DengueWatch: A System for Real-Time Dengue Monitoring and Forecasting in Iloilo Province

A Special Problem

Presented to

the Faculty of the Division of Physical Sciences and Mathematics

College of Arts and Sciences

University of the Philippines Visayas

Miag-ao, Iloilo

In Partial Fulfillment

of the Requirements for the Degree of

Bachelor of Science in Computer Science by

AMODIA, Kurt Matthew A.

BULAONG, Glen Andrew C.

ELIPAN, Carl Benedict L.

Francis DIMZON, Ph.D.

Adviser

June 2025

Approval Sheet

The Division of Physical Sciences and Mathematics, College of Arts and Sciences, University of the Philippines Visayas

certifies that this is the approved version of the following special problem:

DengueWatch: A System for Real-Time Dengue Monitoring and Forecasting in Iloilo Province

Approved by:

Name	**Signature**	**Date**
Francis D. Dimzon, Ph.D. (Adviser)		
Ara Abigail E. Ambita (Panel Member)		
Kent Christian A. Castor (Division Chair)		

Division of Physical Sciences and Mathematics
College of Arts and Sciences
University of the Philippines Visayas

Declaration

We, Kurt Matthew A. Amodia, Glen Andrew C. Bulaong, and Carl Benedict L. Elipan, hereby certify that this Special Problem has been written by us and is the record of work carried out by us. Any significant borrowings have been properly acknowledged and referred.

Name	Signature	Date
Kurt Matthew A. Amodia (Student)		June 2, 2025
Glen Andrew C. Bulaong (Student)		June 2, 2025
Carl Benedict L. Elipan (Student)		June 2, 2025

Dedication

We dedicate this special problem to all the teachers who have guided us throughout our academic journey. Your knowledge and mentorship have laid the foundation for this research, and for that, we are truly grateful.

To our families, friends, and classmates, thank you for your unwavering support, encouragement, and belief in us. Your presence has been a constant source of strength.

Most especially, we dedicate this work to the health offices and frontline personnel who continue to battle dengue cases with courage and dedication. Your tireless efforts and sacrifices are an inspiration. We hope that this research, in its own small way, can contribute to your work and make a meaningful difference in your fight against this disease.

Acknowledgment

This research would not have been possible without the support and guidance of several individuals and institutions.

First and foremost, we express our deepest gratitude to our adviser, Dr. Francis Dimzon, for his invaluable insights, unwavering support, and commitment to our project. His guidance was instrumental in shaping the direction of our research.

We also extend our sincere thanks to the Iloilo Provincial Health Office and the Iloilo Epidemiological Unit, for accomodating our inquiries and sharing vital data and insights. Their cooperation played a crucial role in the success of the research.

Our appreciation also goes to the UPV Health Services Unit, especially the doctors and nurses who participated in our user testing. Your thoughtful feedback provided essential perspectives that greatly contributed to the relevance of our research.

Finally, we are greatly thankful to God Almighty, for granting us the strength, perseverance, and determination to complete this research.

Abstract

Dengue fever remains a significant public health concern in the Philippines, with cases rising dramatically in recent years. Iloilo City experienced a surge in cases, with 4,585 reported cases and 10 deaths as of August 10, 2023, a 319% increase from the previous year's 1,095 cases and one death. This rise overwhelmed local healthcare facilities, with over 76% of non-COVID-19 hospital beds occupied by dengue patients. The lack of a reliable monitoring and forecasting system delayed interventions, worsening the public health burden. To address this, the study developed a centralized system to modernize data management and monitoring of dengue cases in public health institutions. Using data from the Iloilo Provincial Health Office and online sources, several deep learning models were trained to forecast dengue cases on weather variables and historical data. Models tested included LSTM, ARIMA, Seasonal ARIMA, Kalman Filter (KF), and a hybrid KF-LSTM, evaluated with time series cross-validation and error metrics like MSE, RMSE, and MAE. The LSTM model performed best, achieving the lowest RMSE of 20.15, followed by the hybrid KF-LSTM with 25.56. The LSTM model was integrated into the system, providing forecasting capabilities to support proactive interventions and better resource planning in health institutions.

Keywords: Dengue, Deep Learning, Monitoring, Prediction

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview of the Current State of Technology

Dengue cases surged globally in 2023 and continued to rise in 2025, with over five million cases and more than 5,000 deaths across 80 countries (Bosano, 2023). The World Health Organization reported a ten-fold increase in cases from 2000 to 2019, peaking in 2019 with the disease affecting 129 countries (WHO, 2024). In the Philippines, dengue remains endemic, leading to prolonged and widespread outbreaks.

In Iloilo, dengue cases escalated dramatically. By August 2023, the provincial health office reported 4,585 cases and 10 deaths, marking a 319% increase from the previous year (Perla, 2024). Iloilo has reached the outbreak threshold, and local authorities are on blue alert. In Iloilo City, 649 cases were recorded in 2024, with two deaths. Hospital capacity is strained, with non-COVID-19 hospital bed occupancy exceeding 76%. This highlights the increasing pressure on healthcare resources in the region.

In recent years, technology has played a growing role in improving disease surveillance across the globe. Internationally, a study published in Frontiers in Physics utilized the Kalman filter to predict COVID-19 deaths in Cear´ a, Brazil(Ahmadini et al., 2021). A study also suggests that weather-based forecasting models using variables like mean temperature and cumulative rainfall can provide early warnings of dengue outbreaks with high sensitivity and specificity, enabling predictions up to 16 weeks in advance (Hii, Zhu, Ng, Ng, & Rocklöv, 2012). Locally, in Davao City, Ligue (2022) found that deep learning models can accurately predict dengue outbreaks by capturing complex, time-dependent patterns in environmental data. The study of Carvajal et. al. uses machine learning methods to reveal the temporal pattern of dengue cases in Metropolitan Manila and emphasizes the significance of relative humidity as a key meteorological factor, alongside rainfall and temperature, in influencing this pattern (Carvajal et al., 2018).

Most studies remain theoretical or academic, with limited translation into practical tools that communities and local health authorities can use for early warning and response. An example of such application is RabDash, developed by the University of the Philippines Mindanao. RabdashDC (2024) is a web-based dashboard for rabies data analytics. However, while RabDash demonstrates the potential of applying advanced analytics in public health, similar systems are lacking in the context of dengue.

1.2 Problem Statement

Dengue remains a critical public health challenge worldwide, with cases increasing due to the easing of COVID-19 restrictions and heightened global mobility. While a temporary decline in cases was observed during the pandemic (2020–2022) due to reduced surveillance efforts, 2023 marked a resurgence, with over five million cases and more than 5,000 deaths reported across 80 countries (Bosano, 2023). In Iloilo City and Province, dengue cases rose by 319% as of August 2023, overwhelming local healthcare systems. This surge strained resources, with over 76% of non-COVID-19 hospital beds occupied by dengue patients (Perla, 2024), highlighting the urgent need for effective monitoring and predictive tools. Despite all these studies, there remains a significant gap in the development of publicly accessible systems that apply these predictive models in real-world settings. Most existing studies remain confined to academic or theoretical contexts, with little translation into practical tools for local communities and public health authorities. In particular, there is a lack of research focused specifically on dengue prediction and surveillance in Iloilo. While deep learning models have shown high accuracy in other regions, their application in the local context of Iloilo is minimal. The lack of a reliable system to monitor and forecast dengue outbreaks contributes to delayed interventions, exacerbating public health risks and healthcare burdens in the region.

1.3 Research Objectives

1.3.1 General Objective

This study aims to develop a centralized monitoring and analytics system for dengue cases in Iloilo City and Province with data management and forecasting capabilities. The researchers will train and compare multiple deep learning models to predict dengue case trends based on climate data and historical dengue cases to help public health officials in dengue surveillance.

1.3.2 Specific Objectives

Specifically, this study aims to:

1. gather dengue data from the Iloilo Provincial Health Office and climate data (including temperature, rainfall, wind, and humidity) from online sources, and combine and aggregate these into a unified dataset to facilitate comprehensive dengue case forecasting;

2. train and evaluate deep learning models for predicting dengue cases using metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Squared Error (MSE), and determine the most accurate forecasting approach; and

3. develop a web-based analytics dashboard that integrates the predictive model, provides a data management system for dengue cases in Iloilo City and the

Province, and assess its usability and effectiveness through structured feedback from health professionals and policymakers.

1.4 Scope and Limitations of the Research

This study aims to gather dengue data from the Iloilo Provincial Health Office and climate data from online sources such as PAGASA or weatherandclimate.com. These data will be preprocessed, cleaned, and combined into a unified dataset to facilitate comprehensive dengue case forecasting. However, the study is limited by the availability and completeness of historical data. Inconsistent or missing data points may introduce biases and reduce the quality of predictions. Furthermore, the granularity of the data will be in a weekly format.

To evaluate deep learning models for predicting dengue cases, the study will train and compare the performance of various models, using metrics like Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). While these models aim to provide accurate forecasts, their performance is heavily influenced by the quality and size of the dataset. Limited or low-quality data may lead to suboptimal predictions. Additionally, the models cannot fully account for external factors such as public health interventions or socio-economic conditions which may impact dengue transmission dynamics.

The study also involves developing a web-based analytics dashboard that integrates predictive models and provides a data management system for dengue cases in Iloilo City and the Province. This dashboard will offer public health officials an interactive interface to visualize dengue trends, input new data, and identify risk areas. However, its usability depends on feedback from stakeholders, which may vary based on their familiarity with analytics tools. Moreover, external factors such as limited internet connectivity or device availability in remote areas may affect the system's adoption and effectiveness. While the dashboard provides valuable insights, it cannot incorporate all factors influencing dengue transmission, emphasizing the need for ongoing validation and refinement.

1.5 Significance of the Research

This study's development of an AI-based dengue forecasting and monitoring system has wide-reaching significance for various stakeholders in Iloilo City:

- Public Health Agencies: Organizations like the Department of Health (DOH) and local health units in Iloilo City and Province stand to benefit greatly from the system. With dengue predictions, we can help these agencies optimize their response strategies and implement targeted prevention measures in high-risk areas before cases escalate.

- Local Government Units (LGUs): LGUs can use the system to support their disaster management and health initiatives by proactively addressing dengue outbreaks. The predictive insights allow for more efficient planning and resource deployment in barangays and communities most vulnerable to outbreaks, improving overall public health outcomes.

- Healthcare Facilities: Hospitals and clinics, which currently face high bed occupancy rates during dengue season will benefit from early outbreak forecasts that can help in managing patient inflow and ensuring adequate hospital capacity.

- Researchers and Policymakers: This AI-driven approach contributes valuable insights for researchers studying infectious disease patterns and policymakers focused on strengthening the national AI Roadmap. The system's data can support broader initiatives for sustainable health infrastructure and inform policy decisions on resource allocation for dengue control.

- Community Members: By reducing the frequency and severity of outbreaks, this study ultimately benefits the community at large. This allows for timely awareness campaigns and community engagement initiatives, empowering residents with knowledge and preventative measures to protect themselves and reduce the spread of dengue.

Chapter 2

Review of Related Literature

2.1 Dengue

Dengue disease is a tropical and subtropical mosquito-borne viral illness and is a major health concern in the Philippines (Bravo, Roque, Brett, Dizon, & L'Azou, 2014). The majority of individuals with dengue experience no symptoms. Fever is the most common symptom, typically 4 to 7 days after being bitten by an infected mosquito (Zhou & Malani, 2024). In recent years, the trend of dengue cases in the Philippines has shown notable fluctuations, with periodic outbreaks occurring every 3 to 5 years, often influenced by climatic and environmental changes. According to the Department of Health (DOH), the number of reported cases has steadily increased over the past decades, attributed to urbanization, population growth, and inadequate vector control measures (World Health Organization (WHO), 2018). Moreover, studies suggest that El Niño and La Niña events have significant effects on dengue incidence, with warmer temperatures and increased rainfall providing favorable breeding conditions for mosquitoes (Watts, David M and Burke, Donald S and Harrison, Bruce A and Whitmire, Ralph E and Nisalak, Ananda, 2020). The study of Carvajal et. al. highlights the temporal pattern of dengue cases in Metropolitan Manila and emphasizes the significance of relative humidity as a key meteorological factor, alongside rainfall and temperature, in influencing this pattern (Carvajal et al., 2018).

2.2 Outbreak Definition

The definition of an outbreak is a critical factor in disease surveillance, as it determines the threshold at which an unusual increase in cases is considered a public health concern. Studies suggest that outbreak thresholds should be context-specific, given the variability in transmission dynamics across different locations (Runge-Ranzinger et al., 2016). Outbreak definitions defined using the Endemic Channel often base thresholds on 2 standard deviations (SD) above the mean number of historic dengue cases. Other studies (Hemisphere, 2015) also used an alert threshold of a 5-year mean plus 2 standard deviations. A study by (Brady, Smith, Scott, & Hay, 2015) discussed that definitions of dengue outbreaks differ significantly across regions and time, making them inconsistent and incomparable.

2.3 Existing System: RabDash DC

RabDash, developed by the University of the Philippines Mindanao, is a web-based dashboard for rabies data analytics. It combines predictive modeling with genomic data, enabling local health authorities to optimize interventions and allocate resources more effectively. RabDash's modules include trend visualization, geographic hotspot mapping, and predictive forecasting, utilizing Long Short-Term Memory (LSTM) models for time-series forecasting (RabDashDC, 2024).

For DengueWatch, RabDash serves as a strong inspiration, particularly in its monitoring, historical trend visualization, and forecasting capabilities. These features align well with the needs of dengue control efforts, providing real-time insights into outbreak trends and enabling more effective, data-driven decision-making. RabDash's architecture is relevant to the DengueDash, as dengue outbreaks similarly require time-series forecasting models. By using LSTM, RabDash effectively models trends in outbreak data, which provides a framework for adapting LSTM to dengue forecasting. Research indicates that LSTM models outperform traditional methods, such as ARIMA and MLP, in handling the complexities of time-dependent epidemiological data (Ligue & Ligue, 2022).

2.4 Deep Learning

The study of (Ligue & Ligue, 2022) highlights how data-driven models can help predict dengue outbreaks. The authors compared traditional statistical methods, such as non-seasonal and seasonal autoregressive integrated moving average (ARIMA), and traditional feed-forward network approach using a multilayer perceptron (MLP) model with a deep learning approach using the long short-term memory (LSTM) architecture in their prediction model. They found that the LSTM model performs better in terms of accuracy. The LSTM model achieved a much lower root mean square error (RMSE) compared to both MLP and ARIMA models, proving its ability to capture complex patterns in time-series data (Ligue & Ligue, 2022). This superior performance is attributed to LSTM's capacity to capture complex, time-dependent relationships within the data, such as those between temperature, rainfall, humidity, and mosquito populations, all of which contribute to dengue incidence (Ligue & Ligue, 2022).

2.5 Kalman Filter

The Kalman Filter is another powerful tool for time-series forecasting that can be integrated into our analysis. It provides a recursive solution to estimating the state of a linear dynamic system from a series of noisy measurements. Its application in epidemiological modeling can enhance prediction accuracy by accounting for uncertainties in the data(Li et al., 2022). Studies have shown that Kalman filters are effective in predicting infectious disease outbreaks by refining estimates based on observed data. A study published in Frontiers in Physics utilized the Kalman filter to predict COVID-19 deaths in Ceará, Brazil. They found that the Kalman filter effectively tracked the progression of deaths and cases, providing critical insights for public health decision-making (Ahmadini et al., 2021). Another research article in PLOS ONE focused on tracking the effective reproduction number (R_t) of COVID-19 using a Kalman filter. This method estimated the growth rate of new infections from noisy data, demonstrating that the Kalman filter could maintain accurate estimates even when case reporting was inconsistent(Arroyo-Marioli, Bullano, Kucinskas, & Rondón-Moreno, 2021).

This study compares ARIMA, seasonal ARIMA, Kalman Filter, and LSTM models using collected dengue case data along with weather data to identify the most effective model for real-time forecasting.

2.6 Weather Data

The relationship between weather patterns and mosquito-borne diseases is inherently nonlinear, meaning that fluctuations in disease cases do not respond proportionally to changes in climate variables(Colón-González, Fezzi, Lake, & Hunter, 2013) Weather data, such as minimum temperature and accumulated rainfall, are strongly linked to dengue case fluctuations, with effects observed after several weeks due to mosquito breeding and virus incubation cycles. Integrating these lagged weather effects into predictive models can improve early warning systems for dengue control(Cheong, Burkart, Leitão, & Lakes, 2013). A study also suggests that weather-based forecasting models using variables like mean temperature and cumulative rainfall can provide early warnings of dengue outbreaks with high sensitivity and specificity, enabling predictions up to 16 weeks in advance(Hii et al., 2012).

This study utilizes weather data, including variables such as temperature, rainfall, and humidity, as inputs for our dengue forecasting model. Given the strong, nonlinear relationship between climate patterns and dengue incidence, these weather variables, along with their lagged effects, are essential for enhancing prediction accuracy and providing timely early warnings for dengue outbreaks.

2.7 Chapter Summary

This chapter reviewed key literature relevant to our study, focusing on existing systems, predictive modeling techniques and the role of weather data in forecasting dengue outbreaks. We examined systems like RabDash DC, which integrates predictive modeling with real-time data to inform public health decisions, providing a foundational structure for our Dengue Watch System. Additionally, deep learning approaches, particularly Long Short-Term Memory (LSTM) networks, were highlighted for their effectiveness in time-series forecasting, while alternative methods such as ARIMA and Kalman Filters were considered for their ability to model complex temporal patterns and handle noisy data.

The literature further underscores the significance of weather variables—such as temperature and rainfall—in forecasting dengue cases. Studies demonstrate that these variables contribute to accurate outbreak prediction models. Leveraging these insights, our study will incorporate both weather data and historical dengue case counts to build a reliable forecasting model.

Chapter 3

Research Methodology

This chapter lists and discusses the specific steps and activities that were performed to accomplish the project. The discussion covers the activities from pre-proposal to Final SP Writing.

Figure 3.1 summarizes the workflow for forecasting the number of weekly dengue cases. This workflow focuses on using statistical, deep learning, and probabilistic models to forecast the number of reported dengue cases. The approach involves deploying several models for prediction, including ARIMA and Seasonal ARIMA as statistical approaches, LSTM as a deep learning approach, and the Kalman Filter as a probabilistic approach. These methods are compared with each other to determine the most accurate model.

