Forecasting Dengue Incidences in Bangladesh: A Univariate Time Series Approach

By

Shahidul Islam

ID: 0722210005101007

Supervisor:

Md. Rifat Hassan

Lecturer

Fall 2023

Dissertation submitted in partial fulfillment for the degree of Bachelor of Science in Computer Science and Engineering

Department of Computer Science & Engineering

Fareast International University

Letter of Transmittal

December, 2023

Md. Rifat Hassan
Lecturer
Department of Computer Science & Engineering
Fareast International University
Dhaka, Bangladesh

Subject: Submission of thesis paper on **"Forecasting Dengue Incidences in Bangladesh: A univariate Time Series Approach".**

Dear Sir,

With due respect, I would like to submit my thesis report on **'Forecasting Dengue Incidences in Bangladesh: A Univariate Time Series Approach'** as a part of my BSc program. The report analyzes the intricate patterns of the dengue virus within the context of Bangladesh and suggests prediction models to forecast the occurrence of dengue fever for the following year.

I sincerely appreciate your permission to compile this report, and I have done our utmost to adhere to your instructions. I would be delighted if you found this work to be insightful and helpful in gaining an awareness of the situation.

Sincerely,
Shahidul Islam
ID: 0722210005101007

APPROVAL

Shahidul Islam (ID # 0722210005101007) from the Department of Computer Science & Engineering of Fareast International University has worked on the Final Year Project titled **"Forecasting Dengue Incidences in Bangladesh: A univariate Time Series Approach"** under the supervision of Md. Rifat Hassan in partial fulfillment of the requirement for the degree of Bachelor of Science in Engineering and the thesis has been accepted as satisfactory.

Supervisor's Signature

…….

Md Rifat Hasan
Lecturer
Department of Computer Science & Engineering
Fareast International University
Dhaka, Bangladesh.

Head of the Department's Signature

…….

Dr. Fauzia Yasmeen
Associate Professor
Department of Computer Science & Engineering
Fareast International University
Dhaka, Bangladesh.

DECLARATION

I hereby affirm that this project is a product of original effort. No portion of this project has been presented elsewhere, in part or in whole, for the attainment of any other degree or certification. All information pertaining to this project will be kept confidential and will not be revealed without the explicit approval of the project supervisor. Previous works relevant to this report have been duly recognized and referenced. The guidelines on plagiarism as outlined by the supervisor have been adhered to.

Signature:

Shahidul Islam

ID: 0722210005101007

ACKNOWLEDGEMENTS

The author wishes to convey his sincere appreciation towards every living organism thriving every moment to make the world a better place.

Shahidul Islam
Department of CSE
Fareast International University
Banani, Dhaka.

Abstract

Dengue fever, an acute health concern in Bangladesh, has dramatically increased in frequency and severity in recent years. The dengue epidemic experienced in 2023 represented an unprecedented peak in the incidence of the disease within the nation's recorded history. In this study, our main goal was to create forecasting models for dengue-affirmed cases in the unique context of Bangladesh, primarily driven by the escalating health crisis of the dengue epidemics. We examined historical confirmed cases of dengue infection in Bangladesh from 2008 to 2023 to comprehend the underlying pattern and seasonal fluctuations with the help of univariate time series analysis. We further proposed the SARIMA, Holt-Winters, Prophet, and LSTM models for dengue prediction. The novel aspect of this research was to introduce the utilization of the Holt-Winters method and new parameters for the SARIMA model, within Bangladesh's particular circumstances of dengue virus. To identify the best model, each one was assessed using simple evaluation metrics like MAE and RMSE. The most effective predictive model indicated a potential surge in future dengue cases that exceeded any previously recorded outbreaks, underscoring the critical need for immediate policy intervention. Finally, the paper concludes with future research scope to enhance dengue surveillance and response. We anticipate that the insights gained from this study will be instrumental in enabling proactive measures to effectively address upcoming dengue outbreaks.

Keywords: Dengue Forecast, Bangladesh, Time Series Analysis, SARIMA, Holt-Winters.

Table of Contents

List of Figures

List of Tables

List of Acronyms

RNA- Ribonucleic Acid

DENV- Dengue Virus

DF- Dengue Fever

AD- Anno Domini

WHO- World Health Organization

DGHS- Directorate General of Health Services

MAE- Mean Absolute Error

MSE- Mean Squared Error

RMSE- Root Mean Squared Error

AIC- Akaike's Information Criterion

BIC- Bayesian Information Criterion

ARIMA- Autoregressive Integrated Moving Average

ETS- Error, Trend, Seasonal

ACF- Autocorrelation Function

PACF- Partial Autocorrelation Function

ADF- Augmented Dickey-Fuller

SARIMA- Seasonal Autoregressive Integrated Moving Average

COVID- Coronavirus Disease

OGM- Online Gradient Method

SARS- severe acute respiratory syndrome

LSTM- Long Short-Term Memory

HFMD- Hand, Foot, and Mouth Disease

HWES- Holt-Winters Exponential Smoothing

Chapter 1

Introduction

This chapter is the introductory phase of our research study. It highlights the historical background of dengue, especially in Bangladesh. We also embark on the importance of forecasting dengue cases. This section also encompasses the study's contributions and wraps up with an outline of the research structure.

1.1 Background

Dengue, a mosquito-borne viral infection, is a rising global public health concern. It is mostly an urban tropical disease caused by the viruses that circulate in a cycle involving human hosts and the *Aedes aegypti* mosquitoes feeding on them [1]. Dengue is carried by one of the four single-stranded, positive sense RNA viruses of the genus *Flavivirus* (family *Flaviviridae*), generally known as serotypes (DENV 1 through 4) [2]. Somehow, infection by any of these serotypes fails to confer cross-protective immunity, heightening the risk of experiencing more aggressive manifestations, such as dengue shock syndrome (DSS) and dengue hemorrhagic fever (DHF) [3]. Historical data indicates a prolonged interaction between these viruses and mankind [4]. Symptoms resembling dengue have been documented as far back as the Chin Dynasty, spanning the years 265–420 AD [5].

Dengue fever outbreaks were initially documented in 1779–1780 in North America, Africa, and Asia; the nearly simultaneous emergence of epidemics across three continents suggests that these viruses, along with their mosquito vectors, have been widely distributed in tropical regions for more than 200 years [1]. The frequency of dengue has increased thirtyfold in the last fifty years and currently, over half of the world's population resides in areas spanning over 100 countries where there is a risk of DENV (dengue virus) infection [6]. An estimated 390 million dengue virus infections happen each year, of which 67–136 million are thought to be clinical cases [7]. While dengue represents a worldwide health challenge, marked by an increasing tally of affected countries, around 75% of the global population susceptible to dengue is concentrated in the Asia-Pacific region [8].

1.2 Dengue Epidemic in Bangladesh

Notably, about 52% of the worldwide population vulnerable to dengue is located in 10 countries within the WHO South-East Asia Region, with Bangladesh being one of these nations [9]. Initially reported in the 1960s as “Dacca fever” in what was then East Pakistan, now Bangladesh, dengue has shown a trend since 2010 of correlating with the May to September rainy season and elevated temperatures. Dengue reports were intermittent from 1964 to 1999, leading up to the first significant outbreak in 2000, which recorded 5,551 hospitalized cases and 93 fatalities [10]. The 2019 dengue outbreak saw a total of 100,201 confirmed dengue cases where 51,179 cases were reported in the capital city Dhaka and 49,022 across the rest of the country [11]. The outbreak in 2000 was the second largest in Bangladesh's history, surpassed only by the 2023 epidemic, the most catastrophic dengue fever outbreak ever recorded in the nation.

In 2023, the Directorate General of Health Services (DGHS) reported a total of 321,073 dengue cases in Bangladesh. Of these, 109,973 cases (34%) were in Dhaka, while the remaining 211,100 (66%) were reported from areas outside Dhaka. The resurgence of dengue have significantly increased health burdens, exacerbating morbidity and mortality rates, amid inadequate resources for the Health, Population, and Nutrition Sector Program (HPNSP) [12]. Several factors contribute to creating habitats for dengue vectors, including high densities of infected mosquitoes, varying immunity levels in people to different dengue serotypes, poor housing conditions, inadequate waste management, sanitation, and drainage systems [13], [14]. With its abundant rainfall and comfortable temperatures, Bangladesh's tropical monsoon environment dramatically increases mosquito density, which raises the incidence of dengue cases [9], [13]. Regrettably, dengue currently lacks a specific cure or vaccination [15]. Besides, dengue prevention efforts in Bangladesh have been hampered by a lack of effective vector control methods and weak disease surveillance systems, limiting the ability to prevent transmission and manage outbreaks effectively.

1.3 Importance of Dengue Forecasting

In today's world, being well-prepared for a disease outbreak requires a precise risk assessment [16]. Significant efforts have been dedicated to creating early warning systems for timely detection and control of major dengue epidemics, considering the high disease burden and forecasting challenges [17] [18]. Historical patterns of dengue outbreaks in Bangladesh indicate that initiating mosquito control measures earlier in the transmission season could effectively reduce both the monthly growth rate and overall scale of dengue epidemics [19]. So, forecasting dengue cases extends beyond mere prediction; it is a crucial tool in shaping public health responses and policies.

Although, for infectious diseases, weekly or monthly dengue epidemic trajectory forecasting is still a relatively young field of study [20]. Unlike the more predictable dynamics governed by physical laws in weather and climate forecasting, the complex social and biological factors driving dengue epidemics make their prediction particularly challenging [21]. Despite the obstacles, researchers are strenuously allocating resources effectively for anticipating outbreaks.

These proactive approaches not only helps in curbing the immediate impact of dengue outbreaks but also aids in minimizing the long-term public health and economic burdens.

Furthermore, forecasting dengue cases enriches research, contributing to the development of more targeted and efficient strategies to combat the disease. In endemic regions like Bangladesh, where dengue poses a significant threat, these forecasts are indispensable for planning and executing public health initiatives. They serve as a cornerstone in devising comprehensive strategies that encompass not just immediate outbreak response but also longer-term dengue control and prevention programs.

1.4 Motivation

The motivation behind this study stems from the growing concern over the rising incidence of dengue in Bangladesh and its significant impact on public health. The unpredictable nature of dengue epidemics, compounded by the challenges in early detection and management, underscores the need for a more robust approach to predict and control dengue outbreaks. This study is driven by the goal to contribute towards a better understanding of dengue dynamics and to develop effective strategies for its prediction and prevention.

Another driving factor for this study is the limited availability of reliable predictive models for dengue outbreaks in the context of Bangladesh. Existing models often do not fully account for the unique environmental and sociocultural factors prevalent in the region. There is a critical need for models that can accurately predict dengue outbreaks, enabling health authorities to prepare and respond more effectively. Furthermore, the study is motivated by a commitment to public health and the well-being of communities in Bangladesh. The ability to forecast dengue outbreaks accurately can significantly enhance the effectiveness of public health interventions, reducing both the incidence and severity of the disease. This not only translates to better health outcomes but also to economic savings for both individuals and the healthcare system.

1.5 Objectives of the Study

The overarching objectives of this study is to advance the understanding and management of dengue fever in Bangladesh through innovative research and practical applications. We aim to contribute novel insights and methodologies in the field of dengue forecasting and control. Specific contributions include:

- Investigate the impact of urbanization and environmental changes in Bangladesh, on the spread and intensity of dengue outbreaks.

- Delve into prior studies on the dengue virus to extract key findings and draw inspiration for further exploration.
- Conduct a comprehensive analysis of the historical dengue case data in Bangladesh.
- Identify underlying patterns, temporal dynamics and seasonal fluctuations of DENV within the context of Bangladesh, with a focus on discerning long-term trends that characterize the epidemiology of the disease in the region.
- Develop optimal univariate time series models that accurately forecast dengue cases in Bangladesh.
- Evaluate the performance of the models using relevant statistical metrics and compare each model to find the superlative one.
- Explore the practical applications of the model in public health planning and dengue control strategies in Bangladesh.
- Foster international collaboration by sharing insights and methodologies with the global health community, contributing to the worldwide fight against dengue.
- Assess the effectiveness of current dengue prevention and control measures implemented in Bangladesh, identifying gaps and opportunities for enhancing public health strategies.
- Acknowledge limitations and therefore provide future research suggestions to enhance the predictive accuracy and applicability of the findings.

By achieving these goals, the study hopes to lessen the impact of dengue on the country's population by promoting efficient management and control of the disease in Bangladesh.

1.6 Organization of the Research

This thesis is organized into seven main chapters, each focusing on different aspects of our study. Following this introductory chapter, the structure of the report is as follows.

- **Chapter 2: Literature Review** – This chapter provides a comprehensive review of existing literature related to dengue epidemiology.
- **Chapter 3: Methodology** – This section delineates the research methodologies employed to execute the comprehensive study.
- **Chapter 4: Implementation** – In this chapter, we provide outline and practical implications of the methods necessary for building the forecasting models.
- **Chapter 5: Results** – We delve into the outcomes of our research along with critical discussion of the findings in this pivotal chapter.
- **Chapter 6: Limitations & Future Works** – This segment includes the challenges, and limitations of this paper and also give suggestions for future research endeavor.
- **Chapter 7: Conclusion** – This chapter provides a concluding summary of the study.

Chapter 2

Literature Review

In this chapter, we meticulously explore the extensive field of research relevant to forecasting cases of infectious diseases, with a deep focus on dengue epidemiology. This literature review seeks to build a bridge between past research and the current study, highlighting the evolution and gaps in the field of forecasting dengue-affected cases.

2.1 Overview of Existing Research

Several studies have contributed to the field of epidemic forecasting, each offering a distinct perspective on the dynamics of disease progression. In this context, time series analysis has emerged as a highly pivotal instrument, also pertinent in dengue forecasting.

The spread of Dengue Virus (DENV) has a high correlation with climate variables like rainfall, temperature, and humidity and it has been established by several studies [22], [23], [24] based on different geolocations. Likewise, Banu et al. (2014) [25] demonstrated how weather variations affected dengue transmission in Dhaka, Bangladesh, and estimated the potential threat of dengue that would arise in the future due to climate change. This research involved gathering data on the monthly incidences of dengue in Dhaka from January 2000 to December 2010, alongside monthly weather data for the same period. The authors defined the association between dengue transmission and meteorological factors using Spearman's correlation coefficients. Furthermore, a Poisson time series model was developed in conjunction with a distributed lag model (DLM) to evaluate the influence of meteorological factors on dengue transmission. The findings demonstrated that humidity and temperature, in particular, were strongly correlated with the spread of DENV.

So, one of the strategies researchers have opted for is to use climatic data-based models for predicting dengue outbreaks. To improve decision-making regarding the extent of vector control measures, Hii et al. (2012) [26] proposed an early forecast system capable of anticipating dengue cases and delivering timely warnings up to 16 weeks in advance. A Poisson multivariate regression model was created utilizing weekly average temperature and total precipitation in Singapore for the years 2000-2010. The standardized Root Mean Square Error (RMSE), residual diagnosis, and Akaike’s Information Criteria (AIC) were used for model selection and evaluation. According to the study, the frequency of dengue was most prominent from June to October in Singapore. Although the study focused on the relationship between temperature,

rainfall, and dengue incidence, it did not explore the potential influence of population density, socioeconomic factors, or human mobility patterns.

In contrast to contemplating the positive association between weather and dengue, some researchers have depended on a univariate approach of time series forecasting based solely on the previous years' affected cases. Naher et al. (2022) [27] highlighted the importance of early warning systems for controlling dengue epidemics in Bangladesh and emphasized on the need for proactive measures to minimize the sufferings caused by outbreaks. A secondary data set of monthly dengue cases from January 2008 to January 2020 was taken into consideration in this investigation. Initially, the researchers considered the Autoregressive Integrated Moving Average (ARIMA), Error, Trend, Seasonal (ETS), and Trigonometric seasonality, Box-Cox transformation, ARMA errors, Trend, and Seasonal (TBATS) model for forecasting purposes. The preliminary order of the ARIMA model was determined by analyzing graphs from the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF), and the stationarity of the time series data was tested using the Augmented Dickey-Fuller (ADF) test. Subsequently, Akaike's Information Criterion (AIC) served as the definitive criterion for model selection. After checking the accuracy of the fitted models with Root Mean Square Percentage Error (RMSPE), Mean Percent Forecast Error (MPFE), and Theil Inequality Coefficient (TIC), ARIMA (2,1,2) turned out to be the superlative predictive model. However, the absence of variation in dengue incidence frequency and incorrect detection of asymptotic infections were two of this model's shortcomings.

Among the statistical models, ARIMA and its seasonal counterpart Seasonal Autoregressive Integrated Moving Average (SARIMA) have been extensively used as viable predictive models to forecast infectious diseases. These models were adopted by ArunKumar et al. (2021) [28] in their study to estimate the COVID-19 pandemic's epidemiological trends for the top 16 countries, which account for 70–80% of the total cases worldwide. The open-source COVID-19 database at John Hopkins University provided the data utilized in this study which spanned from January 22nd,2020 to July 24th,2020. The study found that ARIMA models were outperformed by the SARIMA models in predictability. The evaluation was carried out using reliable evaluation metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE).

Mrityunjay Panda (2020) [29] introduced the Holt-Winters Exponential Smoothing (HWES) model along with ARIMA to predict the spreading of COVID-19 in India. The required data spanning from 30th January to 29th June 2020 was taken from Kaggle, and it was also released on the official website of the Ministry of Health and Family Welfare under the Government of India. The research discovered that ARIMA (4,1,1), and ARIMA (3,1,1) are the most vigorous models with the lowest Akaike Information Criterion (AIC) value for forecasting confirmed and mortality cases in the scenario of India and its states. The authors conducted multiple attempts to

verify the optimal values for the smoothing parameters α and β of the Holt-Winters model. With low AIC and RMSE, the forecasting results using α=0.9, and β=0.3 yielded the best result for daily confirmed cases, while α=0.5, and β=0.7 gave the most optimal results for daily mortality. However, in this study, the SARIMA model was not considered or evaluated, which had been established as the more powerful method to forecast COVID-19 cases according to another study [28].

