), driven through aggregated precipitation forecast models with a 39-hour lead time. The study revealed that the ability to forecast flash floods differed for the mentioned flood events.

Anusha & Bharathi (2020) utilized a much higher detail instrument called Synthetic Aperture Radar (SAR), combined with visualizations, to identify inundated areas caused by continuous rainfall and rising water levels in Rapti and Ghaghara Rivers in Uttar Pradesh, India, in August 2017. Floodwater mapping was conducted at the district level by overlaying the extracted water layer. The findings demonstrated that SAR data is highly effective for flood observation. In another study, a map that details flood forecasts in Wadi El-Laqeita, located in Egypt's Central Eastern Desert, was created via ML techniques, incorporating two algorithms: XGBoost and KNN. XGBoost outperformed KNN and was applied as a more effective method for flood mapping. These generated results offered valuable insights to decision-makers for future site development in the area (Ahmed et al., n.d.).

Meanwhile, deep learning algorithms continue to be utilized in flood prevention to address the advantages and disadvantages of models that heavily rely on numbers, which, although accurate, tend to be slow. Additionally, these techniques help enhance the performance of conventional flood mapping methods (Bentivoglio et al., 2022). Tien Bui et al. (2020) showcased a novel solution for FSM using a DLNN, with presented evidence in the high-frequency tropical storm region of the northwest mountains in Vietnam. The DLNN model was designed to create an inference system capable of predicting various levels of flash flood susceptibility.

Similarly, in another study by Panahi et al. (2021), the use of two advanced deep learning network frameworks, CNN and RNN, resulted in improved accuracy in predicting and mapping flash flood probabilities, surpassing the performance of previous research that used various approaches in other tropical areas.

Although there are various applications for flood-prone area identification, flood extent, and risk layouts, further research is required to investigate how deep learning can support immediate flood alerts during emergencies and contribute to flood risk assessment. A significant problem is the further creation of DL models capable of generalizing to new and undiscovered studies. Additionally, every examined model and its results were fixed or predictable, along with restrictions given to the uncertainties in results and the need for probabilistic predictions (Bentivoglio et al., 2022).

Conversely, there is increasing interest in applying advanced technology for disaster management. However, few undertakings examined the shift of these technologies when it comes to flood or fluvial monitoring. Al-Rawas et al. (2024) reviewed advanced technologies, such as AI/ML, IoT, robotics, and cloud computing, for flash flood EWS. The study revealed that AI/ML has been utilized in 64% of published research for these purposes, followed by IoT with 19%, cloud computing with 6%, and robotics with 2%. Techniques such as random forests and support vector machines showed high accuracy but require further refinement with larger datasets for testing. Additionally, AI/ML, IoT, and cloud computing enable the real-time distribution of early warnings to at-risk locales by utilizing digitalized channels like social media and messaging applications.

Significant progress has been made in using IoT and ML techniques to determine flood occurrences using factors such as humidity, rainfall, temperature, water flow, and water level. (Sankaranarayanan et al., 2020). However, the challenge in the Philippines is the lack of early warning systems using the mentioned technology that is more community-centered and can accurately predict flooding hours early.

Flood prediction and extent mapping models are being developed to address existing gaps in DRRM. These advancements aim to reduce the impacts of flood hazards and enhance flexibility in flood management strategies. Additionally, considering the potential effects of climate change, future susceptibility, and EWS implementations are a must to consider under various weather factors to support the development of long-term adaptation strategies.

Machine Learning (ML) in Predictive Flood Mapping

Over the years, the implementation of much more efficient and updated techniques for flood prediction has been heavily researched, including machine learning algorithms. Previous attempts have already been made using this approach with varying results based on the algorithm used. Mainly, the utilization of machine learning algorithms falls into two categories, supervised and unsupervised. Sharma et al. (n.d.) demonstrated the first category by employing five commonly used supervised learning algorithms, like SVM, K-NN, RF, Logistic Regression, and DTs. These five models were tested in terms of correctness, precision, recall, and ROC-score. Based on their testing, the model with the best metric score was Logistic Regression, with K-NN and SVM coming close.

Another attempt at testing multiple supervised algorithms was made by Seethepalli et al. (2024), where similar algorithms were used but with the added inclusion of two techniques, Deep Learning Models and Clustering techniques. In this test, they also had a similar metric system but opted for the use of F1-score rather than ROC-score. However, the result of their test had a different outcome compared to the first, with SVM having the highest accuracy of 95%, precision score of 1.0, recall of 92%, and F1-Score of 96%.

This result can also be supported by the study of Zehra (n.d.), in which she utilized SVM to predict critical lake floods in Chiang Mai, Thailand, and Rawal Lake in Islamabad, as well as medium-sized area floodings in Bird Creek, USA. The research emphasizes the efficiency of SVM in these scenarios as it learns through trial and error and its consideration of structural risk minimization.

However, SVM is prone to the curse of dimensionality and the instance of overfitting. Zehra also presented Nonlinear Autoregressive Neural Network (NARX) as a comparable alternative, having closer performance and is better suited for flood forecasting, as it is better suited for time-series data analysis, as SVM is suited for a higher degree of polynomials in its data.

Comparative studies have used algorithms such as Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), Naïve Bayes (NB), Artificial Neural Network (ANN), Deep Neural Network (DNN), Decision Tree (DT), Alternating Decision Tree (AD Tree), Logistic Model Tree (LM Tree), Reduced-error Pruning Tree (REP Tree), J48 Decision Tree (J48), Light Gradient Boosting Machine (LightGBM), Categorical Boosting (CatBoost), Extreme Gradient Boosting (XGBoost), and Cascade Forest Model (CFM) for Flood Susceptibility Mapping, producing relatively good performance with Cascade Forest Model (Seydi et al., 2022), Random Forest (Sellami et al., 2022), and Naïve Bayes Tree (Luu et al., 2021) outperforming them.

Regarding flood prediction models, the most popular choice is Neural Networks, specifically Artificial Neural Networks (ANN). This is further broken down by Asokan (n.d.), in which their study emphasizes the viability of the structure of ANN in error calculation and prediction comparison in patterns recognized in the target data. ANN is structured similarly to a human brain, with the network integrating three specialized layers. Outside of flood estimation, this ML technique is also used in different scenarios such as computer vision, speech recognition, and artificial intelligence, to name a few.

A case study by Abdul Rahman & Ramli (2024) showcased the actual use of ANN to predict flood susceptibility in Temerloh, Pahang, using rainfall data gathered from January 1, 2021, to December 31, 2022. Four input attributes were taken into consideration for the data—rainfall, water level, stream flow, and weather—while the target attribute was the actual flood occurrence, which would output “Flood” or “No Flood”. The resulting accuracy of their created model was around 0.9909, with a low error of 0.009 on its MSE and 0.096 on its RMSE. In this case, ANN is a highly viable algorithm for flood prediction, but has the downside of being fairly slow, with the tradeoff of high accuracy. Also mentioned was that the potential improvement of this algorithm would be to use it in tandem with other flood models to improve its speed and efficiency.

Random Forest is another ML algorithm that is utilized synonymously in labeling or grouping tasks. Hidayat (2023) showcased the use of Random Forest to detect floods based on rainfall data. They followed the same procedure of only considering the same attributes as their input, with the difference being the dates included, which range from January 2014 to December 2018, 5 years, as compared to the previous study of 2 years. This showed a comparison between ANN, Naïve Bayes, and Random Forest in model performance, which differentiates their accuracy. The data for this comparison were taken from other studies that followed a similar procedure, only using a different algorithm. This test displayed RF having the highest correctness score of 95.8%, while ANN and Naïve Bayes had 87.6% and 79.16%, respectively. This pertains to how Random Forest can be a potential algorithm for future flood early warning systems.

A similar study was undertaken in the Philippines, utilizing regression models to analyze weather patterns to predict floods. Neil Ruaro II et al. (n.d.) made use of Random Forest and Gradient Boosting to predict potential flooding in Makati, Cebu, and Iloilo. They gathered the weather data using a Python script from Stormglass.io and WeatherAPI, which scraped data from December 12, 2021, to March 12, 2022. They evaluated the performance of their optimized Random Forest and Gradient Boosting models with an existing prediction Naïve Model, for previous predictions of the area. The Random Forest and Gradient Boosting models did slightly worse compared to the Naïve Model in terms of accuracy, but have overall faster performance in dealing with time-series data.

Another was made to create a similar system in Metro Manila. Garcia et al. (2016) used ground pressure sensors and udometers to gather data, combined with an online server connection to receive observations remotely. They implemented RF to infer potential flooding on the streets of España Boulevard, proposing its ability to utilize bootstrap aggregation and the determination of randomized features by incorporating DTs to induce randomness. It was found to be beneficial in creating a better prediction model as the randomness factor can help decrease its variance. Using flood and rainfall data from October 2014 to July 2015, the Random Forest model was able to produce an OOB score of 0.9375 and a coefficient of determination R^2 of 0.938, indicating an overall good performance. With further optimization, the researchers mentioned that the model could potentially yield a higher coefficient and overall accuracy value.

A study by Lee & Kim (2021) proposed a unique method for faster fluvial prediction and overall improvement of its performance. The method they used was logistic regression, which produced a 2D grid that showed the flood inundation level and flood probability for each grid. They highlighted the importance of using logistic regression as compared to linear regression to avoid linear classifications, which do not take into account other distinct thresholds, unlike logistic regression. They tested the flood probability in the Taehwa River from January 1, 2016, to October 5, 2016, which was caused by Typhoon Chaba. The test compared the time it took to predict and measure how accurate the result was between the normal hydrological method of determining floods and the proposed method. The proposed algorithm showed promising results with 83.6% at its worst and 98.4% at its best in its prediction, and was able to do it within 30 seconds, as compared to the normal method, which took 100 minutes. They also mentioned that the abundance of data trained on the model would produce higher results with a similar time given.

Although supervised machine learning algorithms are much more in favor of flood prediction, there were also considerations in using unsupervised algorithms. A case of this was illustrated by Nhangumbe et al. (2023) in which they utilized a mix of unsupervised and supervised ML methods to analyze damage, as well as flood risk in Beira, Mozambique, by using Sentinel data. In this study, they wanted to compare the performance of an unsupervised algorithm to a supervised algorithm with varying types of images, ones with various classes and merging classes, ones with more than two classes, and those with no classes at all. The unsupervised algorithm yielded the highest mean (0.819 to 0.856) and lowest variability across the tests, which were based on the intersection over the union metric (IOU). Although the performance of the supervised algorithm was better in the case of images with labels, it can be said that unsupervised algorithms can be proven reliable in given situations.

The study by Li et al. (2023) also used K-Means clustering with the addition of DBSCAN to assess urban flood risks. Their scope was the eastern coast of Fujian Province in China, where they procured the historical flood data. In their proposed framework, after preprocessing the data and calculating their index weight, DBSCAN was used to extract the noise and while K-Means was used to predict the actual risk levels (high to low). The results show that DBSCAN was able to extract 36,796 noise points, accounting for about 22.2% of the total data, given the density threshold of the neighborhood radius and neighborhood point threshold. K-Means was now used to further divide the risk level of the gathered data with the K = 5 parameter set. Afterwards, the result was verified in comparison to the historical data, which shows a 96.84% accuracy. As mentioned in their study, potential improvements can be made by considering other factors such as natural, social, and economic aspects, which can affect urban flood assessment. The proposed method also heavily relies on the quality of the data being fed, so high-quality data is required for the model to be accurate. In addition, DBSCAN is prone to subjectivity, and larger parameters can heavily affect its clustering effect. Therefore, it is important to explore different parameter thresholds and provide accurate data to use.

The study by Seydi et al. (2023) featured a different machine learning algorithm for their flood prediction model. This model made use of the Cascade Forest Model (CFM), which generates a cascade of decision tree forests that gives much more accurate predictions due to its discriminative learning process through various ensembles of random forests. To put this to the test, the study evaluates their CFM with other algorithms such as RF, SVM, DNN, XGBoost, CatBoost, and LightGBM. Additionally, the data used for the testing consists of historical and satellite images. The notable results show that CatBoost had the highest prediction accuracy in non-flooded areas, but had an overall low performance in flooded areas, while LightGBM had the opposite result, of having flooded areas having more accuracy than non-flooded ones. However, in consideration of the metrics used, CFM had the highest overall accuracy of 92%, out of all test cases. The researchers stated that to achieve a high OA, they made use of 21 flood conditioning factors, and potential fine-tuning can still be made to this model.

Spatial Analysis Software in Predictive Flood Mapping

Spatial analysis software such as ArcGIS, QGIS, and languages like R and Python have become essential in flood mapping. They have enabled the integration of large datasets and various geographic data to do analysis, geoprocessing, simulation of rainfall-runoff processes, statistical modeling, and visualization. Mudashiru et al. (2021) provided a comprehensive review of several publications under Flood Hazard Mapping over the last 20 years, categorizing them into three parts based on existing modelling methods: numerical/physically-based, empirical, and physical approach. Geographic Information System (GIS) software is utilized in both physically-based and numerical models for data processing. The paper showed that based on application rates, empirical and physically-based modeling methods had higher application rates than the physical modeling method with 43.8%, 46.2%, and 10%, respectively.

In recent studies, there has been a surge in using both Machine Learning and Spatial Analysis Software for flood prediction and mapping. An example is Ighile et al. (2022) illustrating GIS and Machine Learning working hand-in-hand in predicting flood-prone areas along the Niger-Benue River Floods, which demonstrated the damage caused by torrential, continuous rains, overflows from water reservoirs, and dam failure. Their methodology consisted of four steps: (1) Geospatial database of all conditioning factors and historical flood occurrences, (2) development of the ML model, (3) validation, and (4) production of the flood vulnerability map. The production of final maps showcasing slope, Topographic Wetness and Standardized Precipitation Index, slope, and other factors was enabled through the use of QGIS and SAGA GIS. ArcGIS further contributed by standardizing datasets with its resampling feature and categorizing the flood susceptibility index map using a quantile-based method. This integration of machine learning (ML) and GIS highlights the recent advancements in the field of flood prediction modeling.

In a flood susceptibility assessment of Keelung City, a recently highly urbanized coastal city in Taiwan, Khoirunisa et al. (2021) used ArcGIS' tools in multiple stages of the procedure. The thematic layers, like elevation, angle, and aspect, were generated using the software. The line density tool calculated and presented a drainage density map produced from the drainage network, while the kernel density tool identified flood point densities across regions, highlighting areas with frequent past flood events and potentially greater future vulnerability. The work of the ANN was exported for ArcGIS to produce and visualize the map, subject to interpretation. The researchers declared that the mix of ANN and GIS was able to produce accurate flood models and, additionally, have these on top of a spatial environment.

Building on the integration of machine learning and GIS, Motta et al. (2021) adopted a data-driven approach, integrating both a custom machine learning classifier and GIS statistics to extend the model's predictions across an entire city. The study focused on binary classification, identifying whether there would be a flood would be observed or not, using six widely used algorithms in flood prediction. A GIS model was then utilized to fill in the lack of spatial representation that the ML models could provide. The model identified spatial clusters that are hot or cold spots, meaning that they are likely to be flooded or not, based on their spatial relationship using Gi* statistics. This two-step method revealed hidden patterns and spatial heterogeneity, despite the challenge of the limited data from just three weather stations.

A case study by Mourato et al. (2023) demonstrated the potential of incorporating stakeholders' input with GIS-based flood vulnerability modeling in mainland Portugal. By combining AHP-GDM—based on geospatial data and expert survey results—with GIS mapping, they computed a flood vulnerability index for mainland Portugal. This process utilized small territorial unit data and land-use/land-cover datasets to build a database, designing the structure of a vulnerability model, administering an expert survey, utilizing AHP-GDM for criteria weighting, and generating vulnerability maps. For the step of vulnerability mapping, spatial data geoprocessing steps were done in ArcGIS. Mapping flood vulnerability at the neighborhood level is critical for decision-making, offering spatial insights into each area's vulnerability, allowing targeted interventions for specific criteria, and enabling validation with prior knowledge.

Abedin & Stephen (2019) developed a framework to model and map the spatiotemporal dynamics in urban flooding, addressing the complexity of mapping such phenomena using GIS. The framework includes rainfall-runoff modeling, flow transformation, watershed delineation, inundation mapping, and validation, each element potentially influencing or being influenced by others. Tested on two urban sites, a DEM of 5m and 1m produced accurate results consistent with actual reports and observations. It provided flexibility of implementation in GIS software, each component having the possibility to be implemented in compatible software such as ArcGIS and QGIS. ArcGIS offers Python scripting capabilities for automating workflows within this framework, though its limitation lies in its inability to handle complex models effectively.

Aside from commercial software such as ArcGIS, there exists open-source software like QGIS that offers all, if not most, of the features that are needed in flood modeling. Sharif (2024) makes use of this exact software in their 2D hydronumeric flood modeling with the addition of a freeware BASEMENT for simulation. Through QGIS, regions were defined and delineated, and the 2D model was created based on readily available datasets like Digital Elevation Models, surrounding dike structures, and land use of the area, implying that the methodology used could be applied to data-scarce regions.

A study by Smith (2022) proposed a generalized approach to forecasting flooding event impacts with ArcGIS Pro, QGIS, and Python. The study revealed comparable geoprocessing workflows involving flood extent maps and global exposure datasets to evaluate agriculture, infrastructure, and population within a region. It also provided workflows for ArcGIS Pro, Python, and QGIS, and open-source modules using the Model Builder feature, Graphical Modeler, and premade modules. One limitation was when converting raster data to point vectors, minor discrepancies occurred as flood maps only partially cover raster cells, and the assigned point might not represent the inundated area accurately. Another, the general geoprocessing workflow generated from a flood extent map revealed minor variations in population counts and affected agricultural areas across methods. This consistency suggests that any of the three approaches can reliably evaluate flood impacts in inundated areas. Most of the differences in computations were connected to the variations in programming implementation and rounding differences.

In deploying the model as a web application, existing solutions like Mourato et al's (2021) Web-GIS system for fluvial flood forecasting and alerts were evaluated. This system has been in operation since 2019 for Portugal's Agueda River basin, combining three models: the WRF model for rainfall forecasting, HEC-HMS for hydrological simulations, and HEC-RAS 2D for hydraulic analysis. The Flood Frequency Analysis System (FFAS) predicts hourly flood depth, extent maps, and velocity for the next 72 hours, with each forecast taking approximately 90 minutes to complete. The areas are assigned to three classes of alert levels, visualized on publicly accessible maps through the Web-GIS platform. Users can select buildings and receive email alerts if it is part of the flood extent. In addition to this, automatic alerts are sent to authorities when hot spots are inundated. This type of deployment, accessible to the public, authorities, and emergency services, was able to provide an increased lead time to reduce damage and protect the lives and property of citizens.

While previous studies focused on river basin forecasting, Thi Hang et al. (2021) developed a flash flood susceptibility map for the national highway in Hoa Binh, Vietnam, highlighting areas prone to flash flooding. ArcGIS tools facilitated modeling causative factors of flash floods, using the weighted sum tool to integrate the final weights of 12 identified factors, ultimately producing a flash flood susceptibility index (FFSI) map. Validation using Shannon's Entropy, a simple yet effective method, yielded an AUC of 0.811. Its efficiency and low resource requirements make it suitable for settings with limited technology, software, or expertise.

And lastly, Nkwunonwo et al.’s (2020) literature review examined the current state of flood modeling, summarizing advancements and identifying key challenges in the field in the context of developing countries we try to generally identify what are some limitations in flood modeling studies that made use of spatial analysis software, they have identified several literatures with use cases of GIS. The review pointed out that there is the usual limitation of lack of access to data, if not, the unavailability of high-resolution data, which is a significant barrier in developing countries. While some offer open-access topographic data such as ASTERGDEM, they do not yield flood inundation model results as accurate as those from high-resolution data. The literature included in the review that was GIS-based was described to be quite suitable for urban flooding and other applications, and was comparatively good compared to numerical-based models, but some could not capture the dynamic flooding phenomenon and lacked extensive validation in terms of limitations.

The literature review explored a variety of existing studies that demonstrated the use of spatial analysis software for flood mapping, highlighting how many approaches are now able to integrate data with machine learning, stakeholders, or expert inputs, and spatio-temporal context. Despite challenges that were present, such as limited data availability, spatial analysis tools continue to prove their value in flood mapping.

Evaluation of Algorithms in Machine Learning for Predictive Flood Mapping

In a study conducted by Hadi et al. (2024) it can be seen in their use of the following metrics for evaluation – precision, accuracy, recall, and F1-score, which they cited as being one of the most prevalent indices to use in examining the outcomes' robustness when a Machine Learning or AI model is used. In this case, they were using the said metrics to compare different ML models to see which of them had the better performance. Afterward, these values can then be summarized into a proper visualization, either through graphs or matrices. These simple metrics are mostly used because of their commonality among model evaluations, including flood prediction.

However, some studies opt to add more key metrics to cater to their study. An example of this is the study of Khan et al. (2024), in which they further break down the said metrics. The core of the common metrics is derived from the confusion matrix, which displays the counts of correct and incorrect cases. These factors were incorporated into a formula that represents the performance metrics such as the model's correctness, sensitivity, specificity, and its F1 score. In addition to these metrics, this study included the calculation of Matthew's correlation coefficient, or MCC, an evaluation function that considers every category of the error matrix. Essentially, it gives straight-to-the-point evaluations as long as it has a positive value, meaning optimal results, and vice versa for poor model performance.

In the study by Srivanit et al. (2024), they included the use of another metric that can be added on top of the common metrics used, the Area Under the Receiver Operating Characteristic Curve, or AUC for short. This metric also gives a straightforward output, only giving a value ranging from 0 to 1, which serves as a measure of the model's predictive accuracy for a given outcome variable. The closer the value of the AUC is to 1, the greater the measure of the effectiveness of the model.

Although the use of common metrics is often seen in studies related to flood prediction, some use different key metrics. In their performance evaluation, they employed four distinct methods: MSE, R-Squared (R^2), RMSE, and MAE to assess overall accuracy on their regression model. MAE gives a numerical measure by determining the average of the absolute differences among the resulting values. Typically, the measure is used to assess the effectiveness of evaluation models, especially when large errors are frequently overlooked.

MSE, on the other hand, calculates the mean of squared differences in the distinguished and predicted outputs, which are important in evaluating models with an emphasis on accuracy when it comes to extreme values. RMSE is simply the squared value of MSE, which helps represent errors in units that are consistent with the dependent variable. The R^2 metric measures the amount of differences in the determined dependent factor accounted for according to the system.

Another function that is typically used in flood model evaluation is the Nash-Sutcliffe Coefficient (NSE). In the study by Jie et al. (2016), they describe

NSE as one of the objective functions that are primarily used to measure the performance or prediction power of hydrological models, alongside R^2 and Relative Volume Error (RE). In this study, they integrated these three single objective functions to see which has the highest accuracy of measuring and time lag peak flow. The result was that NSE had the overall highest accuracy and time delay in peak flow between the functions.

In support of these metrics, Hakim et al. (2024) further show an application of these metrics across different evaluation methods when it comes to flood prediction. The most commonly mentioned in the study was RMSE, followed by R^2 and NSE. The study also included other methods that can be used for flood prediction that are rarely mentioned, such as SPE, ANOVA, Chi-Square method, Willmott Index (WI), and Silhouette Coefficient.

In terms of model performance comparisons, Mosavi et al. (2018) conducted a survey that highlighted multiple different ML algorithms and compared them based on performance in different scenarios. This study illustrates two different forecast categories, short-term and long-term. The comparative performance analysis was based on the R^2 and RMSE metrics, and all models were trained using historical data. The study also features multiple single and hybrid models such as ANN, NARX, SVM, SVR, ANFIS, WNN, and BPNN. Although the study shows the prominence of ANN as a very popular algorithm for flood prediction, its performance can be outdone by other algorithms depending on the amount of data and attributes included, as well as the method used for data decomposition and pre-processing.

Another study by Byaruhanga et al. (2024) performed a similar comparison in which they conducted a scoping review of various machine learning algorithms used in various articles regarding Flood Early Warning Systems. Here, they included multiple models and categorized them based on popularity and types of data used. According to their scope, ANN represents approximately 40% of the methods used in flood forecasting, with Time-Series models and Fuzzy Logic following closely behind. The study also highlights the importance of physics-based deterministic models like HEC-HMS, HEC-RAS, and Xianjiang, among others. Additionally, the authors mention the utilization of models based on data such as Auto-Regressive Moving Averages (ARMA), as well as hybrid models like ANN-Fuzzy Logic (ANN-FL) and ANN-Gated Recurrent Unit (ANN-GRU).

Synthesis

The most common theme found in the literature is that it is viable to integrate machine learning with spatial analysis software. From the scoped studies, multiple algorithms were introduced for flood forecasting and prediction. Most studies made use of popular flood modeling algorithms such as ANN, SVM, and K-NN. Some studies introduced different machine learning algorithms, such as RF, GB, and NARX, as potential options, as these algorithms are better suited for time-series data and can greatly increase their performance depending on the quality of data used. The study made by Hidayat (2023) further demonstrates this as they compare the performance of Random Forest to other algorithms and check their viability as a flood prediction model. In addition, the potency of unsupervised algorithms was also introduced as an alternative method to employ hybrid models and potential data-cleaning techniques.

It was observed that there has been a surge of studies that aimed to make use of both the strengths of machine learning and spatial analysis software to enhance flood predictions. They are usually used in tandem through the machine learning model as the source of predictions and the GIS software for the visualization and incorporation of spatial context. Examples of these are Ighile et al.'s (2022), who developed a machine learning model to predict the flood-prone areas along a river and used QGIS, SAGA GIS, and ArcGIS to produce the flood susceptibility maps. The GIS software could also be used to fill in a limitation of ML models, such as the lack of spatial representation, as evident in Motta et al. (2021).

With its importance for real-life situations and decision-making, studies have started to incorporate stakeholders' opinions, experts' suggestions, and real-world profiles such as agriculture and population maps into the flood mapping models. Mourato et al. (2023) exhibited this by using the Portuguese Statistics for neighborhoods, land use, and land cover, and perceptions and priorities of experts in vulnerability assessment in forming a flood vulnerability index for the mainland of Portugal. Smith's (2022) generalized approach utilized agriculture, infrastructure, and population datasets in assessing a flood event's impact on an area. It was able to map which agricultural, infrastructure, or population points were affected, along with tables of flood impact results that can be accessed by stakeholders for flood impact mitigation measures.

In developing flood mapping models, studies emphasized the need for high-resolution data to achieve the best performance. Nkwunonwo et al. (2020) expressed that the lack of access to high-resolution data is a significant barrier in developing countries, which prohibits the development of an accurate flood mapping model. And along with the importance of quality input, all the studies at a minimum provide a visualized output that is key in policy making and public awareness. Khoirunisa et al. (2021) used ArcGIS to visualize elevation and angles onto maps, while Mourato et al. (2021) used Web-GIS to provide a flood extent map of the area along a river basin that can be categorized into classes of alert levels, along with the option of selecting buildings in the map that are used to trigger an alert if it has been flooded. Both applications for spatial analysis software were able to help in providing an increased lead time for the authorities and the public through sound and informed decision-making.

Data-driven approaches have requirements such as the need for large and quality datasets, as noted by Al-Rawas et al. (2024), Li et al. (2023), and Nkwunonwo et al. (2020). This exact requirement poses a barrier, particularly to developing countries like the Philippines, which may lack the data despite being a disaster-prone country. What comes with the advantages of machine learning algorithms is the demand for expertise and computational requirements for training and maintenance. These advancements discussed in the review are often limited to regions with extensive datasets and technological infrastructure, leaving developing areas with data scarcity and limited resources underexplored, pointing to the need for further research that is scalable and can work around these limitations.

Overall, the literature consistently emphasizes the critical role of advanced technologies like AI, machine learning (ML), and spatial analysis working together to enhance flood prediction and management. There is broad agreement on the effectiveness of ML algorithms, such as artificial neural networks (e.g., CNN, DLNN), in improving the accuracy of flood susceptibility mapping and real-time forecasting. Studies highlight the superior performance of these models over traditional methods, particularly in predicting flood-prone areas. However, there are challenges related to their generalization across diverse geographies and the lack of robust datasets in developing regions. While researchers agree on the potential of integrating smart technologies like IoT and cloud computing into Early Warning Systems (EWS), the effectiveness of these systems remains limited by infrastructural and data constraints, especially in disaster-prone but resource-limited areas. These findings inform this study by highlighting the need for predictive flood mapping solutions that are both accurate and adaptable to local contexts. The gaps identified in current research—such as the limited use of probabilistic models and the need for community-centered early warning systems—underscore the importance of developing a solution tailored to the specific challenges of regions, especially the Municipality of Camaligan, Camarines Sur.

Gap Bridge by the Study

While significant advancements have been made in flood prediction and mapping using AI and ML, there are still challenges related to data availability, model accuracy, and the adaptability of these technologies in diverse settings. Despite significant progress, there is a noticeable gap in research focused on regions with limited data availability and technological resources, particularly in developing countries like the Philippines.

Current studies predominantly use deterministic models, which do not account for uncertainties, and are largely applied to well-researched areas with abundant data. Moreover, while IoT and ML technologies have shown promise in enhancing early warning systems, their implementation in community-centered, resource-limited environments remains underexplored (Lim et al., 2019; Bentivoglio et al., 2022). There is also a lack of solutions specifically tailored to predict floods with sufficient lead time (e.g., six hours ahead) in geographically diverse regions.

This study aims to address these gaps by developing a predictive flood mapping web application specifically designed for the Municipality of Camaligan, an area vulnerable to frequent flooding. Unlike previous research, this project will integrate machine learning algorithms with spatial analysis to predict flood levels six to 12 hours in advance, even in data-constrained environments. By focusing on a community-centered approach, the system will provide localized early warning capabilities, ensuring timely evacuation and disaster response.

This research will contribute to new insights by evaluating the effectiveness of NARX in predicting flood levels with limited data inputs, thus providing a scalable and adaptable solution for flood-prone, resource-limited regions. Additionally, this study will fill the existing research gap by incorporating spatial analysis and community-centered early warning strategies, ultimately supporting disaster risk management and resilience in underrepresented areas.

Conceptual Framework

The conceptual framework for this study emphasizes the creation and execution of a predictive flood mapping web application using Machine Learning and Spatial Analysis Software. This model outlines how different components contribute to the achievement of the study's objectives.

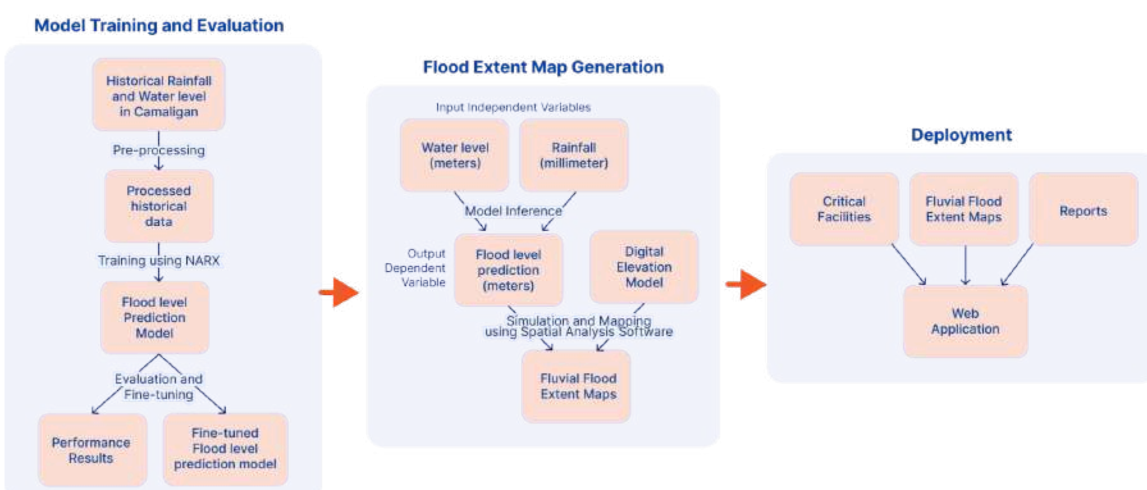


Figure 2.0. **Conceptual Framework**

Drawing from Figure 2.0, this study makes use of historical rainfall data in Camaligan with the historical water level of the Bicol River alongside their real-time data as inputs for model development. The real-time data will also be used as grounds for predetermined verification for model testing and evaluation. The study aims to produce an accurate flood prediction model that takes into account the rainfall and river water level as factors. As such, NARX will be employed for the model. This study will also utilize other technologies such as spatial analysis software, Web-GIS libraries, and web application languages for web development.

During the developmental process, the data procured will initially undergo data pre-processing and analysis to decrease potential noise and outliers. The model development process will start with the application of the specified machine learning techniques. This evaluation will use R^2, NSE, RMSE, and prediction latency as its primary metrics. HEC-RAS will be utilized to serve as a mapping software to provide the flood extent map in Camaligan based on the water level at the point of the monitoring station. In tandem, a web application will also be developed to host the said map overview that includes features such as alert levels, critical facilities, and real-time viewing.

This study features two main outputs. First, the study seeks to create a precise model for predicting fluvial floods that can handle a six-hour timeframe and be able to predict the next six hours of potential flood activity. The second is to produce a flood map of Camaligan, Camarines Sur, which any public user can access.

This review of related literature and studies investigated existing findings, comparisons, frameworks, use cases, and cons of ML and spatial analysis software in data-driven predictive flood mapping relative to traditional methods. By exploring various approaches, models, and technologies, the review highlights significant advancements in flood prediction, including the integration of machine learning with GIS tools and the use of data-driven techniques for improved accuracy. However, challenges such as data scarcity, computational demands, and the need for community-centered early warning systems persist, especially in developing regions. These insights inform the current study, emphasizing the importance of developing a predictive flood mapping system that not only addresses these gaps but is also adaptable and effective for flood-prone areas like Camaligan.

CHAPTER 3

METHODOLOGY

This chapter outlines the research methodology for developing the Predictive Flood Mapping Web Application for Camaligan. It discusses the methods in detail for each of the objectives of the study. The Machine Learning (ML) process is thoroughly discussed, covering data gathering preparation, model training, evaluation, fine-tuning, and integration with spatial analysis software for spatial context across Camaligan. The chapter details the phases of system development, including user story creation, product backlog definition, sprint planning, execution, review, retrospectives, and final deployment. Together, these methodologies and tools ensured a structured and efficient development process for the application. The overall system development followed an Agile methodology, specifically the Scrum framework, to allow for incremental progress, continuous feedback, and early delivery of key features. This approach supported the five main objectives of the study, with an emphasis on iterative improvements and user-focused development, particularly for the web application component.

I. Data Gathering and Preparation

To support the predictive flood modeling system for the Municipality of Camaligan, this study gathered and prepared several key datasets. Historical water level and rainfall data were obtained in .xlsx format from DOST-PAGASA's Bicol River Flood Forecasting and Warning Center (BRFFWC). These included rainfall data from November 1980 to January 2025 and water level data from January 2008 to January 2025. For consistency in model training, the rainfall data was trimmed to match the water level timeframe.

