

1 DENGUEWATCH: A SYSTEM FOR REAL-TIME  
2 DENGUE MONITORING AND FORECASTING IN ILOILO  
3 PROVINCE

4 A Special Problem Proposal  
5 Presented to  
6 the Faculty of the Division of Physical Sciences and Mathematics  
7 College of Arts and Sciences  
8 University of the Philippines Visayas  
9 Miag-ao, Iloilo

10 In Partial Fulfillment  
11 of the Requirements for the Degree of  
12 Bachelor of Science in Computer Science by

13 AMODIA, Kurt Matthew A.  
14 BULAONG, Glen Andrew C.  
15 ELIPAN, Carl Benedict L.

16 Francis D. DIMZON  
17 Adviser

18 January 23, 2025

## Abstract

Dengue fever remains a significant public health concern in the Philippines, with cases rising dramatically in recent years. Nationwide outbreaks have placed immense strain on healthcare systems, underscoring the need for innovative approaches to surveillance and response. In Iloilo City, this national trend is reflected in a significant surge, with the Iloilo Provincial Health Office reporting 4,585 cases and 10 fatalities as of August 10, 2023—a 319% increase from the previous year’s 1,095 cases and one death. This research focuses on developing a centralized system for monitoring and forecasting dengue trends in Iloilo City, incorporating graphical visualizations such as heatmaps, trends, and historical graphs. The study explores the application of artificial intelligence (AI) in dengue prediction, utilizing deep learning models. The performance of the Long Short-Term Memory (LSTM) model is compared with traditional statistical methods, including non-seasonal and seasonal Autoregressive Integrated Moving Average (ARIMA) models and the Kalman Filter. Evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Square Error (RMSE) were used to identify the most effective model for integration into the system. Forecasting is based on climate variables such as temperature, rainfall, relative humidity, and previous monthly case counts. The LSTM model emerged as the best performer with an RMSE of 16.66, demonstrating its suitability for time-series predictions, while the Kalman Filter showed the poorest performance with an RMSE of 38.40. By integrating predictive analytics with real-time data visualization, the proposed system aims to support public health agencies, such as the Department of Health (DOH), by providing actionable insights for proactive intervention strategies. This AI-driven solution enhances traditional outbreak reporting systems by enabling timely, data-informed decisions to mitigate the impact of dengue in the region.

**Keywords:** ARIMA, artificial intelligence, dengue prediction, LSTM, Kalman Filter, deep learning, climate variables, public health, outbreak mitigation

# Contents

48	<b>1 Introduction</b>	<b>1</b>
49	1.1 Overview . . . . .	1
50	1.2 Problem Statement . . . . .	2
51	1.3 Research Objectives . . . . .	2
52	1.3.1 General Objective . . . . .	2
53	1.3.2 Specific Objectives . . . . .	2
54	1.4 Scope and Limitations of the Research . . . . .	3
55	1.5 Significance of the Research . . . . .	4
56	<b>2 Review of Related Literature</b>	<b>6</b>
57	2.1 Dengue . . . . .	6
58	2.2 Existing System: RabDash DC . . . . .	6
59	2.3 Deep Learning . . . . .	7
60	2.4 Kalman Filter . . . . .	7
61	2.5 Weather Data . . . . .	8
62	2.6 Chapter Summary . . . . .	9
63	<b>3 Research Methodology</b>	<b>10</b>

64	3.1	Research Activities . . . . .	11
65	3.1.1	Gather Dengue Data and Climate Data to Create a Com- plete Dataset for Forecasting . . . . .	11
66			
67	3.1.2	Develop and Evaluate Deep Learning Models for Dengue Case Forecasting . . . . .	12
68			
69	3.1.3	Integrate the Predictive Model into a Web-Based Data An- alytics Dashboard . . . . .	16
70			
71	3.1.4	System Development Framework . . . . .	16
72	3.1.5	Design, Building, Testing, and Integration . . . . .	16
73	3.2	Development Tools . . . . .	17
74	3.2.1	Software . . . . .	17
75	3.2.2	Hardware . . . . .	18
76	3.2.3	Packages . . . . .	19
77	3.3	Calendar of Activities . . . . .	21
78	<b>4</b>	<b>Preliminary Results/System Prototype</b>	<b>22</b>
79	4.1	Data Gathering . . . . .	22
80	4.2	Exploratory Data Analysis . . . . .	23
81	4.3	Model Training . . . . .	25
82	4.3.1	LSTM Model . . . . .	26
83	4.3.2	ARIMA Model . . . . .	28
84	4.3.3	Kalman Filter Model . . . . .	31
85	4.4	Preliminary System Requirements . . . . .	34
86	4.4.1	Backend Requirements . . . . .	34
87	4.4.2	Security and Validation Requirements . . . . .	34

88	4.5	System Prototype . . . . .	35
89	4.5.1	Guest Interface . . . . .	35
90	4.5.2	Personnel Interface . . . . .	36
91		<b>References</b>	<b>41</b>
92		<b>A Appendix Title</b>	<b>43</b>
93		<b>B Resource Persons</b>	<b>44</b>

# 94 List of Figures

95	3.1	Workflow for forecasting the number of weekly dengue cases . . .	10
96	4.1	Snippet of the Combined Dataset . . . . .	23
97	4.2	Trend of Dengue Cases . . . . .	23
98	4.3	Correlation Heatmap . . . . .	24
99	4.4	Ranking of Correlations . . . . .	24
100	4.5	Ranking of Correlations (with lagged effects) . . . . .	25
101	4.6	Comparison of Window Sizes . . . . .	27
102	4.7	ARIMA Prediction Results for Test Set . . . . .	28
103	4.8	Seasonal ARIMA Prediction Results for Test Set . . . . .	30
104	4.9	Kalman Filter Prediction Results for Test Set . . . . .	31
105	4.10	Entity-Relationship Database Schema Hybrid Diagram for DengueDash	
106		Database Structure . . . . .	34
107	4.11	Dashboard for Guests . . . . .	36
108	4.12	Login Page for Users . . . . .	37
109	4.13	First Part of Case Report Form . . . . .	38
110	4.14	Second Part of Case Report Form . . . . .	38
111	4.15	Dengue Reports . . . . .	39

112	4.16 Detailed Case Report . . . . .	40
113	A.1 LSTM Prediction Results for Test Set . . . . .	43

# 114 List of Tables

<small>115</small>	3.1	Timetable of Activities for 2024 . . . . .	21
<small>116</small>	3.2	Timetable of Activities for 2025 . . . . .	21
<small>117</small>	4.1	Comparison of Models . . . . .	26
<small>118</small>	4.2	Comparison of Window Sizes . . . . .	27



# Chapter 1

## Introduction

### 1.1 Overview

From 2020 to 2022, dengue cases declined due to reduced surveillance during the COVID-19 pandemic (WHO, 2023), but cases surged in 2023 as restrictions were lifted. This year saw an increase in dengue outbreaks worldwide, with over five million cases and more than 5,000 deaths reported in over 80 countries (Bosano, 2023). Dengue is endemic in the Philippines, leading to longer and more widespread seasonal outbreaks. Globally, dengue infections have increased significantly, posing a major public health challenge. The World Health Organization reported a ten-fold rise in cases between 2000 and 2019, with a peak in 2019 when the disease spread across 129 countries (WHO, 2024).

Iloilo City and Province are intensifying efforts to curb the rising dengue cases (Lena, 2024). As of August 10, 2023, the Iloilo Provincial Health Office recorded 4,585 cases and 10 deaths, a 319% increase from last year's 1,095 cases and one death. Governor Arthur Defensor Jr. confirmed that the province has reached the dengue outbreak threshold based on Department of Health (DOH). Local government units (LGUs) have been informed, and the province's disaster management office is on blue alert, indicating disaster mode. (Perla, 2024)

In Iloilo City, 649 dengue cases were recorded this year 2024, with two deaths. Cases cluster in 40 out of 180 barangays, meaning multiple cases are being reported in these areas over several weeks. The city's health officer, Dr. Roland Jay Fortuna, reported high utilization of non-COVID-19 hospital beds, reaching over 76%, prompting concerns about hospital capacity.

143 This study explores the monitoring and forecasting of dengue outbreaks by an-  
144 alyzing key factors such as temperature, relative humidity, and historical dengue  
145 cases, using different models. The findings aim to provide an advanced, AI-driven  
146 alternative for dengue prevention and control, targeting agencies like the Depart-  
147 ment of Health (DOH). By aligning with the national AI Roadmap, particularly  
148 in Iloilo City, this research aspires to improve outbreak responses through cutting-  
149 edge technology rather than traditional reporting methods.

## 150 1.2 Problem Statement

151 Dengue remains a critical public health challenge worldwide, with cases increasing  
152 due to the easing of COVID-19 restrictions and heightened global mobility. While  
153 a temporary decline in cases was observed during the pandemic (2020–2022) due  
154 to reduced surveillance efforts, 2023 marked a resurgence, with over five million  
155 cases and more than 5,000 deaths reported across 80 countries. In dengue-endemic  
156 regions like the Philippines, the threat is particularly severe. In Iloilo City and  
157 Province, dengue cases rose by 319% as of August 2023, overwhelming local health-  
158 care systems. This surge strained resources, with over 76% of non-COVID-19 hos-  
159 pital beds occupied by dengue patients, highlighting the urgent need for effective  
160 predictive tools. The lack of a reliable system to monitor and forecast dengue  
161 outbreaks contributes to delayed interventions, exacerbating public health risks  
162 and healthcare burdens in the region.