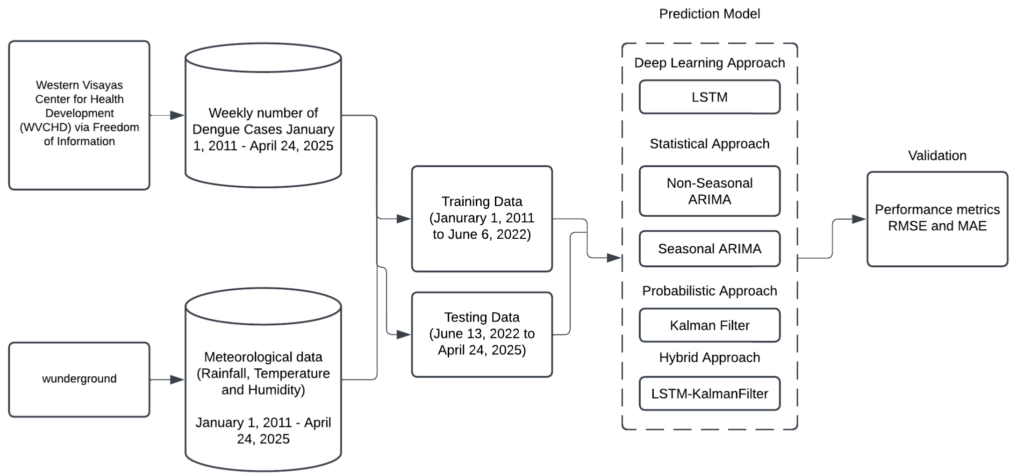


Figure 3.1: Workflow for forecasting the number of weekly dengue cases

3.1 Research Activities

3.1.1 Dengue and Climate Data Collection

Acquisition of Dengue Case Data

The historical dengue case dataset used in this study was obtained from the Humanitarian Data Exchange and the Western Visayas Center for Health Development (WVCHD) via Freedom of Information (FOI) requests. The decision to use weekly intervals was driven by the need for precision and timeliness in capturing fluctuations in dengue cases and weather conditions. Dengue transmission is influenced by short-term changes in weather variables such as rainfall and temperature, which impact mosquito breeding and virus transmission cycles. A weekly granularity allowed the model to better capture these short-term trends, enabling more accurate predictions and responsive public health interventions.

Moreover, using a weekly interval provided more data points for training the models compared to a monthly format. This is particularly critical in time series modeling, where larger datasets help improve the robustness of the model and its ability to generalize to new data. Also, the collection of weather data was done by utilizing web scraping techniques to extract weekly weather data (e.g., rainfall, temperature, and humidity) from Weather Underground (wunderground.com).

Data Fields

- **Time.** Represents the specific year and week corresponding to each entry in the dataset.
- **Rainfall.** Denotes the observed average rainfall, measured in millimeters, for a specific week.
- **Humidity.** Refers to the observed average relative humidity, expressed as a percentage, for a specific week.
- **Max Temperature.** Represents the observed maximum temperature, measured in degrees Celsius, for a specific week.
- **Average Temperature.** Represents the observed average temperature, measured in degrees Celsius, for a specific week.
- **Min Temperature.** Represents the observed minimum temperature, measured in degrees Celsius, for a specific week.
- **Wind.** Represents the observed wind speed, measured in miles per hour (mph), for a specific week.

- **Cases.** Refers to the number of reported dengue cases during a specific week.

Data Integration and Preprocessing

The dengue case data was integrated with the weather data to create a com prehensive dataset, aligning the data based on corresponding timeframes. The dataset undergoed a cleaning process to address any missing values, outliers, and inconsistencies to ensure its accuracy and reliability. To ensure that all features and the target variable were on the same scale, a MinMaxScaler was applied to normalize both the input features (climate data) and the target variable (dengue cases).

Exploratory Data Analysis (EDA)

Trends, seasonality, and correlations between reported dengue cases and weather factors were thoroughly analyzed to identify potential relationships in the dataset. To support and illustrate these findings, a series of visualizations, including time-series plots and scatterplots, were developed, to highlight key patterns and relationships within the dataset.

Outbreak Detection

To detect outbreaks, we computed the outbreak threshold value of dengue cases using the formula,

$$\text{Outbreak Threshold Value} = \mu + 2\sigma \tag{3.1}$$

where μ is the historical mean and σ is the standard deviation.

It is important to take note that definitions of dengue outbreaks differ significantly across regions and time. This computation is subject to changes depending on how the surveillance units detect outbreaks themselves.

3.1.2 Develop and Evaluate Deep Learning Models for Dengue Case Forecasting

The deep learning models were developed and trained to forecast weekly dengue cases using historical weather data (rainfall, temperature, wind, and humidity) and dengue case counts. The dataset was normalized and divided into training and testing sets, ensuring temporal continuity to avoid data leakage. The methodology for preparing and training the model are outlined below.

Data Preprocessing

The raw dataset included weekly aggregated weather variables (rainfall, temperature, wind, humidity) and dengue case counts. The "Time" column was converted to a datetime format to ensure proper temporal indexing. To standardize the data for training, MinMaxScaler was employed, normalizing the feature values and target variable to a range of 0 to 1. This step ensured that the models could efficiently process the data without being biased by feature scaling differences.

LSTM Model

The dataset was split into training and test sets to evaluate the model's performance and generalizability:

- **Training Set:** 80% of the data (572 sequences) was used for model training, enabling the LSTM to learn underlying patterns in historical dengue case trends and their relationship with weather variables.
- **Test Set:** The remaining 20% of the data (148 sequences) was reserved for testing

To prepare the data for LSTM, a sliding window approach was utilized. Sequences of weeks of normalized features were constructed as input, while the dengue case count for the subsequent week was set as the target variable. This approach ensured that the model leveraged temporal dependencies in the data for forecasting. To enhance the performance of the LSTM model in predicting dengue cases, Bayesian Optimization was employed using the Keras Tuner library. The tuning process aimed to minimize the validation loss (mean squared error) by adjusting key model hyper-parameters. Table 3.1 summarizes the search space below:

Search Space	**LSTM Units**	**Learning Rate**
Min Value	32	0.0001
Max Value	128	0.01
Step	16	None
Sampling	Linear	Log
Tuner Configuration		
Max Trials	10	
Executions per Trial	3	
Validation Split	0.2	

Table 3.1: Hyperparameter Tuning: Search Space and Tuner Configuration

The hyperparameter tuning was conducted for three different window sizes of data: 5, 10, and 20. This allows the model to have the optimal hyperparameters used for each window size. Training was conducted over 100 epochs with early stopping to prevent overfitting while maintaining computational efficiency. A batch size of 1 was used, enabling the model to process individual sequences, which is suitable for smaller datasets but results in longer training times. The Adam optimizer, known for its adaptive learning capabilities and stability was employed.

To validate the effectiveness of the model, cross-validation was implemented. However, standard k-fold cross-validation randomly shuffles the data, which isn't suitable for time series since the order of observations is important. To address this, a time series-specific cross-validation strategy was used with TimeSeriesSplit from the scikit-learn library. This method creates multiple train-test splits where each training set expands over time and each test set follows sequentially. This approach preserves the temporal structure of the data while helping reduce overfitting by validating the model across different time segments.

After training, predictions on both the training and test datasets were rescaled to their original scale using the inverse transformation of MinMaxScaler. Model performance was evaluated using the mean squared error (MSE), root mean squared error (RMSE) and mean absolute error (MAE).

ARIMA

The ARIMA model was utilized to forecast weekly dengue cases, leveraging historical weather data—including rainfall, maximum temperature, and humidity—as exogenous variables alongside historical dengue case counts as the primary dependent variable. The dataset was partitioned into training (80%) and testing (20%) sets while maintaining temporal consistency.

To identify the optimal ARIMA configuration, a comprehensive grid search was performed across the following parameter ranges:

- Autoregressive order (p): 0 to 3
- Differencing order (d): 0 to 2
- Moving average order (q): 0 to 3

Each combination of (p,d,q) was used to fit an ARIMA model, and performance was evaluated based on the mean squared error (MSE) between the predicted and actual dengue cases on the test set. The parameter set that achieved the lowest MSE was selected as the final model configuration.

Following model selection, the best-fit ARIMA model was retrained on the training set and subsequently used to forecast dengue cases for the test period. The predictions were assigned to the `PredictedCases` column in the test dataset. Model performance was further assessed using key evaluation metrics, including MSE, root mean squared error (RMSE), and mean absolute error (MAE). Visual comparisons between actual and predicted dengue cases were produced through line plots to better illustrate the model's forecasting accuracy.

Seasonal ARIMA (SARIMA)

The SARIMA modeling process began with data preprocessing, which included handling missing values through interpolation or imputation, and standardizing features to ensure stable model training. The dataset was then split into training and testing sets in an 80:20 ratio, preserving the temporal order of observations. Seasonality analysis was conducted using time series decomposition and autocorrelation plots, which revealed a periodicity of 52 weeks—justifying the adoption of a seasonal model. To fine-tune the model, a grid search was performed over a range of SARIMA parameters (p,d,q)(P,D,Q)[S], while stationarity was validated using the Augmented Dickey-Fuller (ADF) test. The model was then trained on the dataset using rainfall, temperature, and humidity as exogenous variables, with convergence ensured by setting a maximum number of iterations. Residual diagnostics were used to confirm that residuals were uncorrelated, indicating a good model fit. For evaluation, forecasts were compared against actual values, and results were visualized with line plots. Finally, to validate the model's generalizability across different time periods, Time Series Cross-Validation with three folds was applied. This allowed assessment of the model's performance on multiple time segments, providing insights into its robustness in real-world forecasting scenarios.

Kalman Filter:

- Input Variables: The target variable (Cases) was modeled using three regressors: rainfall, max temperature, and humidity.
- Training and Testing Split: The dataset was split into 80% training and

20% testing to evaluate model performance.

- Observation Matrix: The Kalman Filter requires an observation matrix, which was constructed by adding an intercept (column of ones) to the regressors.

The Kalman Filter's Expectation-Maximization (EM) method was employed for training, iteratively estimating model parameters over 10 iterations. After training, the smoothing method was used to compute the refined state estimates across the training data. Observation matrices for the test data were constructed in the same manner as for the training set, ensuring compatibility with the learned model parameters. On the test data, the Kalman Filter applied these parameters to predict and correct the estimated dengue cases, providing more stable and accurate forecasts compared to direct regression models. Additionally, a hybrid Kalman Filter–LSTM (KF-LSTM) model was developed to combine the strengths of both approaches. In this setup, the LSTM model was first used to predict dengue cases based on historical data and weather features. The Kalman Filter was then applied as a post-processing step to the LSTM predictions, smoothing out noise and correcting potential errors.

Model Evaluation

- **MSE** represents the average of the squared differences between predicted and actual values. It penalizes larger errors more heavily.
- **RMSE**, the square root of MSE, provides a more interpretable value in the same units as the target (i.e., number of dengue cases).

- **MAE** calculates the average magnitude of the errors without considering their direction, giving a more straightforward understanding of the average prediction error.

Model Simulation:

After identifying the best-performing model among all the trained deep learning models, a simulation was conducted. Using the same parameters from the initial training, the selected model was retrained with the original dataset along with new data up to January 2025. The retrained model was then used to forecast dengue cases for the period from February 2025 to May 2025. Listing 3.1 shows a code snippet of the model training.

Listing 3.1: Code Snippet for Model Training

```
# Fit on train set
history = model.fit(
X_train, y_train,
epochs=100,
batch_size=1,
validation_split=0.2,
callbacks=[early_stop],
verbose=1
)

# Predict on 2025
y_pred_test = model.predict(X_test, verbose=0)
```

3.1.3 Integrate the Predictive Model into a Web-Based Data Analytics Dashboard

Dashboard Design and Development

- Design an intuitive, user-friendly web-based dashboard incorporating:
  - Interactive visualizations of yearly dengue case trends.
  - Data input and update forms for dengue and weather data.
  - Map display of dengue cases in each district in Iloilo City

Model Integration and Deployment

- Deploy the best-performing model within the dashboard as a backend service to enable real-time or periodic forecasting.

3.1.4 System Development Framework

The Agile Model is the birthchild of both iterative and incremental approaches in Software Engineering. It aims to be flexible and effective at the same time by being adaptable to change. It's also important to note that small teams looking to construct and develop projects quickly can benefit from this kind of methodology. As the Agile Method focuses on continuous testing, quality assurance is a guarantee since bugs and errors are quickly identified and patched.

Design and Developlment

After brainstorming and researching the most appropriate type of application to accommodate both the prospected users and the proposed solutions, the team has decided to proceed with a web application. Given the time constraints and available resources, it has been decided that the said means is the most pragmatic and practical move. The next step is to select modern and stable frameworks that align with the fundamental ideas learned by the researchers in the university. The template obtained from WVCHD and Iloilo Provincial Epidemiology and Surveillance Unit was meticulously analyzed to create use cases and develop a preliminary well-structured database that adheres to the requirements needed to produce a quality application. The said use cases serve as the basis of general features. Part by part, these are converted into code, and with the help of selected libraries and packages, it resulted in the desired outcome that may still modified and extended to achieve scalability.

Testing and Integration

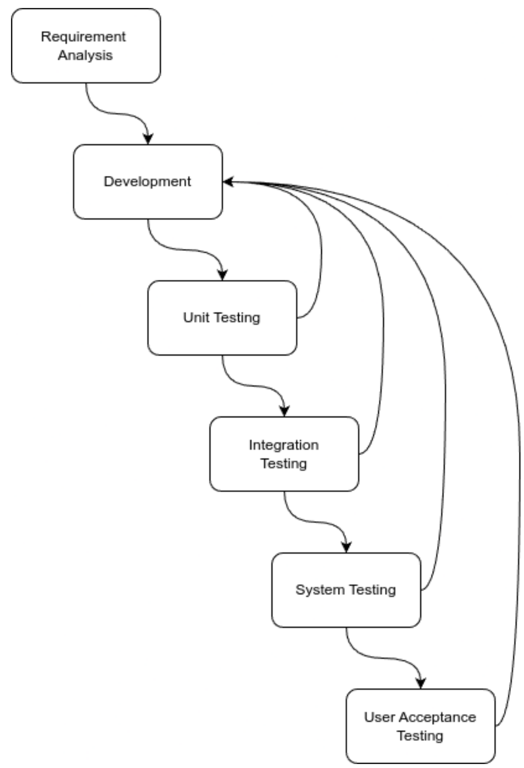


Figure 3.2: Testing Process for DengueWatch

Implementing testing is important to validate the system's performance and efficacy. Thus a series of tests were conducted to identify and resolve bugs during the developmental phase. Each feature was rigorously tested to ensure quality assurance, with particular emphasis on prerequisite features, as development cannot progress properly if these fail. Because of this, integration between each feature serves as a pillar for a cohesive user experience. Since dengue reports include confidential information, anonymized historical dengue reports were used to train the model and create the foundational architecture of the system. By using functional tests, data validation and visualization can be ensured for further continual improvements. Security testing is also important as it is needed to safeguard confidential information when the system is deployed. It includes proper authentication, permission views, and mitigating common injection attacks. Finally, a user acceptance test from the prospected users, in this case, doctors, nurses, and other health workers is crucial to assess its performance and user experience. It enables the developers to confirm if the system meets the needs of the problem.

3.2 Development Tools

3.2.1 Software

Github

GitHub is a cloud-based platform that tracks file changes using Git, an open-source version control system (*About GitHub and Git - GitHub Docs*, n.d.). It is used in the project to store the application's source code, manage the system's source version control, and serve as a repository for the Latex files used in the actual research.

Visual Studio Code

Visual Studio Code is a free, lightweight, and cross-platform source code editor developed by Microsoft (*Why Visual Studio Code?*, 2021). As VS Code supports this project's programming and scripting languages, it was chosen as the primary source code editor.

Django

Django is a free and open-sourced Python-based web framework that offers an abstraction to develop and maintain a secure web application. As this research aims to create a well-developed and maintainable application, it is in the best interest to follow an architectural pattern that developers and contributors in the future can understand. Since Django adheres to Model-View-Template (MVT) that promotes a clean codebase by separating data models, business logic, and presentation layers, it became the primary candidate for the application's backbone.

Next.js

A report by Statista (2024) claims that React is the most popular front-end framework among web developers. However, React has limitations that can be a nuisance in rapid software development, which includes routing and performance optimizations. This is where Next.js comes in—a framework built on top of React. It offers solutions for React's deficiency, making it a rising star in the framework race.

Postman

As the application heavily relies on the Application Programming Interface (API) being thrown by the backend, it is a must to use a development tool that facilitates the development and testing of the API. Postman is a freemium API platform that offers a user-friendly interface to create and manage API requests (*What is*

Postman? Postman API Platform, n.d.).

3.2.2 Hardware

The web application was developed on laptop computers with minimum specifications of an 11th-generation Intel i5 CPU and 16 gigabytes of RAM. Furthermore, an RTX 3060 mobile GPU was also utilized to retrain the integrated model locally.

3.2.3 Packages

Django REST Framework

Django Rest Framework (DRF) is a third-party package for Django that provides a comprehensive suite of features to simplify the development of robust and scalable Web APIs (Christie, n.d.). These services include Serialization, Authentication and Permissions, Viewsets and Routers, and a Browsable API .

Leaflet

One of the features of the web application is the ability to map the number of cases using a Choropleth Map. Leaflet is the only free, open-sourced, and most importantly, stable JavaScript package that can do the job. With its ultra-lightweight size, it offers a comprehensive set of features that does not trade off performance and usability (*Leaflet — an open-source JavaScript library for interactive maps*, n.d.).

Chart.js

Another feature of the application is to provide users with informative, approachable data storytelling that is easy for everyone to understand. The transformation of pure data points and statistics into figures such as charts is a big factor. Thus, there is a need for a package that can handle this feature without compromising the performance of the application. Chart.js is a free and open-source JavaScript package that is made to meet this criteria as it supports various types of charts (*Chart.js*, n.d.).

Tailwind CSS

Using plain CSS in production-quality applications can be counterproductive. Therefore, CSS frameworks were developed to promote consistency and accelerate the rapid development of web applications (Joel, 2021). One of these is Tailwind, which offers low-level utility classes that can be applied directly to each HTML element to create a custom design (*Tailwind CSS - Rapidly build modern websites without ever leaving your HTML.*, n.d.). Given the limited timeline for this project, using this framework is a wise choice due to its stability and popularity among developers.

Shadcn

Shadcn offers a collection of open-source UI boilerplate components that can be directly copied and pasted into one's project. With the flexibility of the provided components, Shadcn allows developers to have full control over customization and styling. Since this is built on top of Tailwind CSS and Radix UI, it is supported by most modern frontend frameworks, including Next.js (Shadcn, n.d.).

Zod

Data validation is integral in this web application since it will handle crucial data that will be used for analytical inferences and observations. Since Zod is primarily used for validating and parsing data, it ensures proper communication between the client and the server (Zod, n.d.).

3.3 Application Requirements

3.3.1 Backend Requirements

Database Structure Design

Determining how data flows and how it would be structured is crucial in creating the system as it defines how extendible and flexible it would be for future features and updates. Thus, creating a comprehensive map of data ensures proper normalization that eliminates data redundancy and improves data integrity. Figure 3.3 depicts the designed database schema that showcases the relationship between the application's entities.