Year	Month	Day	Hour	Rainfall
1980	11	1	9	0
1980	11	1	10	
1980	11	1	11	
1980	11	1	12	
1980	11	1	13	
1980	11	1	14	
1980	11	1	15	
1980	11	1	16	
1980	11	1	17	
1980	11	1	18	
1980	11	1	19	
1980	11	1	20	
1980	11	1	21	
1980	11	1	22	
1980	11	1	23	
1980	11	1	0	
1980	11	1	1	
1980	11	1	2	
1980	11	1	3	
1980	11	1	4	
1980	11	1	5	
1980	11	1	6	
1980	11	1	7	1
1980	11	1	8	0
1980	11	2	9	

Year	Month	Day	Hour	WaterLevel
2008	1	1	0	
2008	1	1	1	1.66
2008	1	1	2	1.62
2008	1	1	3	1.57
2008	1	1	4	1.52
2008	1	1	5	1.44
2008	1	1	6	1.37
2008	1	1	7	1.3
2008	1	1	8	1.34
2008	1	1	9	1.45
2008	1	1	10	1.63
2008	1	1	11	1.83
2008	1	1	12	1.93
2008	1	1	13	1.95
2008	1	1	14	
2008	1	1	15	1.81
2008	1	1	16	1.68
2008	1	1	17	1.57
2008	1	1	18	1.47
2008	1	1	19	
2008	1	1	20	1.34
2008	1	1	21	1.34
2008	1	1	22	1.35
2008	1	1	23	1.44
2008	1	2	0	1.54

Figure 3.1. **Samples from the Original Rainfall and Water Level Dataset**

Using Python's pandas library in JupyterLab, the data was converted into CSV format and cleaned for processing. The datasets were merged into a single file with the following columns: {Year, Month, Day, Hour, Rainfall, WaterLevel}. Interpolation techniques were applied to fill in missing values, and all entries were reformatted to retain only numerical types suitable for machine learning models.

Year	Month	Day	Hour	Rainfall	WaterLevel
2008	1	1	9	0	1.45
2008	1	1	10	0	1.63
2008	1	1	11	0	1.83
2008	1	1	12	0	1.93
2008	1	1	13	0	1.95
2008	1	1	14	1	0
2008	1	1	15	0	1.81
2008	1	1	16	0	1.68
2008	1	1	17	0	1.57
2008	1	1	18	0	1.47
2008	1	1	19	0	0
2008	1	1	20	0	1.34
2008	1	1	21	0	1.34
2008	1	1	22	0	1.35
2008	1	1	23	0	1.44
2008	1	1	0	0	0
2008	1	1	1	0	1.66
2008	1	1	2	0	1.62
2008	1	1	3	1	1.57
2008	1	1	4	0	1.52

Figure 3.2. **Samples from the Merged Dataset**

The prepared dataset was then split into 80-20 – 80% for training and 20% for validation subsets to facilitate model evaluation. The training data consisted of values recorded before September 2021, while the testing data only included values from September 2021 onwards to assess the model's predictive performance on unseen data. Additionally, the dataset was scaled to ensure consistency and improve the convergence of the model during training. A new column {FWaterLevel} was added to serve as the dependent variable for prediction. This column represents the water level six hours into the future, allowing the model to learn temporal patterns for flood forecasting.

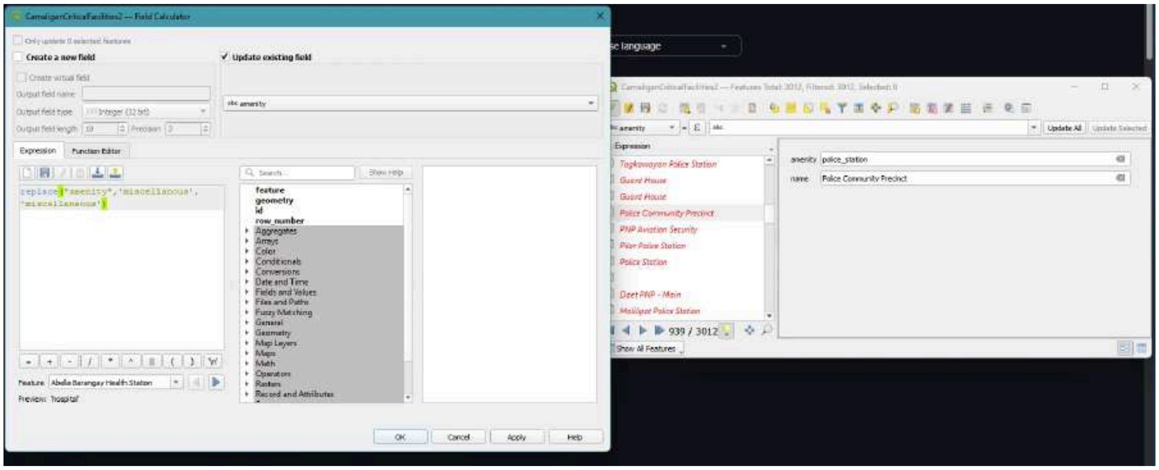


Figure 3.3. **Correcting Critical Facilities**

In addition to the time-series data, critical facilities data were also gathered and processed, an example of which is shown in Figure 3.3. These were sourced from the UP Resilience Institute and the Camaligan Planning Office. Discrepancies in Coordinate Reference Systems (CRS) were resolved using QGIS, converting all spatial data to a format compatible with Mapbox (EPSG:3857). Critical facility records were further refined by only including facilities in the vicinity of Camarines Sur. The merged dataset was composed of points of critical facilities plotted throughout Camaligan. Each facility was also categorized based on the amenity type, which was expanded to include other amenities such as places of worship and evacuation centers.

As for the water simulation and identifying flood extents in low-lying areas such as Camaligan, an accurate Digital Elevation Model (DEM) was procured. Although a preliminary DEM was generated using elevation data sourced through the use of Google Earth Pro, GPS Visualizer, and QGIS, it failed to capture the fine-scale elevation differences between the Bicol River and the surrounding land that are needed for the simulation. This limitation was primarily attributed to the coarse resolution of publicly available DEMs (e.g., SRTM), which typically provide a spatial resolution of up to 30 meters, rendering them insufficient for modeling narrow and low-lying riverbanks and adjacent floodplains. Following consultation with a GIS engineer, it was determined that a higher resolution DEM would be more appropriate. To achieve this, the researchers explored the use of Sentinel-1 satellite imagery, an openly available source of synthetic aperture radar (SAR) data.

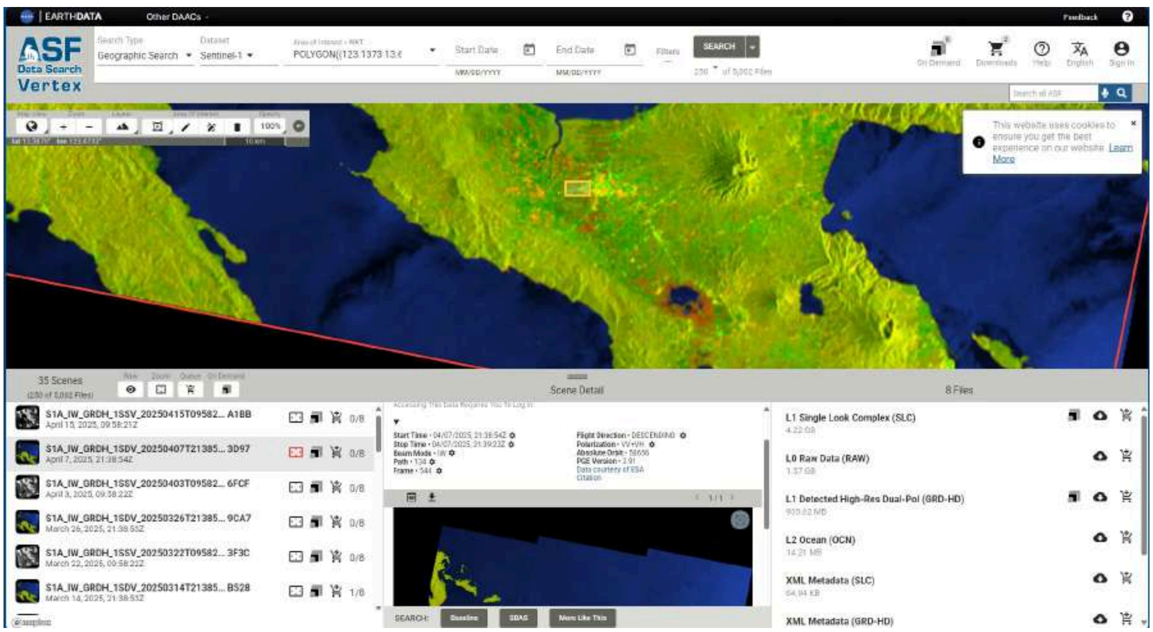


Figure 3.4. **Downloaded SAR Data**

Sentinel-1 data that covered Camaligan was accessed via the Alaska Satellite Facility (ASF) Data Portal, as shown in Figure 3.4, using a geographic search query, with a polygon drawn around the extent of the boundary of Camaligan. A recent capture was selected, and the Level-1 Ground Range Detected High-Resolution Dual-Polarization (GRD-HD) product was downloaded.

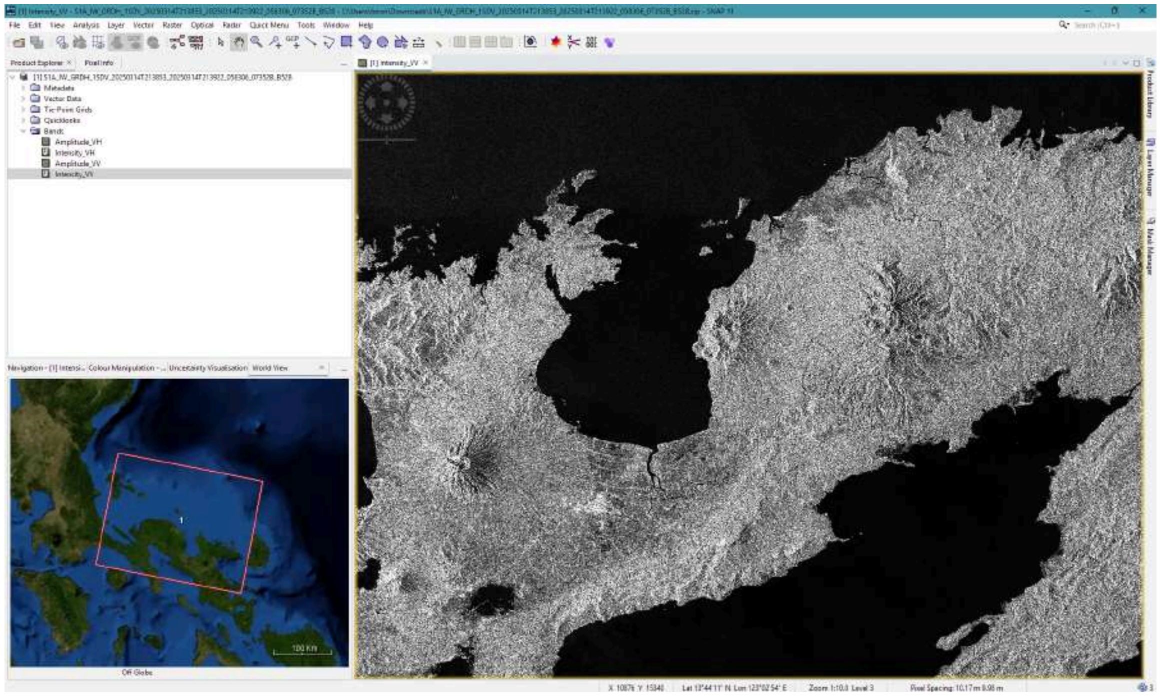


Figure 3.5. **Processing Satellite Capture in SNAP**

The downloaded SAR data was processed in the Sentinel Application Platform (SNAP), as shown in Figure 3.5. Within SNAP, a Range-Doppler Terrain Correction was applied using the Amplitude_VV band, and a Digital Elevation Model was generated from the corrected product.

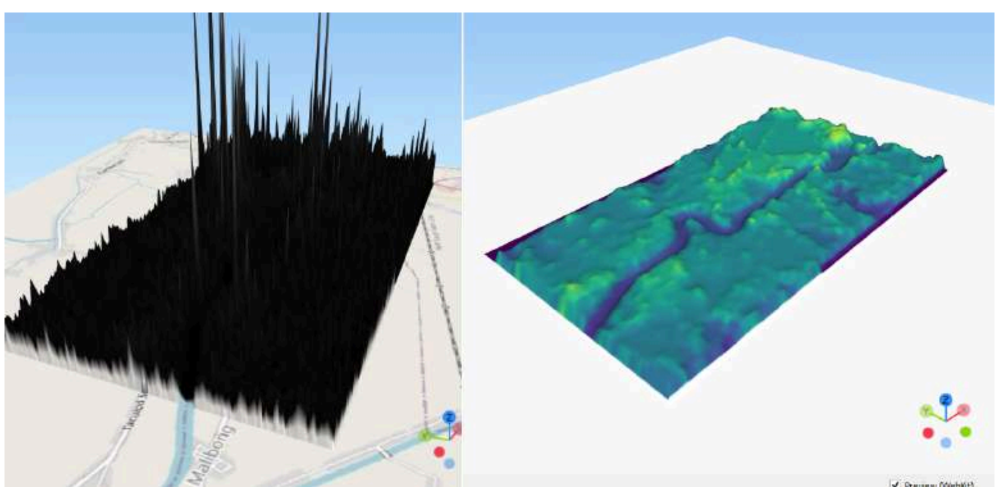


Figure 3.6. **Initial vs Smoothened DEM Comparison**

Despite capturing the river's course more clearly than the generated DEMs from the previous method, the initial DEM displayed unnatural elevation spikes across the land area. To address this, a series of raster smoothing operations were performed in QGIS using tools such as r.neighbors, Gaussian filtering, and sink-filling algorithms from GRASS GIS and SAGA NextGen toolboxes, as seen in Figure 3.6. The DEM was also clipped to an adjusted boundary polygon that extended 100 meters beyond the official Camaligan boundary, allowing it to encompass the full width of the river and reduce simulation errors. Furthermore, the elevation values were scaled down, reducing the maximum elevation from ~315 meters to a more realistic value, given Camaligan's flat topography.

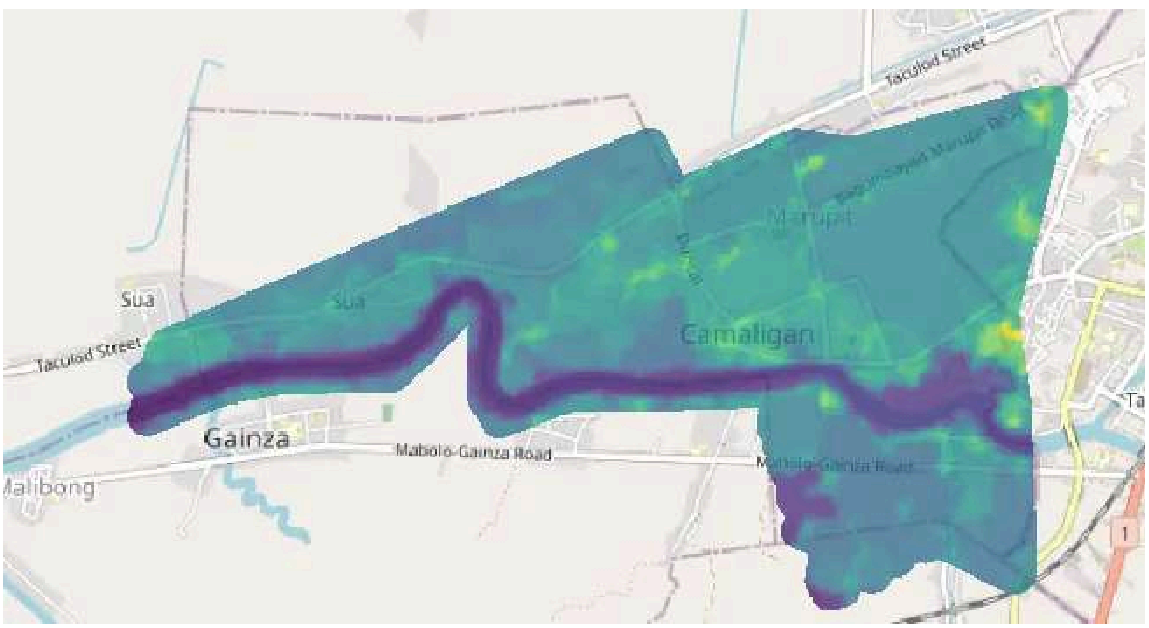


Figure 3.7. **Final DEM**

Figure 3.7 above shows the final Digital Elevation Model that was used for the water simulation in HEC-RAS.

II. Model Training and Fine-Tuning

The first objective addressed the initial phase of the machine learning process, which involved the preparation of historical water level and rainfall data. This section proceeds to examine the subsequent two phases.

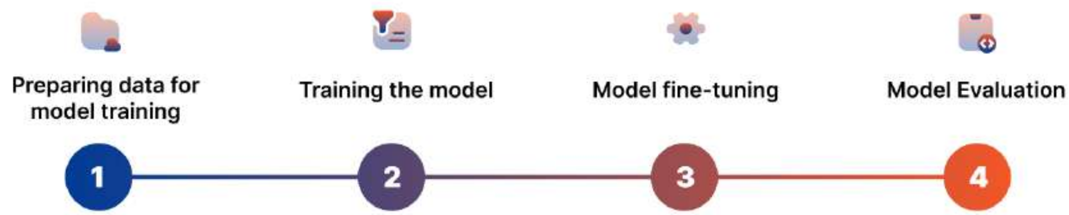


Figure 3.8. **ML Process**

The second and third step, as shown in Figure 3.8, handles the training of the model that will output the flood level predictions, and fine-tune it to produce more accurate inferences. This included preparing the needed packages, setting up the model, loading in the data, running the training process, and adjusting the hyperparameters for training the model further. The initial model was developed in Jupyter Lab using Python as the programming language. The prepared dataset was loaded using the pandas library and structured into data frames. Rainfall and water level values were then reshaped and scaled using NumPy to standardize the input features for model training. These features were combined to form the input for the predictive model.

The model employed the Nonlinear Autoregressive with eXogenous Input (NARX) algorithm, using the SysIdentPy library, which is based on the NARMAX (Nonlinear Autoregressive Moving Average with eXogenous inputs) framework.

This approach involved specifying input lags to capture temporal dependencies between past rainfall, water levels, and future flood levels. The model was trained on 17 years of historical data to detect meaningful patterns and correlations between the independent variables (past rainfall and water level) and the dependent variable (predicted water level).

However, the initial architecture failed to produce accurate results, returning a negative R^2 score of -0.387, which indicated poor model performance and an inability to generalize from the data. Due to these shortcomings, the approach was discontinued, and an alternative model architecture was explored to achieve better predictive accuracy.

An alternative model architecture was utilized that made use of a straightforward, NARX implementation function, which automatically created a NARX dataset that implemented input lags as additional parameters. This method also included data scaling towards the input, which helped make the prediction much more accurate. This approach proved to be better as it yielded a better R^2 score of 0.23, which indicated the model was now learning as compared to the previous iteration.

To improve the predictive performance of the model, it underwent several rounds of fine-tuning. One of the major adjustments involved modifying the input-output configuration to use 48 past rainfall features and 48 past water level features, effectively increasing the dimensionality of the NARX dataset to 96 features. This allowed the model to capture a broader temporal context from past rainfall and water level data to enhance its ability to generalize across time sequences.

Another adjustment made during fine-tuning was the specification of the activation layer, in which the default use was rectified linear unit (ReLU). This was later changed to hyperbolic tangent (tanh) as it captured the sinusoidal nature of the input data. Additionally, the optimizer was changed to Adam, a widely used optimization algorithm known for its adaptive learning rate and effective performance in training deep neural networks. The learning rate was set to 0.0001, a value chosen to ensure stable convergence during training while preventing overshooting of the loss function.

The number of training batch sizes was also varied throughout the re-training process to empirically determine the configuration that would yield the best performance. Initially, the batch size was set to 32 but was gradually decreased in order to improve model performance. A batch size of 8 was determined to be the optimal number when training with the second model architecture, as it produced more consistent and better results.

III. Model Evaluation

To assess the performance of each model iteration, the study applied key regression metrics, particularly the coefficient of determination (R^2) score, Nash-Sutcliffe Evaluation (NSE), Root Mean Squared Error (RMSE), and prediction latency.

The evaluation process was conducted after multiple iterations of model tuning and adjustment of parameters to improve accuracy. This allowed performance comparisons across different model configurations. The model was evaluated using the validation/test dataset. The predicted values were compared against the actual flood levels for the same timestamps, and performance results were recorded.

Each version was tested using the same evaluation metrics, and the model that satisfied the predefined threshold metrics of $R^2 \geq 0.8$, NSE ≥ 0.8, RMSE < 0.2, and prediction latency of less than 5 minutes was selected as the final predictive model for deployment.

IV. Web Application Development

To develop the web application component of the system, the research adopted the Scrum framework, an agile methodology suited for iterative and user-centered software development. Scrum enabled the development team to respond quickly to feedback, adapt features incrementally, and maintain a structured yet flexible workflow. Scrum supports this study through its key practices, including regular sprint planning, daily stand-ups, and sprint reviews, which promote collaboration and accommodate the dynamic nature of developing the predictive flood mapping web application.

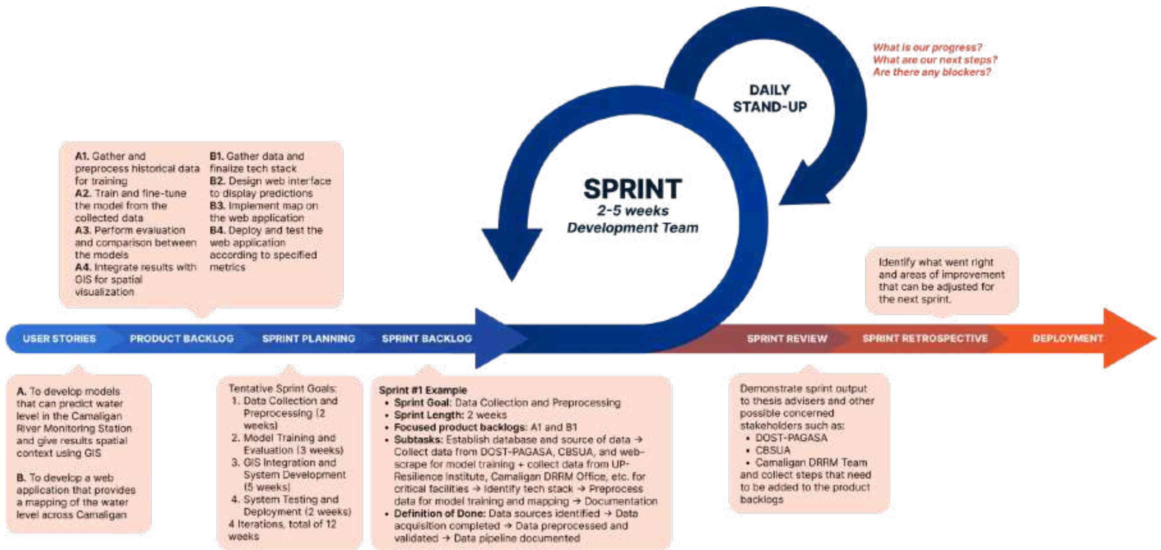


Figure 3.9. **SCRUM Framework**

As shown in Figure 3.9, the process was divided into sprints, each typically lasting one to two weeks, focusing on delivering specific system features such as location monitoring, critical facility overlays, flood level visualization, and mobile responsiveness. At the beginning of each sprint, the team conducted sprint planning to identify tasks from the product backlog, which contained all feature requirements gathered from stakeholder input and user needs.

During development, daily stand-ups helped track progress and resolve issues, ensuring team alignment. After each sprint, a review and retrospective were conducted to evaluate the output, gather feedback, and refine upcoming tasks. This cycle continued until the full-featured system was completed.

The first sprint was concerned with creating and finalizing the structure and design of the web application. Diagrams and designs were created to illustrate the system's functionality, user interactions, and interface design. The first sprint was allotted two weeks and was considered done when all necessary diagrams and designs had been created and reviewed.

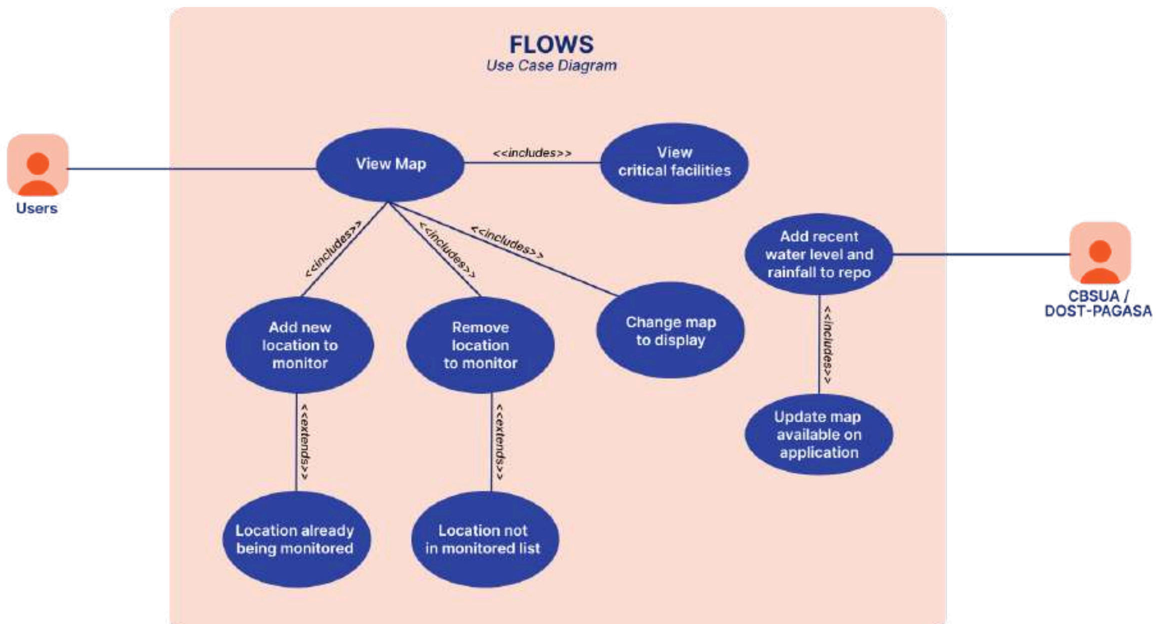


Figure 3.10. **Use Case Diagram**

The use case diagram in Figure 3.10 above illustrates the interaction between the users, such as the citizens and Camaligan DRRMO, and the Flood Mapping Web Application. The user interacts with the system to view the predictive flood map that provides a detailed representation of flood-prone zones, including critical facilities, to assist in decision-making during emergencies. The system allows users to view the predicted flood water level in a specific location. An area from the monitored list can also be removed when it is no longer of interest. The system was designed to update the available flood extent maps whenever a new set of data is provided by CBSUA/DOST-PAGASA.

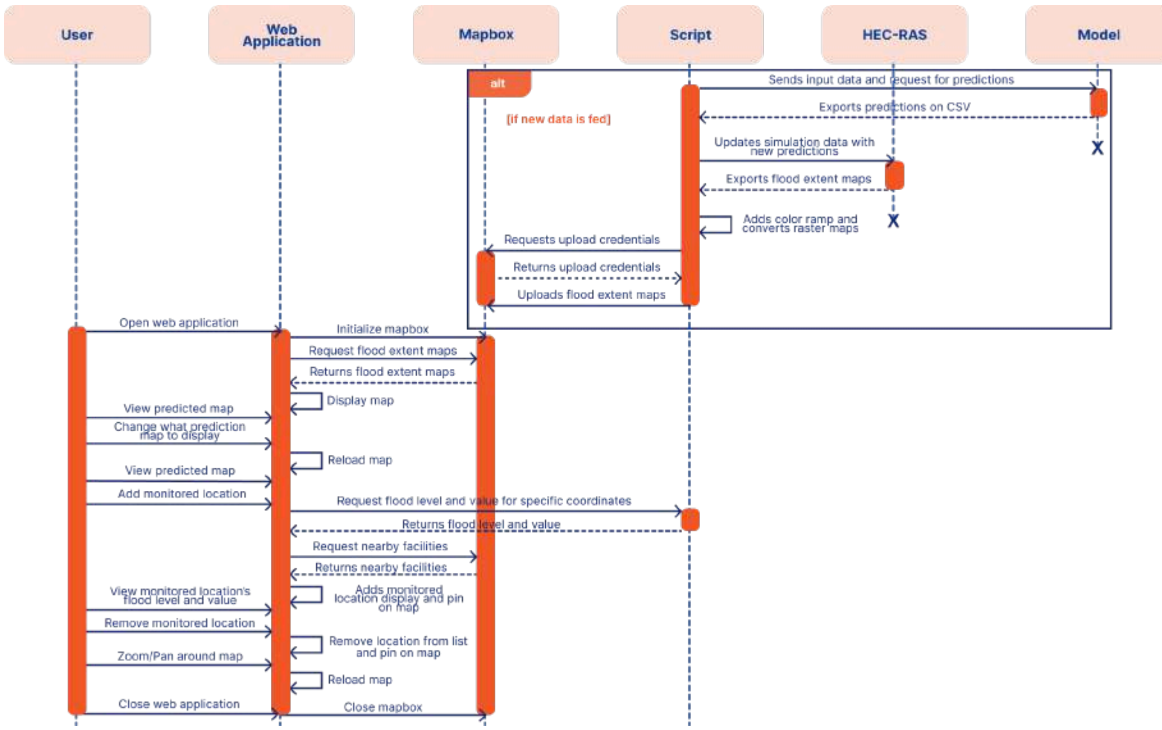


Figure 3.11. **Sequence Diagram**

To describe the back-end processes of the web application, a sequence diagram in Figure 3.11 was created. The diagram shows how the system works whenever new data is fed. This is handled by the automation script programmed to take in the new data, feed it to the model for prediction, pass the predicted values for simulation, apply post-processing to the flood extent maps, generate the CSV files for the reports, and upload the converted flood extent maps to Mapbox. As for the web application side, most of the processing is handled by the web application's scripts and through Mapbox.

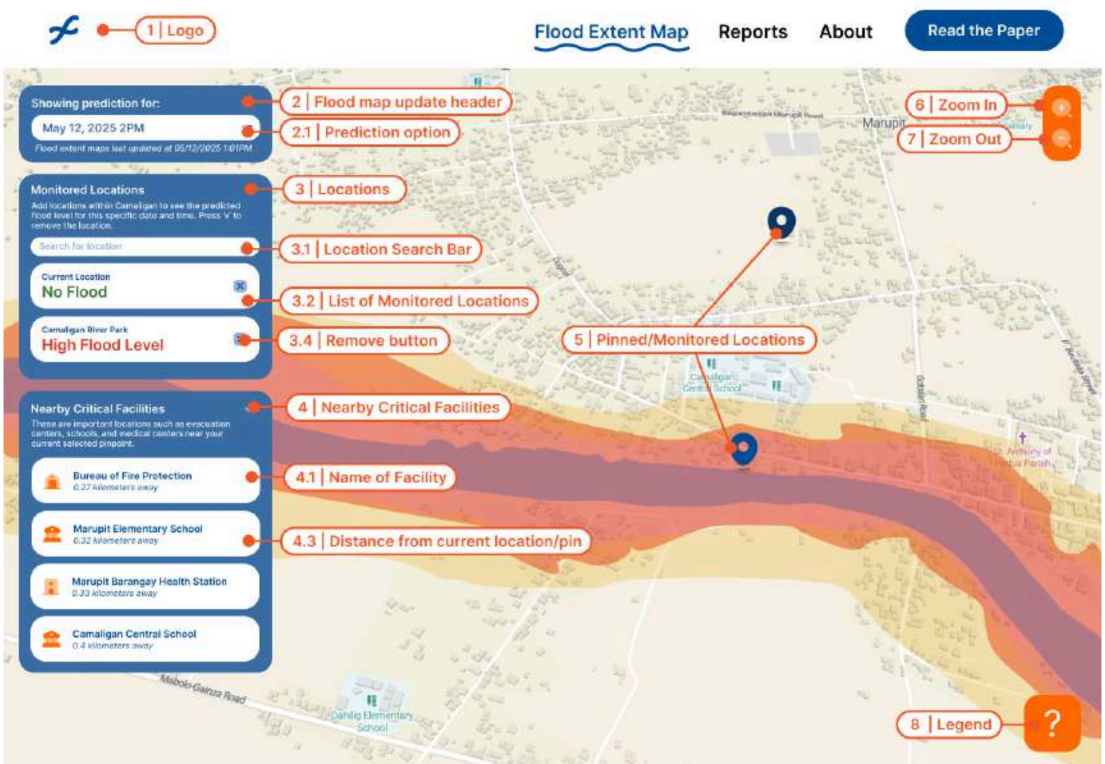


Figure 3.12. **UI Mockup**

Figure 3.12 provides a clear and user-friendly interface for interacting with the predictive flood mapping web application. Each element serves a specific function to ensure efficient monitoring and navigation of flood-related data.

Positioned at the top of the interface is the website logo in which represents the application's branding and serves as a visual anchor for the UI. Located at the bottom of the logo is the Prediction Tab, which displays the date and time of the most recent flood map update, ensuring users are informed of the data's timeliness. Clicking this would bring down a dropdown menu allowing users to select from available map updates (e.g., hourly predictions).

Next to this is the Monitored Location Tab, which provides users with tools to manage and monitor specific locations. A location search bar is included that enables users to search for a specific location to monitor. This location is then added to the monitored location list that displays all locations being tracked by the user. This also indicates the flood status of each location (e.g., "No Flood," "High Flood Level"). The added monitored locations are visualized by location markers on the map, indicating locations being tracked. Optionally, users can remove a location from the monitored list with ease with the remove button on the side of the location.

Following this is the Nearby Critical Facilities Tab, which displays the critical infrastructure near the user's current or monitored location. This displays the name of nearby critical facilities, such as hospitals or evacuation centers, and indicates how far each facility is from the selected location.

At the bottom right corner of the website is the help button, which provides access to additional guidance or information about the application's functionality. Here also includes the legend used in the map's color coding that helps ensure clarity and usability for interpreting flood-related information. The map itself can be interacted with to adjust the map's scale for detailed or broader views of the flood map.

The second sprint was focused on setting up the integration of the model's predictions with the water simulation that produced the flood extent maps for the web application. The sprint lasted for three weeks, with the definition of done set as when the process of the model predictions until the uploading of the flood extent maps was already programmed and functional. For the first section of this sprint, the model's predictions are input along with the processed Digital

Elevation Model for the water simulation in HEC-RAS. The goal was to simulate potential flood propagation within Camaligan based on forecasted water levels about the elevation across the area.

The processed DEM was imported into the program's RAS Mapper alongside its projected CRS. The 2D Flow Area of the map is then delineated along the DEM's boundaries, which form the polygon area that represents the floodplain extent of Camaligan. Additionally, the river boundaries were also configured with the inflow of the upstream being mapped to the left side of the river, while the outflow of the downstream was mapped to the right. These boundary conditions were defined through the Unsteady Flow tab of the program, where the upstream is assigned by the stage hydrograph, which is filled in with the six-hourly water level values as generated by the model, while the downstream was set to a normal depth of 0 friction slope.

