DNN Implementation for Disease Outbreak Forecasting (HealthTrace)

Ethan Jed V. Carbonell and Kirk Henrich C. Gamo

West Visayas State Universtiry
Luna St, La Paz, Iloilo City, 5000 Iloilo

December 12, 2025

ABSTRACT

Aims. This study aims to develop "HealthTrace," a deep learning-based forecasting system designed to predict disease incidence 14 days in advance. The primary objective is to utilize Long Short-Term Memory (LSTM) networks to provide public health officials in Iloilo City with an accurate early warning system, facilitating proactive resource allocation and timely intervention.
Methods. The system was built using a comprehensive dataset from the CCHAIN project, comprising daily records of 52 environmental, socioeconomic, and health features over two years. Data processing involved normalizing these features using MinMax scaling and structuring them into 30-day temporal sequences to capture incubation periods. We implemented and compared two distinct recurrent neural network architectures—Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU)—using TensorFlow and Keras. A rigorous hyperparameter tuning process was conducted, evaluating 16 different configurations across varying learning rates, batch sizes, and optimizers (Adam, RMSprop, SGD) to identify the optimal model structure for minimizing regression error.
Results. The study revealed that the LSTM architecture utilizing the Adam optimizer (Configuration 15) yielded the highest predictive accuracy, achieving a Test R^2 score of approximately 0.52 and effectively modeling non-linear disease trends. While GRU models offered a lighter parameter footprint and faster inference times, they demonstrated slightly lower accuracy compared to the LSTM baseline. The comparative analysis highlighted a distinct hierarchy in optimizer performance, with adaptive methods (Adam) significantly outperforming Stochastic Gradient Descent (SGD). The final deployed system successfully generates 14-day forecasts with low Mean Absolute Error, providing a functional web-based dashboard for real-time monitoring and risk assessment.

1. Problem & Solution Overview

1.1. The Problem

Public health officials in Iloilo City and the Philippines face significant challenges in allocating resources for disease outbreaks such as Dengue, Typhoid, and Leptospirosis. Traditional statistical methods often fail to capture the complex, non-linear dependencies between climate factors (rainfall, humidity, temperature) and disease incubation periods, leading to delayed responses and overwhelmed healthcare facilities.

1.2. The Solution (HealthTrace)

To address this, we developed HealthTrace, a Python-based forecasting system. The core of this solution is a Deep Neural Network utilizing Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) architectures. Unlike standard regression, these recurrent networks are capable of learning long-term temporal dependencies in time-series data, allowing us to predict disease incidence 14 days in advance based on historical climate patterns.

2. Tools, Ethics & Dataset

2.1. Tools & Technologies

- **Backend:** Flask
- **Deep Learning:** TensorFlow/Keras
- **Data Processing:** Pandas, NumPy, Scikit-learn
- **Visualization:** Plotly.js
- **Frontend:** HTML5, CSS3, JavaScript

2.2. Privacy & Bias Mitigation

The dataset consists entirely of aggregated daily counts. No Personally Identifiable Information (PII) or specific patient records were used, ensuring full compliance with user privacy standards.

We normalized all features using MinMaxScaler to prevent features with larger ranges (like Rainfall) from biasing the network weights against smaller range features (like Disease Count).

2.3. Dataset Source

The model was trained using historical climate and health data aggregated from CCHAIN (Climate Change and Health in the Philippines).

2.3.1. Input Features

The forecasting model utilizes 2 primary input feature categories: Environmental & Socioeconomic Features, and Health Features. The Environmental & Socioeconomic Features consists of nine subcategories: Climate & Precipitation, Temperature, Air Quality, Socioeconomic, Wealth Index, Vegitation, Sanitation & Water Access, Water Bodies, and Healthcare Access. These environmental and health metrics are processed into 30-day time-series sequences, allowing the LSTM network to effectively learn the incubation periods and delayed effects of climate on disease transmission. The dataset consists of approximately 11,680 total data points, representing two full years of continuous daily records across the four tracked diseases.

Table 1: Complete List of Environmental & Socioeconomic Features

Climate & Precipitation	
precipitation	spi3
spi6	precip_anomaly
precipitation_7day	precipitation_30day
Temperature	
tmin	tmax
tave	temp_range
tave_7day	tave_30day
Air Quality	
no2	co
so2	o3
pm10	pm25
Socioeconomic	
pop_count_total	pop_density_mean
avg_rad_mean	
Wealth Index	
rwi_mean	rwi_median
rwi_std	
Vegetation	
ndvi	
Sanitation & Water Access	
drinking_water_count	drinking_water_nearest
water_well_count	water_well_nearest
toilet_count	toilet_nearest
waste_basket_count	waste_basket_nearest
wastewater_plant_count	wastewater_plant_nearest
Water Bodies	
osm_wetland_nearest	osm_reservoir_nearest
osm_water_nearest	osm_riverbank_nearest
osm_river_nearest	osm_stream_nearest
osm_canal_nearest	osm_drain_nearest
Healthcare Access	
clinic_count	clinic_nearest
hospital_count	hospital_nearest
pharmacy_count	pharmacy_nearest
doctors_count	doctors_nearest

Table 2: Target Health Variable

Health Feature
disease_cases

The following figures present sample time-series visualizations derived from the dengue historical dataset. These graphs illustrate the temporal dynamics and trends of four distinct feature categories:

- **Precipitation:** Daily rainfall patterns, seasonal intensity and variability.
- **Disease Cases:** Incidence of dengue, the primary epidemiological target variable.
- **SPI6:** Medium-term hydrological conditions, identifies prolonged wet or dry periods affecting vector breeding.
- **Population Count:** Demographic trend and scale of the exposed population over the study period.

