A windowed correlation based feature selection method to improve time series prediction of dengue fever cases

Tanvir Ferdousi, Lee W. Cohnstaedt, and Caterina M. Scoglio

Abstract—The performance of data-driven prediction models depends on the availability of data samples for model training. A model that learns about dengue fever incidence in a population uses historical data from that corresponding location. Poor performance in prediction can result in places with inadequate data. This work aims to enhance temporally limited dengue case data by methodological addition of epidemically relevant data from nearby locations as predictors (features). A novel framework is presented for windowing incidence data and computing time-shifted correlation-based metrics to quantify feature relevance. The framework ranks incidence data of adjacent locations around a target location by combining the correlation metric with two other metrics: spatial distance and local prevalence. Recurrent neural network-based prediction models achieve up to 33.6% accuracy improvement on average using the proposed method compared to using training data from the target location only. These models achieved mean absolute error (MAE) values as low as 0.128 on [0,1] normalized incidence data for a municipality with the highest dengue prevalence in Brazil’s Espírito Santo. When predicting cases aggregated over geographical ecoregions, the models achieved accuracy improvements up to 16.5%, using only 6.5% of incidence data from ranked feature sets. The paper also includes two techniques for windowing time series data: fixed-sized windows and outbreak detection windows. Both of these techniques perform comparably, while the window detection method uses less data for computations. The framework presented in this paper is application-independent, and it could improve the performances of prediction models where data from spatially adjacent locations are available.

Index Terms—Dengue, Feature Selection, Machine Learning, Recurrent Neural Networks, Time Series.

1 INTRODUCTION

Accurate time series prediction of dengue fever outbreaks can be useful in planning mitigation strategies for hundreds of tropical and subtropical regions around the world. Data-driven models such as neural networks are flexible in design and can ease the difficulties of estimating unknown parameters that mechanistic models often require [1]. However, the prediction accuracy of such models depends on the quality and the quantity of training data. For dengue fever outbreaks, the availability of incidence data varies across regions. A data aggregation center in a region may not have adequate data to achieve an acceptable level of accuracy in out-of-sample projections. In such cases, selected incidence data from adjacent centers in the same region could improve model performance as additional features. We propose a framework with quantitative methodologies to rank and select nearby case data as supplementary features. Our method uses windowed time-lagged cross-correlation combined with distance and prevalence metrics to identify relevances and potential causal relationships.

Dengue virus is primarily spread by several species of mosquito vectors (Aedes aegypti and Aedes albopictus), and the outbreaks infect 390 million people every year [2]. The viral strains also cause about 40,000 annual deaths with hemorrhagic fever, and dengue shock syndrome [3]. Dengue virus transmission is prevalent in regions where competent vector mosquitoes are present. In those regions, the mosquito abundance varies throughout the year and depends on factors including air temperature, precipitation, vegetation, and urbanization [4], [5], [6]. The availability of extensive data on these factors makes statistical and machine learning analyses feasible. However, researchers experience missing data, uneven reporting intervals, lack of granularity, inadequacy, and accuracy with dengue case counts for many regions around the globe. These factors limit the prediction performance of data-driven models. To complicate the situation further, time series outbreak data for most diseases are non-stationary (e.g., the underlying processes evolve with time). In many regions, climatic variations affect Aedes vector populations [7]. In addition to that, outbreak start times and sizes may vary because of imported cases caused by short and long-range mobility. Co-circulation of multiple viral strains adds to the complexity. Despite those issues, correlations exist between outbreaks in adjacent human populations (county, municipality, district, etc.) for most communicable diseases, including dengue [8]. While a correlation may not always imply causation, the use of incidence data from adjacent regions can improve the model training because of similarities in meteorological
factors, host population density, and a high probability of mobility-based viral transmissions.

For sequential data (e.g., multivariate time series), different methods of feature selection have been used in the past including correlation-based filters [9], [10]. Granger causality tests [11], [12], genetic algorithm [13], and several other methods [14], [15]. The applications include forecasts of electrical energy consumption, meteorological variables, financial markets, etc. Correlation-based methods are widely used in machine learning problems [16], [17]. Few works extend feature sets for disease outbreak prediction using incidence data from spatially adjacent locations. Such geospatial clustering techniques rely on similarity metrics to improve model performance. A recent work [18] uses pair-wise correlation to cluster location data to train models for COVID-19 outbreak projection in Chinese provinces. Another work targeted towards dengue also uses correlation-based similarity measures to extend the training feature set [19]. However, these implementations assume an instantaneous correlation of incidence data between regions and do not consider the temporal order of outbreaks (whether one location is leading or lagging the other). There can be a considerable amount of time delay between outbreaks occurring in adjacent regions. Time lagged cross-correlation can help identify such phase relations [20]. The phase information may help quantify relationships between outbreaks of adjacent locations. We hypothesize that if the temporal incidence dynamics of one outbreak lead the incidence dynamics of another, the former outbreak might have a causal influence on the latter, given that these places lie spatially close enough to affect one another, and the outbreaks are of a reasonable size. We weigh the available features (incidence data) based on a combination of these factors: leading phase correlation, distance, and prevalence. Because of the population’s dynamically changing immunity patterns caused by the co-circulation of multiple dengue virus strains, the regions experiencing large outbreaks may also evolve. A single computation of the correlation coefficient over the entire timeline of data may mislead the analysis. Hence, splitting the time series into multiple windows and comparing sequences at each time window for correlation and phase analyses can provide better insights into the dengue outbreak’s seasonal patterns. A limitation with some existing methods is that clustering large data sets for model training can make the process complex and inefficient, while unrelated features may deteriorate prediction performance. This phenomenon is known as the curse of dimensionality [21], [22]. This work aims to reach optimal model performance with the smallest subset of highly relevant features based on their ranks. There are two broad categories of feature selection methods: filters and wrappers [23]. Wrapper approaches [24] are computationally expensive as the search space for optimal feature subsets increases exponentially with the number of available features. Our method is primarily a filter approach to rank features.

In this work, we present a framework of feature enhancement for data-driven prediction of dengue outbreaks. To achieve that goal, this paper details: i) a windowed time-shifted cross-correlation method to compute correlation weights, ii) two correlation-window allocation methods, iii) a procedure of ranking feature using metrics based on correlation, distance, and prevalence, and iv) analysis of prediction performance across windowing schemes, prediction models, and spatial aggregations. This work’s novel contribution lies in how we interpret and process incidence data to compute correlation metrics using our knowledge of how outbreaks spread.