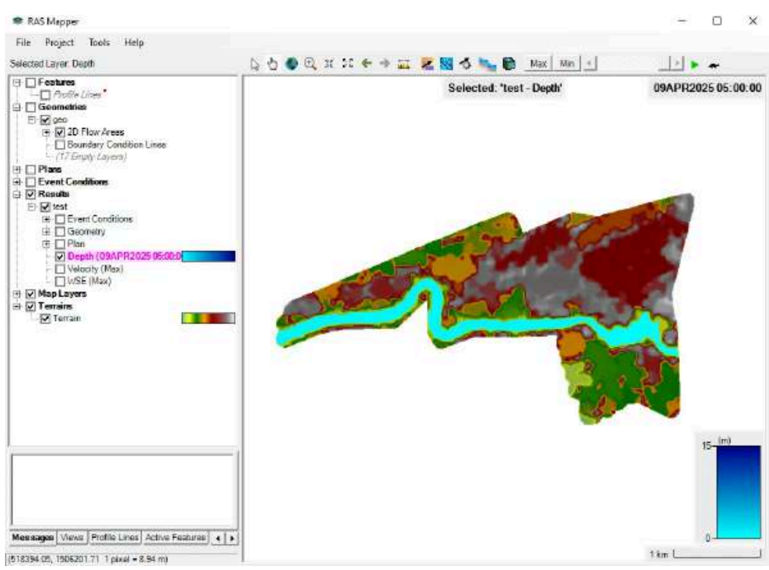


Figure 3.13. **HEC-RAS Simulation**

After setting up the geometry and water level inputs, the simulation plan was executed with output intervals set at one hour, producing six raster maps representing the predicted flood extent for each hour. Simulation runs were automated using HEC-RAS's internal controller via the Compute_CurrentPlan() method. The simulation results can be viewed as seen in Figure 3.13. However, HEC-RAS lacks direct methods for programmatically exporting GeoTIFF files. As a workaround, PyAutoGUI was used to simulate manual interaction with the software interface to export the flood layers as images. This approach introduced a platform dependency, limiting automation to Windows-based systems.

After exporting, elevation values were classified into their respective flood levels and mapped to RGB color codes. The original 32-bit files from HEC-RAS were also converted into 8-bit GeoTIFF files, which were needed for the web application. This conversion was carried out using Python scripts built around the Rasterio package and applied across all six hourly flood extent outputs.

After preparing all files and back-end scripts needed for the web application, the third sprint was initiated to develop the functional web application. This sprint lasted for three weeks, with the definition of done set to be when the web application's all features have been added and are deemed ready for user evaluation. The map functionality was handled using the Mapbox GL JS Framework. The map was first initialized, then the uploaded flood extent maps were loaded in using the addSource and addLayer methods. The flood extent maps for each hour were set to be shown using a dropdown menu by adding an event listener. Each dropdown item corresponds to a specific time frame and is assigned a tileset index ranging from 0 to 5, representing predictions for the upcoming six hours. The selected flood extent map layer is made visible, while all other layers are hidden to maintain clarity. In conjunction with this process, the refreshMonitoredLocations function is triggered to fetch and update the flood level values for all actively monitored locations based on the newly selected prediction hour.

To facilitate adding monitored locations, a search function was developed to assist users in adding locations for flood monitoring. On top of the Mapbox dataset for locations, the critical facilities dataset was added for more localized search results. The implementation was done through GL Geocoder, which allowed integration of additional POIs from a GeoJSON dataset. A local Geocoder function was developed to filter and display these features within the search results, while a MapboxGeocoder instance was initialized using the provided access token, including the localGeocoder function, to enable both global place search and local dataset search.

The monitored locations feature enables users to track predicted flood levels at selected points on the map. This was implemented by making an addMonitored function that will store the place name and coordinates of a selected location. The location details (name, coordinates, flood level, and flood value) are added to the list and marked on the map interface. The fetching of the flood level and value was done by referring to the original 32-bit GeoTiff file using the coordinates. Another function then refreshes the sidebar with the new set of monitored locations. Whenever another flood extent map is selected in the dropdown menu, the refreshMonitoredLocations function is run to fetch the flood level and values of the monitored locations for that flood extent map.

The Nearby Critical Facilities feature was also programmed to update whenever a location is added. This feature displays all critical facilities within a two-kilometer radius using Mapbox' Tilequery API. The API returns a JSON file of all the nearby facilities, and a function was programmed to display these facilities in the Nearby Critical Facilities section of the web page. The list items have event listeners that, when clicked, will trigger the map to center on its location and display the facility's flood level and value. In addition to this, all critical facilities are displayed on the map itself with icons representing the amenity type.

As the web application can be accessed by any device, mobile responsiveness was considered by providing layout adjustments for smaller screens. Instead of a static sidebar overlay, elements such as monitored locations and nearby facilities were programmed to be presented as modals. On another page, flood level reports of per barangay and critical facilities were added to show a general view of the status of Camaligan based on the predictions. The most at-risk barangays are also shown as toasts on the map page. This was done by reading a CSV file generated by scripts that fetch the maximum flood level predicted within each barangay's boundary and the flood level of each critical facility. Other pages include the landing and about pages to show more information and guides about the web application.

In the development of the Predictive Flood Mapping Web Application for Camaligan, incorporating both machine learning and spatial analysis software, hardware specifications, as presented in Table 3.1 below, were deemed essential to ensure smooth performance, efficient data handling, and continuous processing of requests and processes.

Table 3.1. **Hardware Requirements**

Component	Minimum
Development Machine	Processor: Intel i5/i7 or AMD Ryzen 5/7, 6-core RAM: 16 GB Storage: 512 GB SSD GPU: NVIDIA card 8GB VRAM
Server Machine	Processor: Intel Xeon, multi-core RAM: 12 GB, Storage: 1 TB SSD GPU: NVIDIA card

The software requirements presented in Table 3.2 cover a range of options for tools and platforms that were used in developing, deploying, and maintenance of the machine learning models, maps, and web application. The system must be run on Windows-based systems for the exporting of the raster maps from HEC-RAS to work, as the automation scripts use macros. Python was the primary programming language for handling Machine Learning, simulation, and other miscellaneous tasks. For back-end development, frameworks like Flask were used to create APIs to facilitate communication between the models and other components. The front-end side used Figma for the design of the user interface and interactions, and utilized Mapbox GL JS for dynamic map visualizations and a responsive interface across devices. The machine learning models were developed and fine-tuned using Python libraries such as Scikit-learn or Tensorflow. Version control was managed through Git, and the progress of the project was monitored through Notion and FigJam, especially with the implementation of the SCRUM framework. As for spatial analysis software, QGIS was used for modifying geospatial datasets, and HEC-RAS for simulating according to the model's results based on related literature.

Table 3.2. **Software Requirements**

Component	Recommended
Operating System	Windows 10/11
Programming Languages	Python (ML and Scripts), HTML/CSS/JavaScript (Front-end)
Spatial Analysis Software / Libraries	QGIS, HEC-RAS
Frameworks & Libraries	Flask, Figma, MapBox GL JS, TippyJS (Web development) TensorFlow, Keras, Scikit-Learn, Numpy, Pandas (Model Training) HECRASController, pywin32, PyAutoGui, Rasterio, GDAL, requests, boto3 (Simulation and Uploading)
Development Tools	Git/GitHub (Version Control), Notion and FigJam (Project Management)

V. System Usability and Effectiveness Testing through User Surveys

To assess the system's functionality and usability, user surveys were designed and conducted to gauge actual user satisfaction and feedback. These were used to further improve the application and to garner any potential improvements. A user testing phase was conducted involving relevant stakeholders and community members. (See Appendix A for the full survey instrument.) The study targeted individuals from the Camaligan Disaster Risk Reduction and Emergency Services Management Office (CADRRESMO), Central Bicol State University of Agriculture (CBSUA), and DOST-PAGASA, ensuring that key institutional and community perspectives were represented. A total of 22 respondents participated in the survey, with a majority being residents or representatives from the Municipality of Camaligan.

User testing involved a simulation of Typhoon Kristine using the developed system. Participants were asked to assess how accurately the system reflected the real-life flood events that occurred during the typhoon. This simulation was chosen to demonstrate the practical application and predictive performance of the system under realistic conditions.

The survey was structured into six major sections: User Profile – gathered demographic and background information of respondents; Usability and Learnability – assessed how intuitive and user-friendly the system interface was; System Efficiency and Effectiveness – evaluated the system's performance, response time, and functional accuracy; Reliability and Security – examined the perceived trustworthiness and stability of the application; Overall Satisfaction and Recommendations – collected users' overall impressions and suggestions for improvement; Typhoon Kristine Case Comparison – compared the predicted flood simulation results to real-world observations from the typhoon's impact. The evaluation material used is in Appendix D.

Quantitative data were collected through Likert-scale ratings and summarized using descriptive statistics such as mean, frequency, and percentage distribution. Qualitative responses from open-ended items were thematically analyzed to identify common sentiments or areas for improvement. The aggregated findings from this evaluation informed the final iteration and refinement of the system before deployment.

CHAPTER 4

RESULTS AND DISCUSSIONS

This chapter presents the results of the study based on its research objectives. The findings are analyzed and supported by relevant figures, tables, and system screenshots to provide a comprehensive understanding of the system's development and performance. The results are categorized into key areas: data requirements, algorithm implementation and performance evaluation, system development, and usability and effectiveness. Each section details the corresponding findings, demonstrating how the objectives were met and how the system contributes to flood prediction and risk assessment in Camaligan.

Acquired and Processed Data

This section presents the historical flood level and rainfall data gathered from DOST-PAGASA's Bicol River Flood Forecasting and Warning Center (Camaligan Station), as well as spatial mapping data and critical facility information obtained from the Camaligan Municipal Planning Office and the UP Resilience Institute. These datasets formed the foundation for flood prediction modeling.

The historical rainfall and water level datasets were successfully obtained and processed for use in the predictive model. After trimming and aligning, the rainfall dataset began in January 2008 to match the water level dataset. These datasets were merged into a single, consistent format containing key attributes necessary for model input.

Year	Month	Day	Hour	Rainfall	WaterLeve	FWaterLevel
2008	1	1	9	0	1.45	1.81
2008	1	1	10	0	1.63	1.68
2008	1	1	11	0	1.83	1.57
2008	1	1	12	0	1.93	1.47
2008	1	1	13	0	1.95	0
2008	1	1	14	1	0	1.34
2008	1	1	15	0	1.81	1.34
2008	1	1	16	0	1.68	1.35
2008	1	1	17	0	1.57	1.44
2008	1	1	18	0	1.47	0
2008	1	1	19	0	0	1.66
2008	1	1	20	0	1.34	1.62
2008	1	1	21	0	1.34	1.57
2008	1	1	22	0	1.35	1.52
2008	1	1	23	0	1.44	1.44
2008	1	1	0	0	0	1.37
2008	1	1	1	0	1.66	1.3
2008	1	1	2	0	1.62	1.34

Figure 4.1. **Samples from the Final Dataset**

Interpolation effectively addressed missing entries, and numerical formatting ensured compatibility with machine learning tools. The addition of the {FWaterLevel} column enabled supervised learning for future water level predictions, with each value representing a six-hour lead time based on current inputs. The final dataset garnered a total of 153,004 rows, totaling up to 17 years' worth of data. This can be seen in Figure 4.1, in which the first few rows of the final preprocessed dataset are showcased. The training set had a total row count of 122,364, including dates from January 2008 to August 2021. The test data, on the other hand, had a total of 30,592, spanning from August 2021 to January 2025.

For the geospatial component, the critical facilities dataset was successfully compiled from two separate sources and harmonized in terms of structure and Coordinate Reference Systems (CRS). The dataset was imported into QGIS and converted to a matching CRS that is compatible with Mapbox. The final dataset includes properly categorized facilities—such as schools, hospitals, and evacuation centers—across the Camaligan area, now ready for integration into the interactive map visualization component of the system. The final critical facility dataset is then prepared and exported in a CRS-compatible format.

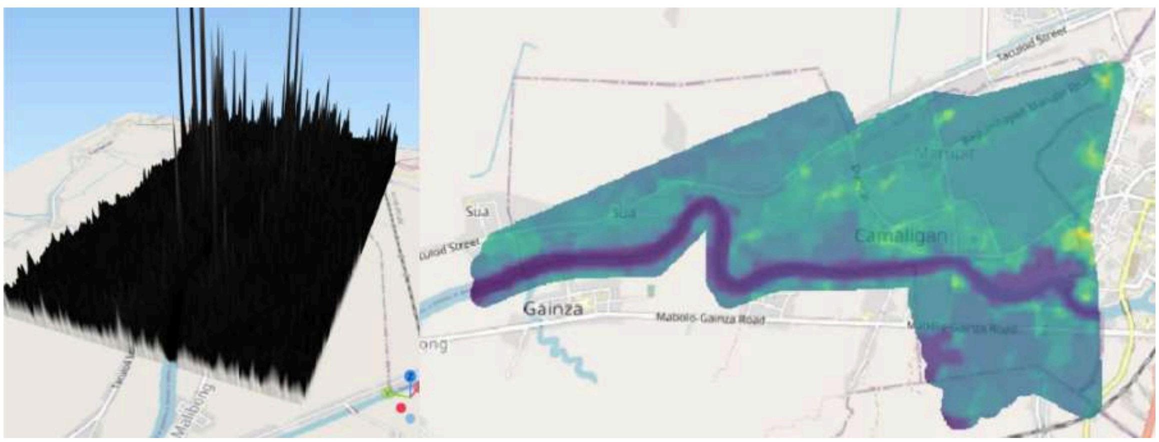


Figure 4.2. **Initial vs Final Digital Elevation Model**

In addition to the preprocessed training data, the Digital Elevation Model was prepared and finalized in QGIS, which is used in the actual flood extent render. The preprocessed DEM was much smoother and more visible compared to the initial gathered DEM, as shown in Figure 4.2. A notable improvement is the visible river line, which is important to the flood extent mapping and unsteady flow simulation. This can be observed with the bluish outline that goes through the terrain map. The terrain's elevation values were also fine-tuned to be normalized and scaled as compared to the spiky and exaggerated values shown in the initial DEM.

Trained and Fine-Tuned Machine Learning Model

This section discusses the machine learning model outputs used to predict flood levels in Camaligan. The final model architecture was developed using a structured NARX dataset generation function. This function constructed a dataset by appending different input lags, thus allowing for more effective time-series forecasting. The model specified the parameters of the input, hidden, and output layers, with a tanh activation function applied to the input and its two hidden layers, while the output layer utilized a linear activation function. The model was trained over 100 epochs with a batch size of 8.

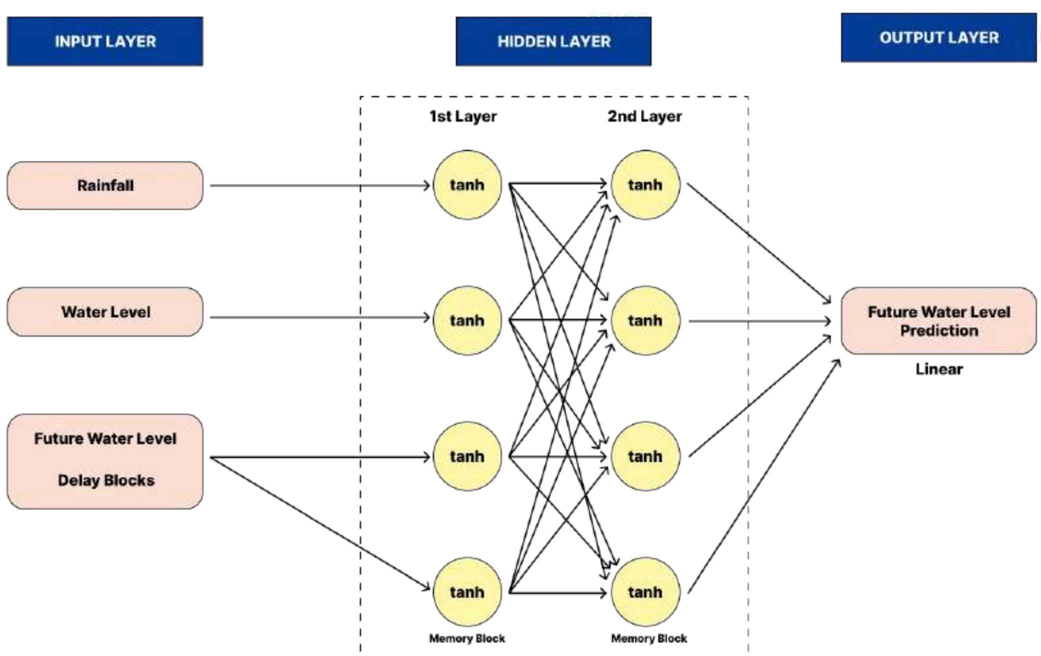


Figure 4.3. **Model Architecture**

Figure 4.3 above presents an overview of the final model architecture. The neural network model developed for water level prediction is composed of three primary components: the input layer, the hidden layers, and the output layer. These components work together to capture temporal dependencies and non-linear relationships within the data, enabling accurate predictions of water levels six hours in advance.

The input layer receives three key factors: rainfall, water level, and the future water level. While rainfall and current water level data serve as the model’s direct inputs, the future water level acts as a reference point for generating input lags following the Nonlinear Auto-Regressive with eXogenous inputs (NARX) model structure. These input lags provide the model with a "memory" of previous states, simulating time-dependent learning. The specified delay blocks define how many past time steps are considered during the prediction process, thus allowing the network to account for temporal patterns within the dataset.

The model incorporates two hidden layers, both employing the hyperbolic tangent (tanh) activation function. This function is well-suited for capturing sinusoidal or oscillating patterns, which are common in environmental and hydrological data. These hidden layers play a critical role in discovering and modeling temporal correlations and non-linear interactions among the 96 generated features from the NARX dataset. The output layer is responsible for producing the model’s final prediction, which is a single floating-point value representing the forecasted water level six hours ahead of the current time step.

III. Model Evaluation Performance

This section presents a summary of the evaluation results for all developed models, with emphasis on the performance and selection of the final predictive model.

As presented in Table 4.1, several model configurations were tested and assessed based on key performance metrics: R^2 (Coefficient of Determination), NSE (Nash–Sutcliffe Efficiency), RMSE (Root Mean Square Error), and prediction latency. The seventh iteration emerged as the most optimal configuration, meeting and exceeding the predefined thresholds—$R^2 \geq 0.8$, NSE ≥ 0.8, RMSE < 0.2, and prediction latency < 5 minutes. Although a subsequent iteration also satisfied the criteria, the seventh yielded the best overall performance.

Table 4.1. **Model Evaluation Results**

Model Versions	Epochs	Batch Size	R^2	NSE	RMSE	Prediction Latency (seconds)
1	100	6	-0.387	-0.387	0.439	1.32
2	100	32	0.23	0.23	0.38328	1.54
3	100	16	0.48	0.48	0.36288	1.56
4	100	8	0.54	0.54	0.34136	2.04
5	100	8	0.84	0.84	0.20136	3.13
6	100	8	0.85	0.85	0.19234	3.11
7	100	8	0.87	0.87	0.17807	2.93
8	100	8	0.84	0.84	0.20025	3.95

This model achieved an R^2 of 0.87, indicating that 87% of the variance in observed water levels is accurately explained by the model's predictions. The NSE of 0.87 further confirms its high predictive reliability, demonstrating that the model closely tracks actual data trends. Additionally, the RMSE of 0.17807 meters reflects minimal average prediction error—less than 18 centimeters—which is suitable for practical flood forecasting and preparedness efforts. The prediction latency of just 2.93 seconds also highlights the model's efficiency, enabling near real-time application for early warning systems.

Collectively, these results affirm that the finalized model is both accurate and computationally efficient, making it a strong candidate for deployment in the web-based flood prediction system. Its reliable performance ensures that it can effectively support flood risk management and decision-making for the Municipality of Camaligan.

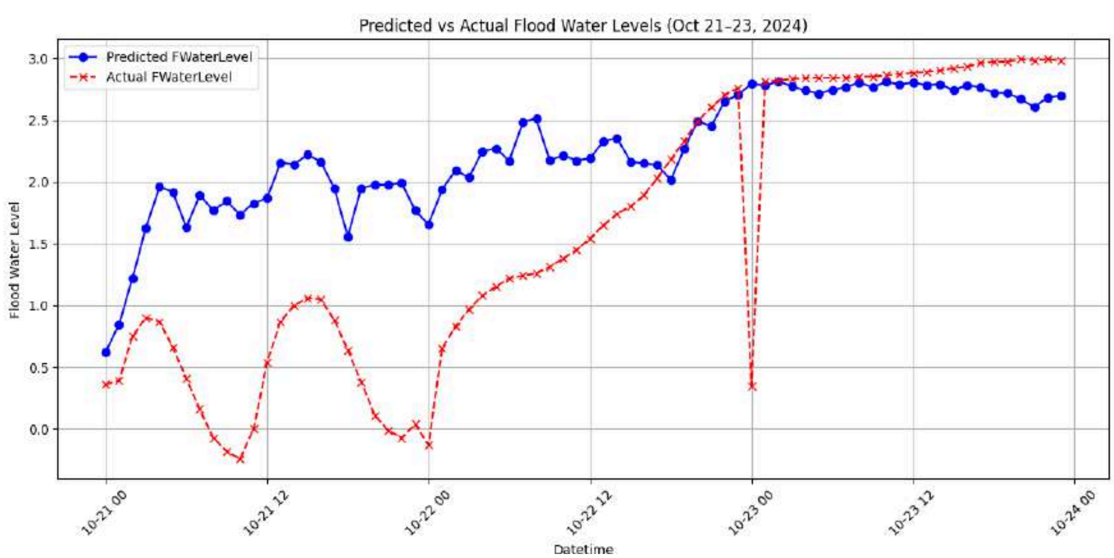


Figure 4.4. **Predicted vs Actual Water Levels for Typhoon Kristine**

The finalized model was tested using a simulated scenario of Typhoon Kristine, focusing on the period when the typhoon was most active—October 21 to October 23, 2024, as shown in a graph in Figure 4.4. The test aimed to assess the model's performance in forecasting flood levels under real-world extreme weather conditions. On the initial day of the simulation, the model tended to overpredict water levels, but its outputs gradually stabilized and aligned more closely with actual observations in the subsequent days.

The relative error between the predicted and actual water levels ranged from 0.1 meters to a maximum of 1.74 meters, suggesting that while the model exhibited some overestimation, it was still able to capture the timing and magnitude of peak flood levels with reasonable accuracy. This performance demonstrates the model's potential to support early warning efforts during major weather events, although further tuning may improve prediction precision for the initial stages of extreme weather scenarios.

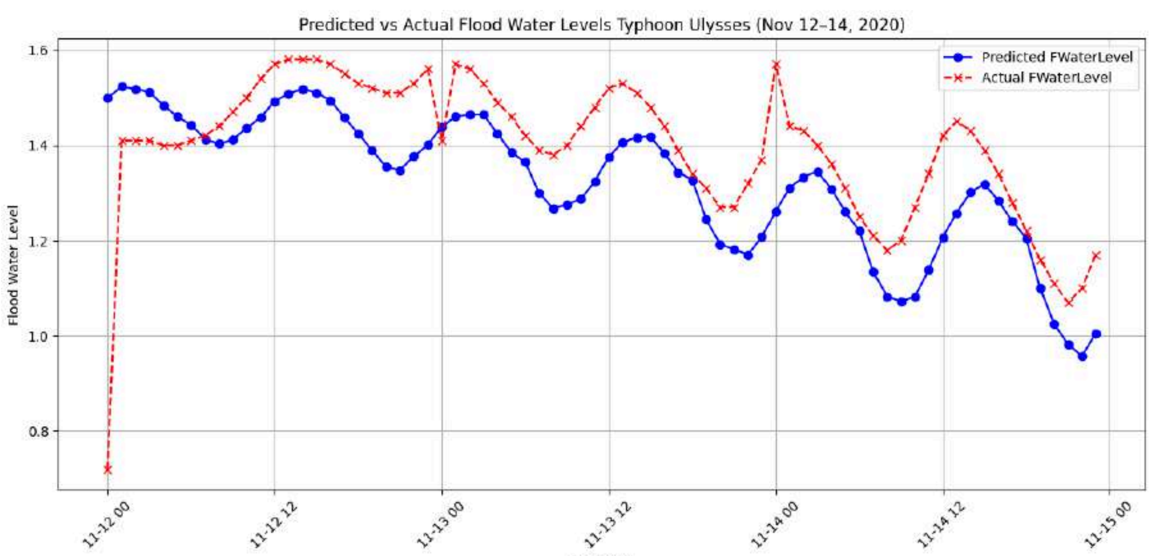


Figure 4.5. **Predicted vs Actual Water Levels for Typhoon Ulysses**

Another simulation was conducted for Typhoon Ulysses. This occurred back on November 12 to 14, 2020, where Camaligan was reportedly affected by the river overflow. As shown in Figure 4.5, the prediction result was relatively close to the sinusoidal pattern exhibited by the actual water level, showing that the model is capable, even if the water level forecast is not as high as compared to Typhoon Kristine.

Developed FLOWS Web Application

This section highlights the GIS integration and describes the design and implementation of the web application that visualizes the flood extent map for Camaligan and its core functionalities.

As part of the final output for flood prediction, the raster-based flood extent maps generated through HEC-RAS simulations were classified and visually styled to reflect different levels of flood severity. These severity levels—low, moderate, and high—were defined based on the Metropolitan Manila Development Authority (MMDA)'s official flood gauge standards and validated in consultation with the Camaligan Disaster Risk Reduction and Emergency Services Management Office (CADRRESMO).

The original files were in 32-bit format, which is incompatible with Mapbox, as it only supports 8-bit GeoTIFFs. Therefore, all raster files were successfully converted from 32-bit to 8-bit format.

Figure 4.6. **Sample Converted 8bit GeoTIFF**

Additionally, the color scheme of the maps was adjusted to match the standardized color representation used as aforementioned, with the final converted and colored GeoTIFF maps being visually similar to the sample in Figure 4.6. These preprocessing steps ensured accurate and visually coherent rendering of the flood maps on the platform. The entire processing pipeline was implemented through Python scripts utilizing the Rasterio library, which efficiently handled both the format conversion and the application of the appropriate color mapping.

The developed web application consists of four primary pages – the Landing Page, the Flood Extent Map Page, the Reports Page, and the About Page. The landing page serves as the first default screen that loads when the website is accessed. The screen contains a brief introduction about the project and the website itself, with buttons that redirect to the flood extent map page.

Figure 4.7. **Flood Extent Map Page**

Figure 4.7 Flood Extent Map page displays the map of Camaligan through Mapbox. The flood extent is also visible with the use of colored indicators on the map. The indicators are described in the bottom right corner of the page, where the legend of the color values is shown. The map also consists of the critical facility markers and location icons that are fetched from the critical facility dataset. The map itself can be zoomed in, panned, and rotated for a 3D view.

At the top right corner of the map is a feature sidebar which includes the Prediction Dropdown, Search Function and Monitored Locations, and Nearby Critical Facilities.

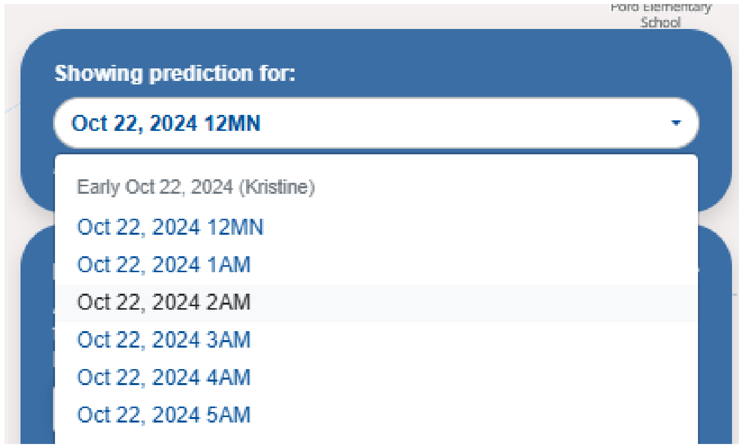


Figure 4.8. **Prediction/Flood Extent Drop Down Feature**

The application supports visualization of flood predictions through a dropdown menu, as shown in Figure 4.8, that allows users to switch between hourly flood extent maps. Each dropdown item corresponds to a specific time frame of six hours. Choosing one of the predictions automatically loads it onto the map for public viewing.

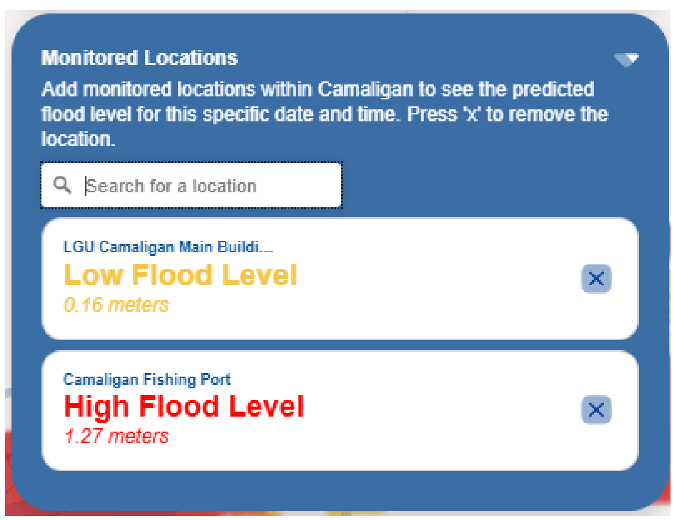


Figure 4.9. **Monitored Locations Feature**

The monitored locations, as shown in Figure 4.9, enable users to track predicted flood levels at selected points on the map. A search bar is located at the top of this feature, where users can input a location. A recommendation dropdown is shown based on the user's input for easier accessibility. The searched location is automatically added to the monitored location list shown at the bottom portion of the search bar. This feature ensures that selected locations are added to the monitoring list. The corresponding flood levels are also shown with matching font colors to match the flood level legend. These monitored locations can also be removed with the remove icon on the right portion of the location bar.

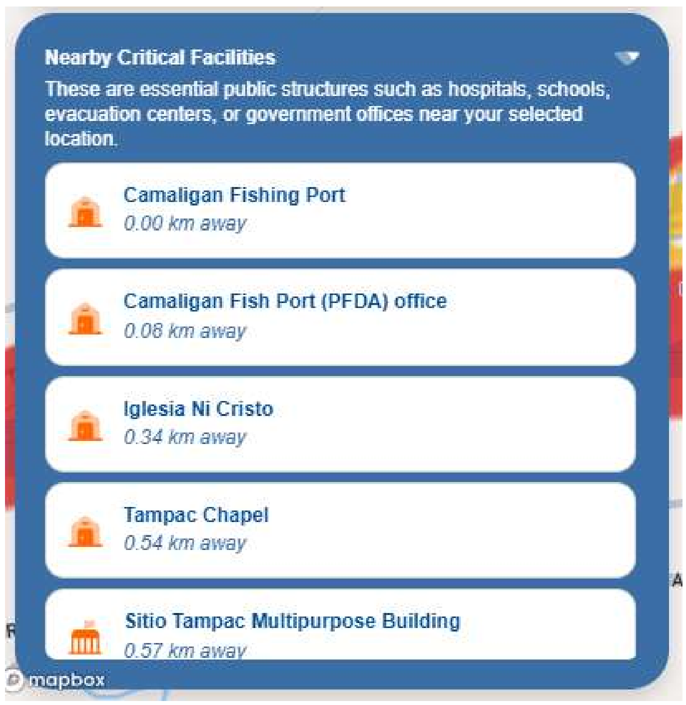


Figure 4.10. **Nearby Critical Facilities Feature**

Upon choosing a monitored location, the application dynamically displays the critical facilities near the inputted monitored location list in the span of a two-kilometer radius (as shown in Figure 4.10). Each critical facility is also displayed on the map with its name and an icon representing its respective category (e.g., school, evacuation center, hospital, government office).


Flood Level Reports

Barangay Flood Levels for October 23, 2024

These are based on the max flood level that was predicted within the boundaries of the barangay, including the river section.

Barangay	12 AM	1 AM	2 AM	3 AM	4 AM	5 AM
Sua	High	High	High	High	High	High
Marupit	High	High	High	High	High	High
Sto. Tomas	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate
San Juan - San Ramon	High	High	High	High	High	High
San Francisco	High	High	High	High	High	High
Tarosanan	High	High	High	High	High	High

Critical Facilities Flood Levels for October 23, 2024

These are essential public structures such as hospitals, schools, evacuation centers, or government offices near your selected location.

Government | Evacuation center | School | Hospital | Place of worship | Miscellaneous

Hospital

Critical Facility	12 AM	1 AM	2 AM	3 AM	4 AM	5 AM
San Juan San Ramon Barangay Health Center	High	High	High	High	High	High
San Mateo Health Center (Rent)	None	None	None	None	None	None
Municipal Health Office	Low	Low	Low	Low	Low	Low
Birthing Center	Moderate	Moderate	Moderate	Moderate	Moderate	Low
Municipal Nutrition Action Office	Moderate	Low	Moderate	Moderate	Low	Low
Sto.Domingo Barangay Health Center (1F)	High	High	High	High	High	High

Figure 4.11. **Flood Level Reports Page**

Figure 4.11 displays the Flood Level Reports page, which is accessible via the ‘Reports’ option in the header. This interface provides an hourly overview of flood risk levels for each barangay, along with the corresponding risk levels for nearby critical facilities.

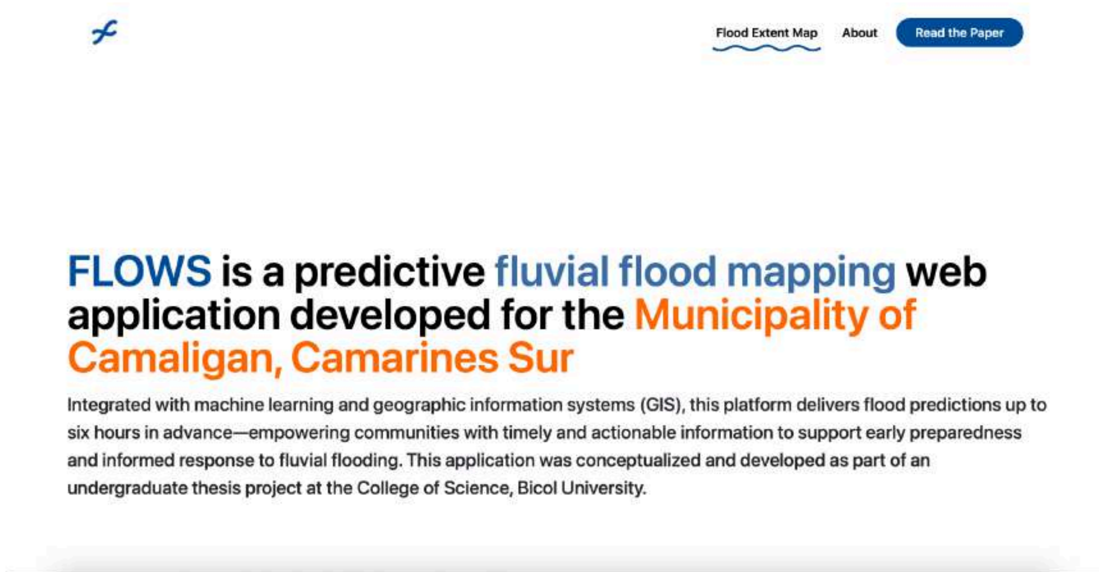


Figure 4.12. **About Page**

The About page (Figure 4.12) displays the web application's credentials and proper recognitions. Scrolling down the page shows an instructional video on how to use the core functionalities of the web application and a brief introduction to the project's cause and objectives. At the bottom of the page, the researcher's contact information can be found.