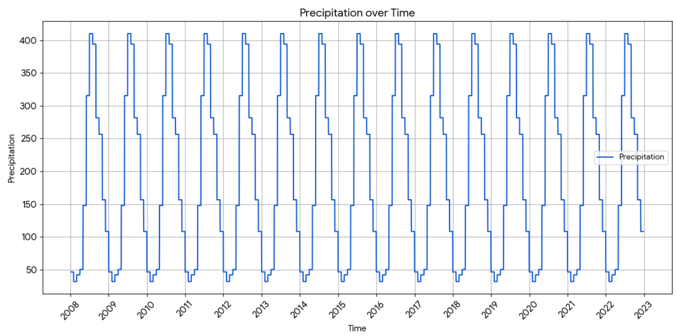


Fig. 1: Precipitation over Time

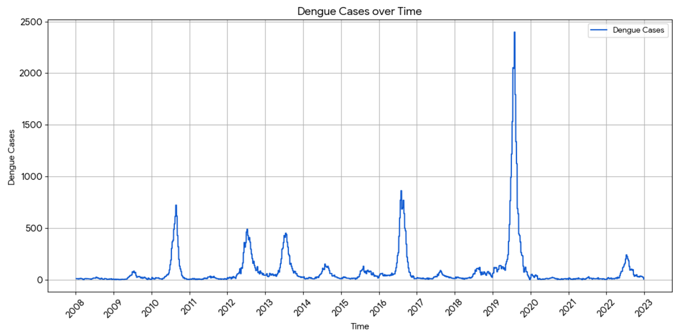


Fig. 2: Dengue Cases over Time

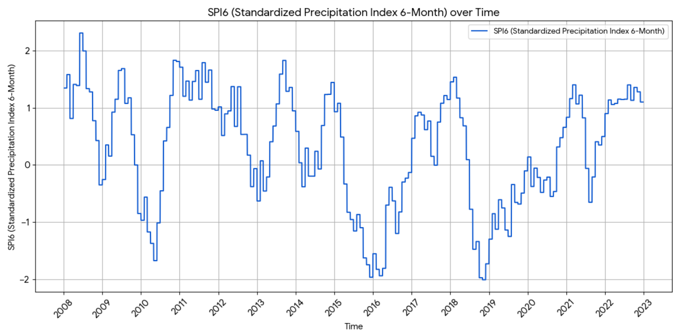


Fig. 3: SPI6 over Time

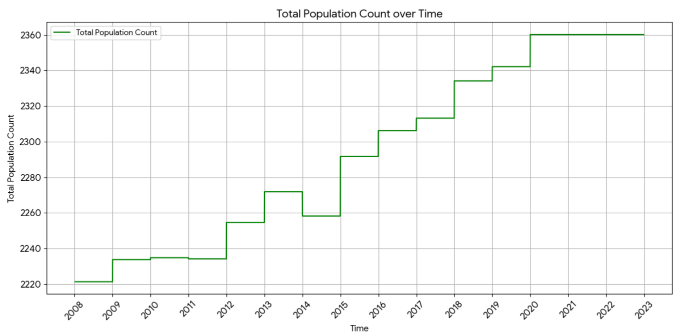


Fig. 4: Population Count over Time

3. Neural Network Structure

The forecasting core utilizes a sequential Deep Learning architecture designed for time-series regression. The model consists of two stacked recurrent layers—either Long Short-Term Memory (LSTM) or Gated Recurrent Units (GRU)—to capture temporal dependencies in the disease data. The first layer contains 64 units and returns sequences to retain temporal structure, while the second layer reduces complexity with 32 units. To prevent overfitting, Dropout regularization (rate of 0.3) is applied after each recurrent layer. The architecture concludes with a fully connected Dense layer of 32 units using ReLU activation for non-linear feature processing, followed by a single-unit output layer that predicts the specific number of disease cases.

Table 3: Neural Network Architecture

Layer	Type	Units/Rate	Activation	Configuration / Purpose
		Recurrent Layers (Feature Extraction)		
1	Input	52	–	Seq Length=30, Features=52
2	LSTM / GRU	64	Tanh	Return Sequences=True
3	Dropout	0.2	–	Regularization (Prevent Overfitting)
4	LSTM / GRU	32	Tanh	Return Sequences=False
5	Dropout	0.3	–	Regularization (Prevent Overfitting)
		Dense Layers (Prediction)		
6	Dense	32	ReLU	Feature Interpretation
7	Dropout	0.3	–	Regularization (Prevent Overfitting)
8	Output	1	Linear	Final Regression Prediction

Note: Both LSTM and GRU architectures were implemented to conduct a direct performance comparison and identify the optimal model for this specific dataset. While GRU models generally offer a lighter parameter footprint and faster inference times, the comprehensive evaluation metrics demonstrated that the LSTM architecture provided superior predictive accuracy for these disease outbreak patterns, leading to its selection as the primary model.