## 163 1.3 Research Objectives

### 164 1.3.1 General Objective

165 This study aims to develop an AI-based dengue forecasting and monitoring system  
166 for Iloilo City and Province. The researchers will train and compare multiple deep  
167 learning models to predict dengue case trends based on climate data and historical  
168 dengue cases to help public health officials in possible dengue case outbreaks.

### 169 1.3.2 Specific Objectives

170 Specifically, this study aims to:

- 171 1. Gather dengue data from the Iloilo Provincial Health Office and climate data  
172 (including temperature, rainfall, wind, and humidity) from online sources.  
173 Combine and aggregate these data into a unified dataset to facilitate com-  
174 prehensive dengue case forecasting.
- 175 2. Evaluate deep learning models for predicting dengue cases using metrics  
176 such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE),  
177 and Mean Squared Error (MSE). Compare the performance of these models  
178 to determine the most accurate forecasting approach.
- 179 3. Develop a web-based analytics dashboard that integrates a predictive model  
180 and provides data management system for dengue cases in Iloilo City and  
181 the Province.
- 182 4. Assess the usability and effectiveness of the analytics dashboard through  
183 structured feedback and surveys involving health professionals and policy-  
184 makers.

## 185 1.4 Scope and Limitations of the Research

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

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

201 The study also involves developing a web-based analytics dashboard that in-  
202 tegrates predictive models and provides a data management system for dengue  
203 cases in Iloilo City and the Province. This dashboard will offer public health  
204 officials an interactive interface to visualize dengue trends, input new data, and

205 identify risk areas. However, its usability depends on feedback from stakeholders,  
206 which may vary based on their familiarity with analytics tools. Moreover, exter-  
207 nal factors such as limited internet connectivity or device availability in remote  
208 areas may affect the system’s adoption and effectiveness. While the dashboard  
209 provides valuable insights, it cannot incorporate all factors influencing dengue  
210 transmission, emphasizing the need for ongoing validation and refinement.

## 211 1.5 Significance of the Research

212 This study’s development of an AI-based dengue forecasting and monitoring sys-  
213 tem has wide-reaching significance for various stakeholders in Iloilo City:

- 214 • **Public Health Agencies:** Organizations like the Department of Health (DOH)  
215 and local health units in Iloilo City and Province stand to benefit greatly  
216 from the system. With dengue predictions, we can help these agencies opti-  
217 mize their response strategies and implement targeted prevention measures  
218 in high-risk areas before cases escalate.
- 219 • **Local Government Units (LGUs):** LGUs can use the system to support  
220 their disaster management and health initiatives by proactively addressing  
221 dengue outbreaks. The predictive insights allow for more efficient planning  
222 and resource deployment in barangays and communities most vulnerable to  
223 outbreaks, improving overall public health outcomes.
- 224 • **Healthcare Facilities:** Hospitals and clinics, which currently face high bed  
225 occupancy rates during dengue season will benefit from early outbreak fore-  
226 casts that can help in managing patient inflow and ensuring adequate hos-  
227 pital capacity.
- 228 • **Researchers and Policymakers:** This AI-driven approach contributes valu-  
229 able insights for researchers studying infectious disease patterns and policy-  
230 makers focused on strengthening the national AI Roadmap. The system’s  
231 data can support broader initiatives for sustainable health infrastructure  
232 and inform policy decisions on resource allocation for dengue control.
- 233 • **Community Members:** By reducing the frequency and severity of outbreaks,  
234 this study ultimately benefits the community at large. This allows for timely

235 awareness campaigns and community engagement initiatives, empowering  
236 residents with knowledge and preventative measures to protect themselves  
237 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, Burke, Harrison, Whitmire, & Nisalak, 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 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

261 genomic data, enabling local health authorities to optimize interventions and al-  
262 locate resources more effectively. RabDash’s modules include trend visualization,  
263 geographic hotspot mapping, and predictive forecasting, utilizing Long Short-  
264 Term Memory (LSTM) models for time-series forecasting (RabDashDC, 2024).

265 For DengueWatch, RabDash serves as a strong inspiration, particularly in  
266 its monitoring, historical trend visualization, and forecasting capabilities. These  
267 features align well with the needs of dengue control efforts, providing real-time  
268 insights into outbreak trends and enabling more effective, data-driven decision-  
269 making. RabDash’s architecture is relevant to the DengueDash, as dengue out-  
270 breaks similarly require time-series forecasting models. By using LSTM, RabDash  
271 effectively models trends in outbreak data, which provides a framework for adapt-  
272 ing LSTM to dengue forecasting. Research indicates that LSTM models outper-  
273 form traditional methods, such as ARIMA and MLP, in handling the complexities  
274 of time-dependent epidemiological data (Ligue & Ligue, 2022).

## 275 2.3 Deep Learning

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

## 289 2.4 Kalman Filter

290 The Kalman Filter is another powerful tool for time-series forecasting that can be  
291 integrated into our analysis. It provides a recursive solution to estimating the state  
292 of a linear dynamic system from a series of noisy measurements. Its application  
293 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).

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

## 2.5 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, Zhu, Ng, Ng, & Rocklöv, 2012).

We will utilize 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.



## 2.6 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 will be performed to accomplish the project. The discussion covers the activities from pre-proposal to Final SP Writing.

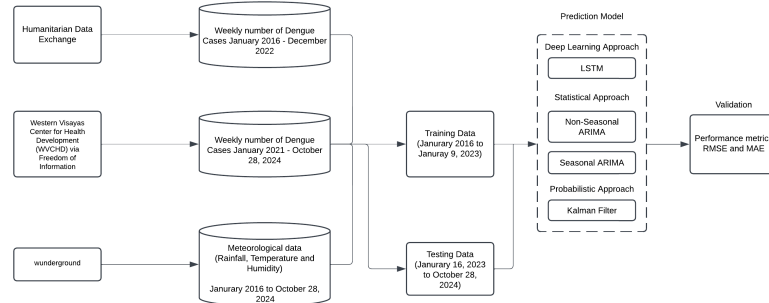


Figure 3.1: Workflow for forecasting the number of weekly dengue cases

This 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.

## 3.1 Research Activities

### 3.1.1 Gather Dengue Data and Climate Data to Create a Complete Dataset for Forecasting

#### 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.

- 384     • **Min Temperature.** Represents the observed minimum temperature, mea-  
385       sured in degrees Celsius, for a specific week.
- 386     • **Wind.** Represents the observed wind speed, measured in miles per hour  
387       (mph), for a specific week.
- 388     • **Cases.** Refers to the number of reported dengue cases during a specific  
389       week.

## 390   **Data Integration and Preprocessing**

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

## 398   **Exploratory Data Analysis (EDA)**

- 399     • Analyze trends, seasonality, and correlations between dengue cases and  
400       weather factors.
- 401     • Create visualizations like time series plots and scatterplots to highlight re-  
402       lationships and patterns in the data.

## 403   **3.1.2   Develop and Evaluate Deep Learning Models for** 404       **Dengue Case Forecasting**

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

## 410 Data Preprocessing

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

## 417 LSTM Model

418 To prepare the data for LSTM, a sliding window approach was utilized. Sequences  
419 of weeks of normalized features were constructed as input, while the dengue case  
420 count for the subsequent week was set as the target variable. This approach en-  
421 sured that the model leveraged temporal dependencies in the data for forecasting.

422 The LSTM model was designed using the TensorFlow and Keras libraries. The  
423 architecture comprised the following layers:

- 424 • Input Layer: Accepting sequences of weeks with three features (rainfall, max  
425 temperature, and humidity).
- 426 • LSTM Layer: A single LSTM layer with 64 units and ReLU activation,  
427 capturing temporal dependencies and feature interactions.
- 428 • Dense Output Layer: A fully connected layer with a single neuron to predict  
429 the dengue cases for the next week.

430 The model was trained for 100 epochs implementing early stopping with a batch  
431 size of 1, enabling fine-grained weight updates. The training dataset consisted  
432 of 80% of the sequences, while the remaining 20% was used as the test set to  
433 evaluate model performance. Validation loss was monitored during training to  
434 assess model generalization.

435 The training process was conducted using three distinct window sizes (5 weeks,  
436 10 weeks, and 20 weeks) to determine the optimal sequence length of weeks to  
437 input into the LSTM model for improved forecasting performance.

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

## 442 Seasonal ARIMA (SARIMA):

443 The SARIMA (Seasonal ARIMA) model was utilized to forecast weekly dengue  
444 cases, incorporating seasonal patterns and exogenous weather variables (rainfall,  
445 max temperature, and humidity). The dataset was divided into training (80%)  
446 and testing (20%) sets while maintaining temporal continuity for validation. The  
447 input data consisted of weekly dengue case counts as the target variable and  
448 weather-related features as exogenous regressors.

