Cluster-based LSTM models to improve Dengue cases forecast

Juan V. Bogado, Christian E. Schaerer
National University of Asunción, Polythecnic School,
San Lorenzo, Paraguay
{*juan.vicente.bm, cschaer*}*@pol.una.py*

and

Diego H. Stalder, Giohanna Martínez
National University of Asunción, Engineering School,
San Lorenzo, Paraguay
dstalder@ing.una.py, gmartinez@fiuna.edu.py

Abstract

Public health problems such as dengue fever need accurate forecasts so governments can take effective preventive measures. Machine learning, in particular deep learning (DL) have become increasingly popular as the volume of data increases continuously. Nevertheless, performing accurate predictions in areas with fewer cases can be challenging. When we apply DL models using long short-term memory (LSTM) in different cities considering weekly dengue incidence and climate, some models may present heterogeneous behaviors and poor accuracy because of the need for more data. To mitigate this problem, clustering analysis across time series is performed based on scores to measure the clustering quality in 217 Paraguayan cities. First, we compare the raw and feature-based clustering techniques considering several metrics. Our results indicate that hierarchical clustering combined with Spearman correlation is the most appropriate approach. Finally, several LSTM models built using clustering results were compared. The root-mean-square error confirms that the clustered models improve accuracy by $19.48 \pm 18.80\%$. Finally, we present a comparison between one-dimensional statistical clustering and our best clustering setup. The main contribution of this work is a technique that can improve the performance of time series models that combine information from similar time series and weather data.

Keywords: times series forecasting, LSTM, epidemiology, dengue.

1 Introduction

Dengue virus is an arthropod-borne virus transmitted mainly by the *Aedes aegypti* mosquito acting as a vector, with a higher incidence in urban areas. Symptoms include fever, headaches, joint and muscle pain, and nausea, disease varies from mild fever to severe conditions of dengue hemorrhagic fever and shock syndrome [1]–[3]. Over 50,000,000 infections are thought to occur annually globally, including 500,000 hospitalizations for dengue hemorrhagic fever [4]. Dengue infections have dramatically increased during the past ten years in South American nations like Colombia, Ecuador, Paraguay, Peru, Venezuela, and Brazil. Dengue is also recognized as an endemic trait, which is why it is regarded as a public health issue in tropical and subtropical areas. [5].

In Paraguay, since 2009, a constant circulation has been observed, reporting between the years 2009 to 2015 a sustained increase in cases and a third major epidemic in 2013, the year in which 153,793 reported cases were observed [6]. More than one hundred deaths were found among all the cases the Ministry of Public Health and Social Welfare (MSPBS) documented during this time.

Deterministic and statistical models have been developed to predict the incidence of the disease as a function of time, based on epidemiological and entomological studies, including meteorological factors,

vector population density, or social media data [7]–[11]. Therefore, multivariate metrics were developed to select the variables for the models [12]–[14]. However, the relationship between dengue incidence and meteorological data is highly complex and cannot be easily inferred [15]–[18]. Additionally, some data from healthcare providers arrive in the reporting system in a delayed manner. This is why data-driven approaches based on machine learning and deep learning have become competitive alternatives to traditional models [19]–[22].

Deep learning approaches, specifically LSTM (Long Short-Term Memory), have proven that they can outperform state-of-the-art models and have been used to forecast influenza trends successfully [23]–[27]. However, fitting LSTM models to the time series of many cities is a challenging task because some cities display low and high incidences in a sort of heterogeneous behavior. In [28], [29], authors proposed to group similar time series instead of fitting one model for each city. However, they have chosen their groups based only on the performance of the model.

Recently we presented a comparison between traditional machine learning approaches (e.g., Lasso regression (LR), Random Forest (RF), Support Vector Regression (SVR), and Deep Learning (DL)) by measuring the RMSE [30]. This work presents improvements to cluster analysis, raw-based and feature-based approaches, and additional metrics such as Kalinski Harabasz and Dunn to measure the group's quality. Finally, group-based DL models are considered to demonstrate the improvement due to the clustering approach, and new metrics such as the standard deviation of percentual Errors and the maximum of percentual errors are considered. Another important comparison between one-dimensional statistical clustering and our best clustering setup is presented to analyze the interpretability of our clusters.