Apart from pure statistical models, neural networks (NN) based deep learning models have become sufficiently good at predicting and analyzing the complex patterns and uncertainty of infectious diseases. A study by Bai et al. (2005) [30] stated that the SARS pandemic was better predicted using backward propagation (BP) neural networks which enhance current computational techniques and increase forecast accuracy. A variation of the standard gradient method called the Online Gradient Method (OGM) was implemented to accelerate the convergence speed. It was additionally stated that using OGM to network training required the incorporation of a stochastic mechanism. Finally, the authors found a number of predicting results of the SARS epidemic in Beijing and Shanxi, China.

In another research, Xu et al. (2020) [31] observed that the Long Short-Term Memory (LSTM) model fared better in forecasting dengue cases than already-published forecasting models, and the use of Transfer Learning (TL) may improve the model's capacity to draw conclusions about areas with fewer dengue cases. From January 2005 to December 2018, case-level monthly records of DENV infections and meteorological data of the top 20 cities of China were collected for model training and testing. To forecast dengue fever (DF), the study built an LSTM-based model and evaluated its efficiency using Root Mean Squared Error (RMSE) against that of other potential models, including Gradient Boosting Machine (GBM), Support Vector Regression (SVR), Back Propagation Neural Network (BPNN), and Generalized Additive Model (GAM). In comparison, the LSTM model lowered the average RMSE of the forecasts by 12.99% to 24.91% and the average RMSE of the predictions during the epidemic period by 15.09% to 26.82%.

Xie et al. (2020) [32] demonstrated the Prophet model's potential to analyze the trend of Hand, Foot, and Mouth Disease (HFMD) in Hubei province, China. The study found that the Prophet model performed more effectively than the ARIMA model in terms of daily case count of HFMD. Daily incidence of HFMD data in Hubei Province from 2009 to 2018 were collected from the Chinese Ministry of Health to be included in the whole dataset. The training set utilized by the authors was the HFMD incidence data from 2009 to 2017, whereas the test set consisted of data from 2018 to produce forecasting models that are robust while avoiding over-fitting. The prediction of ARIMA (4,1,4) (0,1,0, 365) failed the Ljung–Box test even though its AIC value was lower than that of ARIMA (5,1,3) (0,1,0, 365). As a result, it was determined that the ARIMA (5,1,3) (0,1,0, 365) model fitted the data the best. In the case of the Prophet model, the trend component was configured to follow a saturating growth pattern, and the carrying

capacity for the logistic growth model was established at 8.5. The change points selection was automatic with an interval with of 0.95 and the number of uncertainty samples as 1000. To compare both models, evaluation metrics like RMSE, MAE, and Pearson R values were computed. The findings demonstrated that both Prophet and ARIMA matched the training sets quite well, with ARIMA outperforming Prophet by a small margin. Nevertheless, ARIMA performed much worse than Prophet in the test sets. The study also established that pronounced seasonal influences in time series data and multiple periods of historical data are optimal for the Prophet model's performance. Although more comparative analysis with several other models was absent in this research, which was prominent in another study [27].

To create a reliable forecasting technology that would enable medical professionals to put into practice efficient monkeypox prevention measures, Long et al. (2023) [33] discussed the use of ARIMA, Prophet, NeuralProphet, and LSTM models. The study obtained historical data of monkeypox reported cases in the United States of America from 17th May 2022 to 14th September 2022. NeuralProphet performed exceptionally well according to the results, with an RMSE of 49.27 and a $R2$ score of 0.76, while exhibiting 95% accuracy based on sample instances. Additionally, as a consequence of data gathering techniques, the authors found that monkeypox, a highly infectious zoonotic disease, has a seasonal pattern and a sexually transmitted pattern in addition to following the SIR epidemic distribution. Consequently, the potential of NeuralProphet was also established in predicting illnesses like chickenpox, HIV, SARS, and COVID-19 which have common traits with monkeypox.

Satrio et al. (2021) [34] explored time series analysis and thus prediction of coronavirus disease in Indonesia with the help of the ARIMA model and Prophet. This study aimed to assess the performance of ARIMA and Prophet models in analyzing time-series data characterized by random patterns, absence of seasonality, and limited data points. The COVID-19 dataset from January 20th, 2020 to May 21st, 2020 was obtained from the Kaggle website, containing 27618 observations from various provinces around the world. Early on, the Prophet prediction data exhibits good accuracy with very little deviation from the real data. But as time goes on, the disparities tend to widen and become more apparent. However, both the real and Prophet results indicate an increasing tendency; however, Prophet displays a more linear increase in the recovered instances. The recovered cases section shows the largest discrepancy between the actual data and Prophet’s forecast. Nevertheless, ARIMA was a bit difficult to utilize. To fit the data into ARIMA, the ADF test, Log-scale transformation, and Time-shifting modification were employed. To obtain the p, d, q order of the model, ACF, PACF plots, and seasonal differencing methods were utilized. Despite this, Prophet continued to outperform ARIMA in comparison, even without any adjustments, with 91% precision. Furthermore, the Mean Forecast Error (MFE) of both models indicated that they were biassed positively, with the exception of the Prophet's forecast of the deaths section, where it showed negative bias.

To conclude, this review underscores the potential of SARIMA, HWES, Prophet, and LSTM models in forecasting epidemics, especially DF outbreaks. Furthermore, the distinct context of Bangladesh, characterized by its unique epidemiological and environmental factors, calls for an in-depth analysis of these models' applicability and effectiveness. In the DF forecast scenario of Bangladesh, the Holt-Winters method has still been an unexplored region up until now, which our study aims to delve into. We have also endeavored to find new parameters for the SARIMA model based on the lowest AIC score. To our substantial knowledge, our study is the first to analyze the Prophet model’s potential in forecasting dengue cases in Bangladesh. By filling these identified gaps, this research aims to contribute significantly to the field of dengue forecasting and public health planning in Bangladesh.

Chapter 3

Methodology

This chapter delineates the methodologies employed for data collection and the requisite preprocessing steps undertaken to ensure data integrity and analysis. Furthermore, it provides a concise overview of the methods pertaining to the SARIMA, HWES, Prophet, and LSTM forecasting models, elucidating their respective roles and applications within the context of this research.

3.1 Data Collection

We collected the required data on monthly dengue cases in Bangladesh, spanning from January 2008 to December 2023, from two primary sources: the Institute of Epidemiology, Disease Control and Research (IEDCR) and the Directorate General of Health Services (DGHS) website. This comprehensive dataset forms the basis for the subsequent analysis and modeling in this study.

3.2 Data Preprocessing

Given that the initial format of the data was unsuitable for Time Series Analysis, we resorted to using the Python programming language and its resourceful libraries to process the data and prepare it for additional analysis. What follows is the detailed inscription.

3.2.1 Data Preprocessing for Time Series Analysis

The original dataset, housed in an Excel file, was structured into a wide format where months were represented as rows and years as columns with each cell having the corresponding value of dengue cases. The dataset contained no missing values or null values. We imported the file into a Python environment using the Pandas library. Further refinement processes using the same library were employed which resulted in a streamlined chronological Data Frame consisting of two columns named ‘date’ and ‘cases’ respectively. Finally, we set the ‘date’ column as the index of the dataset. The culmination of these steps yielded a well-structured dataset, primed for time series analysis.

3.2.2 Data Transformation for Model Training

The dataset consisting of the ‘date’ and ‘cases’ columns was applicable for training the SARIMA model. However, data transformation steps were essential for the HWES, Prophet, and LSTM models.

For the prophet model, the transformation involved renaming the ‘date’ column to ‘ds’ in a ‘date-time’ format and the ‘cases’ column to ‘y’ in a numerical format. This renaming aligned with the Prophet model's expectation where it required inputs as a data frame with two column: one for the metric value and the other for timestamps, reflecting the quantity to be forecasted [35]. Such a transformation was critical for the model to correctly interpret the inputs, enabling it to effectively leverage its underlying algorithms for time series forecasting.

HWES model’s multiplicative seasonality method, used in this study, cannot handle zero values in time series data since the components of the time series are multiplied together to make forecasts. Since our data contained zero values for dengue cases of some months, every 0 value was replaced with a 1. Although this choice of substitution introduced a tiny bias in the data, it was a crucial part of error-free calculation and model training.

In the preparation of the dataset for LSTM model training, we employed a scrupulous data transformation approach. Initially, the dataset underwent a normalization process, using the ‘Scikit-learn’ library. This step ensured that all values were uniformly scaled within the range of 0 to 1, a prerequisite for the sensitivity of LSTM models to input data scale. Normalizing the data streamlines the analysis process and decreases the time required to run the model. [36]. It also helps in mitigating the risk of disproportionately large gradients which can destabilize the learning process. Subsequently, the transformed data was methodically restructured to form a sequence dataset conducive to LSTM training. Each sequence comprised a set of consecutive data points used as input features, with the subsequent data point serving as the label. This transformation catered to the LSTM's intrinsic requirement for time-stepped sequences, enabling the model to effectively capture temporal dependencies inherent in the time series data, thereby reinforcing the foundation for precise forecasting of dengue cases.

3.3 Data Visualization and Analysis

In the field of time series analysis, basic data visualization and decomposition techniques are essential tools for understanding underlying patterns and behaviors, especially when studying the number of dengue cases over time. The initial phase of our analysis focused on these aspects, employing graphical representations and decomposition methods to unravel the temporal dynamics of the dataset.

3.3.1 Time Series Graph

The time series plot produced with the help of the 'Matplotlib' library of Python, served as the cornerstone of this exploration phase. This simple yet powerful visualization provided an immediate sense of the trends, cycles, and irregularities present in the data. We could visually assess those aspects by plotting the number of dengue cases against time. This graphical representation served not just as a tool for preliminary assessment but also as a foundation for more detailed analytical approaches.

3.3.2 Seasonal-Trend Decomposition

Expanding upon the foundational understanding established through basic data visualization, the analysis progressed to a more nuanced exploration via seasonal-trend decomposition. Seasonal-trend decomposition of time series can unearth underlying patterns and offer clear decomposition outcomes, making it a valuable tool for eliminating seasonal components or forecasting future values in time series analysis. [37]. Therefore, we used this method in our study to methodically parse the intrinsic patterns embedded within our time series data. Using the Python library 'stats models', we sequentially dissected the time series data into its trend, seasonality, and residuals. By isolating these elements, we could better understand the persistent patterns, such as long-term increases or decreases (trend), and recurring patterns at fixed intervals (seasonality). Moreover, the residual component helped identify the unexplained variance, offering clues about the noise or data irregularities. Such decomposition not only reinforced the findings derived from the initial visual examination but also laid a more quantitatively robust groundwork for subsequent modeling and forecasting endeavors.

3.3.3 Autocorrelation Analysis

Autocorrelation analysis is an essential step in the exploratory data analysis of time series forecasting. A comprehensive visualization and interpretation of the Autocorrelation Function (ACF) and the Partial Autocorrelation Function (PACF) is indispensable in this segment, particularly in model identification for SARIMA modeling. Copious studies [38], [39], [40] have resorted to using the plots of these two functions to determine and analyze the parameter values of ARIMA and SARIMA-based models.

The Autocorrelation Function, commonly referred to as ACF, is used to measure and analyze the extent of correlation present within a time series relative to its previous values. It evaluates the linear correlation of a current observation in a time series and its preceding observations at specified time intervals, referred to as lags. An illustration of the correlation coefficients over various lags is provided by the ACF plot. On this graphic, every point represents the correlation between the time series and its own lagged values. The number of lags is shown on the x-axis, while the correlation coefficients are indicated on the y-axis. This plot's spikes above the confidence interval reveal considerable autocorrelation at certain lags and shed light on the data's cyclicality and repeating patterns. It is crucial for determining the order of Moving Average (MA) processes.

In contrast, the Partial Autocorrelation Function, or PACF, measures the correlation between a variable and its lag, while eliminating the impact of correlations at shorter lag intervals. This function is pivotal in revealing the pure correlation of a lag, absent of the effects of earlier lags. Like the ACF plot, it has lags on the x-axis and correlation coefficients on the y-axis. Significant spikes within the plot, which exceed the confidence intervals, suggest a strong partial correlation at those specific lags. This is particularly valuable in determining the order of Autoregressive (AR) terms in time series modeling.

In our investigation, the ‘stats models’ and ‘Matplotlib’ libraries in Python were employed to generate ACF and PACF graphs, with a specified lag value of 40. This number was chosen to provide a detailed view of the autocorrelations over an extended period, thus offering a deeper understanding of the data’s temporal structure. The interpretation of these plots was undertaken with careful consideration, particularly since the properties of autocorrelation in time series data can be nuanced and complex. The ACF and PACF analyses formed a cornerstone of our model selection process, particularly in the context of SARIMA modeling, allowing us to discern whether AR or MA terms (p, q, P, Q) were required, and if so, their respective orders.

3.3.4 Augmented Dickey-Fuller Test

The Augmented Dickey-Fuller test, or ADF test, is a statistical procedure known as a unit root test and it’s executed is to ascertain the intensity of a trend's influence on a time series [41]. The primary objective of the ADF test is to ascertain if a time series has stationarity. Stationarity means that the statistical characteristics of the time series such as mean, variance, and autocorrelation are unchanged across time. In our context, we used the ADF test and differencing of the time series data at various lags to conclude the non-seasonal and seasonal differencing parameters (d, D) of the SARIMA model.

3.4 Methods of the Forecasting Models

Upon the conscientious finalization of data acquisition, processing, and analytical examination, the focus of our research transitioned to the crucial stage of model training and predicting. Central to this segment of our methodology is an in-depth exploration of four distinct forecasting models: SARIMA, HWES, Prophet, and LSTM. This subsection will clarify the basic theoretical ideas and practical foundations of every chosen model.

3.4.1 SARIMA

SARIMA (Seasonal Auto Regressive Integrated Moving Average) is an advanced iteration of the ARIMA model, introduced by Box and Jenkins in 1976. ARIMA is capable of forecasting future values based on its past values [42]. The generalized form of the ARIMA model is given in Equation (1) [43],

$$\left(1-\sum_{i=1}^{p}\phi_i L^I\right)(1-L)^d X_t = \delta + \left(1+\sum_{i=1}^{q}\theta_i L^I\right)\varepsilon_t \quad (1)$$

Where L is the lag operator,

ϕ_i is the moving average part parameter, and

ε_t is the error term [43].

The Seasonal ARIMA model (SARIMA) is formed by adding seasonal terms in the ARIMA framework, as delineated in Equation (2),

$$\text{ARIMA}\ (p,d,q)\times(P,D,Q,S) \quad (2)$$

Here, p is the non-seasonal autoregressive order, d is the order of common difference, q is the non-seasonal moving average order, P is the seasonal autoregressive order, D is the order of seasonal difference, Q is the seasonal moving average order, and S is the length of the seasonal

cycle [43]. Nonetheless, the model can also be succinctly represented in a more mathematical form, as outlined in Equation (3) [44],

$$(1-\phi_1 B^{\omega} - \phi_2 B^{2\omega} - \cdots - \phi_p B^{P\omega}) \times (1 - \varphi_1 B - \varphi_2 B^2 - \cdots - \varphi_p B^{p}) \times (1 - B^{\omega})^{D}(1 - B)^{d} Q_n(t) = (1 - \Theta_1 B^{\omega} - \Theta_2 B^{2\omega} - \cdots - \Theta_{1Q} B^{Q\omega}) \times (1 - \phi_1 B - \theta_2 B^2 - \cdots - \theta_q B\omega^{q}) e(t) \quad (3)$$

Where, ϕ is the non-seasonal parameter of autoregression,

θ is the non-seasonal parameter of moving average,

φ is the seasonal parameter of autoregression,

Θ is the seasonal parameter of parameter of moving average,

ω is the frequency, and

B is the differential variable [44].

In the preliminary stage of our SARIMA model construction, the determination of the appropriate order for the parameters *p*, *q*, *P*, and *Q* was derived from an empirical analysis of the ACF and PACF plots. Then, we ascertained the orders of *d*, and *D* based on the degree of differencing required to achieve stationarity of the time series. We selected the order of *S* based on the annually recurring seasonal pattern observed in the data. Finally, we compared this initial SARIMA $(p, d, q) \times (P, D, Q, S)$ model against an array of SARIMA models characterized by varying parameters by using a grid-search methodology that explored the parameter space. The apex of this process was the selection of the model that exhibited the lowest Akaike Information Criterion (AIC) value. The model was further examined with diagnostic checking of the residuals. The enclosed flowchart in *Figure 3.1* [45] portrays the procedural steps inherent in the SARIMA model methodology, also known as Box-Jenkins iterative approach.