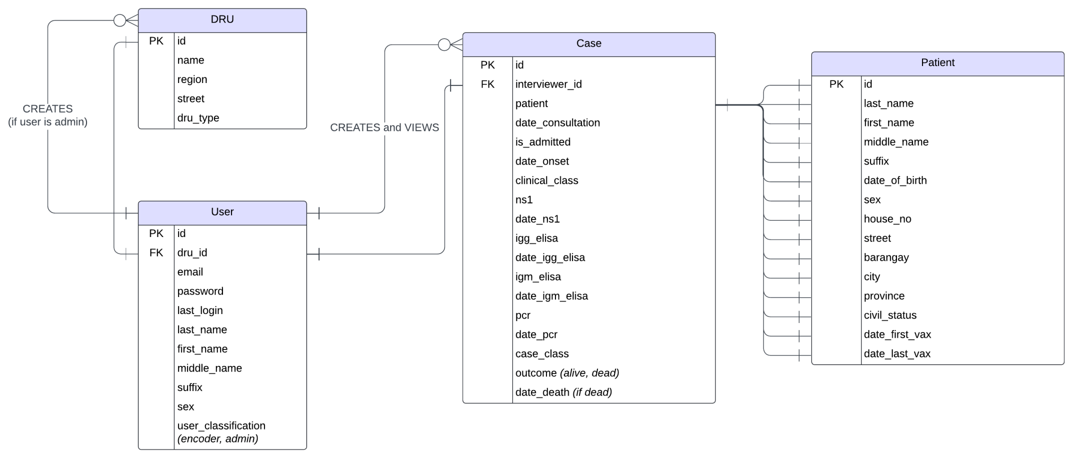


Figure 3.3: Entity-Relationship Database Schema Hybrid Diagram for DengueDash Database Structure

3.3.2 User Interface Requirements

Admin Interface

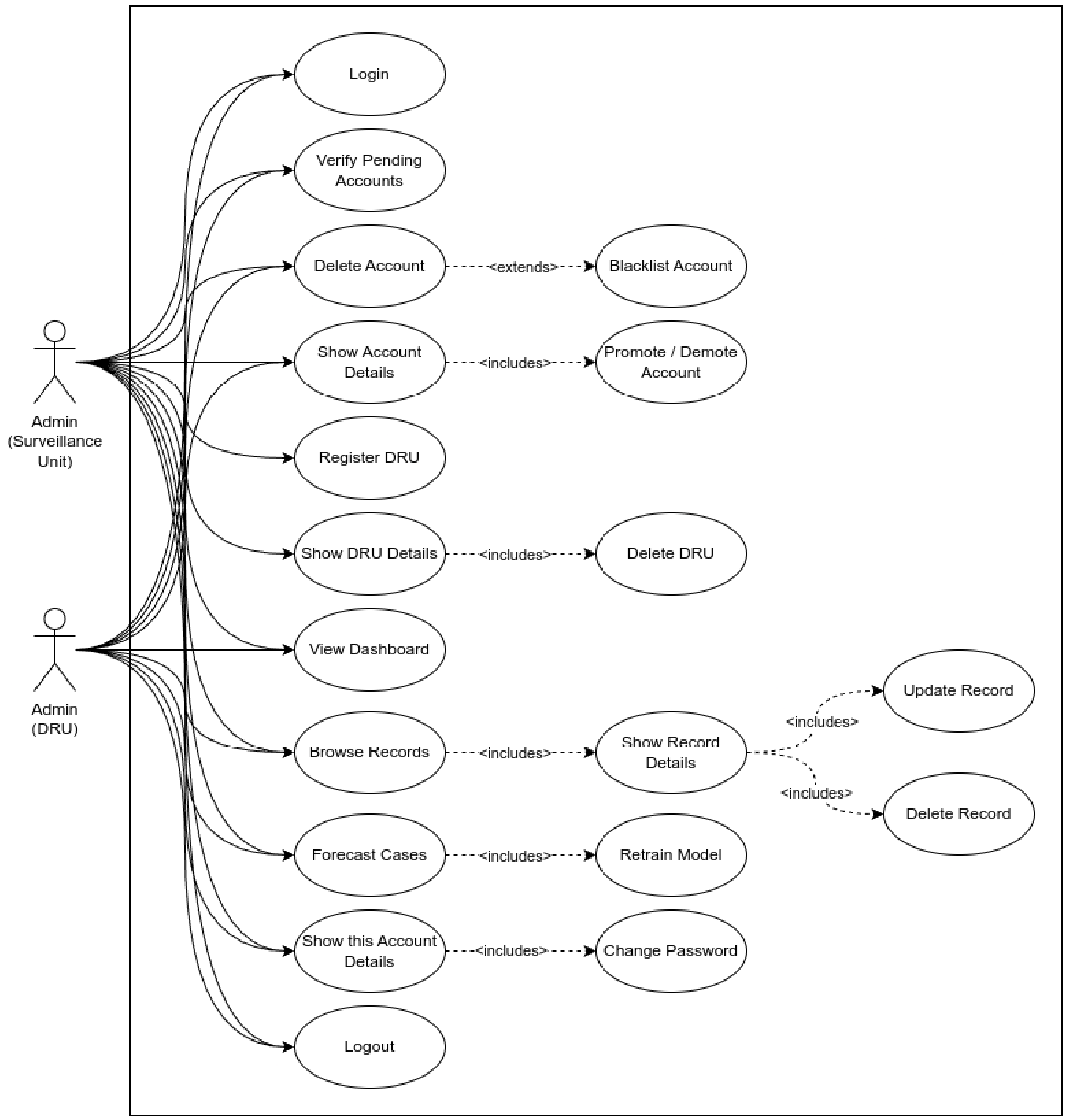


Figure 3.4: Use Case Diagram for Admins

Figure 3.4 shows the actions of an admin for a specific Disease Reporting Unit (DRU) and an admin for a specific Surveillance Unit can take in the application. Both of them include the management of accounts, browsing records, and forecasting and retraining all the consolidated data under their supervision. Most importantly, these users must verify the encoders who register under their jurisdiction before allowing their account to access the application in the name of safeguarding the integrity of the data. The only advantage of the latter type of administrator is that it has a one-step higher authorization as it manages the DRUs. In addition, only the authorized surveillance unit administrator can register and create a DRU to uphold transparency and accountability.

Encoder Interface

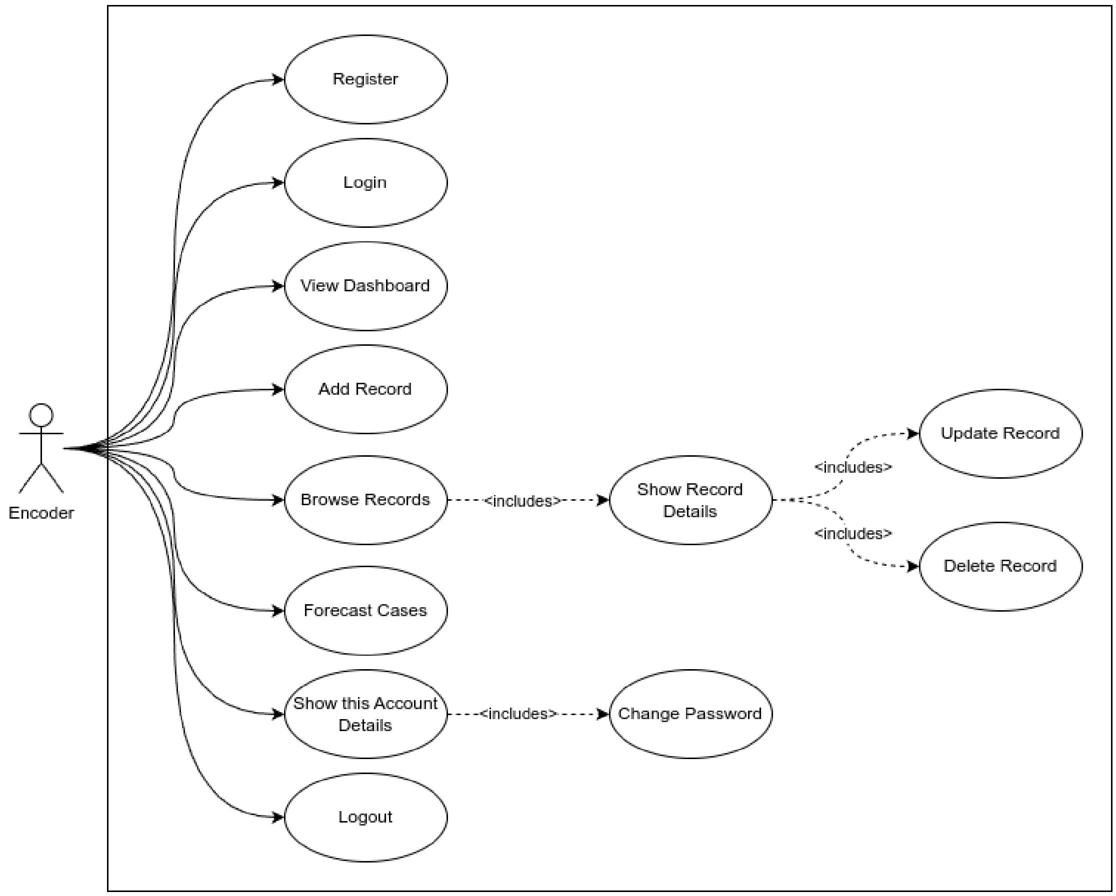


Figure 3.5: Use Case Diagram for Encoder

Figure 3.5, on the other hand, illustrates the use cases for the system's primary users. These users can register but must wait for further verification to access the application. Similar to the previous interfaces, encoders can browse and manage records, as well as forecast the consolidated cases under a specific surveillance or disease reporting unit, but they are not allowed to retrain the model. Lastly, they are the only type of user that can file and create dengue cases by filling out a form with the required details.

3.3.3 Security and Validation Requirements

Password Encryption

Storing passwords as plain text in the database is a disgrace and a mortal sin in production. It is important to implement precautionary methods such as hashing and salting, followed by encryption with a strong algorithm, to prevent bad actors from using the accounts for malicious transactions. By default, Django generates a unique random salt for each password and encrypts it with Password-Based Key Derivation Function 2 (PBKDF2) with a SHA256 hash function. Utilizing these techniques ensures that in the event of a data breach, cracking these passwords would be time-consuming and useless for the attackers.

Authentication

DengueWatch utilizes JSON Web Tokens (JWT) to authenticate the user. Since the mechanism operates in a stateless manner, tokens are served only after a successful login, eliminating the need for the server to keep a record of the token, which is vulnerable to session hijacking. In addition, these tokens are signed with a secret key, ensuring they have not been tampered with.

Data Validation

Both the backend and frontend should validate the input from the user to preserve data integrity. Thus, Zod is implemented in the latter to help catch invalid inputs from the user. By doing this, the user can only send proper requests to the server which streamlines the total workflow. On the other hand, Django has also a built-in validator that checks the data type and ensures that the input matches the expected format on the server side. These validation processes ensure that only valid and properly formatted data is accepted, which reduces the risk of errors and ensures consistency across the web application.

Chapter 4

Results and Discussion

4.1 Data Gathering

The data for dengue case prediction was gathered from a variety of reliable sources, enabling a comprehensive dataset spanning from January 2011 to October 2024. This dataset includes 720 rows of data, each containing weekly records of dengue cases along with corresponding meteorological variables, such as rainfall, temperature, and humidity.

1. Dengue Case Data: The primary source of historical dengue cases came from the Humanitarian Data Exchange and the Western Visayas Center for Health Development (WVCHD). The dataset, accessed through Freedom of Information (FOI) requests, provided robust case numbers for the Western Visayas region. The systematic collection of these data points was essential for establishing a reliable baseline for model training and evaluation.

2. Weather Data: Weekly weather data was obtained by web scraping from Weather Underground, allowing access to rainfall, temperature, wind, and humidity levels that correlate with dengue prevalence.

Time	Rainfall	MaxTemperature	AverageTemperature	MinTemperature	Wind	Humidity	Cases
2011-01-03	9.938571	29.444400	25.888890	23.888900	11.39	86.242857	5
2011-01-10	8.587143	30.000000	26.705556	24.444444	7.32	88.028571	4
2011-01-17	5.338571	30.000000	26.616667	25.000000	7.55	84.028571	2
2011-01-24	5.410000	30.555556	26.483333	20.555556	10.67	80.971429	7
2011-01-31	2.914286	28.333333	25.283333	18.650000	11.01	74.885714	2

Table 4.1: Snippet of the combined dataset

4.2 Exploratory Data Analysis

From Table 4.2, the dataset consists of 720 weekly records with 8 columns:

- **Time.** Weekly timestamps (e.g. "2011-w1")
- **Rainfall.** Weekly average rainfall (mm)
- **MaxTemperature, AverageTemperature, MinTemperature.** Weekly temperature data (°C)
- **Wind.** Wind speed (m/s)
- **Humidity.** Weekly average humidity (%)
- **Cases.** Reported dengue cases

From the statistics in Table 4.3, the number of cases ranges from 0 to 319. The average number of dengue cases per week is 23.74, with a median of 12 cases and a standard deviation of 37.14. The distribution is highly skewed, with some

#	Column	Non-Null Count	Data Type
0	Time	720 non-null	datetime64[ns]
1	Rainfall	720 non-null	float64
2	MaxTemperature	720 non-null	float64
3	AverageTemperature	720 non-null	float64
4	MinTemperature	720 non-null	float64
5	Wind	720 non-null	float64
6	Humidity	720 non-null	float64
7	Cases	720 non-null	int64

Table 4.2: Data Schema: Column Names, Non-Null Counts, and Data Types

weeks experiencing significant number of cases (up to 319 cases). Rainfall shows a wide variation (0 to 445mm), while temperature remains relatively stable, with an average of 28.1 degree celsius. Humidity levels ranges from 73% to 89% with a mean of 81.6%.

Statistic	Time	Rainfall	MaxTemperature	AverageTemperature	MinTemperature	Wind	Humidity	Cases
Count	720	720.000000	720.000000	720.000000	720.000000	720.000000	720.000000	720.000000
Mean	2017-12-02 11:22:00	13.957499	32.191142	28.110319	25.038472	6.172417	81.609442	23.744444
Min	2011-01-03 00:00:00	0.000000	-14.600000	24.494444	12.222222	1.910000	73.185714	0.000000
25%	2014-06-21 06:00:00	1.270000	31.666667	27.504167	25.000000	4.117500	79.885713	5.000000
50% (Median)	2017-12-07 12:00:00	4.318000	32.222222	28.161111	25.000000	5.725000	81.771429	12.000000
75%	2021-05-11 18:00:00	10.414000	32.777778	28.751389	25.555556	7.860000	83.503571	26.000000
Max	2024-10-28 00:00:00	445.008000	58.333333	30.916667	32.222222	19.200000	89.571429	319.000000
Std Dev	N/A	35.448846	2.616379	0.999800	1.291659	2.446703	2.831674	37.144813

Table 4.3: Descriptive Statistics of the Combined Dataset

Figure 4.1 illustrates the trend of weekly dengue cases over time. The data reveals periodic spikes in the number of cases, suggesting a seasonal pattern in dengue cases. Notably, peak cases are observed during certain periods approximately 3 years, potentially aligning with specific climatic conditions such as increased rainfall or temperature changes. This underscores the importance of incorporating climate variables into the forecasting model.

Figure 4.2 presents a time series subplot that combines rainfall and dengue cases to highlight potential non-linear associations between the two variables. In this figure, raw rainfall data is represented by blue scatter points (aligned with

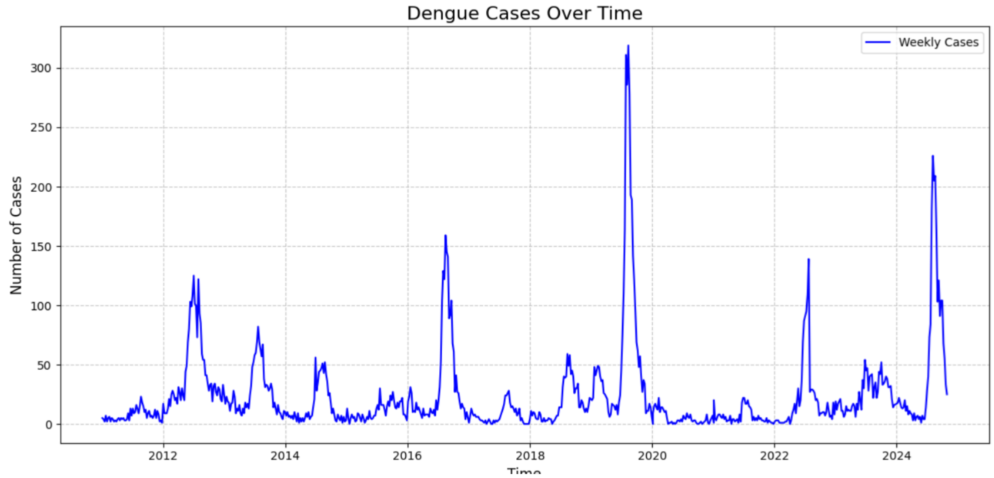


Figure 4.1: Trend of Dengue Cases

the left y-axis), while a blue solid line traces its 4-week rolling average to reveal underlying trends. Simultaneously, the red dashed line illustrates the smoothed trajectory of dengue cases (aligned with the right y-axis), also using a 4-week rolling average to reduce short-term fluctuations and emphasize longer-term patterns.

Notably, the plot suggests a recurring pattern. Periods of increased rainfall often precede or coincide with spikes in dengue cases. This observed relationship supports existing literature which proposes that higher rainfall contributes to the proliferation of mosquito breeding sites, particularly in stagnant water, thereby elevating the risk of dengue transmission.

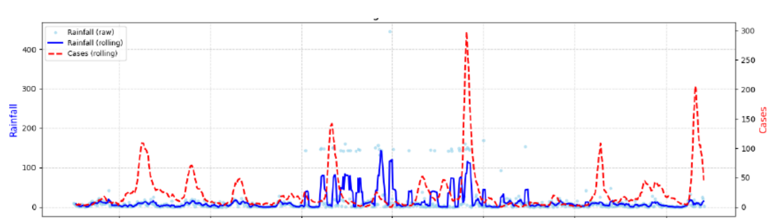


Figure 4.2: Rainfall and Dengue Cases Over Time

The KDE plots in Figure 4.3 illustrate the distributions of meteorological variables during outbreak and normal dengue weeks. The x-axes represent the actual values of each feature, while the y-axes show density, indicating how frequently values occur within each category. The graphs reveal that outbreak weeks tend to have moderately higher rainfall than weeks with no outbreak. This is evident in the way the curve for outbreak weeks is positioned slightly to the right of the curve for normal weeks. In terms of temperature, the distributions for both normal and outbreak weeks appear very similar; however, upon closer inspection, the curve for maximum temperature shows a slightly higher density at higher values during outbreak weeks. The same is true for humidity, with outbreak weeks showing greater density at higher humidity levels. These patterns suggest that dengue outbreaks are more likely to occur during warm, humid periods with relatively high rainfall. Based on these observations, rainfall, maximum temperature, and humidity were selected as the meteorological features for training the predictive models.

