

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 29, 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	1 Introduction	1
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	2 Review of Related Literature	6
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	3 Research Methodology	10

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	4	Preliminary Results/System Prototype	22
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	29
84	4.3.3	Kalman Filter Model	32
85	4.4	Preliminary System Requirements	34
86	4.4.1	Backend Requirements	34
87	4.4.2	User Interface Requirements	35

88	4.4.3	Security and Validation Requirements	37
89	4.4.4	Testing Process	38
90	4.5	System Prototype	39
91	4.5.1	Guest Interface	39
92	4.5.2	Personnel Interface	40
93		References	44
94		A Appendix Title	46
95		B Resource Persons	47

96 List of Figures

97	3.1 Workflow for forecasting the number of weekly dengue cases . . .	10
98	4.1 Snippet of the Combined Dataset	23
99	4.2 Trend of Dengue Cases	23
100	4.3 Correlation Heatmap	24
101	4.4 Ranking of Correlations	24
102	4.5 Ranking of Correlations (with lagged effects)	25
103	4.6 Comparison of Window Sizes	27
104	4.7 ARIMA Prediction Results for Test Set	29
105	4.8 Seasonal ARIMA Prediction Results for Test Set	30
106	4.9 Kalman Filter Prediction Results for Test Set	32
107	4.10 Entity-Relationship Database Schema Hybrid Diagram for DengueDash	
108	Database Structure	34
109	4.11 Use Case Diagram for Admin	35
110	4.12 Use Case Diagram for Encoder	36
111	4.13 Testing Process for DengueWatch	38
112	4.14 Dashboard for Guests	39
113	4.15 Login Page for Users	40

114	4.16 First Part of Case Report Form	41
115	4.17 Second Part of Case Report Form	41
116	4.18 Dengue Reports	42
117	4.19 Detailed Case Report	43
118	A.1 LSTM Prediction Results for Test Set	46

119 List of Tables

<small>120</small>	3.1	Timetable of Activities for 2024	21
<small>121</small>	3.2	Timetable of Activities for 2025	21
<small>122</small>	4.1	Comparison of Models	26
<small>123</small>	4.2	Comparison of Window Sizes	28

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.

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

155 **1.2 Problem Statement**

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

168 **1.3 Research Objectives**

169 **1.3.1 General Objective**

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

174 **1.3.2 Specific Objectives**

175 Specifically, this study aims to:

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

190 1.4 Scope and Limitations of the Research

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

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

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

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

216 1.5 Significance of the Research

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

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

240 awareness campaigns and community engagement initiatives, empowering
241 residents with knowledge and preventative measures to protect themselves
242 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

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

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

280 2.3 Deep Learning

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

294 2.4 Kalman Filter

295 The Kalman Filter is another powerful tool for time-series forecasting that can be
296 integrated into our analysis. It provides a recursive solution to estimating the state
297 of a linear dynamic system from a series of noisy measurements. Its application
298 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.

331 2.6 Chapter Summary

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

341 The literature further underscores the significance of weather variables—such
342 as temperature and rainfall—in forecasting dengue cases. Studies demonstrate
343 that these variables contribute to accurate outbreak prediction models. Lever-
344 aging these insights, our study will incorporate both weather data and historical
345 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.

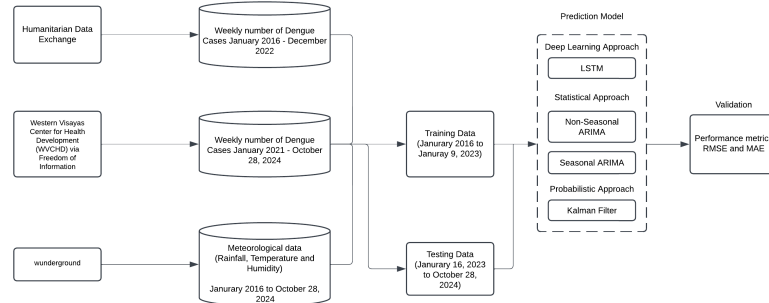


Figure 3.1: Workflow for forecasting the number of weekly dengue cases

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.

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

395 **Data Integration and Preprocessing**

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

403 **Exploratory Data Analysis (EDA)**

- 404 • Analyze trends, seasonality, and correlations between dengue cases and
405 weather factors.
- 406 • Create visualizations like time series plots and scatterplots to highlight re-
407 lationships and patterns in the data.

408 **3.1.2 Develop and Evaluate Deep Learning Models for** 409 **Dengue Case Forecasting**

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

415 Data Preprocessing

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

422 LSTM Model

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

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

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

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

440 The training process was conducted using three distinct window sizes (5 weeks,
441 10 weeks, and 20 weeks) to determine the optimal sequence length of weeks to
442 input into the LSTM model for improved forecasting performance.

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

447 Seasonal ARIMA (SARIMA):

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

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

- 455 • Order: (2, 0, 2)
- 456 • Seasonal Order: (0, 1, 1, 52)

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

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

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

465 ARIMA

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

- 472 • p (autoregressive order): 0 to 3
- 473 • d (differencing order): 0 to 2
- 474 • q (moving average order): 0 to 3

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

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

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

486 **Kalman Filter:**

- 487 • Input Variables: The target variable (Cases) was modeled using three re-
488 gressors: rainfall, max temperature, and humidity.
- 489 • Training and Testing Split: The dataset was split into 80% training and
490 20% testing to evaluate model performance.
- 491 • Observation Matrix: The Kalman Filter requires an observation matrix,
492 which was constructed by adding an intercept (column of ones) to the re-
493 gressors.

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

499 **Model Evaluation and Optimization**

- 500 • Compare the performance of all models to identify the most accurate fore-
501 casting approach.
- 502 • Iteratively optimize the selected model.