4. Hyperparameter Tuning

To optimize the performance of the HealthTrace forecasting system, we conducted a comprehensive series of controlled experiments. Logging for the hyperparameter tuning configurations was performed on the Dengue dataset. We evaluated a total of 16 different training configurations, systematically varying hyperparameters such as the optimizer, learning rate, batch size, units, dropout rate, and optimizer-specific parameters. The goal was to balance model complexity with generalization capabilities to avoid both underfitting and overfitting. Below are the detailed results for all 16 configurations. As shown in the results, Configuration 15 achieved the best performance on the validation set, demonstrating the highest *Test_*R^2.

4.1. Hyperparameter Tuning

Table 4: Configuration 1

Training Config	Training Results	Validation Results	Test Results
Epochs: 31	*Loss:* 0.00012	*Val_Loss:* 0.01333	*Test_Loss:* 0.00052
Units: 64	*MAE:* 0.00672	*Val_MAE:* 0.05339	*Test_MAE:* 0.01329
Dropout: 0.2	*RMSE:* 0.01105	*Val_RMSE:* 0.11547	*Test_RMSE:* 0.02278
Optimizer: Adam	R^2: 0.95645	*Val_*R^2: 0.61038	*Test_*R^2: -0.22324
LR: 0.001			
Batch: 32			

Table 5: Configuration 2

Training Config	Training Results	Validation Results	Test Results
Epochs: 38	*Loss:* 0.00008	*Val_Loss:* 0.00852	*Test_Loss:* 0.00037
Units: 64	*MAE:* 0.00713	*Val_MAE:* 0.04193	*Test_MAE:* 0.01268
Dropout: 0.2	*RMSE:* 0.00920	*Val_RMSE:* 0.09232	*Test_RMSE:* 0.01929
Optimizer: Adam	R^2: 0.96985	*Val_*R^2: 0.75096	*Test_*R^2: 0.12294
LR: 0.01			
Batch: 32			

Table 6: Configuration 3

Training Config	Training Results	Validation Results	Test Results
Epochs: 32	*Loss:* 0.00021	*Val_Loss:* 0.01237	*Test_Loss:* 0.00090
Units: 64	*MAE:* 0.00961	*Val_MAE:* 0.05191	*Test_MAE:* 0.02016
Dropout: 0.2	*RMSE:* 0.01439	*Val_RMSE:* 0.11120	*Test_RMSE:* 0.03007
Optimizer: Adam	R^2: 0.92626	*Val_*R^2: 0.63863	*Test_*R^2: -1.13115
LR: 0.0001			
Batch: 32			

Table 7: Configuration 4

Training Config	Training Results	Validation Results	Test Results
Epochs: 69	*Loss:* 0.00006	*Val_Loss:* 0.00779	*Test_Loss:* 0.00040
Units: 64	*MAE:* 0.00556	*Val_MAE:* 0.03939	*Test_MAE:* 0.01163
Dropout: 0.2	*RMSE:* 0.00783	*Val_RMSE:* 0.08826	*Test_RMSE:* 0.01992
Optimizer: Adam	R^2: 0.97813	*Val_*R^2: 0.77235	*Test_*R^2: 0.06451
LR: 0.001			
Batch: 32			

Table 8: Configuration 5

Training Config	Training Results	Validation Results	Test Results
Epochs: 42	*Loss:* 0.00067	*Val_Loss:* 0.01421	*Test_Loss:* 0.00035
Units: 64	*MAE:* 0.01892	*Val_MAE:* 0.05346	*Test_MAE:* 0.01188
Dropout: 0.2	*RMSE:* 0.02593	*Val_RMSE:* 0.11920	*Test_RMSE:* 0.01882
Optimizer: RMSprop	R^2: 0.76042	*Val_*R^2: 0.58479	*Test_*R^2: 0.16509
LR: 0.001			
Batch: 32			

Table 9: Configuration 6

Training Config	Training Results	Validation Results	Test Results
Epochs: 44	*Loss:* 0.00122	*Val_Loss:* 0.02133	*Test_Loss:* 0.00077
Units: 64	*MAE:* 0.02842	*Val_MAE:* 0.06652	*Test_MAE:* 0.02619
Dropout: 0.2	*RMSE:* 0.03490	*Val_RMSE:* 0.14605	*Test_RMSE:* 0.02775
Optimizer: RMSprop	R^2: 0.56587	*Val_*R^2: 0.37662	*Test_*R^2: -0.81475
LR: 0.01			
Batch: 32			

