# **Introduction:**

Dengue, commonly known as break-bone fever, is a viral illness transmitted to humans through Aedes mosquitoes during their larval stages when they harbor the dengue virus. Its prevalence is higher in tropical and subtropical regions (1). The dengue virus comprises four distinct serotypes: DENV-1, DENV-2, DENV-3, and DENV-4, originating from sylvatic strains in Southeast Asian forests. Recently, a new serotype, DENV-5, has been identified exclusively in the sylvatic cycle (2). Symptoms of dengue fever typically manifest when the person is in critical condition, as significant symptoms are initially mild. Seroepidemiological studies reveal that the immune system activates upon infection, providing resistance to the specific dengue serotype (3). Severe cases can lead to life-threatening complications like dengue hemorrhagic fever. It is estimated that around 3.6 billion people reside in regions where dengue transmission is possible, with an annual occurrence of 50 million to 200 million dengue infections, including 500,000 severe cases and over 20,000 dengue-related deaths (4). As of December 31st, 2022, a total of 4,110,465 dengue cases and 4099 fatalities have been reported (5). Additional symptoms include a metallic taste, loss of appetite, diarrhea, nausea, stomachache, severe headaches, fatigue, myalgia, and painful joints. The term 'break-bone fever' stems from the myalgia and joint discomfort often experienced (6). The proliferation of Aedes aegypti and sustained dengue transmission is influenced by various factors, including serotype circulation, human movement, mosquito population size, and local climate conditions like temperature and precipitation (7, 8). The rainy monsoon significantly aids Aedes mosquito reproduction, making Bangladesh conducive to mosquito breeding throughout the year, especially during the monsoon season (9, 10).

In Bangladesh, a study aimed to predict dengue outbreaks from 2000 to 2008, focusing on seasonal climate data. The study employed feature selection and cross-validation to develop and assess a prediction model (11). Another study by Dourjoy et al. used machine learning, specifically Support Vector Machine (SVM) and Random Forest (RF), achieving accuracies of 68 percent and 64 percent, respectively, in predicting dengue incidences in Bangladesh (12). Majeed et al. successfully developed and validated an LSTM model with spatial attention for dengue prediction, demonstrating its effectiveness in predicting dengue cases in Malaysia (13). Another study in Bangladesh aimed to build a machine learning model utilizing the DengueBD dataset and employing Multiple Linear Regression (MLR) and Support Vector Regression (SVR). The results illustrated the significant potential of machine learning algorithms in predicting dengue epidemics (9).

However, the mentioned study focused solely on patient data patterns, overlooking crucial factors like meteorological data, socioeconomic influences, geographic location, and weather correlations with monthly cases. Furthermore, the study lacked result validation and had a limited dataset for application in machine learning approaches. Hence, this current study seeks to predict dengue virus occurrences by analyzing past admissions of dengue patients. To track dengue patient data, this research utilized officially released information from the Directorate General of Health Services (DGHS) under the Ministry of Health and Family Welfare, responsible for health services in Bangladesh. Alongside recorded dengue cases, daily weather data from the Bangladesh Meteorological Department (www.bmd.gov.bd) was also utilized. This study explored three different machine learning models (Long Short-Term Memory (LSTM) Network, Recurrent Neural Network (RNN), and Gated Recurrent Units (GRU)) and evaluated their performance using Mean Absolute Error (MAE), RMSE, and AUROC.

# **Methodology:**

## *Long Short-Term Memory (LSTM) Network:*

The Long Short-Term Memory (LSTM) neural network, often referred to simply as LSTM, is recognized as an improved version of the Recurrent Neural Network (RNN) class. Its origin can be attributed to Hochreiter and Schmidhuber [1], who introduced LSTM in 1997. Over the years, LSTM has gained increasing prominence and has found applications across diverse domains and challenges. What sets LSTM apart from the conventional RNN is its remarkable capacity to retain information over extended time intervals, a characteristic known as long-term dependency. This architectural attribute enables LSTM to overcome the limitations of standard RNNs.