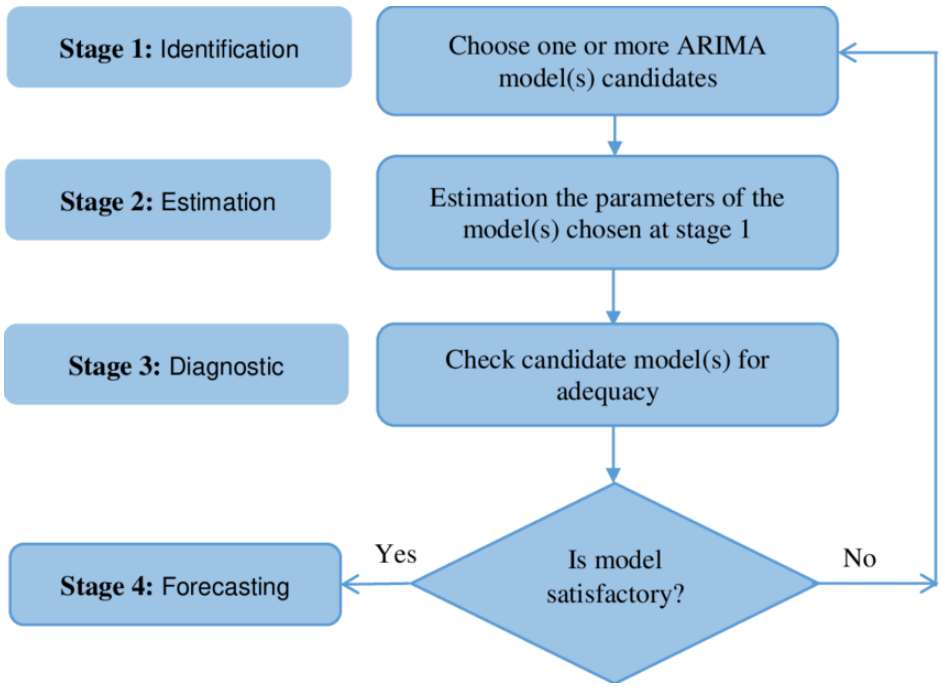

Figure 3.1 SARIMA model building flowchart

3.4.2 HWES

Holt-Winters Exponential Smoothing (HWES), developed by Charles Holt and Peter Winters, is also known as Holt Winters method. The algorithm extends upon Exponential Smoothing to capture various aspects like level, trend, and seasonality in the data. Employing exponentially decreasing weights and values for historical data, exponential smoothing is a technique for smoothing time series data [46]. Holt Winters model with additive trend and multiplicative seasonality can be defined by the following Equations (4-7) [47],

$$S_t = \alpha\left(\frac{X_t}{\prod I^i_{t-s_i}}\right) + (1-\alpha)(S_{t-1} + T_{t-1}) \tag{4}$$

$$T_t = \gamma(S_t - S_{t-1}) + (1-\gamma)T_{t-1} \tag{5}$$

$$I^i_t = \delta^i\left(\frac{X_t}{S_t \prod_{j\neq 1} I^i_{t-s_j}}\right) + \left(1-\delta^i\right)I^i_{t-s_i} \tag{6}$$

$$\hat{X}_t(k) = (S_t + kT_t)\prod_i I^i_{t-S_{i+k}} + \varphi^k_{AR}\left(X_t - (S_{t-1} + kT_{t-1})\prod_i I^i_{t-s_i}\right) \tag{7}$$

Where S_t represents the level,

T_t is the trend,

I_t^i corresponds to multiple seasonality,

$\hat{X}_t(k)$ is the k-step ahead forecast,

α denotes the level smoothing parameter,

γ is the trend smoothing parameter,

δ^i represents the smoothing parameters of each seasonal pattern with cycle length of s_i, and

φ_{AR}^k is an adjustment for the first autocorrelation error [47].

Based on our domain knowledge and a detailed examination of the features present in our time series data on dengue cases, we decided to apply the HWES model, configuring it with an additive trend and multiplicative seasonality. This decision was underpinned by a detailed understanding of the dengue case patterns, where the additive trend component aptly represented the gradual changes in dengue occurrences over time, while the multiplicative seasonal component captured the proportional variability of seasonal fluctuations.

3.4.3 Prophet

Prophet is an algorithm that uses an additive mode to forecast time series data. Engineered in 2017 by the Data Science Team at Facebook, it handles strong seasonal effects and outliers well. The algorithm contains with three main components: trend, seasonality and holidays and can be given by the following Equations (8-10) [48],

$$y_t = b_t + s_t + h_t + \varepsilon_t \tag{8}$$

$$b_t = \frac{C}{1 + \exp\big(-k(t-m)\big)} \tag{9}$$

$$s_t = \sum_{n=1}^{N} \left[\alpha_n \cos\left(\frac{2\pi nt}{P}\right) + \beta_n \sin\left(\frac{2\pi nt}{P}\right)\right] \tag{10}$$

$$h_t = [\mathbf{1}(t \in D_1), \ldots, \mathbf{1}(t \in D_L)]\kappa \tag{11}$$

Where ε_t is the error term,

C is the carrying capacity that is the maximum value of the logistic curve,

k is the growth rate which controls the steepness of the curve,

m is an offset parameter corresponding to the curve's midpoint,

s_t is the seasonality with a regular period P,

D_i is the set of dates for holidays, and

κ is the change in the forecast caused by holidays [48].

In our research, implementing the prophet model was relatively easy. After doing the necessary data processing for the prophet model as discussed in the preprocessing section, we let the Prophet model train on the whole data on default parameters and optimization. Then we forecasted for the next 12 months using the fitted model.

3.4.3 LSTM

A variant of the Recurrent Neural Network (RNN), the Long-Short Memory Network (LSTM) is adept at learning from sequences of time series data. It integrates short-term memory and long-term memory via gating mechanisms, effectively addressing the issue of vanishing gradients [49]. A basic architecture of the LSTM model is shown in *Figure 3.2* [50]. The mathematical intuitions behind this model are given below by the equations (12-17) [51],

$$i_t = \sigma\left(x_t U^i + h_{t-1} W^i\right) \quad (12)$$

$$f_t = \sigma(x_t U^f + h_{t-1} W^f) \quad (13)$$

$$o_t = \sigma(x_t U^o + h_{t-1} W^o) \quad (14)$$

$$\tilde{C}_t = \tanh\left(x_t U^c + h_{t-1} W^c\right) \quad (15)$$

$$C_t = \sigma\left(f_t * C_{t-1} + i_t * \tilde{C}_t\right) \quad (16)$$

$$h_t = \tanh\left(C_t\right) * o_t \quad (17)$$

Where, i_t, f_t , and o_t represents input, forget and output gates, at time t respectively,

x_t is the number of input features,

h_t is the number of hidden units,

W and U are weight matrices,

$\tilde{C}_t$ is the intermediate cell state, and

C_t refers to current cell memory [51].

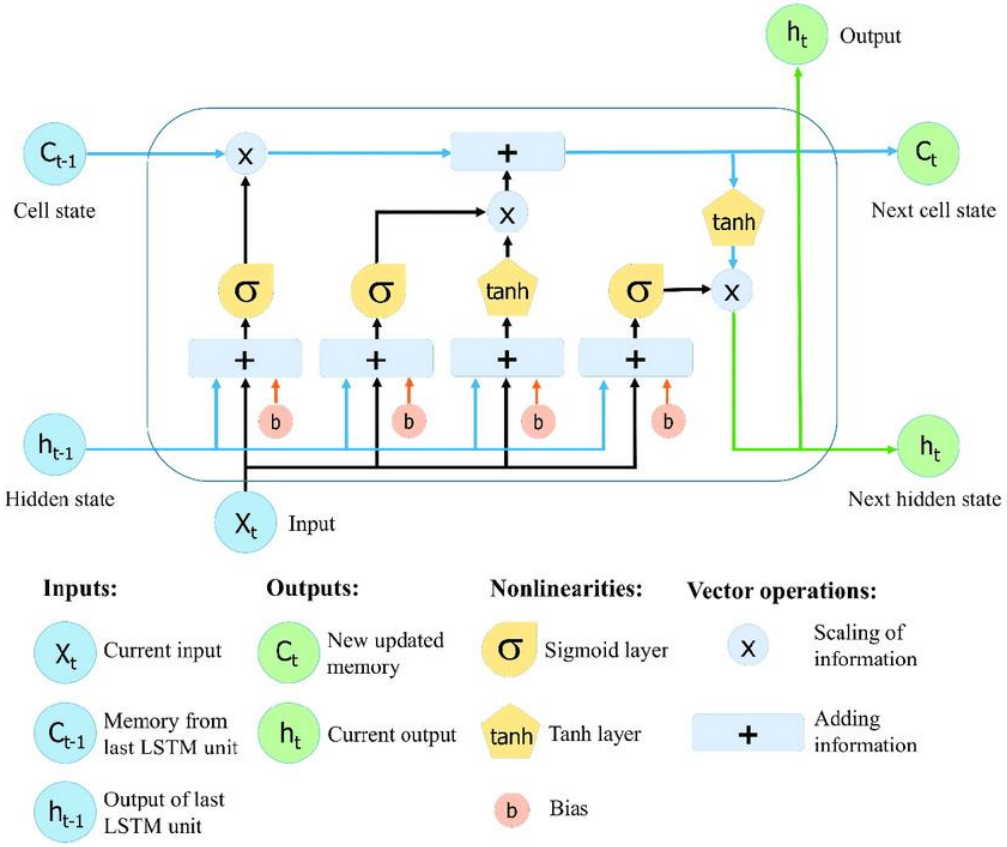

Fig 3.2 LSTM architecture

To capture the complex temporal fluctuations, present in our time series data, we built the LSTM as a sequential network with numerous layers. Mandatory data normalization and transformation was applied to the training data as discussed in an earlier section of data preprocessing. The model was initiated with two hidden layers, each consisting of 50 units. A dense layer with a single unit was employed as the output layer to consolidate the learned features into a single predicted value, representing the forecasted number of dengue cases. We adopted state of the art Adam optimizer and a diminutive learning rate of 0.0001. Training the model involved iterating through the data for 100 epochs with a batch size of 16. Post-training, the predicted values and actual training values were inverse-transformed from their normalized state to revert the scaled data back to its original scale. The following pseudo-code shown in *Figure 3.3* concisely summarizes the whole procedural approach of the LSTM algorithm.

```
Psuedo-code for the LSTM algorithm:

Normalize the 'cases' column of the dataset.

Set TIME_STEPS to 12.

Prepare X_train, y_train using create_dataset.
create_dataset(X, y, time_steps):
    Return array of X, y sequences.

Initialize Sequential LSTM model with:
    LSTM layer (50 units, input shape from X_train).
    LSTM layer (50 units).
    Dense output layer (1 unit).
Compile with Adam optimizer (0.0001 learning rate), MSE loss.
Train for 100 epochs, batch size 16.

Predict and inverse transform training set predictions.
Calculate MAE, and RMSE on training data.

Forecast for the next 12 periods:
    Use the last TIME_STEPS of 'scaled_cases' for initial input.
    Predict and append each next value, updating the input sequence.

Output and plot the forecasted values.
```

Fig 3.3 Pseudo-code of the LSTM algorithm

3.5 Model Selection & Evaluation Criteria

In this final part of the methodology section, we delve into the comprehensive approach employed for selecting the most appropriate forecasting model for our study on dengue cases. Our methodology hinged on a two-tiered model evaluation strategy. This strategy encompassed determining the best SARIMA model based on the lowest AIC value, and comparing all the forecasting models using two simple evaluation metrics, MAE, and RMSE. A further residual analysis of the best model was also explored. Given below are the theoretical foundations of these criteria.

3.5.1 AIC

Named after the Japanese statistician Hirotsugu Akaike, the Akaike Information Criterion (AIC) is a measure of prediction error, and thus, the relative quality of statistical models for a particular set of data. Mathematically,

$$\mathrm{AIC} = 2k - 2\ln(L) \tag{18}$$

Where, *k* represents the number of parameters in the model,

L is the maximum value of the likelihood function for the model.

3.5.2 MAE & RMSE

Within the scope of our study on time series data of dengue cases, Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE) measured the accuracy of the fitted models. In research, it's generally expected that the MAE should be smaller than the RMSE to yield more accurate prediction outcomes [52]. These two evaluation metrics can be expressed by the given Equations (19, 20) [53],

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |x_i - x| \tag{19}$$

$$\mathrm{RMSE} = \sqrt{\sum \frac{\left(y_{\mathrm{pred}} - y_{\mathrm{ref}}\right)^2}{n}} \tag{20}$$

Where, *n* is the total quantity of data points,

x, y_{ref} are experimental values in the data set, and

x_i, y_{pred} are projected values from the models [53].

3.5.3 Residuals Analysis

Following the identification of our best-fitted model through a rigorous comparison using metrics such as MAE and RMSE, we proceeded to conduct a detailed residual analysis. This critical phase of our methodology was aimed at ensuring the robustness of the model and

validating its predictive capability for dengue case forecasting. The residual analysis encompassed the examination of standardized residuals, the Autocorrelation Function (ACF) plot of these residuals, and the Ljung-Box test.

Standardized residuals analysis evaluates how much the predicted values differ from the actual values, and also checks for any discernible patterns or systematic biases in the residuals. Ideally, standardized residuals should resemble white noise suggesting that the model has successfully incorporated all the pertinent information contained within the data. ACF of the standardized residuals investigates any autocorrelation in the residuals, which can indicate model misspecifications or omitted variables. The absence of significant autocorrelations (i.e., most autocorrelations are within the confidence interval) in the ACF plot is desirable, as it implies that the residuals are random and the model has adequately captured the time-dependent structure of the data. Furthermore, the Ljung-Box test performs a statistical test for overall randomness in the residuals by examining the null hypothesis that the residuals are independently distributed. A failure to reject the null hypothesis (p-value > 0.005) would indicate that the residuals are random, suggesting that the model does not exhibit significant lack of fit.

3.6 Conclusive Summary

The entirety of the methodology can be briefly encapsulated within a streamline flowchart containing the procedural steps, given by *Figure 3.4*. In the course of our arduous study, we identified various characteristics, and limitations of the models. These findings are concisely presented in the *Table 3.1* below.

Model	**Characteristics**	**Data Transformation Requirements**	**Limitations**
SARIMA	• Forecasts future values based on past values. • Adds seasonal terms to ARIMA framework.	• No transformation required.	• Assume linear relationship. • Tedious selection of the parameters.
HWES	• Captures level, trend, and seasonality. • Includes additive and multiplicative options.	• Zero values have to be replaced with ones in multiplicative option.	• Linear in nature. • The multiplicative method is not suitable for handling zero values in its original form.
Prophet	• Predicts time series	• Minimal	• Subpar performance

	data with an additive model, handling seasonality and outliers efficiently.	transformation required; renaming of the columns.	compared to other classical models. • Requires fine tuning to boost performance.
LSTM	• A variant of RNN, adept at learning sequences in time series data.	• Normalization of data is essential, along with sequential restructuring of the dataset.	• Computationally expensive. • Requires large dataset and complex parameter tuning.

Table 3.1 Unique characteristics of the forecasting models

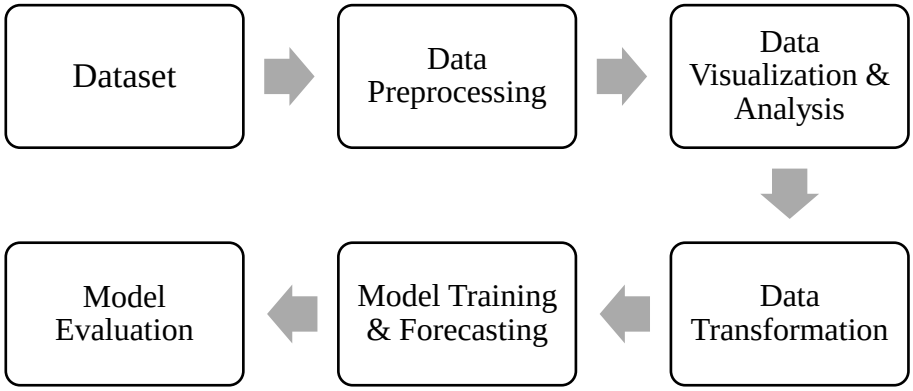

Fig 3.4 Summarized pipeline of the methodology

In conclusion, the methodology section of our research provides a thorough and methodical approach for forecasting dengue cases by utilizing a range of sophisticated time series models. This methodological journey demonstrates our dedication to a comprehensive and data-driven approach by combining statistical rigor with practical model selection. The strategies adopted in this study paved the way for robust and insightful forecasts, contributing significantly to our understanding and management of dengue occurrence trends.

Chapter 4

Implementation

This chapter focuses on the practical aspects of implementing the forecasting models for dengue cases in Bangladesh. It covers the specifics of the hardware and software utilized, the experimental setup, and the detailed aspects of tuning and executing the models

4.1 Hardware & Software Specifications

To ensure that researchers can replicate the experimental environment and reproduce the results, it is essential to specify the hardware and software used in the experiments. We mainly used a generic consumer laptop as the workstation. The important hardware specifications and software versions have been given in *Table 4.1*.

Model	MSI Modern 15 A5M
Processor	Ryzen 5500U (2.1 GHz)
Storage	512GB (PCIe Gen3)
Ram	16GB DDR4 (3200MHz)
GPU	RX Vega 7
GPU Memory	512MB (1800MHz)
Operating System	Windows 11
Programming Language	Python (v3.9.15)
Python Distribution	Anaconda (v23.3.1)

Table 4.1 Overview of Hardware & Software

4.2 Data Exploration

In this section, we will discuss the process of data reading, preprocessing, and visualization in detail. These steps were accomplished in ‘Jupyter Notebook’ (v6.5.4), an interactive computing environment that combines code execution with rich visualizations, making it particularly suited for data exploration. It comes by default with the ‘Anaconda’ distribution of the ‘Python’ programming language.

4.2.1 Data Reading & Initial Treatment

At first, we imported necessary libraries along with their aliases in the ‘Jupyter Notebook’ environment. The time series data of monthly dengue cases, stored in CSV format, was read using the ‘Pandas’ library, used for working with datasets. The dataset was put into a ‘Pandas DataFrame’ format, named ‘df’, and from the two columns ‘date’ and ‘cases’, the ‘date’ column was converted into ‘DateTime’ format and set as the index, which is a requirement for analyzing time series data. Also, the dataset was checked for the presence of any missing values. These steps were implemented using the following code snippet of *Figure 4.1*.