503 **3.1.3 Integrate the Predictive Model into a Web-Based** 504 **Data Analytics Dashboard**

505 **Dashboard Design and Development**

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

510 **Model Integration and Deployment**

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

513 **3.1.4 System Development Framework**

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

520 **3.1.5 Design, Building, Testing, and Integration**

521 **Design and Development**

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

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

535 **Testing and Integration**

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

542 **3.2 Development Tools**

543 **3.2.1 Software**

544 **Github**

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

550 **Visual Studio Code**

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

555 Django

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

564 Next.js

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

571 Postman

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

577 3.2.2 Hardware

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

581 3.2.3 Packages

582 Django REST Framework

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

587 Leaflet

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

594 Chart.js

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

602 Tailwind CSS

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

611 **Shadcn**

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

617 **Zod**

618 Data validation is integral in this web application since it will handle crucial data
619 that will be used for analytical inferences and observations. Since Zod is primarily
620 used for validating and parsing data, it ensures proper communication between
621 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 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.

```
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 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.

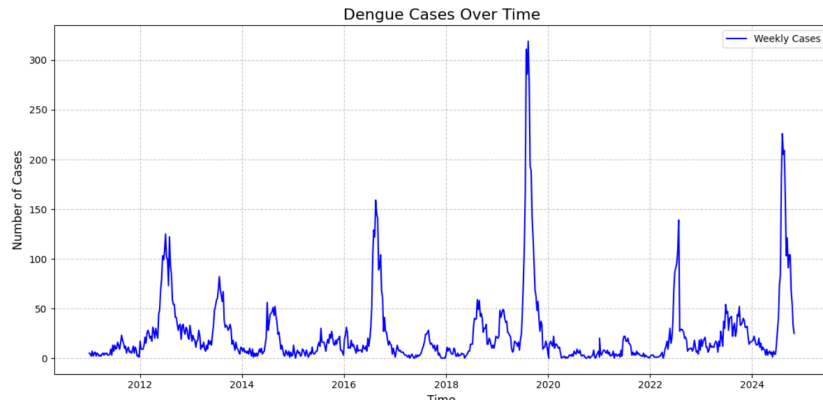


Figure 4.2: Trend of Dengue Cases

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

Figure 4.5 shows the ranking of correlation coefficients between dengue cases and selected features, with the addition of lagged effects. The analysis reveals no improvement in correlation when lagged variables are compared to direct observations. This suggests that the observed values of rainfall, humidity, and maximum temperature remain the most significant predictors for dengue case forecasting.

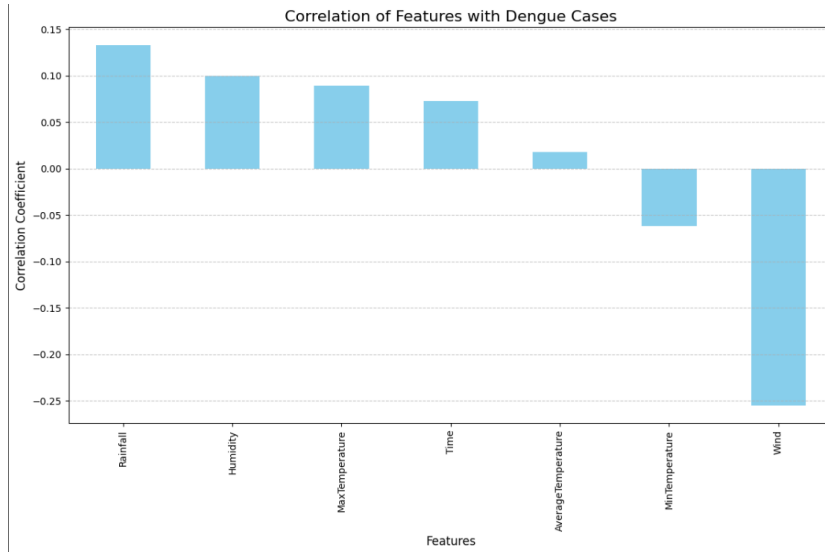


Figure 4.3: Ranking of Correlations

Overall, the exploratory data analysis highlights the significance of rainfall, humidity, and max temperature variables in dengue case forecasting.