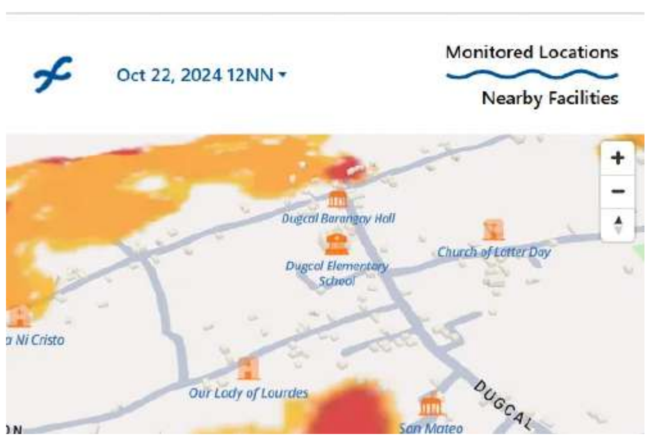


Figure 4.13. **Mobile Layout**

Given that the application may be accessed across a range of devices, mobile responsiveness was a key consideration in the interface design. Layout adjustments, as shown in Figure 4.13, were implemented to enhance usability on smaller screens. Instead of a static sidebar overlay, elements such as monitored locations and nearby facilities are presented as modals.

Assessed System Usability and Effectiveness

This section presents the results of system testing and user satisfaction surveys, which were conducted using a simulation of the flooding caused by Typhoon Kristine. The evaluation focuses on the web application's usability, efficiency, and effectiveness in delivering predictive flood visualizations. Feedback was gathered from stakeholders and end users to assess the system's performance in replicating real-world conditions and its potential to support early decision-making and disaster preparedness.

This user satisfaction was measured through a survey based on ISO/IEC 25010 usability and user experience guidelines, focusing on effectiveness, efficiency, and satisfaction. The survey was conducted among the recognized stakeholders, such as LGU Officials, DRR Officers, Emergency Responders, Community Members, Researchers, and Programmers.

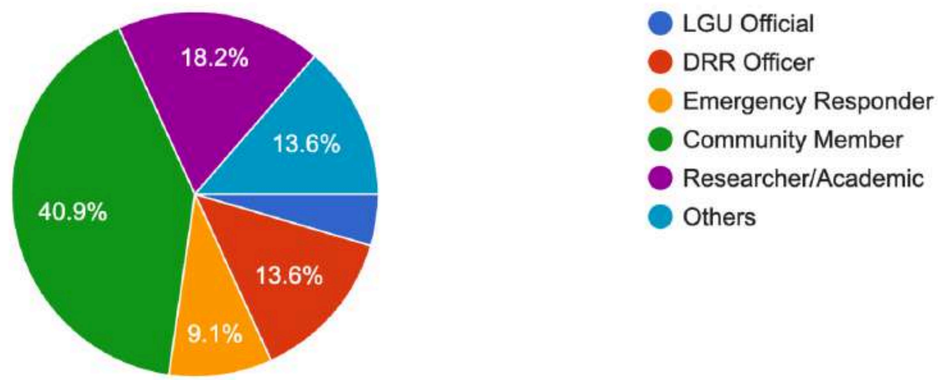


Figure 4.14. **Distribution of Respondents by Role**

Figure 4.14 above shows the distribution of respondents by role within the community organization who took the survey. A total of 22 responses were collected, comprising LGU officials (4.6%), DRR officers (13.6%), emergency responders (9.1%), community members (40.9%), researchers/academics (18.2%), and others (13.6%).

Table 4.2 summarizes the usability and learnability survey results for the FLOWS web application. The findings indicate strong user satisfaction, with an average rating ranging from 4.45 to 4.82 across key metrics. The highest rating was recorded for simplicity and efficiency of accessing flood predictions (4.82), followed by ease of navigation and visual clarity of the interface (4.64). Over 70% of the respondents rated the system as highly usable and intuitive. Based on these results, it suggests that FLOWS met its objective of providing a user-friendly platform for flood prediction and visualization.

Table 4.2. **Usability and Learnability Survey Results**

User Satisfaction Metric	Percentage of Rating			Average Rating
	3	4	5	
System is easy to navigate and use	9.1%	18.2%	72.7%	4.64
Interface is user-friendly and visually clear	9.1%	18.2%	72.7%	4.64
Learned to use the system quickly	13.6%	27.3%	59.1%	4.45
System provides clear instructions and labels	9.1%	22.7%	68.2%	4.59
Accessing flood predictions is simple and efficient	0.0%	18.2%	81.8%	4.82

Table 4.3 below is a summary of system efficiency and effectiveness survey results. Respondents rated the system high across all measured metrics, with an average score ranging from 4.27 to 4.77. The highest satisfaction was observed in the clarity of flood hazard level distinctions (4.77) and the system's smooth and fast operation (4.73). Accurate flood prediction was perceived a slightly lower average rating of 4.27. Overall, the results suggest that FLOWS delivers timely updates, effective flood visualization, and efficient performance.

Table 4.3. **System Efficiency and Effectiveness Results**

User Satisfaction Metric	Percentage of Rating			Average Rating
	3	4	5	
System provides accurate flood predictions	9.1%	54.5%	36.4%	4.27
System updates predictions in a timely manner	0.0%	63.6%	36.4%	4.36
System effectively integrates visualization for flood mapping	4.5%	31.8%	63.6%	4.59
Flood hazard levels (low, medium, high) are clearly distinguishable	4.5%	13.6%	81.8%	4.77
System loads quickly and operates smoothly	4.5%	18.2%	77.3%	4.73

Table 4.4 summarizes the reliability and security survey results, showing strong positive feedback, with average ratings from 3.32 to 4.59 across evaluated aspects. Respondents indicated that the system provides reliable data with minimal errors (4.32), functions consistently without unexpected crashes (4.36), and ensures data security while in use (4.59). These results demonstrate that FLOWS meets critical standards for system stability and user trust.

Table 4.4. **Reliability and Security**

User Satisfaction Metric	Percentage of Rating			Average Rating
	3	4	5	
System provides reliable data without frequent errors	9.1%	50.0%	40.9%	4.32
system functions consistently without unexpected crashes	4.5%	54.5%	40.9%	4.36
Data and information feel secure while using the system	4.5%	31.8%	63.6%	4.59

Table 4.5 below shows the user feedback comparing the FLOWS system's predictions with actual occurrences during Typhoon Kristine in Camaligan. High average ratings for the alignment of predicted flood events with actual occurrences (4.35) and the consistency of identified high-risk areas (4.30) suggest a reasonable degree of accuracy in the system's performance. The perceived potential of FLOWS to improve early warning, preparation, and evacuation planning garnered the highest average of 4.60, highlighting user confidence in its future utility for disaster preparedness based on its observed performance. Overall, the findings suggest that FLOWS is viewed favorably by users as a potentially valuable and relatively accurate tool in Camaligan.

Table 4.5. **Typhoon Kristine Case Comparison**

User Satisfaction Metric	Percentage of Rating			Average Rating
	3	4	5	
Alignment of FLOWS-predicted flood levels with actual reported water levels and flood-affected areas in Camaligan during Typhoon Kristine	0.0%	65.0%	35.0%	4.35
Consistency of FLOWS-identified high flood-risk areas and critical facilities with actual affected locations during Typhoon Kristine	0.0%	70.0%	30.0%	4.30
Perceived potential of FLOWS to improve early warning, evacuation planning, and resource allocation in Camaligan if deployed before Typhoon Kristine	0.0%	40.0%	60.0%	4.60

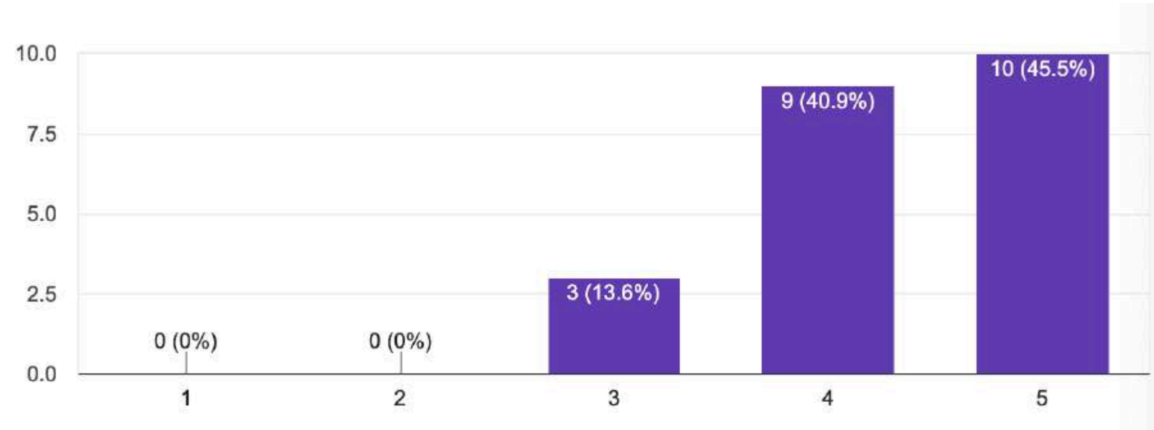


Figure 4.15. **Overall User Satisfaction Level**

Figure 4.15 shows that user satisfaction for FLOWS indicates a generally positive reception. Among the 22 respondents who reported high levels of satisfaction, 40.9% rated a 4, and 45.5% gave the highest rating of 5. Only 13.6% reported a neutral rating of 3, while no respondents expressed dissatisfaction with ratings of 1 or 2. This distribution suggests that users are largely pleased with the predictive fluvial flood mapping web application.

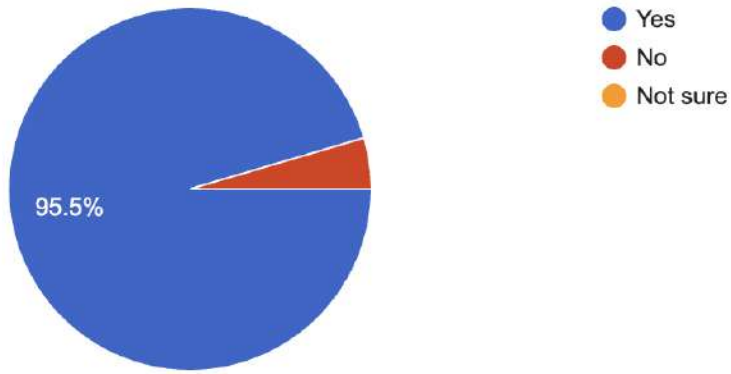


Figure 4.16. **Recommendation Rate**

Figure 4.16 shows the recommendation rate of the system. The majority (95.5%) of the 22 respondents indicate that they would recommend the system to others. This exceptionally high recommendation underscores a strong positive perception and a high level of satisfaction with the system among its users. The minimal percentage of respondents who would not recommend (4.5%) suggests that any potential drawbacks or areas of improvement are not significant enough to deter endorsement of the system.

Based on the conducted user satisfaction surveys and simulation testing using Typhoon Kristine as a reference event, the FLOWS web application demonstrated a high level of usability, effectiveness, and efficiency. Respondents highlighted the system's ease of use, user-friendly interface, accurate flood predictions, and responsive, clear visualization of flood scenarios. These strengths affirmed that the system successfully met its goal of providing an accessible and informative platform for community-based flood risk awareness and preparedness.

Despite the positive reception, user feedback also pointed toward areas for further enhancement. Suggestions included integrating real-time rainfall data to enhance prediction timeliness, enabling barangay-level statistics for more localized insights, and introducing customizable simulation parameters for more flexible use. Minor improvements to the user interface, such as providing additional information on flood visualization, were also recommended. In addition, participants emphasized the importance of continued model validation using real-world extreme flood events to maintain and improve predictive accuracy.

These insights offer a clear pathway for future development. Enhancing FLOWS with real-time data integration, advanced customization features, and expanded validation methods would significantly strengthen its role as a practical decision-support tool for local governments, emergency responders, and communities. The feedback collected reinforces FLOWS' potential not only as a research output but as a scalable early warning platform capable of contributing to disaster risk reduction efforts in flood-prone municipalities.

The successful implementation of ISO/IEC 25010-aligned evaluation metrics confirms that the system is efficient, reliable, and user-friendly.

CHAPTER 5

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

Summary

Flooding is one of the most frequent and destructive natural disasters in the Philippines, with fluvial (river-induced) floods posing a recurring threat to low-lying communities such as the Municipality of Camaligan, Camarines Sur. Despite the presence of flood monitoring stations and hazard maps, existing systems often lack predictive capabilities and do not provide localized, timely, or accessible flood warnings. This study aimed to address this gap by developing a Predictive Fluvial Flood Mapping Web Application that forecasts flood levels six hours in advance and visualizes the results through a flood-extent map.

This research was guided by five key objectives: (1) To gather and process historical flood level and rainfall data, mapping, and critical facilities from relevant sources; (2) To train a machine learning model capable of predicting flood levels based on historical flood level and rainfall; (3) To evaluate the model's performance using regression metrics; (4) To develop a user-friendly web application that visualizes the flood extent map; and (5) To assess the system's usability and effectiveness through user testing and surveys.

The methodology involved collecting and processing hydrological data from DOST-PAGASA, and was used to train an NARX-based neural network implemented using TensorFlow, Keras, and Scikit-learn. The model's outputs were incorporated into flood simulations using HEC-RAS and processed into raster maps, which were visualized via Mapbox GL JS on the developed web platform. Automation scripts handled end-to-end system operations—from prediction to map visualization and cloud upload. Evaluation metrics such as R^2 (0.87), NSE (0.87), RMSE (0.17807), and latency (2.93s) demonstrated the model's reliability and efficiency.

The final output is a fully functional and mobile-responsive web application that allows users to search for specific locations, monitor these locations and their predicted flood levels, and view nearby critical facilities. Results from user surveys confirmed the system's usability and practical value for disaster preparedness and local decision-making. Overall, the study demonstrates the potential of integrating machine learning, GIS, and web technologies to build more proactive and localized flood early warning systems.

Conclusions

Following the accomplishment of the research objectives and a thorough evaluation of the system's performance, the conclusions derived from the findings are presented below.

1. Relevant datasets, including historical flood levels, rainfall, Digital Elevation Models, and critical facilities, were successfully collected, preprocessed, and standardized for the modelling and development methods.
2. A NARX-based neural network model was successfully trained using 96 features derived from past rainfall and water level data, achieving high performance, which indicates strong predictability.

3. The final prediction model met the desired threshold metrics for accuracy with an R2 of 0.87, NSE of 0.87, and an RMSE of 0.17807, and a prediction latency of 2.93 seconds, confirming the model's efficiency and suitability for real-time applications.
4. A functional web application was developed for visualization and predictive mapping, featuring search functionality, monitored locations with nearby critical facilities, barangay-level reports, and mobile responsiveness.
5. System evaluation through user surveys showed that FLOWS effectively met its goal of providing a usable, efficient, and practical platform for flood prediction and visualization, aiding in early decision-making and community preparedness.

Recommendations

Based on the conclusions drawn, the following recommendations are proposed to enhance the system's functionality, ensure its sustainability, and guide future developments in line with the study's objectives.

1. Future studies may consider integrating actual real-time data streams and pluvial flood data for improved prediction accuracy, particularly in urban areas prone to surface flooding, or including other points for the water level prediction for larger river networks.
2. Explore other machine learning algorithms to improve temporal prediction performance and support multi-point forecasting.

3. Consider using other simulation software or other methods that can be programmatically achieved without the need for macros, so the automation will not be limited to Windows.
4. Implement automated alert systems (e.g., SMS or push notifications), additional statistics, and enhance the system with multi-language support to broaden accessibility for diverse communities.
5. Conduct further testing with local government units and emergency responders, and deploy pilot implementations during active typhoon seasons to assess impact in real-time settings.

BIBLIOGRAPHY

Abdul Rahman, A. J., & Ramli, N. A. (2024). Flood Prediction Using Artificial Neural Networks: A Case Study in Temerloh, Pahang. Qeios. https://doi.org/10.32388/tuz29y

Abedin, S. J. H., & Stephen, H. (2019). GIS Framework for Spatiotemporal Mapping of Urban Flooding. *Geosciences 2019, Vol. 9, Page 77, 9*(2), 77. https://doi.org/10.3390/GEOSCIENCES9020077

Ahmed, S., El-Magd, A., Pradhan, B., & Alamri, A. (n.d.). Machine learning algorithm for flash flood prediction mapping in Wadi El-Laqeita and surroundings, Central Eastern Desert, Egypt. https://doi.org/10.1007/s12517-021-06466-z/Published

Alcantara, J. C., & Christopher, J. (2019). *Briefing Note Regional Fellowship Program Overview of the Societal Impacts of Floods in the Philippines*.

Al-Rawas, G., Nikoo, M. R., Al-Wardy, M., & Etri, T. (2024). A Critical Review of Emerging Technologies for Flash Flood Prediction: Examining Artificial Intelligence, Machine Learning, Internet of Things, Cloud Computing, and Robotics Techniques. In Water (Switzerland) (Vol. 16, Issue 14). *Multidisciplinary Digital Publishing Institute (MDPI)*. https://doi.org/10.3390/w16142069

Anusha, N., & Bharathi, B. (2020). Flood detection and flood mapping using multi-temporal synthetic aperture radar and optical data. *Egyptian Journal of Remote Sensing and Space Science, 23*(2), 207–219. https://doi.org/10.1016/j.ejrs.2019.01.001

Asian Disaster Reduction Center（ADRC）. (n.d.). Retrieved September 26, 2024, from https://www.adrc.asia/nationinformation.php?NationCode=608&Lang=en

Asokan, A. (n.d.). PREDICTING FLOOD USING ARTIFICIAL NEURAL NETWORKS. http://www.ripublication.com

Atwii, F., Sandvik, K. B., Kirch, L., Mucke, P., Paragi, B., Radtke, K., Schneider, S., & Weller, D. (2022). *World Risk Report 2022 Focus: Digitalization* (L. Cohen, Trans.). https://weltrisikobericht.de/worldriskreport/

Bates, P. D. (2022). Flood Inundation Prediction. Annual Review of Fluid Mechanics Downloaded from Www.Annualreviews.Org. *Guest, 23*, 27. http://dx.doi.org/10.1146/annurev-fluid-030121-113138

Bentivoglio, R., Isufi, E., Jonkman, S. N., & Taormina, R. (2022). Deep learning methods for flood mapping: a review of existing applications and future research directions. In Hydrology and Earth System Sciences (Vol. 26, Issue 16, pp. 4345–4378). *Copernicus GmbH*. https://doi.org/10.5194/hess-26-4345-2022

Bentoso, L. D., Juan, E. O., Brosas, D. G., Paragas, J. R., Nuevas, L. K., & Velarde, M. W. C. (2021). Web-Based Solution for Flood Warning Decision Support in the Province of Leyte, Philippines. *ICRACOS 2021 - 2021 3rd International Conference on Research and Academic Community Services: Sustainable Innovation in Research and Community Services for Better Quality of Life towards Society 5*, 185–190. https://doi.org/10.1109/ICRACOS53680.2021.9701990

Berkhahn, S., Fuchs, L., & Neuweiler, I. (2019). An ensemble neural network model for real-time prediction of urban floods. *Journal of Hydrology*, *575*, 743–754. https://doi.org/10.1016/J.JHYDROL.2019.05.066

Byaruhanga, N., Kibirige, D., Gokool, S., & Mkhonta, G. (2024). Evolution of Flood Prediction and Forecasting Models for Flood Early Warning Systems: A Scoping Review. In Water (Switzerland) (Vol. 16, Issue 13). *Multidisciplinary Digital Publishing Institute (MDPI)*. https://doi.org/10.3390/w16131763

Cabrera, J. S., & Lee, H. S. (2019). Flood-prone area assessment using GIS-based multi-criteria analysis: A case study in Davao Oriental, Philippines. *Water (Switzerland)*, *11*(11). https://doi.org/10.3390/w11112203

Cadiz, N. R. (2018). UP NOAH in Building Resilient Philippines; Multi-hazard and Risk Mapping for the Future. *Procedia Engineering*, *212*, 1018–1025. https://doi.org/10.1016/J.PROENG.2018.01.131

Cheong, S. M., Sankaran, K., & Bastani, H. (2022). Artificial intelligence for climate change adaptation. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *12*(5), e1459. https://doi.org/10.1002/WIDM.1459

Climate Adaptation Platform. (n.d.). *Philippines: working towards climate adaptation | PreventionWeb*. Retrieved September 26, 2024, from https://www.preventionweb.net/news/vulnerable-philippines-working-towards-climate-adaptation

Dano, U. L., Balogun, A. L., Matori, A. N., Yusouf, K. W., Abubakar, I. R., Mohamed, M. A. S., Aina, Y. A., & Pradhan, B. (2019). Flood Susceptibility Mapping Using GIS-Based Analytic Network Process: A Case Study of Perlis, Malaysia. *Water 2019, Vol. 11, Page 615*, *11*(3), 615. https://doi.org/10.3390/W11030615

Deutsche Gesellschaft für Internationale Zusammenarbeit. (n.d.). Local Flood Early Warning Systems; Mozambique & the Philippines. *Global Initiative on Disaster Risk Management I*. Retrieved September 26, 2024, from https://gidrm.net/user/pages/get-started/resources/files/GIDRM_LFEWS_Mozambique_Philippines.pdf

El Naqa, I., & Murphy, M. J. (2015). What Is Machine Learning? *Machine Learning in Radiation Oncology*, 3–11. https://doi.org/10.1007/978-3-319-18305-3_1

El-Haddad, B. A., Youssef, A. M., Pourghasemi, H. R., Pradhan, B., El-Shater, A. H., & El-Khashab, M. H. (2021). Flood susceptibility prediction using four machine learning techniques and comparison of their performance at Wadi Qena Basin, Egypt. *Natural Hazards*, *105*(1), 83–114. https://doi.org/10.1007/S11069-020-04296-Y/FIGURES/9

Frege, I. A., Befus, E., Bih, K. B., Fawwad, M., Ide, T., Kiefer, P., Kienzl, P., Klüppel, D., Korte, L., Kortendieck, M., Moloinyane, S., Peters, K., Radtke, K., Schneider, S., & Weller, D. (2024). *World Risk Report 2024 Focus: Multiple Crises* (K. Budelmann, C. Bücher, S. Stein, & N. Grünhagen-Rösler, Trans.). https://weltrisikobericht.de/worldriskreport/

Frege, I. A., Bradshaw, S., Dijkzeul, D., Funke, C., Kienzl, P., Küsters, K., Masusch, L., Nasreen, M., Radtke, K., Schneider, S., Thielbörger, P., Weller, D., Wiegers, O., Yaman, D. K., & Zennig, K. (2023). *World Risk Report 2023 Focus: Diversity* (A. Carstensen, S. Kirsch, & K. Schieritz, Trans.). https://weltrisikobericht.de/worldriskreport/

Garcia, F. C. C., Retamar, A. E., & Javier, J. C. (2016). A real time urban flood monitoring system for metro Manila. *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, *2016-January*. https://doi.org/10.1109/TENCON.2015.7372990

Garcia, L. N. (2022, September 1). *Rise beyond the tide*. https://kalahi.dswd.gov.ph/press/features/item/57-rise-beyond-the-tide

Hadi, F. A. A., Sidek, L. M., Salih, G. H. A., Basri, H., Sammen, S. S., Dom, N. M., Ali, Z. M., & Ahmed, A. N. (2024). Machine learning techniques for flood forecasting. *Journal of Hydroinformatics, 26*(4), 779–799. https://doi.org/10.2166/hydro.2024.208

Hakim, D. K., Gernowo, R., & Nirwansyah, A. W. (2024). Flood prediction with time series data mining: Systematic review. In Natural Hazards Research (Vol. 4, Issue 2, pp. 194–220). *KeAi Communications Co*. https://doi.org/10.1016/j.nhres.2023.10.001

Hidayat, N. (2023). Flood Disaster Detection Based on Rainfall Using Random Forest Algorithm. In Future Computer Science Journal (FCSJ) (Vol. 1, Issue 2). http://asasijournal.id/index.php/fcsjhttp://doi.org/10.xxxxx/fcsj.xxxx.xxx

Hou, J., Nie Zhou, Chen, G., Huang, M., & Bai, Guangbi. (2021). Rapid forecasting of urban flood inundation using multiple machine learning models. *Natural Hazards*, *108*, 2335–2356. https://doi.org/10.1007/s11069-021-04782-x

Ighile, E. H., Shirakawa, H., & Tanikawa, H. (2022). Application of GIS and Machine Learning to Predict Flood Areas in Nigeria. Sustainability 2022, Vol. 14, Page 5039, 14(9), 5039. https://doi.org/10.3390/SU14095039

Jie, M. X., Chen, H., Xu, C. Y., Zeng, Q., & Tao, X. E. (2016). A comparative study of different objective functions to improve the flood forecasting accuracy. *Hydrology Research,* *47*(4), 718–735. https://doi.org/10.2166/nh.2015.078

Jones, A., Kuehnert, J., Fraccaro, P., Meuriot, O., Ishikawa, T., Edwards, B., Stoyanov, N., Remy, S. L., Weldemariam, K., & Assefa, S. (2023). AI for climate impacts: applications in flood risk. *Npj Climate and Atmospheric Science 2023 6:1*, *6*(1), 1–8. https://doi.org/10.1038/s41612-023-00388-1

Karim, F., Armin, M. A., Ahmedt-Aristizabal, D., Tychsen-Smith, L., & Petersson, L. (2023). A Review of Hydrodynamic and Machine Learning Approaches for Flood Inundation Modeling. *Water 2023, Vol. 15, Page 566*, *15*(3), 566. https://doi.org/10.3390/W15030566

Khan, M., Khan, A. U., Ullah, B., & Khan, S. (2024). Developing a machine learning-based flood risk prediction model for the Indus Basin in Pakistan. *Water Practice and Technology,* *19*(6), 2213–2225. https://doi.org/10.2166/wpt.2024.151

Khoirunisa, N., Ku, C. Y., & Liu, C. Y. (2021). A GIS-Based Artificial Neural Network Model for Flood Susceptibility Assessment. International Journal of Environmental Research and Public Health 2021, Vol. 18, Page 1072, 18(3), 1072. https://doi.org/10.3390/IJERPH18031072

Lee, J., & Kim, B. (2021). Scenario-based real-time flood prediction with logistic regression. *Water (Switzerland), 13*(9). https://doi.org/10.3390/w13091191

Li, J., Zheng, A., Guo, W., Bandyopadhyay, N., Zhang, Y., & Wang, Q. (2023). Urban flood risk assessment based on DBSCAN and K-means clustering algorithm. *Geomatics, Natural Hazards and Risk, 14*(1). https://doi.org/10.1080/19475705.2023.2250527

Lim, M. B. B., Lim, H. R., & Piantanakulchai, M. (2019). Flood evacuation decision modeling for high risk urban area in the Philippines. *Asia Pacific Management Review, 24*(2), 106–113. https://doi.org/10.1016/j.apmrv.2019.01.001

Luu, C., Bui, Q. D., Costache, R., Nguyen, L. T., Nguyen, T. T., Van Phong, T., Van Le, H., & Pham, B. T. (2021). Flood-prone area mapping using machine learning techniques: a case study of Quang Binh province, Vietnam. *Natural Hazards*, *108*(3), 3229–3251. https://doi.org/10.1007/S11069-021-04821-7/METRICS

Madhuri, R., Sistla, S., & Srinivasa Raju, K. (2021). Application of machine learning algorithms for flood susceptibility assessment and risk management. *Journal of Water and Climate Change*, *12*(6), 2608–2623. https://doi.org/10.2166/WCC.2021.051

Mahar, A., & Lagmay, F. A. (2014). *Lessons from Disasters in the Philippines: The Project NOAH Experience 1*.

Mccarthy, J. (2007). *What is Artificial Intelligence?* http://www-formal.stanford.edu/jmc/

Mehmood, H., Conway, C., & Perera, D. (2021). Mapping of Flood Areas Using Landsat with Google Earth Engine Cloud Platform. *Atmosphere 2021, Vol. 12, Page 866*, *12*(7), 866. https://doi.org/10.3390/ATMOS12070866

Mejico, L. (2020). *This web-based app helps citizens map flooded areas during disasters*. https://www.rappler.com/moveph/agos/mapakalamidad-ph-app-helps-citizens-map-flooded-areas-disasters/

Mohanty, A., Hussain, M., Mishra, M., Kattel, D. B., & Pal, I. (2019). Exploring community resilience and early warning solution for flash floods, debris flow and landslides in conflict prone villages of Badakhshan, Afghanistan. *International Journal of Disaster Risk Reduction, 33*, 5–15. https://doi.org/10.1016/j.ijdrr.2018.07.012

Mosavi, A., Ozturk, P., & Chau, K. W. (2018). Flood prediction using machine learning models: Literature review. In Water (Switzerland) (Vol. 10, Issue 11). MDPI AG. https://doi.org/10.3390/w10111536

Motta, M., de Castro Neto, M., & Sarmento, P. (2021). A mixed approach for urban flood prediction using Machine Learning and GIS. International *Journal of Disaster Risk Reduction, 56*, 102154. https://doi.org/10.1016/J.IJDRR.2021.102154

Mourato, S., Fernandez, P., Marques, F., Rocha, A., & Pereira, L. (2021). An interactive Web-GIS fluvial flood forecast and alert system in operation in Portugal. *International Journal of Disaster Risk Reduction, 58*, 102201. https://doi.org/10.1016/J.IJDRR.2021.102201

Mourato, S., Fernandez, P., Pereira, L. G., & Moreira, M. (2023). Assessing Vulnerability in Flood Prone Areas Using Analytic Hierarchy Process—Group Decision Making and Geographic Information System: A Case Study in Portugal. Applied Sciences 2023, Vol. 13, Page 4915, 13(8), 4915. https://doi.org/10.3390/APP13084915

Mudashiru, R. B., Sabtu, N., Abustan, I., & Balogun, W. (2021). Flood hazard mapping methods: A review. *Journal of Hydrology, 603*, 126846. https://doi.org/10.1016/J.JHYDROL.2021.126846

Munawar, H. S., Hammad, A. W. A., & Waller, S. T. (2022). Remote Sensing Methods for Flood Prediction: A Review. In Sensors (Vol. 22, Issue 3). MDPI. https://doi.org/10.3390/s22030960

Natarajan, L., Usha, T., Gowrappan, M., Palpanabhan Kasthuri, B., Moorthy, P., & Chokkalingam, L. (2021). Flood Susceptibility Analysis in Chennai Corporation Using Frequency Ratio Model. *Journal of the Indian Society of Remote Sensing, 49*(7), 1533–1543. https://doi.org/10.1007/s12524-021-01331-8

Nearing, G., Cohen, D., Dube, V., Gauch, M., Gilon, O., Harrigan, S., Hassidim, A., Klotz, D., Kratzert, F., Metzger, A., Nevo, S., Pappenberger, F., Prudhomme, C., Shalev, G., Shenzis, S., Tekalign, T. Y., Weitzner, D., & Matias, Y. (2024). Global prediction of extreme floods in ungauged watersheds. *Nature 2024 627:8004*, *627*(8004), 559–563. https://doi.org/10.1038/s41586-024-07145-1

Neil Ruaro II, E. G., Harry So, N. S., & Cu, G. G. (n.d.). Analyzing Weather Patterns for Predicting Floods with Regression Models.

Nhangumbe, M., Nascetti, A., Georganos, S., & Ban, Y. (2023). Supervised and unsupervised machine learning approaches using Sentinel data for flood mapping and damage assessment in Mozambique. Remote Sensing Applications: Society and Environment, 32. https://doi.org/10.1016/j.rsase.2023.101015

Nkwunonwo, U. C., Whitworth, M., & Baily, B. (2020). A review of the current status of flood modelling for urban flood risk management in the developing countries. Scientific African, 7, e00269. https://doi.org/10.1016/J.SCIAF.2020.E00269

PAGASA. (n.d.). *Learning Tools - Floods*. Retrieved September 26, 2024, from https://www.pagasa.dost.gov.ph/learning-tools/floods

Panahi, M., Jaafari, A., Shirzadi, A., Shahabi, H., Rahmati, O., Omidvar, E., Lee, S., & Bui, D. T. (2021). Deep learning neural networks for spatially explicit prediction of flash flood probability. *Geoscience Frontiers, 12*(3). https://doi.org/10.1016/j.gsf.2020.09.007

Pascual, L. A. C. A., Ong, A. K. S., Briggs, C. M., Diaz, J. F. T., & German, J. D. (2024). Factors affecting the intention to prepare for flash floods in the Philippines. *International Journal of Disaster Risk Reduction*, *112*. https://doi.org/10.1016/j.ijdrr.2024.104794

Sankaranarayanan, S., Prabhakar, M., Satish, S., Jain, P., Ramprasad, A., & Krishnan, A. (2020). Flood prediction based on weather parameters using deep learning. *Journal of Water and Climate Change, 11*(4), 1766–1783. https://doi.org/10.2166/wcc.2019.321

Sayama, T., Yamada, M., Sugawara, Y., & Yamazaki, D. (2020). Ensemble flash flood predictions using a high-resolution nationwide distributed rainfall-runoff model: case study of the heavy rain event of July 2018 and Typhoon Hagibis in 2019. *Progress in Earth and Planetary Science, 7*(1). https://doi.org/10.1186/s40645-020-00391-7

Seethepalli, S., Navya, S., Gadhiraju, H., Lanka, A., & Kumar, M. (2024). Flood Prediction: A Comparative Study of Machine Learning Algorithms. In Tuijin Jishu/Journal of Propulsion Technology (Vol. 45, Issue 3).