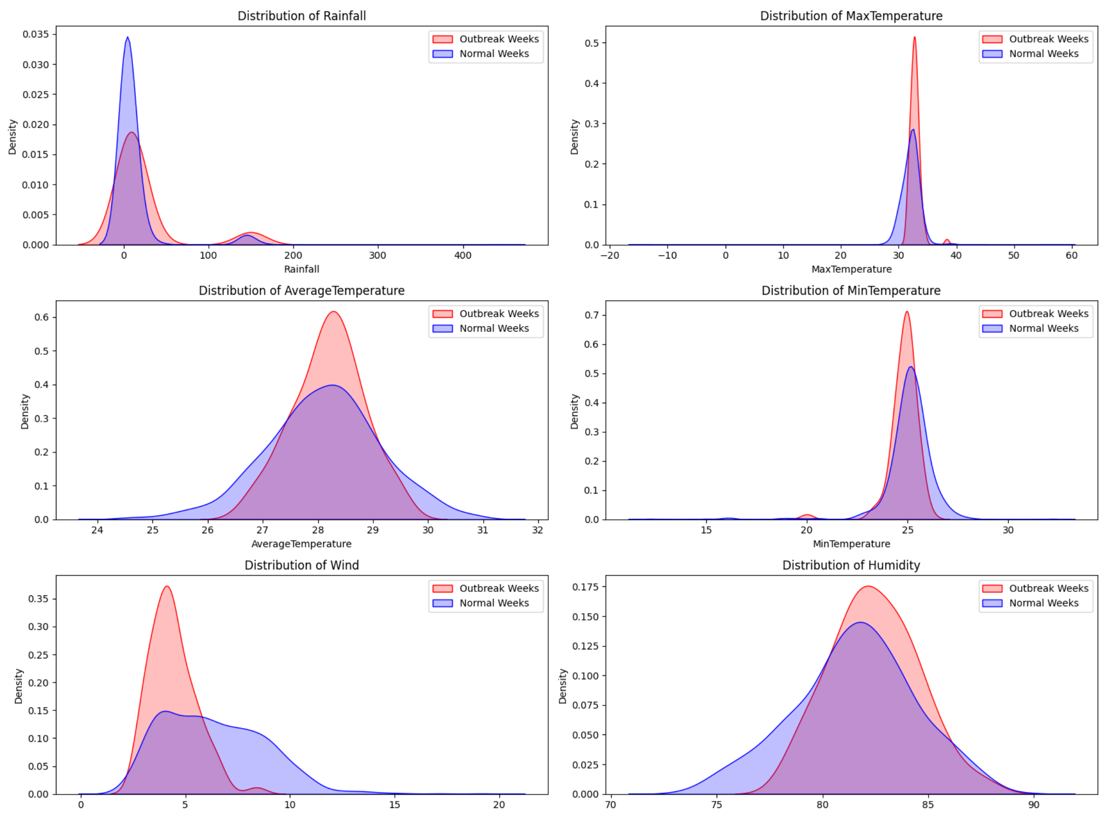


Figure 4.3: Kernel Density Estimate Plots of Meteorological Features

4.3 Outbreak Detection

To identify outbreaks, the researchers calculated the outbreak threshold value using the historical mean as the endemic channel. The threshold is determined using the formula:

$$\begin{aligned} \text{Outbreak Threshold Value} &= \mu + 2\sigma && (4.1)\\ &= 23.744444 + 2(37.144813) && (4.2)\\ &= 23.744444 + 74.289626 && (4.3)\\ &= 98.03407 && (4.4) \end{aligned}$$

where μ is the historical mean and σ is the standard deviation.

This result indicates that dengue cases exceeding 98 in Iloilo City can be considered an outbreak. However, it is important to note that this threshold serves only as a baseline.

4.4 Model Training Results

The models were evaluated using three commonly used regression metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). These metrics help assess how accurately each model forecasts dengue cases based on historical data. Table 4.4 presents a comparative analysis of the models using these metrics.

- **MSE** represents the average of the squared differences between predicted and actual values. It penalizes larger errors more heavily.
- **RMSE**, the square root of MSE, provides a more interpretable value in the same units as the target (i.e., number of dengue cases).
- **MAE** calculates the average magnitude of the errors without considering their direction, giving a more straightforward understanding of the average prediction error.

In simpler terms, lower values in these metrics indicate that the model is making more accurate predictions.

Method	**LSTM**	**Seasonal ARIMA**	**ARIMA**	**Kalman Filter**	**KF-LSTM**
Testing MSE	406.03	1261.20	1521.48	1474.82	785.35
Testing RMSE	20.15	34.45	39.00	38.40	25.56
Testing MAE	12.61	18.73	25.80	22.33	14.55
Best Parameters	Window Size: 5 Learning Rate: 0.01 Units: 64	(2,0,2)(0,1,1)	(1,2,2)	Observation Covariance: 10.0 Transition Covariance: 0.1 × Identity	Same as LSTM

Table 4.4: Comparison of different models for dengue prediction

As shown in Table 4.4, the LSTM model consistently achieved the lowest MSE (406.03), RMSE (20.15), and MAE (12.61) among all evaluated models. This suggests that, on average, the LSTM's predictions were about 12 to 20 cases away from the actual values, which is a strong indication of reliability for practical use in public health decision-making.

In contrast, traditional time series models like Seasonal ARIMA and ARIMA showed higher errors, indicating less accurate predictions. For example, the Seasonal ARIMA model had an RMSE of 34.45, which implies that its forecasts deviated from actual dengue case counts by around 34 cases on average, a significant discrepancy for health officials planning resource allocation.

The Kalman Filter and hybrid KF-LSTM models showed moderate performance. Although they did not outperform LSTM, the hybrid model (KF-LSTM) still reduced errors compared to the standalone Kalman Filter.

These results highlight the potential of LSTM-based models to provide timely and accurate forecasts that can support early intervention, resource planning, and policy formulation to combat dengue outbreaks in Iloilo City.

4.4.1 LSTM Model

The LSTM model was tuned for the following parameters: learning rate and units. The hyperparameter tuning was conducted for each window size, finding the best parameters for each window size. Further evaluating which window size is most suitable for the prediction model, Table 4.5 shows the evaluation metrics for each window size used in the LSTM model training.

Window Size	**MSE**	**RMSE**	**MAE**	**R^2**
5	**406.03**	**20.15**	**12.61**	**0.76**
10	**1037.77**	**32.21**	**26.79**	**0.39**
20	**427.39**	**20.67**	**13.61**	**0.75**

Table 4.5: Comparison of Window Sizes

The results indicate that a window size of 5 weeks provides the most accurate predictions, as evidenced by the lowest MSE and RMSE values. Furthermore, the R^2 score of 0.76 indicates that 76% of the variability in the target variable (cases) is explained by the independent variables (the inputs) in the model, making it a reliable configuration overall.

As shown in Table 4.6, the results from time series cross-validation indicate consistent performance trends, with a window size of 5 yielding the highest average RMSE across all folds compared to the other window sizes.

Window Size	Average RMSE	Average MAE	Average R^2
5	**16.69**	**9.06**	**0.79**
10	**17.08**	**10.40**	**0.75**
20	**16.93**	**8.75**	**0.81**

Table 4.6: Time-Series Cross Validation Results: Comparison of Window Sizes

Figure 4.4 illustrates the model's performance in predicting dengue cases for each fold using a window size of 5. As shown in the plot, the training set progressively increases with each fold, mimicking a real-world scenario where more data becomes available over time for dengue prediction. Figure 4.5 demonstrates that the predicted cases closely follow the trend of the actual cases, indicating that the LSTM model successfully captures the underlying patterns in the data. It is also evident that as the fold number increases and the training set grows, the accuracy of the predictions on the test set improves. Despite the test data being unseen, the model exhibits a strong ability to generalize, suggesting it effectively leverages past observations to predict future trends.

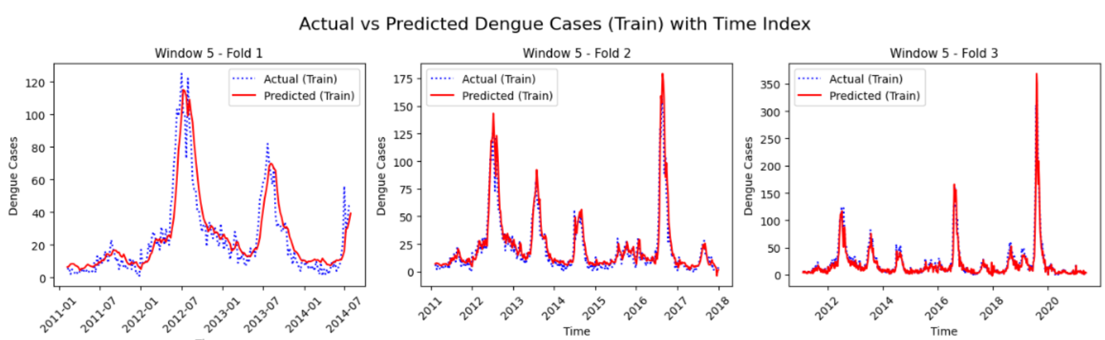


Figure 4.4: Training Folds - Window Size 5

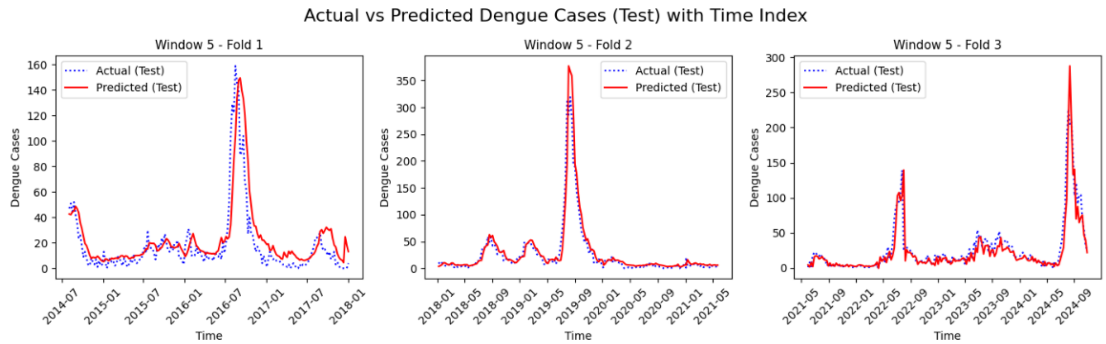


Figure 4.5: Testing Folds - Window Size 5

4.4.2 ARIMA Model

The ARIMA model was developed to capture non-seasonal trends in the data. To determine the best model configuration, grid search was used to explore various combinations of ARIMA parameters, ultimately selecting **ARIMA(1, 2, 2)**. The model was iteratively refined over **400 iterations** to ensure convergence to an optimal solution. Figure 4.6 illustrates the comparison between actual and predicted dengue cases in the test set. As shown in the plot, the ARIMA model struggled to capture the non-linear characteristics and abrupt spikes in the data. Consequently, it failed to accurately reflect the fluctuations and outbreak patterns seen in the actual case counts.

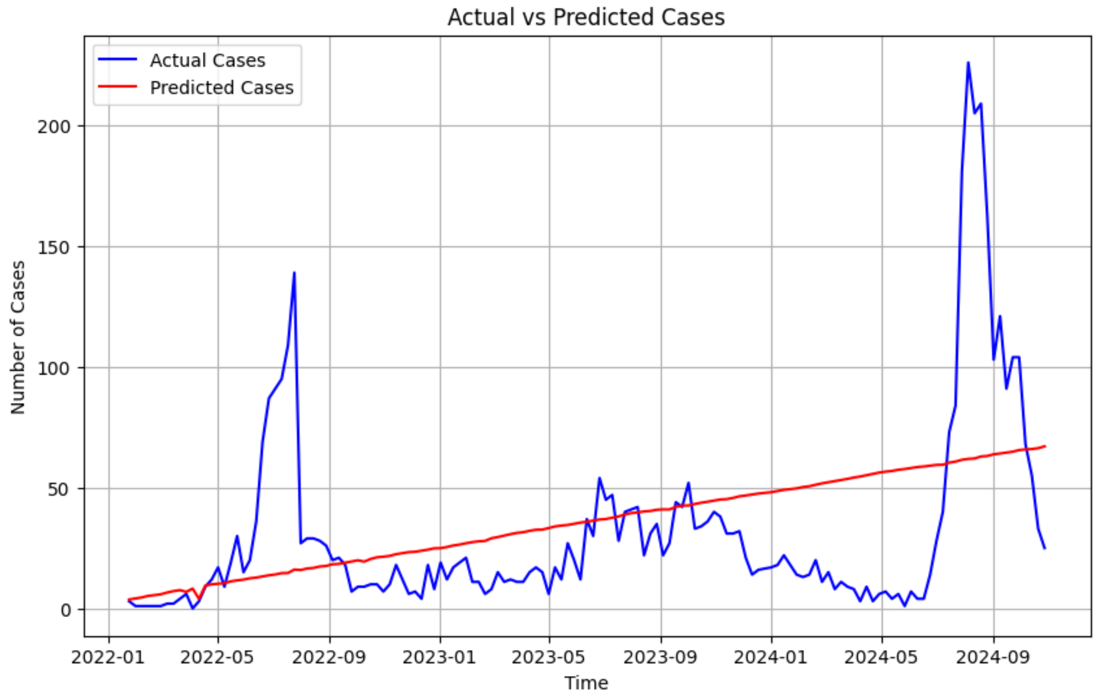


Figure 4.6: ARIMA Prediction Results for Test Set

The model's performance was assessed using regression metrics to evaluate its forecasting capability. The ARIMA model yielded the following error metrics:

- **MSE (Mean Squared Error)**: 1521.48
- **RMSE (Root Mean Squared Error)**: 39.01
- **MAE (Mean Absolute Error)**: 25.80

4.4.3 Seasonal ARIMA (SARIMA) Model

To address the limitations of the ARIMA model, a Seasonal ARIMA (SARIMA) model was developed to capture both non-seasonal and seasonal variations in the data.

This model incorporates seasonal parameters, which were tuned using grid search to find the best configuration: **SARIMA(2, 0, 2)(0, 1, 1)[52]**. As with ARIMA, **400 iterations** were applied to ensure a robust fit. As shown in Figure 4.7, the SARIMA model demonstrates a notable improvement in performance. Unlike its non-seasonal counterpart, it effectively captures the general trend and aligns more closely with the peaks observed in the actual dengue cases, indicating its ability to model seasonal dynamics.

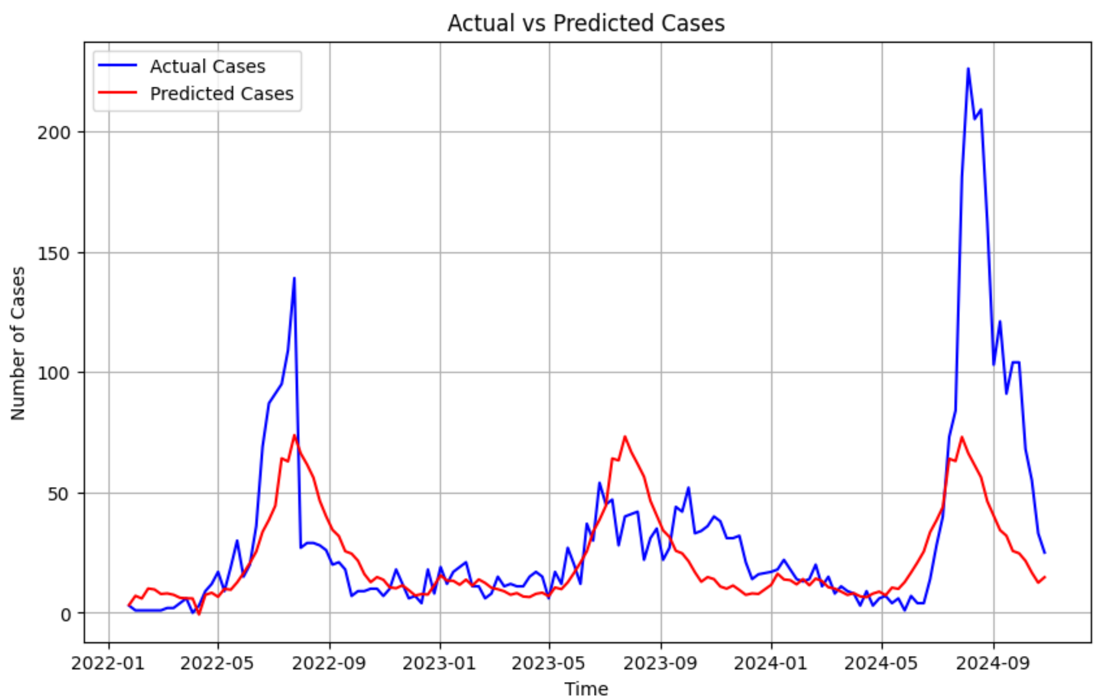


Figure 4.7: Seasonal ARIMA Prediction Results for Test Set

The model's performance was assessed using regression metrics to evaluate its forecasting capability. The SARIMA model yielded the following error metrics:

- **MSE**: 1109.69
- **RMSE**: 33.31
- **MAE**: 18.09

The lower error values, when compared to the ARIMA model, highlight the SARIMA model's superior capability in forecasting dengue cases. Its effectiveness in capturing seasonal patterns contributed to a more accurate representation of the actual cases.

After training the model, the SARIMA model was validated using the same Time Series Cross-Validation strategy employed in the LSTM model. Table 4.7 presents the performance metrics for each fold, as well as the average metrics across all folds. The average RMSE and MAE values were close to those obtained during the initial training phase, indicating that the SARIMA model performed consistently across different time segments.

Fold	**MSE**	**RMSE**	**MAE**
1	659.68	25.68	16.00
2	2127.49	46.12	21.30
3	996.43	31.56	18.89
Average	**1261.20**	**34.45**	**18.73**

Table 4.7: Comparison of SARIMA performance for each fold

4.4.4 Kalman Filter Model

Figure 4.8 shows the comparison between the actual dengue cases and the predicted values on the test set. As illustrated in the plot, the Kalman Filter model demonstrates a moderate ability to follow the general trend of the actual data.

While it effectively captures some rising and falling patterns, it still struggles to accurately replicate the sharp peaks and extreme values found in the real case counts. This limitation is particularly noticeable during the large spikes in 2022 and 2024. The model's performance was evaluated using standard regression metrics. The results are as follows:

$$\text{MSE} = 1474.82, \quad \text{RMSE} = 38.40, \quad \text{MAE} = 22.34$$

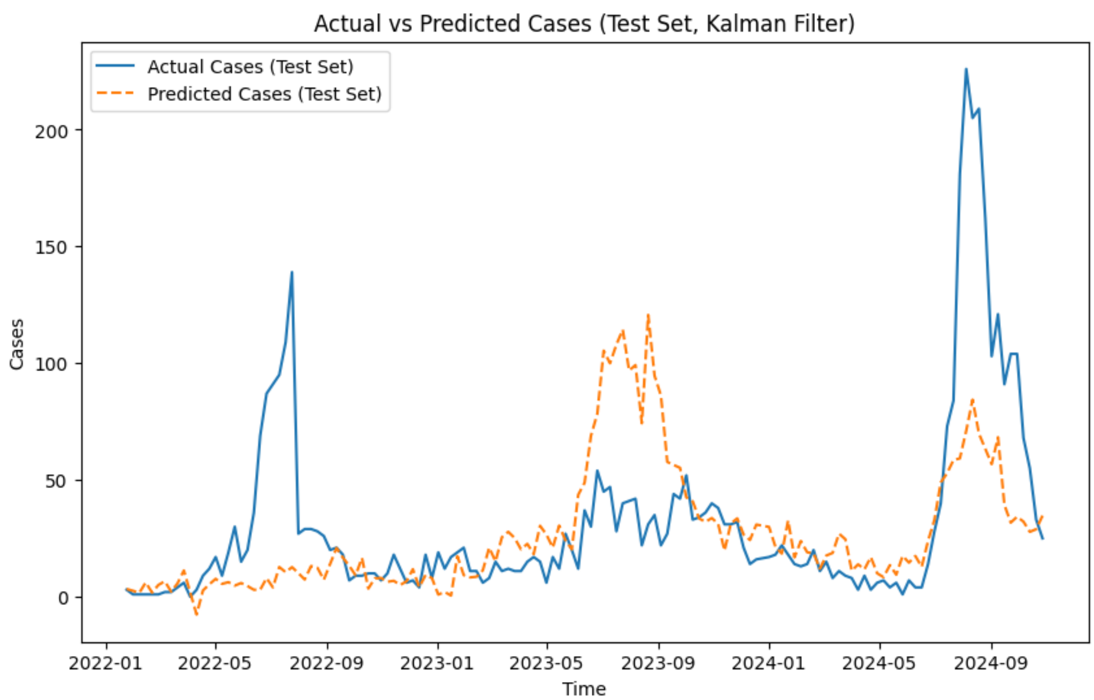


Figure 4.8: Kalman Filter Prediction Results for Test Set

The Kalman Filter was then combined with the LSTM model in order to see improvements in its predictions. Table 4.8 shows the metrics across three folds using the same Time Series Cross Validation Strategy employed in the previous models to see how it performed on different time segments.