Importing Dependencies

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sms
import warnings
import itertools

%matplotlib inline
```

Data Preprocessing

```
df = pd.read_csv(r'C:\Users\Admin\Desktop\Project\dataset\dengue08.csv')
df['date'] = pd.to_datetime(df['date'])
```

```
df.info()
df.isnull().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 192 entries, 0 to 191
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   date    192 non-null    datetime64[ns]
 1   cases   192 non-null    int64
dtypes: datetime64[ns](1), int64(1)
memory usage: 3.1 KB

date     0
cases    0
dtype: int64
```

```
df.set_index('date', inplace=True)
```

Figure 4.1 Code for data reading & preprocessing

4.2.2 Graphical Data Presentation

We used the ‘Matplotlib’ library as our primary visualization tool, along with ‘Seaborn’, both capable of drawing informative statistical graphics with a touch of aesthetics. At first, we rendered the whole dataset in a time series plot with the help of the code snippet which is given

below in *Figure 4.2*. Necessary axis labeling, grid configuration, and plot styling were also done to enhance readability without cluttering the visual space.

Time Series Plot

```
: # Plotting using Matplotlib
plt.figure(figsize=(13, 6))
plt.plot(df.index, df['cases'], label='Number of Cases', color='darkblue')
plt.title('Time Series Plot of Dengue Fever In Bangladesh',
          fontsize=15, fontweight='bold', pad=20)
plt.xlabel('Year', fontsize=12, labelpad=20)
plt.ylabel('Number of Cases', fontsize=12, labelpad=20)
plt.xticks(fontsize=10)
plt.yticks(fontsize=10)
plt.grid(True, alpha=0.2)
```

Figure 4.2 Code for time series graph

Then, we performed a visual analysis by decomposing the time series into its constituent components, which typically include the trend, seasonality, and residuals. The ‘statsmodels’ package’s ‘seasonal_decompose’ function was imported and along with the ‘Seaborn’ library, we created the visualization with necessary labels and styling, summarized in the following code of *Figure 4.3*.

Seasonal Decomposition

```
from pandas.plotting import register_matplotlib_converters

register_matplotlib_converters()
sns.set_style("darkgrid")

plt.rc("figure", figsize=(12, 8))
plt.rc("font", size=11)

from statsmodels.tsa.seasonal import seasonal_decompose
```

```
# Perform seasonal decomposition
result = seasonal_decompose(df)

# Plot the trend, seasonal, and residual components
plt.subplot(3, 1, 1)
plt.plot(result.trend)
plt.title('Trend')

plt.subplot(3, 1, 2)
plt.plot(result.seasonal)
plt.title('Seasonality')

plt.subplot(3, 1, 3)
plt.scatter(df.index, result.resid)  # Use scatter for residuals
plt.title('Residual')

# Add some whitespace between the subplots
plt.tight_layout(pad=3.0)

# Add a title to the entire plot
plt.suptitle('Seasonal Decomposition of The Time Series Data', fontsize=16, y=1.01, fontweight='bold')
```

Figure 4.3 Code for seasonal decomposition

In the final part of our data visualization, we did an autocorrelation analysis of the data from ACF, and PACF plot, as mentioned in the Methodology section. The resulting plots helped us to determine the parameters *p, P, d,* and *D* of the SARIMA model at the preliminary stage. We assorted to using the 'statsmodel.graphics.tsaplots' module within the 'statsmodels' library in Python to generate the two aforementioned plots by constructing the code given in *Figure 4.4*.

ACF, PACF Plot

Order "p, P" of the AR term --> PACF plot

Order "q, Q" of the MA term --> ACF plot

```
plt.rcParams.update({'font.size': 11})
```

```
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

# Plotting the ACF `
plt.figure(figsize=(10, 4))
plot_acf(df['cases'], lags=40, ax=plt.gca())
plt.title('Autocorrelation Function (ACF)')
plt.show()

# Plotting the PACF
plt.figure(figsize=(10, 4))
plot_pacf(df['cases'], lags=40, ax=plt.gca())
plt.title('Partial Autocorrelation Function (PACF)')

# Save the plot to a file
plt.savefig("003.acf_pacf.png", bbox_inches='tight')
plt.show()
```

Figure 4.4 Code for ACF, PACF plots

4.2.3 Stationarity Test

The Augmented Dickey-Fuller (ADF) test is devoted to assessing whether a time series data exhibits stationarity, a key consideration when using ARIMA-based models. We used ‘adfuller’ function from the ‘stattools’ module within the ‘statsmodels.tsa’ package to perform this test on the main time series, the first differenced series, and the twelfth differenced series data to establish the *d*, *D* parameters of the SARIMA model. The code in *Figure 4.5* describes this process.

ADF test for stationarity

```
from statsmodels.tsa.stattools import adfuller
def adf_test(series):
    result=adfuller(series)
    print('ADF Statistics: {}'.format(result[0]))
    print('p- value: {}'.format(result[1]))
    if result[1] <= 0.05:
        print("strong evidence against the null hypothesis, reject the null hypothesis. Data has no unit root and is stationary")
    else:
        print("weak evidence against null hypothesis, time series has a unit root, indicating it is non-stationary")

adf_test(df['cases'])

df['first_diff'] = df['cases'] - df['cases'].shift(1)
adf_test(df['first_diff'].dropna())

df['twelve_diff'] = df['cases'] - df['cases'].shift(12)
adf_test(df['twelve_diff'].dropna())
```

Figure 4.5 Code for ADF Test

4.3 Model Implementation

Now, we delve into the practical aspects of implementing four sophisticated statistical models: SARIMA, HWES, Prophet, and LSTM. Each of these models represented a unique approach to time series forecasting, offering a diverse toolkit for addressing the complex challenge of predicting dengue fever cases in Bangladesh. This subsection is dedicated to a detailed exploration of the implementation process for each model, including the configuration of their respective parameters, the handling of the dataset, and the nuances of their application to our specific research context.

4.3.1 SARIMA Implementation

Before implementing the SARIMA model on our dataset, we followed the simplified guide mentioned in *Table 4.1* for the initial selection of the seasonal and non-seasonal parameters of the model.

Parameter	**Value**	**Description**	**Check**	**How to Find**
p	2	AR order	PACF plot	Number of lags crossing the significance limit
d	1	Differencing order	ADF test	Number of differences needed to make the series stationary
q	2	MA order	ACF plot	Number of lags crossing the significance limit
P	0	Seasonal AR order	PACF plot	Observe significant spikes at seasonal lags
D	0	Seasonal	ADF test	Number of seasonal differences required

		differencing order		to make the series stationary
Q	0	Seasonal MA order	ACF plot	Observe significant spikes at seasonal lags
S	12	Seasonal period	Time Series plot	The length of the seasonal cycle in the data

Table 4.2 Guide for the initial selection of model parameters of SARIMA

Then, we imported the 'SARIMAX' class from the 'statespace.sarimax' module within the 'statsmodels.tsa' package to implement SARIMA on our dataset, given in *Figure 4.6*.

SARIMAX from initial parameter selection

```
from statsmodels.tsa.statespace.sarimax import SARIMAX

# Set the frequency of the DateTime index
df.index.freq = 'MS'

# Fit the SARIMAX model
model = SARIMAX(df['cases'], order=(2, 1, 2), seasonal_order=(0, 0, 0, 12), freq="MS")
results = model.fit()

# Display the model summary, and AIC
print(results.summary())
print(f"AIC: {results.aic}")
```

Figure 4.6 Code for SARIMA

Then we performed a grid search of the parameters to find which collection showed the lowest AIC score. The non-seasonal parameters (*p, d, q*) and seasonal parameters (*P, D, Q*) were defined with ranges from 0 to 2, and 0 to 1, respectively. Using Python's 'itertools.product', all possible combinations of those parameters were generated and for each combination, a SARIMA model was instantiated and fitted to the dengue cases data.

Grid Search for the best parameters of SARIMA

```
from statsmodels.tsa.statespace.sarimax import SARIMAX
import itertools
import sys

# Set the frequency of the DateTime index
df.index.freq = 'MS'

# Define non-seasonal p, d, q combinations
p = d = q = range(0, 3)
pdq = list(itertools.product(p, d, q))

# Define seasonal p, d, q combinations with a smaller range
seasonal_p = seasonal_d = seasonal_q = range(0, 2)  # Limiting seasonal parameters to 0 and 1
seasonal_pdq = [(x[0], x[1], x[2], 12) for x in itertools.product(seasonal_p, seasonal_d, seasonal_q)]

# Store AIC, and parameter combinations
aic_results = []

# Number of iterations for progress tracking
total_iterations = len(pdq) * len(seasonal_pdq)
current_iteration = 0

# Iterate through SARIMA models
for param in pdq:
    for param_seasonal in seasonal_pdq:
        try:
            model = SARIMAX(df['cases'],
                            order=param,
                            seasonal_order=param_seasonal,
                            enforce_stationarity=False,
                            enforce_invertibility=False)
            results = model.fit()
            aic_results.append((param, param_seasonal, results.aic))
        except Exception as e:
            continue  # Skip iteration on exception
        finally:
            current_iteration += 1
            print(f"Completed: {current_iteration}/{total_iterations}", end='\r')
            sys.stdout.flush()

# Sort results by AIC
aic_results.sort(key=lambda x: (x[2] if isinstance(x[2], float) else float('inf')))

# Display the top 5 models based on AIC
aic_results[:5]
```

Figure 4.7 Code for parameter grid-search of SARIMA

The results were sorted by the AIC score, and the top 5 models were acquired with their respective AIC scores. The code snippet mentioned in *Figure 4.7* highlights the total procedure. Then, we resorted to the model with the lowest AIC score to train and test with our dataset.

The model was further used to forecast monthly dengue cases for the year 2024. The Python script in *Figure 4.8* created a comprehensive visualization of the observed data, the model's predictions, and the forecasts along with their confidence intervals.

Forecast

```
# Forecast for 2024
forecast = results.get_forecast(steps=12)
pred_conf = forecast.conf_int()
```

Plot for observation

```
sns.set_style("darkgrid")
plt.figure(figsize=(15, 6))
plt.plot(df.index, df['cases'], label='Observed')
plt.plot(y_pred.index, y_pred, label='In-sample Prediction', color='r')
plt.plot(forecast.predicted_mean.index, forecast.predicted_mean, label='Forecast', color='g')
plt.fill_between(pred_conf.index, pred_conf.iloc[:, 0], pred_conf.iloc[:, 1], color='green', alpha=0.2)
plt.xlabel('Date')
plt.ylabel('Cases')
plt.title('SARIMAX Forecast')
plt.legend()

plt.savefig("005.sarima_grid_forecast.png", bbox_inches='tight')

plt.show()
```

Figure 4.8 Code for SARIMA forecast and comparative visualization

4.3.2 HWES Implementation

Data transformation was necessary when implementing the Holt-Winters Exponential Smoothing method on our data. The Holt-Winters method with additive trend and multiplicative seasonality was used in our research. Due to its multiplicative nature, it was unable to handle any zero values in the dataset. So, any 0 value was changed into 1 to avoid calculation error. After that, the transformed dataset was fitted into the model and trained for forecasting, as given in *Figure 4.9*. From the 'statsmodels.tsa.holtwinters' module, the 'ExponentialSmoothing' class was imported to implement the model. The 'Basinhopping' method was used as the optimization algorithm which is effective for avoiding local optima in complex models. *Table 4.2* briefly discusses the parameters used in this model training.

The zero values need slight modification when seasonal='mul'

```
# Find the smallest value in the dataset
min_val = df['cases'].min()

# If the smallest value is less than or equal to 0, add a constant to make all values positive
if min_val <= 0:
    constant_to_add = abs(min_val) + 1  # The "+1" ensures all values are strictly positive
    df['cases'] = df['cases'] + constant_to_add
```

```
# Fit the HWES model with multiplicative seasonality
from statsmodels.tsa.holtwinters import ExponentialSmoothing

model_hwes = ExponentialSmoothing(df['cases'], trend='add', seasonal='mul', seasonal_periods=12)
fit_hwes = model_hwes.fit(method='basinhopping')

# Get the fitted values
fitted_values_hwes = fit_hwes.fittedvalues
```

Figure 4.9 Data transformation & model training of HWES

Parameters	**Value**	**Description**
Trend Component	'add'	Specifies an additive trend component
Seasonal Component	'mul'	Indicates a multiplicative seasonal component
Seasonal Periods	12	The number of time steps in the seasonal cycle

Table 4.3 Parameter selection for HWES

Then, the model was used for forecasting for the year 2024. Unlike SARIMA, the default forecasting of the Holt-Winters method doesn't show confidence intervals. Hence, using the 'Scipy' library, confidence intervals were manually calculated for each forecasted value, assuming the approximate z-score to be 1.96 for a 95% confidence interval. The following code snippet in *Figure 4.10* was run to calculate the confidence intervals and visualize the forecast alongside previous predictions for comparative analysis.

Forecast with CI

```
# Creating a new date range for the forecast period
forecast_index = pd.date_range(start='2024-01-01', periods=12, freq='MS')

# Forecasting for the year 2024 (12 months ahead)
forecast_values = fit_hwes.forecast(steps=12)

# Manually calculating the confidence intervals based on the residuals' standard deviation
residual_std = fit_hwes.resid.std()
conf_int = pd.concat([
    forecast_values - 1.96 * residual_std,
    forecast_values + 1.96 * residual_std
], axis=1)
conf_int.columns = ['Lower_CI', 'Upper_CI']

# Creating a DataFrame to hold the forecast values and confidence intervals
forecast_df = pd.DataFrame({
    'Forecast': forecast_values
}, index=forecast_index)

# Merging the confidence intervals
forecast_df = pd.concat([forecast_df, conf_int], axis=1)

# Plotting the actual, fitted, and forecasted values along with confidence intervals
sns.set_style("darkgrid")

plt.figure(figsize=(15, 6))
plt.plot(df['cases'], label='Actual', color='blue')
plt.plot(fitted_values_hwes, label='In-sample Prediction', color='red')
plt.plot(forecast_df['Forecast'], label='Forecast', color='green')
plt.fill_between(forecast_df.index, forecast_df['Lower_CI'], forecast_df['Upper_CI'], color='green', alpha=0.3
plt.title('Holt-Winters Exponential Smoothing Forecast',fontsize=16)
plt.xlabel('Date', fontsize=14)
plt.ylabel('Cases', fontsize=14)
plt.legend()
```

Figure 4.10 Code for forecasting with HWES model

4.3.3 Prophet Implementation

As discussed in the previous chapter, the Prophet algorithm needs the correct renaming of the columns in a dataset to be fitted into model training. *Figure 4.11* shows the dataset transformation and training part of the Prophet model for our time series data and *Table 4.3* shows the selection of parameters utilized in this model.

Prophet Model Training

```
#Preparing data for Prophet model
df.rename(columns={'date': 'ds', 'cases': 'y'}, inplace=True)

from prophet import Prophet
model = Prophet(yearly_seasonality=True,
                weekly_seasonality=False,
                daily_seasonality=False,
                holidays=None,
                seasonality_mode='multiplicative',
                changepoint_prior_scale=0.04
                )
model.fit(df)

forecast = model.predict(df)
```

Figure 4.11 Code for implementing the Prophet model

Parameter	Value	Explanation
'yearly_seasonality'	True	Enables yearly seasonality in the model
'weekly_seasonality'	False	Disables weekly seasonality
'daily_seasonality'	False	Disables daily seasonality
'holidays'	None	Specifies that no holiday effects are included
'seasonality_mode'	'multiplicative'	Sets the seasonality to be multiplicative
'changepoint_prior_scale'	0.04	Adjusts the flexibility of the automatic changepoint selection. A smaller value makes the trend change more conservative.

Table 4.4 Parameter selection for the Prophet model

Then, we used the Prophet model for forecasting future values of the time series into 2024, given in the code snippet of *Figure 4.12*. The code extended the forecasting period to include the months of 2024 and then generated predictions for this period. Furthermore, for a comparative analysis, the predicted values and their confidence intervals for 2024 were extracted and plotted alongside the actual and previously predicted values of the whole dataset.

Prophet Model Forecast

```
# Extend the future dataframe to include the months of 2024
future_extended = model.make_future_dataframe(periods=12, freq='M')  # 12 months for 2024

# Predict
forecast_extended = model.predict(future_extended)

# Extract the predicted values and confidence intervals for 2024
predicted_values_2024 = forecast_extended['yhat'][-12:].values
lower_bound_2024 = forecast_extended['yhat_lower'][-12:].values
upper_bound_2024 = forecast_extended['yhat_upper'][-12:].values

# Extract dates for 2024
dates_2024 = future_extended['ds'][-12:].values

# Plotting
plt.figure(figsize=(14, 7))
# Actual vs Predicted for test data
plt.plot(df['ds'], actual_values, label="Actual Values", color="blue")
plt.plot(df['ds'], predicted_values, label="Predicted Values", color="red")
# Forecast for 2024
plt.plot(dates_2024, predicted_values_2024, label="Forecast for 2024", color="green")
# Confidence interval
plt.fill_between(dates_2024, lower_bound_2024, upper_bound_2024, color='green', alpha=0.2)

plt.title("Prophet Forecast", fontsize=16)
plt.xlabel("Date", fontsize=14)
plt.ylabel("Number of Cases", fontsize=14)
plt.legend()
plt.grid(True)
plt.tight_layout()
```

Figure 4.12 Code for Prophet model forecasting

4.3.4 LSTM Implementation

The data preparation process for training an LSTM neural network is comparatively more complex than the transformation steps taken during the Holt-Winters method or the Prophet model. Initially, the dataset was put through normalization, scaling the dengue case values to a range between 0 and 1. We opted ‘MinMaxScaler’ function from ‘sklearn.preprocessing’ for this step. Subsequently, a function ‘create_dataset’ was defined to restructure the scaled data into a format suitable for LSTM training. *Figure 4.13* illustrates these procedures given below.

Data Transformation for LSTM

```
from sklearn.preprocessing import MinMaxScaler

# Normalizing the dataset
scaler = MinMaxScaler()
df['scaled_cases'] = scaler.fit_transform(df['cases'].values.reshape(-1, 1))

# Prepare data for LSTM
def create_dataset(X, y, time_steps=1):
    Xs, ys = [], []
    for i in range(len(X) - time_steps):
        v = X.iloc[i:(i + time_steps)].values
        Xs.append(v)
        ys.append(y.iloc[i + time_steps])
    return np.array(Xs), np.array(ys)

TIME_STEPS = 12

X_train, y_train = create_dataset(df[['scaled_cases']], df['scaled_cases'], TIME_STEPS)
```

Figure 4.13 Code for LSTM data transformation

In our study, 'Keras' API from the 'TensorFlow' library was imported to construct and train the neural network model. The model comprised two LSTM layers with 50 units each, where the first LSTM layer was designed to return sequences, facilitating the stacking of LSTM layers, and the second layer prepared the output for the final dense layer. The dense layer, having a single unit, served as the output layer, providing the forecasted value. Crucial hyperparameters of the model are given in *Table 4.4*, which is as follows.

Hyperparameter	**Value**
Optimizer	Adam
Learning Rate	0.0001
Loss Function	Mean Squared Error
Epochs	100
Batch Size	16
Verbose	0

Table 4.5 Selection of LSTM hyperparameters

Post-training, the model was employed to make predictions on the training set, which were then inversely transformed to their original scale. Additionally, the model was utilized to forecast the values for the next 12 months of 2024. This was achieved through a sequential approach, where each prediction was fed into the input for the next, emulating a rolling forecast. The forecasts were also inversely scaled to ensure they were presented in the original scale. *Figure 4.14* summarizes the overall process of this integrated model training and prediction.

LSTM

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense

# Building the LSTM model
model = Sequential()

# First LSTM layer
model.add(LSTM(50, return_sequences=True, input_shape=(X_train.shape[1], X_train.shape[2])))

# Additional LSTM Layer
model.add(LSTM(50, return_sequences=False))  # return_sequences is False for the last LSTM layer

# Output layer
model.add(Dense(1))

# Compiling the model with an Adam optimizer and a learning rate of 0.0001
optimizer = tf.keras.optimizers.Adam(learning_rate=0.0001)
model.compile(optimizer=optimizer, loss='mean_squared_error')

# Training the model
model.fit(X_train, y_train, epochs=100,batch_size=16, verbose=0)


# Making predictions on the training set
train_preds = model.predict(X_train)
train_preds = scaler.inverse_transform(train_preds)
actual_train_values = scaler.inverse_transform(y_train.reshape(-1, 1))

# Forecasting for 2024
forecasts = []
last_data = df[['scaled_cases']].values[-TIME_STEPS:]

for _ in range(12):  # forecasting for each month of 2024
    pred = model.predict(last_data.reshape(1, TIME_STEPS, 1))
    forecasts.append(scaler.inverse_transform(pred)[0, 0])
    last_data = np.append(last_data[1:], pred)
```

Figure 4.14 Code for LSTM forecasting

Furthermore, similar to the previous models, a visualization of the forecasts for 2024 along with actual and fitted data was constructed, which is given in *Figure 4.15*.