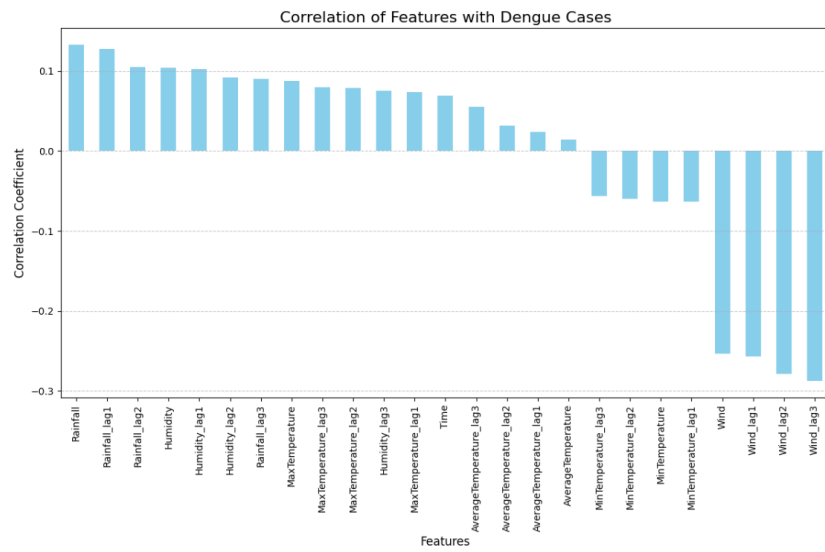


Figure 4.4: Ranking of Correlations (with lagged effects)

4.3 Model Training

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

Using SARIMA and LSTM for dengue forecasting requires an adaptive approach due to seasonal changes and long-term trends. By continuously monitoring performance, incorporating external factors, and updating the model regularly (preferably semi-annually or annually), forecasting accuracy can be maintained. If drastic environmental or epidemiological changes occur, more frequent retraining is necessary. This ensures that public health interventions remain proactive, effectively mitigating dengue outbreaks.

To optimize predictive performance, hyperparameter tuning was conducted individually for each model, refining parameters to achieve the most accurate and reliable forecasts. Following training, the models were rigorously evaluated against the dataset using a set of key performance metrics, including Mean Squared Error (MSE) and Root Mean Squared Error (RMSE).

The table below provides a summary and comparative analysis of each model's results across these metrics, offering insights into the strengths and limitations of 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

4.3.1 LSTM Model

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

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

698 The dataset was split into training and test sets to evaluate the model’s per-
699 formance and generalizability:

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

705 The training process was conducted using three distinct window sizes—5 weeks,
706 10 weeks, and 20 weeks—to identify the optimal sequence length of weeks for input
707 into the LSTM model, thereby enhancing forecasting performance. The following
708 plots illustrate the performance of the model in predicting dengue cases for each
709 of the specified window sizes.

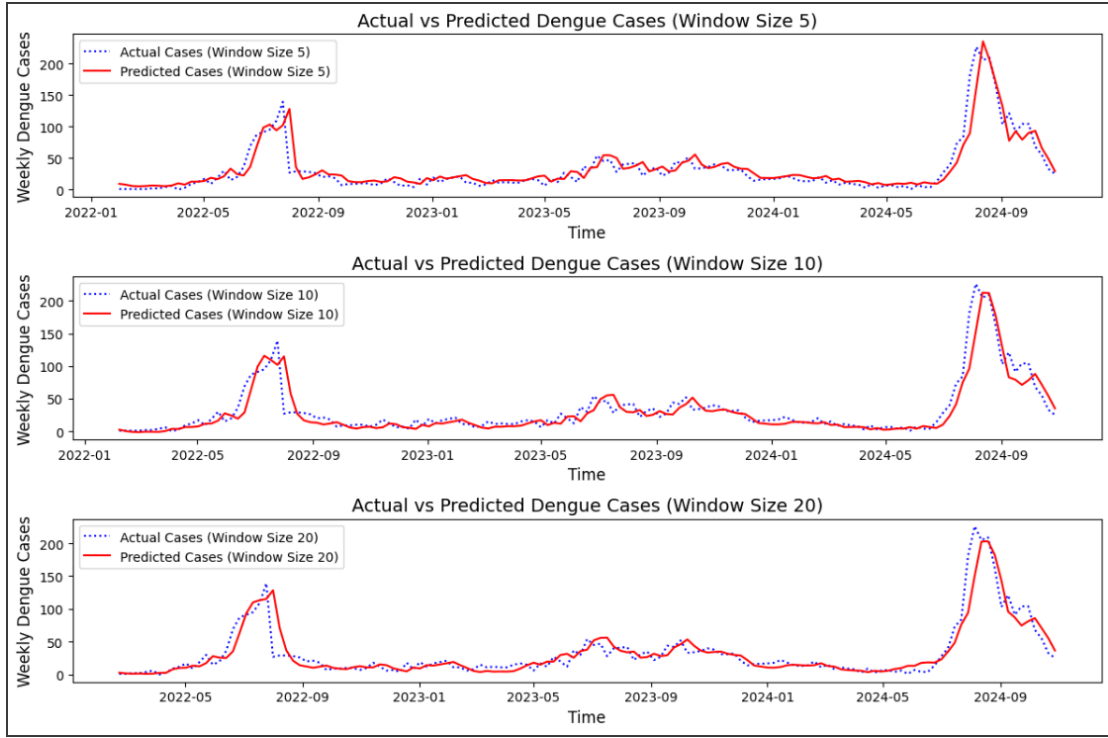


Figure 4.5: Comparison of Window Sizes

710 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

711

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

716 Training and Testing Data Division for ARIMA 717 and Seasonal Arima

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

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

723 4.3.2 ARIMA Model

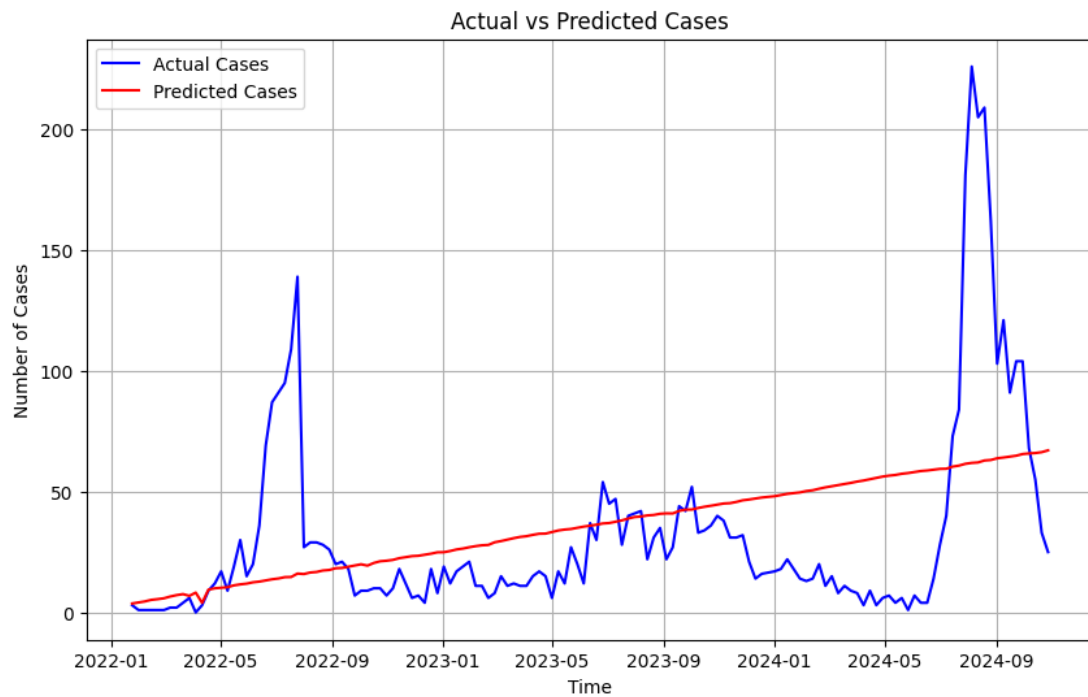


Figure 4.6: ARIMA Prediction Results for Test Set

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

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

743 Seasonal ARIMA (SARIMA) Model

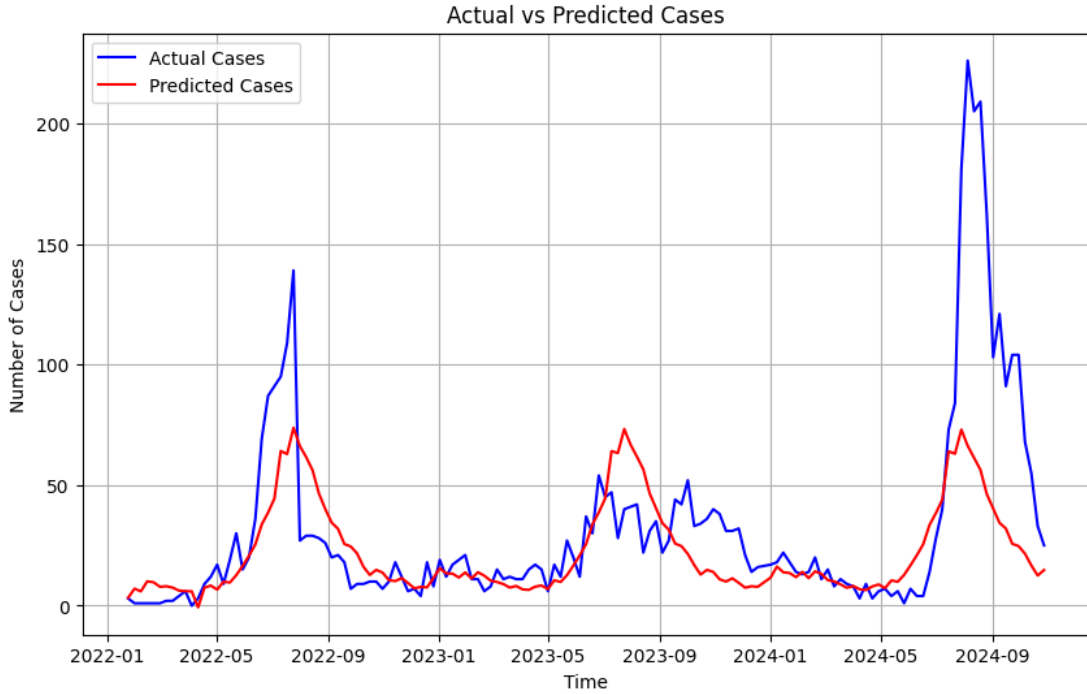


Figure 4.7: Seasonal ARIMA Prediction Results for Test Set