Fold	**MSE**	**RMSE**	**MAE**
1	113.59	10.66	6.42
2	752.51	27.43	12.11
3	1489.95	38.60	25.13
Average	**785.35**	**25.56**	**14.55**

Table 4.8: Comparison of KF-LSTM performance for each fold

As can be seen in the table above, the performance of the hybrid model demonstrated improvements in all metrics as compared to just using the Kalman Filter alone.

4.5 Model Simulation

To evaluate the LSTM model's real-world forecasting ability, a simulation was conducted to predict dengue cases for the year 2025. The model was retrained exclusively, using the parameters found from the initial training, on data from 2011 to January 2025, using both dengue cases and weather variables. Importantly, the actual dengue case values for 2025 were never included during training. Instead, only the weather variables collected for 2025 were input into the model to generate predictions for that year. After prediction, the forecasted dengue cases for 2025 were compared against the true observed cases to assess the model's accuracy. Figure 4.9 shows that the predicted values closely follow the trend, although it may overestimate the dengue cases in some weeks.

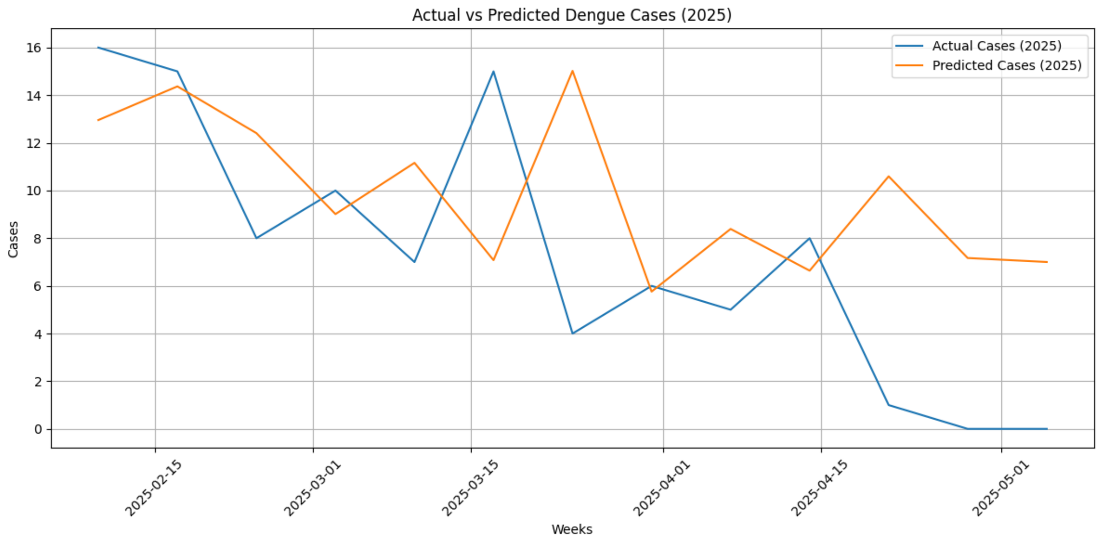


Figure 4.9: Predicted vs Actual Dengue Cases 2025

Retraining the model is essential to ensure it remains accurate and responsive to the evolving trends of dengue case patterns over time. Ideally, the model should be updated whenever new data becomes available to capture recent dynamics. However, given the computational cost associated with retraining, a more practical approach is to update the model on a monthly basis. This allows the incorporation of approximately four weeks' worth of new data, providing a meaningful update to the model's predictive capabilities without excessive resource consumption. Furthermore, this schedule aligns with the typical data release cycle of provincial health offices, which, based on the researchers' experience, usually occurs monthly. This balance between accuracy and efficiency ensures that the model remains both up-to-date and manageable in real-world deployment.

4.6 System Prototype

4.6.1 Home Page

The Home Page is intended for all visitors to the web application. The Analytics Dashboard, which displays relevant statistics for dengue cases at a certain time and location, is the primary component highlighted, as seen in Figure 4.10. This component includes a combo chart that graphs the number of dengue cases and deaths per week in a specific year, a choropleth map that tracks the number of dengue cases per barangay in a location, and various bar charts that indicate the top constituent places affected by dengue.

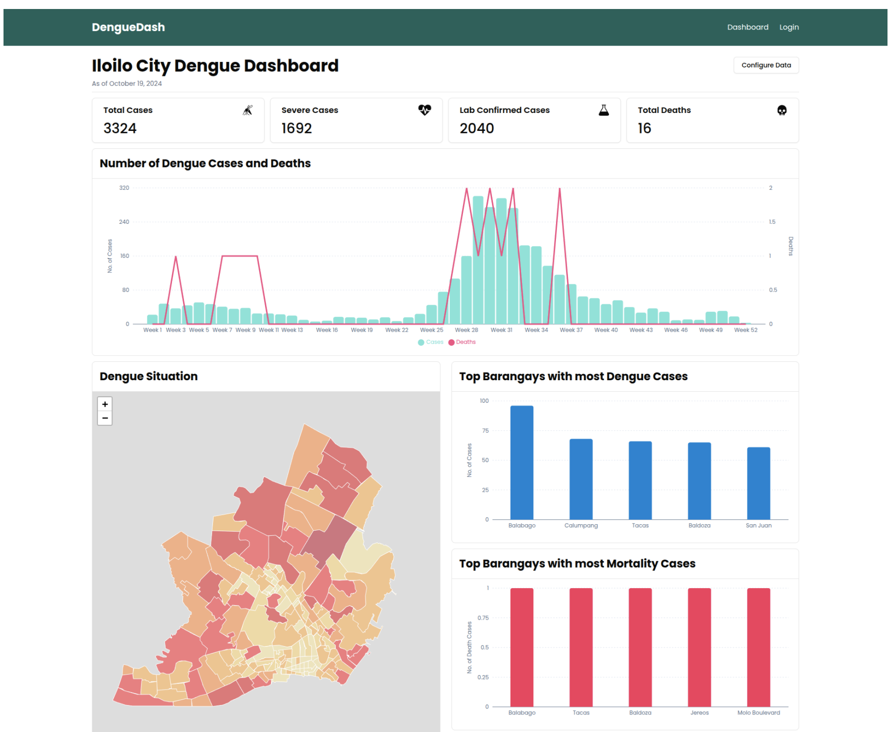


Figure 4.10: Home Page

4.6.2 User Registration, Login, and Authentication

The registration page, as shown in 4.11 and 4.12, serves as a gateway to access the authenticated pages of the web application. Only prospective encoders can register an account, as administrator accounts are created by existing administrator accounts to protect the integrity of the data in production. After registering, the ”encoder account” cannot access the authorized pages yet as it needs to be verified first by an administrator managing the unit the user entered. Because of this, proper identification (user’s picture and employee identification card) is mandatory to help the admins verify the identity of the registrant. Once verified, the user can log in to the system through the page shown in Figure 4.13. After entering the correct credentials, which consist of an email and password, the system uses HTTP-only cookies containing JSON Web Tokens (JWT) to prevent vulnerability to Cross-site Scripting (XSS) attacks. It will then proceed to the appropriate page for the type of user it belongs to. Logging out, on the other hand, will remove both the access and refresh tokens from the browser and will blacklist the latter token to make it unusable for security purposes.

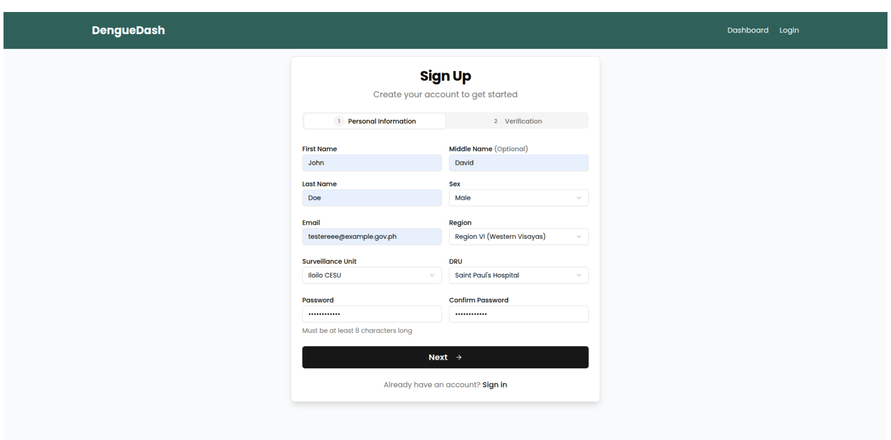


Figure 4.11: Personal Information Tab of Sign Up Page

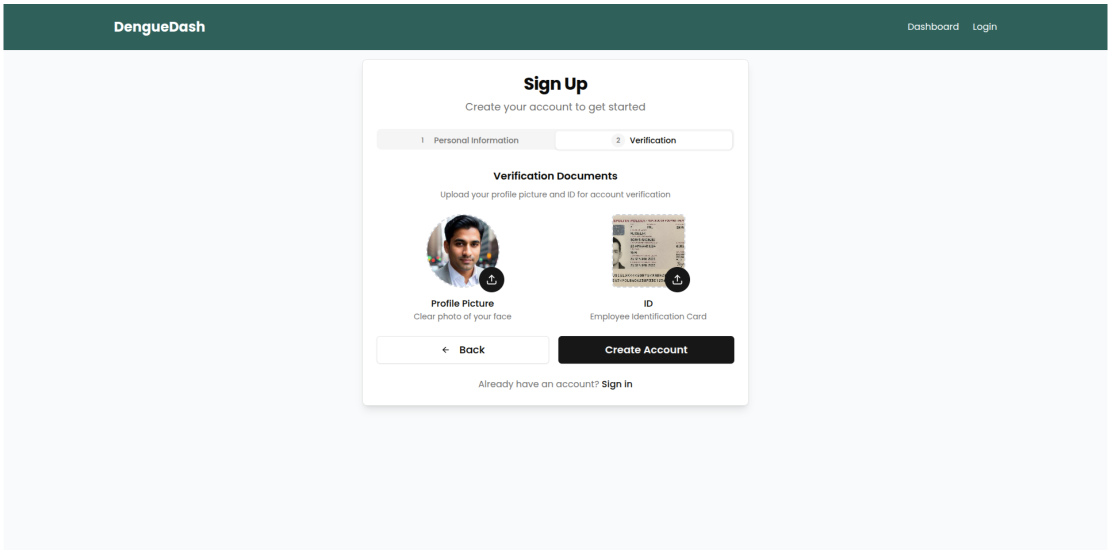


Figure 4.12: Verification Tab of Sign Up Page

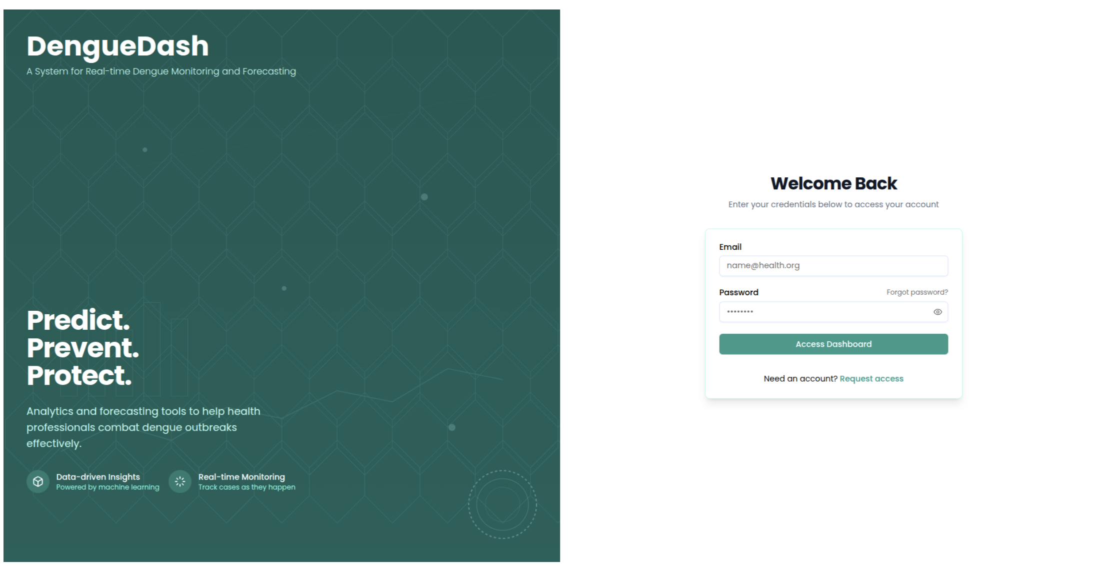


Figure 4.13: Login Page

4.6.3 Encoder Interface

Case Report Form

Figures 4.14 and 4.15 show the digitized counterpart of the form obtained from the Iloilo Provincial Epidemiology and Surveillance Unit. As the system aims to support expandability for future features, some fields were modified to accommodate more detailed input. It is worth noting that all of the included fields adhere to the latest Philippine Integrated Disease and Surveillance Response (PIDSR) Dengue Forms, which the referenced form was based on. By doing this, if implemented on a national scale, the transition between targeted users will be easier. Moreover, the case form includes the patient's basic information, dengue vaccination status, consultation details, laboratory results, and the outcome. On the other hand, encoders can also create case records using a "bulk upload" feature that makes use of a formatted CSV file template. As shown in Figure 4.16, an encoder can download the template using the "Download Template" button, and insert multiple records inside the file, then upload it by clicking the "Click to upload" button. The web application automatically checks the file for data inconsistencies and validation.

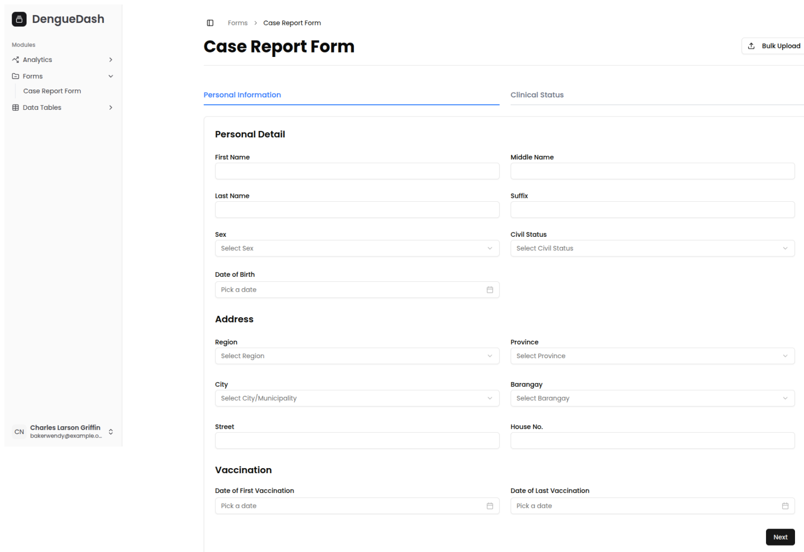


Figure 4.14: First Part of Case Report Form

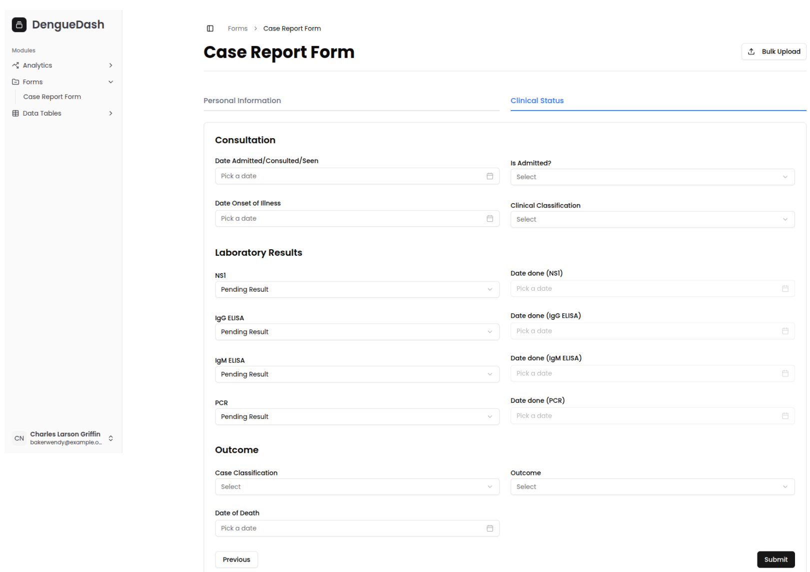


Figure 4.15: Second Part of Case Report Form

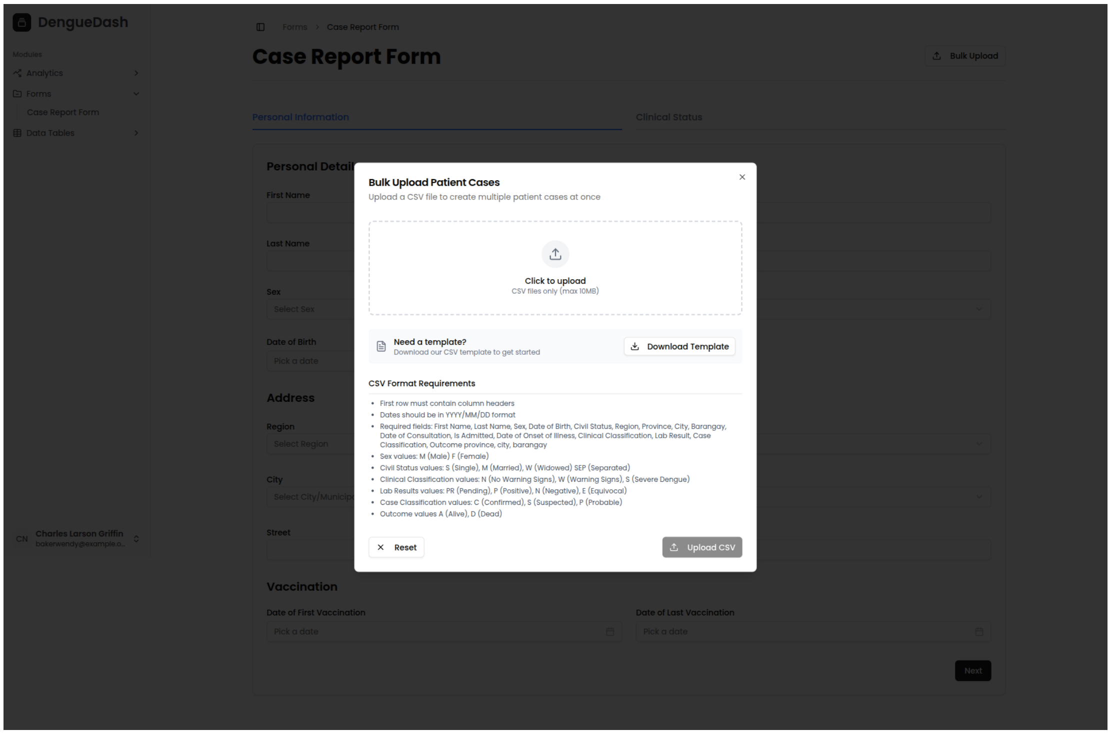


Figure 4.16: Bulk Upload of Cases using CSV

Browsing, Update, and Deletion of Records

Once the data generated from the case report form or the bulk upload is validated, it will be assigned as a new case and can be accessed through the Dengue Reports page, as shown in Figure 4.17. The said page displays basic information about the patient related to a specific case, including their name, address, date of consultation, and clinical and case classifications. It is also worth noting that it only shows cases that the user is permitted to view. For example, in a local Disease Reporting Unit (DRU) setting, the user can only access records that belong to the same DRU. Additionally, users can search for cases by name, location, date of consultation, or classifications associated with the specific query, making it easier to find pertinent information quickly and efficiently. On the other hand, in a consolidated surveillance unit such as a regional, provincial, or city quarter, its users can view all the records from all the DRUs that report to them. Moving forward, Figure 4.18 shows the detailed case report of the patient on a particular consultation date.