Table 10: Configuration 7

Training Config	Training Results	Validation Results	Test Results
Epochs: 55	*Loss:* 0.00017	*Val_Loss:* 0.01073	*Test_Loss:* 0.00165
Units: 64	*MAE:* 0.00909	*Val_MAE:* 0.05525	*Test_MAE:* 0.03528
Dropout: 0.2	*RMSE:* 0.01312	*Val_RMSE:* 0.10359	*Test_RMSE:* 0.04058
Optimizer: RMSprop	R^2: 0.93869	*Val_*R^2: 0.68640	*Test_*R^2: -2.88182
LR: 0.0001			
Batch: 32			

Table 11: Configuration 8

Training Config	Training Results	Validation Results	Test Results
Epochs: 100	*Loss:* 0.00078	*Val_Loss:* 0.03393	*Test_Loss:* 0.00289
Units: 64	*MAE:* 0.01487	*Val_MAE:* 0.10111	*Test_MAE:* 0.04244
Dropout: 0.2	*RMSE:* 0.02788	*Val_RMSE:* 0.18420	*Test_RMSE:* 0.05375
Optimizer: SGD	R^2: 0.72296	*Val_*R^2: 0.00850	*Test_*R^2: -5.81000
LR: 0.01			
Batch: 32			

Table 12: Configuration 9

Training Config	Training Results	Validation Results	Test Results
Epochs: 100	*Loss:* 0.00016	*Val_Loss:* 0.01243	*Test_Loss:* 0.00496
Units: 64	*MAE:* 0.00916	*Val_MAE:* 0.06911	*Test_MAE:* 0.06674
Dropout: 0.2	*RMSE:* 0.01252	*Val_RMSE:* 0.11148	*Test_RMSE:* 0.07040
Optimizer: SGD	R^2: 0.94411	*Val_*R^2: 0.63685	*Test_*R^2: -10.68321
LR: 0.01			
Batch: 32			

Table 13: Configuration 10

Training Config	Training Results	Validation Results	Test Results
Epochs: 100	*Loss:* 0.00075	*Val_Loss:* 0.02351	*Test_Loss:* 0.00082
Units: 64	*MAE:* 0.01607	*Val_MAE:* 0.07065	*Test_MAE:* 0.02237
Dropout: 0.2	*RMSE:* 0.02747	*Val_RMSE:* 0.15333	*Test_RMSE:* 0.02866
Optimizer: SGD	R^2: 0.73120	*Val_*R^2: 0.31295	*Test_*R^2: -0.93624
LR: 0.001			
Batch: 32			

Table 14: Configuration 11

Training Config	Training Results	Validation Results	Test Results
Epochs: 71	*Loss:* 0.00010	*Val_Loss:* 0.00634	*Test_Loss:* 0.00039
Units: 64	*MAE:* 0.00758	*Val_MAE:* 0.03821	*Test_MAE:* 0.01505
Dropout: 0.2	*RMSE:* 0.01002	*Val_RMSE:* 0.07964	*Test_RMSE:* 0.01969
Optimizer: Adam	R^2: 0.96421	*Val_*R^2: 0.81465	*Test_*R^2: 0.08643
LR: 0.001			
Batch: 16			

Table 15: Configuration 12

Training Config	Training Results	Validation Results	Test Results
Epochs: 37	*Loss:* 0.00014	*Val_Loss:* 0.01053	*Test_Loss:* 0.00035
Units: 64	*MAE:* 0.00850	*Val_MAE:* 0.04787	*Test_MAE:* 0.00994
Dropout: 0.2	*RMSE:* 0.01179	*Val_RMSE:* 0.10261	*Test_RMSE:* 0.01879
Optimizer: Adam	R^2: 0.95045	*Val_*R^2: 0.69230	*Test_*R^2: 0.16789
LR: 0.001			
Batch: 64			

Table 16: Configuration 13

Training Config	Training Results	Validation Results	Test Results
Epochs: 31	*Loss:* 0.00012	*Val_Loss:* 0.01075	*Test_Loss:* 0.00030
Units: 128	*MAE:* 0.00702	*Val_MAE:* 0.04805	*Test_MAE:* 0.01340
Dropout: 0.2	*RMSE:* 0.01109	*Val_RMSE:* 0.10368	*Test_RMSE:* 0.01733
Optimizer: Adam	R^2: 0.95618	*Val_*R^2: 0.68585	*Test_*R^2: 0.29169
LR: 0.001			
Batch: 32			

Table 17: Configuration 14

Training Config	Training Results	Validation Results	Test Results
Epochs: 55	*Loss:* 0.00009	*Val_Loss:* 0.01155	*Test_Loss:* 0.00058
Units: 32	*MAE:* 0.00685	*Val_MAE:* 0.04244	*Test_MAE:* 0.02152
Dropout: 0.2	*RMSE:* 0.00967	*Val_RMSE:* 0.10746	*Test_RMSE:* 0.02401
Optimizer: Adam	R^2: 0.96670	*Val_*R^2: 0.66254	*Test_*R^2: -0.35871
LR: 0.001			
Batch: 32			