An LSTM module utilizes a gating mechanism to regulate the information flow within its memory cells, also overseeing the duration of information retention. Comprising three distinct gate types - the input gate (it), forget gate (ft), and output gate (ot) - each memory cell also accommodates a cell state that facilitates information movement and interconnects with the gates. These gates manage the quantity of data directed to the cell state (Ct). At a specific time instance, denoted as "t," all three gates receive input parameters: the new input (xt), the prior time step's output or hidden state (ht-1), and the previous cell state (Ct-1) incorporated into the memory cell. The forget gate (ft) determines the extent to which past information (ht-1) is to be discarded, allowing certain portions to pass and be stored in the cell state. Meanwhile, the input gate (it) controls the amount of new input (xt) to permit and store within the cell state. Lastly, the output gate (ot) ascertains the quantity of information to extract and filter from the current cell state, generating a new hidden state (ht). A basic illustration of an LSTM memory cell is provided in Figure 1.

A diagram of a flowchart

Description automatically generated

Figure 1: LSTM memory cell structure

## *Recurrent Neural Network (RNN):*

The fundamental architecture for predicting recurrent neural networks is presented in Figure 2 [2]. An input sequence vector, denoted as *x = (x1, . . . , xT)*, undergoes processing through weighted connections across a series of *N* hidden layers which are recurrently connected. This process computes the hidden sequence vectors *hn = (hn1, . . . , hnT),* followed by the output sequence vector *y = (y1, . . . , yT).* Each output vector *yt* serves as the basis for defining a predictive distribution, *Pr(xt+1|yt)*, which characterizes the potential subsequent inputs *xt+1*. Notably, the initial element *x1* within each input sequence always consists of a null vector containing all zero entries. Consequently, the network generates a prediction for *x2*, the actual first input, without any preceding information. Remarkably, the network exhibits depth in both spatial and temporal dimensions. This implies that any information traversing the computational graph, either vertically or horizontally, encounters successive weight matrices and nonlinear operations.

A diagram of a diagram

Description automatically generated

Figure 2: Recurrent Neural Network (RNN) architecture

## *Gated Recurrent Units (GRU):*

Gated Recurrent Units (GRU), a variant of the RNN, offers enhanced performance with a simpler structure [3, 4]. Similar to RNN and LSTM, GRU can also be employed for time-series predictions. The model architecture of GRU, illustrated in Figure 3, consists of just two components: a reset gate (*rt*) and an update gate (*zt*). The reset gate (*rt*) governs the extent to which information from the new input (*xt*) is incorporated into the current hidden layer vector from the previous state. On the other hand, the update gate (*zt*) manages the retention of the previous hidden layer (*ht-1*). In comparison to LSTM, GRU boasts a more streamlined design with fewer gates, leading to increased efficiency.

**Results and Discussions**

**Data Collection Overview**: From January 2000 up to December 2022, we gathered data on the total number of dengue cases. In addition to this, we also recorded relevant environmental factors that might influence dengue transmission. These factors included average wind speed, temperature, dew point, relative humidity, rainfall, and surface pressure.

**Data Preprocessing Steps**:

**Lookback Window Implementation**: A lookback window of 7 days was utilized, implying that we're predicting the dengue cases on a given day (let's say "t") based on the data from the previous seven days (t-1 to t-7).

**Normalization**: Considering that the data contained differences in scales and units, normalization was essential. We employed the sklearn preprocessing tool to achieve this, specifically using the MinMaxScalar. This ensured that our data values were adjusted to fall within a range of -1 to 1.

**Determining Inputs and Outputs**:

**Inputs (X)**: The data that fed into our model consisted of the following parameters with a 7-day lookback window:

Number of dengue cases

Wind speed

Temperature

Dew point

Relative humidity

Rainfall

Surface pressure

**Output (y)**: The primary goal was to predict the number of dengue cases based on the aforementioned inputs (X).

**Data Splitting**: The dataset was divided into a 95:5 ratio. The majority (95%) was dedicated to training the model, while the remaining 5% was reserved for testing its accuracy. For efficient processing, the data was organized into batches of 16.