744 This model incorporates seasonal parameters, which were tuned using grid search
 745 to find the best configuration: **SARIMA(2, 0, 2)(0, 1, 1)[52]**. As with ARIMA,
 746 **400 iterations** were applied to ensure a robust fit.

747 Steps to Create the SARIMA Model:

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

- (P, D, Q) are the seasonal parameters, and
- S is the season length.

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

4. Model Training:

- Set the iteration count to 400 for enhanced model robustness.
- Train the model on the 80% training dataset and reserve the remaining 20% for testing.

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

- **MSE:** 1109.69
- **RMSE:** 33.31

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.

4.3.3 Kalman Filter Model

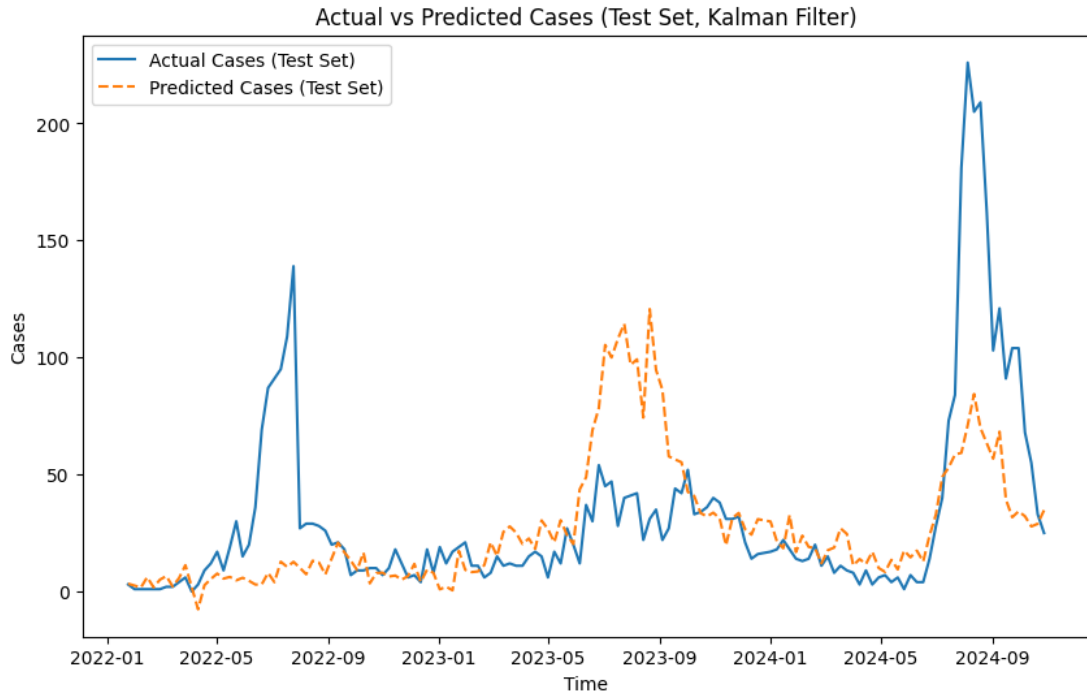


Figure 4.8: Kalman Filter Prediction Results for Test Set

770 Kalman Filter Methodology with Matrix Calculations 771

772 **Measurement Acquisition:** Obtain the measurement z_k of the system's state
773 with associated confidence. This measurement matrix provides a noisy observation
774 of the true state.

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

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

780 Prediction Step:

- 781 • Predict the next state:

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

782 where A is the state transition matrix and B is the control matrix.

- 783 • Update the state covariance:

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

784 where Q is the process noise covariance matrix.

785 **Compute Residual:** Calculate the residual

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

786 where H is the observation matrix. This residual represents the new information
787 from the measurement.

788 Scaling Factor (Kalman Gain):

- 789 • Compute the Kalman Gain:

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

790 where R is the measurement noise covariance matrix.

- 791 • The Kalman Gain determines the weight of the measurement relative to the
792 prediction.

793 **State Update:**

- 794 • Update the state estimate:

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

795 blending the prediction and measurement.

796 **Uncertainty Update:**

- 797 • Update the state covariance:

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

798 where I is the identity matrix.

799 **Model Evaluation:** Upon testing, the Kalman Filter produced a Mean
800 Squared Error (MSE) of 1474.82 and a Root Mean Squared Error (RMSE) of
801 38.40.

802 4.4 Preliminary System Requirements

803 4.4.1 Backend Requirements

804 Database Structure Design

805 Determining how data flows and how it would be structured is crucial in creating