Sellami, E. M., Maanan, M., & Rhinane, H. (2022). Performance of Machine Learning Algorithms for Mapping and Forecasting of Flash Flood Susceptibility in Tetouan, Morocco. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLVI-4-W3-2021*(4/W3-2021), 305–313. https://doi.org/10.5194/ISPRS-ARCHIVES-XLVI-4-W3-2021-305-2022

Seydi, S. T., Kanani-Sadat, Y., Hasanlou, M., Sahraei, R., Chanussot, J., & Amani, M. (2023). Comparison of Machine Learning Algorithms for Flood Susceptibility Mapping. *Remote Sensing 2023, Vol. 15, Page 192*, *15*(1), 192. https://doi.org/10.3390/RS15010192

Shah, S. A., & Ai, S. (2024). Flood susceptibility mapping contributes to disaster risk reduction: A case study in Sindh, Pakistan. International Journal of Disaster Risk Reduction, 108. https://doi.org/10.1016/j.ijdrr.2024.104503

Sharif, S. (2024). 2D hydronumeric flood modeling using open-source software QGIS and freeware BASEMENT (V3) on a section of the Elbe River in Magdeburg, Germany. *Modern Cartography Series, 12*, 219–238. https://doi.org/10.1016/B978-0-443-23890-1.00009-8

Sharma, T., Prasad, S. K., & Mishra, P. (n.d.). FLOOD PREDICTION ANALYSIS USING SUPERVISED MACHINE LEARNING TECHNIQUES. http://ymerdigital.com

Silvestro, F., Rossi, L., Campo, L., Parodi, A., Fiori, E., Rudari, R., & Ferraris, L. (2019). Impact-based flash-flood forecasting system: Sensitivity to high resolution numerical weather prediction systems and soil moisture. *Journal of Hydrology, 572*, 388–402. https://doi.org/10.1016/j.jhydrol.2019.02.055

Smith, K. (n.d.). *What is flood mapping | Royal HaskoningDHV*. Retrieved September 26, 2024, from https://www.royalhaskoningdhv.com/en/twinn/blogs/2024/what-is-flood-mapping

Smith, R. E. (2022). Flood Warning: A Generalized Approach to Forecast the Impacts of Flooding Events Using ArcGIS Pro, QGIS, and Python of Flooding Events Using ArcGIS Pro, QGIS, and Python. https://scholarsarchive.byu.edu/etd

Srivanit, M., Pattanasri, S., Phichetkunbodee, N., Manokeaw, S., Sitthikankun, S., & Rinchumphu, D. (2024). Analysis of sensitive urban form indicators of flood susceptible prediction based on machine learning models. Global *Journal of Environmental Science and Management, 10*(4), 1501–1518. https://doi.org/10.22034/gjesm.2024.04.02

Swain, K. C., Singha, C., & Nayak, L. (2020). Flood Susceptibility Mapping through the GIS-AHP Technique Using the Cloud. *ISPRS International Journal of Geo-Information 2020, Vol. 9, Page 720*, *9*(12), 720. https://doi.org/10.3390/IJGI9120720

Tavares da Costa, R., Manfreda, S., Luzzi, V., Samela, C., Mazzoli, P., Castellarin, A., & Bagli, S. (2019). A web application for hydrogeomorphic flood hazard mapping. *Environmental Modelling & Software*, *118*, 172–186. https://doi.org/10.1016/J.ENVSOFT.2019.04.010

Thi Hang, H., Duy Hoa, P., Ngoc Tru, V., & Viet Phuong, N. (2021). Application of Shannon's entropy model and GIS in flash flood forecasting along National Highway-6, Hoa Binh Province, Vietnam. *International Journal of GEOMATE, 21*, 50–57. https://doi.org/10.21660/2021.87.j2316

Tien Bui, D., Hoang, N. D., Martínez-Álvarez, F., Ngo, P. T. T., Hoa, P. V., Pham, T. D., Samui, P., & Costache, R. (2020). A novel deep learning neural network approach for predicting flash flood susceptibility: A case study at a high frequency tropical storm area. Science of the Total Environment, 701. https://doi.org/10.1016/j.scitotenv.2019.134413

Trif, S., Bilașco, Ștefan, Petrea, D., Roșca, S., Fodorean, I., & Vescan, I. (2023). Spatial Modeling through GIS Analysis of Flood Risk and Related Financial Vulnerability: Case Study: Turcu River, Romania. *Applied Sciences (Switzerland)*, *13*(17). https://doi.org/10.3390/app13179869

United Nations-Spider. (n.d.). *Data Application of the Month: Overview of Web Maps for Flood Monitoring | UN-SPIDER Knowledge Portal*. Retrieved September 26, 2024, from https://www.un-spider.org/links-and-resources/data-sources/daotm-flood-web-maps

Wahba, M., Essam, R., El-Rawy, M., Al-Arifi, N., Abdalla, F., & Elsadek, W. M. (2024). Forecasting of flash flood susceptibility mapping using random forest regression model and geographic information systems. Heliyon, 10(13). https://doi.org/10.1016/j.heliyon.2024.e33982

Zehra, N. (n.d.). Prediction Analysis of Floods Using Machine Learning Algorithms (NARX & SVM). International Journal of Sciences: Basic and Applied Research. http://gssrr.org/index.php?journal=JournalOfBasicAndApplied

APPENDICES

Appendix A

Letters and Appointments

Letter to CBSUA, DOST-PAGASA as Partner Stakeholders

Republic of the Philippines
BICOL UNIVERSITY
College of Science
Legazpi City, Albay

BAGONG PILIPINAS

October 16, 2024

CENTRAL BICOL STATE UNIVERSITY OF AGRICULTURE
RECORDS SECTION
RECEIVED
BY:
DATE: NOV 25 2024 TIME: 8:25
CODE: 11 - 1724

ALBERTO N. NAPERI, DPA
SUC President IV
Central Bicol State University of Agriculture

ATTN: PROF. VLADIMIR R. FORONDA, Ph.D.
Director, Extension Services Division & Digital Agriculture and Innovation Center

Dear Sir Naperi,

Subject: Request for Datasets and Consultation for Research Purposes

I hope this letter finds you well. We, Alexandra Nicole Eclarinal, Yna Gabrielle Foronda, and Francis Maurice Miranda, are fourth year students from Bicol University pursuing a degree in BS Computer Science. We are currently conducting our thesis entitled *"FLOWS: An Early Predictive Flood Mapping Web Application for Camarines Sur Using Machine Learning Algorithms and Spatial Analysis Software,"* under the supervision of Prof. Aris J. Ordoñez and Prof. Arlene A. Satuito.

Our research aims to develop a web-based 3D flood mapping application capable of predicting flood levels for Camarines Sur with a lead time of six to twelve hours, using both elevation and rainfall data. By utilizing machine learning algorithms and spatial analysis, our goal is to create a reliable flood prediction tool that can aid in emergency response, disaster risk management, and the timely dissemination of critical flood level information to the public and relevant authorities.

To successfully implement this project, we require comprehensive **historical and current rainfall and water level data**, including other related datasets.

This data will be instrumental in training the machine learning models and validating the accuracy of our flood predictions. We assure you that all data provided will be used solely for academic purposes and in compliance with any confidentiality or data-sharing protocols set by your agency.

We kindly request your assistance in facilitating access to this data, and we would greatly appreciate any guidance on the procedures for obtaining it. Should you need any further information or clarification regarding our research or the requested data, please feel free to contact us using the details below.

Thank you in advance for your support and cooperation in this important endeavor. We look forward to your positive response and hope to collaborate with your agency in this vital project.

Republic of the Philippines
BICOL UNIVERSITY
College of Science
Legazpi City, Albay

BAGONG PILIPINAS

Sincerely,

ALEXANDRA NICOLE D. ECLARINAL
alexandranicoledimabayao.eclarinal@bicol-u.edu.ph
+639 484 046 495

YNA GABRIELLE P. FORONDA
ynagabrielleplopinio.foronda@bicol-u.edu.ph
+639 624 144 064

FRANCIS MAURICE B. MIRANDA
francismauricebolaza.miranda@bicol-u.edu.ph
+639 199 647 029

Noted:

ARIS J. ORDOÑEZ, DIT
Programming Adviser

PROF. ARLENE A. SATUITO
Content Adviser

Republic of the Philippines
CENTRAL BICOL STATE UNIVERSITY
San Jose, Pili, Camarines Sur 4418
Website: www.cbsua.edu.ph
Email Address: op@cbsua.edu.ph
Trunkline: (054) 871-5531;871-5533

BAGONG PILIPINAS

OFFICE OF THE PRESIDENT

TO: Dr. R. L. Ramirez
VP for Res. & Innovation
Attn: Dr. V Foronda
Director, ESD & OAIC

DATE: 4-25-2024

SUBJECT:
please for your information and possible consideration of the request.

- [ ] For Appropriate Action
- [ ] For Information and Reference
- [ ] For Comments/Suggestion/Recommendation
- [ ] Please provide me feedback on/before ____________
- [ ] Please see me on this
- [ ] Please represent me on this

FROM: **ALBERTO N. NAPERI, DPA**
SUC President IV

OUP-FR-001
Effectivity Date: June 3, 2024

Rev: 2
Page 1 of 1

Letter to CADRRESMO

January 15, 2025

Camaligan Disaster Risk Reduction and Management Office (DRRMO)
Municipality of Camaligan
Camarines Sur

Subject: Request for a Courtesy Visit to Camaligan DRRMO and Flood-Prone Communities

Dear Sir/Madam,

We hope this letter finds you well. We are a group of researchers and developers from Bicol University, currently undertaking a project titled **"FLOWS: A Predictive Fluvial Flood Mapping Web Application for the Municipality of Camaligan, Camarines Sur Using Machine Learning".** This initiative aims to create a web-based tool that integrates machine learning and spatial analysis to provide real-time flood predictions and 3D flood maps, specifically tailored to Camaligan.

To ensure the success and community relevance of this project, we would like to request a courtesy visit to the Camaligan DRRMO office and nearby communities frequently affected by fluvial flooding. Our primary objectives for this visit are as follows:

1. **Understand the current flood response strategies** employed by your office and identify how our system can support and enhance these efforts.
2. **Gather insights and feedback** from the DRRMO and community members about their needs and challenges in dealing with fluvial flooding.
3. **Present and discuss** the proposed features of our predictive flood mapping system to ensure alignment with the municipality's disaster preparedness goals.

We believe this collaboration will significantly benefit both your office and the local community by enabling better preparation and decision-making during flood events. We kindly propose scheduling the visit on the **22nd of January**.

Should you need further details or wish to discuss this request, please feel free to contact us using the details below. We are looking forward to working closely with the Camaligan DRRMO to make this project impactful and meaningful to the community.

Thank you very much for your time and consideration. We eagerly await your positive response.

Sincerely,

ALEXANDRA NICOLE D. ECLARINAL
Bicol University
alexandranicoledimabayao.eclarinal@bicol-u.edu.ph
09484046495

Proposal Defense Appointments

BICOL UNIVERSITY
COLLEGE OF SCIENCE
Computer Science and Information Technology Department
Legazpi City

APPOINTMENT OF SPECIAL PROBLEM 1 EVALUATORS

December 19, 2024

Chairman: **RYAN A. RODRIGUEZ, MSCS, MIT**
Member: **MARY JANE B. BURCE, MIT**
Member: **MARY JOY P. CANON, DIT**

You are hereby appointed to constitute the Special Problem Panel as indicated above to evaluate the research work of **Eclarinal, Alexandra Nicole D., Foronda, Yna Gabrielle P., and Miranda, Francis Maurice B.** who will work on the topic, **"FLOWS: A Predictive Fluvial Flood Mapping Web Application for the Municipality of Camaligan Using Machine Learning and Spatial Analysis Software"**, which is scheduled for its Proposal Defense on December 19, 2024 at 2:30-4:00 p.m. in CSB2 Room 201.

As member of the panel you are asked to:

1) Appraise the validity and acceptability of the thesis work in terms of its scholarly quality, correctness of the facts and claims contained therein; and completeness as to its basic components.
2) Make sure that all the suggestions are judiciously incorporated.
3) Evaluate the research report based on adopted.
4) Provide ample time to his advisee in relation to the thesis work.
5) Orient the advisee on what might/will transpire in the defense session and
6) Be physically present during the oral defense.

You shall be entitled to an honorarium as chairman and as member of the panel, as per Board Resolution No.93, s 2006.

Very truly yours,

JOCELYN E. SERRANO, M. Sc.
Dean, College of Science

Conforme:

RYAN A. RODRIGUEZ, MSCS, MIT
Chairman

MARY JANE B. BURCE, MIT
Member

MARY JOY P. CANON, DIT
Member

BICOL UNIVERSITY
COLLEGE OF SCIENCE
Computer Science and Information Technology Department
Legazpi City

APPOINTMENT OF SPECIAL PROBLEM 1 EVALUATORS

December 19, 2024

RYAN A. RODRIGUEZ, MSCS, MIT
College of Science
Legazpi City

You are hereby appointed to constitute the Special Problem Panel as indicated above to evaluate the research work of **Eclarinal, Alexandra Nicole D., Foronda, Yna Gabrielle P., and Miranda, Francis Maurice B.** who will work on the topic, **"FLOWS: A Predictive Fluvial Flood Mapping Web Application for the Municipality of Camaligan Using Machine Learning and Spatial Analysis Software",** which is scheduled for its Proposal Defense on December 19, 2024 at 2:30-4:00 p.m. in CSB2 Room 201.

As member of the panel, you are asked to:

1) Appraise the validity and acceptability of the thesis work in terms of its scholarly quality, correctness of the facts and claims contained therein; and completeness as to its basic components.
2) Make sure that all the suggestions are judiciously incorporated.
3) Evaluate the research report based on adopted.
4) Provide ample time to her advisee in relation to the thesis work.
5) Orient the advisee on what might/will transpire in the defense session and
6) Be physically present during the oral defense.

You shall be entitled to an honorarium as chairman of the panel, as per Board Resolution No.93, s 2006.

Very truly yours,

JOCELYN E. SERRANO, M. Sc.
Dean, College of Science

Conforme:

RYAN A. RODRIGUEZ, MSCS, MIT
Chairman of the Panel

BICOL UNIVERSITY
COLLEGE OF SCIENCE
Computer Science and Information Technology Department
Legazpi City

APPOINTMENT OF SPECIAL PROBLEM 1 EVALUATORS

December 19, 2024

MARY JANE B. BURCE, MIT
College of Science
Legazpi City

You are hereby appointed to constitute the Special Problem Panel as indicated above to evaluate the research work of **Eclarinal, Alexandra Nicole D., Foronda, Yna Gabrielle P., and Miranda, Francis Maurice B.** who will work on the topic, **"FLOWS: A Predictive Fluvial Flood Mapping Web Application for the Municipality of Camaligan Using Machine Learning and Spatial Analysis Software"**, which is scheduled for its Proposal Defense on December 19, 2024 at 2:30-4:00 p.m. in CSB2 Room 201.

As member of the panel, you are asked to:

1) Appraise the validity and acceptability of the thesis work in terms of its scholarly quality, correctness of the facts and claims contained therein; and completeness as to its basic components.
2) Make sure that all the suggestions are judiciously incorporated.
3) Evaluate the research report based on adopted.
4) Provide ample time to her advisee in relation to the thesis work.
5) Orient the advisee on what might/will transpire in the defense session and
6) Be physically present during the oral defense.

You shall be entitled to an honorarium as a member of the panel, as per Board Resolution No.93, s 2006.

Very truly yours,

JOCELYN E. SERRANO, M. Sc.
Dean, College of Science

Conforme:

MARY JANE B. BURCE, MIT
Member of the Panel

BICOL UNIVERSITY
COLLEGE OF SCIENCE
Computer Science and Information Technology Department
Legazpi City

APPOINTMENT OF SPECIAL PROBLEM 1 EVALUATORS

December 19, 2024

MARY JOY P. CANON, DIT
College of Science
Legazpi City

You are hereby appointed to constitute the Special Problem Panel as indicated above to evaluate the research work of **Eclarinal, Alexandra Nicole D., Foronda, Yna Gabrielle P., and Miranda, Francis Maurice B.** who will work on the topic, **"FLOWS: A Predictive Fluvial Flood Mapping Web Application for the Municipality of Camaligan Using Machine Learning and Spatial Analysis Software"**, which is scheduled for its Proposal Defense on December 19, 2024 at 2:30-4:00 p.m. in CSB2 Room 201.

As member of the panel, you are asked to:

1) Appraise the validity and acceptability of the thesis work in terms of its scholarly quality, correctness of the facts and claims contained therein; and completeness as to its basic components.
2) Make sure that all the suggestions are judiciously incorporated.
3) Evaluate the research report based on adopted.
4) Provide ample time to her advisee in relation to the thesis work.
5) Orient the advisee on what might/will transpire in the defense session and
6) Be physically present during the oral defense.

You shall be entitled to an honorarium as member of the panel, as per Board Resolution No.93, s 2006.

Very truly yours,

JOCELYN E. SERRANO, M. Sc.
Dean, College of Science

Conforme:

MARY JOY P. CANON, DIT
Member of the Panel

BICOL UNIVERSITY
COLLEGE OF SCIENCE
Computer Science and Information Technology Department
Legazpi City

APPOINTMENT OF SPECIAL PROBLEM 1 PROGRAMMING ADVISER

December 19, 2024

ARIS J. ORDOÑEZ, DIT
College of Science
Legazpi City

You are hereby appointed to constitute the Special Problem Panel as indicated above to evaluate the research work of **Eclarinal, Alexandra Nicole D., Foronda, Yna Gabrielle P., and Miranda, Francis Maurice B.** who will work on the topic, **"FLOWS: A Predictive Fluvial Flood Mapping Web Application for the Municipality of Camaligan Using Machine Learning and Spatial Analysis Software"**, which is scheduled for its Proposal Defense on December 19, 2024 at 2:30-4:00 p.m. in CSB2 Room 201.

As an adviser, you shall perform the following task:

1) Check the format of the manuscript.
2) Provide general editing of thesis work.
3) Attend the defense session of the advisees and record suggestions and recommendations at the panel.
4) Orient the advisee on what might/will transpire in the defense session
5) Be physically present during the oral defense.

This designation shall be entitled to a professional fee as authorized under Board Resolution No.093, s 2006.

Very truly yours,

JOCELYN E. SERRANO, M. Sc.
Dean, College of Science

Conforme:

ARIS J. ORDOÑEZ, DIT
Programming Adviser

BICOL UNIVERSITY
COLLEGE OF SCIENCE
Computer Science and Information Technology Department
Legazpi City

APPOINTMENT OF SPECIAL PROBLEM 1 CONTENT ADVISER

December 19, 2024

ARLENE A. SATUITO
Professor
College of Science
Legazpi City

You are hereby appointed to constitute the Special Problem Panel as indicated above to evaluate the research work of **Eclarinal, Alexandra Nicole D., Foronda, Yna Gabrielle P., and Miranda, Francis Maurice B.** who will work on the topic, **"FLOWS: A Predictive Fluvial Flood Mapping Web Application for the Municipality of Camaligan Using Machine Learning and Spatial Analysis Software",** which is scheduled for its Proposal Defense on December 19, 2024 at 2:30-4:00 p.m. in CSB2 Room 201.

As an adviser, you shall perform the following task:

1) Check the format of the manuscript.
2) Provide general editing of thesis work.
3) Attend the defense session of the advisees and record suggestions and recommendations at the panel.
4) Orient the advisee on what might/will transpire in the defense session
5) Be physically present during the oral defense.

This designation shall be entitled to a professional fee as authorized under Board Resolution No.065, s 2004.

Very truly yours,

JOCELYN E. SERRANO, M. Sc.
Dean, College of Science

Conforme:

ARLENE A. SATUITO
Content Adviser

Final Defense Appointments

BICOL UNIVERSITY
COLLEGE OF SCIENCE
Computer Science Department
Legazpi City

APPOINTMENT OF SPECIAL PROBLEM 2 EVALUATORS

May 4, 2025

Chairman: **RYAN A. RODRIGUEZ, MSCS, MIT**
Member: **MARY JANE B. BURCE, MIT**
Member: **MARY JOY P. CANON, DIT**

You are hereby appointed to constitute the Special Problem Panel as indicated above to evaluate the research work of **Eclarinal, Alexandra Nicole D., Foronda, Yna Gabrielle P., and Miranda, Francis Maurice B.** who will work on the topic, **"FLOWS: A Predictive Fluvial Flood Mapping Web Application for the Municipality of Camaligan, Camarines Sur Using Machine Learning"**, which is scheduled for its Final Defense on May 4, 2025 at 1:00-2:30 p.m. in CSB2 Room 200.

As a member of the panel, you are asked to:

1) Appraise the validity and acceptability of the thesis work in terms of its scholarly quality, correctness of the facts and claims contained therein, and completeness as to its basic components.
2) Make sure that all the suggestions are judiciously incorporated.
3) Evaluate the research report based on the adopted guidelines.
4) Provide ample time to his advisee in relation to the thesis work.
5) Orient the advisee on what might/will transpire in the defense session and
6) Be physically present during the oral defense.

You shall be entitled to an honorarium as the chairman and as members of the panel, as per Board Resolution No.93, s 2006.

Very truly yours,

JOCELYN E. SERRANO, M. Sc.
Dean, College of Science

Conforme:

RYAN A. RODRIGUEZ, MSCS, MIT
Chairman

MARY JANE B. BURCE, MIT
Member

MARY JOY P. CANON, DIT
Member

BICOL UNIVERSITY
COLLEGE OF SCIENCE
Computer Science Department
Legazpi City

APPOINTMENT OF SPECIAL PROBLEM 2 EVALUATORS

May 4, 2025

RYAN A. RODRIGUEZ, MSCS, MIT
College of Science
Legazpi City

You are hereby appointed to constitute the Special Problem Panel as indicated above to evaluate the research work of **Eclarinal, Alexandra Nicole D., Foronda, Yna Gabrielle P., and Miranda, Francis Maurice B.** who will work on the topic, **"FLOWS: A Predictive Fluvial Flood Mapping Web Application for the Municipality of Camaligan, Camarines Sur Using Machine Learning"**, which is scheduled for its Final Defense on May 4, 2025 at 1:00-2:30 p.m. in CSB2 Room 200.

As a member of the panel, you are asked to:

1) Appraise the validity and acceptability of the thesis work in terms of its scholarly quality, correctness of the facts and claims contained therein, and completeness as to its basic components.
2) Make sure that all the suggestions are judiciously incorporated.
3) Evaluate the research report based on the adopted guidelines.
4) Provide ample time to his advisee in relation to the thesis work.
5) Orient the advisee on what might/will transpire in the defense session and
6) Be physically present during the oral defense.

You shall be entitled to an honorarium as the chairman of the panel, as per Board Resolution No.93, s 2006.

Very truly yours,

JOCELYN E. SERRANO, M. Sc.
Dean, College of Science

Conforme:

RYAN A. RODRIGUEZ, MSCS, MIT
Chairman of the Panel

BICOL UNIVERSITY
COLLEGE OF SCIENCE
Computer Science Department
Legazpi City

APPOINTMENT OF SPECIAL PROBLEM 2 EVALUATORS

May 4, 2025

MARY JANE B. BURCE, MIT
College of Science
Legazpi City

You are hereby appointed to constitute the Special Problem Panel as indicated above to evaluate the research work of **Eclarinal, Alexandra Nicole D., Foronda, Yna Gabrielle P., and Miranda, Francis Maurice B.** who will work on the topic, "**FLOWS: A Predictive Fluvial Flood Mapping Web Application for the Municipality of Camaligan, Camarines Sur Using Machine Learning**", which is scheduled for its Final Defense on May 4, 2025 at 1:00-2:30 p.m. in CSB2 Room 200.

As a member of the panel, you are asked to:

1) Appraise the validity and acceptability of the thesis work in terms of its scholarly quality, correctness of the facts and claims contained therein, and completeness as to its basic components.
2) Make sure that all the suggestions are judiciously incorporated.
3) Evaluate the research report based on the adopted guidelines.
4) Provide ample time to his advisee in relation to the thesis work.
5) Orient the advisee on what might/will transpire in the defense session and
6) Be physically present during the oral defense.

You shall be entitled to an honorarium as a member of the panel, as per Board Resolution No.93, s 2006.

Very truly yours,

JOCELYN E. SERRANO, M. Sc.
Dean, College of Science

Conforme:

MARY JANE B. BURCE, MIT
Member of the Panel

BICOL UNIVERSITY
COLLEGE OF SCIENCE
Computer Science Department
Legazpi City

APPOINTMENT OF SPECIAL PROBLEM 2 EVALUATORS

May 4, 2025

MARY JOY P. CANON, DIT
College of Science
Legazpi City

You are hereby appointed to constitute the Special Problem Panel as indicated above to evaluate the research work of **Eclarinal, Alexandra Nicole D., Foronda, Yna Gabrielle P., and Miranda, Francis Maurice B.** who will work on the topic, "**FLOWS: A Predictive Fluvial Flood Mapping Web Application for the Municipality of Camaligan, Camarines Sur Using Machine Learning",** which is scheduled for its Final Defense on May 4, 2025 at 1:00-2:30 p.m. in CSB2 Room 200.

As a member of the panel, you are asked to:

1) Appraise the validity and acceptability of the thesis work in terms of its scholarly quality, correctness of the facts and claims contained therein, and completeness as to its basic components.
2) Make sure that all the suggestions are judiciously incorporated.
3) Evaluate the research report based on the adopted guidelines.
4) Provide ample time to his advisee in relation to the thesis work.
5) Orient the advisee on what might/will transpire in the defense session and
6) Be physically present during the oral defense.

You shall be entitled to an honorarium as a member of the panel, as per Board Resolution No.93, s 2006.

Very truly yours,

JOCELYN E. SERRANO, M. Sc.
Dean, College of Science

Conforme:

MARY JOY P. CANON, DIT
Member of the Panel

BICOL UNIVERSITY
COLLEGE OF SCIENCE
Computer Science Department
Legazpi City

APPOINTMENT OF SPECIAL PROBLEM 2 PROGRAMMING ADVISER

May 4, 2025

ARIS J. ORDOÑEZ, DIT
College of Science
Legazpi City

You are hereby appointed to constitute the Special Problem Panel as indicated above to evaluate the research work of **Eclarinal, Alexandra Nicole D., Foronda, Yna Gabrielle P., and Miranda, Francis Maurice B.** who will work on the topic, **"FLOWS: A Predictive Fluvial Flood Mapping Web Application for the Municipality of Camaligan, Camarines Sur Using Machine Learning",** which is scheduled for its Final Defense on May 4, 2025 at 1:00-2:30 p.m. in CSB2 Room 200.

As an adviser, you shall perform the following tasks:

1) Check the format of the manuscript.
2) Provide general editing of thesis work.
3) Attend the defense session of the advisees and record suggestions and recommendations at the panel.
4) Orient the advisee on what might/will transpire in the defense session
5) Be physically present during the oral defense.

This designation shall be entitled to a professional fee as authorized under Board Resolution No.093, s 2006.

Very truly yours,

JOCELYN E. SERRANO, M. Sc.
Dean, College of Science

Conforme:

ARIS J. ORDOÑEZ, DIT
Programming Adviser

BICOL UNIVERSITY
COLLEGE OF SCIENCE
Computer Science Department
Legazpi City

APPOINTMENT OF SPECIAL PROBLEM 2 CONTENT ADVISER

May 4, 2025

ARLENE A. SATUITO
College of Science
Legazpi City

You are hereby appointed to constitute the Special Problem Panel as indicated above to evaluate the research work of **Eclarinal, Alexandra Nicole D., Foronda, Yna Gabrielle P., and Miranda, Francis Maurice B.** who will work on the topic, **"FLOWS: A Predictive Fluvial Flood Mapping Web Application for the Municipality of Camaligan, Camarines Sur Using Machine Learning",** which is scheduled for its Final Defense on May 4, 2025 at 1:00-2:30 p.m. in CSB2 Room 200.

As an adviser, you shall perform the following tasks:

1) Check the format of the manuscript.
2) Provide general editing of thesis work.
3) Attend the defense session of the advisees and record suggestions and recommendations at the panel.
4) Orient the advisee on what might/will transpire in the defense session
5) Be physically present during the oral defense.

This designation shall be entitled to a professional fee as authorized under Board Resolution No.065, s 2004.

Very truly yours,

JOCELYN E. SERRANO, M. Sc.
Dean, College of Science

Conforme:

ARLENE A. SATUITO
Content Adviser

Appendix B

Diagrams

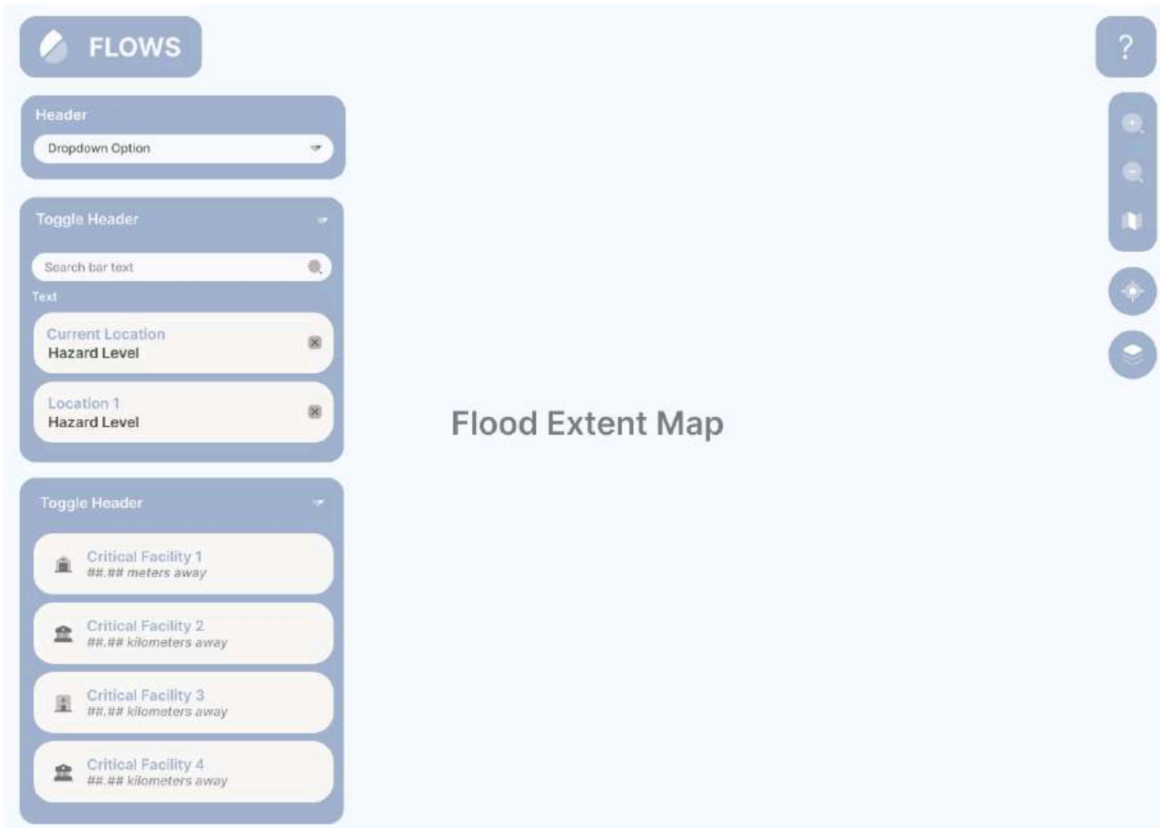


Figure 5. **Wireframe**

Figure 6. **Level 0 Data Flow Diagram**

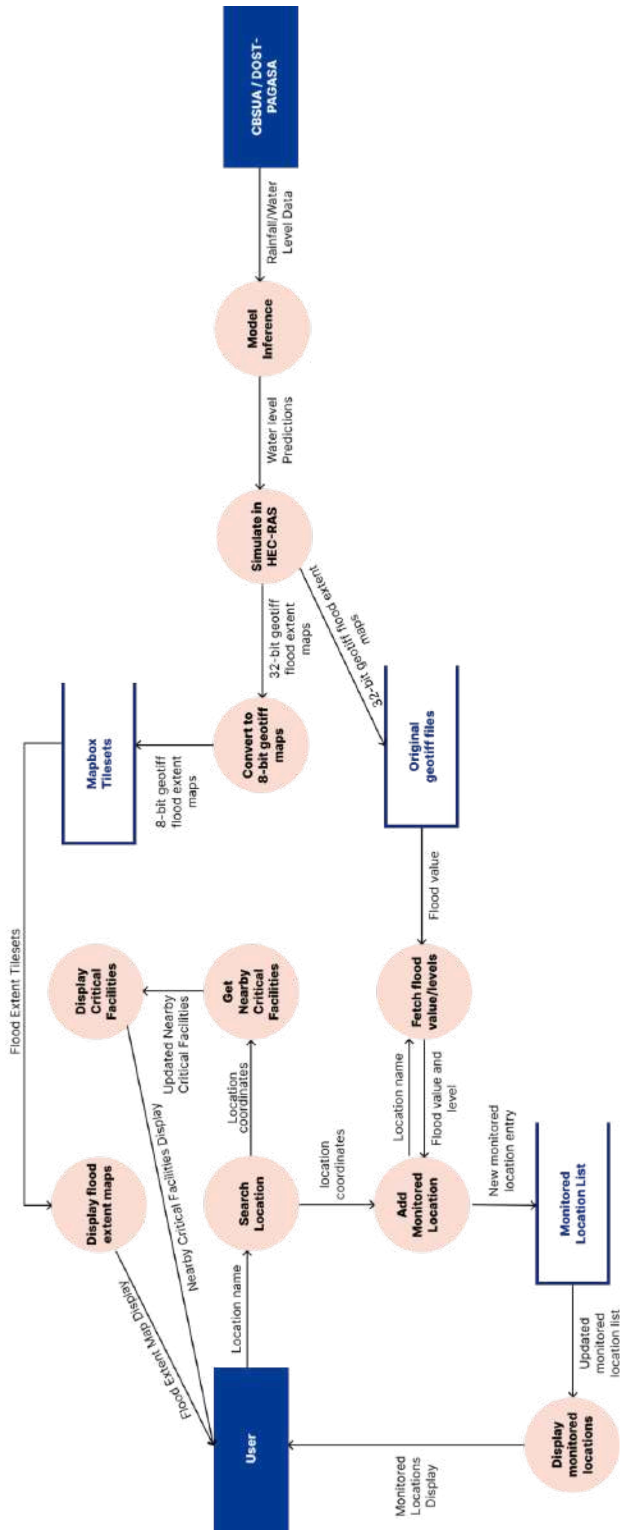


Figure 7. **Level 1 Data Flow Diagram**

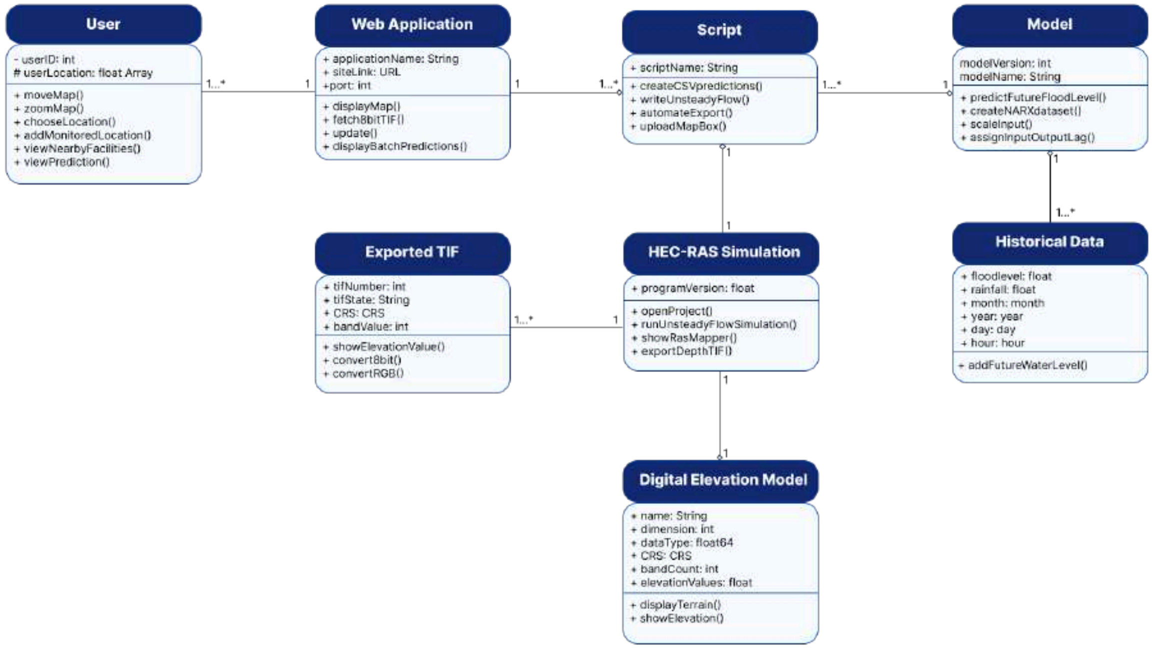


Figure 8. **Class Diagram**

The diagram in Figure 8 showcases the various attributes and relationships of each class that the project will utilize. The above diagram helps visualize how the project would function on the back-end side by showing how each one interacts and logically binds together. The various classes include the User, Web Application, the Scripts used in the automation process, the flood prediction model and its corresponding historical data, and the utilized flood mapping software with its generated geotiff files. All these classes work together to support the functionality of the project.