LSTM forecast visualization

```
# Visualizing the forecasts for 2024 along with actual and fitted data
sns.set_style("darkgrid")
plt.figure(figsize=(15, 6))
plt.plot(df.index[TIME_STEPS:], actual_train_values, label='Actual')
plt.plot(df.index[TIME_STEPS:], train_preds, label='Fitted')
plt.plot(pd.date_range(start='2024-01-01', periods=12, freq='M'), forecasts, label='Forecast')
plt.title('LSTM Forecast')
plt.xlabel("Date", fontsize=14)
plt.ylabel("Number of Cases", fontsize=14)
plt.legend()

plt.savefig("010.LSTM_forecast.png", bbox_inches='tight')
plt.show()
```

Figure 4.15 Code for LSTM forecast plot

4.4 Model Evaluation

After wrapping up the implementation of the models, the next phase of our critical study embarked on evaluating each model. We selected two primary statistical metrics for this evaluation: Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). Applying these metrics to the actual and predicted values of our time series data was almost identical concerning the aspect of coding. We used 'sklearn.metrics' library to calculate the MAE and RMSE of the models. The following code given in *Figure 4.16* shows the basic idea of these calculations.

Evaluating the model

```
#Calculate MAE
from sklearn.metrics import mean_absolute_error
# Calculate the Mean Absolute Error (MAE) between the actual and fitted values
mae_hwes = mean_absolute_error(df['cases'], fitted_values_hwes)
print(f'MAE: {mae_hwes}')

# Calculate RMSE
from sklearn.metrics import mean_squared_error

rmse_hwes = np.sqrt(mean_squared_error(df['cases'], fitted_values_hwes))
print(f'RMSE: {rmse_hwes}')
```

Figure 4.16 Code for evaluating the model

Since calculating the evaluation metrics was almost identical to every model, only one example of a code snippet has been provided to avoid redundancy. After the best model with the lowest MAE and RMSE scores was identified, it also underwent analysis of residuals to further verify the robustness. The residual analysis included the Ljung-Box test, the Autocorrelation Function (ACF) plot of the standardized residuals, and also their inspection over time. The Ljung-Box test

was carried out using ‘statsmodels’ library. *Figure 4.17* shows the code required to perform this test.

Ljung-Box Test

```
from statsmodels.stats.diagnostic import acorr_ljungbox

residuals_hwes = df['cases'] - fitted_values_hwes

# Perform the Ljung-Box test
ljung_box_results = acorr_ljungbox(residuals_hwes, lags=12)

# Directly print the results
print("Ljung-Box test results:")
print(ljung_box_results)
```

Figure 4.17 Code for performing Ljung-Box test

Finally, we constructed plots for inspecting standardized residuals over time and the ACF of those residuals by using ‘Matplotlib’, ‘scipy’, and ‘statsmodels’ libraries. The code snippet given in *Figure 4.18* illustrates the residual analysis segment.

Residual Analysis

```
import statsmodels.api as sm
import matplotlib.pyplot as plt
from scipy.stats import zscore

def plot_residuals_and_acf(residuals, lags=20):
    """
    Plot standardized residuals over time and ACF for standardized residuals.
    """
    # Standardizing residuals
    std_residuals = zscore(residuals)

    # Creating subplots
    fig, axes = plt.subplots(2, 1, figsize=(10, 10))

    # Plot standardized residuals over time
    axes[0].plot(std_residuals)
    axes[0].set_title('Standardized Residuals Over Time')

    # Plotting ACF of standardized residuals
    sm.graphics.tsa.plot_acf(std_residuals, lags=lags, ax=axes[1])
    axes[1].set_title('ACF of Standardized Residuals')

    plt.tight_layout()
    plt.show()

plot_residuals_and_acf(fit_hwes.resid)
```

Figure 4.18 Code for residual analysis

Chapter 5

Results & Discussion

In this pivotal chapter, we delve into the outcomes of our research endeavor, aimed at forecasting dengue cases in Bangladesh, along with an analysis of the time series data. This investigation seeks to assess how well the models predicted the dynamics of dengue disease, a significant public health issue in our nation. The findings are based on the methodology and implementations described in the earlier chapters. In addition to presenting the quantitative results, we intend to engage in a critical discussion about the findings, both in terms of theoretical contribution and practical application in public health strategies.

5.1 Results

This section provides the findings derived from each segment of our study's methodological framework and the strategies employed in the practical application thereof. We are dedicated to showcasing the raw output of our research efforts, visually represented through graphs and charts, and quantitatively represented through statistical measures, providing a foundational basis for the subsequent discussion segment.

5.1.1 Overview of the Time Series Data

Our dataset encompassed monthly incidences of Dengue Virus (DENV) infections in Bangladesh, from 2008 to 2023, given in *Figure 5.1*. Then, *Figure 5.2* exhibits the entirety of the compiled dataset in a time series plot, where the vertical axis represents the case count and the horizontal axis illustrates dengue cases.

Month	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023
January	0	0	0	0	0	6	15	0	13	92	26	38	199	32	126	566
February	0	0	0	0	0	7	7	0	3	58	7	18	45	9	20	166
March	0	0	0	0	0	3	2	2	17	36	19	17	27	13	20	111
April	0	0	0	0	0	3	0	6	38	73	29	58	25	3	23	143
May	0	1	0	0	0	12	8	10	70	134	52	193	10	43	163	1036
June	0	0	0	61	10	50	9	28	254	267	295	1884	20	272	737	5956
July	160	4	61	255	129	172	82	171	926	286	946	16253	23	2286	1571	43854
August	473	125	183	691	122	339	80	765	1451	346	1796	52636	68	7698	3521	71976
September	334	188	120	193	246	385	76	965	1544	430	3087	16856	47	7841	9911	79598
October	184	154	45	114	107	501	63	869	1077	512	2406	8143	164	5458	21932	67769
November	0	0	0	36	27	218	22	271	522	409	1192	4011	546	3567	19334	40716
December	0	0	0	9	0	53	11	75	145	126	293	1247	231	1203	5024	9288

Figure 5.1 Raw data of monthly dengue cases in Bangladesh from 2008 to 2023

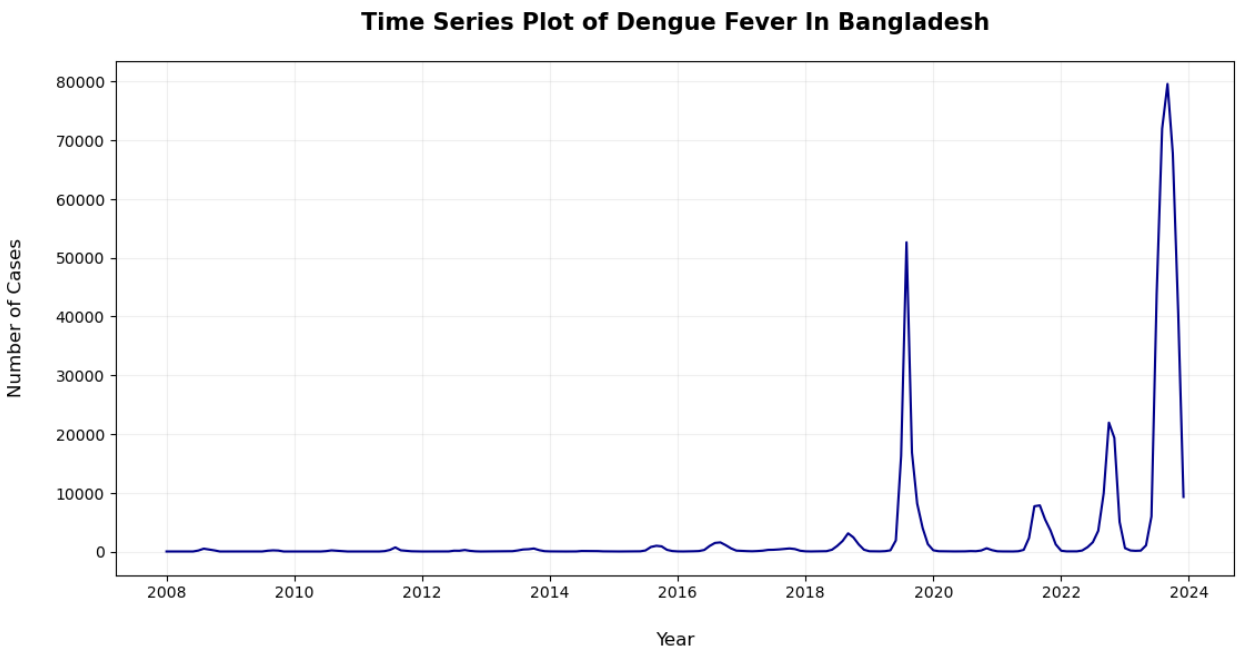

Figure 5.1 Chronological sequence of dengue cases in Bangladesh

From visual inspection of the time series plot, the number of cases was relatively slow and stable from 2008 to around 2018, with subtle minor peaks. The trend from 2019 onwards was more volatile, with higher peaks and more variation from year to year. The largest spikes occurred around 2019 and 2022, indicating severe outbreaks of dengue fever during those times. A seasonal-trend decomposition of the time series data, illustrated in *Figure 5.3*, was further analyzed to understand the underlying patterns more concisely. The decomposition broke down the time series into three components: trend, seasonality, and residuals. The trend component showed an overall increasing pattern, which was comparatively sluggish until about 2018, then a clear upward trend towards 2022 and 2023. The seasonality component presented clear seasonal patterns in the data repeating annually. The residuals, which are the differences between the actual data and the combined model of trend and seasonality, were centered around the zero mean, with some variations. Large residuals coincided with the peaks in the time series, suggesting that the outliers were not fully explained by the seasonal and trend components.

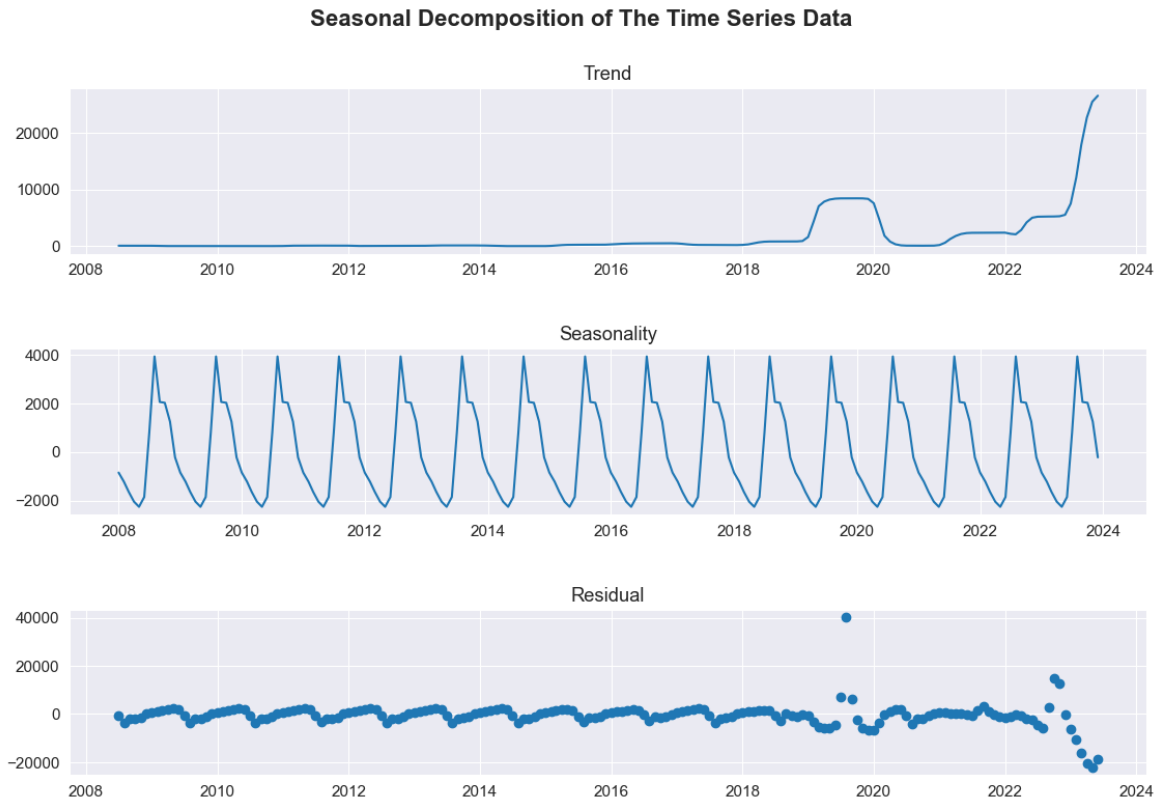

Figure 5.3 Seasonal-Trend decomposition of the time series data

Figure 5.4 shows the Autocorrelation Function (ACF) plot of the time series data. The blue bars represent the autocorrelation at different lags (up to 40) with the blue shaded area as the confidence interval. The autocorrelation at lag 0 was 1, since the time series was perfectly correlated with itself at lag 0. The plot showed significant positive autocorrelation at the initial lags of 1 and 2. Beyond the initial lags, the values hovered around zero and within the confidence interval. Similarly, *Figure 5.5* illustrates the Partial Autocorrelation Function (PACF) plot of our dataset, where the vertical blue bars represent the partial autocorrelation coefficients for each lag, up to 40 lags. After lag 0, we noticed significant spikes at the initial lags 1 and 2, while the rest were under the 95% CI (confidence interval). These two plots ACF, and PACF helped us to determine the initial parameters *p, P, q,* and *Q* for SARIMA to be 2, 0, 2, and 0 respectively.

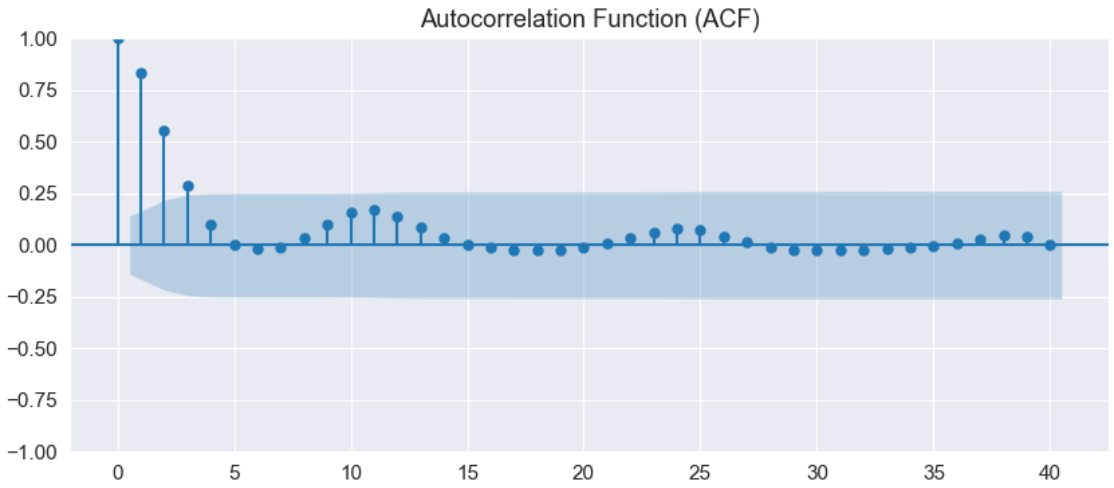

Figure 5.4 ACF plot

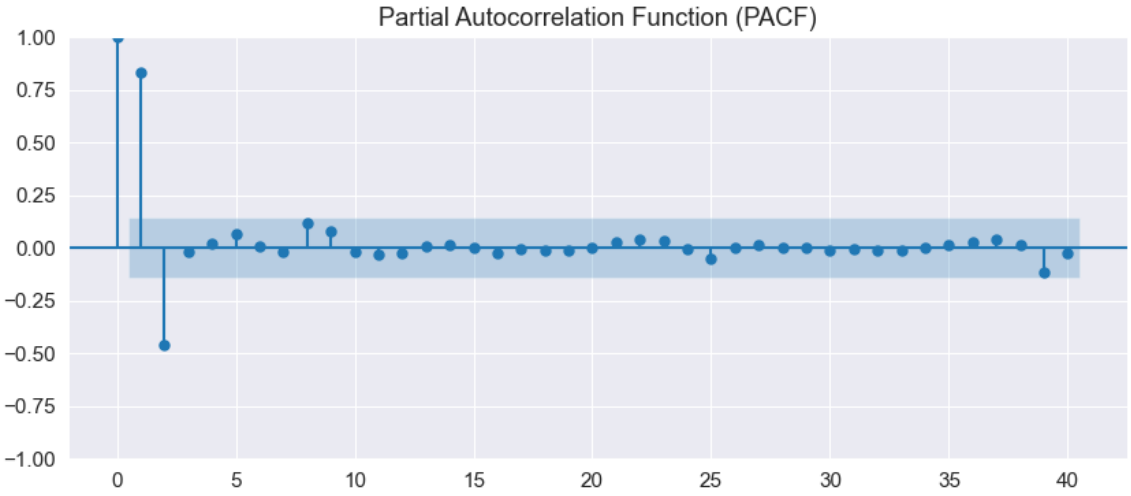

Figure 5.5 PACF plot

5.1.2 Outcome of the ADF Test

We examined the stationarity of our time series data using the Augmented Dickey-Fuller (ADF) test, where the test results included an ADF statistic and corresponding p-value. We performed the ADF test on the original series, the first difference of the series, and the twelfth difference of the series, and the outcome is given in *Table 5.1*.