 806 the system as it defines how extendible and flexible it would be for future features
 807 and updates. Thus, creating a comprehensive map of data ensures proper normal-
 808 ization that eliminates data redundancy and improves data integrity. Figure 4.10
 809 depicts the designed database schema that showcases the relationship between the
 810 application's entities.

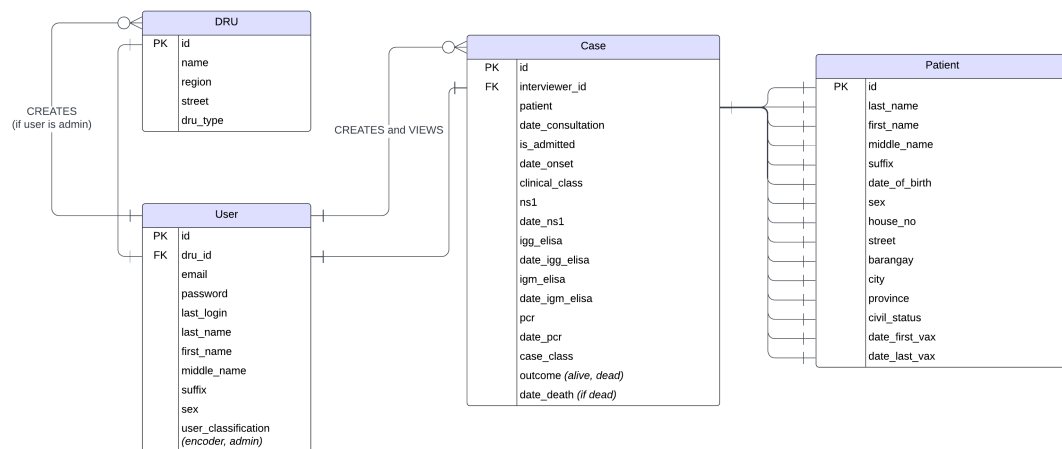


Figure 4.9: Entity-Relationship Database Schema Hybrid Diagram for DengueDash Database Structure

811 4.4.2 User Interface Requirements

812 Admin Interface

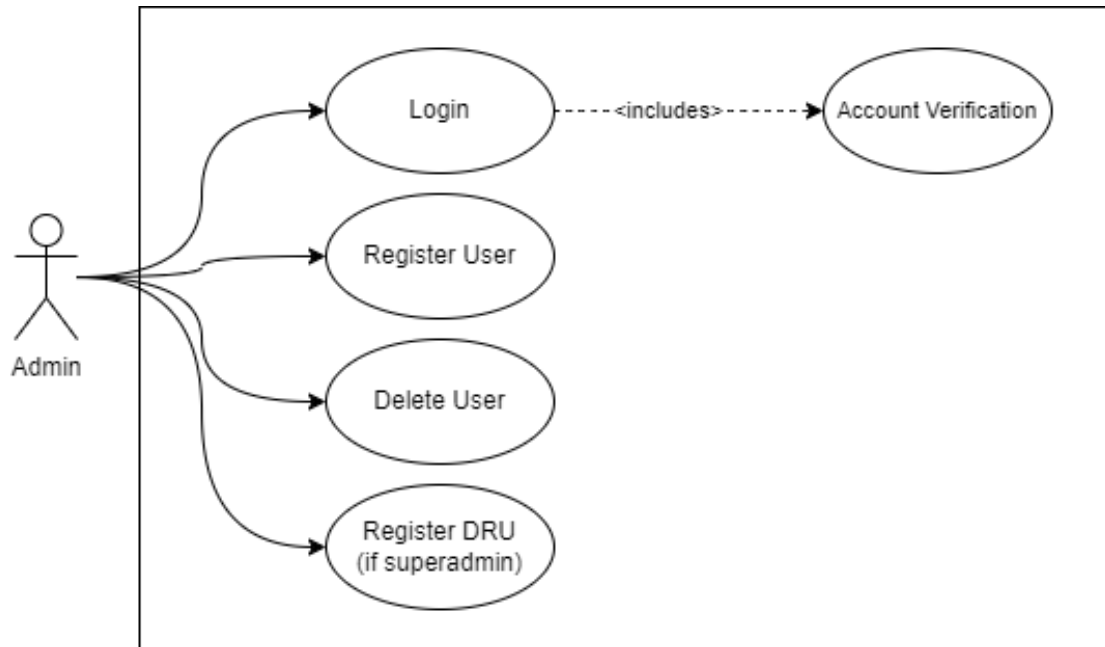


Figure 4.10: Use Case Diagram for Admin

813 Figure 4.11 shows the possible tasks that the admin can do in the application. To
814 protect the integrity of data, only the admins can register and delete accounts.
815 Both account creation and deletion will be done within the application.

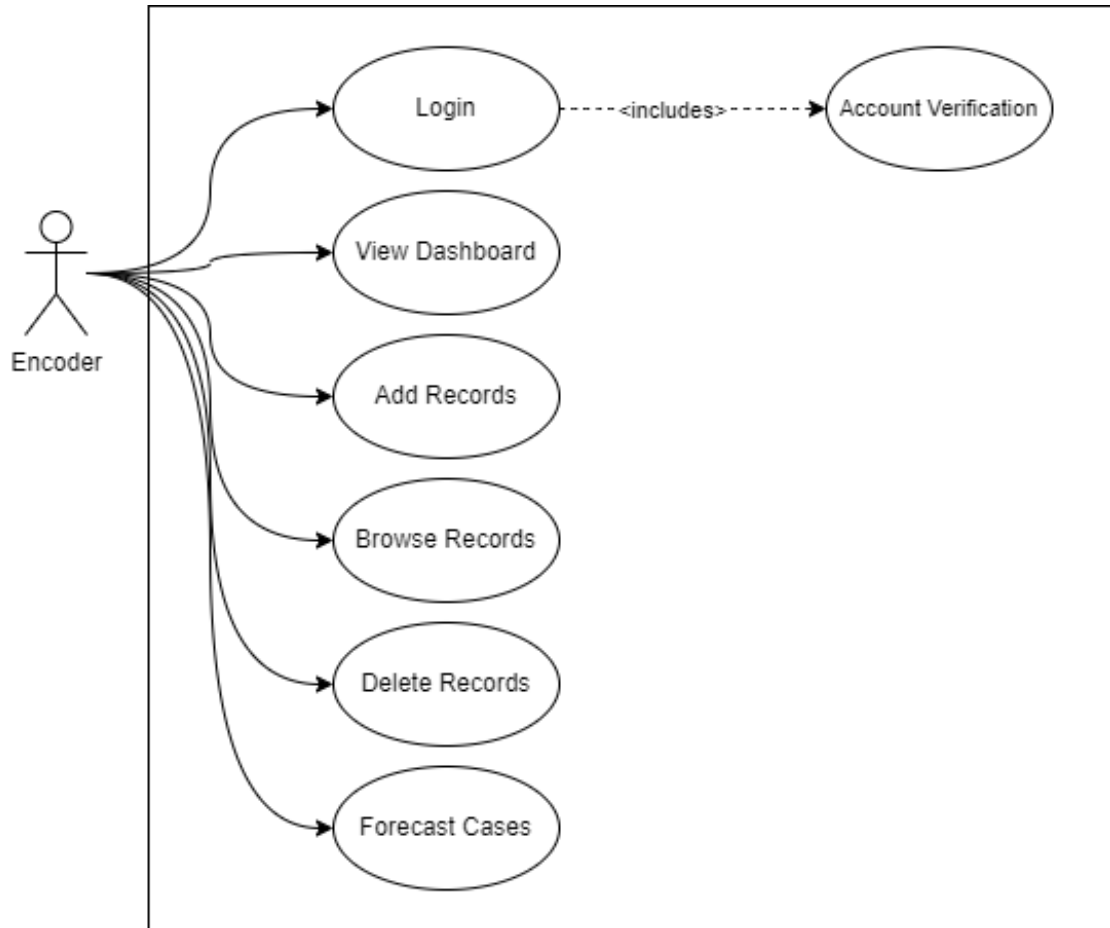


Figure 4.11: Use Case Diagram for Encoder

817 Figure 4.12, on the other hand, illustrates the use cases for the system's primary
818 users. Since only the admin accounts can register a user, the registration process
819 is not part of it. Instead, the main features, which include reporting and viewing
820 records, are the only permitted actions for this type of user. The said processes
821 can be done in the application by filling out a form with details required for each
822 dengue case. As data is entered, it will be consolidated for model training and
823 used for further forecasting of dengue cases.

824 4.4.3 Security and Validation Requirements

825 Password Encryption

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

834 Authentication

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

840 Data Validation

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

849 4.4.4 Testing Process

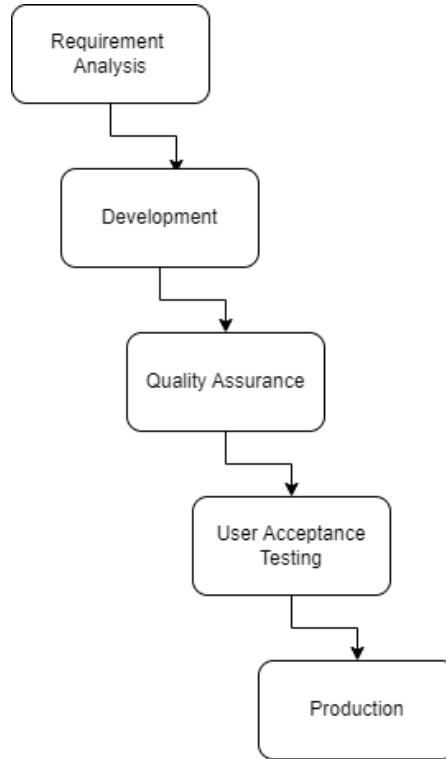


Figure 4.12: Testing Process for DengueWatch

850 As the system requirements and functionalities have been mentioned above, it
851 is important to implement testing to validate the system's performance and effi-
852 cacy. Since dengue reports include confidential information, anonymized historical
853 dengue reports were used to train the model and create the foundational architec-
854 ture of the system. By using functional tests, data validation and visualization can
855 be ensured for further continual improvements. Security testing is also important
856 as it is needed to safeguard confidential information when the system is deployed.
857 It includes proper authentication, permission views, and mitigating common in-
858 jection attacks. Finally, a user acceptance test from the prospected users, in this