**Model Construction with LSTM**: Our chosen model was the LSTM (Long Short-Term Memory), which we built using 10 hidden layers. For optimization, we deployed the well-known Adam optimizer and set the learning rate at 0.01. To gauge the model's performance and accuracy, we used the Mean Squared Error (MSE) as our loss function. The training process involved 200 epochs for updating the model's weights. This experiment was run on an Intel Core i7-2.4GHz processor workstation equipped with 16GB RAM and an NVIDIA GeForce 940MX graphics card, utilizing Pytorch as the main platform.

**LSTM Model Results & Analysis**: Figure 1 displays a month-wise comparison of actual dengue cases against those predicted by our LSTM model, covering the period from January 2000 to October 2021. For training, we utilized both weather data and actual dengue cases up to October 2021. Figure 2 then presents a direct comparison between LSTM-predicted dengue cases and actual figures. The model's effectiveness was assessed using a specific accuracy formula, which revealed that our LSTM model achieved an accuracy rate of 87.98%. The formula is given below:

**Modeling with RNN**: Our chosen architecture for this phase of the project was the Recurrent Neural Network (RNN). We built our RNN with a total of 10 hidden layers. For the training process, we employed the Adam optimizer with a learning rate set at 0.01. To assess the effectiveness of our model, we used the Mean Squared Error (MSE) as our standard loss function. We subjected our model to a rigorous training process that spanned 200 epochs. All these computational tasks were executed on an Intel Core i7-2.4GHz processor with 16GB RAM, and NVIDIA GeForce 940MX graphics. Pytorch served as our main software platform.

**RNN Results & Analysis**: In Figure 3, we have depicted a month-by-month comparison between actual dengue cases and those predicted by our RNN for the period spanning January 2000 to October 2021. This period also constituted our training data, combining weather patterns and actual dengue cases. Figure 4 offers a clearer visual of the RNN's predictive capacity juxtaposed against the actual dengue case data. After computation, our RNN model's accuracy stood at 57.18%.

**Modeling with GRU**: For this segment, we opted for the Gated Recurrent Units (GRU) model, structured with 10 hidden layers. Similar to the previous models, we engaged the Adam optimizer for training, setting the learning rate at 0.01. The Mean Squared Error (MSE) was once again our chosen metric for gauging model performance. The GRU model underwent 200 epochs of weight adjustment. The computational setup remained consistent: Intel Core i7-2.4GHz, 16GB RAM, and NVIDIA GeForce 940MX graphics, all running on the Pytorch platform.

**GRU Results & Analysis**: Figure 5 delineates the month-wise comparative analysis of the actual dengue cases versus the predictions made by the GRU model from January 2000 to October 2021. This same period, enriched by weather data and dengue case records, was the foundation of our training data. Figure 6 showcases the GRU's prediction against the actual dengue statistics. Upon evaluating our model, we found that the GRU yielded an accuracy of 79.81%.

In summary, among the three models (LSTM, RNN, and GRU), LSTM showed the highest accuracy at 87.98%, followed by GRU at 79.81%, and RNN lagged behind with 57.18%.