449 The SARIMA model's parameters were set as follows:

- 450 • Order: (2, 0, 2)
- 451 • Seasonal Order: (0, 1, 1, 52)

452 The SARIMA model was trained using the training dataset, including exoge-  
453 nous variables. The maximum number of iterations was set to 400 to ensure  
454 convergence during fitting.

455 The model's performance was assessed using regression metrics to evaluate its  
456 forecasting capability:

- 457 • Mean Squared Error (MSE): Quantifies average squared prediction error.
- 458 • Root Mean Squared Error (RMSE): Measures average prediction error on  
459 the data's original scale.

## 460 ARIMA

461 The ARIMA model was employed to forecast weekly dengue cases using historical  
462 weather data (rainfall, max temperature, and humidity) as exogenous variables  
463 and historical case counts as the primary dependent variable. The dataset was  
464 split into training (80%) and testing (20%) sets. To determine the optimal con-  
465 figuration for the ARIMA model, a grid search was conducted over the following  
466 parameter ranges:

- 467 • p (autoregressive order): 0 to 3
- 468 • d (differencing order): 0 to 2
- 469 • q (moving average order): 0 to 3

470 The combinations of these parameters were evaluated by fitting an ARIMA model  
471 for each set of (p, d, q) values. The model's performance was assessed using the  
472 mean squared error (MSE) between the predicted and actual dengue cases in the  
473 test set. The combination yielding the lowest MSE was selected as the optimal  
474 parameter configuration.

475 The fitted ARIMA model was used to forecast weekly dengue cases for the  
476 test dataset. Predictions were directly assigned to the PredictedCases column in  
477 the test dataset. Model performance was evaluated using the following metrics:

- 478 • Mean Squared Error (MSE): Quantifies average squared prediction error.
- 479 • Root Mean Squared Error (RMSE): Measures average prediction error on  
480 the data's original scale.

#### 481 **Kalman Filter:**

- 482 • Input Variables: The target variable (Cases) was modeled using three re-  
483 gressors: rainfall, max temperature, and humidity.
- 484 • Training and Testing Split: The dataset was split into 80% training and  
485 20% testing to evaluate model performance.
- 486 • Observation Matrix: The Kalman Filter requires an observation matrix,  
487 which was constructed by adding an intercept (column of ones) to the re-  
488 gressors.

489 The Kalman Filter's em method was employed for training, iteratively esti-  
490 mating model parameters over 10 iterations. The smooth method was used to  
491 compute the smoothed state estimates for the training data. Observation matri-  
492 ces for the test data were constructed similarly, ensuring compatibility with the  
493 trained model.

#### 494 **Model Evaluation and Optimization**

- 495 • Compare the performance of all models to identify the most accurate fore-  
496 casting approach.
- 497 • Iteratively optimize the selected model.

### 498 **3.1.3 Integrate the Predictive Model into a Web-Based** 499 **Data Analytics Dashboard**

#### 500 **Dashboard Design and Development**

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

#### 505 **Model Integration and Deployment**

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

### 508 **3.1.4 System Development Framework**

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

### 515 **3.1.5 Design, Building, Testing, and Integration**

#### 516 **Design and Development**

517 After brainstorming and researching the most appropriate type of application to  
518 accommodate both the prospected users and the proposed solutions, the team has  
519 decided to proceed with a web application. Given the time constraints and avail-  
520 able resources, we believe this is the most pragmatic and practical move. The next  
521 step is to select modern and stable frameworks that align with the fundamental  
522 ideas we have learned at the university. The template obtained from WVCHD  
523 and Iloilo Provincial Epidemiology and Surveillance Unit was meticulously ana-  
524 lyzed to create use cases and develop a preliminary well-structured database that



525 adheres to the requirements needed to produce a quality application. The said use  
526 cases serve as the basis of general features. Part by part, these are converted into  
527 code, and with the help of selected libraries and packages, it resulted in the de-  
528 sired outcome that may still modified and extended since it is continuously being  
529 developed.

## 530 **Testing and Integration**

531 Each feature will be rigorously user-tested to ensure quality assurance, with par-  
532 ticular emphasis on prerequisite features, as development cannot progress properly  
533 if these fail. Moreover, integration between each feature serves as a pillar for a  
534 cohesive user experience. Presently, we have not been able to use performance  
535 metrics to measure the system’s performance, as developing and connecting the  
536 core features is the utmost priority.

## 537 **3.2 Development Tools**

### 538 **3.2.1 Software**

#### 539 **Github**

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

#### 545 **Visual Studio Code**

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

## 550 Django

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

## 559 Next.js

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

## 566 Postman

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

## 572 3.2.2 Hardware

573 The web application is continuously being developed on laptop computers with  
574 minimum specifications of an 11th-generation Intel i5 CPU and 16 gigabytes of  
575 RAM.

### 576 3.2.3 Packages

#### 577 Django REST Framework

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

#### 582 Leaflet

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

#### 589 Chart.js

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

#### 597 Tailwind CSS

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

## 606 **Shadcn**

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

## 612 **Zod**

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

### 3.3 Calendar of Activities

A Gantt chart showing the schedule of the activities is included below. Each bullet represents approximately one week of activity.

Table 3.1: Timetable of Activities for 2024

Activities	Aug	Sept	Oct	Nov	Dec
Project Initiation and Team Formation	••				
Literature Review and Data Gathering	••	••••			
Data Cleaning and Feature Selection		••		•	•
Creating System Dashboard		••	••••	•	
Analysis and Interpretation of Results			•		•
Documentation	••	••••	••••	••••	••••

Table 3.2: Timetable of Activities for 2025

Activities	Jan	Feb	Mar	Apr	May
Create Admin Dashboard	•	•••			
Integrate the Best Model to the System	•	••••			
Extend Features to Accommodate a National Setting		•	••		
User Testing			••	•	
System Deployment				•••	
Documentation	••	••••	••••	••••	••••

## Chapter 4

# Preliminary Results/System Prototype

### 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 2010 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.

```
data.head()
```

	Time	Rainfall	MaxTemperature	AverageTemperature	MinTemperature	Wind	Humidity	Cases
0	2011-01-03	9.938571	29.444400	25.888890	23.888900	11.39	86.242857	5
1	2011-01-10	8.587143	30.000000	26.705556	24.444444	7.32	88.028571	4
2	2011-01-17	5.338571	30.000000	26.616667	25.000000	7.55	84.028571	2
3	2011-01-24	5.410000	30.555556	26.483333	20.555556	10.67	80.971429	7
4	2011-01-31	2.914286	28.333333	25.283333	18.650000	11.01	74.885714	2

Figure 4.1: Snippet of the Combined Dataset

## 4.2 Exploratory Data Analysis

Figure 4.2 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, 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.

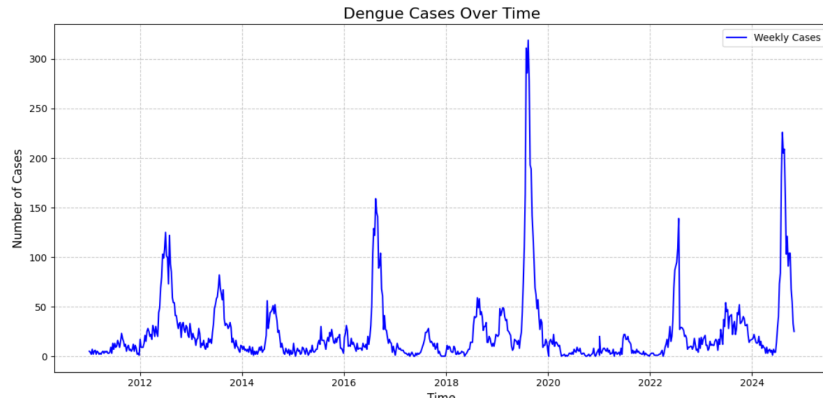


Figure 4.2: Trend of Dengue Cases

Figure 4.3 presents a detailed heatmap showing the correlations among all variables. The heatmap highlights the interdependencies between climatic variables and their respective relationships with dengue cases. Such relationships provide a deeper understanding of how these variables interact and affect dengue case trends, which can guide feature selection for the predictive model.