Table 18: Configuration 15

Training Config	Training Results	Validation Results	Test Results
Epochs: 30	*Loss:* 0.00009	*Val_Loss:* 0.01175	*Test_Loss:* 0.00020
Units: 64	*MAE:* 0.00680	*Val_MAE:* 0.05335	*Test_MAE:* 0.01015
Dropout: 0.3	*RMSE:* 0.00964	*Val_RMSE:* 0.10838	*Test_RMSE:* 0.01427
Optimizer: Adam	R^2: 0.96688	*Val_*R^2: 0.65676	*Test_*R^2: 0.51966
LR: 0.001			
Batch: 32			

Table 19: Configuration 16

Training Config	Training Results	Validation Results	Test Results
Epochs: 77	*Loss:* 0.00004	*Val_Loss:* 0.00635	*Test_Loss:* 0.00041
Units: 64	*MAE:* 0.00394	*Val_MAE:* 0.03356	*Test_MAE:* 0.01292
Dropout: 0.1	*RMSE:* 0.00635	*Val_RMSE:* 0.07969	*Test_RMSE:* 0.02014
Optimizer: Adam	R^2: 0.98562	*Val_*R^2: 0.81442	*Test_*R^2: 0.04415
LR: 0.001			
Batch: 32			

4.2. Best Configuration

The figure 9 visualizes the high-dimensional relationship between the model's hyperparameters—specifically Learning Rate, Batch Size, Units, and Dropout—and its final performance metric (Test R²). Each line represents a single training run, color-coded to indicate success: red lines denote poor performance, while green lines indicate high accuracy.
Figure 6 provides a comparative analysis of model accuracy (y-axis, Test R²) against computational cost (x-axis, Training Time in seconds). Each data point represents a trained model configuration, with the size of the marker corresponding to the model's complexity (number of LSTM units).

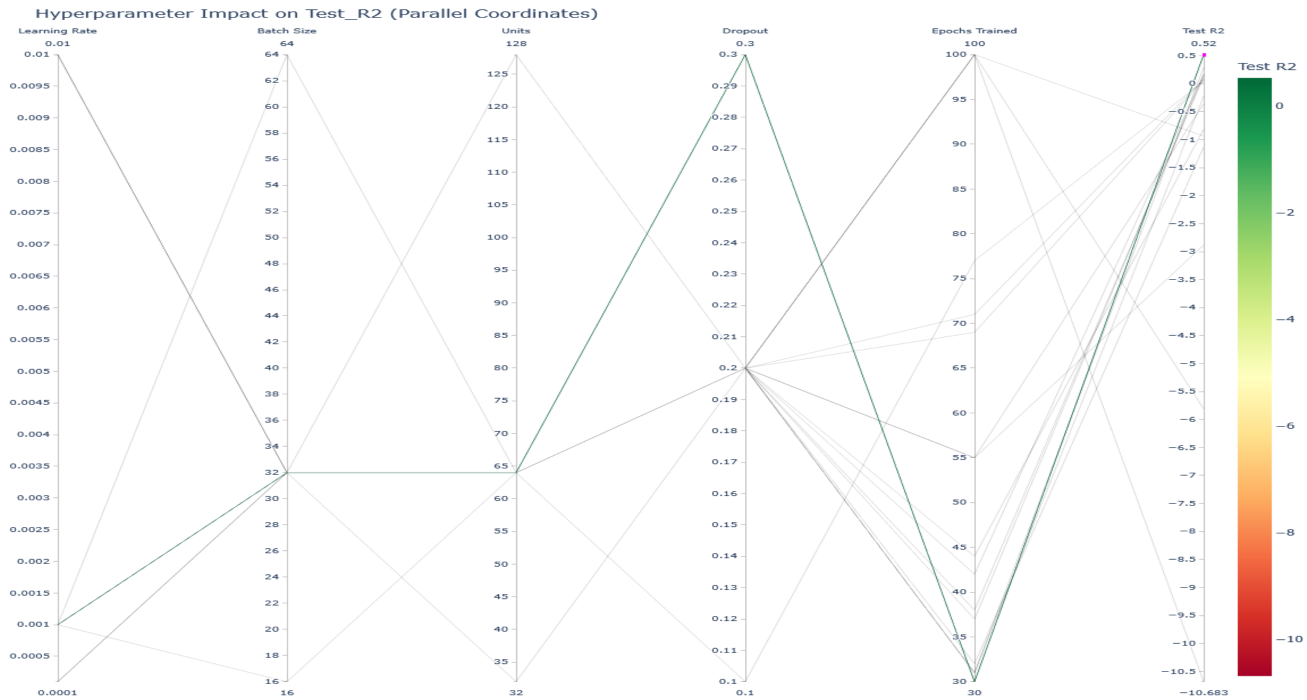


Fig. 5: Hyperparameter Parallel Coordinates

By tracing the path of the "best" line (configuration 15 [18]), we can observe the specific combination of settings that maximized the R² score. This visualization is critical for identifying stability patterns, such as whether high dropout rates consistently degrade performance or if larger unit counts are necessary for capturing complex temporal dependencies in the Dengue dataset.