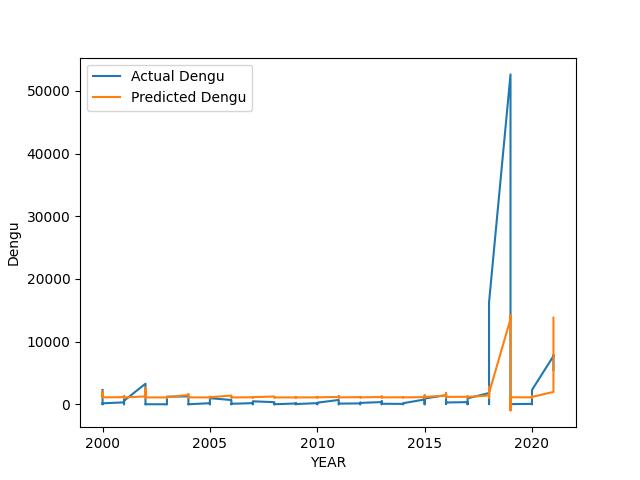


Figure 1: Actual dengue vs. predicted dengue using LSTM

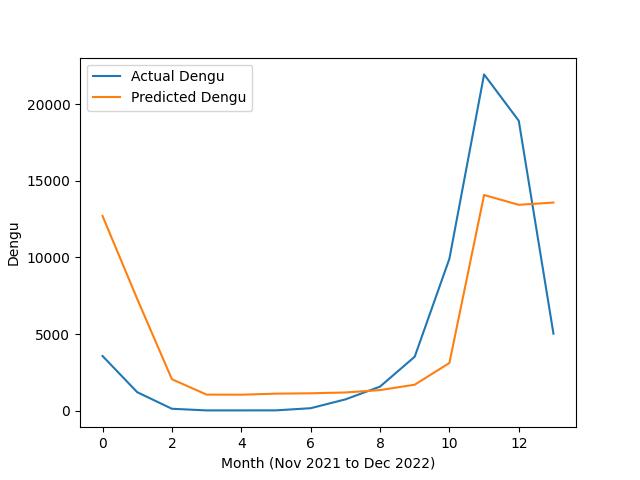


Figure 2: Testing LSTM Actual dengue vs. predicted dengue

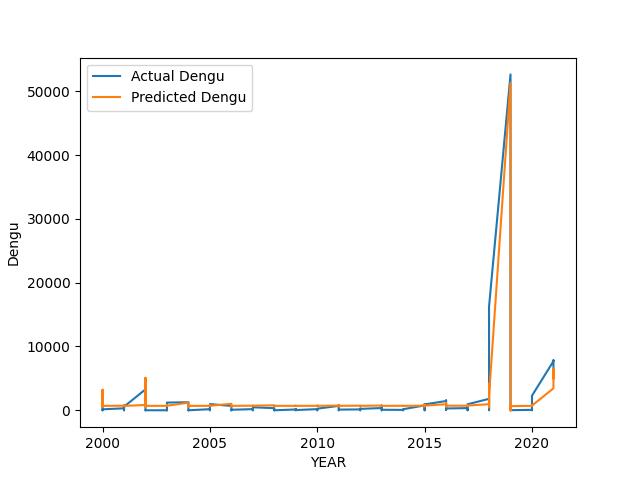


Figure 3: Actual dengue vs. predicted dengue using RNN

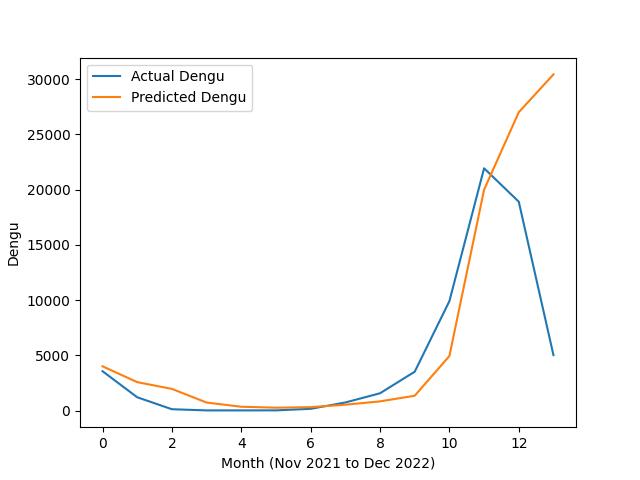


Figure 4: Testing RNN Actual dengue vs. predicted dengue

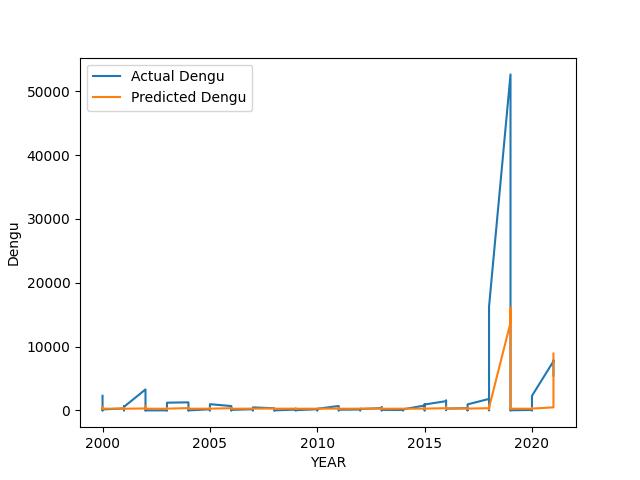


Figure 5: Actual dengue vs. predicted dengue using GRU

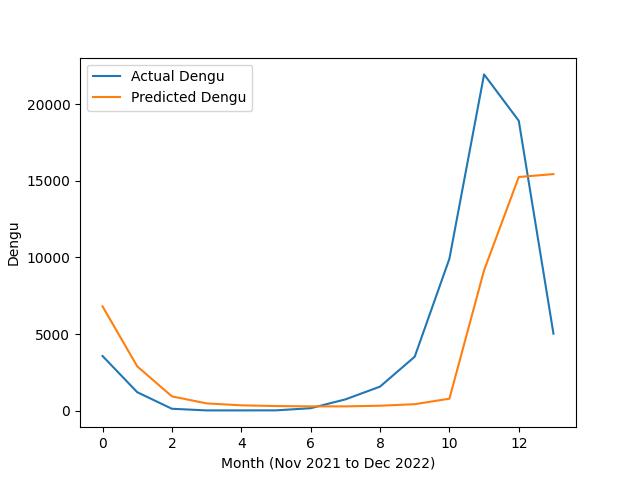


Figure 6: Testing GRU Actual dengue vs. predicted dengue

**Discussion:**

The results from the comparison indicate that the LSTM model outperforms others, displaying the highest accuracy and the lowest average RMSE. Notably, the GRU models perform better than the RNN model, irrespective of their utilization of attention mechanisms. This implies that these models might excel in capturing intricate data patterns and making precise predictions. However, it's crucial to acknowledge that the outcomes of this comparison may not universally apply to other datasets or tasks. Evaluating these models on diverse datasets and tasks is essential to validate their relative performance in those particular contexts.

Several potential reasons could explain why LSTM models, especially those with attention mechanisms and stacked LSTM layers, demonstrated superior performance in this comparison. One reason might be their enhanced capability to process a diverse and intricate set of input variables, encompassing climate, topographic, demographic, and land-use variables, all potentially pertinent for dengue case prediction. For instance, climate factors like temperature and humidity can impact the survival and breeding of dengue-transmitting mosquitoes, while topographic elements such as water bodies can influence mosquito breeding site distribution. Demographic factors like population density and land-use metrics such as urbanization levels can also influence dengue