This paper is organized as follows: §2 presents the methodology, the used dataset, the sampling, and preprocessing. Section 3 provides experiments for the selection of the clustering technique of the time series and its evaluation. §4 provides a brief review of the selected model and explains how the data is grouped for experiments, showing and discussing the results. Finally, §5 draws the conclusions.

2 Methodology

A grouping technique comparison is performed based on the similarity of the incidence time series. To verify whether the clustering improves the model performance, a LSTM model is trained considering other natural grouping schemes, such as the administrative division of the country and global model. Clustering models will be presented at §3. The overall framework of this work is illustrated in Figure 1, and the detailed steps are presented later.

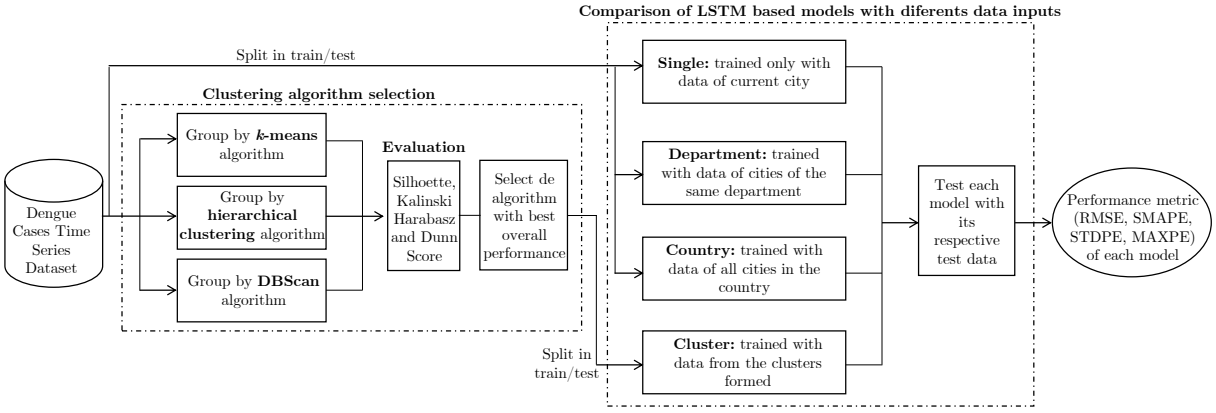


Figure 1: Summarized workflow of the model selection presenting order of procedures: a) reading the databases, b) carrying out the tests to determine the most suitable clustering technique, c) the evaluation of the LSTM model using the city data, and grouped by using the clustering criteria described in §3. Finally, the interpretability of our clusters is presented.

2.1 Datasets

Dengue fever cases (c) from January 2009 to December 2013, organized in 217 cities of 17 states, also called departments in Paraguay, and population (p) of each city, were provided by the COMIDENCO project [31]–[33]. The COMIDENCO team curated data to develop epidemiological models. The meteorological information was gathered from weather stations located all around the country [34]. Meteorological data

available included daily reports of minimum, average, and maximum temperature; minimum, average, and maximum atmospheric pressure; rainfall; maximum, average, and minimum wind speed and cloudiness.

To develop predictive models, the data is organized weekly. Once the time series for each city is obtained, the Dengue fever incidence is computed. In this article, we consider the following definition for the incidence of Dengue in a city in terms of percentage.

The incidence of Dengue fever in a city, denoted as Icd, is given by

$$Icd := 100\frac{c}{p}, \tag{1}$$

where c is the number of cases per week and p is the population.

Once the incidence is calculated, weather features associated with each time series have to be chosen. According to [28], the features selected are average temperature, average atmospheric pressure, and rainfall.