2 PRELIMINARIES

2.1 Definitions

In supervised learning problems, we collect data on multiple variables. A target variable (label) is the designated output of a machine learning model for prediction. A feature is an input variable that is expected to influence the target variable. Each instance of data (e.g., a point in time) contains several feature values and usually a single label value. A data set comprises many such instances. For a supervised learning problem, a data set is split into 3 subsets in order to: train, evaluate, and test the models. With the training subset fed in batches (collection of instances), a supervised model learns to predict targets based on features in an iterative process. It optimizes parameters by minimizing a loss function. A neural network comprises artificial neurons (cells) in one or multiple layers. Each neural cell is a node in the network with connections to other nodes across layers, inputs, or outputs. The recurrent neural networks (RNN) are special neural cells that can store internal states in their memory, which helps them predict sequence data (data points that are temporally related) better. Model training aims to get optimal parameter values to generalize beyond the training data and perform well with test data. Sometimes, a model can over-fit the training data and perform poorly with unseen test examples. To prevent such scenarios, we use regularization techniques.

2.2 Time series prediction of outbreaks

Time series forecasting is a popular research area because of its applicability in many disciplines, such as forecasting weather patterns, stock prices, market trends, and resource allocation. In epidemiology, these methods enable the prediction of future outbreaks by fitting models with past disease incidence data and carefully chosen covariates. Such predictions come at varying degrees of accuracy and depend on the characteristics of the target variable, quality of sample data available for fitting, the covariates (predictors) being used, and the models themselves. There is rarely a single model that works best for every application. Classical forecasting methods such as exponential smoothing, autoregressive integrated moving average (ARIMA), and seasonal autoregressive integrated moving average (seasonal ARIMA) have been widely used to predict time series data [25]. The ARIMA model can handle non-stationary data, which is an important advantage. Besides that, the seasonal ARIMA model can incorporate repeating patterns in the data to enable forecasting of diseases that show seasonal patterns [26]. However, these models have tendencies to follow the mean values of past data, and it is not easy to associate these with rapidly changing processes [27]. In addition to that, many classical models require manual tuning of their parameters and may fail to capture
complex nonlinear interactions. Data-driven forecasting of vector-borne diseases such as dengue fever is complicated due to complex interactions of several factors with disease dynamics. A list of these factors include but is not limited to seasonally dependent *Aedes* mosquito growth and feeding patterns [28], co-circulations of multiple viral strains [29], environmental (e.g., temperature) effects on dengue virus transmission [30], and human mobility patterns [31]. Neural networks can automatically interpret features from observable variables and can model complex nonlinear phenomena. Hybrid methodologies that combine neural networks with classical models (e.g., ARIMA) are also popular and have been used for dengue outbreak forecasting [32]. In recent times, long-short term memory (LSTM) networks [33] (a type of recurrent neural networks), and its derivatives [34], [35] have shown a considerable amount of success in predicting sequential data and has frequently outperformed other classical and machine learning methods [36]. These recurrent architectures have also performed well to predict disease outbreaks [37], [38], [39]. Hence, we consider these viable candidates to test the performance of the proposed feature enhancement framework in this paper.

### 2.3 Factors that affect dengue disease dynamics

Dengue virus is primarily spread by female mosquito vector species: (*Aedes aegypti* and *Aedes albopictus*). Hence, the outbreaks depend on the abundance of such vectors. The relationships between *Aedes* mosquitoes and environmental variables (i.e., temperature, rainfall) are already well characterized by many researchers. Environmental temperature affects the growth, host-seeking, blood-meal intakes of mosquitoes. *Aedes aegypti* cannot develop below 16°C or above 34°C [40]. Within that range, the development from larva to adult was found to be faster at higher temperatures (30°C) compared to lower temperatures (21°C) [41]. *Aedes albopictus* can develop in wider temperature ranges and can survive better in lower temperatures [42] compared to *Aedes aegypti*. The optimum flight temperature for *Aedes aegypti* females was found to be 21°C [43]. Studies have observed that a large diurnal temperature range decreases female fecundity [44]. Temperature fluctuations also affect extrinsic incubation periods (EIP) of dengue viruses. An experiment with DEN-2 strain found that EIP was 12 days at 30°C and reduced to 7 days for 32°C and 35°C [45]. Besides temperature, rainfall has a significant role in dengue outbreaks. Rainwater stuck in different places creates breeding spaces for *Aedes* mosquitoes. Previous works have studied the association of dengue transmission with rainfall [4, 46]. In this work, we use several variables, including observed and reanalyzed temperatures, precipitations, relative humidity, and surface-level pressure. These can directly or indirectly affect mosquito vector suitability and dengue outbreak dynamics.

### 2.4 Data collection and processing

#### 2.4.1 Data acquisition

To test the proposed framework for dengue outbreak predictions, we collect data for several regions of Brazil. The InfoDengue project [47] monitors outbreak data on over 700 municipalities of Brazil. Their server contains weekly dengue fever case counts with a surveillance period starting from 2010. Besides dengue incidence data, we collect weather observation data from NOAA (National Oceanic and Atmospheric Administration) ground weather station database and reanalysis data from the NCEP /NCAR Reanalysis 1 dataset published by the NOAA physical sciences laboratory (PSL) database (NCEP and NCAR stand for National Centers for Environmental Prediction and National Center for Atmospheric Research, respectively). We list all the variables in Table 1. We use the weekly case counts as labels and further process the remaining variables to use those as baseline features.

| Variable Name | Time Resolution | Space Resolution | Source |
|---------------|-----------------|------------------|--------|
| Reported dengue fever cases | Weekly | Municipality level | INFO Dengue |
| Observed Temperatures (min, max, and avg) | Daily | Ground station dependent | NCEI-NOAA |
| Observed Precipitation | Daily | Ground station dependent | NCEI-NOAA |
| Avg surface air temperature | Daily | 2.5° × 2.5° | NCEP/NCAR Reanalysis 1 |
| Avg surface relative humidity | Daily | 2.5° × 2.5° | NCEP/NCAR Reanalysis 1 |
| Avg surface pressure | Daily | 2.5° × 2.5° | NCEP/NCAR Reanalysis 1 |
| Avg precipitable water | Daily | 2.5° × 2.5° | NCEP/NCAR Reanalysis 1 |

#### 2.4.2 Re-sampling and feature engineering

The available raw data cannot be readily used in the machine learning methods. All the features and labels are matched and aligned in both spatial and temporal dimensions. We align data based on available labels (i.e., case data). For each municipality of Brazil (smallest spatial unit available), we search for the nearest ground weather station to collect weather data. We extract reanalysis data from NCEP/NCAR Reanalysis 1 data sets using each municipality’s centroid’s coordinates. Once the spatial granularity is taken care of, we fix the temporal dimension mismatches by converting all data to match incidence data resolution. As incidence data are available in weekly intervals and the remaining variables are available daily, this process involves context-aware down-sampling of those remaining variables (temperature, precipitation, humidity, etc.). During this process, we derive 4 additional features from observed weather data of ground stations: average diurnal temperature range of the week, minimum diurnal temperature range of the week, maximum diurnal temperature range of the week, and the number of rainy days in the week. We compute these from daily observed temperature and precipitation data. In total, we have 12 feature variables related to weather and environment: i) 4 observed variables: temperature (average, minimum, and maximum) and precipitation, ii) 4 derived variables based on the time interval (week): diurnal temperature range (average, minimum, and maximum) and the number of rainy days, iii) 4 reanalysis variables:
temperatures, relative humidity, pressure, precipitable water (all are averages and at earth surface level).