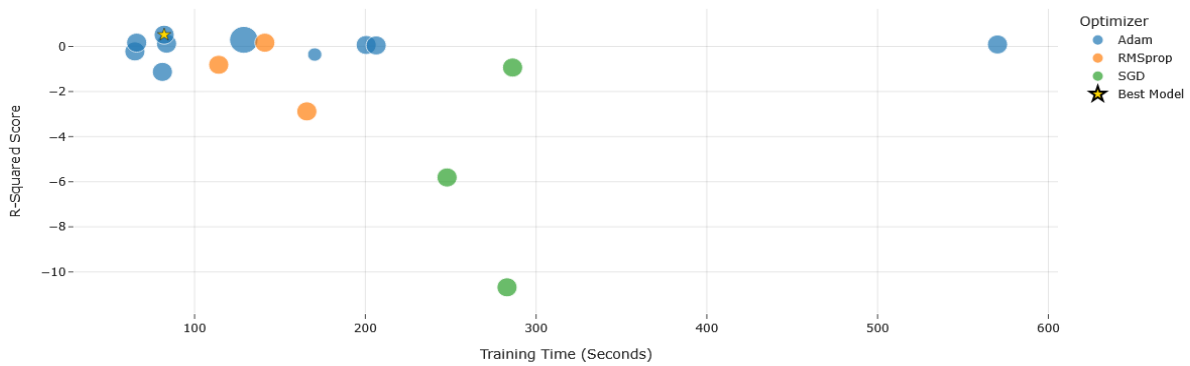


Fig. 6: Hyperparameter Parallel Coordinates

The gold star explicitly marks configuration 15 [18], identifying it as the optimal trade-off between performance and efficiency. Importantly, the color-coding reveals a distinct hierarchy in optimizer performance: the Adam optimizer consistently achieved the top results and was used by the best model, followed by RMSprop in second place. In contrast, SGD (Stochastic Gradient Descent) produced the lowest R² scores, demonstrating that adaptive learning rate methods are significantly more effective for this specific forecasting task.

5. System Implementation Results

The HealthTrace application visualizes complex epidemiological data through three primary components, designed to provide actionable insights for health officials.

5.1. Forecast Analysis Graph

This component serves as the primary visualization for the disease outbreak timeline, bridging historical data with future predictions.

- Historical Context: Displays the confirmed disease cases from the preceding 30 days, providing a baseline for the current epidemiological trend.
- Predictive Model Output: Projects the expected number of cases for the next 14 days.