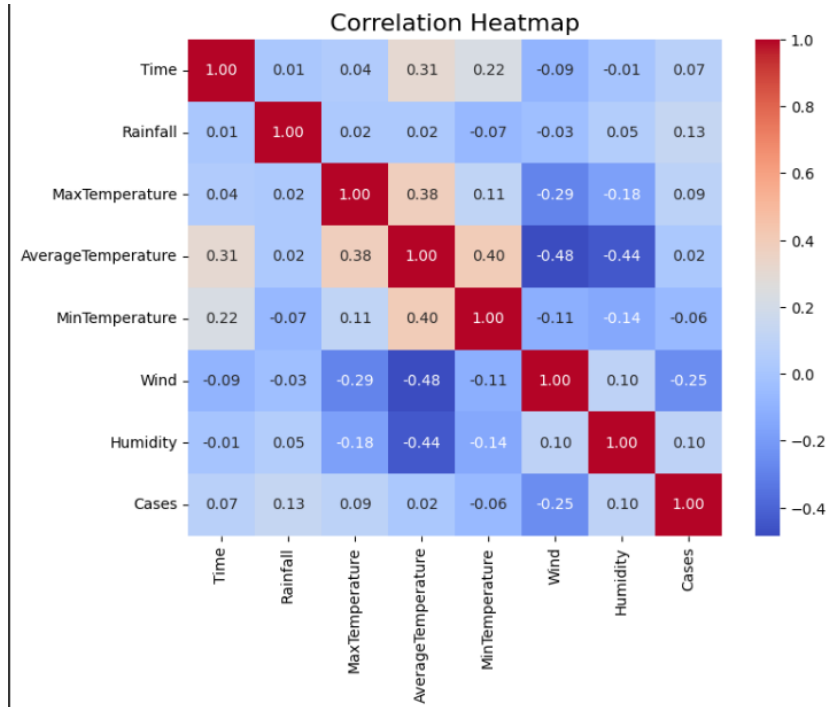


Figure 4.3: Correlation Heatmap

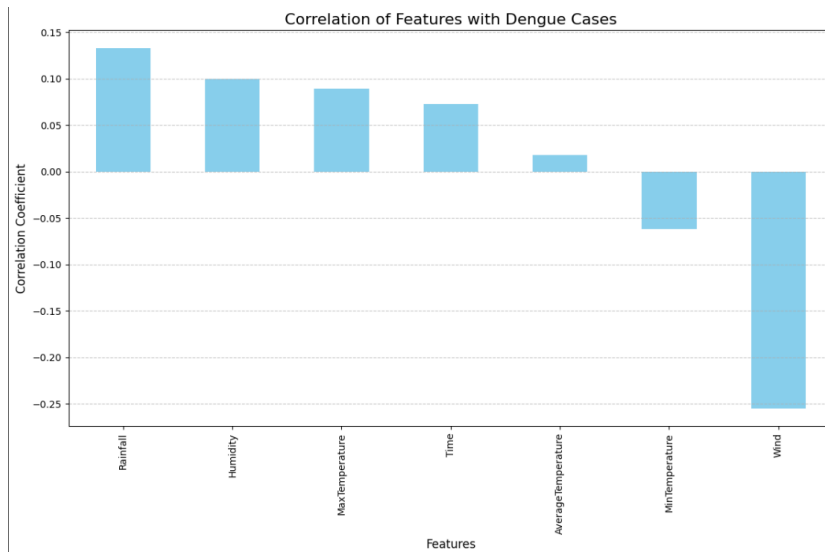


Figure 4.4: Ranking of Correlations

Figure 4.4 shows the ranking of correlation coefficients between dengue cases and selected features, including rainfall, humidity, maximum temperature, average temperature, minimum temperature, and wind speed. Among these, rainfall



653 exhibits the highest positive correlation with dengue cases (correlation coefficient  
654 0.13), followed by humidity (0.10) and maximum temperature (0.09).

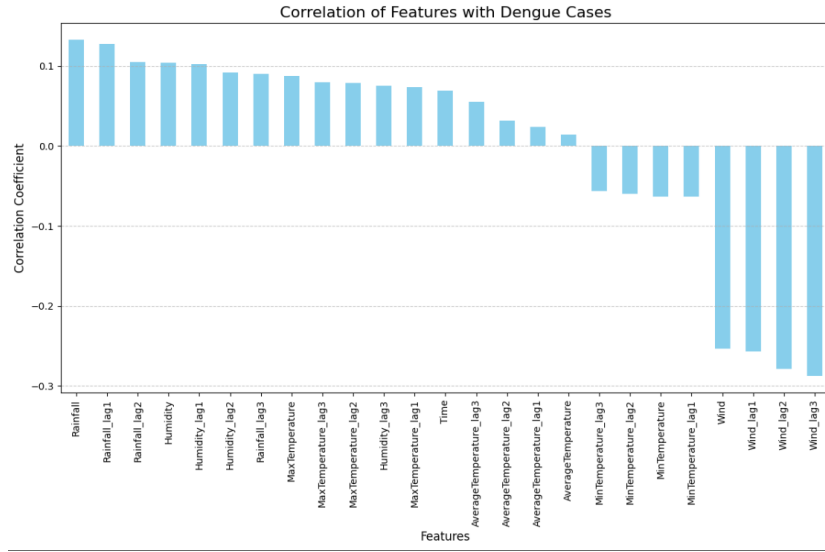


Figure 4.5: Ranking of Correlations (with lagged effects)

655 Figure 4.5 shows the ranking of correlation coefficients between dengue cases  
656 and selected features, with the addition of lagged effects. The analysis reveals no  
657 improvement in correlation when lagged variables are compared to direct observa-  
658 tions. This suggests that the observed values of rainfall, humidity, and maximum  
659 temperature remain the most significant predictors for dengue case forecasting.  
660 Overall, the exploratory data analysis highlights the significance of rainfall, hu-  
661 midity, and max temperature variables in dengue case forecasting.

## 662 4.3 Model Training

663 The proposed Dengue Watch system utilized four distinct models to forecast  
664 weekly dengue cases: Long Short-Term Memory (LSTM) networks, Autoregres-  
665 sive Integrated Moving Average (ARIMA), Seasonal ARIMA (SARIMA), and  
666 Kalman Filter. Each model was trained on a dataset containing 720 weeks of  
667 historical dengue cases from 2010 to 2024, with meteorological variables such as  
668 max temperature, humidity, and rainfall.

669 To optimize predictive performance, hyperparameter tuning was conducted  
670 individually for each model, refining parameters to achieve the most accurate and  
671 reliable forecasts. Following training, the models were rigorously evaluated against

672 the dataset using a set of key performance metrics, including Mean Squared Error  
673 (MSE) and Root Mean Squared Error (RMSE).

674 The table below provides a summary and comparative analysis of each model's  
675 results across these metrics, offering insights into the strengths and limitations of  
676 each forecasting technique for dengue case prediction in Iloilo City.

Model	MSE	RMSE
LSTM	277.71	16.66
Seasonal ARIMA (2, 0, 2) (0, 1,1)	1109.69	33.31
ARIMA (1, 2, 2)	1521.48	39.01
Kalman Filter	1474.82	38.40

Table 4.1: Comparison of Models

### 677 4.3.1 LSTM Model

678 The LSTM model architecture consisted of an input layer, a single LSTM layer  
679 with 64 units and ReLU activation, followed by a dense layer with a single output  
680 neuron to predict the dengue case count. Key hyperparameters included:

- 681 • Window Size: 5, 10, and 20 weeks, representing the time steps used in the  
682 sequence data for each prediction.
- 683 • Epochs: 100 epochs were used for training, balancing sufficient training  
684 time with computational efficiency also implementing early stopping to avoid  
685 overfitting.
- 686 • Batch Size: 1, allowing the model to process one sequence at a time, which  
687 is beneficial for small datasets but increases training time.
- 688 • Optimizer: The Adam optimizer was chosen for its adaptive learning capa-  
689 bilities and stability in training. A custom learning rate of 0.0001 was set  
690 to ensure gradual convergence and minimize risk of overfitting.

691 The dataset was split into training and test sets to evaluate the model's per-  
692 formance and generalizability:

- 693 • **Training Set:** 80% of the data (572 sequences) was used for model training,  
694 enabling the LSTM to learn underlying patterns in historical dengue case  
695 trends and their relationship with weather variables.

696 • **Test Set:** The remaining 20% of the data (148 sequences) was reserved for  
697 testing

698 The training process was conducted using three distinct window sizes—5 weeks,  
699 10 weeks, and 20 weeks—to identify the optimal sequence length of weeks for input  
700 into the LSTM model, thereby enhancing forecasting performance. The following  
701 plots illustrate the performance of the model in predicting dengue cases for each  
702 of the specified window sizes.

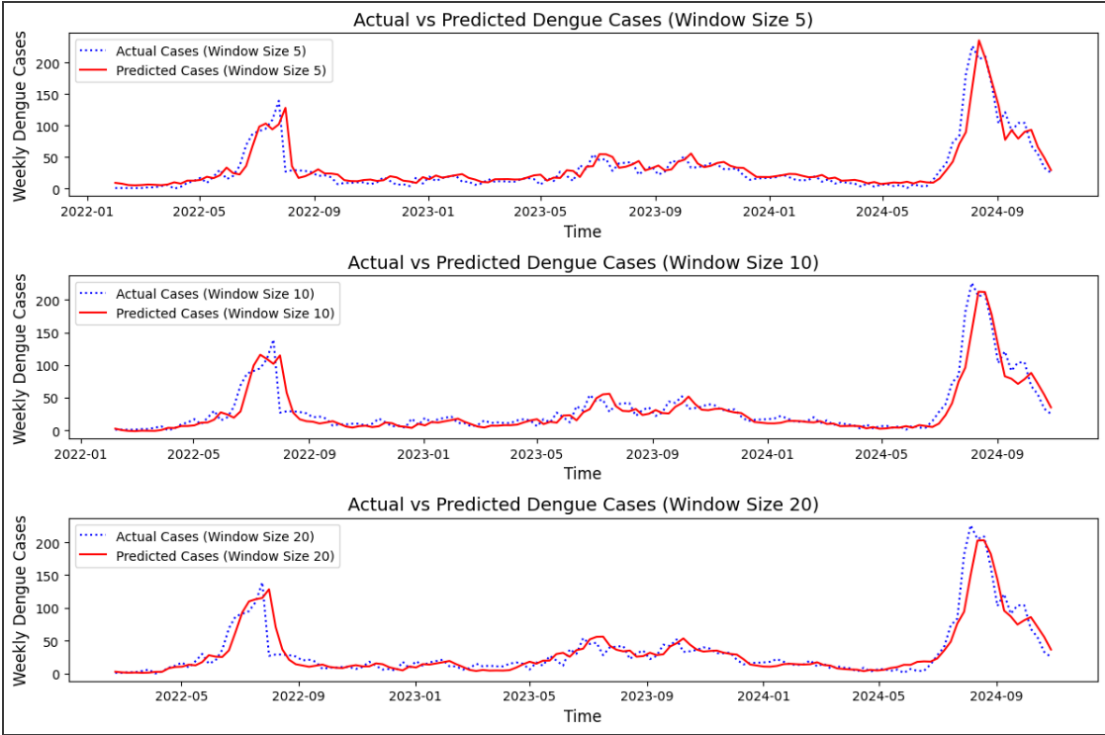


Figure 4.6: Comparison of Window Sizes

703 The evaluation metrics included Mean Squared Error (MSE) and Root Mean  
Squared Error (RMSE), which assess the accuracy of the model's predictions.