2.4.3 Data splitting and normalization

The complete set of data is a two-dimensional array \( X \) with features on one dimension and time on the other. The set of features consists of: i) the variables listed in section 2.4.2 ii) dengue incidence data of the target location, and iii) dengue incidence data of locations selected as predictors by the methods presented in this paper. We split \( X \) along its time dimension into three parts: training (\( X_{\text{train}} \)), validation (\( X_{\text{val}} \)), and test (\( X_{\text{test}} \)). The validation set is required during the training phase as we implement early stopping [48] as a regularization mechanism. We normalize all three sets of data before model training and evaluation. The normalization formula is the following,

\[
\begin{align*}
\mu_{\text{train}} &= \text{MEAN}(X_{\text{train}}) \\
\sigma_{\text{train}} &= \text{SD}(X_{\text{train}}) \\
X_{\text{train}} &= \frac{(X_{\text{train}} - \mu_{\text{train}})}{\sigma_{\text{train}}} \\
X_{\text{val}} &= \frac{(X_{\text{val}} - \mu_{\text{train}})}{\sigma_{\text{train}}} \\
X_{\text{test}} &= \frac{(X_{\text{test}} - \mu_{\text{train}})}{\sigma_{\text{train}}}
\end{align*}
\]

All three data sets (training, validation, and test) are normalized using the mean and the standard deviation computed from training data. The complete data set’s summary statistics are not used here to prevent the machine learning models from gaining statistical insights about validation and test data sets.

2.5 Sequence model specifications

Whether it is training or evaluation, the time series data are split into smaller batches and fed into the recurrent neural network models. From a macroscopic perspective, a sliding window moves over batches of data. Because of the sequential nature of dengue case data, the batches are fed according to the time order without randomization. The sliding window is configured with two integer parameters: input length (\( t_{\text{in}} \)), and output length (\( t_{\text{out}} \)). The window is depicted in Figure 1. A trainable model would take \( t_{\text{in}} \) time steps of feature data as input and predict \( t_{\text{out}} \) time steps of target/label data (e.g., dengue case counts) as outputs every iteration. In the configurations used in this paper, there is no temporal overlap between input and output sequences. In this configuration, the models make single shot projections (all the \( t_{\text{out}} \) data points are predicted at once every iteration).

Each batch (window) of data is \( (t_{\text{in}} + t_{\text{out}}) \) steps long in the time dimension. A total of 32 batches are stacked together for model training and evaluation. One batch differs from another by a single time-step (hence, there are temporal overlaps between batches). Each batch is further split in time and data (variable) dimensions to separate inputs (\( t_{\text{in}} \) of features) and outputs (\( t_{\text{out}} \) of labels).

This work focuses on performance gains obtainable using recurrent neural network (RNN) models due to their proven track record in predicting time series data [36]. A recurrent unit’s temporal behavior is illustrated in the lower part of Figure 1. We can see that information is passed through time (also called cell state), enabling the model to predict values based on insights gained from past inputs. We use two popular recurrent neural network cell types: LSTM (long-short term memory) [33] with forget gates [49] and GRU (gated recurrent unit) [50]. We also test with a simple linear neural network model as a trivial baseline. Using the TensorFlow [51] package, we configure the models as described below to produce the results:

- **Linear**: The model consists of a single layer of artificial neurons (Dense units in TensorFlow) without any nonlinear activation functions. The size of the layer (number of neural units) is equal to the output prediction steps, \( t_{\text{out}} \).
- **LSTM**: The model consists of an input layer of 32 long short-term memory (LSTM) units. The output layer consists of a layer similar to the Linear model (described above).
- **GRU**: The model consists of an input layer of 32 gated recurrent units (GRU). The output layer consists of a layer similar to the Linear model (described above).

We initialize the weight metrics of the models as zeros in the beginning. Only the recurrent models (LSTM and GRU) can train and predict based on entire input sequences. The Linear model predicts based on the last input time step. While training, we use the mean squared error (MSE) as the loss function to optimize the model using Adam optimizer [52]. For predictions, we use the mean absolute error (MAE) metric to evaluate model performance. We regularize our training process with the early stopping [48] mechanism, which monitors loss within validation data and stops training if performance does not improve. Based on our tests on different data sets, we configure the training to run for 120 iterations (epochs).
3 Methods

To describe the methodology, first, we define our spatial units. We designate the term infection center (IC) to indicate a spatial building block of the model. An IC is a point in the space (regional map) where observed or estimated incidence data on disease outbreaks are available. The spatial granularity of an IC is not fixed for the model. It can be a country, a state, a city, a suburb, or an administrative unit with some resolution in space based on available disease incidence data. The basic structure of our proposed framework is shown in Figure 2. The circles inside the shaded region are the infection centers. One of the infection centers, marked as IC, indicates the target infection center where we intend to make predictions of a designated label (e.g., dengue cases). The map’s remaining infection centers are marked as peripheral infection centers (PIC). These are locations where similar observations on the designated label of the target IC are available. The temporal resolution of the IC and the PICs are aligned before performing any comparative analysis of the data. Some locations may have daily, weekly, or monthly observations. Some data may need re-sampling in the time domain before they can be compared (e.g., convert daily weather data to weekly values).

**Fig. 2. A framework depicting the method to expand trainable and testable data set on dengue incidence. The shaded region enclosed by dashed borders on the left depicts a map of the region of interest. The circles inside the map indicate multiple infection centers (IC) across the region. The target infection center is marked as IC; while the kth peripheral infection center is PIC. Using the proposed windowed cross-correlation method (section 3.2), a phased cross-correlation matrix is computed (I), which is eventually reduced to a correlation weight, \(\gamma_C(k)\) (II). We also compute a prevalence weight, \(\gamma_P(k)\), using cumulative case data (III) and a geographic distance weight, \(\gamma_D(k)\), using location data (IV). The three weight metrics are combined to compute (V) the predictor metric of PIC\(_k\), \(\Gamma(k)\). The PICs are ranked using these \(\Gamma\) values, and their incidence data are selected accordingly to be stacked together with the IC feature set (VI).**

Assuming that there are \(N\) peripheral infection centers (PIC) on the map. Once we match the spatial and the temporal dimensions of the label data (e.g., weekly dengue cases in a city), we use a windowed-time shifted cross-correlation analysis on each \(IC - PIC_k\) pair (for all \(k \in N\)) and compute a correlation weight, \(\gamma_C(k)\). We also consider the cumulative cases of each PIC and compute a prevalence weight, \(\gamma_P(k)\). Finally, the geodesic distance of each \(IC – PIC_k\) pair is taken into account in the form of a distance metric, \(\gamma_D(k)\). All three metrics are normalized and lie within the range \([0, 1]\) for the selected region. These are combined as following to compute a predictor strength metric for each PIC\(_k\).