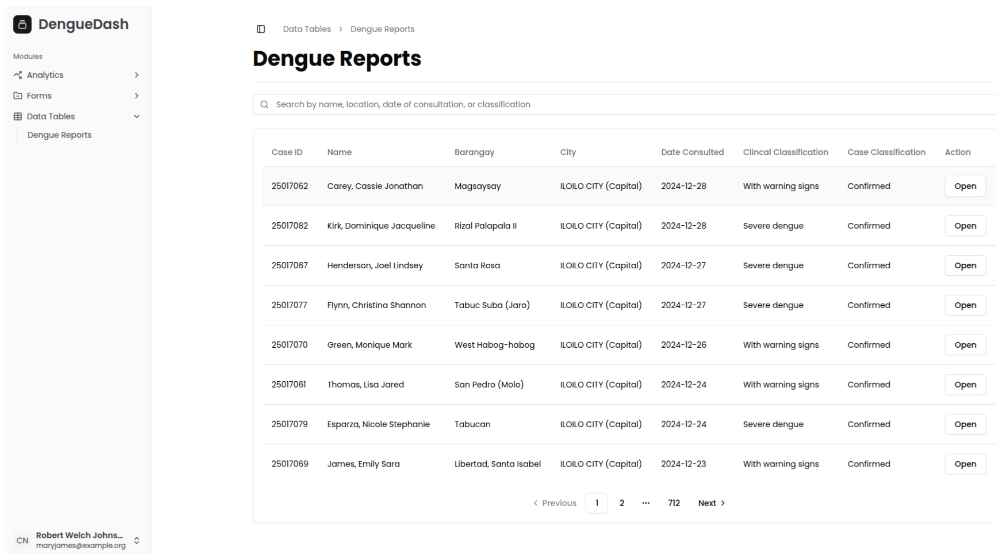


Figure 4.17: Dengue Reports

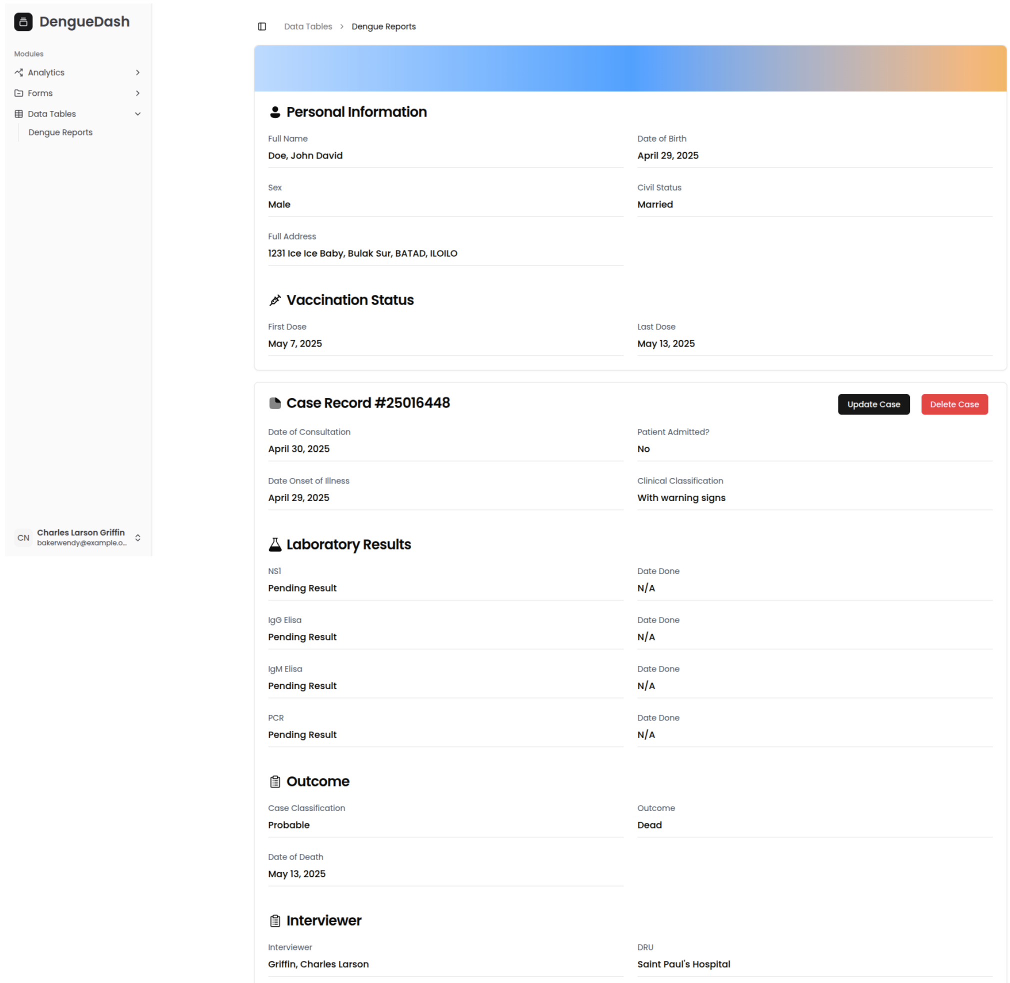


Figure 4.18: Detailed Case Report

To update the case, the user can click the "Update Case" button, where a dialog will appear, and the updateable fields will be shown. It is worth noting that in this case, only fields under Laboratory Results and Outcome are included since they are the only ones that are time-based, where the result may change in the future. After updating, a prompt will show confirming the user's action. Moving forward, to delete a case record, the user must click the "Delete Case" button, and a prompt verifying the action will appear. After confirming, the case will be deleted permanently.

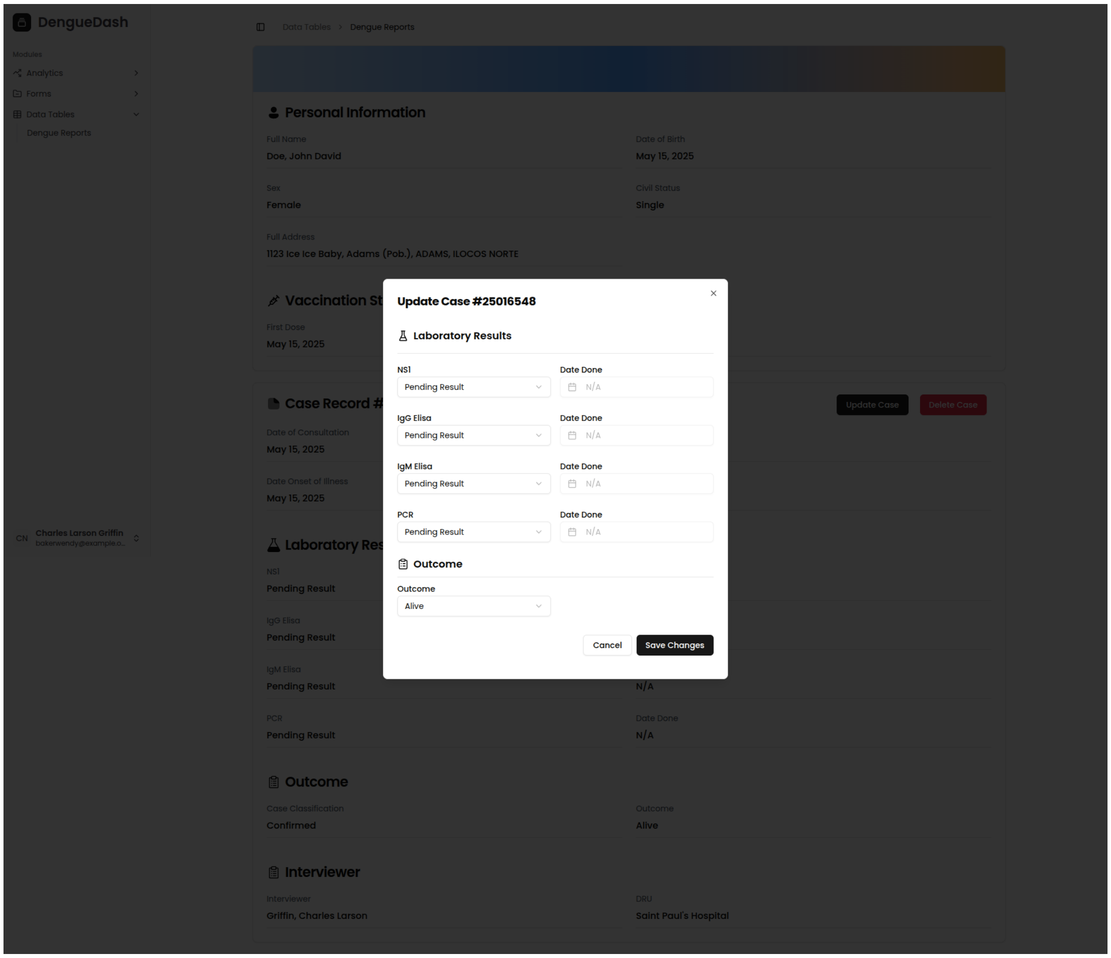


Figure 4.19: Update Report Dialog

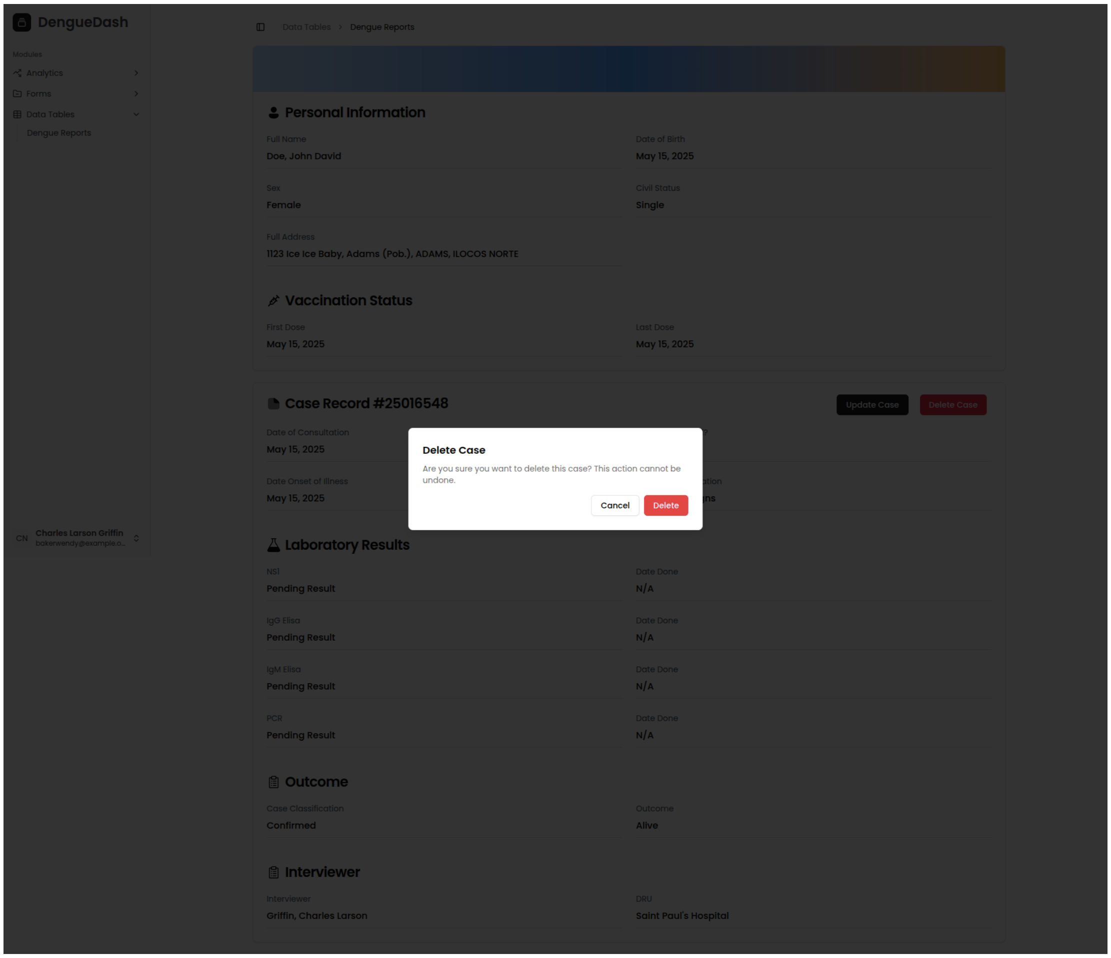


Figure 4.20: Delete Report Alert Dialog

Forecasting

The pièce de résistance of the web application's features is the Forecasting Page. This is where users can forecast dengue cases for the next few weeks. To predict, the application utilizes the exported LSTM model in a Keras format derived from training the consolidated data from the database. The said file stores the model's architecture and the learned parameters, which include the weights and biases to predict cases without training the data again. Furthermore, it requires the recent weekly dengue cases and weather variable data (temperature, humidity, and rainfall) to form a sequence based on the window size, and the forecasted weather data via OpenWeatherAPI. Due to the limitations posed to the current subscribed student plan in the said API, only two weeks of forecasted weather data can be fetched. As a result, the web application can predict dengue cases for the next two following weeks. Moving forward, the Forecasting page, as shown in Figure 4.21, introduces a user-friendly interface that shows the current cases for the week and the predictions for the next two weeks with a range of 90 percent to 110 percent confidence interval that is presented in a simple but organized manner. There is also a line chart that shows the number of cases from the last 5 weeks plus the forecasted weekly cases. In addition, the current weather data for a specific week is also shown, as well as the forecasted weather data fetched from the said API. Lastly, locations where dengue cases have been reported for the current week are listed in the Location Risk Assessment component.

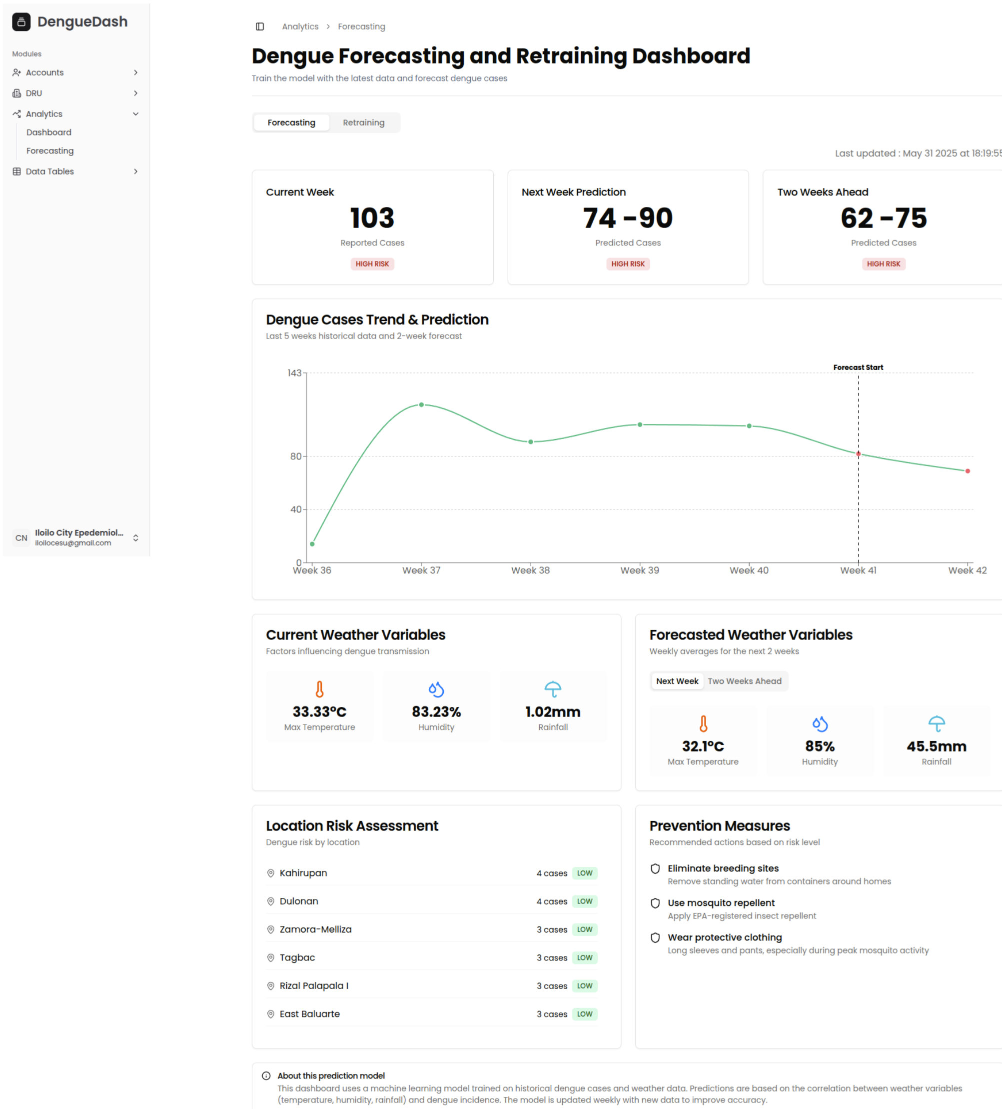


Figure 4.21: Forecasting Page

4.6.4 Admin Interface

Retraining

With LSTM being the best-performing model among the models used in forecasting dengue cases, it is the model chosen to power the prediction and retraining of the consolidated data within the web application. Since the retraining process consumes a lot of processing power and requires a more advanced understanding of how it works, it was decided that the said feature should only be available to admin users of surveillance units. Furthermore, the retraining component in the Forecasting page includes three additional components that include the configuration of LSTM parameters (Figure 4.22), the actual retraining of the consolidated data from the database (Figure 4.23), and the results of the retraining that shows the current and previous model metrics depending on the parameters entered (Figure 4.24). It is also worth noting that when training, the model used a seeded number to promote reproducibility.