Window Size	MSE	RMSE
5	282.69	16.81
10	277.71	16.66
15	289.63	17.02

Table 4.2: Comparison of Window Sizes

704

705 The results indicate that a window size of 10 weeks provides the most accurate  
 706 predictions, as evidenced by the lowest MSE and RMSE values. This suggests that  
 707 using a 10-week sequence length effectively balances the temporal dependencies  
 708 captured by the model and the computational complexity of training.

## 709 Training and Testing Data Division for ARIMA 710 and Seasonal Arima

711 Both models utilized an **80%-20% split** to evaluate generalizability:

- 712 • **Training Set:** 80% of the data was used for training, allowing the models  
 713 to learn underlying patterns in the dataset.
- 714 • **Test Set:** 20% of the data was reserved for testing, providing an unbiased  
 715 assessment of the models' performance on unseen data.

### 716 4.3.2 ARIMA Model

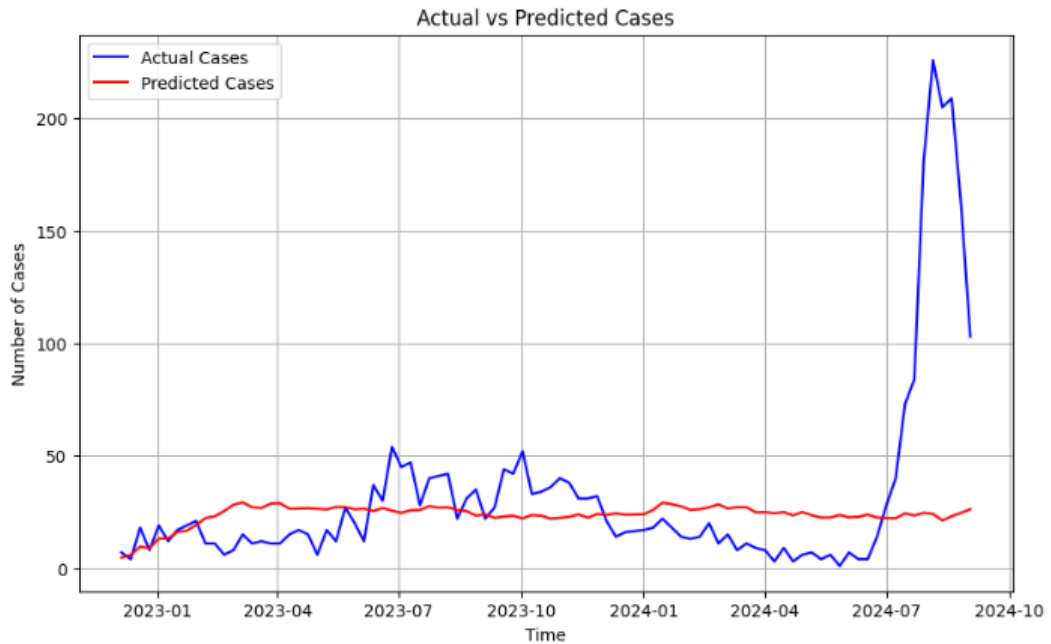


Figure 4.7: ARIMA Prediction Results for Test Set

717 The ARIMA model was developed to capture non-seasonal trends in the data. To  
718 determine the best model configuration, grid search was used to explore various  
719 combinations of ARIMA parameters, ultimately selecting **ARIMA(1, 2, 2)**. The  
720 model was iteratively refined over **400 iterations** to ensure convergence to an  
721 optimal solution. Key details are as follows:

- 722 1. **Data Preprocessing:** Prepare the dataset by handling any missing values  
723 and scaling the data if necessary to improve model convergence and stability.
- 724 2. **Hyperparameter Tuning:** Use a grid search on potential ARIMA param-  
725 eters  $(p, d, q)$  to identify the configuration that minimizes error. The optimal  
726 parameters were found to be **(1, 2, 2)**.
- 727 3. **Model Training:**
  - 728 • Set the number of iterations to 400 to ensure thorough training and  
729 convergence.
  - 730 • Train the ARIMA model on 80% of the data and reserve 20% for test-  
731 ing.
- 732 4. **Evaluation:** After training, the ARIMA model was evaluated on the test  
733 data, yielding the following performance metrics:
  - 734 • **MSE (Mean Squared Error):** 1521.48
  - 735 • **RMSE (Root Mean Squared Error):** 39.01

## 736 Seasonal ARIMA (SARIMA) Model

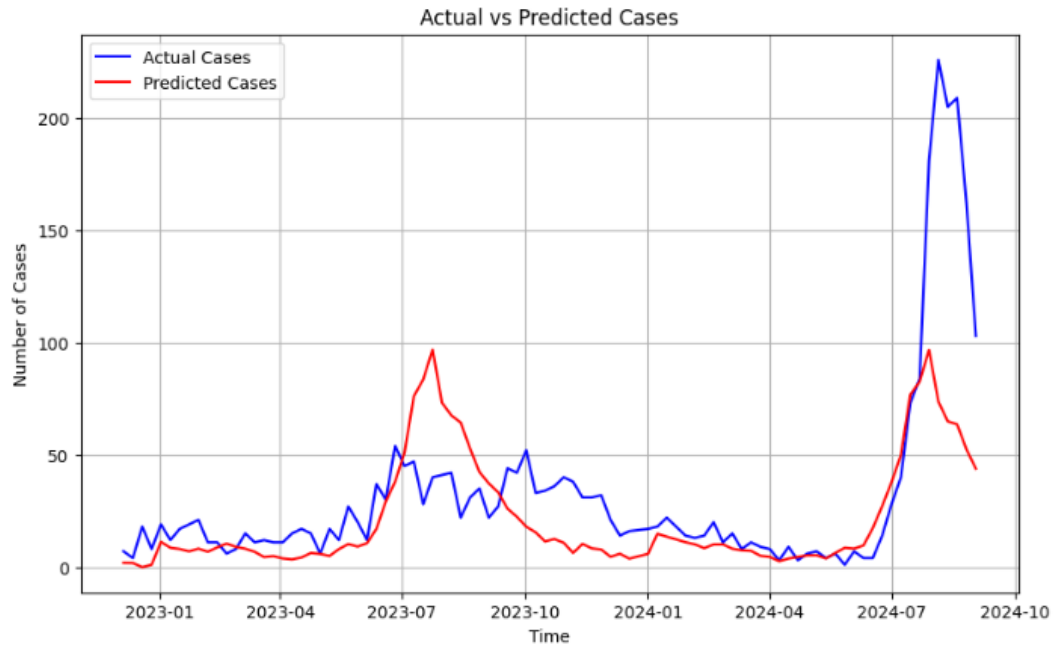


Figure 4.8: Seasonal ARIMA Prediction Results for Test Set

737 This model incorporates seasonal parameters, which were tuned using grid search  
 738 to find the best configuration: **SARIMA(2, 0, 2)(0, 1, 1)[52]**. As with ARIMA,  
 739 **400 iterations** were applied to ensure a robust fit.

### 740 Steps to Create the SARIMA Model:

- 741 1. **Data Preprocessing:** Ensure data readiness by filling any missing values  
 742 and scaling as needed.
- 743 2. **Seasonality Analysis:** Examine the dataset for seasonal patterns. A pe-  
 744 riodicity of **52 weeks** was identified, making SARIMA a suitable choice for  
 745 capturing yearly seasonality.
- 746 3. **Hyperparameter Tuning:** Conduct grid search to identify the best set of  
 747 parameters  $(p, d, q)(P, D, Q)[S]$ , where:
  - 748 • **(p, d, q)** are the non-seasonal parameters,
  - 749 • **(P, D, Q)** are the seasonal parameters, and

750

- $S$  is the season length.

751

The optimal configuration found was  $(2, 0, 2)(0, 1, 1)$ [52].

752

#### 4. Model Training:

753

- Set the iteration count to 400 for enhanced model robustness.

754

- Train the model on the 80% training dataset and reserve the remaining 20% for testing.