859 case, the Iloilo City Epidemiology and Surveillance Unit, is crucial to assess its
860 performance and user experience. It enables the developers to confirm if the sys-
861 tem meets the needs of the problem, and once confirmed, it will be deployed and
862 further evaluated to ensure stability and reliability in live operation.

4.5 System Prototype

4.5.1 Guest Interface

The Guest Interface is intended for all visitors of the web application. It shows the related statistics for dengue cases in a particular area and time. As the system is still in its testing phase, the data converted into charts shown in Figure 4.14 are generated from Python's Faker library.

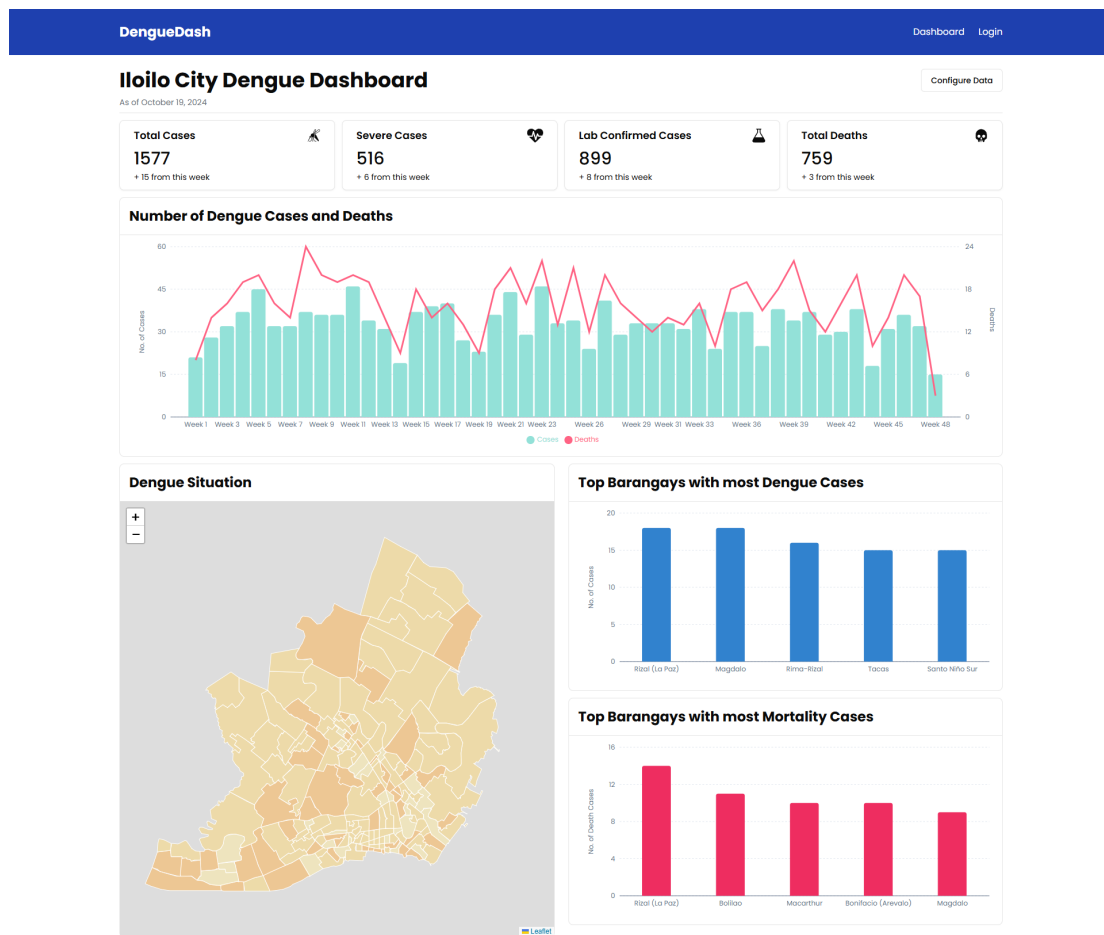


Figure 4.13: Dashboard for Guests

869 4.5.2 Personnel Interface

870 User Authentication, and Login

871 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
872 a different interface. As of the moment, registering a user is done using API via
873 Postman. In the login process, the system implements HTTP-only cookies that
874 contains the JSON Web Tokens (JWT) to protect against XSS attacks. After
875 proper credentials have been provided, it will redirect to the user's home page.
876

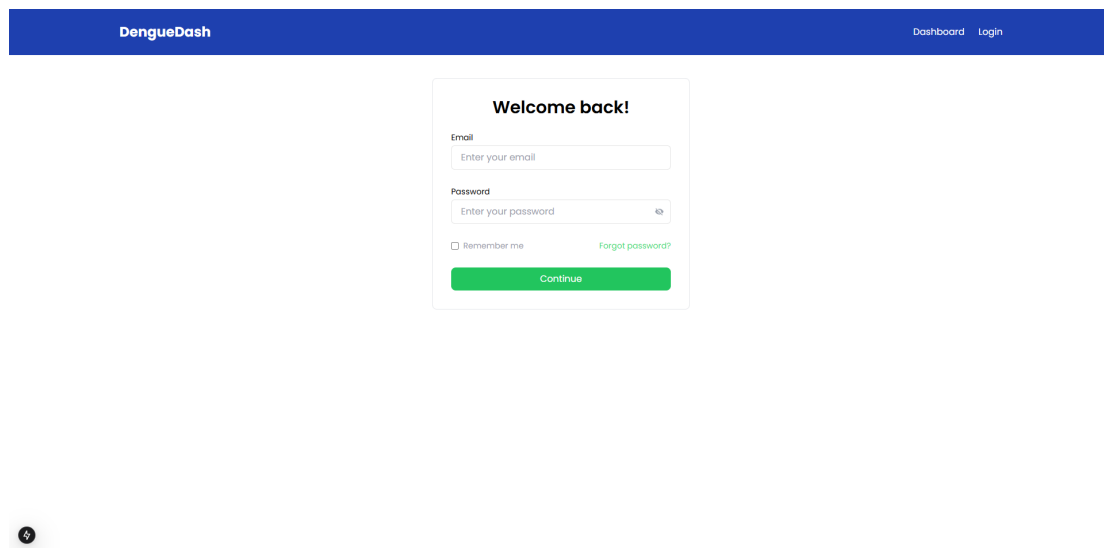


Figure 4.14: Login Page for Users

877 Encoder's View

878 Figures 4.16 and 4.17 show the digitized counterpart of the form obtained from the
879 Iloilo Provincial Epidemiology and Surveillance Unit. As the system aims to support
880 expandability for future features, some fields were modified to accommodate
881 more detailed input. It is worth noting that all of the included fields adhere to the
882 latest Philippine Integrated Disease and Surveillance Response (PIDSR) Dengue
883 Forms, which the referenced form was based on. By doing this, it is assumed
884 that the targeted users will have a familiarity when deployed on a national scale.
885 On a further note, the case form includes the patient's basic information, dengue
886 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 4.15: 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

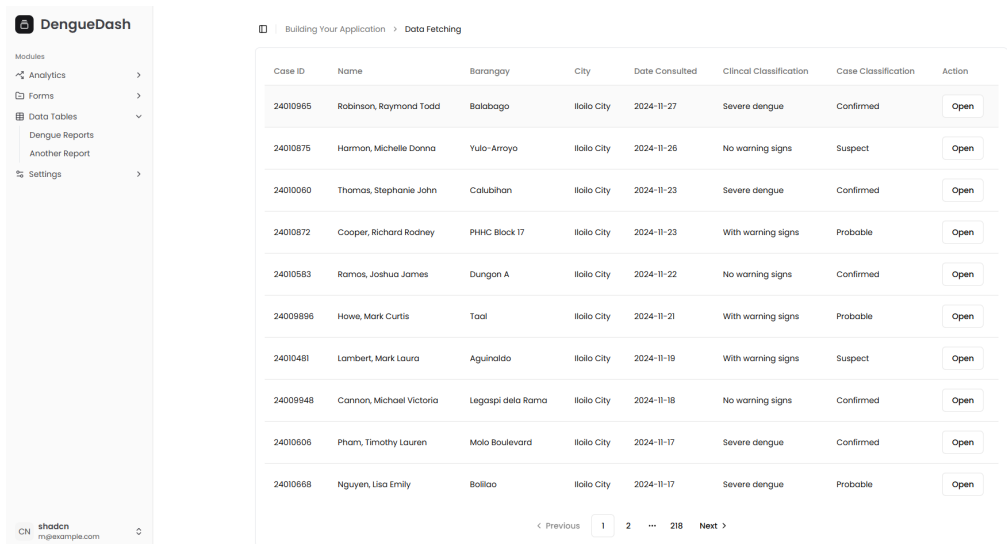
Date of Death

Back

Submit

Figure 4.16: Second Part of Case Report Form

887 Once the data generated from the case report form is validated, it will be
888 assigned as a new case and can be accessed through the Dengue Reports page, as
889 shown in Figure 4.18. The said page displays basic information about the patient
890 related to a specific case, including their name, address, date of consultation, and
891 clinical and case classifications. It is also worth noting that it only shows cases
892 the user is permitted to view. For example, in a local Disease Reporting Unit