\[
\Gamma(k) = \gamma_C(k)[\gamma_P(k) + \gamma_D(k)], \forall k \in N \tag{2}
\]

3.1 Windowing incidence data

We compare the time series of disease incidence data (e.g., weekly cases per 100k people) to find correlations. The comparisons are made for all \(IC – PIC_k\) pairs with available data for a given region. The key intuition behind this approach is, if a PIC in the region has an infectious outbreak at some point in time, \(t = t_0\), this may initiate or affect the course of an outbreak for the target IC at \(t \geq t_0\). A leading PIC outbreak may not always imply causal influence depending on the geographic location and population behavior. Despite that, a PIC having an outbreak will influence adjacent ICs, as it acts as an infection source. This also assumes that a strict isolation measure is not in place and the control measures are not 100% effective due to the vector-borne nature of the infection. It is also important to note that, despite seasonal patterns, outbreaks can occur irregularly. A PIC may lead the target IC in one season and lag in other seasons due to complex interactions of multiple viral strains. Hence, we divide the time series into smaller time windows. We propose two methods for windowing incidence data: i) fixed-length window allocation and ii) variable-length window detection. Both of these methods are depicted in Figure 3.

A straightforward approach is to divide the entire time series into a fixed number of intervals, \(M_f\). If the time series is \(T\) units (day, week, or month) long in total, then fixed window, \(w_m^f\) (where \(m \in [0, M_f - 1]\)) will have \(T/M_f\) units of data. While this is the simplest way to divide the series for correlation analysis, it may not be the most efficient. Setting the appropriate value of \(M_f\) remains an open problem, although we apply some intuitions from the seasonality patterns of dengue outbreaks in our case. The fixed windows might not be appropriately placed to contain the time series curve’s meaningful dynamics, and some windows may even cover regions without an outbreak.

A second approach is to detect windows based on the time series itself. To do this, we normalize the incidence rate of the target IC to be ranged between \([0, 1]\). A typical outbreak curve has irregularities in its shape that make the analysis cumbersome. We use a Savitzky-Golay filter to smooth out the irregularities while preserving the shape of the outbreaks. Our window detection method has two parameters: the incidence threshold \((i_{MIN})\) and the minimum window size \((\Delta_{MIN})\). An window is detected between two time points, \(t_{START}\) and \(t_{END}\) (where, \(0 \leq t_{START} \leq t_{END} \leq T\)), if the normalized incidence rate, \(i_N(t) > i_{MIN}\) for all \(t_{START} \leq t \leq t_{END}\) and \(t_{END} - t_{START} \geq \Delta_{MIN}\). A detected window, \(w_m^d\) will have a length (greater than \(\Delta_{MIN}\)) that depends on the time series curve characteristics. The number of detected windows, \(M_d\), will also vary for the same reason. Figure 3 shows both methods in action using time series data for...
Brazils municipality between 2010-2015. For the assigned value of \( M_f = 5 \), we get equally sized windows, each of which is 1 year in length. With our detection method, we find 4 windows \( (M_d = 4) \) that indicate 4 separate outbreaks \( (\Delta_{MIN} = 10, i_{MIN} = 0.05) \).

3.2 Time-shifted cross correlation

Let \( i_0(t) \) and \( i_k(t) \) be the disease incidence (new cases at time step t) of the target IC and the \( k \)th PIC respectively. We perform bivariate computations of time shifted Pearson’s correlation coefficients \([54]\) using windowed \( (w_m) \) incidence data of the target IC \( (i_0^{w_m}(t)) \) and the \( k \)th PIC \( (i_k^{w_m}(t)) \). The formula used to compute the coefficients is shown in Equation 3. This measure is also known as time lagged (or phased) cross-correlation \([55]\) in statistics and signal processing.

\[
R_{k}^{w_m}(\theta) = r(i_0^{w_m}(t), i_k^{w_m}(t-\theta))
\]  

\( R_{k}^{w_m}(\theta) \) is the correlation coefficient computed for subsets of the time series \( i_0(t) \) and \( i_k(t) \) selected by the time window \( w_m \) when one series is shifted by an amount \( \theta \) with respect to another. The Pearson’s correlation coefficient function is indicated by \( r() \) in Equation 3.

For the two time series curves shown in Figure 4, the time-lagged correlation matrix (obtained by computing \( R_{k}^{w_m}(\theta) \) \( \forall m \in [0, M - 1] \) and \( \theta \in [-8, 8] \)) is plotted as a heatmap in Figure 5. We use the location depicted in Figure 3 as the target IC \( (i_0(t)) \) and another location from the same state (Espirito Santo) of Brazil as the \( k \)th PIC \( (i_k(t)) \). For this demonstration, the time series curves were split using fixed length windows \( (M_f = M = 5) \). The heatmap depicted in Figure 5 can be visualized by comparing with Figure 4. As expected, the two curves are almost in phase in \( w_0 \), negatively correlated in \( w_2 \), IC leads in \( w_1 \), and PIC leads in \( w_3 \) and \( w_4 \).

Heatsmaps like Figure 5 are computed for all \( PIC_k \) with \( k \in [1, N] \). To determine if a \( PIC_k \) is leading in a window \( (w_m) \), we find the location \( (\theta) \) of the peak correlation as shown in Equation 4.

\[
\theta_{k}^{w_m} = \arg\max_{\theta} R_{k}^{w_m}(\theta)
\]  

We define the correlation strength \( S_{k}^{w_m} \) to be the mean correlation measure computed around the peak \( (\theta_{k}^{w_m} + \theta) \), extending by the amount \( \theta_E \) in both directions (Equation 5).

\[
S_{k}^{w_m} = \frac{1}{\sum_{\theta=\theta_E}^{\theta_E}} \sum_{\theta=-\theta_E}^{\theta_E} R_{k}^{w_m}(\theta_{k}^{w_m} + \theta)
\]  

Equation 5 has an additional condition on the values \( \theta \) such that \( R_{k}^{w_m}(\theta_{k}^{w_m} + \theta) \) exists for the given parameters. A \( PIC_k \) leads the IC if the peak of correlation lies on the right half of the heatmap shown in Figure 5, which translates to \( \theta_{k}^{w_m} > 0 \). We only consider if a \( PIC_k \) is at least in phase with the IC and discard the cases where any \( PIC_k \) lags the IC. This is how we compute the predictor probability.
matrix $P$ with dimensions $M \times N$. The individual predictor probabilities ($\forall m \in [0, M-1], \forall k \in [1, N]$) are computed as shown in Equation 6:

$$P_{m,k} = \begin{cases} S_k^{w_m}, & \text{if } \theta_k^{w_m} \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

(6)