755

756

5. **Evaluation:** The SARIMA model yielded the following error metrics:

757

- **MSE:** 1109.69

758

- **RMSE:** 33.31

759

The SARIMA model outperformed the ARIMA model in terms of lower MSE and RMSE values, indicating its effectiveness in capturing the seasonal patterns in the data.

761

762

### 4.3.3 Kalman Filter Model

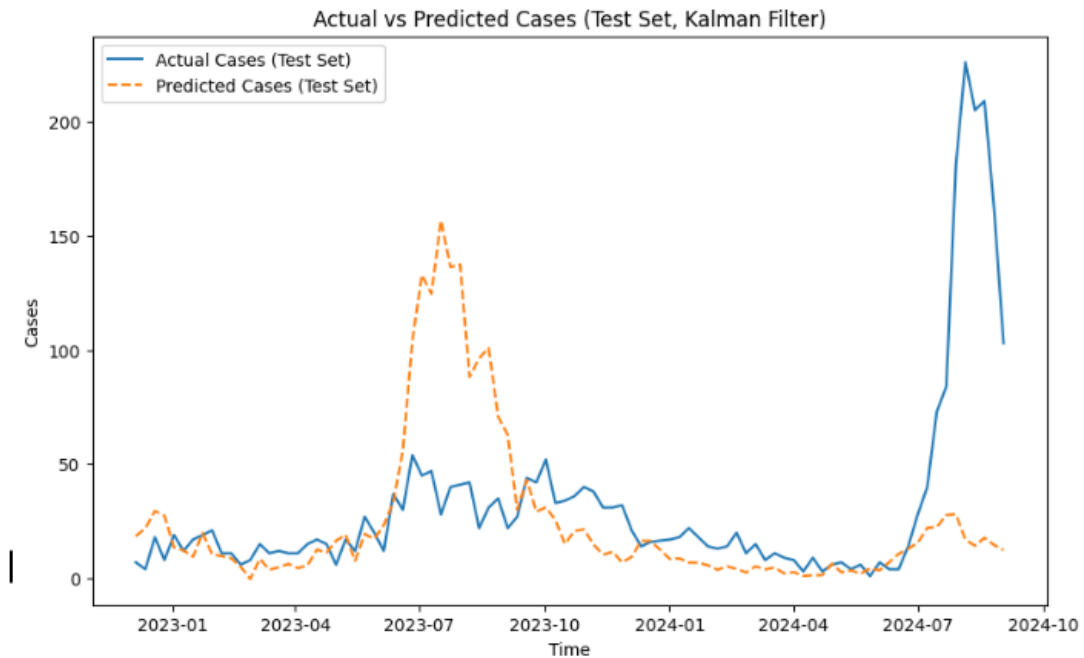


Figure 4.9: Kalman Filter Prediction Results for Test Set

## 763 Kalman Filter Methodology with Matrix Calculations 764

765 **Measurement Acquisition:** Obtain the measurement  $z_k$  of the system's state  
766 with associated confidence. This measurement matrix provides a noisy observation  
767 of the true state.

768 The dataset was split into training and test sets to evaluate the Kalman Filter's  
769 performance and generalizability:

- 770 • **Training Set:** 80% of the data was used for training, enabling the Kalman  
771 Filter model to capture key patterns.
- 772 • **Test Set:** The remaining 20% of the data was reserved for testing.

### 773 Prediction Step:

- 774 • Predict the next state:

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_k$$

775 where  $A$  is the state transition matrix and  $B$  is the control matrix.

- 776 • Update the state covariance:

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q$$

777 where  $Q$  is the process noise covariance matrix.

778 **Compute Residual:** Calculate the residual

$$y_k = z_k - H\hat{x}_{k|k-1}$$

779 where  $H$  is the observation matrix. This residual represents the new information  
780 from the measurement.

### 781 Scaling Factor (Kalman Gain):

- 782 • Compute the Kalman Gain:

$$K_k = P_{k|k-1}H^T(H P_{k|k-1}H^T + R)^{-1}$$

783 where  $R$  is the measurement noise covariance matrix.



- 784     • The Kalman Gain determines the weight of the measurement relative to the  
785     prediction.

786     **State Update:**

- 787     • Update the state estimate:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k y_k$$

788     blending the prediction and measurement.

789     **Uncertainty Update:**

- 790     • Update the state covariance:

$$P_{k|k} = (I - K_k H) P_{k|k-1}$$

791     where  $I$  is the identity matrix.

792     **Model Evaluation:** Upon testing, the Kalman Filter produced a Mean  
793     Squared Error (MSE) of 1474.82 and a Root Mean Squared Error (RMSE) of  
794     38.40.

## 4.4 Preliminary System Requirements

### 4.4.1 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 4.10 depicts the designed database schema that showcases the relationship between the application's entities.

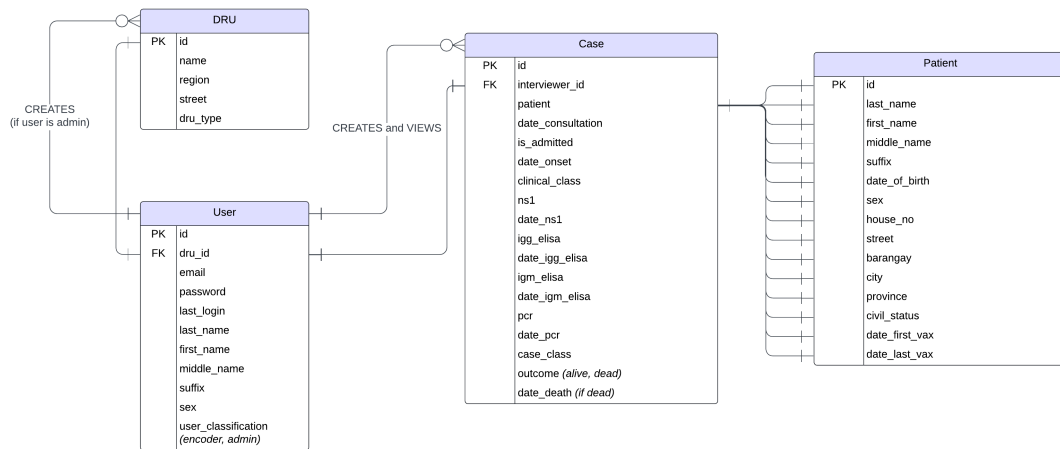


Figure 4.10: Entity-Relationship Database Schema Hybrid Diagram for DengueDash Database Structure

### 4.4.2 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

811 Derivation Function 2 (PBKDF2) with a SHA256 hash function. Utilizing these  
812 techniques ensures that in the event of a data breach, cracking these passwords  
813 would be time-consuming and useless for the attackers.

## 814 **Authentication**

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

## 820 **Data Validation**

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

# 829 **4.5 System Prototype**

## 830 **4.5.1 Guest Interface**

831 The Guest Interface is intended for all visitors of the web application. It shows  
832 the related statistics for dengue cases in a particular area and time. As the system  
833 is still in its testing phase, the data converted into charts shown in Figure 4.11  
834 are generated from Python's Faker library.