transmission. LSTM networks are adept at modeling time series data and capturing patterns across multiple time steps, which proves particularly beneficial in predicting dengue fever affected by both short-term and long-term factors. Spatial attention utilization enables the model to assign varying weights to features during predictions, aiding in focusing on the most relevant factors, like environmental elements or population density, thereby enhancing prediction accuracy. Additionally, LSTM networks handle missing data more effectively by retaining information from preceding time steps, which proves valuable in scenarios with missing or incomplete data.

All models demonstrate the ability to make accurate predictions regarding disease progression and enable early disease diagnosis. A comprehensive understanding of the disease's etiology and the development of precise diagnostic models would significantly aid in combatting dengue fever and reducing associated complications and fatalities. Addressing data uncertainty through modeling is another crucial aspect. However, it's important to recognize potential limitations of using LSTM for dengue fever prediction. LSTM networks can be intricate and demand substantial data and computational resources for effective training, posing challenges in practical implementation. Moreover, these networks are often challenging to interpret as they operate as black-box models, hindering insight into the specific features or patterns driving predictions and potentially impeding the identification of biases in the data.

## ***References:***

1. S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.
2. Graves, A., 2013. Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850.
3. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.
4. Zhao, R., Wang, D., Yan, R., Mao, K., Shen, F. and Wang, J., 2017. Machine health monitoring using local feature-based gated recurrent unit networks. IEEE Transactions on Industrial Electronics, 65(2), pp.1539-1548.