The predictor probabilities are averaged across all windows (Equation 2) to compute overall predictive abilities of all $PIC_k$. For a particular region ($k \in [1, N$]), the predictive ability metrics are normalized between [0,1], the final measure, $\gamma_C(k)$, is defined as the correlation weight of $PIC_k$ as shown in Equation 8:

$$\gamma_C(k) = \frac{1}{M} \sum_{m=0}^{M-1} P_{m,k}$$

(7)

$$\gamma_C(k) = \frac{\gamma_C^{\hat{k}}(k) - \min_k \gamma_C^{\hat{k}}(k)}{\max_k \gamma_C^{\hat{k}}(k) - \min_k \gamma_C^{\hat{k}}(k)}$$

(8)

### 3.3 Distance and prevalence metrics

With increasing distance, the impact of a $PIC$ on the target $IC$ is likely to be reduced due to decreased travel between the locations, increasing differences in environmental conditions (e.g., temperature, rainfall, vegetation), etc. It is intuitive to use a metric proportional to the inverse distance for strengthening the predictive ability measures of $PIC$s. Let, $d_k$ be the geodesic distance (shortest path on the surface of the earth, assuming earth to be an ellipsoid) between the target $IC$ and $PIC_k$. The normalized [0,1] distance of a $PIC_k$ in the region is,

$$\hat{d}_k = \frac{d_k - \min_k d_k}{\max_k d_k - \min_k d_k}$$

(9)

We want the metric to be inversely proportional to the distance. Hence, the distance metric of $PIC_k$ is defined as,

$$\gamma_D(k) = 1 - \hat{d}_k$$

(10)

The outbreak history of a location is an important criterion that indicates the viral pathogen and endemic scenarios’ persistence. For a $PIC_k$, we compute the prevalence $I_k$ by taking a sum of the incidence data $i_k(t)$ for the entire timeline ($\forall t \in [0, T$])

$$I_k = \sum_{t=0}^{T} i_k(t)$$

(11)

The prevalence metric is normalized [0,1] across the region.

$$\gamma_P(k) = \frac{I_k - \min_k I_k}{\max_k I_k - \min_k I_k}$$

(12)

### 4 RESULTS

We present here the results in several stages. The outcomes of the feature analysis are presented first. This is followed by an analysis of prediction performance using the proposed methods. The results are generated using municipality-wise weekly dengue case data between 2010-2019 from Brazil’s Espirito Santo state. We obtained data for 78 municipalities of Espirito Santo and ranked them based on the total number of cases recorded for the entire time period. The top 5 municipalities based on prevalence are listed in Table 2. The location IDs shown in the table are the IBGE (Instituto Brasileiro de Geografia e Estatística) codes for Brazil [57].

| Loc. ID | Name                | Cases  | Cases per 100k |
|---------|---------------------|--------|----------------|
| 3205309 | Vitoria             | 71,348 | 19,901.72      |
| 3205002 | Serra               | 58,424 | 11,031.10      |
| 3201209 | Cachoeiro de Itapemirim | 45,319 | 21,520.12      |
| 3205200 | Vila Velha          | 36,743 | 7,329.18       |
| 3201308 | Cariacica           | 27,103 | 7,059.60       |

### 4.1 Feature selection and analysis

The $PIC$s are sorted and ranked for each $IC$, based on the predictor strength metric, $\Gamma$ (Equation 2), which is computed from the three individual metrics: correlation weight ($\gamma_C$), prevalence weight ($\gamma_P$), and distance weight ($\gamma_D$). We choose the municipality of Vitória in Espirito Santo, Brazil, as the target $IC$, which had the highest total number of cases in the state during 2010-2019, to generate the results. We compute the correlation weight using 20 fixed-length windows ($Mf = M = 20$) for the time range 2010-2019, making each window approximately 26 weeks (6 months) long. The top 5 ranked $PIC$s based on $\Gamma$ are listed in Table 3 with the corresponding weights. While our method prioritizes the correlation weight more than others, $PIC$s with relatively lower correlation weight can still be favored because of the following factors: i) having a significant number of cases or ii) being in proximity of the target $IC$. This is evident in Table 3 as 3201209 ($PIC$ #1) is chosen over 3205200 ($PIC$ #2) and 3205101 ($PIC$ #4) is chosen over 3200607 ($PIC$ #5). Note, the numbers (#) used in ‘PIC #’ indicate rank. This should not be confused with arbitrary indices (k) used to compute metrics ($PIC_k$). This effect is also illustrated in Figure 6, which shows the ranked $PIC$s listed in Table 3. Although $PIC$ #1 lies farthest from the $IC$ among the five (lowest $\gamma_C$), it is ranked at the top due to significantly higher values in the other two factors ($\gamma_C$ and $\gamma_P$).

A time series plot in Figure 7 shows that the top $PIC$s are mostly correlated with the $IC$, Vitória. Among the $PIC$s displayed here, PIC #4 (3205101) shows the weakest correlation with the $IC$. It will be clear in the upcoming results, the proximity and the high incidence fraction of this location help with prediction performance. The variability of the incidence curves prevents our method from classifying a single $PIC$ as the optimal predictor for all time ranges. However, the combination of top-ranked $PIC$s will improve prediction accuracy.
TABLE 3
Top 5 predictor PICs for Vitória. The weights based on our defined predictability metrics (correlation, prevalence, and distance) are shown in columns 3-5. The combined weights (\( \Gamma \)) are shown in the last column.

| Rank # | Loc. ID     | Corr. | \( \gamma_C \) | Prev. | \( \gamma_P \) | Dist. | \( \gamma_D \) | \( \Gamma \) |
|--------|-------------|-------|----------------|-------|----------------|-------|----------------|---------|
| 1      | 3201209     | 0.912 | 0.912          | 0.812 | 0.812          | 0.614 | 0.614          | 1.301   |
| 2      | 3205200     | 1.000 | 1.000          | 0.270 | 0.270          | 1.000 | 1.000          | 1.270   |
| 3      | 3201308     | 0.956 | 0.956          | 0.260 | 0.260          | 0.996 | 0.996          | 1.202   |
| 4      | 3205101     | 0.621 | 0.621          | 0.770 | 0.770          | 0.937 | 0.937          | 1.060   |
| 5      | 3200607     | 0.772 | 0.772          | 0.524 | 0.524          | 0.809 | 0.809          | 1.029   |

Fig. 6. A geospatial map showing an incidence center (IC) and 5 ranked peripheral incidence centers (PIC). The circles’ radii are proportional to the total number of cases reported during 2010-2019 [47]. The shades of the fill color are generated from a color gradient (green-yellow-red) which is proportional to the fraction of cases with respect to the local population of each location (IC or PIC) during 2010-2019 [47]. The greenish shades indicate smaller infected fractions, while the reddish shades indicate larger fractions. Map generated using Folium [58] with OpenStreetMap [59]. Basemap tiles provided by CartoDB [60].