Data	p-value
Original Series	0.92
First Difference Series	1.78e-15
Twelfth Difference Series	0.37

Table 5.1 ADF Test Results

The test indicated original time series data was non-stationary (p-value > 0.05). The first differencing made the data stationary (p-value < 0.05), but the seasonal differencing at lag 12 did not. The results suggested for the initial *d* and *D* parameters of SARIMA to be 1 and 0.

5.1.3 SARIMA Forecast

We selected the initial parameters of SARIMA from our previous examination of the autocorrelation plots and ADF test. So, we considered *SARIMA (2, 1, 2) (0, 0, 0, 12)* to be implemented at first. After fitting the time series dataset with this model, the AIC value was 3822.114. For further investigation, we performed a grid search of the numerous combinations of the parameters. This yielded *SARIMA (2, 1, 2) (0, 1, 1, 12)* as the better model with the lowest AIC value of 3604.277, which was selected for our final training and forecasting purpose. *Figure 5.6* depicts the forecast of this model alongside the actual observed dengue cases. The blue and red line shows the observed and predicted values of the time series data, while the green line with the shaded area gives the projection of dengue cases in 2024 with 95% CI.

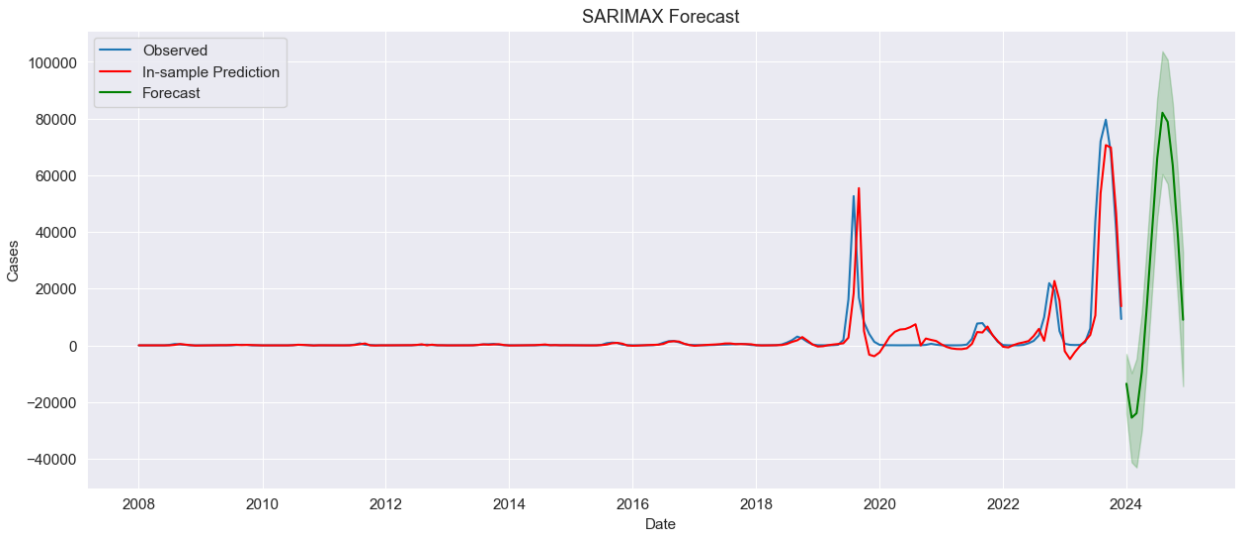

Figure 5.6 SARIMA Forecast

The forecast, juxtaposed with the actual dengue fever cases data, demonstrated a commendable fit during the in-sample period, as evidenced by the alignment of the predicted values with the observed historical data. This observation was further strengthened by the diagnostic plots shown in *Figure 5.7*. The Standardized Residual Plot showed residuals scattered randomly around zero, while systematic peaks suggested inadequacy to capture the epidemic blasts of the years 2019 and 2023, also confirmed by the deviations in the Q-Q plot. The correlogram (ACF plot) of the residuals at different lags showed all autocorrelations within the significant band, suggesting the overall adequacy of the model.

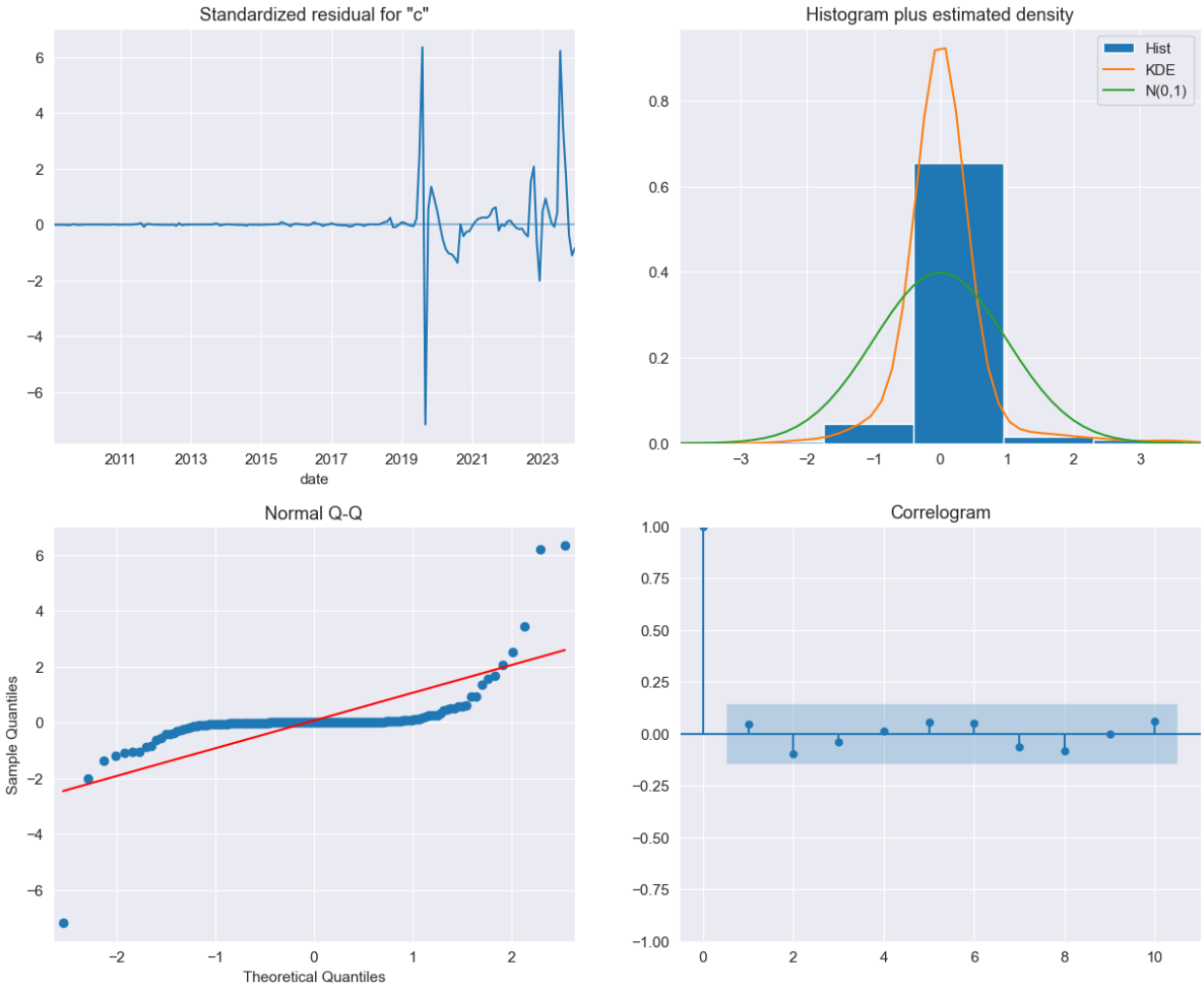

Figure 5.7 Diagnostic plots of the SARIMA model

5.1.4 HWES Forecast

Our investigation employed the Holt-Winters Exponential Smoothing (HWES) model, incorporating an additive trend and multiplicative seasonality method for dengue forecasting. The graph in *Figure 5.8* illustrates the application of the Holt-Winters Exponential Smoothing technique to forecast the time series of dengue case counts. The actual observed data is depicted by the blue line, while the in-sample predictions of the model are represented by the red line, closely tracing the historical data and demonstrating the model's capacity to capture the intrinsic patterns within the observed period. The green line extends beyond the scope of the actual data, presenting the model's forecast into the year 2024, with the shaded area around this line indicating the confidence intervals for these predictions. Notably, the forecast anticipated a pronounced increase in cases, reflecting an epidemy-driven surge.

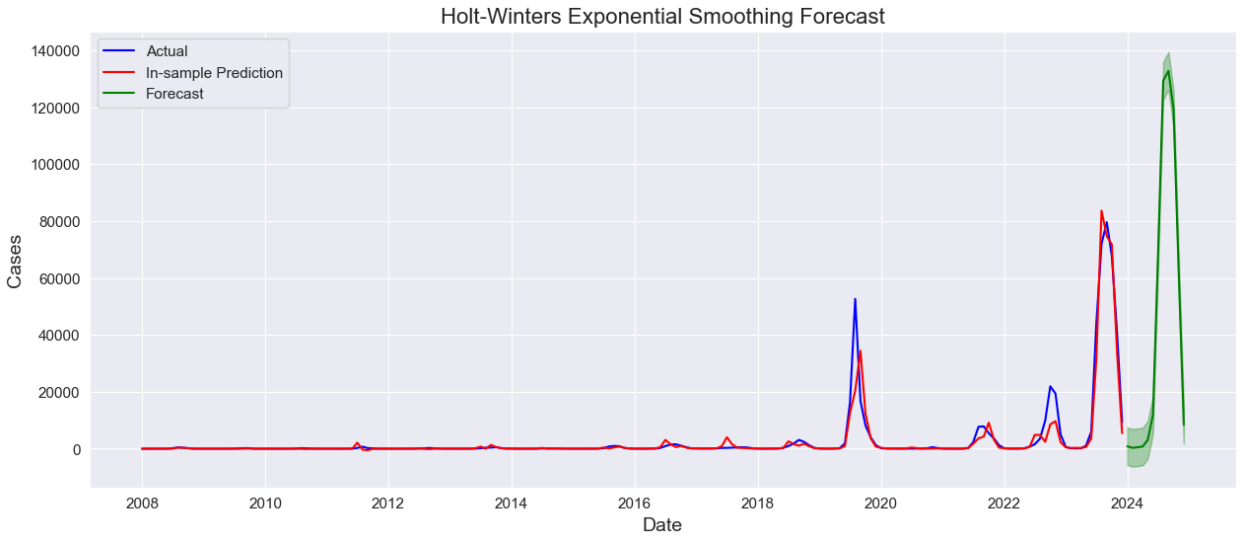

Figure 5.8 Holt-Winters Forecast

5.1.5 Prophet Forecast

Figure 5.9 presents the forecast generated by the Prophet model, similar to the previous forecasting graphs. From observation, the model captured the seasonal peaks and troughs with reasonable accuracy, although couldn’t capture the epidemic outbursts of 2019 and 2023 as well as SARIMA or HWES. The absence of the non-negativity constraints was also observed in the early forecasting values going below the zero line.

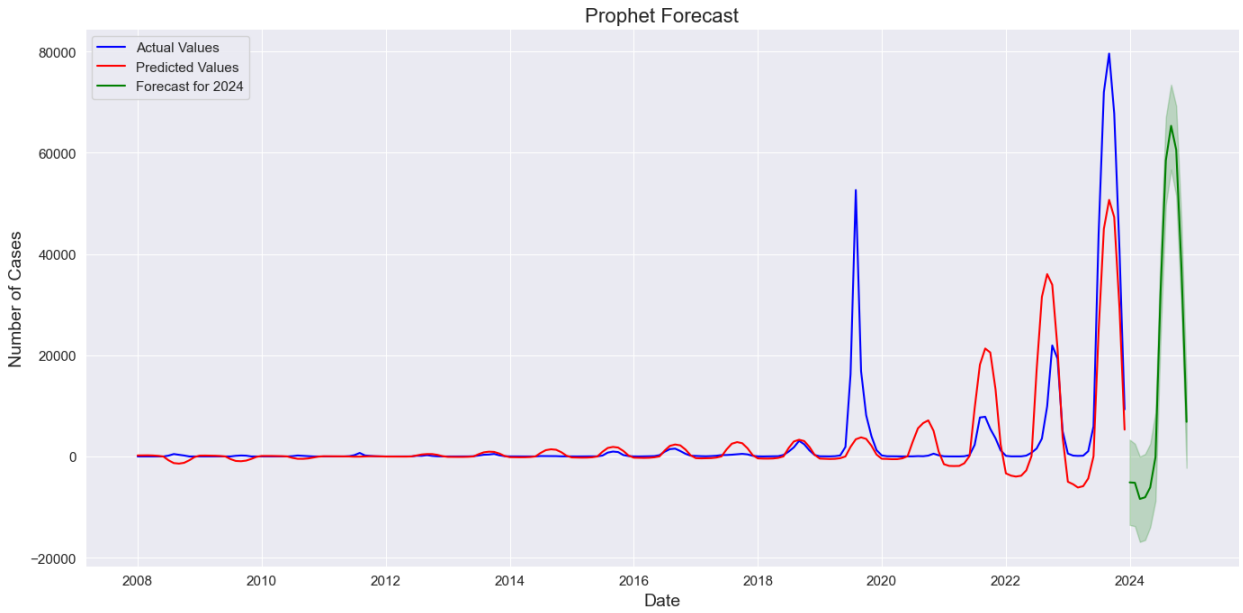

Figure 5.9 Prophet Forecast

5.1.5 LSTM Forecast

In this study, the implemented LSTM model couldn't capture the underlying seasonal patterns and outliers as precisely as the previously examined models. The observations are given in *Figure 5.10*. Investigating the graph, we observed that the fitter values were rather linear instead of having seasonal fluctuations and the model hardly captured the epidemic bloom of 2019 and beyond. The forecast also showed considerable variance, deviating from the historical trend.

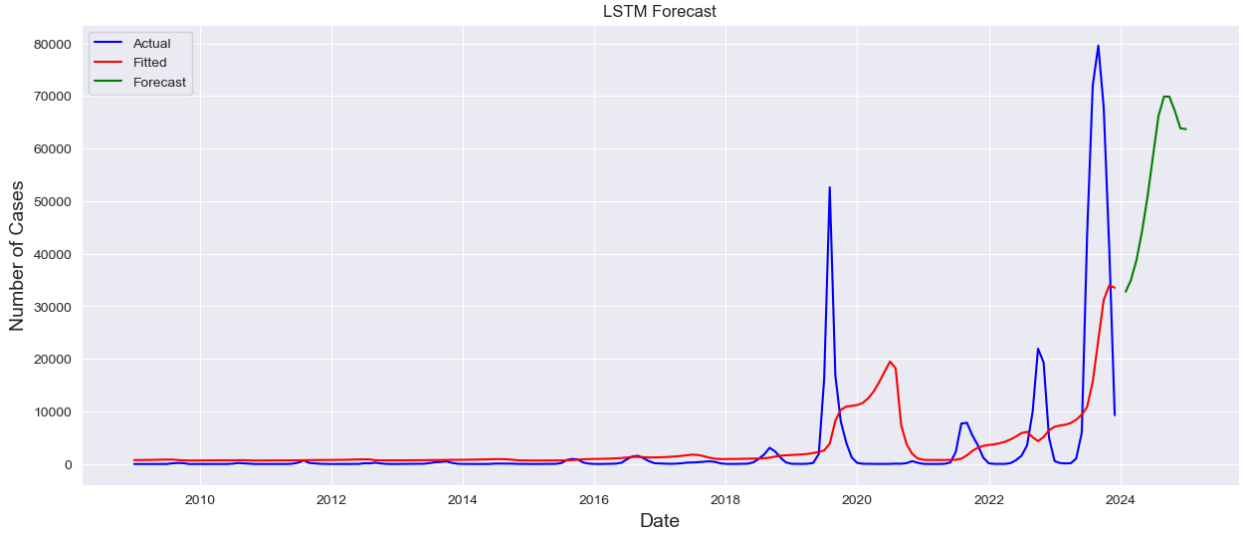

Figure 5.10 LSTM Forecast

5.1.6 Evaluation Metrics Results

The comparative efficacy of each model's performance was evaluated using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) as the principal metrics of accuracy. The results are cataloged in *Table 5.2*.

Model	**MAE**	**RMSE**
SARIMA (2, 1, 2) (0, 1, 1, 12)	1572.096	5173.864
HWES	963.865	3389.622
Prophet	2543.344	6589.956
LSTM	3701.169	9190.474

Table 5.2 Model Evaluation

These figures illustrated the disparities in forecast accuracy across the models, with the Holt-Winters model achieving the lowest MAE and RMSE values, indicating a higher predictive accuracy relative to the SARIMA, Prophet, and LSTM models. Conversely, the LSTM model exhibited the highest MAE and RMSE, suggesting a lesser degree of precision in its prediction ability of our case data. *Table 5.3* presents the forecasted monthly dengue case numbers for the year 2024 as predicted by the Holt-Winters Exponential Smoothing (HWES) model.