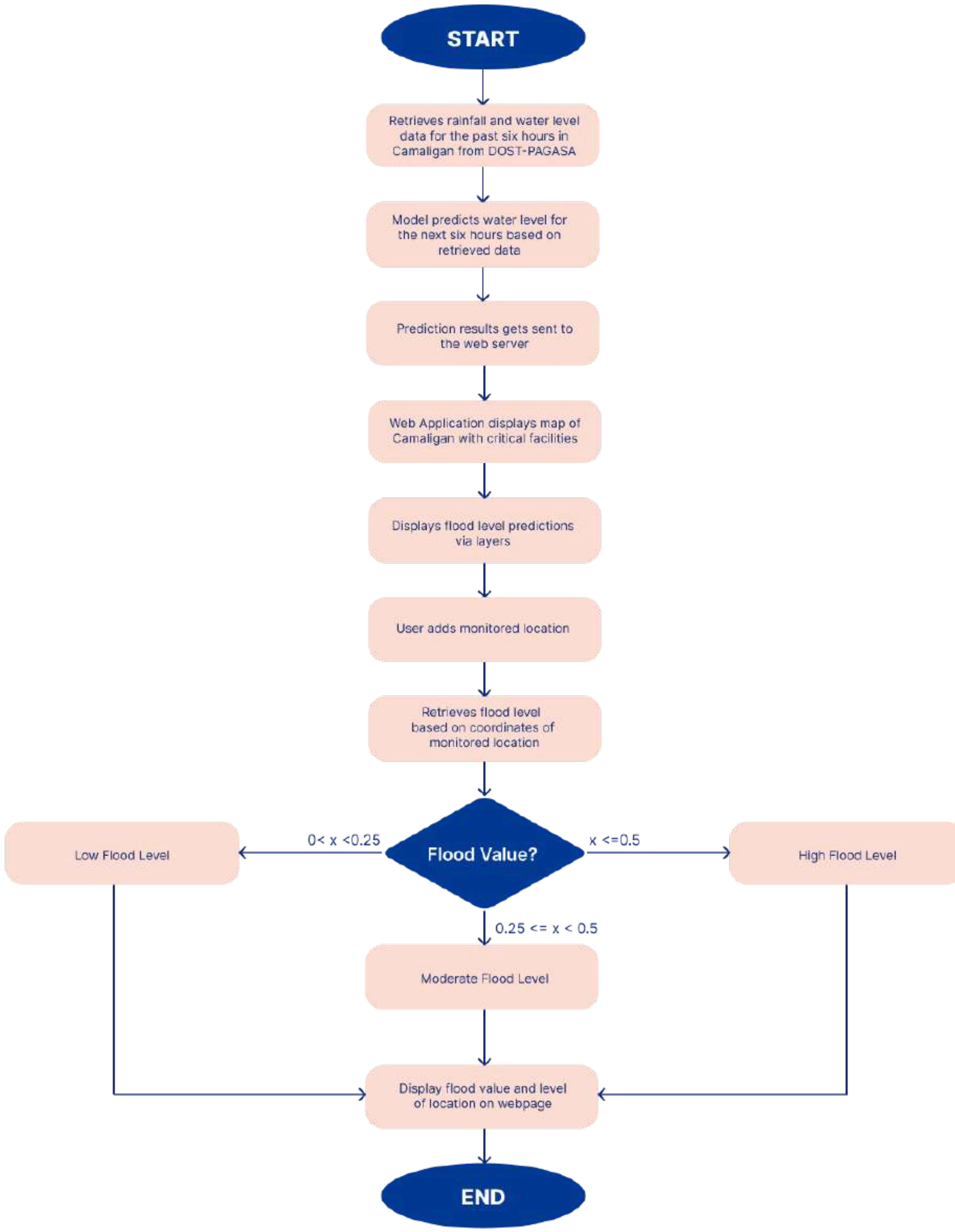


Figure 9. **Flowchart**

Figure 9 illustrates the sequence of operations for the predictive flood mapping process, starting from data retrieval to displaying flood predictions and sending alerts. The flow ensures accurate forecasting and effective communication of results to users and affected communities.

The process begins with the initiation of the system, followed by the retrieval of rainfall and flood data for the past six hours in Camaligan using a repository. This historical data serves as the basis for predictions. The retrieved data is sent to the predictive model, which uses machine learning algorithms to forecast flood levels for the next six hours.

The prediction results from the model up to the simulation are uploaded to Mapbox, which processes the data for visualization. The web application then displays a flood map of Camaligan, including critical facilities and flood level predictions in which are visualized using color-coded layers. These color-coded layers are assigned based on flood level, with low indicated in yellow, moderate indicated in orange, and high indicated in red. The activity concludes after visualizing the flood level and monitored locations with corresponding nearby critical facilities on the web page.

Appendix C

Model Training and Fine-Tuning

```
In [110…  model.eval()
          with torch.no_grad():
              y_pred_tensor = model(X_test_tensor)
              y_pred = y_pred_tensor.numpy()

          y_pred_original = scaler_y.inverse_transform(y_pred)
          y_test_original = scaler_y.inverse_transform(y_test_tensor.numpy().reshape(-1, 1))

In [111…  from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
          import numpy as np

          mae = mean_absolute_error(y_test_original, y_pred_original)
          mse = mean_squared_error(y_test_original, y_pred_original)
          rmse = np.sqrt(mse)
          r2 = r2_score(y_test_original, y_pred_original)

          print("Model Performance Metrics:")
          print(f"Mean Absolute Error (MAE): {mae:.3f}")
          print(f"Mean Squared Error (MSE): {mse:.3f}")
          print(f"Root Mean Squared Error (RMSE): {rmse:.3f}")
          print(f"R² Score: {r2:.3f}")

          Model Performance Metrics:
          Mean Absolute Error (MAE): 0.367
          Mean Squared Error (MSE): 0.193
          Root Mean Squared Error (RMSE): 0.439
          R² Score: -0.387
```

Figure 10. **Initial Model**

```
In [12]:  def nash_sutcliffe(y_true, y_pred):
              numerator = np.sum((y_true - y_pred) ** 2)
              denominator = np.sum((y_true - np.mean(y_true)) ** 2)
              return 1 - (numerator / denominator)

In [13]:  from sklearn.metrics import mean_squared_error

          mse = mean_squared_error(y_test_orig, y_pred_orig)
          rmse = np.sqrt(mse)

          mae = mean_absolute_error(y_test_orig, y_pred_orig)
          r2 = r2_score(y_test_orig, y_pred_orig)
          nse = nash_sutcliffe(y_test_orig, y_pred_orig)

          print(f"\nModel Performance Metrics:")
          print(f"Mean Absolute Error (MAE): {mae:.5f}")
          print(f"R² Score: {r2:.2f}")
          print(f"Nash-Sutcliffe Efficiency (NSE): {nse:.2f}")
          print(f"Mean Squared Error (MSE): {mse:.5f}")
          print(f"Root Mean Squared Error (RMSE): {rmse:.5f}")

          Model Performance Metrics:
          Mean Absolute Error (MAE): 0.10869
          R² Score: 0.87
          Nash-Sutcliffe Efficiency (NSE): 0.87
          Mean Squared Error (MSE): 0.03171
          Root Mean Squared Error (RMSE): 0.17807
```

Figure 11. **Final Model**

```
In [15]:  # Print predictions vs actual values
          print("\nPredictions vs Actual Values:")
          for i in range(min(20, len(y_pred_orig))):  # Print first 20
              print(f"Prediction: {y_pred_orig[i][0]:.3f}, Actual: {y_test_orig[i][0]:.3f}")

          Predictions vs Actual Values:
          Prediction: -1.303, Actual: -1.400
          Prediction: -1.368, Actual: -1.430
          Prediction: -1.408, Actual: -1.450
          Prediction: -1.421, Actual: -1.460
          Prediction: -1.389, Actual: -1.470
          Prediction: -1.312, Actual: -1.470
          Prediction: -1.164, Actual: -1.470
          Prediction: -1.022, Actual: -1.200
          Prediction: -0.962, Actual: -0.930
          Prediction: -0.942, Actual: -0.840
          Prediction: -0.972, Actual: -0.860
          Prediction: -1.084, Actual: -0.970
          Prediction: -1.205, Actual: -1.190
          Prediction: -1.292, Actual: -1.390
          Prediction: -1.371, Actual: -1.420
          Prediction: -1.403, Actual: -1.440
          Prediction: -1.346, Actual: -1.460
          Prediction: -1.247, Actual: -1.450
          Prediction: -1.106, Actual: -1.160
          Prediction: -0.935, Actual: -0.820
```

Figure 12. **Prediction vs Actual Results of the ML Model**

Appendix D

System Evaluation Tool and Results

FLOWS System Evaluation | User Satisfaction Survey

Instructions: Please rate the system based on your experience. Select the most appropriate response for each statement.

Section 1: User Profile

1. **Role in the Community/Organization:**
   ☐ Local Government Official
   ☐ Disaster Risk Reduction Officer
   ☐ Emergency Responder
   ☐ Community Member
   ☐ Researcher/Academic
   ☐ Other (please specify): ___________

On a scale of 1 (Strongly Disagree) to 5 (Strongly Agree), please rate the following statements:

Section 2: Usability and Learnability

Statement	1	2	3	4	5
The system is easy to navigate and use.	☐	☐	☐	☐	☐
The interface is user-friendly and visually clear.	☐	☐	☐	☐	☐
I was able to learn how to use the system quickly.	☐	☐	☐	☐	☐
The system provides clear instructions and labels.	☐	☐	☐	☐	☐
The process of accessing flood predictions is simple and efficient.	☐	☐	☐	☐	☐

Section 3: System Efficiency and Effectiveness

Statement	1	2	3	4	5
The system provides accurate flood predictions.	☐	☐	☐	☐	☐
The system updates predictions in a timely manner.	☐	☐	☐	☐	☐
The system effectively integrates visualization for flood mapping.	☐	☐	☐	☐	☐
The flood hazard levels (low, medium, high) are clearly distinguishable.	☐	☐	☐	☐	☐
The system loads quickly and operates smoothly.	☐	☐	☐	☐	☐

Section 4: Reliability and Security

Statement	1	2	3	4	5
The system provides reliable data without frequent errors.	☐	☐	☐	☐	☐
The system functions consistently without unexpected crashes.	☐	☐	☐	☐	☐
My data and information feel secure while using the system.	☐	☐	☐	☐	☐

Section 5: Overall Satisfaction and Recommendations

1. How satisfied are you with **FLOWS: A Predictive Fluvial Flood Mapping Web Application for the Municipality of Camaligan, Camarines Sur Using Machine Learning?**
   ☐ Very Dissatisfied
   ☐ Dissatisfied
   ☐ Neutral
   ☐ Satisfied
   ☐ Very Satisfied

2. Would you recommend the system to others?
   ☐ Yes

☐ No
☐ Not sure

3. What do you like most about the system?

4. What improvements would you suggest?

Section 6: To assess the real-world performance and reliability of FLOWS, this section presents an evaluation of the system using **Typhoon Kristine (2024)** as a case study. By comparing predicted flood levels, affected areas, and the proximity of critical facilities, this case-based evaluation aims to demonstrate the system's practical relevance, accuracy, and potential contributions to disaster preparedness and response.

On a scale of 1 (Strongly Disagree) to 5 (Strongly Agree), please rate the following statements:

Statement	1	2	3	4	5
How closely did the predicted flood levels from FLOWS during Typhoon Kristine align with the actual reported water levels and flood-affected areas in Camaligan?	☐	☐	☐	☐	☐
Were the predicted high flood-risk areas and critical facilities identified by the system during Typhoon Kristine consistent with the locations that were actually affected or reported?	☐	☐	☐	☐	☐
If FLOWS had been actively deployed before Typhoon Kristine, do you think it might have improved early warning, evacuation planning, or resource allocation in the Municipality of Camaligan?	☐	☐	☐	☐	☐

Other related comment/s:

____

FLOWS System Evaluation | User Satisfaction Survey

Instructions: Please rate the system based on your experience. Select the most appropriate response for each statement.

Section 1: User Profile

1. **Role in the Community/Organization:**
   - ☐ Local Government Official
   - ☑ Disaster Risk Reduction Officer
   - ☐ Emergency Responder
   - ☐ Community Member
   - ☐ Researcher/Academic
   - ☐ Other (please specify): __________

On a scale of 1 (Strongly Disagree) to 5 (Strongly Agree), please rate the following statements:

Section 2: Usability and Learnability ***(ISO/IEC 25010 - Usability, Learnability)***

Statement	1	2	3	4	5
The system is easy to navigate and use.	☐	☐	☐	☐	☑
The interface is user-friendly and visually clear.	☐	☐	☐	☐	☑
I was able to learn how to use the system quickly.	☐	☐	☐	☐	☑
The system provides clear instructions and labels.	☐	☐	☐	☑	☐
The process of accessing flood predictions is simple and efficient.	☐	☐	☐	☐	☑

Section 3: System Efficiency and Effectiveness

Statement	1	2	3	4	5
The system provides accurate flood predictions.	☐	☐	☐	☑	☐

The system updates predictions in a timely manner.	☐	☐	☐	☒	☐
The system effectively integrates visualization for flood mapping.	☐	☐	☐	☐	☒
The flood hazard levels (low, medium, high) are clearly distinguishable.	☐	☐	☐	☐	☒
The system loads quickly and operates smoothly.	☐	☐	☐	☐	☒

Section 4: Reliability and Security

Statement	1	2	3	4	5
The system provides reliable data without frequent errors.	☐	☐	☐	☒	☐
The system functions consistently without unexpected crashes.	☐	☐	☐	☐	☒
My data and information feel secure while using the system.	☐	☐	☐	☐	☐

Section 5: Overall Satisfaction and Recommendations

1. How satisfied are you with **FLOWS: A Predictive Fluvial Flood Mapping Web Application for the Municipality of Camaligan, Camarines Sur Using Machine Learning?**
   ☐ Very Dissatisfied
   ☐ Dissatisfied
   ☐ Neutral
   ☐ Satisfied
   ☒ Very Satisfied

2. Would you recommend the system to others?
   ☒ Yes
   ☐ No
   ☐ Not sure

3. What do you like most about the system? Easy to use by the public

4. What improvements would you suggest? Real time Rainfall data

Section 6: To assess the real-world performance and reliability of FLOWS, this section presents an evaluation of the system using **Typhoon Kristine (2024)** as a case study. By comparing predicted flood levels, affected areas, and the proximity of critical facilities, this case-based evaluation aims to demonstrate the system's practical relevance, accuracy, and potential contributions to disaster preparedness and response.

On a scale of 1 (Strongly Disagree) to 5 (Strongly Agree), please rate the following statements:

Statement	1	2	3	4	5
How closely did the predicted flood levels from FLOWS during Typhoon Kristine align with the actual reported water levels and flood-affected areas in Camaligan?	☐	☐	☐	☑	☐
Were the predicted high flood-risk areas and critical facilities identified by the system during Typhoon Kristine consistent with the locations that were actually affected or reported?	☐	☐	☐	☑	☐
If FLOWS had been actively deployed before Typhoon Kristine, do you think it might have improved early warning, evacuation planning, or resource allocation in the Municipality of Camaligan?	☐	☐	☐	☐	☑

Other related comment/s: ____________________________

Appendix E

FLOWS User's Manual

This user manual is designed to guide community members and emergency responders in effectively utilizing the FLOWS web application. The primary purpose of this manual is to assist users in navigating the system to access real-time predictive flood information for the Municipality of Camaligan. FLOWS is built to be accessible and intuitive, focusing on providing critical data for early decision-making during flood events.

The key features of the system include the ability to:

- Add multiple locations within Camaligan to monitor flood levels;
- View the flood risk level (color-coded and numerically displayed in meters) at each monitored location;
- Automatically identify and display nearby critical facilities, such as schools, hospitals, and evacuation centers; and
- Update the flood map and hazard information based on user-selected locations and prediction times.

The system aims to support flood preparedness, improve situational awareness, and strengthen community resilience during extreme weather events.

System Requirements

FLOWS is a lightweight web application that can be accessed through any modern web browser on both desktop and mobile devices. The system is optimized for fast performance and minimal buffering, ensuring a smooth user experience even on mobile data connections.

For this demonstration, the system is loaded with the flood simulation scenario based on Typhoon Kristine in Camaligan.

To access FLOWS:

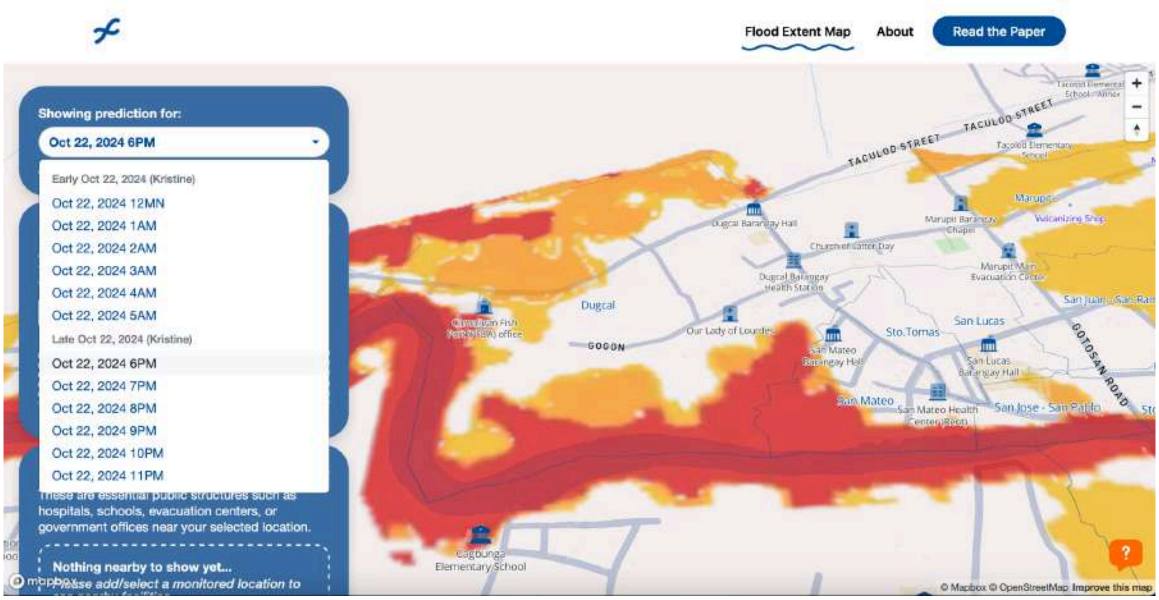
- Open any web browser (e.g., Google Chrome, Mozilla Firefox, Safari)
- Ensure that the device is connected to the internet, either via mobile data or Wi-Fi
- Type the following IP address into the browser's search bar:
  140.245.52.155
- Press Enter. The landing page will immediately display the map extent of Camaligan and available flood prediction layers.

The system is designed to be straightforward and intuitive, requiring no specialized technical skills for operation.

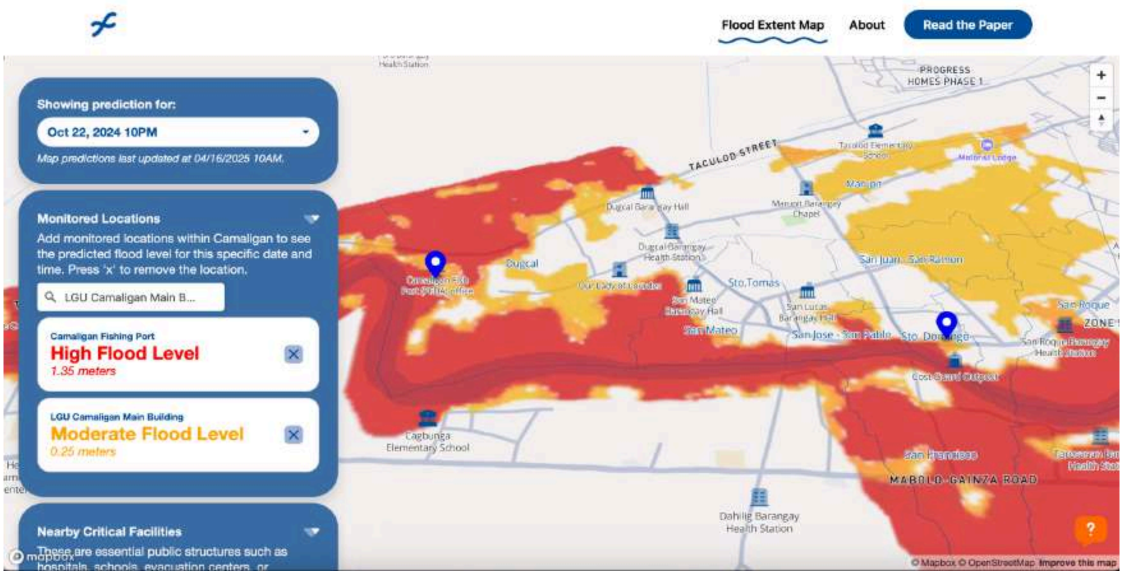
System Navigation

Upon accessing the FLOWS web application, users can easily interact with the following features:

1. **Prediction Dropdown (Upper Left Corner):**
   - Displays a selection of available date and time options for predicted flood levels
   - Users can choose a specific forecast hour to update the flood map accordingly.

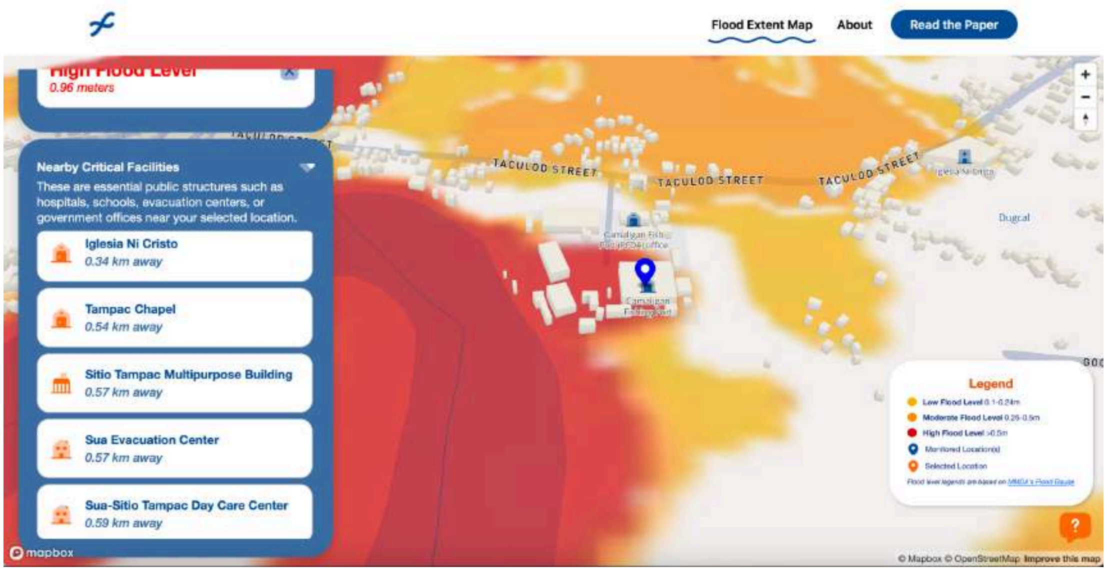


2. **Monitored Locations Panel (Middle Left)**
    - Users can search for a location within Camaligan using the search bar
    - Once selected, the location will be pinned on the map and added to the monitored list
    - For each monitored location, the system displays the associated flood risk level (color-coded) and the predicted water level in meters.

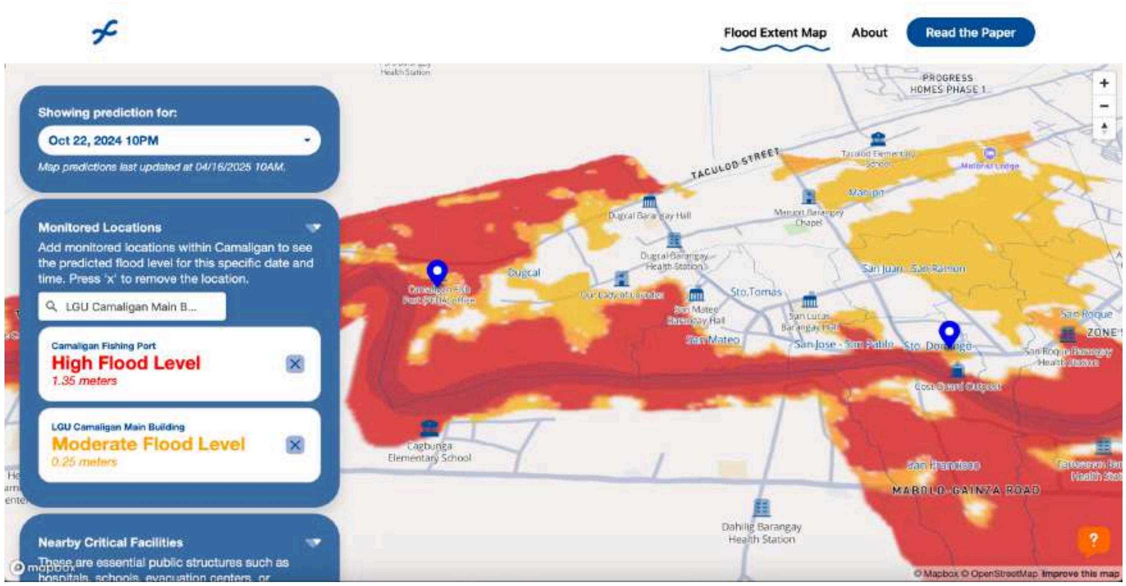


3. **Nearby Critical Facilities Display:**
    - When a location is monitored, the system also automatically lists nearby critical facilities within a 2 km radius.
    - These include hospitals, schools, government offices, evacuation centers, and more.

4. **Legend Button (Bottom Right Corner):**
    - Clicking the question mark icon opens the legend, explaining the color coding of flood risk levels (e.g., yellow for low risk, orange for moderate, red for high).



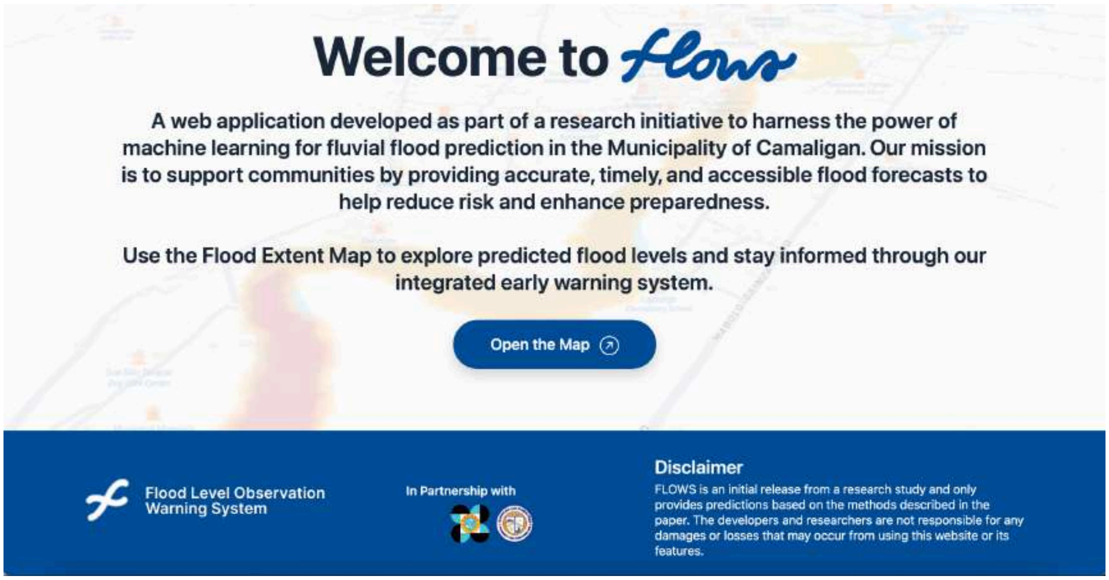
5. **Multiple Location Monitoring:**
    - Users can monitor several locations simultaneously.
    - The flood extent map updates dynamically based on the monitored points and their predicted conditions.



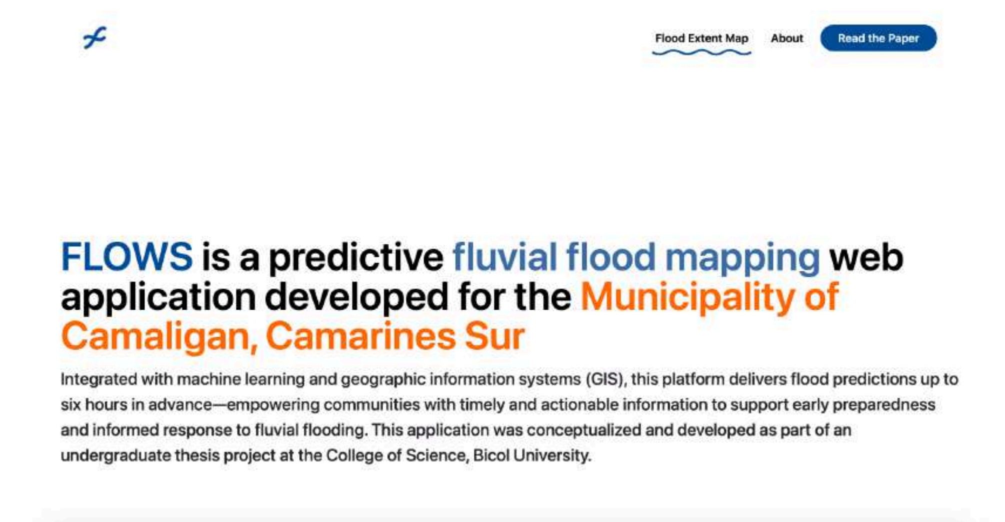
Extras: Navigation from the Header Menu

The header of the web application contains four interactive elements:

- **Logo (F):** Clicking the logo redirects to the Home/Welcome Page, which provides a brief introduction to the FLOWS system, lists the partner institutions, and displays a disclaimer.



- **Flood Extent Map:** Clicking this button brings users back to the main map page to view flood predictions.
- **About:** Redirects to an information page featuring an instructional video, detailed background on the project, partner institutions, and researcher contact information.



- **Reports:** This option redirects the user to the Flood Level Report showing a summary of the affected barangays and critical facilities for the next 6 hours.
- **Read the Paper:** When the research paper is published, this button will direct users to the official research publication for further reading.

Flood Level Reports

Barangay Flood Levels for October 23, 2024

These are based on the max flood level that was predicted within the boundaries of the barangay, including the river section.

Barangay	12 AM	1 AM	2 AM	3 AM	4 AM	5 AM
Sua	High	High	High	High	High	High
Marupit	High	High	High	High	High	High
Sto. Tomas	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate
San Juan - San Ramon	High	High	High	High	High	High
San Francisco	High	High	High	High	High	High
Tarosanan	High	High	High	High	High	High

Critical Facilities Flood Levels for October 23, 2024

These are essential public structures such as hospitals, schools, evacuation centers, or government offices near your selected location.