893 (DRU) setting, the user can only access records that came from the same DRU.
894 On the other hand, in a consolidated surveillance unit such as a regional and
895 provincial quarter, its users can view all the records that came from all the DRUs
896 that report to them. Moving forward, Figure 4.19 shows the detailed case report
897 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 4.17: Dengue Reports

The image is a screenshot of a web application interface. At the top, there is a dark header bar with a logo on the left and the text 'DengueDash' on the right. Below the header, the main content area is divided into two sections. The first section, titled 'Modules', contains a list of items: 'Analytics', 'Forms', 'Data Tables', 'Dengue Reports', and 'Settings'. Each item has a small icon to its left and a right-pointing arrow. The 'Dengue Reports' item is highlighted with a blue background. The second section, titled 'Dengue Reports', shows a list of reports. The first report is 'Another Report', which is highlighted with a blue background. Below it, there is a table with columns for 'Report Name', 'Status', and 'Action'. The table contains one row with the value 'Another Report' in the 'Report Name' column, 'Completed' in the 'Status' column, and a right-pointing arrow in the 'Action' column. At the bottom of the page, there is a footer bar with the text 'shadcn' and 'myexampl.com' on the left, and a small icon on the right.

Building Your Application > Data Feeding	
 Personal Information	
Full Name Thomas, Stephanie John	Date of Birth September 19, 2010
Sex Male	Civil Status Separated
Full Address 4189 Rice Coves, Calubihan, Iloilo City, Iloilo	
 Vaccination Status	
First Dose February 11, 2024	Last Dose April 10, 2024
 Case Record #24010060	
Date of Consultation November 23, 2024	Patient Admitted? Yes
Date Onset of Illness November 22, 2024	Clinical Classification Severe dengue
 Laboratory Results	
NSI Pending Result	Date Done N/A
IgG Elisa Negative	Date Done November 29, 2024
IgM Elisa Equivocal	Date Done December 2, 2024
PCR Pending Result	Date Done N/A
 Outcome	
Case Classification Confirmed	Outcome Dead
Date of Death November 29, 2024	

Figure 4.18: Detailed Case Report

References

- 899 *About GitHub and Git - GitHub Docs.* (n.d.). Retrieved from
900 [https://docs.github.com/en/get-started/start-your-journey/
901 about-github-and-git](https://docs.github.com/en/get-started/start-your-journey/about-github-and-git)
- 902 Ahmadini, A. A. H., Naeem, M., Aamir, M., Dewan, R., Alshqaq, S. S. A., &
903 Mashwani, W. K. (2021). Analysis and forecast of the number of deaths,
904 recovered cases, and confirmed cases from covid-19 for the top four affected
905 countries using kalman filter. *Frontiers in Physics*, 9, 629320.