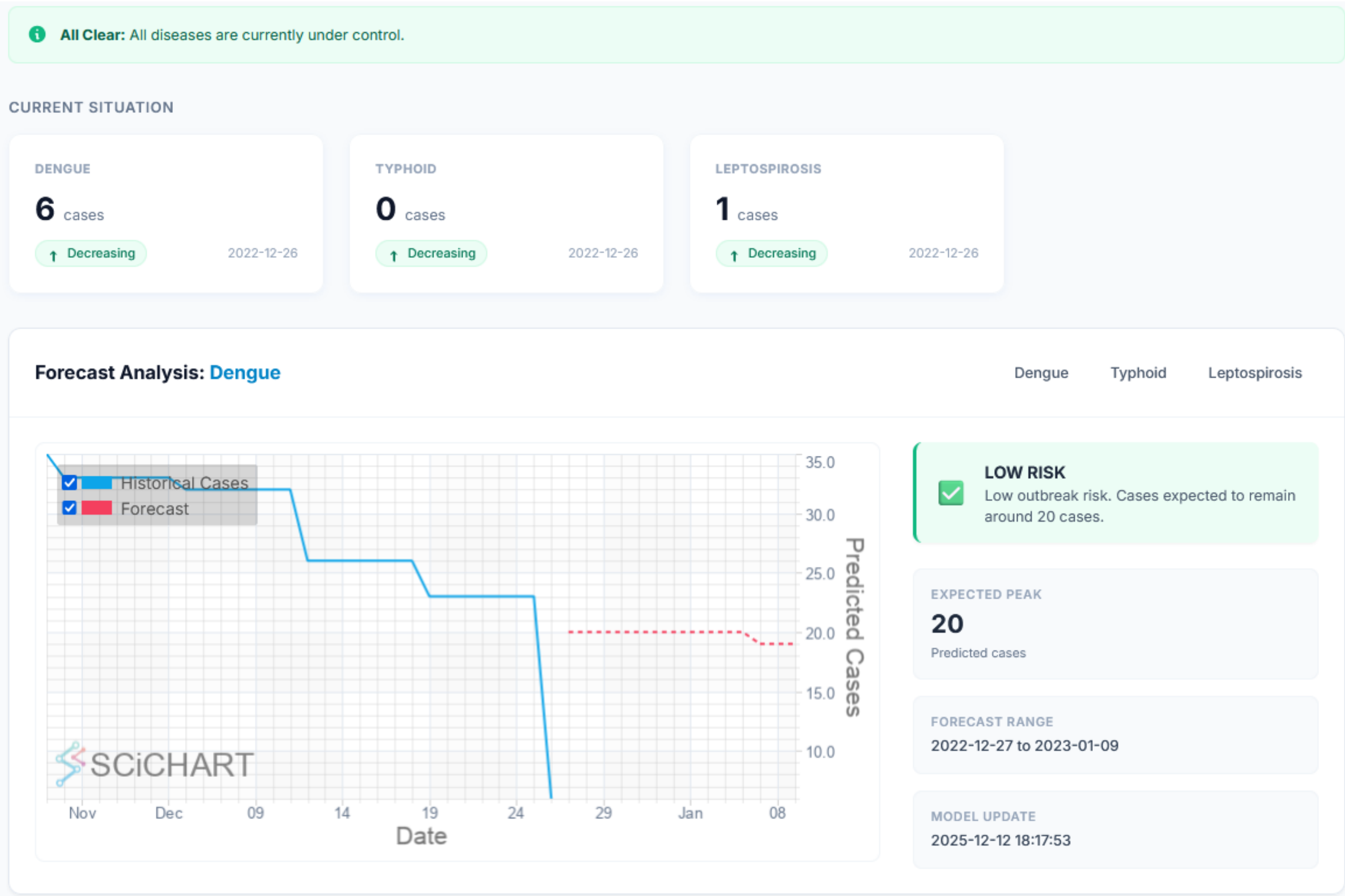


Fig. 7: Forecast Analysis

5.2. Feature Impact Analysis

Understand which environmental factors are driving the model's prediction for this specific timeframe. This module breaks down the "black box" of the AI model by quantifying the relationship between external variables and disease transmission.

- Uses horizontal bar charts to represent the correlation strength (Impact %) of various features.
- Features are organized into distinct categories for readability, including Climate & Precipitation, Socioeconomic Factors, Air Quality, and Sanitation & Water Access.

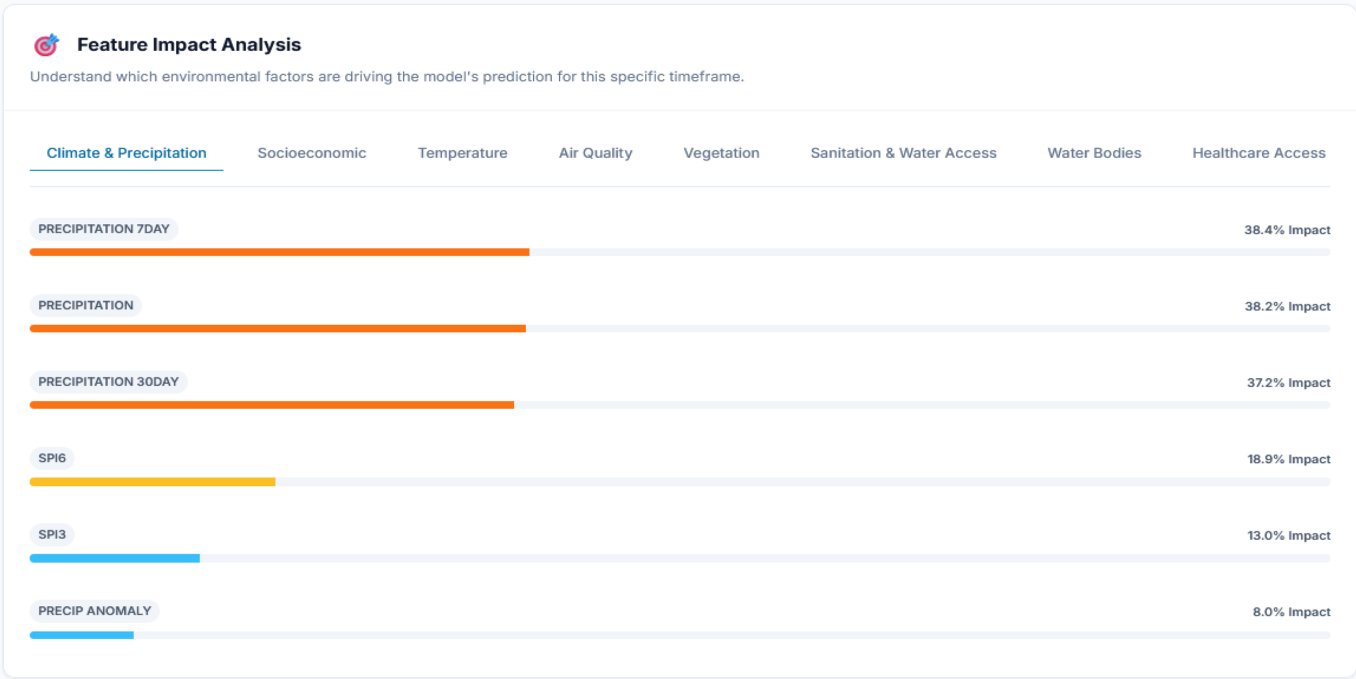
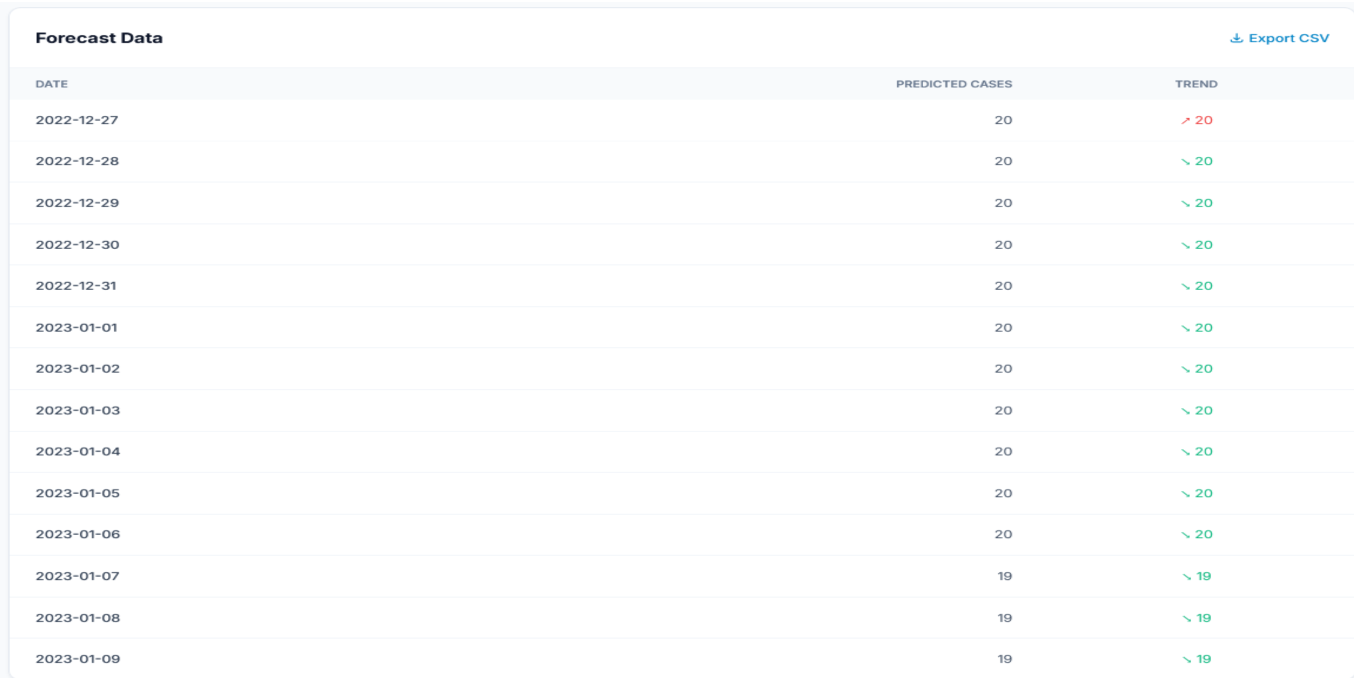