4.2 Prediction performance
After selecting features with the proposed methods, we train and evaluate the prediction performances using the three models (Linear, LSTM, and GRU) described in section 2.5. The time series data are split into 3 distinct sets with ratios of 50:30:20 for model training, evaluation, and testing. A sliding window with input length \( t_{in} \) of 8 and output length \( t_{out} \) 4 is used for a single shot prediction of the next 4 weeks using data of the past 8 weeks every step.

4.2.1 Individual IC prediction
For the IC of Vitória, the PICs are added gradually according to their computed ranks (section 4.1), and the models’ prediction performances are evaluated. The mean absolute error (MAE) values on the normalized test data are plotted in Figure 8 for varying number of additional features, \( N_{PIC} \). For both LSTM and GRU models, the addition of the first two PICs deteriorates the model performance. However, the subsequent additions keep improving the outcomes. The plots quickly reach their minima, after which errors increase. The first few additions increase error due to high variability in the incidence data, as evident in Figure 7. Further additions of PIC create averaging effects on the predictor data set and cause performance improvements. The GRU model eventually reaches a minimum MAE value of 0.128, which is lower than the best optimal LSTM prediction (0.1415) by about 9.54%. The Linear model reaches an optimum MAE value of 0.3384, which is nowhere close to the recurrent models. After reaching the minima, all three error curves rise again. This increase in MAE with larger feature sets \( (N_{PIC}) \) can be attributed to the curse of dimensionality [21], [22]. LSTM and GRU models perform optimally with 4 and 6 additional PICs, respectively. Predicting based on present input and historical context (internal states of LSTM and GRU) of data certainly puts recurrent models ahead in performance, which is evident even without a PIC \( (N_{PIC} = 0 \) in Figure 8) in the feature set. However, the linear model significantly benefits from feature addition as case data from PICs strengthen inductive bias.

Fig. 7. Time series plots of weekly dengue cases per 100,000 people for the IC and top 5 ranked PICs (Table 3). The plots depict cases only between 2010-2015 for improved clarity, but the metrics were computed based on the entire series (2010-2019).

Fig. 8. Prediction performances of the Linear, LSTM, and GRU models in predicting normalized test data for varying number of features \( (N_{PIC}) \). PIC data are added to the feature set based on ranks dictated by computed predictor strengths (Table 3), after which models are trained and evaluated over the test data to compute the mean absolute error (MAE) metrics (lower is better). After determining the optimum number of features \( (N_{PIC}) \) to be added to the predictor data set for an IC, recurrent models are trained with the extended feature set and are used to predict dengue cases for the test data subset of the time series. Figure 9 compares the predictions with actual incidence data for both LSTM and GRU models using Vitória as the IC and choosing the optimum number of PICs for the two models \( (N_{PIC} = 4 \) for LSTM and 6 for
GRU).

Fig. 9. Predicted weekly dengue cases per 100,000 people using test data with the recurrent models (LSTM and GRU). These results were obtained for the IC; Vitória, with 4 and 6 additional PIC data joined with LSTM and GRU’s feature set, respectively.

4.2.2 Effect of window selection methods
The impacts of various correlation window configurations are analyzed under the two proposed windowing methods: i) fixed-length window assignment and ii) variable-length window detection. For the first method (fixed), we vary the number of windows ($M^f$) between 5, 10, 20, and 40. For the second method (detection), we vary the minimum window size ($\Delta_{MIN}$) between 5, 10, 20, and 30 weeks. In both methods, we are consequently varying the number of windows, window lengths, and where the windows are located. In total, we run tests under 8 different scenarios and compare the prediction performances using the 3 models (Linear, LSTM, and GRU). Instead of working with a single IC, we run the tests over multiple locations and report the average performance metrics (e.g., mean of MAE). We take the top 20 ICs based on total cases per 100,000 people and average the performance metrics across ICs. Among the 20 ICs, there were 4 ICs where none of the models could predict with reasonable accuracy (MAE $< 1$) with or without additional features. In those locations, either the data were too limited or the outbreaks were too random for our models to generalize beyond training data. We filter out these 4 locations and take the remaining 16 locations to evaluate our methods.

For a model, predicting on a given IC, the optimal MAE is defined as the minimum mean absolute error (MAE) obtained by varying the number of additional features ($N_{PIC}$) from the PIC ranked list produced by a given windowing method (i.e., minima of the curves in Figure 8 are the optimal MAEs of three models). The average optimal MAE (across 16 ICs) are plotted in Figure 10 for all 8 windowing schemes. The recurrent models perform significantly better than the Linear model: on average, LSTM and GRU outperform the Linear model by 60% and 61%, respectively. LSTM and GRU outperform the best-performing linear model by 41% and 43%, respectively. We expect this kind of advantage, given the importance of considering historical data in predicting future dengue cases. The performances across different windowing schemes with LSTM and GRU models are comparable (both models have an approximate standard deviation of 0.02 around their means of 0.3569 and 0.3435, respectively), with a small trend of increasing mean error values with windowing method configurations. In the case of linear models, there are some stark contrasts when using window detection methods. Using the window detection method with a large $\Delta_{MIN}$ value increases performance sharply. For example, using a $\Delta_{MIN}$ of 30 weeks shows 41.9% improvement over fixed window schemes’ average performance. This indicates that features selected with a small number of large correlation windows detected based on outbreak locations in the time series are more effective than the remaining schemes that use a relatively large number of smaller windows. A similar trend is also visible with fixed window methods, although it is not statistically conclusive. The lower values of $M^f$, which translates to having a lower number of large windows, show slightly better performance over higher values of $M^f$.

To understand how beneficial our proposed methods are over the baseline (without feature enhancement, $N_{PIC} = 0$), we analyze the improvements in prediction accuracy. The term, improvement in accuracy, is defined as the relative decrease in MAE (i.e., increase in prediction performance) with optimum PIC selection (Figure 10) compared to MAE without additional features ($N_{PIC} = 0$). We compute the accuracy improvements across all 16 ICs and plot the mean improvement in accuracy as percentages in Figure 11. The linear model demonstrates average improvements ranging from 18.97% to 27.13% depending on the window selection schemes. The LSTM model improvements range from 22.13% to 33.6% and the GRU model range from 10.79% to 31.92% depending on the window selection schemes. We should be careful in interpreting these values; greater improvements do not translate to better optimal performance. These values are relative to their baselines ($N_{PIC} = 0$). With some incidence centers (IC), a model can already predict with higher accuracy than other incidence centers. Our methods help minimize those gaps and still provide some improvements over the baselines (ranging from 10.79% to 33.6% depending on the model and the scheme). Figure 10 demonstrates that, in general, GRU and LSTM both perform...
well when we deal with average values. For individual ICs, however, a conclusive determination of the best performing model can be done (either GRU or LSTM).

\[
\begin{align*}
\text{Fixed, } M^f = 5 & \quad \text{Detected, } \Delta_{MIN} = 5 \\
\text{Fixed, } M^f = 10 & \quad \text{Detected, } \Delta_{MIN} = 10 \\
\text{Fixed, } M^f = 20 & \quad \text{Detected, } \Delta_{MIN} = 20 \\
\text{Fixed, } M^f = 40 & \quad \text{Detected, } \Delta_{MIN} = 30
\end{align*}
\]

In other words, windowed cross-correlation on a few important outbreak regions performs better than comparing over the entire time series. As the target data is aggregated from all PICs in the region, the optimally trained Linear models sometimes perform better than recurrent models.