Government | Evacuation center | School | Hospital | Place of worship | Miscellaneous

Hospital

Critical Facility	12 AM	1 AM	2 AM	3 AM	4 AM	5 AM
San Juan San Ramon Barangay Health Center	High	High	High	High	High	High
San Mateo Health Center (Rent)	None	None	None	None	None	None
Municipal Health Office	Low	Low	Low	Low	Low	Low
Birthing Center	Moderate	Moderate	Moderate	Moderate	Moderate	Low
Municipal Nutrition Action Office	Moderate	Low	Moderate	Moderate	Low	Low
Sto.Domingo Barangay Health Center (1F)	High	High	High	High	High	High

Appendix F

Coding Snippets for Core Functions

1. Pre-processing the data

This code snippet outlines the data pre-processing of the gathered historical rainfall and water level data. This involved cleaning the data and interpolating the missing values using Python. The script also merges the dataset into one single CSV with the 6-hour interval flood prediction added based on the Water Level column.

```
# Data Preparation Code

rainfall = pd.read_csv("filtered_rainfall.csv")
waterlevel = pd.read_csv("waterlevel.csv")
waterlevel["Year"] = waterlevel["Year"].astype('int64')

merged_data = pd.merge(
   rainfall,
   waterlevel,
   on=['Year', 'Month', 'Day', 'Hour'],
   how='left'
)

merged_data.info()
merged_data.to_csv('merged_data.csv', index=False)
merged_data["WaterLevel"] = pd.to_numeric(merged_data['WaterLevel'],
errors='coerce')

merged_data["Rainfall"] = pd.to_numeric(merged_data['Rainfall'], errors='coerce')
merged_data["WaterLevel"] = merged_data["WaterLevel"].fillna(0)
merged_data["Rainfall"] = merged_data["Rainfall"].fillna(0)
merged_data["WaterLevel"] = merged_data["WaterLevel"].astype('float64')
merged_data["Rainfall"] = merged_data["Rainfall"].astype('float64')
merged_data['WaterLevel'] = merged_data['WaterLevel'].interpolate()
merged_data['Rainfall'] = merged_data['Rainfall'].interpolate()
finaldata = pd.read_csv("finaldata.csv")
finaldata.info()

finaldata["FWaterLevel"] = finaldata["WaterLevel"].shift(-6)
finaldata.dropna(inplace=True)
finaldata.to_csv("finaldata2.csv", index=False)
finaldata = pd.read_csv("finaldata2.csv")
finaldata = finaldata[finaldata["Rainfall"] != 0]
finaldata.info()
```

2. Training the model

This is the notebook script for model training, which utilized TensorFlow and Keras to format its model structure. The data was first formatted into a NARX dataset with the input and output lags as parameters.

```
# Model Training Code

# Imports
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.callbacks import EarlyStopping
import matplotlib.pyplot as plt
from sklearn.metrics import mean_absolute_error, r2_score
from tensorflow.keras.optimizers import Adam
from sklearn.metrics import mean_squared_error

finaldata = pd.read_csv("finaldata2.csv")
rainfall = finaldata['Rainfall'].values
waterlevel = finaldata['WaterLevel'].values
fwaterlevel = finaldata['FWaterLevel'].values

scaler_rainfall = MinMaxScaler()
scaler_waterlevel = MinMaxScaler()
scaler_fwaterlevel = MinMaxScaler()
rainfall_scaled = scaler_rainfall.fit_transform(rainfall.reshape(-1, 1))
waterlevel_scaled = scaler_waterlevel.fit_transform(waterlevel.reshape(-1, 1))
fwaterlevel_scaled = scaler_fwaterlevel.fit_transform(fwaterlevel.reshape(-1, 1))

# Creates NARX dataset from input data
def create_narx_dataset(waterlevel, rainfall, fwaterlevel, n_y, n_u):
    X, y = [], []
    for i in range(len(waterlevel) - max(n_y, n_u)):
        y_input = waterlevel[i:i+n_y].flatten()
        u_input = rainfall[i:i+n_u].flatten()
        X.append(np.concatenate((y_input, u_input)))
        y.append(fwaterlevel[i+n_y])
    return np.array(X), np.array(y)

# Calculates NSE score
def nash_sutcliffe(y_true, y_pred):
    numerator = np.sum((y_true - y_pred) ** 2)
    denominator = np.sum((y_true - np.mean(y_true)) ** 2)
    return 1 - (numerator / denominator)

# Input and output lags
n_y = 48
```

```
n_u = 48
X, y = create_narx_dataset(waterlevel_scaled, rainfall_scaled, fwaterlevel_scaled, n_y,
n_u)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=False)

model = Sequential()
model.add(Dense(64, input_dim=n_y + n_u, activation='tanh'))
model.add(Dense(64, activation='tanh'))
model.add(Dense(32, activation='tanh'))
model.add(Dense(1, activation='linear'))

model.compile(optimizer=Adam(learning_rate=0.0001), loss='mse', metrics=['mae'])

history = model.fit(
    X_train, y_train,
    epochs=100,
    batch_size=8,
    validation_split=0.2,
    verbose=1,
)

y_pred = model.predict(X_test)

y_test_orig = scaler_fwaterlevel.inverse_transform(y_test.reshape(-1, 1))
y_pred_orig = scaler_fwaterlevel.inverse_transform(y_pred)

# Prints Model metrics

mse = mean_squared_error(y_test_orig, y_pred_orig)
rmse = np.sqrt(mse)

mae = mean_absolute_error(y_test_orig, y_pred_orig)
r2 = r2_score(y_test_orig, y_pred_orig)
nse = nash_sutcliffe(y_test_orig, y_pred_orig)

print(f"\nModel Performance Metrics:")
print(f"Mean Absolute Error (MAE): {mae:.5f}")
print(f"R² Score: {r2:.2f}")
print(f"Nash-Sutcliffe Efficiency (NSE): {nse:.2f}")
print(f"Mean Squared Error (MSE): {mse:.5f}")
print(f"Root Mean Squared Error (RMSE): {rmse:.5f}")

plt.figure(figsize=(10, 5))
plt.plot(history.history['loss'], label='Training Loss', color='blue')
plt.plot(history.history['val_loss'], label='Validation Loss', color='red')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.title('Loss Curve')
```

```
plt.legend()
plt.show()

# Print predictions vs actual values
print("\nPredictions vs Actual Values:")
for i in range(min(500, len(y_pred_orig))):  # Print first 20
    print(f"Prediction: {y_pred_orig[i][0]:.3f}, Actual: {y_test_orig[i][0]:.3f}")

# Save model format to tensorflow file
model.save('flows_modelv4.h5')

# Load the model
model = tf.keras.models.load_model("flows_modelv4.h5")
```

3. Connecting the automation scripts

This code is responsible for running all the scripts for the automation sequentially. It includes running the model to provide predictions, editing the input data for the unsteady flow simulation, running the simulation in HEC-RAS, exporting the resulting raster maps, adding a color ramp to the maps and converting them to 8-bit format, and uploading the final raster maps to Mapbox.

```
def run_scripts_with_custom_delays(scripts, delays):
    for idx, script in enumerate(scripts):
        print(f"Running script: {script}")
        try:
            result = subprocess.run(["python", script], capture_output=True, text=True)
            print(result.stdout)

            if result.returncode == 0:
                print(f"Script finished: {script}")
            else:
                print(f"Script failed: {script}")
                print(result.stderr)

        except Exception as e:
            print(f"Error running {script}: {e}")

        if idx < len(delays):
            delay = delays[idx]
            print(f"Waiting {delay} seconds before next script...\n")
            time.sleep(delay)
```

4. Predicting the water levels using the model

This code is responsible for creating a NARX-formatted dataset based on the input historical data. This dataset is fed to the model as input and returns a CSV file containing the 6-hour interval water level predictions.

```
# Load dataset
data = pd.read_csv(CSV_LOCATION)

# Scale training data (before the specified date and time)
def load_historical_data(year, month, day):
    start_dt = datetime(year, month, day)

    historical_data = data.copy()
    historical_data['Datetime'] = pd.to_datetime(
        historical_data[['Year', 'Month', 'Day', 'Hour']], errors='coerce'
    )

    historical_data = historical_data.dropna(subset=['Datetime'])
    historical_data = historical_data[historical_data['Datetime'] < start_dt]

    return historical_data


def apply_scaling(historical_data):
    global rainfall, waterlevel, fwaterlevel, scaler_rainfall, scaler_waterlevel,
scaler_fwaterlevel

    rainfall = historical_data['Rainfall'].values
    waterlevel = historical_data['WaterLevel'].values
    fwaterlevel = historical_data['FWaterLevel'].values

    scaler_rainfall = MinMaxScaler().fit(rainfall.reshape(-1, 1))
    scaler_waterlevel = MinMaxScaler().fit(waterlevel.reshape(-1, 1))
    scaler_fwaterlevel = MinMaxScaler().fit(fwaterlevel.reshape(-1, 1))

model = tf.keras.models.load_model(MODEL_LOCATION, custom_objects={'mse':
mse})

def apply_scaling_to_data(monthly_data):
    monthly_data['Rainfall_scaled'] =
scaler_rainfall.transform(monthly_data['Rainfall'].values.reshape(-1, 1)).flatten()
    monthly_data['WaterLevel_scaled'] =
scaler_waterlevel.transform(monthly_data['WaterLevel'].values.reshape(-1, 1)).flatten()
    return monthly_data

def load_data_for_month(year, month):
    monthly_data = data[(data['Year'] == year) & (data['Month'] == month)].copy()
    monthly_data['Datetime'] = pd.to_datetime(monthly_data[['Year', 'Month', 'Day',
'Hour']])
    monthly_data.dropna(subset=['Datetime'], inplace=True)
```

```
    monthly_data = monthly_data.sort_values('Datetime').reset_index(drop=True)
    return monthly_data

def create_narx_input(dataframe, n_y, n_u, prediction_step=1):
    sequences = []
    timestamps = []
    for i in range(len(dataframe) - max(n_y, n_u) - prediction_step + 1):
        y_input = dataframe['WaterLevel_scaled'].values[i:i+n_y]
        u_input = dataframe['Rainfall_scaled'].values[i:i+n_u]
        seq = np.concatenate((y_input, u_input))
        sequences.append(seq)

        # Predict for the time right after input
        target_time = dataframe['Datetime'].iloc[i + max(n_y, n_u) + prediction_step - 1]
        timestamps.append(target_time)
    return np.array(sequences), pd.to_datetime(timestamps)

def get_start_index(ts_all):
    if os.path.exists(LOG_PATH):
        with open(LOG_PATH, 'r') as f:
            last_index = int(f.read().strip())
    else:
        # Try to find exact match first
        match_idx = np.where(ts_all == pd.Timestamp(START_DATETIME))[0]
        if len(match_idx) == 0:
            print(f"START_DATETIME {START_DATETIME} not found in data. Using first
available timestamp: {ts_all[0]}")
            last_index = 0
        else:
            last_index = match_idx[0]
    return last_index

def move_to_next_month(current_year, current_month):
    next_month = current_month + 1
    next_year = current_year
    if next_month > 12:
        next_month = 1
        next_year += 1
    return next_year, next_month

# === Run Prediction Sequence ===

current_year = START_YEAR
current_month = START_MONTH
current_day = START_DAY

historical_data = load_historical_data(current_year, current_month, current_day)
apply_scaling(historical_data)
```

```
data_load = load_data_for_month(current_year, current_month)
data_load = apply_scaling_to_data(data_load)

X_all, ts_all = create_narx_input(data_load, n_y, n_u)

# Get start index based on log or START_DATETIME
start_idx = get_start_index(ts_all)
end_idx = start_idx + PREDICT_HOURS

if end_idx > len(X_all):
    print("No more predictions available in current month, moving to the next month.")
    current_year, current_month = move_to_next_month(current_year, current_month)
    historical_data = load_historical_data(current_year, current_month, 1)  # Fallback to
day=1
    if len(historical_data) == 0:
        print("Warning: No historical data available for the next month.")
        exit()

    apply_scaling(historical_data)
    data_load = load_data_for_month(current_year, current_month)

    if len(data_load) == 0:
        print("Warning: No data available for the next month.")
        exit()
    else:
        data_load = apply_scaling_to_data(data_load)
        X_all, ts_all = create_narx_input(data_load, n_y, n_u)
        start_idx = 0
        end_idx = start_idx + PREDICT_HOURS

X_batch = X_all[start_idx:end_idx]
ts_batch = ts_all[start_idx:end_idx]

# Predict
pred_scaled = model.predict(X_batch)
pred = scaler_fwaterlevel.inverse_transform(pred_scaled).flatten()

# Offset timestamps by 2 days
ts_batch_offset = ts_batch - pd.Timedelta(days=2)

result_df = pd.DataFrame({
    'Year': ts_batch_offset.year,
    'Month': ts_batch_offset.month,
    'Day': ts_batch_offset.day,
    'Hour': ts_batch_offset.hour,
    'PredictedFWaterLevel': pred
})
```

```
# Save predictions to CSV
result_df.to_csv(OUTPUT_CSV, mode='w', header=True, index=False)

# Update log
with open(LOG_PATH, 'w') as f:
    f.write(str(end_idx))

print(f"Saved predictions for {ts_batch[0]} to {ts_batch[-1]} -> {OUTPUT_CSV}")
```

5. Simulating the predicted water levels on HEC-RAS

This code is responsible for initializing the correct HECRASController provided with the software, updating the u01 file that contains the water level data that will be used for the simulation, and running the unsteady flow data simulation.

```
try:
    # Initialize HEC-RAS Controller
        hec = win32com.client.Dispatch("RAS67.HECRASController")   # Adjust for your
HEC-RAS version
    logging.info("HEC-RAS Controller initialized.")

    # File paths
    project_file = os.path.abspath(os.path.join(BASE_DIR, '..', 'flows.prj'))
    unsteady_file = os.path.abspath(os.path.join(BASE_DIR, '..', 'flows.u01'))
              csv_file    =    os.path.abspath(os.path.join(BASE_DIR,    '..',    '..',    'assets',
'flood_predictions.csv'))

    # Load prediction CSV
    df = pd.read_csv(csv_file)

    if 'PredictedFWaterLevel' not in df.columns:
        raise ValueError("'PredictedFWaterLevel' column not found in CSV.")

    # Get the first 6 predicted water levels
    water_levels = df['PredictedFWaterLevel'].values[:6]
    if len(water_levels) < 6:
        raise ValueError("Not enough predicted water level values (need at least 6).")

    # Optional processing: scale, clip, or adjust as needed
    factor = 1.5
    adjusted_levels = np.clip(water_levels * factor, 0, None)

    # Enforce a minimum of 1.400 for HEC-RAS rendering
    adjusted_levels = np.clip(adjusted_levels, 1.700, None)

    # Create a DataFrame for logging and formatting
    df_levels = pd.DataFrame({"Water_Level": adjusted_levels})

    # Update .u01 unsteady flow file
```

```
    def update_unsteady_flow(file_path, levels_df):
        with open(file_path, "r") as f:
            lines = f.readlines()

        stage_count = len(levels_df)
        formatted_stages = ["{:.3f}".format(val) for val in levels_df["Water_Level"]]

        # Unsteady Flow Formatting
        stage_line = "   " + "   ".join(formatted_stages) + "\n"

        # Find and replace Stage Hydrograph block
        for i, line in enumerate(lines):
            if line.strip().startswith("Stage Hydrograph="):
                lines[i] = f"Stage Hydrograph= {stage_count} \n"
                lines[i + 1] = stage_line
                break
        else:
            raise ValueError("'Stage Hydrograph=' section not found in the .u01 file.")

        with open(file_path, "w") as f:
            f.writelines(lines)

        logging.info("Unsteady flow file updated successfully.")
        logging.info(f"Stage Hydrograph= {stage_count}")
        logging.info("Formatted Values:\n" + stage_line)

    # Run HEC-RAS
    def run_hec_ras():
        RAS = win32com.client.Dispatch("RAS67.HECRASController")
        RAS.Project_Open(project_file)
        current_plan = RAS.CurrentPlanFile()
        if not current_plan:
            raise RuntimeError("Failed to load the HEC-RAS project. Check the file path.")
        # Explicitly open the unsteady flow data
        success = RAS.Compute_CurrentPlan()

        if success:
            print("HEC-RAS Unsteady Flow Simulation completed successfully.")
        else:
            print("Error running HEC-RAS Unsteady Flow Simulation.")

    # Execute steps
    update_unsteady_flow(unsteady_file, df_levels)
    run_hec_ras()

except Exception as e:
    logging.error(f"Error: {e}")
    hec.QuitRas()
    logging.info("HEC-RAS closed successfully.")
```

6. Converting the flood extent raster maps

This code is tasked to change the colors of the raster map according to the legend of yellow for water values of 0<x<0.25, orange for 0.25<=x<0.5, and red for flood values of >=0.5, based on its band-1 value (water level). After adding the color ramp, the raster maps need to be converted from 32-bit geotiff format to 8-bit geotiff format as required by Mapbox for uploading.

```
def convert_to_8bit_rasterio(input_path, output_path_colormap, scale_min=None,
scale_max=None, target_crs='EPSG:32651'):
    with rasterio.open(input_path) as src:
        profile = src.profile.copy()

        # Assign CRS if missing or incorrect
        if not src.crs or src.crs.to_string() != target_crs:
            print(f"Assigning CRS {target_crs} to {os.path.basename(input_path)}...")
            profile.update({'crs': CRS.from_string(target_crs)})
        else:
            print(f"{os.path.basename(input_path)} CRS is already {src.crs}")

        # Read data and mask NoData
        data = src.read(1).astype(np.float32)
        nodata_value = src.nodata if src.nodata is not None else -9999
        valid_mask = data != nodata_value
        data_valid = data[valid_mask]

        if scale_min is None or scale_max is None:
            scale_min = np.nanmin(data_valid)
            scale_max = np.nanmax(data_valid)

        # Scale valid data to 0–255
        scaled = np.zeros_like(data, dtype=np.uint8)
        scaled[valid_mask] = np.clip(
            ((data_valid - scale_min) / (scale_max - scale_min)) * 255, 0, 255
        ).astype(np.uint8)

        # Update profile
        profile.update({
            'dtype': 'uint8',
            'count': 1,
            'nodata': 0,
            'photometric': 'palette'
        })

        # Define colormap
        colormap = {}
        for value in range(1, 22):       # Yellow
            colormap[value] = (244, 180, 0)
```

```
        for value in range(23, 43):     # Orange
            colormap[value] = (251, 140, 0)
        for value in range(44, 256):    # Red
            colormap[value] = (213, 0, 0)

    # Write colormapped 8-bit output
    with rasterio.open(output_path_colormap, 'w', **profile) as dst:
        dst.write(scaled, 1)
        dst.write_colormap(1, colormap)
        dst.update_tags(scale_min=str(scale_min), scale_max=str(scale_max))

    print(f"Converted to 8-bit with colormap: {os.path.basename(output_path_colormap)}")

def convert_colormap_to_rgb(input_path_colormap, output_path_rgb):
    with rasterio.open(input_path_colormap) as src:
        band = src.read(1)
        colormap = src.colormap(1)

        # Convert colormap to RGB bands
        rgb = np.zeros((3, band.shape[0], band.shape[1]), dtype=np.uint8)
        for value, color in colormap.items():
            mask = band == value
            rgb[0][mask] = color[0]  # Red
            rgb[1][mask] = color[1]  # Green
            rgb[2][mask] = color[2]  # Blue

        # Update profile
        profile = src.profile.copy()
        profile.update({
            'count': 3,
            'dtype': 'uint8',
            'photometric': 'RGB',
            'nodata': 0
        })

        # Write RGB output
        with rasterio.open(output_path_rgb, 'w', **profile) as dst:
            dst.write(rgb[0], 1)
            dst.write(rgb[1], 2)
            dst.write(rgb[2], 3)

    print(f"Converted to RGB: {os.path.basename(output_path_rgb)}")

# === Batch Processing ===

BASE_DIR = os.path.dirname(os.path.abspath(__file__))

input_folder = os.path.abspath(os.path.join(BASE_DIR, '..', 'test'))
```

```
output_folder = os.path.abspath(os.path.join(BASE_DIR, '..', '..', 'assets',
'converted_rgb'))
copied_folder = os.path.abspath(os.path.join(BASE_DIR, '..', '..', 'assets', 'original'))

os.makedirs(copied_folder, exist_ok=True)
os.makedirs(output_folder, exist_ok=True)

tif_files = [f for f in os.listdir(input_folder) if f.lower().endswith(".tif")]

for idx, tif_file in enumerate(tif_files):
    input_path = os.path.join(input_folder, tif_file)
    temp_colormap_path = os.path.join(output_folder, f"temp_colormap_{idx}.tif")
    final_rgb_path = os.path.join(output_folder, f"tif_rgb_{idx}.tif")

    convert_to_8bit_rasterio(input_path, temp_colormap_path)
    convert_colormap_to_rgb(temp_colormap_path, final_rgb_path)

    os.remove(temp_colormap_path)
    print(f"Removed temp file: {os.path.basename(temp_colormap_path)}")

    renamed_original_path = os.path.join(copied_folder, f"tif_rgb_{idx}.tif")
    shutil.copy2(input_path, renamed_original_path)

    print(f"Original copied and renamed: {os.path.basename(renamed_original_path)}")

print("\nAll files converted to RGB for Mapbox upload.")
```

7. Uploading the raster maps

This code is responsible for requesting the credentials from S3 and Mapbox that will enable the script to upload the flood extent raster maps to Mapbox, which will be displayed on the web application.

```
def upload_single_tif(tif_path, index):
    try:
        # 1. Request temporary S3 credentials
        cred_res = requests.post(

f'https://api.mapbox.com/uploads/v1/{username}/credentials?access_token={mapbox_to
ken}'
        )
        credentials = cred_res.json()

        # 2. Upload TIFF to S3
        s3 = boto3.client(
            's3',
            aws_access_key_id=credentials['accessKeyId'],
            aws_secret_access_key=credentials['secretAccessKey'],
            aws_session_token=credentials['sessionToken'],
```

```
        )

        bucket = credentials['bucket']
        key = credentials['key']
        with open(tif_path, 'rb') as f:
            s3.upload_fileobj(f, bucket, key)
        print(f"Uploaded {os.path.basename(tif_path)} to S3")

        # 3. Trigger tileset upload
        tileset_id = f'{username}.flood-map-{index}'
        upload_payload = {
            "url": credentials["url"],
            "tileset": tileset_id,
            "name": f"Flood Map {index}"
        }
        upload_res = requests.post(

f'https://api.mapbox.com/uploads/v1/{username}?access_token={mapbox_token}',
            json=upload_payload
        )
        upload_job = upload_res.json()
        print(f"Upload started for {tileset_id} (Upload ID: {upload_job['id']})")

    except Exception as e:
        print(f"Error uploading {os.path.basename(tif_path)}: {e}")

def upload_all_tifs():
    tif_files = sorted(glob(os.path.join(input_folder, "*.tif")))
    print(f"Found {len(tif_files)} TIFF files to upload.\n")

    for index, tif_path in enumerate(tif_files):
        print(f"Processing {index+1}/{len(tif_files)}: {os.path.basename(tif_path)}")
        upload_single_tif(tif_path, index)
        time.sleep(10)  # optional delay to avoid rate limits

upload_all_tifs()
```

8. Flood level reports per barangay and critical facilities

This code is responsible for generating the flood level reports per barangay and critical facilities. This was done by using the rasterio package and overlapping the boundary of each barangay and fetching the maximum flood level seen within it. For the critical facilities, its coordinates are used for fetching the flood level at the location. The results are exported into CSV files: One for barangays and one for critical facilities.

```
# 1. Load Date/Time strings first
date_time_headers = load_and_format_datetime_csv(datetime_csv_path)
if date_time_headers is None:
    print("Exiting due to error loading datetime CSV.")
```

```
    exit()

# ** Crucial Check **: Ensure number of headers matches number of GeoTIFFs
if len(date_time_headers) != len(geotiff_files):
    print(f"Error: Mismatch between number of datetime entries ({len(date_time_headers)}) and GeoTIFF files ({len(geotiff_files)}).")
    print("Please ensure the datetime CSV and the geotiff_files list correspond correctly.")
    exit()

# 2. Load Critical Facilities Data
try:
    print(f"Loading critical facilities from: {facilities_geojson_path}")
    gdf_facilities = gpd.read_file(facilities_geojson_path)
    # Ensure the name column exists
    if facility_name_column not in gdf_facilities.columns:
        raise ValueError(f"Column '{facility_name_column}' not found in {facilities_geojson_path}. Available properties: {list(gdf_facilities.columns)}")

    print(f"Loaded {len(gdf_facilities)} facility features.")
    print(f'Available properties: {list(gdf_facilities.columns)}')
    # Ensure geometry is points
    if not gdf_facilities.geom_type.isin(['Point']).all():
        print("Warning: GeoJSON contains non-point geometries. Sampling will only work for points.")
        # Optionally filter: gdf_facilities = gdf_facilities[gdf_facilities.geom_type == 'Point']

except Exception as e:
    print(f"Error loading facilities GeoJSON data: {e}")
    exit()

# Create the base report DataFrame with facility names
report_df = gdf_facilities[[facility_name_column]].copy()
report_df.rename(columns={facility_name_column: 'Critical Facility'}, inplace=True)
# Optional: Add other attributes from the GeoJSON to the report
report_df['Amenity'] = gdf_facilities[[amenity_name_column]].copy()

# 3. Process each GeoTIFF using an index
print("\nProcessing GeoTIFF files for point sampling...")
for i, filename in enumerate(geotiff_files):
    raster_path = os.path.join(geotiff_directory, filename)
    if not os.path.exists(raster_path):
        print(f"Warning: File not found, assigning 'File Not Found': {raster_path}")
        report_df[date_time_headers[i]] = 'File Not Found'
        continue

    # Use the formatted date/time string from the list as the column header
    col_name = date_time_headers[i]
    print(f"  Sampling raster: {filename} -> Column: '{col_name}'")
```

```
    try:
        with rasterio.open(raster_path) as src:
            raster_crs = "EPSG:32651"
            nodata_val = src.nodata # Get the raster's NoData value

            # Ensure facilities are in the same CRS as the raster
            gdf_facilities_reprojected = gdf_facilities
            if gdf_facilities.crs != raster_crs:
                print(f"    Reprojecting points from {gdf_facilities.crs} to {raster_crs}...")
                gdf_facilities_reprojected = gdf_facilities.to_crs(raster_crs)

            # Extract coordinates in the raster's CRS
            coords = [(p.x, p.y) for p in gdf_facilities_reprojected.geometry]

            # Sample the raster at the facility coordinates
            # src.sample returns a generator yielding lists (one value per band)
            sampled_values_generator = src.sample(coords)

            # Extract value from the first band, handle NoData
            flood_levels = []
            for val_list in sampled_values_generator:
                val = val_list[0] # Get value from first band
                if nodata_val is not None and val == nodata_val:
                    flood_levels.append(None) # Use None if it's NoData
                else:
                    flood_levels.append(val)

            # Add results to the report DataFrame
            if len(flood_levels) == len(report_df):
                report_df[col_name] = flood_levels

            else:
                     print(f"Warning: Mismatch in feature count during sampling for {filename}.
Assigning 'Error'.")
                report_df[col_name] = 'Error'

    except Exception as e:
        print(f"Error sampling raster {filename}: {e}")
        report_df[col_name] = 'Sampling Error'


# 4. Clean up and Export Report
print("\nFinalizing report...")
# Replace None values (NoData or errors) with 'N/A' or 0.0
# Using 0.0 might be better if 0 represents no flood depth
report_df.fillna(0.0, inplace=True) # Or use 'N/A'

# Optional: Round numeric values (be careful with 'N/A' or 'Error')
for col in report_df.columns:
```

```
    if col != 'Critical Facility' and col != 'Amenity':
        report_df[col] = pd.to_numeric(report_df[col], errors='ignore') # Ignore non-numeric
        # Check dtype before rounding
        if pd.api.types.is_numeric_dtype(report_df[col]):
            report_df[col] = report_df[col].round(2) # Example: 2 decimal places
        report_df[col] = report_df[col].apply(classify_flood_level) # Classify flood levels

# Add units description to column headers
#report_df.columns = ['Critical Facility'] + [f'{col} ({stats_output_name})' for col in
report_df.columns if col != 'Critical Facility']

# Export to CSV
try:
    report_df.to_csv(output_report_csv, index=False)
    print(f"\nReport successfully saved to: {output_report_csv}")
except Exception as e:
    print(f"\nError saving report to CSV: {e}")

def get_flood_stats(vector_path, raster_path, stats_list, nodata_val=None):
    """Calculates zonal stats for a given vector and raster file."""
    try:
        with rasterio.open(raster_path) as src:
            affine = src.transform
            array = src.read(1) # Read the first band
            # Use the GeoDataFrame directly if CRS matches, otherwise path is safer
            stats = zonal_stats(
                vector_path, # Use path to handle potential CRS differences initially
                # array, # Using array directly is faster if CRS matches
                # affine=affine,
                raster=raster_path, # Let rasterstats handle CRS check/reprojection if needed
                stats=stats_list,
                    nodata=nodata_val if nodata_val is not None else src.nodata, # Use nodata
value from raster
                all_touched=True # Include pixels touched by the boundary
            )
        # Extract the primary stat (e.g., 'max')
        return [s.get(stats_list[0]) if s else None for s in stats]
    except Exception as e:
        print(f"Error processing {raster_path}: {e}")
        # Return None for all features if this raster fails
        try:
            gdf_tmp = gpd.read_file(vector_path)
            return [None] * len(gdf_tmp)
        except:
            return [] # Or handle error differently

# --- Main Script ---
date_time_headers = load_and_format_datetime_csv(datetime_csv_path)
if date_time_headers is None:
```

```
    print("Exiting due to error loading datetime CSV.")
    exit()

# ** Crucial Check **: Ensure number of headers matches number of GeoTIFFs
if len(date_time_headers) != len(geotiff_files):
    print(f"Error: Mismatch between number of datetime entries ({len(date_time_headers)}) and GeoTIFF files ({len(geotiff_files)}).")
    print("Please ensure the datetime CSV and the geotiff_files list correspond correctly.")
    exit()


# 2. Load Barangay Data
try:
    print(f"Loading barangay boundaries from: {barangay_vector_path}")
    gdf_barangays = gpd.read_file(barangay_vector_path)
    if barangay_name_column not in gdf_barangays.columns:
        raise ValueError(f"Column '{barangay_name_column}' not found in {barangay_vector_path}. Available columns: {gdf_barangays.columns.tolist()}")
    print(f"Loaded {len(gdf_barangays)} barangay features.")
except Exception as e:
    print(f"Error loading barangay data: {e}")
    exit()

# Create the base report DataFrame
report_df = gdf_barangays[[barangay_name_column]].copy()
report_df.rename(columns={barangay_name_column: 'Barangay'}, inplace=True)

# 3. Process each GeoTIFF using an index
print("\nProcessing GeoTIFF files and assigning datetime headers...")
for i, filename in enumerate(geotiff_files):
    raster_path = os.path.join(geotiff_directory, filename)
    if not os.path.exists(raster_path):
        print(f"Warning: File not found, skipping: {raster_path}")
        # Assign 'File Not Found' to the corresponding column
        report_df[date_time_headers[i]] = 'File Not Found'
        continue

    # Use the formatted date/time string from the list as the column header
    col_name = date_time_headers[i]
    print(f"  Calculating stats for: {filename} -> Column: '{col_name}'")

    # Calculate stats for the current raster
    flood_levels = get_flood_stats(gdf_barangays, raster_path, stats_to_calculate) # Pass gdf for CRS check

    # Add results to the report DataFrame with the datetime header
    if len(flood_levels) == len(report_df):
        # Add the statistic description to the value? Or keep header clean? Keep header clean for now.
```

```
        # Maybe add units to header later?
        report_df[col_name] = flood_levels
    else:
        print(f"Warning: Mismatch in feature count for {filename}. Assigning 'Error'.")
        report_df[col_name] = 'Error'




# 4. Clean up and Export Report
print("\nFinalizing report...")
report_df.fillna(0, inplace=True) # Replace None with 'N/A' (no overlap or NoData)

# Optional: Round numeric values (be careful with 'N/A' or 'Error')
for col in report_df.columns:
    if col != 'Barangay':
        report_df[col] = pd.to_numeric(report_df[col], errors='ignore') # Ignore non-numeric
        if pd.api.types.is_numeric_dtype(report_df[col]):
            report_df[col] = report_df[col].round(2) # Example: 2 decimal places
        #report_df[col] = report_df[col].apply(classify_flood_level) # Classify flood levels

count = 0
for hour in report_df.columns[1:]:
    # Sort by flood value (descending), get top 5
    top5 = report_df[['Barangay', hour]].sort_values(by=hour, ascending=False).head(5)
    filename = f"assets/reports/top5_{count}.csv"
    count += 1

    # Save to CSV
    top5.to_csv(filename, index=False)

for col in report_df.columns:
    if col != 'Barangay':
        report_df[col] = report_df[col].apply(classify_flood_level) # Classify flood levels

# Export to CSV
try:
    report_df.to_csv(output_report_csv, index=False)
    print(f"\nReport successfully saved to: {output_report_csv}")
except Exception as e:
    print(f"\nError saving report to CSV: {e}")
```