Fig. 8: Feature Impact

5.3. Forecast Data

While the graphs provide trends, this section provides the concrete numerical data required for resource allocation and reporting.

- Forecast Table: A tabular view listing the specific date and the exact number of predicted cases (rounded to the nearest integer) for the entire 14-day forecast window.



Forecast Data — Export CSV

DATE	PREDICTED CASES	TREND
2022-12-27	20	↗ 20
2022-12-28	20	↘ 20
2022-12-29	20	↘ 20
2022-12-30	20	↘ 20
2022-12-31	20	↘ 20
2023-01-01	20	↘ 20
2023-01-02	20	↘ 20
2023-01-03	20	↘ 20
2023-01-04	20	↘ 20
2023-01-05	20	↘ 20
2023-01-06	20	↘ 20
2023-01-07	19	↘ 19
2023-01-08	19	↘ 19
2023-01-09	19	↘ 19

Fig. 9: Forecast Data

6. Conclusion

This study successfully established "HealthTrace," a functional deep learning application capable of forecasting disease outbreaks in Iloilo City up to 14 days in advance. By integrating 52 distinct environmental and socioeconomic features—ranging from precipitation anomalies to healthcare access indices—we demonstrated that deep learning models can effectively capture the complex, non-linear dependencies between climate patterns and disease incidence.

Our rigorous hyperparameter tuning process identified that a 2-layer LSTM architecture with 64 units and a dropout rate of 0.3, optimized via Adam, provided the most robust generalization (Configuration 15), achieving a Test R^2 of 0.52. The comparative analysis conclusively showed that while GRU architectures offer computational efficiency, the LSTM's ability to retain long-term dependencies proved superior for this specific time-series regression task. Furthermore, the poor performance of the SGD optimizer underscores the necessity of adaptive learning rate methods for high-dimensional health data.

The deployed web application bridges the gap between complex data science and public health utility, offering officials a clear, interactive dashboard for proactive decision-making. Future enhancements will focus on integrating real-time data feeds from local health authorities, exploring ensemble modeling techniques to further improve accuracy, and expanding the system to support multi-region forecasting.

Code Availability

The full source code, datasets, and training scripts are available in the project repository:

GitHub:
`https://github.com/KirkGamo/HealthTrace`

Project CCHAIN Dataset:
`https://www.kaggle.com/datasets/thinkdatasci/project-cchain`