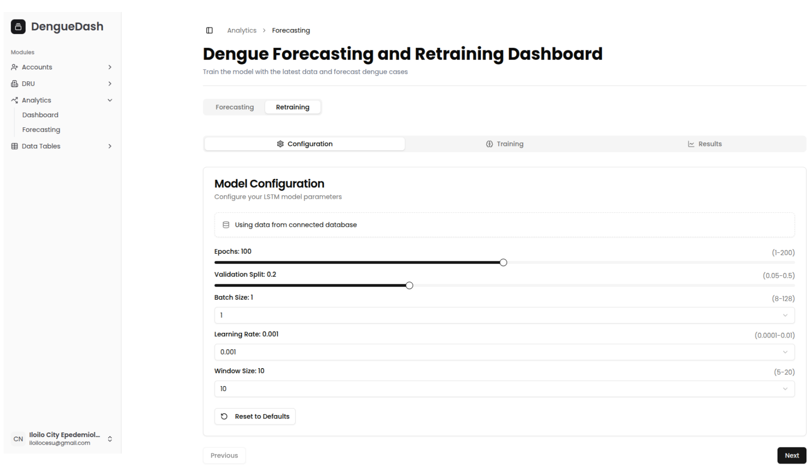


Figure 4.22: Retraining Configurations

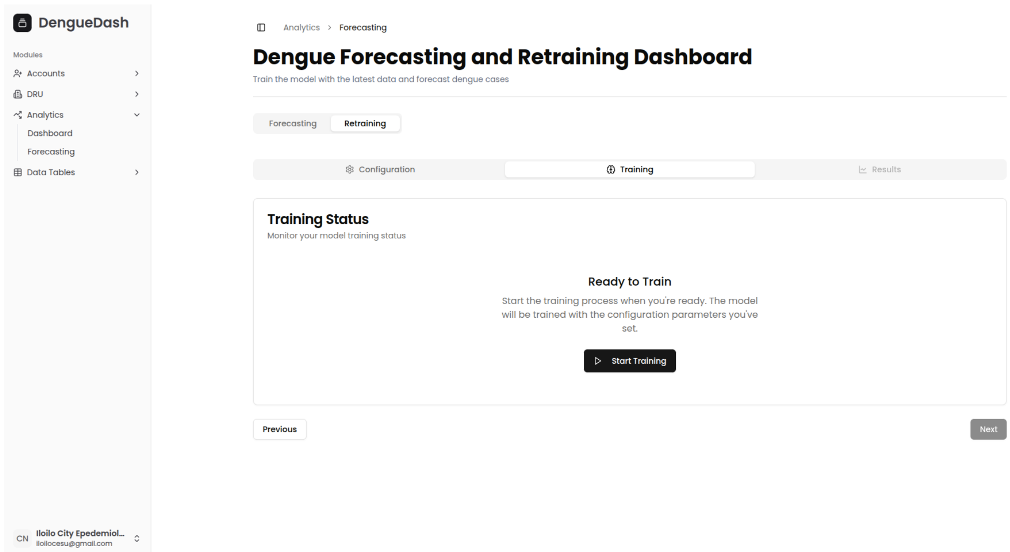


Figure 4.23: Start Retraining

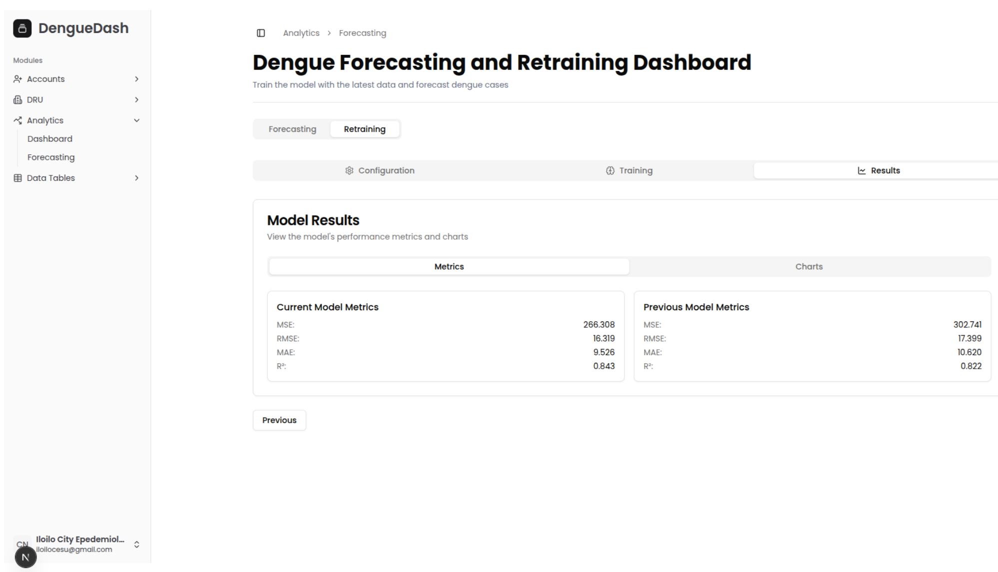


Figure 4.24: Retraining Results

Managing Accounts

Proper management of accounts is important to protect the integrity and confidentiality of data. Thus, it is crucial for administrators to track their users and control the flow of information. As discussed in the user registration of encoders, admin users from a specific DRU or surveillance unit have the power to grant them access to the web application. Figure 4.26 illustrates the interface for this scenario, as the admins can approve or reject their applications. Once approved, these users can access the features given to encoders and may be promoted to have administrative access, as shown in Figure 4.27. Both Figure 4.26 and 4.27 also show the expanded details of the user, which include personal information, proof of identification, and brief activity details within the system. When deleting an account, the user's email will be blacklisted and illegible to use when creating another account, and all the cases reported by this user will be soft-deleted. However, the blacklist status can be reverted by clicking the "Unban" button, which would make the user of the email able to register to the web application again as shown in Figure 4.28.

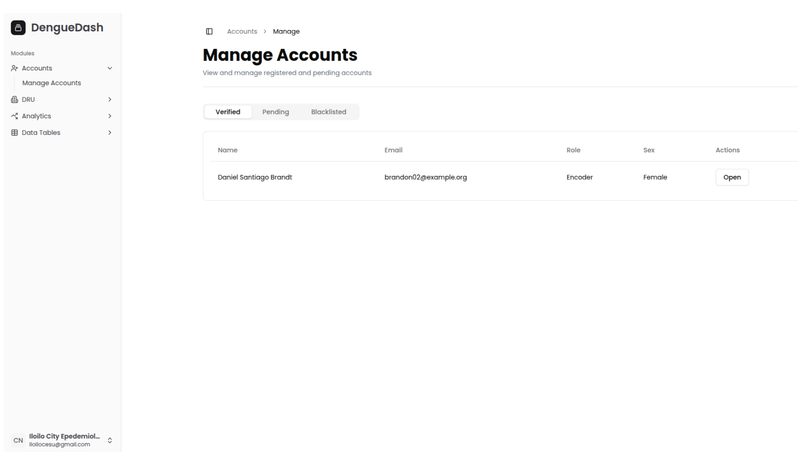


Figure 4.25: List of Verified Accounts

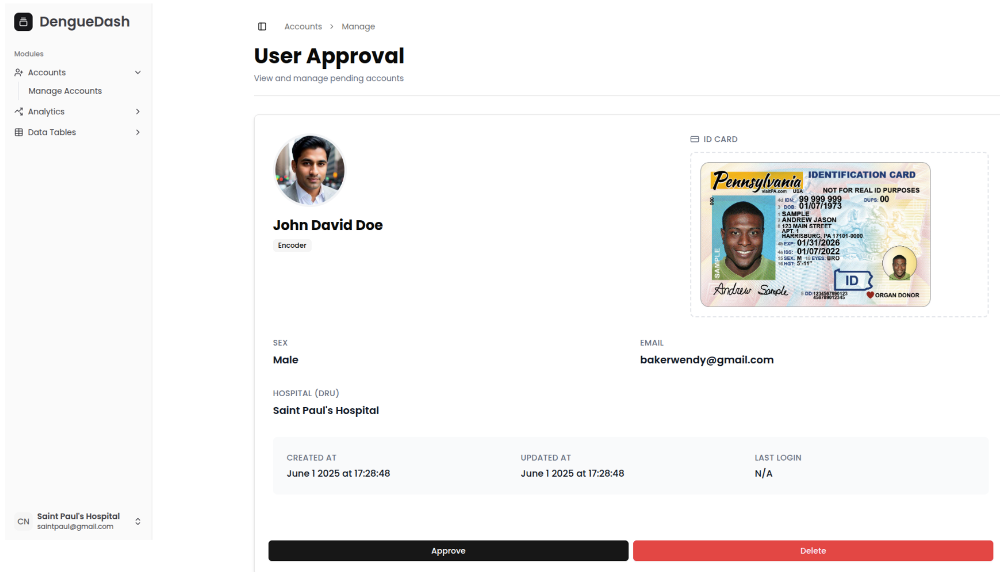


Figure 4.26: Encoder Approval Page

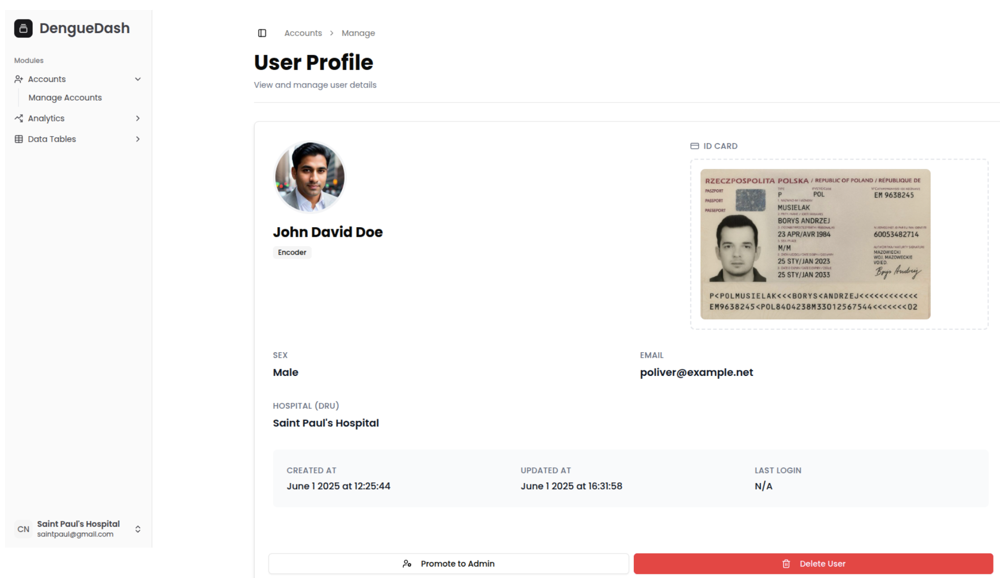


Figure 4.27: Account Management

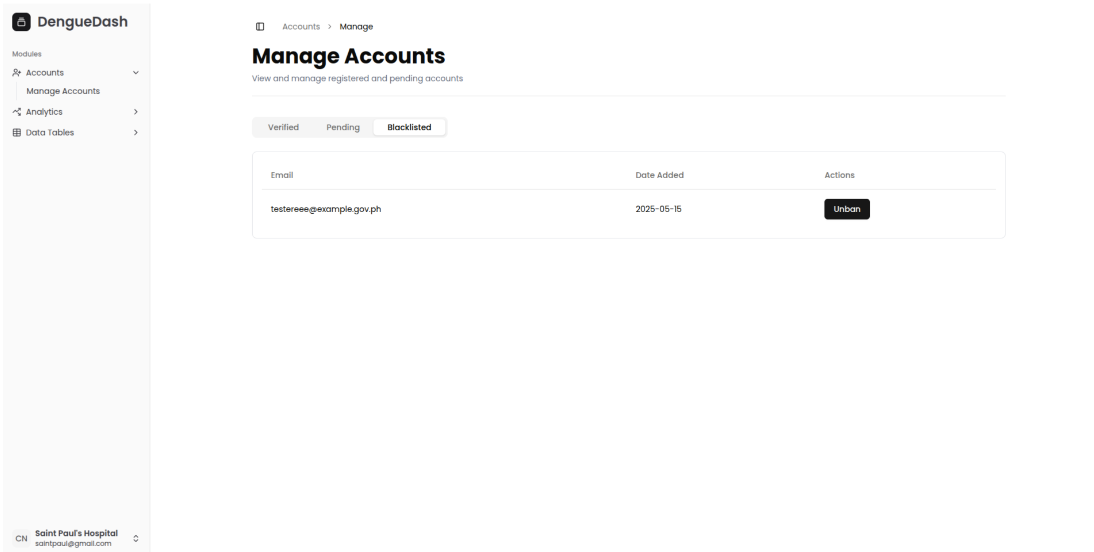


Figure 4.28: List of Blacklisted Accounts

Managing DRUs

Unlike the registration of encoder accounts, the creation of Disease Reporting Units can only be done within the web application, and the user performing the creation must be an administrator of a surveillance unit. Figure 4.29 presents the fields the admin user must fill out, and once completed, the new entry will show as being managed by that unit, as shown in Figure 4.30. Figure 4.31, on the other hand, shows the details provided in the registration form as well as its creation details. There is also an option to delete the DRU, and when invoked, all the accounts being managed by it, and the cases reported under those accounts will be soft-deleted.

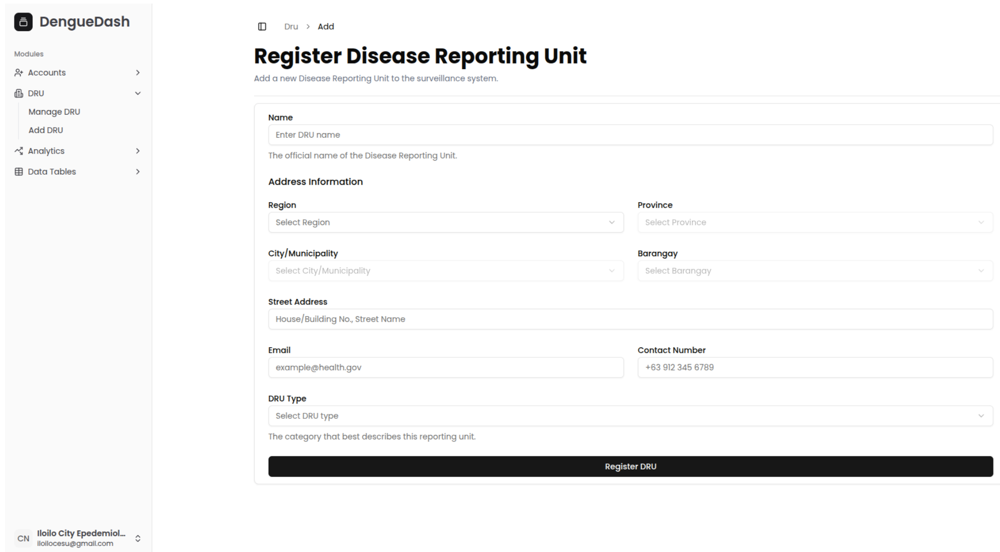


Figure 4.29: Disease Reporting Unit Registration

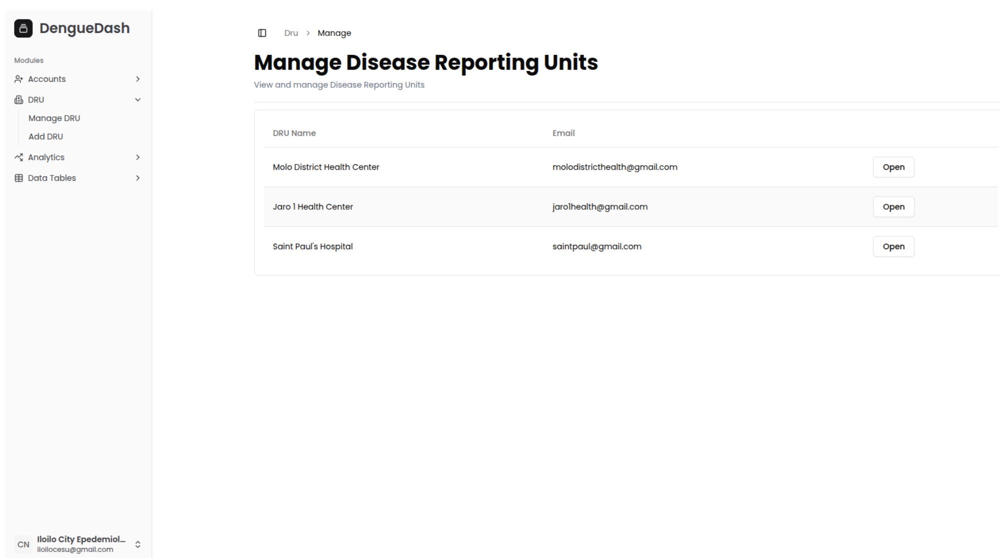


Figure 4.30: List of Disease Reporting Units

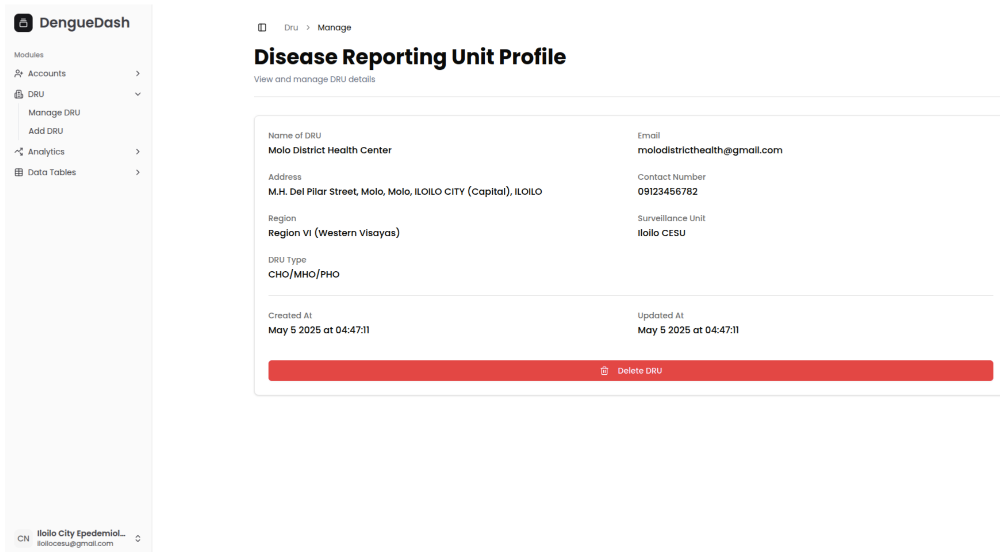


Figure 4.31: Disease Reporting Unit details

4.7 User Testing

To evaluate the usability of the system, the System Usability Scale (SUS) was utilized. SUS is a Likert-scale-based questionnaire comprising 10 items that are critical to assessing system usability. A total of five participants completed the survey. Their responses were processed following the step-by-step calculation method adopted from (Babich, n.d.). The resulting usability scores for each participant are shown in Table 4.9.

The average System Usability Scale (SUS) score across systems is typically 68 (Babich, n.d.). In this testing, the system achieved an average SUS score of 88.5, indicating a highly positive user experience. This score suggests that participants found the system not only enjoyable to use but also intuitive enough

Participant	Usability Score
1	95.0
2	90.0
3	85.0
4	87.5
5	85.0
Average	**88.5**

Table 4.9: Computed System Usability Scores per Participant

to recommend to others. Furthermore, it demonstrates that the system is suitable for real-world applications without presenting significant complexity for first-time users.

Chapter 5

Conclusion

The development of DengueWatch marks a transformative leap forward in public health technology, providing Iloilo City with a centralized system to combat one of the most persistent mosquito-borne diseases. Previously, data was recorded manually on paper, making tracking and analysis slow and error-prone. DengueWatch digitizes this process, enabling faster, more accurate monitoring. More than an academic project, DengueWatch serves as a practical solution aimed at shifting the approach from reactive outbreak response to proactive prevention. By combining deep learning models with real-time climate data integration, the system achieves a level of accuracy and usability that makes it viable for real-world deployment.