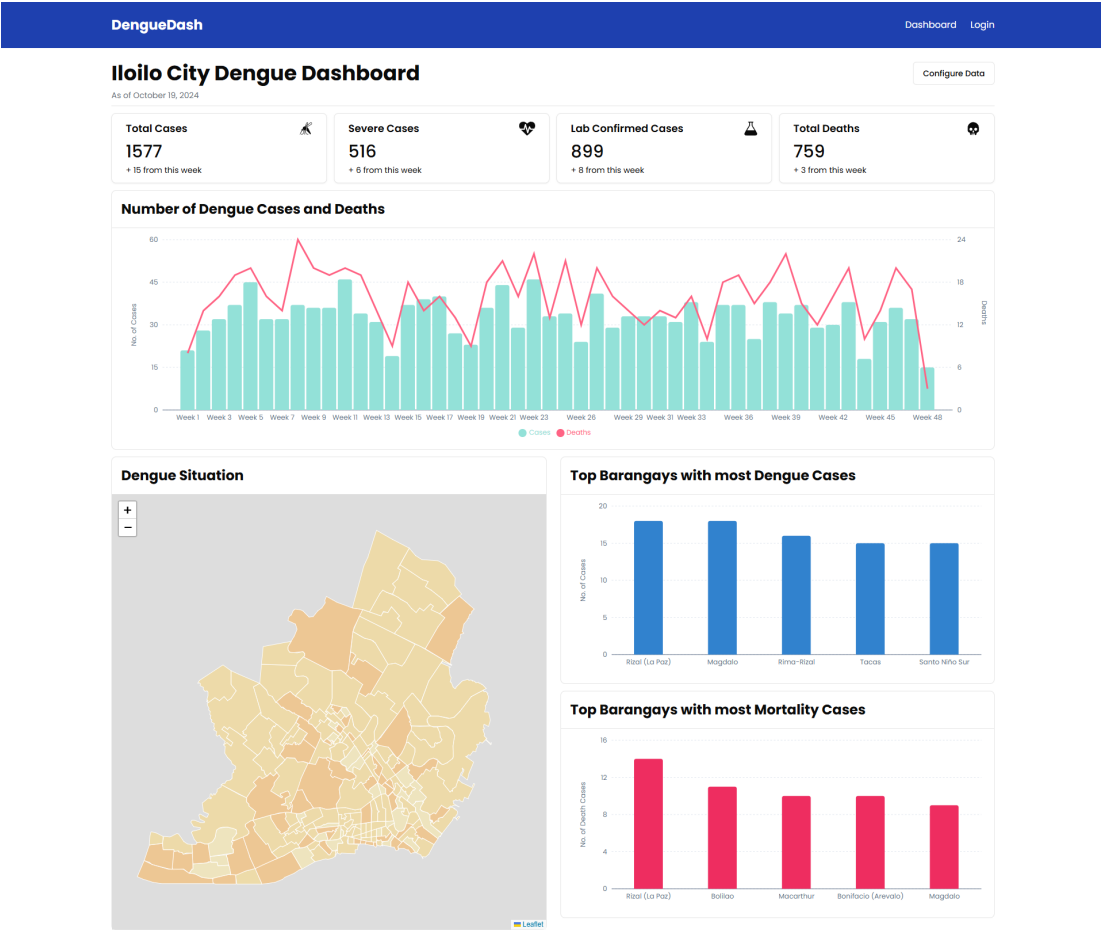


Figure 4.11: Dashboard for Guests

## 4.5.2 Personnel Interface

### User Authentication, and Login

To protect the data's integrity in production, it has been decided that the registration process will not be visible. Instead, an admin must register a user using a different interface. As of the moment, registering a user is done using API via Postman. In the login process, the system implements HTTP-only cookies that contains the JSON Web Tokens (JWT) to protect against XSS attacks. After proper credentials have been provided, it will redirect to the user's home page.

**DengueDash** Dashboard Login

**Welcome back!**

**Email**  
Enter your email

**Password**  
Enter your password

☐ Remember me [Forgot password?](#)

**Continue**

Figure 4.12: Login Page for Users

## Encoder's View

Figures 4.13 and 4.14 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 (PIDS) Dengue Forms, which the referenced form was based on. By doing this, it is assumed that the targeted users will have a familiarity when deployed on a national scale. On a further note, the case form includes the patient's basic information, dengue vaccination status, consultation details, laboratory results, and the outcome.

DengueDash

Modules

Analytics

Forms

Case Report Form

Data Tables

Settings

CN

shadcn

m@example.com

0

Building Your Application

Data Fetching

Personal Information

Clinical Status

Personal Detail

First Name

Middle Name

Last Name

Suffix

Sex

Civil Status

Date of Birth

Address

House No.

Street

Barangay

City

Province

Vaccination

Date of First Vaccination

Date of Last Vaccination

Back

Next

Figure 4.13: First Part of Case Report Form

DengueDash

Modules

Analytics

Forms

Case Report Form

Data Tables

Settings

CN

shadcn

m@example.com

0

Building Your Application

Data Fetching

Personal Information

Clinical Status

Consultation

Date Admitted/Consulted/Seen

Is Admitted?

Date Onset of Illness

Clinical Classification

Laboratory Results

NSI

Date done (NSI)

IgG ELISA

Date done (IgG ELISA)

IgM ELISA

Date done (IgM ELISA)

PCR

Date done (PCR)

Outcome

Case Classification

Outcome

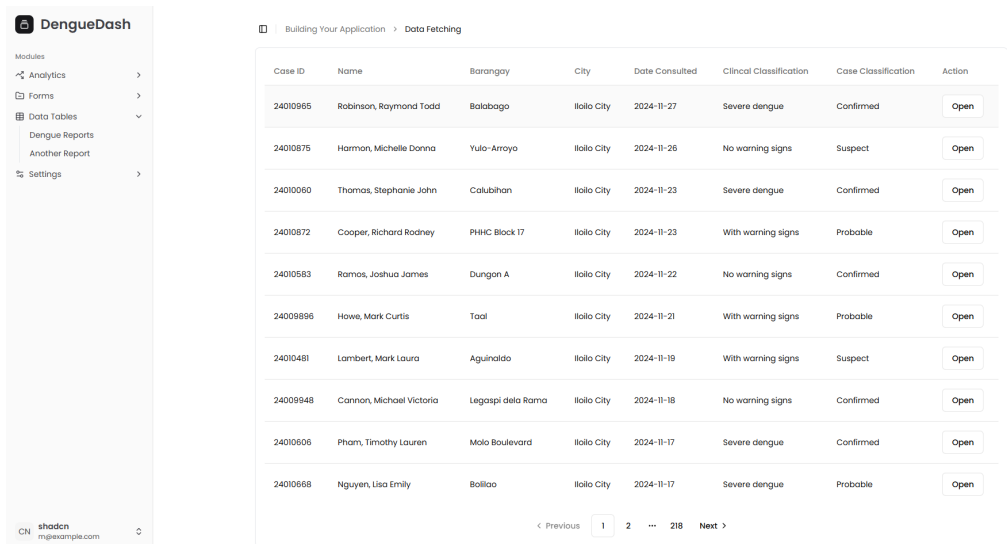
Date of Death

Back

Submit

Figure 4.14: Second Part of Case Report Form

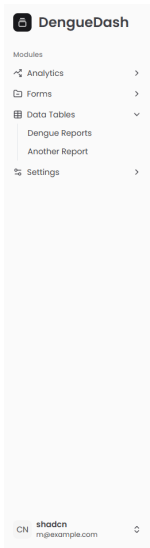
853 Once the data generated from the case report form is validated, it will be  
854 assigned as a new case and can be accessed through the Dengue Reports page, as  
855 shown in Figure 4.15. The said page displays basic information about the patient  
856 related to a specific case, including their name, address, date of consultation, and  
857 clinical and case classifications. It is also worth noting that it only shows cases  
858 the user is permitted to view. For example, in a local Disease Reporting Unit  
859 (DRU) setting, the user can only access records that came from the same DRU.  
860 On the other hand, in a consolidated surveillance unit such as a regional and  
861 provincial quarter, its users can view all the records that came from all the DRUs  
862 that report to them. Moving forward, Figure 4.16 shows the detailed case report  
863 of the patient on a particular consultation date.



The screenshot displays the DengueDash application interface. On the left is a sidebar menu with the following items: Analytics, Forms, Data Tables (expanded), Dengue Reports, Another Report, and Settings. The main content area is titled 'Building Your Application > Data Fetching' and contains a table of dengue reports. The table has columns for Case ID, Name, Barangay, City, Date Consulted, Clinical Classification, Case Classification, and Action. There are 12 rows of data, each with an 'Open' button in the Action column. At the bottom of the table, there is a pagination control showing '1' of 218 items.

Case ID	Name	Barangay	City	Date Consulted	Clinical Classification	Case Classification	Action
24010965	Robinson, Raymond Todd	Balabago	Iloilo City	2024-11-27	Severe dengue	Confirmed	Open
24010975	Harmon, Michelle Donna	Yulo-Arayo	Iloilo City	2024-11-26	No warning signs	Suspect	Open
24010960	Thomas, Stephanie John	Calubihan	Iloilo City	2024-11-23	Severe dengue	Confirmed	Open
24010972	Cooper, Richard Rodney	PHHC Block 17	Iloilo City	2024-11-23	With warning signs	Probable	Open
24010583	Ramos, Joshua James	Dungan A	Iloilo City	2024-11-22	No warning signs	Confirmed	Open
24009896	Howe, Mark Curtis	Taal	Iloilo City	2024-11-21	With warning signs	Probable	Open
24010481	Lambert, Mark Laura	Aguinaldo	Iloilo City	2024-11-19	With warning signs	Suspect	Open
24009948	Cannon, Michael Victoria	Legaspi dela Rama	Iloilo City	2024-11-18	No warning signs	Confirmed	Open
24010606	Pham, Timothy Lauren	Molo Boulevard	Iloilo City	2024-11-17	Severe dengue	Confirmed	Open
24010668	Nguyen, Lisa Emily	Boilao	Iloilo City	2024-11-17	Severe dengue	Probable	Open

Figure 4.15: Dengue Reports



Building Your Application > Data Fetching

### Personal Information

Full Name <b>Thomas, Stephanie John</b>	Date of Birth <b>September 19, 2010</b>
Sex <b>Male</b>	Civil Status <b>Separated</b>
Full Address <b>4189 Rice Coves, Calubihan, Iloilo City, Iloilo</b>	

### Vaccination Status

First Dose <b>February 11, 2024</b>	Last Dose <b>April 10, 2024</b>
--	------------------------------------

### Case Record #24010060

Date of Consultation <b>November 23, 2024</b>	Patient Admitted? <b>Yes</b>
Date Onset of Illness <b>November 22, 2024</b>	Clinical Classification <b>Severe dengue</b>

### Laboratory Results

NSI <b>Pending Result</b>	Date Done <b>N/A</b>
IgG Elisa <b>Negative</b>	Date Done <b>November 29, 2024</b>
IgM Elisa <b>Equivocal</b>	Date Done <b>December 2, 2024</b>
PCR <b>Pending Result</b>	Date Done <b>N/A</b>

### Outcome

Case Classification <b>Confirmed</b>	Outcome <b>Dead</b>
Date of Death <b>November 29, 2024</b>	