1. WHO. Dengue and severe dengue [Internet]. 2023 [cited 2023 Sep 26]. Available from: https://www.who.int/news-room/fact-sheets/detail/dengue-and-severe-dengue

2. Kayesh MEH, Khalil I, Kohara M, Tsukiyama-Kohara K. Increasing Dengue Burden and Severe Dengue Risk in Bangladesh: An Overview. Trop Med Infect Dis [Internet]. 2023 Jan 1 [cited 2023 Sep 26];8(1). Available from: /pmc/articles/PMC9866424/

3. Stolerman LM, Maia PD, Nathan Kutz J. Forecasting dengue fever in Brazil: An assessment of climate conditions. PLoS One [Internet]. 2019 Aug 1 [cited 2023 Sep 26];14(8):e0220106. Available from: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0220106

4. Murray NEA, Quam MB, Wilder-Smith A. Epidemiology of dengue: past, present and future prospects. Clin Epidemiol [Internet]. 2013 Aug 19 [cited 2023 Sep 26];5(1):299. Available from: /pmc/articles/PMC3753061/

5. DGHS. Dengue Press Release [Internet]. 2023 [cited 2023 Sep 26]. Available from: https://old.dghs.gov.bd/index.php/bd/home/5200-daily-dengue-status-report

6. Gupta G, Khan S, Guleria V, Almjally A, Alabduallah BI, Siddiqui T, et al. DDPM: A Dengue Disease Prediction and Diagnosis Model Using Sentiment Analysis and Machine Learning Algorithms. Diagnostics [Internet]. 2023 Mar 1 [cited 2023 Sep 26];13(6):1093. Available from: /pmc/articles/PMC10047105/

7. Teurlai M, Menkès CE, Cavarero V, Degallier N, Descloux E, Grangeon JP, et al. Socio-economic and Climate Factors Associated with Dengue Fever Spatial Heterogeneity: A Worked Example in New Caledonia. PLoS Negl Trop Dis [Internet]. 2015 Dec 1 [cited 2023 Sep 26];9(12). Available from: /pmc/articles/PMC4666598/

8. Wu X, Lu Y, Zhou S, Chen L, Xu B. Impact of climate change on human infectious diseases: Empirical evidence and human adaptation. Environ Int. 2016 Jan 1;86:14–23.

9. Dey SK, Rahman MM, Howlader A, Siddiqi UR, Uddin KMM, Borhan R, et al. Prediction of dengue incidents using hospitalized patients, metrological and socio-economic data in Bangladesh: A machine learning approach. PLoS One [Internet]. 2022 Jul 1 [cited 2023 Sep 26];17(7). Available from: /pmc/articles/PMC9299345/

10. Hashizume M, Dewan AM, Sunahara T, Rahman MZ, Yamamoto T. Hydroclimatological variability and dengue transmission in Dhaka, Bangladesh: A time-series study. BMC Infect Dis [Internet]. 2012 Apr 24 [cited 2023 Sep 26];12(1):1–9. Available from: https://bmcinfectdis.biomedcentral.com/articles/10.1186/1471-2334-12-98

11. Pear M, Id H, Zhou W, Ren C, Marshall J, Yuan HY. Prediction of dengue annual incidence using seasonal climate variability in Bangladesh between 2000 and 2018. PLOS Global Public Health [Internet]. 2022 May 9 [cited 2023 Sep 26];2(5):e0000047. Available from: https://journals.plos.org/globalpublichealth/article?id=10.1371/journal.pgph.0000047

12. Dourjoy SMK, Rafi AMGR, Tumpa ZN, Saifuzzaman M. A Comparative Study on Prediction of Dengue Fever Using Machine Learning Algorithm. Lecture Notes in Networks and Systems [Internet]. 2021 [cited 2023 Sep 26];127:501–10. Available from: https://link.springer.com/chapter/10.1007/978-981-15-4218-3\_49

13. Majeed MA, Shafri HZM, Zulkafli Z, Wayayok A. A Deep Learning Approach for Dengue Fever Prediction in Malaysia Using LSTM with Spatial Attention. Int J Environ Res Public Health [Internet]. 2023 Mar 1 [cited 2023 Sep 26];20(5). Available from: /pmc/articles/PMC10002017/