At the heart of DengueWatch is a Long Short-Term Memory (LSTM) neural network, which outperformed traditional forecasting models such as ARIMA and Kalman Filter. The LSTM model achieved a Root Mean Square Error (RMSE) of 20.15, followed by the hybrid KF-LSTM model with an RMSE of 25.56, demonstrating a substantial improvement in predictive capability. Consequently, the LSTM model was selected for integration into the DengueWatch system. Retraining the model monthly strikes a balance between maintaining accuracy and managing computational costs. It allows the model to incorporate new trends from the latest four weeks of data and aligns with the typical monthly data release schedule of provincial health offices.

Usability testing further underscored DengueWatch's readiness for real-world deployment. The system achieved an average System Usability Scale (SUS) score of 88.5, significantly above the industry benchmark of 68. This indicates that users found the system intuitive, efficient, and suitable for operational use in public health contexts. Key features such as a user-friendly dashboard and a two-week forecasting window ensure that the system supports timely, effective responses.

Beyond its immediate application in Iloilo City, the framework behind DengueWatch holds the potential for broader impact. With minor adaptations, it can be scaled nationally through integration with Department of Health surveillance systems.

Chapter 6

References

About GitHub and Git - GitHub Docs. (n.d.). Retrieved from https://docs.github.com/en/get-started/start-your-journey/about-github-and-git

Ahmadini, A. A. H., Naeem, M., Aamir, M., Dewan, R., Alshqaq, S. S. A., & Mashwani, W. K. (2021). Analysis and Forecast of the Number of Deaths, Recovered Cases, and Confirmed Cases from COVID-19 for the Top Four Affected Countries Using Kalman Filter. *Frontiers in Physics*, *9*, 629320.

Arroyo-Marioli, F., Bullano, F., Kucinskas, S., & Rondón-Moreno, C. (2021). Tracking R of COVID-19: A New Real-Time Estimation Using the Kalman Filter. *PLOS ONE*, *16*(1), e0244474.

Babich, N. (n.d.). *How to Use the System Usability Scale (SUS) to Evaluate the Usability of Your Website, year=2015.* Usability Geek. Retrieved from https://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website/ (Accessed: 2025-04-26)

Bosano, R. (2023). *WHO: PH Most Affected by Dengue in Western Pacific.* Retrieved Use the date of access, from https://news.abs-cbn.com/spotlight/12/22/23/who-ph-most-affected-by-dengue-in-western-pacific

Brady, O. J., Smith, D. L., Scott, T. W., & Hay, S. I. (2015). Dengue Disease Outbreak Definitions Are Implicitly Variable. *Epidemics*, *11*, 92–102.

Bravo, L., Roque, V. G., Brett, J., Dizon, R., & L'Azou, M. (2014). Epidemiology of Dengue Disease in the Philippines (2000–2011): A Systematic Literature Review. *PLOS Neglected Tropical Diseases*, *8*(11), e3027.

Carvajal, T. M., Viacrusis, K. M., Hernandez, L. F. T., Ho, H. T., Amalin, D. M., & Watanabe, K. (2018). Machine Learning Methods Reveal the Temporal Pattern of Dengue Incidence Using Meteorological Factors in Metropolitan Manila, Philippines. *BMC Infectious Diseases*, *18*, 1–15.

Chart.js. (n.d.). Retrieved from https://www.chartjs.org/

Cheong, Y. L., Burkart, K., Leitão, P. J., & Lakes, T. (2013). Assessing Weather Effects on Dengue Disease in Malaysia. *International Journal of Environmental Research and Public Health*, *10*(12), 6319–6334.

Christie, T. (n.d.). *Home - Django REST framework.* Retrieved from https://www.django-rest-framework.org/

Colón-González, F. J., Fezzi, C., Lake, I. R., & Hunter, P. R. (2013). The Effects of Weather and Climate Change on Dengue. *PLOS Neglected Tropical Diseases*, *7*(11), e2503.

Hemisphere, N. (2015). Update on the Dengue Situation in the Western Pacific Region. *Update*.

Hii, Y. L., Zhu, H., Ng, N., Ng, L. C., & Rocklöv, J. (2012). Forecast of Dengue Incidence Using Temperature and Rainfall. *PLOS Neglected Tropical Dis-*

eases, *6*(11), e1908.

Joel, C. (2021, 10). *6 reasons to use Tailwind over traditional CSS.* Retrieved from https://dev.to/charliejoel/6-reasons-to-use-tailwind-over-traditional-css-1nc3

Leaflet — an open-source JavaScript library for interactive maps. (n.d.). Retrieved from https://leafletjs.com/

Li, X., Feng, S., Hou, N., Li, H., Zhang, S., Jian, Z., & Zi, Q. (2022). Applications of Kalman Filtering in Time Series Prediction. In *International conference on intelligent robotics and applications* (pp. 520–531).

Ligue, K. D. B., & Ligue, K. J. B. (2022). Deep Learning Approach to Forecasting Dengue Cases in Davao City Using Long Short-Term Memory (LSTM). *Philippine Journal of Science*, *151*(3).

Perla. (2024). *Iloilo Beefs Up Efforts Amid Hike in Dengue Cases.* Retrieved Use the date of access, from https://www.pna.gov.ph/articles/1231208

RabDashDC. (2024). *RabDash DC.* Retrieved Use the date of access, from https://rabdash.com

Runge-Ranzinger, S., Kroeger, A., Olliaro, P., McCall, P. J., Sánchez Tejeda, G., Lloyd, L. S., ... Coelho, G. (2016). Dengue Contingency Planning: From Research to Policy and Practice. *PLOS Neglected Tropical Diseases*, *10*(9), e0004916.

Shadcn. (n.d.). *Introduction.* Retrieved from https://ui.shadcn.com/docs

Tailwind CSS - Rapidly build modern websites without ever leaving your HTML. (n.d.). Retrieved from https://tailwindcss.com/

Watts, David M and Burke, Donald S and Harrison, Bruce A and Whitmire, Ralph E and Nisalak, Ananda. (2020). Effect of temperature on the transmission of dengue virus by *aedes aegypti*. *The American Journal of Tropical*

Medicine and Hygiene, *36*(1), 143–152.

What is Postman? Postman API Platform. (n.d.). Retrieved from `https://www.postman.com/product/what-is-postman/`

Why Visual Studio Code? (2021, 11). Retrieved from `https://code.visualstudio.com/docs/editor/whyvscode`

World Health Organization (WHO). (2018). Dengue and severe dengue in the philippines. *WHO Dengue Factsheet*. (Available at: `https://www.who.int`)

Zhou, S., & Malani, P. (2024). What Is Dengue? *JAMA*, *332*(10), 850–850.

Zod. (n.d.). *TypeScript-First Schema Validation with Static Type Inference.* Retrieved from `https://zod.dev/?id=introduction` (Accessed: 2025-04-26)

Appendix A

Data Collection Snippets

Time	Rainfall	MaxTemperature	AverageTemperature	MinTemperature	Wind	Humidity	Cases
2011-w1	9.938571429	29.4444	25.88889	23.8889	11.39	86.24285714	5
2011-w2	8.587142857	30	26.70555556	24.44444444	7.32	88.02857143	4
2011-w3	5.338571429	30	26.61666667	25	7.55	84.02857143	2
2011-w4	5.41	30.55555556	26.48333333	20.55555556	10.67	80.97142857	7
2011-w5	2.914285714	28.33333333	25.28333333	18.65	11.01	74.88571429	2
2011-w6	3.44	31.11111111	27.53333333	25	3.69	85.27142857	5
2011-w7	1.444285714	30	27.06111111	25	6.69	82.1	6
2011-w8	0.7685714286	30.55555556	27.03333333	25	7.82	76.18571429	2
2011-w9	4.064285714	30.55555556	26.75	24.44444444	6.64	84.1	5
2011-w10	1.252857143	31.11111111	27.7	25.55555556	7.92	80.18571429	4
2011-w11	7.068571429	28.33333333	26.03888889	23.88888889	10.35	85.7	2
2011-w12	3.567142857	30.55555556	26.4	24.44444444	9.57	87.74285714	3
2011-w13	2.745714286	30	26.61666667	25	10.05	86.25714286	2
2011-w14	0.09285714286	30	27.44444444	25	8.25	76.32857143	4
2011-w15	1.008571429	31.11111111	26.89444444	22.77777778	7.93	80.64285714	5
2011-w16	0.8771428571	31.66666667	28.43333333	24.44444444	4.76	80.35714286	3
2011-w17	2.15	32.77777778	28.87222222	25.55555556	5.54	81.77142857	5
2011-w18	3.062857143	32.77777778	28.62777778	25	5.15	81.9	4
2011-w19	9.828571429	32.77777778	27.97222222	25	4.18	85.77142857	5
2011-w20	2.848571429	33.33333333	29.71111111	25.55555556	3.77	79.65714286	3
2011-w21	11.74142857	32.77777778	28.51111111	25	4.28	81.81428571	3
2011-w22	14.52857143	32.77777778	28.47222222	25.55555556	4.04	82.45714286	4
2011-w23	11.51714286	32.22222222	28.26111111	24.44444444	3.91	84.24285714	9
2011-w24	8.784285714	32.22222222	27.71111111	24.44444444	3.87	84.7	3
2011-w25	10.43714286	32.77777778	27.47777778	24.44444444	4.46	86.52857143	13
2011-w26	6.702857143	32.22222222	28.39444444	25	3.94	81.87142857	7
2011-w27	14.48	32.22222222	27.66666667	25	4.19	85.42857143	13
2011-w28	3.32	32.22222222	27.49444444	19.68333333	6.69	80.78571429	9
2011-w29	17.33714286	32.77777778	27.95555556	25	3.57	80.05714286	10
2011-w30	22.77	31.66666667	26.63888889	24.44444444	5.13	88.01428571	16
2011-w31	9.875714286	32.77777778	27.81111111	25	4.66	83.11428571	13

Figure A.1: Snippet of Consolidated Data

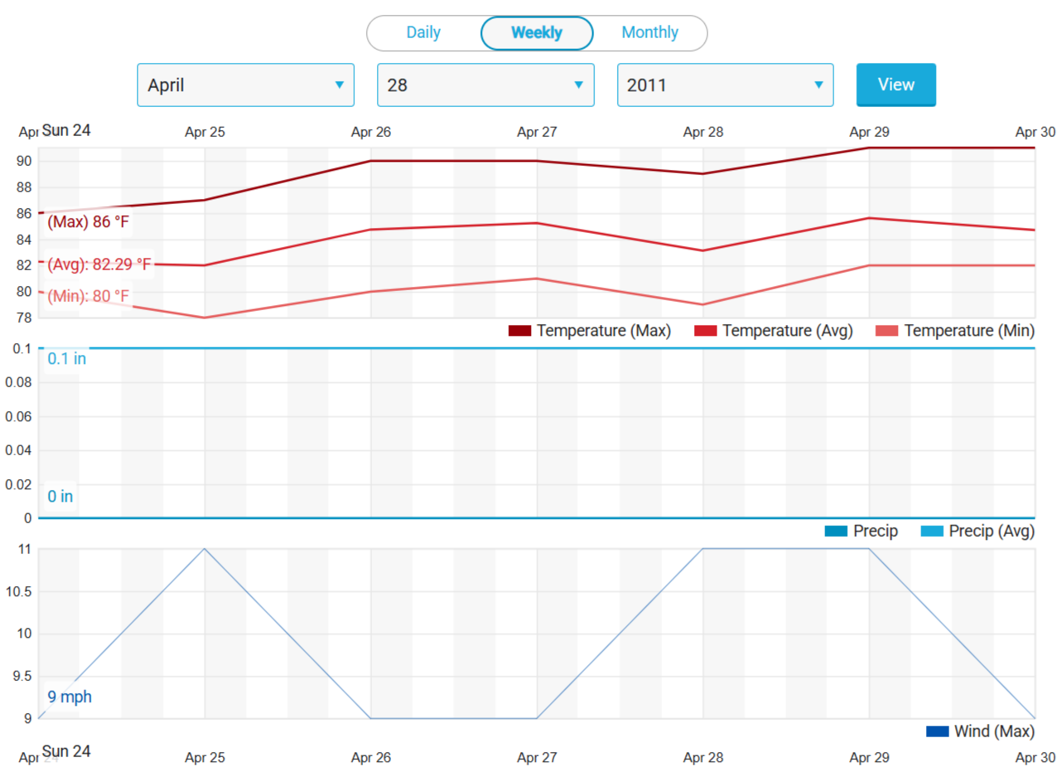


Figure A.2: Snippet of Weather Data Collection

UPV HSU RECEIVED
Date: 22 2025
Received by:

DENGUE WATCH - Smarter Forecasting for Healthier Communities

Dear HSU,

We are a group of students working on our Special Problem project, which is focused on addressing one of the most pressing public health challenges in our communities: dengue outbreaks.

We developed DENGUE WATCH, a web-based system designed to forecast and monitor dengue cases in real time to support early action and informed decision-making. Our system leverages cutting-edge Long Short-Term Memory (LSTM) models to predict future dengue trends based on historical cases and environmental factors such as rainfall, temperature, and humidity.

Our mission is to create a practical, data-driven tool that helps healthcare workers like you respond faster and more effectively to potential dengue surges.

We are currently in the user testing phase, and we would be honored to invite you to try out DENGUE WATCH and share your insights. Your frontline experience and professional input are invaluable in making this tool as useful, accurate, and user-friendly as possible

By participating in testing, you'll help us:

- Evaluate how intuitive and helpful the system is in a healthcare setting
- Improve data presentation and usability for real-world tasks
- Validate our forecasting features from a practical standpoint

With your help, we can enhance DENGUE WATCH into a system that truly supports community health and disease prevention efforts.

Sincerely,

Carl Benedict Elipan
Glen Andrew Buloang
Kurt Matthew Amodia

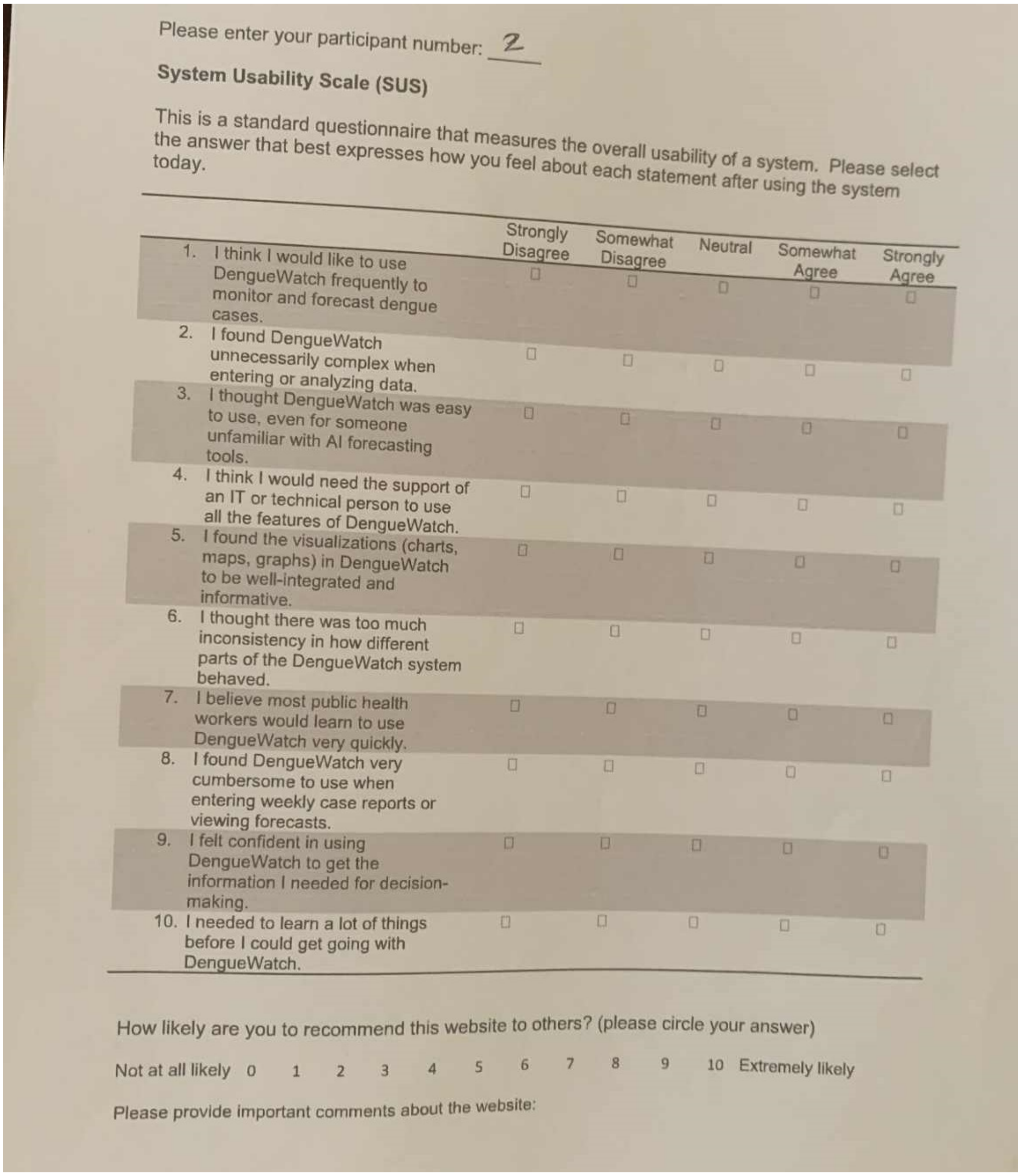
clelipan@up.edu.ph
gcbulaong@up.edu.ph
kmamodia@up.edu.ph

Invitation to try out Dengue Watch is accepted. Pls. keep in touch with the doctor or nurse-on-duty when there is available time (not attending to patients).

, MD. 04-22-2025

MA. CECILIA C. VILLARUZ, M.D.
Chief, UPV - Health Service Unit

Figure A.3: Letter of Approval for User Testing in UPV HSU



Please enter your participant number: 2

System Usability Scale (SUS)

This is a standard questionnaire that measures the overall usability of a system. Please select the answer that best expresses how you feel about each statement after using the system today.

	Strongly Disagree	Somewhat Disagree	Neutral	Somewhat Agree	Strongly Agree
1. I think I would like to use DengueWatch frequently to monitor and forecast dengue cases.	☐	☐	☐	☐	☐
2. I found DengueWatch unnecessarily complex when entering or analyzing data.	☐	☐	☐	☐	☐
3. I thought DengueWatch was easy to use, even for someone unfamiliar with AI forecasting tools.	☐	☐	☐	☐	☐
4. I think I would need the support of an IT or technical person to use all the features of DengueWatch.	☐	☐	☐	☐	☐
5. I found the visualizations (charts, maps, graphs) in DengueWatch to be well-integrated and informative.	☐	☐	☐	☐	☐
6. I thought there was too much inconsistency in how different parts of the DengueWatch system behaved.	☐	☐	☐	☐	☐
7. I believe most public health workers would learn to use DengueWatch very quickly.	☐	☐	☐	☐	☐
8. I found DengueWatch very cumbersome to use when entering weekly case reports or viewing forecasts.	☐	☐	☐	☐	☐
9. I felt confident in using DengueWatch to get the information I needed for decision-making.	☐	☐	☐	☐	☐
10. I needed to learn a lot of things before I could get going with DengueWatch.	☐	☐	☐	☐	☐

How likely are you to recommend this website to others? (please circle your answer)

Not at all likely 0 1 2 3 4 5 6 7 8 9 10 Extremely likely

Please provide important comments about the website:

Figure A.4: System Usability Score Questionnaire