To obtain effective values for each feature associated with a time series, we geographically interpolate the values recorded at the two closest weather stations, as needed. Then the features are linearly combined using weight factors proportional to their corresponding city populations.

Finally, our model uses four variables: the incidence Icd and the three meteorological variables (which are chosen previously). Each time series has 265 records since we have 265 weeks in the period 2009-2013.

2.2 Clustering algorithms

Clustering is an unsupervised machine learning task aimed to classify a large amount of data in groups when there is no prior knowledge about real groups. Partitions in groups are made in such a way that the elements of a group are as similar as possible to each other [35]. Time series clustering can be used to improve the performance of a predictive model for Dengue fever in two ways:

1. Each group can be taken as a unit, in this way, a model can be adjusted for all the individual components of that cluster, thus reducing the number of necessary adjustments per city.
2. The more data available for deep learning models, the better the network performance becomes as it helps to avoid overfitting.

K-means [36] algorithm tries to find a partition of the samples in k clusters so that each sample belongs to one of them, specifically the one whose centroid is closest. A centroid is the middle of a cluster, which can be thought of as the multidimensional average of the cluster. Algorithm 1 shows the k-means partitioning process.

Algorithm 1: k-means

```
Data: time series to cluster, number of clusters to form (k)
Result: clusters
place the centroids c_1, c_2, ..., c_k randomly
do
    foreach datapoint x_i do
        find the nearest centroid (c_i)
        assign the point to that cluster
    end
    foreach cluster j = 1, ..., k do
        c_j = mean of all points assigned to that cluster
    end
while convergence or maximum of iterations
```

In *hierarchical clustering* [37], clusters are generated hierarchically, as the name implies. It starts by taking every data point as a cluster. Then, the closest points merge into a single cluster, and so on, until all points are in a single cluster. Algorithm 2 shows the hierarchical clustering process.

Algorithm 2: Hierarchical clustering

```
Data: n time series to cluster
Result: dendogram
assign each item to a cluster
for i = 1 to n − 1 do
    find the most similar pair of clusters and merge them into a single cluster
    recalculate the distance between the new cluster and the other points
end
```

Finally, a cluster of size n is obtained, where n is the initial number of points to be grouped. It seems pointless to form a single large group with all elements. However, the goal of hierarchical clustering is to form a dendrogram. A dendrogram is a tree that shows the merging process, from this dendrogram, cut points can be defined and form groups as seen in Figure 2.

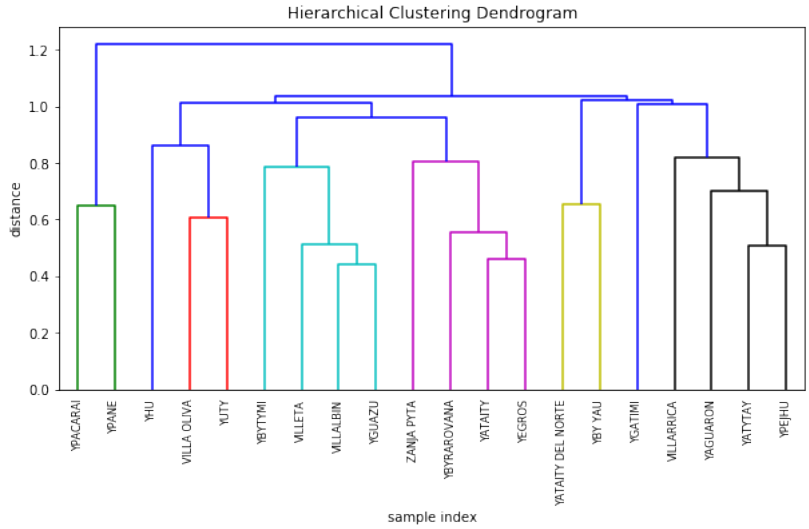


Figure 2: Dendrogram formed with a sample of 20 cities from the COMIDENCO dataset [32], five groups formed can be observed, the lines represent the distances between elements and the lines of the same color represent those that are in the same cluster.