9. Back-end of web application

This code is responsible for requesting the credentials from S3 and Mapbox that will enable the script to upload the flood extent raster maps to Mapbox, which will be displayed on the web application.

```
function successLocation(position) {
        // console.log(position);
        setupMap([position.coords.longitude, position.coords.latitude]);
    }

    function errorLocation() {
        setupMap([123.160705, 13.623460]);
    }

    function setupMap(center) {
        const map = new mapboxgl.Map({
            container: 'map', // container ID
            style: 'mapbox://styles/gavinciii/cm9ro1smy008b01spb1yydza4', // style URL
            //style: 'mapbox://styles/mapbox/satellite-streets-v12',
            center: [123.160705, 13.623460], // starting position [lng, lat]. Note that lat must
be set between -90 and 90
            zoom: 15.5, // starting zoom
            pitch: 45
        });
        const totalTilesets = 6; // 0 through 11

        map.on('load', () => {
            console.log("Map loaded.");

            for (let i = 0; i < totalTilesets; i++) {
                const tilesetId = `flood-map-${i}`;
                const sourceId = `source_${tilesetId}`;
                const layerId = `layer_${tilesetId}`;

                // Safety check: Don't re-add if already exists (useful for hot-reloading dev
envs)
                if (!map.getSource(sourceId)) {
                    map.addSource(sourceId, {
                        type: 'raster',
                        url: `mapbox://${username}.${tilesetId}`,
                        tileSize: 256
                    });
                    console.log(`AddedSource: ${sourceId}`);
                }

                if (!map.getLayer(layerId)) {
                    map.addLayer({
                        id: layerId,
                        type: 'raster',
                        source: sourceId,
                        paint: {
                            'raster-opacity': 0.7
                        },
                        layout: {
```

```
                    'visibility': (i === 0) ? 'visible' : 'none' // Only layer 0 visible initially
                },
                slot: 'middle'
            },
            );
            console.log(`AddedSource: ${layerId}`);
        }
    }
    console.log("Initial layers added or confirmed.");
    loadTop5FromCSV(0);

    // --- NEW Event listener setup for MULTIPLE Bootstrap Dropdowns ---
    const dropdownMenus = document.querySelectorAll('.tileset-dropdown-menu'); //
Select ALL menus by common class

    if (dropdownMenus.length > 0) {
        dropdownMenus.forEach(menu => { // Attach listener to each menu found
            menu.addEventListener('click', (event) => {
                // Check if the clicked element is a dropdown item with the data attribute
                if (event.target.matches('.dropdown-item[data-tileset-index]')) {
                    event.preventDefault(); // Prevent default anchor link behavior

                    const selectedIndex = parseInt(event.target.dataset.tilesetIndex, 10);
                    const selectedText = event.target.textContent;
                    layerIndex = selectedIndex;

                    console.log(`Dropdown item clicked. Selected index:
${selectedIndex}`);

                    // Find the button associated *specifically* with the clicked menu
                    const parentDropdown = event.target.closest('.dropdown'); // Find the
parent .dropdown container
                    if (parentDropdown) {
                        const button = parentDropdown.querySelector('.dropdown-toggle'); //
Find the button within that container
                        if (button) {
                            button.textContent = selectedText; // Update the correct button's
text
                        }
                    }


                    // --- Map Layer Visibility Logic (Stays the Same) ---
                    // This part affects the single map globally based on the index
                    for (let i = 0; i < totalTilesets; i++) {
                        const layerId = `layer_flood-map-${i}`;
                        if (map.getLayer(layerId)) {
                            const visibility = (i === selectedIndex) ? 'visible' : 'none';
                            map.setLayoutProperty(layerId, 'visibility', visibility);
```

```
                } else {
                    console.warn(`Layer ${layerId} not found on map during visibility
toggle.`);
                }
            }
            // --- End Map Layer Logic ---

            refreshMonitoredLocations(selectedIndex, map);
            loadTop5FromCSV(selectedIndex);

        }
    });
});
console.log(`Bootstrap dropdown event listeners added to
${dropdownMenus.length} menus.`);

// Set initial button text for ALL dropdowns based on default layer (layer 0)
dropdownMenus.forEach(menu => {
    const initialItem = menu.querySelector('[data-tileset-index="0"]');
    const parentDropdown = menu.closest('.dropdown');
    if (initialItem && parentDropdown) {
        const button = parentDropdown.querySelector('.dropdown-toggle');
        if (button) {
            button.textContent = initialItem.textContent;
        }
    }
});

} else {
    console.error("Could not find any dropdown menu elements with class
'tileset-dropdown-menu'.");
}
});

map.addControl(new mapboxgl.NavigationControl(), 'top-right');
// map.addControl(new mapboxgl.GeolocateControl({
//     positionOptions: { enableHighAccuracy: true },
//     trackUserLocation: true
// }), 'top-right');

fetch('assets/datasets/Camaligan-Crit-Facilities.geojson') // Load the downloaded
GeoJSON file
.then(response => response.json())
.then(geojsonData => {

    // Define localGeocoder function for searching within the dataset
    function localGeocoder(query) {
        // console.log("Geocoder Search Query:", query); // Debug search query
```

```
    const results = geojsonData.features
      .filter(feature =>
        feature.properties &&
        feature.properties.name &&
        feature.properties.name.toLowerCase().includes(query.toLowerCase())
      )
      .map(feature => ({
        text: feature.properties.name, // Display name in search results
        place_name: feature.properties.name,
        center: feature.geometry.coordinates, // Coordinates for result
        geometry: feature.geometry,
      }));

    // console.log("Filtered Results:", results); // Debug filtered results
    return results;
  }



  // Initialize MapboxGeocoder with local dataset search
  const geocoder = new MapboxGeocoder({
    accessToken: mapboxgl.accessToken,
    localGeocoder: localGeocoder,
    zoom: 15,
    mapboxgl: mapboxgl,
    placeholder: 'Search for a location',
    container: document.getElementById('search-container')
  });

    // Function to add the geocoder to a specified container
  function addGeocoderToContainer(containerId) {
    const container = document.getElementById(containerId);
    container.appendChild(geocoder.onAdd(map));
  }

  // Function to update the visibility of the geocoder containers based on the
window size
  function updateGeocoderVisibility() {
    const container1 = document.getElementById('search-container');
    const container2 = document.getElementById('search-container2');

    if (window.innerWidth <= 767) {
      document.getElementById("search-container").innerHTML = "";
      document.getElementById("search-container2").innerHTML = "";
      addGeocoderToContainer('search-container'); // Add geocoder to the first
container
    } else {

      document.getElementById("search-container").innerHTML = "";
```

```
            document.getElementById("search-container2").innerHTML = "";
            addGeocoderToContainer('search-container2'); // Add geocoder to the
second container
          }
        }

        // Initial visibility check
        updateGeocoderVisibility();

        // Update where geocoder instance is applied when resizing the window!! Not
necessarily needed to be handled sine the resize listener interferes with android
keyboard.
        //window.addEventListener('resize', updateGeocoderVisibility);

        //
document.getElementById('search-container').appendChild(geocoder.onAdd(map));
        //
document.getElementById('search-container2').appendChild(geocoder.onAdd(map));


        geocoder.on('result', async (event) => {
        // When the geocoder returns a result
        const point = event.result.center; // Capture the result
        const placeName = event.result.text;
        const tileset = 'gavinciii.81ck42dr'; // replace this with the ID of the tileset you
created
        const radius = 1609; // 1609 meters is roughly equal to one mile
        const limit = 10; // The maximum amount of results to return

        const query = await fetch(

`https://api.mapbox.com/v4/${tileset}/tilequery/${point[0]},${point[1]}.json?radius=${radiu
s}&limit=${limit}&access_token=${mapboxgl.accessToken}`,
          { method: 'GET' }
        );
        const json = await query.json();
        // console.log(json);

        addMonitored(map, placeName, point);
        updateNearby(json.features, map);
        });



      })
      .catch(error => console.error("Error loading GeoJSON:", error));

  }
```

```
let selectedLocations = [];
let markers = [];

async function addMonitored(map, placeName, coordinates) {
  if (selectedLocations.some(loc => loc.name === placeName)) return;

  try {
    //console.log("Fetching flood level for coordinates:", coordinates);
    const result = await fetchBand1Value(coordinates[0], coordinates[1]);

    // console.log("Flood Value:", result.value);
    // console.log("Flood Level:", result.level);

    // Push with the retrieved flood level
    selectedLocations.push({
      name: placeName,
      coordinates: coordinates,
      floodLevel: result.level,
      floodValue: +result.value.toFixed(2)   // You can also include result.value if
needed
    });

    // Add marker to the map
    let marker = new mapboxgl.Marker({ color: "blue" })
      .setLngLat(coordinates)
      .addTo(map);

    markers.push(marker); // Store marker for later removal
    updateLocationList(map);


  } catch (error) {
    console.error("Error fetching flood level:", error);
  }
}

async function fetchBand1Value(lng, lat) {
  try {
    const flaskServerUrl = 'http://127.0.0.1:5000';
    const response = await fetch(`${flaskServerUrl}/api/get-band1`, {
      method: "POST",
      headers: { "Content-Type": "application/json" },

      body: JSON.stringify({ lng, lat, layerIndex })
    });

    if (!response.ok) {
      throw new Error("API call failed with status " + response.status);
    }
```

```
        const data = await response.json();
        // console.log("Received data from API:", data);

        if (data.value !== undefined) {
            // console.log("Band1 value:", data.value);
            // console.log("Flood level:", data.flood_level);
            return {
                value: data.value,
                level: data.flood_level,
                row: data.row,
                col: data.col
            };
        } else {
            throw new Error("Invalid data structure: " + JSON.stringify(data));
        }
    } catch (err) {
        console.error("Error in fetchBand1Value:", err);
        throw err;  // Re-throw to be handled by the caller
    }

}

    /**
   * Re-fetches flood levels for all currently monitored locations based on the provided
layer index,
   * updates the selectedLocations array, and refreshes the UI list.
   * @param {number} layerIndex The new layer index (0-11) to fetch data for.
   * @param {mapboxgl.Map} map The Mapbox map instance.
   */
  async function refreshMonitoredLocations(layerIndex, map) {
    console.log(`Refreshing ${selectedLocations.length} selected locations for index
${layerIndex}...`);

    if (selectedLocations.length === 0) {
        console.log("No locations selected, nothing to refresh.");
        // Ensure the list shows the "empty" message if needed
        updateLocationList(map); // Call it even if empty to show the correct message
        return;
    }

    // Set a temporary "Loading..." state in the UI if desired
    // selectedLocations.forEach(loc => loc.floodLevel = "Updating...");
    // updateLocationList(map); // Show "Updating..." immediately

    const updatePromises = selectedLocations.map((location, index) => // Use map to
preserve order if needed later
        fetchBand1Value(location.coordinates[0], location.coordinates[1])
            .then(result => {
```

```
                // --- Directly update the object in the selectedLocations array ---
                location.floodLevel = result.level;
                location.floodValue = +result.value.toFixed(2)   // Update value too if you
store it
                console.log(`Refreshed ${location.name}: Level=${location.floodLevel}`);
                return { status: 'fulfilled', index: index }; // Indicate success
            })
            .catch(error => {
                console.error(`Failed to refresh ${location.name}: ${error.message}`);
                // --- Update the object to show an error state ---
                location.floodLevel = "Error";
                location.value = 0;
                return { status: 'rejected', index: index, reason: error.message }; // Indicate
failure
            })
      );

      // Wait for all fetches to complete or fail
      await Promise.allSettled(updatePromises); // Use allSettled
      console.log("Finished refreshing selected locations batch.");

      // --- Now call your existing function to redraw the list ---
      // updateLocationList will use the updated floodLevel values in the
selectedLocations array
      updateLocationList(map);
   }

   function updateLocationList(map) {
      // Select all elements with the class "monitored-list"
      const monitoredLists = document.querySelectorAll(".monitored-list");

      if (selectedLocations.length === 0) {
         const emptyCardHTML = `
         <div class="card mb-2 rounded-4 card-empty mt-3">
            <div class="card-body d-flex align-items-center p-3 ">
            <div>
               <h6 class="fw-bold text-darkBlue text-white">Nothing to show here
yet...</h6>
               <p class="card-subtitle text-white fw-medium fs-6 fst-italic lh-1">Please add
a monitored location to see a flood alert level.</p>
            </div>
            </div>
         </div>
         `;

         // Set the innerHTML of *each* monitored-list element to the empty message
         monitoredLists.forEach(monitoredList => {
            monitoredList.innerHTML = emptyCardHTML;
         });
```

```
        return; // Exit the function early
    }

    // Clear the inner HTML of each monitored list
    monitoredLists.forEach(monitoredList => {
        monitoredList.innerHTML = ""; // Clear existing content
        window._mapRef = map;


        selectedLocations.forEach((loc, index) => {
            let textColor = loc.floodLevel === "High Flood Level" ? "red" :
                        loc.floodLevel === "Moderate Flood Level" ? "orange" :
                        loc.floodLevel === "Low Flood Level" ? "#F5C548" : "green";

            let card = document.createElement("div");
            card.className = "card my-2 rounded-4 monitored-location";
            // Set the data-index attribute here
            card.setAttribute("data-index", index);
            card.innerHTML = `
                <div class="card-body d-flex justify-content-between align-items-center
p-3">
                    <div>
                        <p class="fw-semibold lh-sm text-darkBlue m-0 text-truncate"
style="max-width: 90%;">${loc.name}</p>
                        <h4 class="fw-bold m-0"
style="color:${textColor};">${loc.floodLevel}</h4>
                        <h6 class="fw-normal fst-italic m-0"
style="color:${textColor};">${loc.floodValue} meters</h6>
                    </div>
                    <button class="btn btn-sm remove-btn"
onclick="removeLocation(${index}, window._mapRef)" style="--bs-btn-padding-y: 0rem;
--bs-btn-padding-x: 0rem;">
                        <img src="assets/remove.svg" alt="Remove" width="30" height="30">
                    </button>
                </div>
            `;

            // ✅ Add event listener directly here
            card.addEventListener("click", (e) => {
                // Prevent the click from triggering when clicking the remove button
                if (e.target.closest('.remove-btn')) return;

                const currentIndex = parseInt(card.getAttribute("data-index"));
                const location = selectedLocations[currentIndex];
                if (location && location.coordinates) {
                    // 🔹 Highlight only the selected card
                    document.querySelectorAll(".monitored-location").forEach(card => {
                        card.classList.remove("selected-location");
                    });
```

```
                card.classList.add("selected-location");

                map.flyTo({
                    center: location.coordinates,
                    zoom: 18,
                    essential: true
                });

                markers.forEach(marker => {
                    marker.remove();

                });
                markers = selectedLocations.map((loc, i) => {
                    const color = (i === currentIndex) ? '#FF6700' : 'blue'; // orange for
selected
                    return new mapboxgl.Marker({ color })
                        .setLngLat(loc.coordinates)
                        .addTo(map);
                });
            }
        });

        monitoredList.appendChild(card);
    });
});

}

// Remove location from the list
function removeLocation(index, map) {
    selectedLocations.splice(index, 1);
    markers[index].remove();
    markers.splice(index, 1); // Remove marker from array
    updateLocationList(map);
}

let currentFacilityPopup = null;

function updateNearby(features, map) {
    // Select all elements with the class "location-list"
    const locationLists = document.querySelectorAll(".location-list");

    // Define the HTML for the "Nothing nearby to show" message
    const emptyNearbyHTML = `
        <div class="card mb-2 rounded-4 card-empty mt-3">
        <div class="card-body d-flex align-items-center p-3 ">
            <div>
            <h6 class="fw-bold text-darkBlue text-white">Nothing nearby to show
yet...</h6>
```

```
            <p class="card-subtitle text-white fw-medium fs-6 fst-italic lh-1">Please
add/select a monitored location to see nearby facilities.</p>
            </div>
          </div>
          </div>
        `;

        if (features.length === 0) {
          // If there are no features, set the innerHTML of each location-list to the empty
message
          locationLists.forEach(locationList => {
          locationList.innerHTML = emptyNearbyHTML;
          });

          if (currentFacilityPopup) {
            currentFacilityPopup.remove();
            currentFacilityPopup = null;
          }
          return; // Exit the function early
        }

        // Clear previous entries and populate each location list
        locationLists.forEach(locationList => {
          locationList.innerHTML = ""; // Clear previous entries

          features.forEach(feature => {
            let coordinates = feature.geometry.coordinates; // Extract coordinates
            let name = feature.properties.name || "Unknown Facility";
            let distance = (feature.properties.tilequery.distance / 1000).toFixed(2); //
Convert meters to km
            let amenity = feature.properties.amenity || "default"; // Get the amenity type

            const iconMap = {
              "police_station": "assets/icons/GovernmentService.svg",
              "fire_station": "assets/icons/GovernmentService.svg",
              "school": "assets/icons/School.svg",
              "hospital": "assets/icons/Hospital.svg",
              "evacuation_center": "assets/icons/Evacuation.svg",
              "miscellaneous": "assets/icons/Misc.svg",
              "government": "assets/icons/Government.svg"
            };
            const iconPath = iconMap[amenity] || "assets/icons/Misc.svg";

            // Create Bootstrap Card
            let card = document.createElement("div");
            card.className = "card mb-2 rounded-4 facility-card"; // Add Bootstrap card
classes
            card.innerHTML = `
              <div class="card-body d-flex align-items-center p-3">
```

```
                <div class="me-3">
                    <img src="${iconPath}" alt="${amenity}" width="30" height="30">
                </div>
                <div>
                    <h6 class="fw-bold text-darkBlue">${name}</h6>
                    <p class="card-subtitle text-lightBlue fw-medium fs-6 fst-italic
lh-sm">${distance} km away</p>
                </div>
            </div>
        `;

        // Add the click event listener to the card
        card.addEventListener('click', async() => {



            map.flyTo({
                center: coordinates, // Use the coordinates from the feature
                zoom: 18,         // Set a desired zoom level (adjust as needed)
                essential: true    // Mark this animation as essential
            });
            // You could also use map.easeTo() or map.jumpTo()
            // map.easeTo({ center: coordinates, zoom: 15 });
            // map.jumpTo({ center: coordinates, zoom: 15 });

            // 2. Remove existing popup, if any
            if (currentFacilityPopup) {
                currentFacilityPopup.remove();
                currentFacilityPopup = null; // Clear the reference
            }

            // Fetch flood level asynchronously
            let floodLevel = "Loading...";
            try {
                const result = await fetchBand1Value(coordinates[0], coordinates[1]);
                floodLevel = result.level || "Unknown";
                floodValue = +result.value.toFixed(2) || "Unknown";
            } catch (err) {
                console.error("Failed to fetch flood level:", err);
                floodLevel = "Unavailable";
            }

            let textColor = floodLevel === "High Flood Level" ? "red" :
                    floodLevel === "Moderate Flood Level" ? "orange" :
                    floodLevel === "Low Flood Level" ? "#F5C548" : "green";

            // 3. Create and display the new popup
            // The default popup style has an anchor pointing down ('bottom'), fulfilling
the "dot below it" idea
```

```
            currentFacilityPopup = new mapboxgl.Popup({
                closeButton: true,
                closeOnClick: false,
                offset: 15
            })
            .setLngLat(coordinates)
            .setHTML(`
                <div>
                    <p class="fw-semibold lh-sm text-darkBlue m-0 text-wrap"
style="max-width: 90%;">${name}</p>
                    <h4 class="fw-bold m-0" style="color:${textColor};">${floodLevel}</h4>
                    <h6 class="fw-normal fst-italic m-0"
style="color:${textColor};">${floodValue} meters</h6>


                </div>
            `)
            .addTo(map);

            // Apply rounded corners to the popup container
            const popupEl = currentFacilityPopup.getElement();
            const contentEl = popupEl.querySelector('.mapboxgl-popup-content');
            if (contentEl) {
                contentEl.style.borderRadius = '12px';
                contentEl.style.boxShadow = '0 4px 12px rgba(0, 0, 0, 0.2)';
            }

            currentFacilityPopup.on('close', () => {
                currentFacilityPopup = null;
            });
        });

        // Append the card to the facility list
        locationList.appendChild(card);
    });
  });
}

/**
 * Fetches a CSV file with date/time info, parses it, and formats rows for a dropdown.
 * Assumes CSV columns: Year, Month, Day, Hour, PredictedFWaterLevel
 * Creates objects { text: "Formatted Date/Time", index: rowNumber }
 * @param {string} csvUrl - The path/URL to the CSV file.
 * @returns {Promise<Array<{text: string, index: number}>>} A promise that resolves
with an array of objects.
 */
async function loadAndFormatCsvForDropdown(csvUrl) {
    // --- THIS FUNCTION REMAINS THE SAME AS THE PREVIOUS ANSWER ---
    // (Includes fetch, parse, format logic)
```

```
        console.log(`Workspaceing CSV data for dropdown from: ${csvUrl}`);
        try {
            const response = await fetch(csvUrl);
            if (!response.ok) { throw new Error(`HTTP error! Status: ${response.status} -
Failed to fetch ${csvUrl}`); }

            // --- Get and Log the Last-Modified Header ---
            const headers = response.headers; // Get the Headers object
            const lastModifiedHeader = headers.get('Last-Modified'); // Get the specific
header value

            if (lastModifiedHeader) {
                console.log(`CSV File Last Modified (from header): ${lastModifiedHeader}`);
                try {
                    // Optionally, convert it to a JavaScript Date object for further use/formatting
                    const lastModifiedDate = new Date(lastModifiedHeader);
                    console.log(`CSV File Last Modified (as Date object):`, lastModifiedDate);

                    // Example: You could display this date somewhere on your page
                    const displayElement = document.getElementById('update');
                    if (displayElement) {
                        displayElement.innerHTML = `<small>Map predictions last updated at
${lastModifiedDate.toLocaleString()}.</small>`; // Format as needed
                    }

                } catch(dateParseError) {
                    console.warn("Could not parse Last-Modified header string into a Date
object:", dateParseError);
                }
            } else {
                // This means the server didn't send the header for this file
                console.log("Server did not send a Last-Modified header for the CSV file.");
            }

            const csvText = await response.text();
            console.log("CSV text fetched successfully.");
            const lines = csvText.trim().split('\n');
            const dropdownItemsData = [];
            const dateTimeFormatter = new Intl.DateTimeFormat('en-US', { year: 'numeric',
month: 'long', day: 'numeric', hour: 'numeric', /*minute: '2-digit',*/ hour12: true });

            for (let i = 1; i < lines.length; i++) {
                const line = lines[i].trim();
                if (line === "") continue;
                const columns = line.split(',');
                if (columns.length >= 4) {
                    const year = parseInt(columns[0].trim(), 10);
                    const month = parseInt(columns[1].trim(), 10);
                    const day = parseInt(columns[2].trim(), 10);
```

```
                const hour = parseInt(columns[3].trim(), 10);
                if (isNaN(year) || isNaN(month) || isNaN(day) || isNaN(hour)) {
console.warn(`Skipping line ${i + 1}: Invalid date/time components. Line: "${line}"`);
continue; }
                try {
                   const dateObject = new Date(year, month - 1, day, hour);
                   let formattedString = dateTimeFormatter.format(dateObject);
                   if (!formattedString.includes(':')) { formattedString =
formattedString.replace(/ (AM|PM)$/, ':00 $1'); } // Add :00 if needed
                   dropdownItemsData.push({ text: formattedString, index: i - 1 });
                } catch (dateError) { console.warn(`Skipping line ${i + 1}: Could not create
Date object. Error: ${dateError.message}. Line: "${line}"`); }
             } else { console.warn(`Skipping line ${i + 1}: Expected at least 4 columns,
found ${columns.length}. Line: "${line}"`); }
          }
          console.log(`Successfully parsed ${dropdownItemsData.length} data rows for
dropdown.`);
          return dropdownItemsData;
      } catch (error) { console.error("Error loading or parsing CSV data for dropdown:",
error); return []; }
      // --- END OF UNCHANGED FUNCTION ---
   }

   // --- How to Use (Modified for Multiple Dropdowns) ---

   const csvFilePath = 'assets/flood_predictions.csv'; // <--- CHANGE TO YOUR CSV
FILE PATH
   const menuSelector = '.tileset-dropdown-menu'; // <--- Use the class selector for the
<ul> elements

   // Get all menu elements that need updating
   const dropdownMenus = document.querySelectorAll(menuSelector);

   if (dropdownMenus.length === 0) {
      console.error(`No dropdown menus found using selector "${menuSelector}"!`);
   } else {
      console.log(`Found ${dropdownMenus.length} dropdown menus to populate.`);
      loadAndFormatCsvForDropdown(csvFilePath)
         .then(itemsData => {
            if (itemsData && itemsData.length > 0) {
                console.log("Formatted Dropdown Items Data:", itemsData);

                // --- Loop through each dropdown menu found ---
                dropdownMenus.forEach((menuElement, menuIndex) => {
                   console.log(`Populating menu #${menuIndex + 1}`);

                   // --- Try to find the corresponding button for this menu ---
                   // Assumes button is the direct previous sibling or within the same parent
.dropdown div
```

```
                let buttonElement = menuElement.previousElementSibling;
                // If button is not direct sibling, try finding within parent .dropdown
                if (!buttonElement || !buttonElement.classList.contains('dropdown-toggle'))
{
                const parentDropdown = menuElement.closest('.dropdown');
                if(parentDropdown) {
                      buttonElement =
parentDropdown.querySelector('.dropdown-toggle');
                }
                }

                if (!buttonElement) {
                   console.warn(`Could not find corresponding button for menu
#${menuIndex + 1}`);
                }
                // -----------------------------------------------------------

                // Clear existing dynamic options from *this specific menu*
                menuElement.innerHTML = '<li><h6 class="dropdown-header">Show
predictions for...</h6></li>'; // Reset with header


                // Populate *this specific menu*
                itemsData.forEach(item => {
                   const listItem = document.createElement('li');
                   const link = document.createElement('a');
                   link.classList.add('dropdown-item', 'text-darkBlue'); // Add your classes
                   link.href = '#';
                   link.textContent = item.text;
                   // IMPORTANT: Store the index associated with this row/layer
                   link.dataset.tilesetIndex = item.index; // e.g., data-tileset-index="0"

                   listItem.appendChild(link);
                   menuElement.appendChild(listItem);
                });

                // Optional: Set the corresponding button text to the first item initially
                if (buttonElement) {
                   buttonElement.textContent = itemsData[0].text;
                }

             }); // --- End loop through each menu ---

             console.log("All dropdowns populated dynamically.");

             // REMINDER: Ensure your click event listener(s) for the dropdown items
             // are set up correctly to handle clicks from items in *either* dropdown menu.
             // Using event delegation on a parent element or querySelectorAll might be
needed.
```

```
            } else {
                console.log("No dropdown options generated from CSV.");
                // Update all buttons if no data
                dropdownMenus.forEach((menuElement, menuIndex) => {
                    let buttonElement = menuElement.previousElementSibling;
                    if (!buttonElement || !buttonElement.classList.contains('dropdown-toggle'))
{
                        const parentDropdown = menuElement.closest('.dropdown');
                        if(parentDropdown) { buttonElement =
parentDropdown.querySelector('.dropdown-toggle'); }
                    }
                    if (buttonElement) {
                        buttonElement.textContent = "No predictions";
                    }
                });
            }
        })
        .catch(error => {
            console.error("An unexpected error occurred during dropdown population:",
error);
        });
}

function loadTop5FromCSV(selectedIndex) {
    const filePath = `assets/reports/top5_${selectedIndex}.csv`;

    fetch(filePath)
        .then(response => {
            if (!response.ok) throw new Error("File not found");
            return response.text();
        })
        .then(csvText => {
            const lines = csvText.trim().split("\n");
            const headers = lines[0].split(",").map(h => h.trim());

            const data = lines.slice(1).map(line => {
                const values = line.split(",");
                const obj = {};

                headers.forEach((header, i) => {
                    // Use empty string if value is missing
                    obj[header] = (values[i] || "").trim();
                });

                return obj;
            });

            displayTop5Barangays(data);
```

```
        })
        .catch(err => {
            console.error("Error loading CSV:", err);
            document.getElementById("top5-barangays").innerHTML = "Could not load
data.";
        });
  }

  function displayTop5Barangays(data) {
      const container = document.getElementById("toast-container");
      container.innerHTML = ""; // Clear previous toasts

      const listItems = data.map((item, index) => {
          const keys = Object.keys(item);
          const barangay = item.Barangay || "Unknown";
          const floodValue = item[keys[1]] || "N/A";

          return `<div class="fs-6"><strong>#${index + 1}</strong> ${barangay} — <span
class="fw-bold">${floodValue} m </div>`;
      }).join("");

      const toast = document.createElement("div");
      toast.className = "toast text-darkBlue bg-white border-0 rounded-3 show";
      toast.setAttribute("role", "alert");
      toast.setAttribute("aria-live", "polite");
      toast.setAttribute("aria-atomic", "true");

      toast.innerHTML = `
          <div class="toast-header">
              <strong class="me-auto fs-6 text-darkBlue">Barangays at most risk</strong>
              <button type="button" class="btn-close" data-bs-dismiss="toast"
aria-label="Close"></button>
          </div>
          <div class="toast-body">
              <div class="mb-0 mt-2">${listItems}</div>
          </div>
      `;

      container.appendChild(toast);
      new bootstrap.Toast(toast, { autohide: false }).show();
```

Documentation

Consultation with CADRRESMO; Site Visits



Visit to Municipal Planning Office for Gathering Data

User Testing in Camaligan

User testing with CBSUA / DOST-PAGASA'S BRFFWC

Alexandra Nicole Eclarinal

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

Phone
+63 948 404 6495

Email
alexandranicoleeclarinal@gmail.com

Address
Zone 3, La Purisima, Iriga City, Camarines Sur, 4431

Climate activist, feminist, and human rights advocate with three years of experience in the development sector, working with various organizations and diverse communities across the Philippines. Currently serving as one of the lead organizers of a national youth movement advancing climate justice through awareness campaigns, grassroots initiatives, and youth empowerment. Actively engaged in addressing the intersectionality of climate change, gender equality, and human rights, with a strong focus on inclusive, community-based approaches to sustainable development.

EDUCATION

Zeferino Arroyo High School (2015 - 2019)
Science, Technology, and Engineering (STE) Program
With High Honors

ACLC College of Iriga, Inc. (2019 - 2021)
TVL-CSS
With Highest Honors

Bicol University - College of Science (2021 - 2025)
Bachelor of Science in Computer Science
Dean's Lister, Service Award

CERTIFICATIONS & AWARDS

Contact Center Services NCII
2019 | Passer, Top 1 Award

Computer Systems Servicing NCII
2021 | Passer, Top 1 Award

Youth Advocate in Climate
2022 | Greenpeace Philippines

Kabataang Resilient Awardee
2023 | UNICEF Philippines

Young Women in Climate
2024 | Greenpeace Philippines

Dean's Lister
2021 - 2025 | Bicol University College of Science Academic Award

CONFERENCES & EVENTS

Asia-Pacific Ministerial Conference on Disaster Risk Reduction
2025 | Facilitator

ASEAN Youth Climate Action and Disaster Resilience Conference
2025 | Facilitator

SKILLS

- Climate justice advocacy
- Gender and human rights mainstreaming
- Community organizing and youth engagement
- Program development and coordination
- Workshop facilitation and public speaking
- Research and policy analysis
- Strategic communication and messaging
- Social media management and digital campaigning
- Content creation (copywriting, storytelling, multimedia)
- Public relations and stakeholder engagement
- Video editing and production (e.g., Premiere Pro, CapCut)
- Journalism and media writing
- Graphic design (Adobe Photoshop, Illustrator, Canva)

EXPERIENCE

External Affairs Executive & Lead Campaigner
2022 - Present | Angat Generation Climate
- Led national youth-led campaigns advocating for climate justice, focusing on grassroots mobilization and awareness-raising.
- Built partnerships with local and international organizations to amplify climate-related campaigns.
- Organized forums, workshops, and digital campaigns to educate communities on climate justice.

Communications Team Member
2024 - Present | Amnesty International
- Contributed to a community of practice focused on human rights communication and advocacy.
- Developed educational materials and social media campaigns to raise awareness of human rights issues.
- Provided strategic insights to improve outreach and engagement in human rights movements.

Technical Working Group & Founding Member
2024 - Present | Kabataang Resilient Network
- Developed advocacy and communication initiatives aimed at children and youth's involvement in climate action and disaster resilience.
- Implemented strategic campaigns addressing disaster risk reduction and climate resilience.
- Engaged with government agencies and civil society organizations to push for inclusive policies.

Volunteer
2024 - Present | Greenpeace Philippines
- Facilitated community discussions and public engagements to promote environmental activism.
- Supported communications initiatives by drafting content, creating visual materials, and managing digital outreach efforts.
- Carried out national and local campaigns and assisted in event organization, training workshops, and campaign planning.

Broadcaster & Writer
2022 - 2025 | BU Scientia
- Wrote, edited, and reported news stories covering campus events.
- Conducted interviews and field reports to ensure accurate and relevant storytelling.
- Produced and delivered news broadcasts for student audiences.

Renewable Energy Ambassador
2024 - 2025 | REBOOT Philippines
- Advocated for sustainable energy solutions in marginalized communities through awareness campaigns.
- Collaborated with local government and grassroots organizations to promote renewable energy initiatives.
- Participated in training programs and community immersions bridging the gap between renewable energy and gender equality.

Field Researcher
2024 - 2025 | WR Numero Research
- Identified and recruited a diverse and representative sample of participants for three focus group discussions within Bicol region.
- Facilitated and led FGDs, ensured productive and respectful engagement among participants.
- Created topline insights from the FGDs, summarized key themes, insights, and emerging patterns from the discussions.

Yna Gabrielle Foronda

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

Phone
+63 962 414 4064

Email
foronda.ynagabrielle@gmail.com

Address
016 Canuto St., Brgy. San Isidro, Pili 4418, Camarines Sur, PHILIPPINES

A graduating computer science student that ventures out of her comfort zone by exploring possible applications and solutions that technology can provide. Familiar in Python and R, with hands-on experience in data manipulation, visualization, and machine learning. Also experienced in graphic design and front-end development using HTML, CSS, and JavaScript.

EDUCATION

Naga City Science High School (2015-2021)
Science, Technology, Engineering, and Mathematics (STEM)
With Honors

Bicol University – College of Science (2021 – Present)
Bachelor of Science in Computer Science
Dean's and President's Lister, Scholarship Award, Service Award, Special Citation

CERTIFICATIONS & AWARDS

Foundations of User Experience (UX) Design
2023 | Part of Google UX Design Course

Crash Course on Python
2023 | Part of Google IT Automation with Python Course

Using Python to Interact with the Operating System
2023 | Part of Google IT Automation with Python Course

Introduction to Git and GitHub
2023 | Part of Google IT Automation with Python Course

President's and Dean's Lister
2021 - 2024 | Academic Award

Scholarship Award
2024 | Academic Award

SKILLS

- Familiar with programming languages (Python, R, C, Javascript)
- Familiar with HTML and CSS
- Proficient with design tools (Photoshop, Figma, Clip Studio Paint, Canva)
- Proficient in project management tools (Notion, Slack, Trello)
- Quick to adapt and open-minded
- Strong communication skills
- Strong time-management skills
- Strong organization and management skills

CONFERENCES & EVENTS

VIVES University of Applied Science Summer School in Industrial AI
2024 | Student Delegate

A dynamic two-week program that took place from May 27 to June 7, 2024, at the VIVES Bruges Xaverianenstraat Campus in Belgium. This comprehensive program covered various topics, including AI Fundamentals, Machine Learning, Data Science, AI Industry, Deep Learning, AI EDGE, AI Model Deployment, and AI Ethics. Aimed at equipping participants with the latest trends and skills in industrial AI, the program also featured company visits to provide insights into Bruges' local industry and culture.

EXPERIENCE

Graphic Artist / Illustrator
2019 - Present | Freelance

- Produced engaging social media assets for a boutique spa, enhancing brand visibility and client engagement.
- Developed an educational comic strip utilized within a local high school curriculum.
- Designed the logo for DOST's Project SET, contributing to its impactful branding and recognition.
- Actively engaged in freelance art commissions, consistently delivering quality creative solutions to diverse clientele.

Graphic Artist
2022 - 2023 | Google Developers Student Club - Bicol University

Designed publication materials and its official mascot that captures the organization's brand identity and recognition.

Creatives Junior Councilor
2023 | Academic Consortium of Computer Science Students

- Designed publication materials and assets for the organization's Facebook page and events.
- Helped with organizing the department-wide Computer Programming Competition

Kadet Engagement Mission Specialist (Intern)
2023 | KadaKareer

- Started the development of the organization's Discord server bot.
- Helped with various organization events, including career coaching sessions, ensuring seamless execution.
- Conducted user interviews to garner feedback for the organization's own community platform
- Analyzed engagement metrics across the organization's Discord and Facebook group, providing actionable insights to optimize community interaction and growth.
- Implemented strategies to boost member engagement on Discord and Facebook platforms.

Model Developer (Student)
2023-2024| Bicol University (BU)

- Trained an Object Detection Model that identifies signs of rotting in lettuce varieties such as discoloration, and wilting leaves for prediction of its remaining shelf-life.

Front-End Developer (Intern)
2024 | BU Information and Communications Technology Office (BU - ICTO)

- Produced assets for the Bicol University website and Bids and Awards Committee new system
- Developed the front-end side of the new Bids and Awards Committee website and admin panel that will facilitate the online posting of biddings, provide easy access to bid results, and enhance transparency.

Basic Digital Phenotyping (Intern)
2024 | International Rice Research Institute (IRRI)

- Learned about digital imaging and machine learning tools like Ilastik and ImageBreed, using rice as the model crop.
- Honed skills on High-Throughput Phenotyping and R for data analysis and visualization
- Trained an instance segmentation model that can segment wheat diseases in pictures while comparing the performance of developing the model from Yolo v8 and Yolo v11 models, for her special project.

Francis Maurice Miranda

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

Phone
+63 919 964 7029

Email
francismauricemiranda@gmail.com

Address
2059 - A Locsin Subdivision, Maroroy, Daraga, Albay

An aspiring software developer and game developer with over 8 years worth of programming experience. Has basic understanding and fundamentals of C++, C, Python, Java, HTML, Javascript, Kotlin and SQL. Also has experience in graphic design, networking, unity, machine learning and app development.

EDUCATION

Philippine Science High School - Bicol Region Campus (2015 - 2021)
Chemistry Track
Director's Lister

Bicol University - College of Science (2021 - Present)
Bachelor of Science in Computer Science
Dean's and President's Lister, Scholarship Award, Service Award

CERTIFICATIONS & AWARDS

National Olympiad Informatics
2017 | Qualifier Award

GDSC Loyola HackFest - Gen Z
2022 | Top 10 Qualifier Award

Ateneo MISA IMSummit
2022 | Top 5 Qualifier Award

Introduction to Git and GitHub
2023 | Part of Google IT Automation with Python Course

President's and Dean's Lister
2021 - 2024 | Academic Award

Scholarship Award
2022 | Academic Award

SKILLS

- Creative thinker
- A good team player
- Good communication
- Adaptable to any situation
- Flexible in any role
- Capable of being a leader
- Familiar with various programming languages (Java, C, C++, Javascript, etc.)
- Has prior experience in 3D modeling and asset creation
- Back-end focused expertise
- Strong communication skills
- Strong time-management skills
- Strong organization and management skills

CONFERENCES & EVENTS

Bicol Blockchain Conference
2022 | Student Volunteer/Attendee

A two-day event of one of the biggest gathering of crypto and blockchain enthusiasts where they organize workshops, talks, and conferences regarding crypto technologies.

De La Salle Computer Science Workshop
2018 | Student Intern

A week-long event held in De La Salle University Manila were interns from various high schools gather and participate in a workshop that teaches Jupyter, Python, Cybersecurity, and Game Development.

EXPERIENCE

Workshop Organizer
2017 / Bicol University Affiliate

- Organized a Scratch Workshop for different elementary schools in Ligao, Albay
- Taught over three schools under the affiliation of Bicol University

Programming Competitor
2017 - 2018 / Philippine Science High School - BRC

Participated in coding competitions such as NOI and APC.

Mobile Developer (Student)
2020 - present | Philippine Science High School - BRC, Bicol University

- Created a mobile application for organizing doctor's appointments with Dr. Swipe.
- Created a mobile application for a game project called OttO.
- Created an image processing mobile application for detecting lettuce spoilages with CropSPT.

Creatives Junior Councilor
2021 | Academic Consortium of Computer Science Students

Designed various publication materials and certificates for the organization.

External Vice-President
2022 - 2023 | BU Academic Consortium of Computer Science Students

- Held office for two consecutive years.
- Held partnerships with different computer science organizations such as FEU ACM, Ateneo MISA, UP Cursor, etc.
- Initiated year-long partnerships with design companies.
- Organized college events such as symposiums and tech showcases.

Game Developer (Asset Creator)
2024 | Lamina Studios

- Produced game file assets for Lamina Studios.
- Developed multiple pixel art animations, tile sets, and portraits.
- Collaborated in game development for a game called Tempus Paragon.

Model Developer (Student)
2023 - present | Bicol University

- Created a model for a school project involving Image Detection and Processing using TensorFlow and Keras.
- Created a model for flood prediction using NARX and Python.

Technical Junior Councilor
2024 | Bicol University Integrated League of DOST Scholars

Participated as a technical member for DOST BUILDS.