One key takeaway is that cases for the entire region can be predicted with improved accuracy while using a small subset (PICs) of data as features. For the results shown in Figure 12, the optimal MAE values can be reached for \(N_{PIC} = 4\) values between 2 and 5. This aggregation provides a fast way to predict cases, with a small fraction of regional outbreak data. While the accuracy achieved at the ecoregion level is not on par with the accuracy achieved for single ICs, this prediction is useful to generate risk maps.

4.2.3 Performance on aggregated data

It is sometimes reasonable to aggregate data to larger scales based on geographic adjacency and environmental similarities (e.g., weather). From a macroscopic point of view, predicting for an ecoregion may be more meaningful for policymakers to interpret the outcomes. According to Omernik (2004), ecoregions are defined as areas within which there is a spatial coincidence in characteristics of geographical phenomena (e.g., geology, physiography, vegetation, land use, climate, hydrology, terrestrial and aquatic fauna, etc.) associated with differences in the quality, health, and integrity of ecosystems \([61]\). We use the terrestrial ecoregions defined by The Nature Conservancy (TNC) in this work \([62]\). The following analysis is performed for Brazilian locations with available data in the Bahia Coastal Forest ecoregion.

Comparing the prediction performance on aggregated data shows that the advantages of recurrent models compared to the Linear model (which we had for individual ICs) are diminished. The optimal performances across prediction models become more comparable, as shown in Figure 12. The mean optimal MAE values obtained for 4 fixed window schemes are 0.38, 0.40, and 0.36 when using Linear, LSTM, and GRU models, respectively. The mean optimal MAEs for 4 window detection schemes are 0.31, 0.35, and 0.33 using the same three models. We observe a distinct advantage of the window detection method for selecting PICs. On average, the window detection methods improve over their baselines (\(N_{PIC} = 0\)) by 16.50%, 12.68%, and 10.07% for Linear, LSTM, and GRU models, respectively. Variable window allocation based on outbreak window detection outperforms fixed window allocation methods.

\[
\begin{align*}
\text{Fixed, } M^f = 5 & \quad \text{Detected, } \Delta_{MIN} = 5 \\
\text{Fixed, } M^f = 10 & \quad \text{Detected, } \Delta_{MIN} = 10 \\
\text{Fixed, } M^f = 20 & \quad \text{Detected, } \Delta_{MIN} = 20 \\
\text{Fixed, } M^f = 40 & \quad \text{Detected, } \Delta_{MIN} = 30
\end{align*}
\]

We construct a risk map for the Bahia Coastal Forest ecoregion using the predicted cases and weights of the top-ranked PICs in Figure 13. The top 40 PICs are included in the map, and the PICs that contribute more towards the weight (\(f\)) are shown as high-risk regions (e.g., red). The aggregated prediction was done using the optimal Linear model with \(\Delta_{MIN} = 30\) and \(N_{PIC} = 4\).

5 Conclusion

In this work, we develop a method to select relevant incidence data from peripheral locations as features to improve the prediction of dengue fever outbreaks. In order to rank features, we use windowed cross-correlation analysis on dengue case data. We propose two methods for allocating correlation windows (position and size) over the time series to compute correlation weights. For a target location (IC), peripheral locations (PIC) are ranked based on a combination of correlation, distance, prevalence metrics. The predictive models benefit from the ranked feature sets, as these reach model and location-specific optimal performances with a relatively small subset of features. We tested
three predictive models using dengue case data from Brazil, showing different levels of accuracy gains.

On average, the proposed feature enhancement methods improve prediction performance by 10.79% to 33.6% over the baseline feature set for the locations we tested, depending on the prediction model and the window allocation scheme. For the location with the highest total cases (2010-2019) in the Espírito Santo region of Brazil, we could get MAE values as low as 0.13 (normalized case data) using the GRU model with data from just 6 locations added to the feature set. In a test across multiple locations, both RNN models (LSTM and GRU) performed with comparable accuracy (average MAE ranging from 0.3435 to 0.3569) when using an optimal number of additional features. The Linear model also benefited (18.97% to 27.13% improvement when using an optimal number of additional features. The proposed method can be generalized and used for projecting any infectious outbreak where temporal data at a reasonable spatial resolution are available.

**ACKNOWLEDGMENT**

This research has been supported by the Department of the Army, U.S. Army Contracting Command, Aberdeen Proving Ground, Natick Contracting Division, Fort Detrick (DWFP Grant W911QY-19-1-0004).

**REFERENCES**

[1] R. E. Baker, J.-M. Peña, J. Jayamohan, and A. Jerusalem, “Mechanistic models versus machine learning, a fight worth fighting for the biological community?” *Biological Letters*, vol. 14, no. 5, p. 20170660, 2018.

[2] S. Bhatt, P. W. Gething, O. J. Brady, J. P. Messina, A. W. Farlow, C. L. Moyes, J. M. Drake, J. S. Brownstein, A. G. Hoen, O. Sankoh et al., “The global distribution and burden of dengue,” *Nature*, vol. 496, no. 7446, pp. 504–507, 2013.

[3] G. A. Roth, D. Abate, K. H. Abate, S. M. Abay, C. Abaffy, N. Abbasi, H. Abbastabar, F. Abd-Allah, J. Abdela, A. Abdelalim et al., “Global, regional, and national age-sex-specific mortality for 282 causes of death in 195 countries and territories, 1980–2017: a systematic analysis for the global burden of disease study 2017,” *The Lancet*, vol. 392, no. 10159, pp. 1736–1788, 2018.

[4] R. Li, L. Xu, O. N. Bjørnstad, K. Liu, T. Song, A. Chen, B. Xu, Q. Liu, and N. C. Stenseth, “Climate-driven variation in mosquito density predicts the spatiotemporal dynamics of dengue,” *Proceedings of the National Academy of Sciences*, vol. 116, no. 9, pp. 3624–3629, 2019.

[5] D. O. Fuller, A. Troyo, and J. C. Beier, “El Nino Southern Oscillation and vegetation dynamics as predictors of dengue fever cases in Costa Rica,” *Environmental Research Letters*, vol. 4, no. 1, p. 014011, 2009.

[6] P.-C. Wu, J.-G. Lay, H.-R. Guo, C.-Y. Lin, S.-C. Lung, and H.-J. Su, “Higher temperature and urbanization affect the spatial patterns of dengue fever transmission in subtropical Taiwan,” *Science of the Total Environment*, vol. 407, no. 7, pp. 2224–2233, 2009.
algorithm in traffic flow forecasting,” Neural Computing and Applications, vol. 21, no. 3, pp. 583–593, 2012.