- 906 Arroyo-Marioli, F., Bullano, F., Kucinskas, S., & Rondón-Moreno, C. (2021).
907 Tracking r of covid-19: A new real-time estimation using the kalman filter.
908 *PloS one*, 16(1), e0244474.
- 909 Bosano, R. (2023). *Who: Ph most affected by dengue in western pacific*. Retrieved
910 Use the date of access, from [https://news.abs-cbn.com/spotlight/12/
911 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)
- 912 Bravo, L., Roque, V. G., Brett, J., Dizon, R., & L’Azou, M. (2014). Epidemiology
913 of dengue disease in the philippines (2000–2011): a systematic literature
914 review. *PLoS neglected tropical diseases*, 8(11), e3027.
- 915 Carvajal, T. M., Viacrusis, K. M., Hernandez, L. F. T., Ho, H. T., Amalin, D. M.,
916 & Watanabe, K. (2018). Machine learning methods reveal the temporal
917 pattern of dengue incidence using meteorological factors in metropolitan
918 manila, philippines. *BMC infectious diseases*, 18, 1–15.
- 919 *Chart.js.* (n.d.). Retrieved from <https://www.chartjs.org/>
- 920 Cheong, Y. L., Burkart, K., Leitão, P. J., & Lakes, T. (2013). Assessing weather
921 effects on dengue disease in malaysia. *International journal of environmental
922 research and public health*, 10(12), 6319–6334.
- 923 Christie, T. (n.d.). *Home - Django REST framework*. Retrieved from [https://
924 www.django-rest-framework.org/](https://www.django-rest-framework.org/)
- 925 Colón-González, F. J., Fezzi, C., Lake, I. R., & Hunter, P. R. (2013). The effects
926 of weather and climate change on dengue. *PLoS neglected tropical diseases*,
927 7(11), e2503.
- 928 Hii, Y. L., Zhu, H., Ng, N., Ng, L. C., & Rocklöv, J. (2012). Forecast of dengue
929 incidence using temperature and rainfall. *PLoS neglected tropical diseases*,

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/>

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

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>

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, D. M., Burke, D. S., Harrison, B. A., Whitmire, R. E., & Nisalak, A. (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/>

WHO. (2023). *Dengue - global situation*. Retrieved Use the date of access, from <https://www.who.int/emergencies/disease-outbreak-news/item/2023-DON498>

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

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>

971 **Appendix A**

972 **Appendix Title**

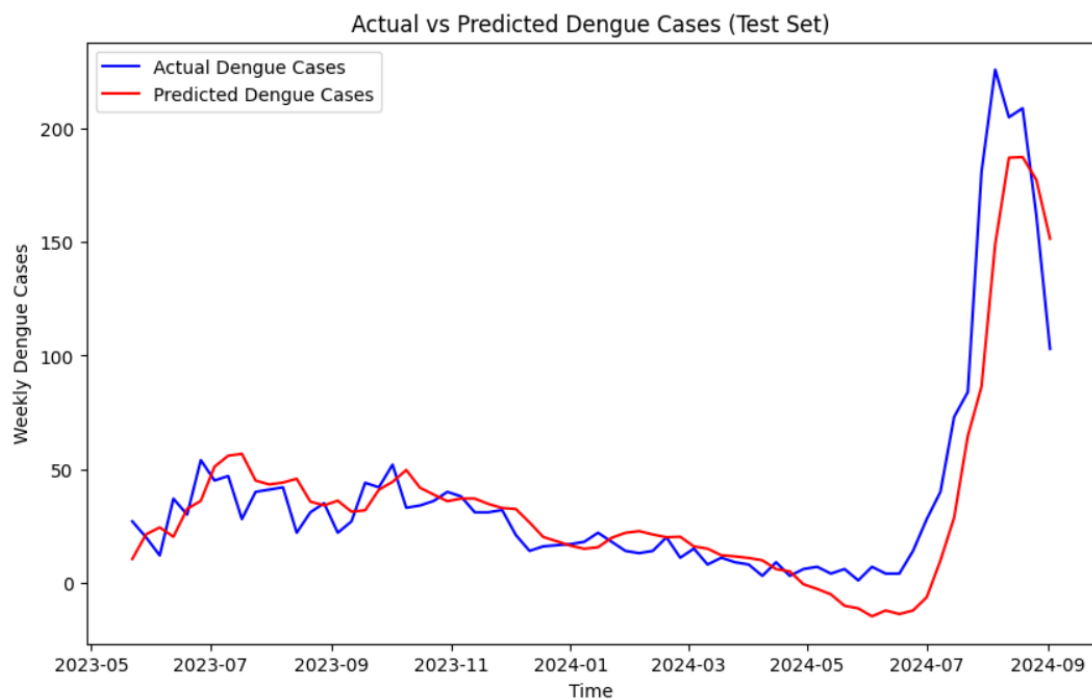


Figure A.1: LSTM Prediction Results for Test Set

973 **Appendix B**

974 **Resource Persons**

975 **Mr. Firstname1 Lastname1**
976 Role1
977 Affiliation1
978 emailaddr1@domain.com

979 **Ms. Firstname2 Lastname2**
980 Role2
981 Affiliation2
982 emailaddr2@domain.net

983