Month	**Cases**
January	826
February	303
March	470
April	773
May	3048
June	11880
July	67961
August	129307
September	132736
October	119331
November	61413
December	8399

Table 5.3 2024 Dengue Forecast by HWES

5.1.7 Residual Analysis of the Best Model

Based on the assessment criteria utilized, we found the HWES model to be the superlative one in our research study, compared to the other experimented models. After the initial evaluation, a comprehensive residual analysis was further conducted to verify the model's robustness. *Figure 5.11* contains the two plots related to the residual analysis.

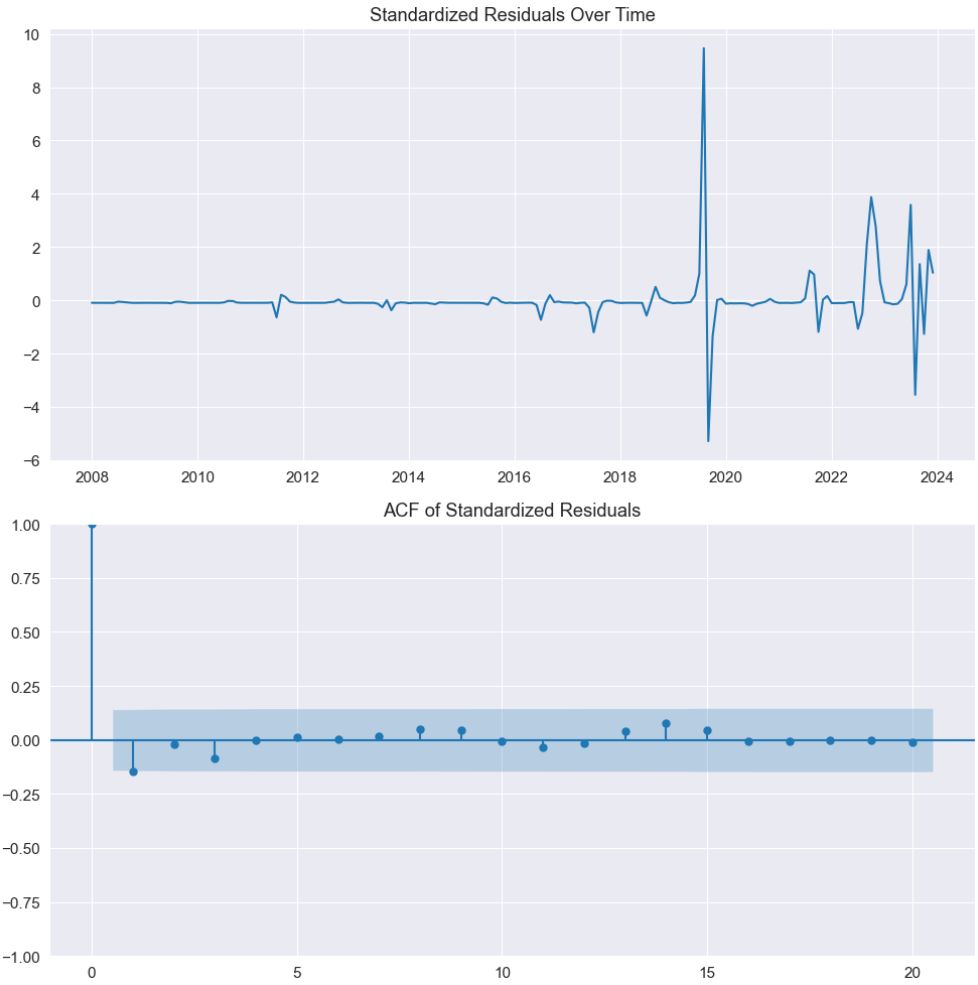

Figure 5.11 Residual Analysis of HWES model

The first plot depicts standardized residuals over time. We observed that the residuals were scattered around randomly around the zero line without any apparent patterns. This randomness suggested that the robustness of the model was solid. The second plot illustrates the autocorrelation function (ACF) of the standardized residuals up to 20 lags. The model being well-fitted, all autocorrelation coefficients fell within the blue-shaded area (95% CI), except lag 1 which showed minuscule autocorrelation. The independence of the residuals was further

confirmed by the Ljung-Box test results given in *Table 5.4*. According to the results, aside from the first lag having a p-value smaller than 0.05, the residuals appeared random, suggesting that the model adequately captured the underlying pattern of the time series data.

Lag	**p-value**
1	0.041
2	0.119
3	0.131
4	0.227
5	0.339
6	0.460
7	0.570
8	0.621
9	0.672
10	0.756
11	0.807
12	0.861

Table 5.4 Ljung-Box Test results

5.2 Discussion

In this section, we contextualize and critically discuss the results of our empirical analysis, focusing on forecasting dengue cases in Bangladesh through various statistical models. The research outcomes, as detailed in the preceding section, have highlighted the differential performance of the SARIMA, HWES, Prophet, and LSTM models in capturing the temporal dynamics of dengue prevalence.

This study aimed to identify the pattern of dengue epidemics and generate a brief forecast for the following year. The time series plot of the dengue dataset showed an overall increase of DENV-affected cases and a comparatively volatile rise in recent years, especially in 2023, surpassing the previous historical records. The re-emergence and dominance of the DENV-3 serotype have been identified as the probable cause of these recent dengue outbreaks in Bangladesh [13] [54]. There's also the potential for an extended dengue fever season, possibly resulting in continuous transmission throughout the year, with outbreaks happening at unpredictable times [55]. Our SARIMA (2, 1, 2) (0, 1, 1, 12) model introduced new seasonal and non-seasonal parameters in comparison to other studies [56] [27] conducted in the scenario of forecasting dengue cases in Bangladesh. The model demonstrated noteworthy alignment with the actual data during in-sample periods and also turned out to be the second-best forecasting model in comparison. This

purely statistical model was applied to predict the pattern of influenza [57], hemorrhagic conjunctivitis [58], renal syndromed fever [59], and so on.

Interestingly, the HWES model, incorporating both an additive trend and a multiplicative seasonal component, outperformed other models in terms of both MAE and RMSE. This superiority indicated its robustness in capturing both the trend and seasonal fluctuations of the dengue case time series. The predictions for the year 2024, as forecasted by this model, suggested an impending rise in dengue cases warranting immediate attention from public health authorities for potential outbreak management. Previously, various forms of the Holt-Winters method have been applied in disease studies [60], [61], [62], [63], [64], but not in the field of DENV forecasting in Bangladesh. Our study, being the first, surprisingly found that the model's ability to capture the cyclic nature of dengue cases was quite powerful. Its robustness against the volatile fluctuations in the time series data likely contributed to its superior performance. Nevertheless, in the residuals analysis, we observed some limitations. The ACF of standardized residuals and Ljung-Box tests revealed that the autocorrelation was satisfactorily addressed, except for the first lag. Since the model applied exponential smoothing, it could not fully capture the relationship between successive observations. In contrast, our SARIMA model perfectly captured the autocorrelations, although it lagged behind in the evaluation test. Furthermore, Holt-Winters method depend on alpha, beta, and gamma parameters and it's necessary to assign weights to these parameters [65]. Our study depended on the default optimization of these parameters instead of manual weight selection. This also could a probable cause for that miniscule error found in residual analysis. Despite that, the model provided a satisfactory result overall.

Notably, the Prophet model, while capturing the general seasonal pattern, fell short in projecting the more dramatic epidemic outbreaks. The model's lack of constraints to prevent negative forecasts raised concerns about its applicability to this particular domain, where case counts should not have fallen below zero. This underscored the need for careful consideration when choosing models for count data, which inherently should not predict negative values. In our study, we also examined LSTM model's ability to forecast the dengue cases in Bangladesh. However, the model performed significantly poor in contrast to the other forecasting models. Such substandard performance could be the results of a number of factors. Large dataset had been recommended for optimal functioning of the LSTM model [66], while our dataset only contained only 192 datapoints. Moreover, previous studies [67], [68], [69] that experimented LSTM to study the complex pattern of infectious diseases, opted for large scale multivariate time series analysis. So, training an LSTM model on a univariate time series with small amount of data might have not been a fine decision.

The comparison of models in this study provides valuable insights into the selection and implementation of time series forecasting models in public health. Although one model stood out

based on the evaluation metrics, each model's characteristics must also be weighed against the specific requirements and nuances of the epidemiological data being modeled. The authors believe that this research will furnish empirical insights to the policymakers so that they can formulate the proper decisions to minimize future occurrences of dengue in Bangladesh.

Chapter 6

Limitations & Future Works

Chapter 6 includes the obstacles the authors faced while conducting this study. Limitations and therefore suggestions for future research have also been discussed in this segment.

6.1 Challenges

In the course of our study, we faced some obstacles and tackled them in the most optimal ways possible. The challenges and their solutions are given below in a sequential manner. One of the first challenges was to correctly find the seasonal and non-seasonal parameters of our SARIMA model. Using differencing of the time series data, visual analysis of the ACF, PACF plot, and AIC score, we found the best parameters that intricately captured the underlying patterns and seasonal fluctuations of dengue fever. Then, the additive trend and multiplicative seasonal approach of the HWES model was unable to take zero values existing in our dataset, which was solved by changing them into ones. Although a small bias was added, it was a crucial step to solve this issue.

Unlike SARIMA, Holt-Winters didn’t provide the upper and lower bound of the forecasted values by default, so the confidence intervals were manually calculated by running a Python script. Solving the challenge of the Prophet model was relatively easy, just changing the name of the columns of our dataset cleared up the hitch. The parameters of the Prophet model were implemented strictly from our domain knowledge of the dengue occurrences in Bangladesh. Data preparation for the LSTM model was rather hectic in contrast to the other models. After the normalization of the dataset, we had to create a user-defined function to restructure the scaled data into a format suitable for LSTM training. Moreover, the predicted dataset had to be inversely transformed to its original scale as the final step.

Overall, it was quite a hurdle to accurately implement models effectively with our limited data and resources.

6.2 Limitations

The scope of the research was confined to the occurrence of dengue fever exclusively within Bangladesh. While this focus allowed for a detailed and context-specific analysis, it also posed a

limitation in terms of the generalizability of the findings. The complex patterns and seasonal behaviors observed in the dataset, and the corresponding analyses, are highly specific to the geographical, climatic, and socio-economic conditions of Bangladesh. Therefore, the applicability of the study's conclusions and the effectiveness of the predictive models may not extend accurately to other countries with different environmental and demographic settings.

The dataset used in this study was relatively small for complex statistical modeling, especially for deep learning models like LSTM. LSTM models require large datasets to effectively learn from long-term dependencies and patterns in data. The limited size and perhaps the quality of the dataset could have restricted the model's ability to learn intricate patterns in dengue case trends, leading to less accurate forecasts

Both SARIMA and Prophet models projected negative results due to their linear modality. Failure to constrain negative values was a flaw of these models, which could distort overall understanding and decision-making. Also, the Prophet model was unable to capture the previous epidemic peaks of the data undermining the model’s efficiency in predicting future epidemics or sudden outbursts of dengue fever.

The Holt-Winters Exponential Smoothing (HWES) model was noted for its minor inaccuracies in the residual analysis. These inaccuracies could be attributed to the reliance on default parameter settings. Optimizing parameters is crucial for improving model accuracy, and relying on default settings may not be adequate for capturing the unique aspects of dengue case data.

The study's exclusive focus on univariate analysis, considering only the number of dengue cases over time, was a significant limitation. This approach overlooked other potentially influential factors, such as climate variables (temperature, humidity, rainfall), demographic changes, urbanization, and public health interventions. These factors can have a substantial impact on the spread and intensity of dengue outbreaks. Multivariate analysis, which includes these variables, could provide a more comprehensive understanding and more accurate predictions. However, integrating these factors into the analysis requires access to more complex datasets and may involve more sophisticated modeling techniques.

6.3 Future Works

The identified limitations of this study provide a foundation for expanded research in this field, indicating significant opportunities for further scholarly inquiry. Some recommendations for the scope of future studies are given below. The purpose of these recommendations for future

research is to improve our knowledge in the prediction of dengue outbreaks, which will ultimately lead to improved public health outcomes. Expanding the research to include data from multiple regions or countries could provide a more generalizable understanding of dengue fever patterns. This more comprehensive approach would enable the investigation of several environmental and demographic elements influencing the spread of dengue.

Further research should focus on refining current models, particularly in terms of parameter optimization. Enhancing the accuracy and reliability of these models can be achieved by addressing the limitations identified in this study. Researchers should aim to include larger datasets with more historical data points to enhance model performance, particularly for complex deep-learning models. Incorporating climate variables will also enrich the dataset, allowing for a more comprehensive analysis. Large-scale multivariate analysis consisting of environmental and socio-economic variables can offer more nuanced insights into the complex patterns of infectious diseases like dengue.

Given the evolving nature of dengue fever transmission, employing more sophisticated machine learning and deep learning models can be beneficial. Also, ensembles of the statistical and ML models may uncover deeper insights and patterns that simpler models miss. Creating models capable of long-term forecasting would be invaluable in planning and resource allocation for public health authorities. Additionally, real-time predictive modeling could play a crucial role in rapid response and intervention strategies, helping to mitigate the impact of outbreaks as they occur.

Chapter 7

Conclusion

Containing dengue epidemics remains a critical wellness concern in tropical and semi-tropical countries, including Bangladesh. The annual dengue fever outbreak continuously threatens the country's population and healthcare organizations. This persistent problem is made worse by the nation's existing environmental circumstances, which encourage the growth of epidemics. Precise forecasting of dengue fever epidemics is essential because it allows health officials to put effective plans in place before and after unanticipated epidemic situations. Keeping that in mind, the researchers of this study aimed to forecast dengue cases in Bangladesh using established prediction models.

The research provided an in-depth analysis of the time series data of monthly DENV-affected cases in Bangladesh from 2008 to 2023. Several models, including SARIMA, Holt-Winters, Prophet, and LSTM, were employed to predict the underlying trend and variations in dengue incidence. The results indicated that the Holt-Winters Exponential Smoothing (HWES) model outperformed the other prediction models in terms of MAE, and RMSE, suggesting its effectiveness in capturing the cyclic nature of dengue cases and providing robust forecasts. This model’s forecast for 2024 indicated a significant rise in dengue cases, highlighting the need for increased public health and preparedness.

The findings underscore the paramount importance of accurate dengue epidemic forecasting, which empowers health officials to implement effective preemptive and responsive strategies in anticipation of epidemic outbreaks. The authors expect that the knowledge gained from this study will not only advance the field of epidemiological modeling but also pave the way for future studies to explore innovative methods for enhancing dengue forecasting and management strategies.

References

[1] D. J. Gubler and G. G. Clark, ‘Dengue/Dengue Hemorrhagic Fever: The Emergence of a Global Health Problem’, *Emerg. Infect. Dis.*, vol. 1, no. 2, pp. 55–57, Jun. 1995, doi: 10.3201/eid0102.952004.

[2] C. P. Simmons, J. J. Farrar, N. Van Vinh Chau, and B. Wills, ‘Dengue’, *N. Engl. J. Med.*, vol. 366, no. 15, pp. 1423–1432, Apr. 2012, doi: 10.1056/NEJMra1110265.

[3] S. B. Halstead, ‘Dengue’, *The Lancet*, vol. 370, no. 9599, pp. 1644–1652, Nov. 2007, doi: 10.1016/S0140-6736(07)61687-0.

[4] S. C. Weaver and N. Vasilakis, ‘Molecular evolution of dengue viruses: Contributions of phylogenetics to understanding the history and epidemiology of the preeminent arboviral disease’, *Infect. Genet. Evol.*, vol. 9, no. 4, pp. 523–540, Jul. 2009, doi: 10.1016/j.meegid.2009.02.003.

[5] A. Wilder-Smith, Murray, and M. Quam, ‘Epidemiology of dengue: past, present and future prospects’, *Clin. Epidemiol.*, p. 299, Aug. 2013, doi: 10.2147/CLEP.S34440.

[6] H. Harapan, A. Michie, R. T. Sasmono, and A. Imrie, ‘Dengue: A Minireview’, *Viruses*, vol. 12, no. 8, p. 829, Jul. 2020, doi: 10.3390/v12080829.

[7] S. Bhatt *et al.*, ‘The global distribution and burden of dengue’, *Nature*, vol. 496, no. 7446, pp. 504–507, Apr. 2013, doi: 10.1038/nature12060.

[8] World Health Organization, *Global strategy for dengue prevention and control 2012-2020*. Geneva: World Health Organization, 2012. Accessed: Jan. 07, 2024. [Online]. Available: https://iris.who.int/handle/10665/75303

[9] S. Sharmin, E. Viennet, K. Glass, and D. Harley, ‘The emergence of dengue in Bangladesh: epidemiology, challenges and future disease risk’, *Trans. R. Soc. Trop. Med. Hyg.*, vol. 109, no. 10, pp. 619–627, Oct. 2015, doi: 10.1093/trstmh/trv067.

[10] M. S. Hossain, A. A. Noman, S. A. A. Mamun, and A. A. Mosabbir, ‘Twenty-two years of dengue outbreaks in Bangladesh: epidemiology, clinical spectrum, serotypes, and future disease risks’, *Trop. Med. Health*, vol. 51, no. 1, p. 37, Jul. 2023, doi: 10.1186/s41182-023-00528-6.

[11] M. S. Hossain, M. H. Siddiqee, U. R. Siddiqi, E. Raheem, R. Akter, and W. Hu, ‘Dengue in a crowded megacity: Lessons learnt from 2019 outbreak in Dhaka, Bangladesh’, *PLoS Negl. Trop. Dis.*, vol. 14, no. 8, p. e0008349, Aug. 2020, doi: 10.1371/journal.pntd.0008349.

[12] P. Mutsuddy, S. Tahmina Jhora, A. K. M. Shamsuzzaman, S. M. G. Kaisar, and M. N. A. Khan, ‘Dengue Situation in Bangladesh: An Epidemiological Shift in terms of Morbidity and Mortality’, *Can. J. Infect. Dis. Med. Microbiol.*, vol. 2019, pp. 1–12, Mar. 2019, doi: 10.1155/2019/3516284.