[28] T. W. Scott, E. Chow, D. Strickman, P. Kittayapong, R. A. Wirtz, L. H. Lorenz, and J. D. Edman, “Blood-feeding patterns of Aedes aegypti (Diptera: Culicidae) collected in a rural Thai village,” Journal of Medical Entomology, vol. 5, pp. 922–927, 1983.

[29] S. Shrivastava, D. Tiraki, A. Diwan, S. K. Lalwani, M. Modak, A. C. Mishra, and V. A. Arankalle, “Co-circulation of the all four dengue virus serotypes and detection of a novel clade of DENV-4 (genotype I) virus in Pune, India during 2016 season,” PLOS ONE, vol. 13, no. 2, p. e0192672, 2018.

[30] J. H. Huber, M. L. Childs, J. M. Caldwell, and E. A. Mordecai, “Seasonal temperature variation influences climate suitability for dengue, chikungunya, and Zika transmission,” PLOS neglected tropical diseases, vol. 12, no. 5, p. e006451, 2018.

[31] A. Wesołowski, T. Qureshi, M. F. Boni, P. R. study, M. A. Johansson, S. B. Rashed, K. Enge-Monsen, and C. O. Buckee, “Impact of human mobility on the emergence of dengue epidemics in Pakistan,” Proceedings of the National Academy of Sciences, vol. 112, no. 38, pp. 11887–11892, 2015.

[32] T. Chakraborty, S. Chattopadhyay, and I. Ghosh, “ Forecasting dengue epidemics using a hybrid methodology,” Physica A: Statistical Mechanics and its Applications, vol. 527, p. 121266, 2019.

[33] J. A. S. da Silva, J. K. L. Almeida, and L. A. L. Pessoa, “Time-series extreme event forecasting with neural networks at Uber,” in International Conference on Machine Learning, vol. 34, 2017, pp. 1–5.

[34] Y. Hua, Z. Zhao, R. Li, X. Chen, Z. Liu, and H. Zhang, “ Deep learning with long short-term memory for time series prediction,” IEEE Communications Magazine, vol. 57, no. 6, pp. 114–119, 2019.

[35] L. Zhu and N. Latpev, “ Deep and confident prediction for time series at Uber,” in 2017 IEEE International Conference on Data Mining Workshops (ICDMW). IEEE, 2017, pp. 103–110.

[36] J. Gu, L. Liang, H. Song, Y. Kong, R. Ma, Y. Hou, J. Zhao, J. Liu, N. He, and Y. Zhang, “A method for hand-foot-mouth disease prediction using GeoDetector and LSTM model in Guangxi, China,” Scientific Reports, vol. 9, no. 1, pp. 1–10, 2019.

[37] F. Shabid, A. Zameer, and M. Muneeb, “Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM,” Chaos, Solitions & Fractals, p. 110212, 2020.

[38] J. Xu, K. Xu, Z. Li, F. Meng, T. Tu, L. Xu, and Q. Liu, “ Forecast of dengue cases in 20 chinese cities based on the deep learning method,” International journal of environmental research and public health, vol. 17, no. 2, p. 453, 2020.

[39] S. R. Christophers, Aedes aegypti: the yellow fever mosquito. CUP Archive, 1960.

[40] J. Courret, E. Dotson, and M. Q. Benedict, “Temperature, larval diet, and density effects on development rate and survival of Aedes aegypti (Diptera: Culicidae),” PLOS ONE, vol. 9, no. 2, p. e87468, 2014.

[41] H. Delatte, G. Gimonneau, A. Triboire, and D. Fontenille, “Influence of temperature on immature development, survival, longevity, fecundity, and gonotrophic cycles of Aedes albopictus, vector of chikungunya and dengue in the Indian Ocean,” Journal of medical entomology, vol. 46, no. 1, pp. 33–41, 2009.

[42] W. A. Rowley and C. L. Graham, “The effect of temperature and relative humidity on the flight performance of female Aedes aegypti,” Journal of Insect Physiology, vol. 14, no. 9, pp. 1251–1257, 1968.

[43] L. B. Carrington, S. N. Feist, N. H. Willits, L. Lambrechts, and T. W. Scott, “Large diurnal temperature fluctuations negatively influence Aedes aegypti (Diptera: Culicidae) life-history traits,” Journal of medical entomology, vol. 50, no. 1, pp. 43–51, 2013.

[44] D. M. Watts, D. S. Burke, B. A. Harrison, R. E. Whitmire, and A. Nisalak, “Effect of temperature on the vector efficiency of Aedes aegypti for dengue 2 virus,” The American journal of tropical medicine and hygiene, vol. 36, no. 1, pp. 143–152, 1987.

[45] M. A. Johansson, F. Dominici, and G. E. Glass, “Local and global effects of climate on dengue transmission in Puerto Rico,” PLOS Negl Trop Dis, vol. 3, no. 2, p. e382, 2009.

[46] I. Dengue, “Info dengue surveillance project,” 2021. [Online]. Available [https://info.dengue.mat.br/]

[47] L. Prechelt, “Early stopping—but when ?” in Neural Networks: Tricks of the trade. Springer, 1998, pp. 55–69.
[49] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to Forget: Continual Prediction with LSTM,” *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[50] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.

[51] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, “Tensorflow: A system for large-scale machine learning,” in *12th USENIX symposium on operating systems design and implementation (OSDI ’16)*, 2016, pp. 265–283.

[52] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[53] A. Savitzky and M. J. Golay, “Smoothing and differentiation of data by simplified least squares procedures.” *Analytical chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[54] J. Benesty, J. Chen, Y. Huang, and I. Cohen, “Pearson correlation coefficient,” in *Noise reduction in speech processing*. Springer, 2009, pp. 1–4.

[55] “Cross correlation,” 2021. [Online]. Available: https://en.wikipedia.org/wiki/Cross-correlation

[56] C. F. Karney, “Algorithms for geodesics,” *Journal of Geodesy*, vol. 87, no. 1, pp. 43–55, 2013.

[57] “Instituto Brasileiro de Geografia e Estatística,” 2021. [Online]. Available: https://www.ibge.gov.br/

[58] Python-Visualization, “Folium,” 2021. [Online]. Available: https://python-visualization.github.io/folium/

[59] O. Foundation, “OpenStreetMap,” 2021. [Online]. Available: https://www.openstreetmap.org

[60] C. Inc., “CartoDB Positron,” 2021. [Online]. Available: https://carto.com/

[61] J. M. Omernik, “Perspectives on the nature and definition of ecological regions,” *Environmental Management*, vol. 34, no. 1, pp. S27–S38, 2004.

[62] D. M. Olson and E. Dinerstein, “The global 200: Priority ecoregions for global conservation,” *Annals of the Missouri Botanical garden*, pp. 199–224, 2002.