Figure 4.16: Detailed Case Report



## References

- 865 *About GitHub and Git - GitHub Docs.* (n.d.). Retrieved from  
866 [https://docs.github.com/en/get-started/start-your-journey/](https://docs.github.com/en/get-started/start-your-journey/about-github-and-git)  
867 [about-github-and-git](https://docs.github.com/en/get-started/start-your-journey/about-github-and-git)
- 868 Ahmadini, A. A. H., Naeem, M., Aamir, M., Dewan, R., Alshqaq, S. S. A., &  
869 Mashwani, W. K. (2021). Analysis and forecast of the number of deaths,  
870 recovered cases, and confirmed cases from covid-19 for the top four affected  
871 countries using kalman filter. *Frontiers in Physics*, 9, 629320.
- 872 Arroyo-Marioli, F., Bullano, F., Kucinskas, S., & Rondón-Moreno, C. (2021).  
873 Tracking r of covid-19: A new real-time estimation using the kalman filter.  
874 *PloS one*, 16(1), e0244474.
- 875 Bosano, R. (2023). *Who: Ph most affected by dengue in western pacific*. Retrieved  
876 Use the date of access, from [https://news.abs-cbn.com/spotlight/12/](https://news.abs-cbn.com/spotlight/12/22/23/who-ph-most-affected-by-dengue-in-western-pacific)  
877 [22/23/who-ph-most-affected-by-dengue-in-western-pacific](https://news.abs-cbn.com/spotlight/12/22/23/who-ph-most-affected-by-dengue-in-western-pacific)
- 878 Bravo, L., Roque, V. G., Brett, J., Dizon, R., & L’Azou, M. (2014). Epidemiology  
879 of dengue disease in the philippines (2000–2011): a systematic literature  
880 review. *PLoS neglected tropical diseases*, 8(11), e3027.
- 881 Carvajal, T. M., Viacrusis, K. M., Hernandez, L. F. T., Ho, H. T., Amalin, D. M.,  
882 & Watanabe, K. (2018). Machine learning methods reveal the temporal  
883 pattern of dengue incidence using meteorological factors in metropolitan  
884 manila, philippines. *BMC infectious diseases*, 18, 1–15.
- 885 *Chart.js.* (n.d.). Retrieved from <https://www.chartjs.org/>
- 886 Cheong, Y. L., Burkart, K., Leitão, P. J., & Lakes, T. (2013). Assessing weather  
887 effects on dengue disease in malaysia. *International journal of environmental*  
888 *research and public health*, 10(12), 6319–6334.
- 889 Christie, T. (n.d.). *Home - Django REST framework*. Retrieved from [https://](https://www.django-rest-framework.org/)  
890 [www.django-rest-framework.org/](https://www.django-rest-framework.org/)
- 891 Colón-González, F. J., Fezzi, C., Lake, I. R., & Hunter, P. R. (2013). The effects  
892 of weather and climate change on dengue. *PLoS neglected tropical diseases*,  
893 7(11), e2503.
- 894 Hii, Y. L., Zhu, H., Ng, N., Ng, L. C., & Rocklöv, J. (2012). Forecast of dengue  
895 incidence using temperature and rainfall. *PLoS neglected tropical diseases*,

896 6(11), e1908.

897 Joel, C. (2021, 10). *6 reasons to use Tailwind over traditional CSS*. Re-  
898 trieved from [https://dev.to/charliejoel/6-reasons-to-use-tailwind](https://dev.to/charliejoel/6-reasons-to-use-tailwind-over-traditional-css-1nc3)  
899 [-over-traditional-css-1nc3](https://dev.to/charliejoel/6-reasons-to-use-tailwind-over-traditional-css-1nc3)

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

902 Lena, P. (2024). *Iloilo beefs up efforts amid hike in dengue cases*. Retrieved Use  
903 the date of access, from <https://www.pna.gov.ph/articles/1231208>

904 Li, X., Feng, S., Hou, N., Li, H., Zhang, S., Jian, Z., & Zi, Q. (2022). Applications  
905 of kalman filtering in time series prediction. In *International conference on*  
906 *intelligent robotics and applications* (pp. 520–531).

907 Ligue, K. D. B., & Ligue, K. J. B. (2022). Deep learning approach to forecasting  
908 dengue cases in davao city using long short-term memory (lstm). *Philippine*  
909 *Journal of Science*, 151(3).

910 Perla. (2024). *Iloilo beefs up efforts amid hike in dengue cases*. Retrieved Use the  
911 date of access, from <https://www.pna.gov.ph/articles/1231208>

912 RabDashDC. (2024). *Rabdash dc*. Retrieved Use the date of access, from [https://](https://rabdash.com)  
913 [rabdash.com](https://rabdash.com)

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

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

917 Watts, D. M., Burke, D. S., Harrison, B. A., Whitmire, R. E., & Nisalak, A.  
918 (2020). Effect of temperature on the transmission of dengue virus by aedes  
919 aegypti. *The American Journal of Tropical Medicine and Hygiene*, 36(1),  
920 143–152.

921 *What is Postman? Postman API Platform*. (n.d.). Retrieved from [https://](https://www.postman.com/product/what-is-postman/)  
922 [www.postman.com/product/what-is-postman/](https://www.postman.com/product/what-is-postman/)

923 WHO. (2023). *Dengue - global situation*. Retrieved Use the date of ac-  
924 cess, from [https://www.who.int/emergencies/disease-outbreak-news/](https://www.who.int/emergencies/disease-outbreak-news/item/2023-DON498)  
925 [item/2023-DON498](https://www.who.int/emergencies/disease-outbreak-news/item/2023-DON498)

926 WHO. (2024). *Dengue and severe dengue*. Retrieved Use the date  
927 of access, from [https://www.who.int/news-room/fact-sheets/detail/](https://www.who.int/news-room/fact-sheets/detail/dengue-and-severe-dengue)  
928 [dengue-and-severe-dengue](https://www.who.int/news-room/fact-sheets/detail/dengue-and-severe-dengue)

929 *Why Visual Studio Code?* (2021, 11). Retrieved from [https://code](https://code.visualstudio.com/docs/editor/whyvscode)  
930 [.visualstudio.com/docs/editor/whyvscode](https://code.visualstudio.com/docs/editor/whyvscode)

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

934 Zhou, S., & Malani, P. (2024). What is dengue? *Jama*, 332(10), 850–850.

935 Zod. (n.d.). *TypeScript-first schema validation with static type inference*. Re-  
936 trieved from <https://zod.dev/?id=introduction>

937 **Appendix A**

938 **Appendix Title**

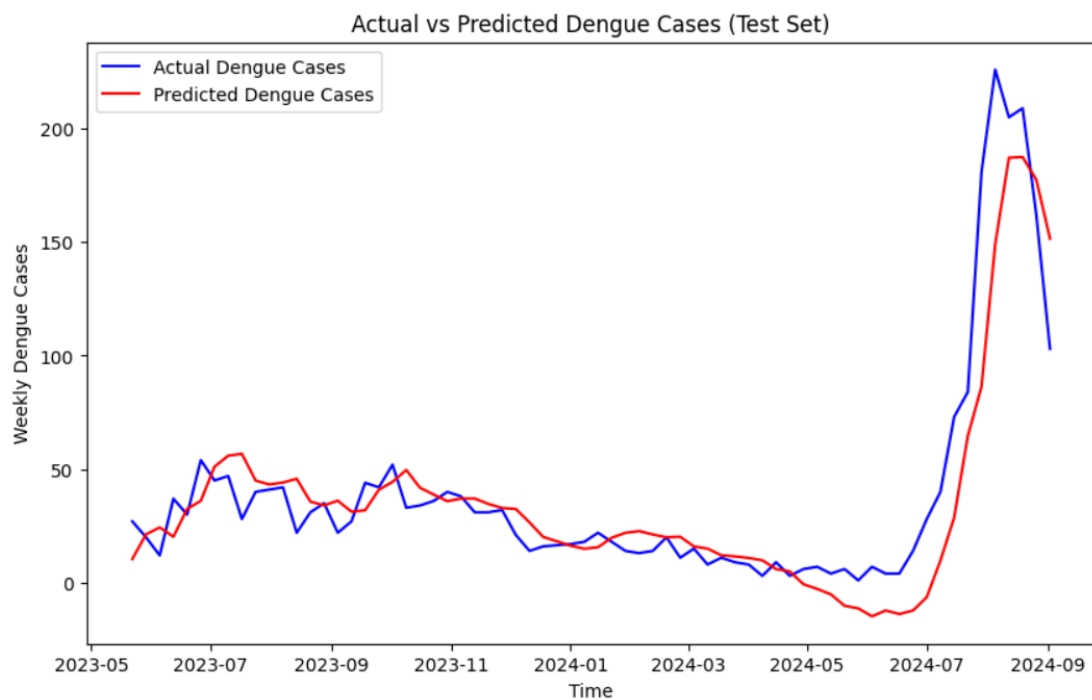


Figure A.1: LSTM Prediction Results for Test Set

## 939 **Appendix B**

### 940 **Resource Persons**

941 **Mr. Firstname1 Lastname1**

942 Role1

943 Affiliation1

944 emailaddr1@domain.com

945 **Ms. Firstname2 Lastname2**

946 Role2

947 Affiliation2

948 emailaddr2@domain.net

949 ....