[13] T. Shirin *et al.*, ‘Largest dengue outbreak of the decade with high fatality may be due to reemergence of DEN-3 serotype in Dhaka, Bangladesh, necessitating immediate public health attention’, *New Microbes New Infect.*, vol. 29, p. 100511, May 2019, doi: 10.1016/j.nmni.2019.01.007.

[14] F. H. Mone, S. Hossain, M. T. Hasan, G. Tajkia, and F. Ahmed, ‘Sustainable actions needed to mitigate dengue outbreak in Bangladesh’, *Lancet Infect. Dis.*, vol. 19, no. 11, pp. 1166–1167, Nov. 2019, doi: 10.1016/S1473-3099(19)30541-9.

[15] F. A. Siregar, T. Makmur, and S. Saprin, ‘Forecasting dengue hemorrhagic fever cases using ARIMA model: a case study in Asahan district’, *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 300, p. 012032, Jan. 2018, doi: 10.1088/1757-899X/300/1/012032.

[16] M. H. Riad, L. W. Cohnstaedt, and C. M. Scoglio, ‘Risk Assessment of Dengue Transmission in Bangladesh Using a Spatiotemporal Network Model and Climate Data’, *Am. J. Trop. Med. Hyg.*, vol. 104, no. 4, pp. 1444–1455, Apr. 2021, doi: 10.4269/ajtmh.20-0444.

[17] M. E. Beatty *et al.*, ‘Best Practices in Dengue Surveillance: A Report from the Asia-Pacific and Americas Dengue Prevention Boards’, *PLoS Negl. Trop. Dis.*, vol. 4, no. 11, p. e890, Nov. 2010, doi: 10.1371/journal.pntd.0000890.

[18] V. Racloz, R. Ramsey, S. Tong, and W. Hu, ‘Surveillance of Dengue Fever Virus: A Review of Epidemiological Models and Early Warning Systems’, *PLoS Negl. Trop. Dis.*, vol. 6, no. 5, p. e1648, May 2012, doi: 10.1371/journal.pntd.0001648.
[19] N. Haider, Y.-M. Chang, M. Rahman, A. Zumla, and R. A. Kock, ‘Dengue outbreaks in Bangladesh: Historic epidemic patterns suggest earlier mosquito control intervention in the transmission season could reduce the monthly growth factor and extent of epidemics’, *Curr. Res. Parasitol. Vector-Borne Dis.*, vol. 1, p. 100063, 2021, doi: 10.1016/j.crpvbd.2021.100063.
[20] K. Liu *et al.*, ‘Spatiotemporal patterns and determinants of dengue at county level in China from 2005–2017’, *Int. J. Infect. Dis.*, vol. 77, pp. 96–104, Dec. 2018, doi: 10.1016/j.ijid.2018.09.003.
[21] M. V. Kiang *et al.*, ‘Incorporating human mobility data improves forecasts of Dengue fever in Thailand’, *Sci. Rep.*, vol. 11, no. 1, p. 923, Jan. 2021, doi: 10.1038/s41598-020-79438-0.
[22] E. Descloux *et al.*, ‘Climate-Based Models for Understanding and Forecasting Dengue Epidemics’, *PLoS Negl. Trop. Dis.*, vol. 6, no. 2, p. e1470, Feb. 2012, doi: 10.1371/journal.pntd.0001470.
[23] E. Pinto, M. Coelho, L. Oliver, and E. Massad, ‘The influence of climate variables on dengue in Singapore’, *Int. J. Environ. Health Res.*, vol. 21, no. 6, pp. 415–426, Dec. 2011, doi: 10.1080/09603123.2011.572279.
[24] S. G. Kakarla *et al.*, ‘Lag effect of climatic variables on dengue burden in India’, *Epidemiol. Infect.*, vol. 147, p. e170, 2019, doi: 10.1017/S0950268819000608.
[25] S. Banu, W. Hu, Y. Guo, C. Hurst, and S. Tong, ‘Projecting the impact of climate change on dengue transmission in Dhaka, Bangladesh’, *Environ. Int.*, vol. 63, pp. 137–142, Feb. 2014, doi: 10.1016/j.envint.2013.11.002.
[26] Y. L. Hii, H. Zhu, N. Ng, L. C. Ng, and J. Rocklöv, ‘Forecast of Dengue Incidence Using Temperature and Rainfall’, *PLoS Negl. Trop. Dis.*, vol. 6, no. 11, p. e1908, Nov. 2012, doi: 10.1371/journal.pntd.0001908.
[27] S. Naher, F. Rabbi, Md. M. Hossain, R. Banik, S. Pervez, and A. B. Boitchi, ‘Forecasting the incidence of dengue in Bangladesh—Application of time series model’, *Health Sci. Rep.*, vol. 5, no. 4, p. e666, Jun. 2022, doi: 10.1002/hsr2.666.
[28] K. E. ArunKumar, D. V. Kalaga, Ch. M. Sai Kumar, G. Chilkoor, M. Kawaji, and T. M. Brenza, ‘Forecasting the dynamics of cumulative COVID-19 cases (confirmed, recovered and deaths) for top-16 countries using statistical machine learning models: Auto-Regressive Integrated Moving Average (ARIMA) and Seasonal Auto-Regressive Integrated Moving Average (SARIMA)’, *Appl. Soft Comput.*, vol. 103, p. 107161, May 2021, doi: 10.1016/j.asoc.2021.107161.
[29] M. Panda, ‘Application of ARIMA and Holt-Winters forecasting model to predict the spreading of COVID-19 for India and its states’, Public and Global Health, preprint, Jul. 2020. doi: 10.1101/2020.07.14.20153908.
[30] Y. Bai and Z. Jin, ‘Prediction of SARS epidemic by BP neural networks with online prediction strategy’, *Chaos Solitons Fractals*, vol. 26, no. 2, pp. 559–569, Oct. 2005, doi: 10.1016/j.chaos.2005.01.064.
[31] J. Xu *et al.*, ‘Forecast of Dengue Cases in 20 Chinese Cities Based on the Deep Learning Method’, *Int. J. Environ. Res. Public. Health*, vol. 17, no. 2, p. 453, Jan. 2020, doi: 10.3390/ijerph17020453.
[32] C. Xie *et al.*, ‘Trend analysis and forecast of daily reported incidence of hand, foot and mouth disease in Hubei, China by Prophet model’, *Sci. Rep.*, vol. 11, no. 1, p. 1445, Jan. 2021, doi: 10.1038/s41598-021-81100-2.
[33] B. Long, F. Tan, and M. Newman, ‘Forecasting the Monkeypox Outbreak Using ARIMA, Prophet, NeuralProphet, and LSTM Models in the United States’, *Forecasting*, vol. 5, no. 1, pp. 127–137, Jan. 2023, doi: 10.3390/forecast5010005.
[34] C. B. Aditya Satrio, W. Darmawan, B. U. Nadia, and N. Hanafiah, ‘Time series analysis and forecasting of coronavirus disease in Indonesia using ARIMA model and PROPHET’, *Procedia Comput. Sci.*, vol. 179, pp. 524–532, Jan. 2021, doi: 10.1016/j.procs.2021.01.036.

[35] J. Sadhasivam, V. Muthukumaran, J. Thimmia Raja, V. Vinothkumar, R. Deepa, and V. Nivedita, 'Applying data mining technique to predict trends in air pollution in Mumbai', *J. Phys. Conf. Ser.*, vol. 1964, no. 4, p. 042055, Jul. 2021, doi: 10.1088/1742-6596/1964/4/042055.
[36] M. Adil, N. Javaid, U. Qasim, I. Ullah, M. Shafiq, and J.-G. Choi, 'LSTM and Bat-Based RUSBoost Approach for Electricity Theft Detection', *Appl. Sci.*, vol. 10, no. 12, p. 4378, Jun. 2020, doi: 10.3390/app10124378.
[37] R. He, L. Zhang, and A. W. Z. Chew, 'Modeling and predicting rainfall time series using seasonal-trend decomposition and machine learning', *Knowl.-Based Syst.*, vol. 251, p. 109125, Sep. 2022, doi: 10.1016/j.knosys.2022.109125.
[38] P. Mishra *et al.*, 'State of the art in total pulse production in major states of India using ARIMA techniques', *Curr. Res. Food Sci.*, vol. 4, pp. 800–806, 2021, doi: 10.1016/j.crfs.2021.10.009.
[39] H. Alabdulrazzaq, M. N. Alenezi, Y. Rawajfih, B. A. Alghannam, A. A. Al-Hassan, and F. S. Al-Anzi, 'On the accuracy of ARIMA based prediction of COVID-19 spread', *Results Phys.*, vol. 27, p. 104509, Aug. 2021, doi: 10.1016/j.rinp.2021.104509.
[40] Y. Wang *et al.*, 'An Advanced Data-Driven Hybrid Model of SARIMA-NNNAR for Tuberculosis Incidence Time Series Forecasting in Qinghai Province, China', *Infect. Drug Resist.*, vol. Volume 13, pp. 867–880, Mar. 2020, doi: 10.2147/IDR.S232854.
[41] P. T. Yamak, L. Yujian, and P. K. Gadosey, 'A Comparison between ARIMA, LSTM, and GRU for Time Series Forecasting', in *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, Sanya China: ACM, Dec. 2019, pp. 49–55. doi: 10.1145/3377713.3377722.
[42] A. Kumar Dubey, A. Kumar, V. García-Díaz, A. Kumar Sharma, and K. Kanhaiya, 'Study and analysis of SARIMA and LSTM in forecasting time series data', *Sustain. Energy Technol. Assess.*, vol. 47, p. 101474, Oct. 2021, doi: 10.1016/j.seta.2021.101474.
[43] K. K. R. Samal, K. S. Babu, S. K. Das, and A. Acharaya, 'Time Series based Air Pollution Forecasting using SARIMA and Prophet Model', in *Proceedings of the 2019 International Conference on Information Technology and Computer Communications*, Singapore Singapore: ACM, Aug. 2019, pp. 80–85. doi: 10.1145/3355402.3355417.
[44] R. W. Divisekara, G. J. M. S. R. Jayasinghe, and K. W. S. N. Kumari, 'Forecasting the red lentils commodity market price using SARIMA models', *SN Bus. Econ.*, vol. 1, no. 1, p. 20, Jan. 2021, doi: 10.1007/s43546-020-00020-x.
[45] M. R. Abonazel and A. I. Abd-Elftah, 'Forecasting Egyptian GDP using ARIMA models', *Rep. Econ. Finance*, vol. 5, no. 1, pp. 35–47, 2019, doi: 10.12988/ref.2019.81023.
[46] I. Djakaria and S. E. Saleh, 'Covid-19 forecast using Holt-Winters exponential smoothing', *J. Phys. Conf. Ser.*, vol. 1882, no. 1, p. 012033, May 2021, doi: 10.1088/1742-6596/1882/1/012033.
[47] A. I. Almazrouee, A. M. Almeshal, A. S. Almutairi, M. R. Alenezi, S. N. Alhajeri, and F. M. Alshammari, 'Forecasting of Electrical Generation Using Prophet and Multiple Seasonality of Holt–Winters Models: A Case Study of Kuwait', *Appl. Sci.*, vol. 10, no. 23, p. 8412, Nov. 2020, doi: 10.3390/app10238412.
[48] Y. Dang, Z. Chen, H. Li, and H. Shu, 'A Comparative Study of non-deep Learning, Deep Learning, and Ensemble Learning Methods for Sunspot Number Prediction', *Appl. Artif. Intell.*, vol. 36, no. 1, p. 2074129, Dec. 2022, doi: 10.1080/08839514.2022.2074129.
[49] W. Zha *et al.*, 'Forecasting monthly gas field production based on the CNN-LSTM model', *Energy*, vol. 260, p. 124889, Dec. 2022, doi: 10.1016/j.energy.2022.124889.
[50] Le, Ho, Lee, and Jung, 'Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting', *Water*, vol. 11, no. 7, p. 1387, Jul. 2019, doi: 10.3390/w11071387.
[51] R. Chandra, A. Jain, and D. Singh Chauhan, 'Deep learning via LSTM models for COVID-19 infection forecasting in India', *PLOS ONE*, vol. 17, no. 1, p. e0262708, Jan. 2022, doi: 10.1371/journal.pone.0262708.
[52] S. Nazar, J. Yang, W. Ahmad, M. F. Javed, H. Alabduljabbar, and A. F. Deifalla, 'Development of the New Prediction Models for the Compressive Strength of Nanomodified Concrete Using Novel

Machine Learning Techniques', *Buildings*, vol. 12, no. 12, p. 2160, Dec. 2022, doi: 10.3390/buildings12122160.
[53] K. Khan, W. Ahmad, M. N. Amin, F. Aslam, A. Ahmad, and M. A. Al-Faiad, 'Comparison of Prediction Models Based on Machine Learning for the Compressive Strength Estimation of Recycled Aggregate Concrete', *Materials*, vol. 15, no. 10, p. 3430, May 2022, doi: 10.3390/ma15103430.
[54] M. E. H. Kayesh, I. Khalil, M. Kohara, and K. Tsukiyama-Kohara, 'Increasing Dengue Burden and Severe Dengue Risk in Bangladesh: An Overview', *Trop. Med. Infect. Dis.*, vol. 8, no. 1, p. 32, Jan. 2023, doi: 10.3390/tropicalmed8010032.
[55] K. K. Paul, I. Macadam, D. Green, D. G. Regan, and R. T. Gray, 'Dengue transmission risk in a changing climate: Bangladesh is likely to experience a longer dengue fever season in the future', *Environ. Res. Lett.*, vol. 16, no. 11, p. 114003, Oct. 2021, doi: 10.1088/1748-9326/ac2b60.
[56] M. A. Islam *et al.*, 'Correlation of Dengue and Meteorological Factors in Bangladesh: A Public Health Concern', *Int. J. Environ. Res. Public. Health*, vol. 20, no. 6, Art. no. 6, Jan. 2023, doi: 10.3390/ijerph20065152.
[57] Cong, Ren, Xie, and Wang, 'Predicting Seasonal Influenza Based on SARIMA Model, in Mainland China from 2005 to 2018', *Int. J. Environ. Res. Public. Health*, vol. 16, no. 23, p. 4760, Nov. 2019, doi: 10.3390/ijerph16234760.
[58] H. Liu *et al.*, 'Forecast of the trend in incidence of acute hemorrhagic conjunctivitis in China from 2011–2019 using the Seasonal Autoregressive Integrated Moving Average (SARIMA) and Exponential Smoothing (ETS) models', *J. Infect. Public Health*, vol. 13, no. 2, pp. 287–294, Feb. 2020, doi: 10.1016/j.jiph.2019.12.008.
[59] Y. Xiao *et al.*, 'Estimating the Long-Term Epidemiological Trends and Seasonality of Hemorrhagic Fever with Renal Syndrome in China', *Infect. Drug Resist.*, vol. Volume 14, pp. 3849–3862, Sep. 2021, doi: 10.2147/IDR.S325787.
[60] D. King *et al.*, 'Changing patterns in the epidemiology and outcomes of inflammatory bowel disease in the United Kingdom: 2000-2018', *Aliment. Pharmacol. Ther.*, vol. 51, no. 10, pp. 922–934, May 2020, doi: 10.1111/apt.15701.
[61] D. A. Adeyinka and N. Muhajarine, 'Time series prediction of under-five mortality rates for Nigeria: comparative analysis of artificial neural networks, Holt-Winters exponential smoothing and autoregressive integrated moving average models', *BMC Med. Res. Methodol.*, vol. 20, no. 1, p. 292, Dec. 2020, doi: 10.1186/s12874-020-01159-9.
[62] Y. Li *et al.*, 'Trends of surgical treatment for spinal degenerative disease in China: a cohort of 37,897 inpatients from 2003 to 2016', *Clin. Interv. Aging*, vol. Volume 14, pp. 361–366, Feb. 2019, doi: 10.2147/CIA.S191449.
[63] X. Xian *et al.*, 'Comparison of SARIMA model, Holt-winters model and ETS model in predicting the incidence of foodborne disease', *BMC Infect. Dis.*, vol. 23, no. 1, p. 803, Nov. 2023, doi: 10.1186/s12879-023-08799-4.
[64] H. Qiu *et al.*, 'Forecasting the incidence of acute haemorrhagic conjunctivitis in Chongqing: a time series analysis', *Epidemiol. Infect.*, vol. 148, p. e193, 2020, doi: 10.1017/S095026882000182X.
[65] R. Siddiqui, M. Azmat, S. Ahmed, and S. Kummer, 'A hybrid demand forecasting model for greater forecasting accuracy: the case of the pharmaceutical industry', *Supply Chain Forum Int. J.*, vol. 23, no. 2, pp. 124–134, Apr. 2022, doi: 10.1080/16258312.2021.1967081.
[66] K. Cho and Y. Kim, 'Improving streamflow prediction in the WRF-Hydro model with LSTM networks', *J. Hydrol.*, vol. 605, p. 127297, Feb. 2022, doi: 10.1016/j.jhydrol.2021.127297.
[67] A. B. Said, A. Erradi, H. A. Aly, and A. Mohamed, 'Predicting COVID-19 cases using bidirectional LSTM on multivariate time series', *Environ. Sci. Pollut. Res.*, vol. 28, no. 40, pp. 56043–56052, Oct. 2021, doi: 10.1007/s11356-021-14286-7.
[68] Y.-T. Tsan, D.-Y. Chen, P.-Y. Liu, E. Kristiani, K. L. P. Nguyen, and C.-T. Yang, 'The Prediction of Influenza-like Illness and Respiratory Disease Using LSTM and ARIMA', *Int. J. Environ. Res. Public. Health*, vol. 19, no. 3, p. 1858, Feb. 2022, doi: 10.3390/ijerph19031858.

[69] E. Mussumeci and F. Codeço Coelho, ‘Large-scale multivariate forecasting models for Dengue - LSTM versus random forest regression’, *Spat. Spatio-Temporal Epidemiol.*, vol. 35, p. 100372, Nov. 2020, doi: 10.1016/j.sste.2020.100372.