In *DBScan* [38], for each point, the neighborhood of a given radius must contain at least a minimum number of points to belong to a cluster. DBScan needs two parameters:

- *eps*. Is the radius of distance to define a neighborhood, *i.e.*, if the two points are at a distance $\leq eps$ it means that they are in the same neighborhood.
- $MinPts$. Minimum number of neighbors, *i.e.*, data points within the eps radius.

The algorithm starts by visiting a random point, the neighborhood of this point is visited, and if it has enough points ($\geq MinPts$) it is said that it is dense enough and a cluster is started on it. If not, the point is labeled as noise. This process continues until a densely connected cluster is built. Then a new point is visited to discover another cluster or noise. Algorithm 3 describes the DBScan.

Algorithm 3: DBScan

```
Data: time series to cluster, eps, MinPts
Result: clusters
foreach p unvisited points do
    mark p as visited
    mark as neighbors points with distance ≤ eps from p
    N = neighborhood length of p
    if N ≥ MinPts then
        C = clusters of p neighborhood
        if p is not a member of any cluster then
            add p' to cluster
        end
    else
        mark p as noise
    end
end
```

As seen, there are parameters that must be entered beforehand to run the algorithms, the most crucial being the number of clusters (k) is a complex task. *Elbow*

method is a heuristic method that consists of graphing the variation of an error metric as a function of the number of clusters and choosing the elbow of the curve as the number of clusters to use. This method works by computing the algorithm method, *e.g.*, k-means for different values of k, varying k from 1 to n ($n \geq 2$), then choosing the value where the error starts to stop being significant.

2.3 LSTM Model

The deep neural networks considered for this work are specifically designed to forecast time series. This is due to the memory cells (LSTM) which preserve long and short dependencies [39], [40]. LSTM cells have input (i), forget (f), and output (o) gates which determine the addition of new information to cell state (C), deletion of less important information from memory, and output gate that controls the output prediction (h). Similarly to Recurrent Neural Networks, LSTM networks use sequential information in which the output depends not only on the current inputs but also on previous ones, e.g. the input of a point x_t is a value x_{t-n} in the same series, where n is the look back. These gates work together to learn and store long- and short-term information related to the sequence.

States of LSTM cells are computed as follows [41]:

$$i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i), \tag{2}$$
$$f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f), \tag{3}$$
$$o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o), \tag{4}$$
$$\hat{C}_t = \tanh(W_C h_{t-1} + U_C x_t + b_C), \tag{5}$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \hat{C}_t, \tag{6}$$
$$h_t = o_t \odot \tanh(C_t), \tag{7}$$

where $W \in \mathbb{R}^{h \times d}$, and $U \in \mathbb{R}^{h \times h}$ and $b_q \in \mathbb{R}^h$ are weights matrices and bias respectively, the subscript q can be either for input gate i, output gate o, forget gate f, or memory cell c depending on what is being calculated. The subscripts d and h refer to the number of input features and the number of hidden units, respectively. The $\odot$ is the Hadamard entrywise product. Vectors $i_t \in \mathbb{R}^h$, $f_t \in \mathbb{R}^h$ and $o_t \in \mathbb{R}^h$ are the input, forget, and output gates, respectively. Vector $C_t \in \mathbb{R}^h$ is the current cell state, and vector $\hat{C} \in \mathbb{R}^h$ is the new candidate value for the cell state. The function $\sigma(\cdot)$ is a Sigmoid function and modulates equations (2)-(4) between 0 and 1.

The decisions for these three gates are dependent on the current input $x_t \in \mathbb{R}^d$ and the previous output $h_{t-1} \in \mathbb{R}^h$. If the gate is 0, then the signal is blocked by the gate. Forget Gate f_t defines how much of the previous state $h_{t.1}$is allowed to pass. Input gate i_t decides which new information from the input to update or add to the cell state. Output gate o_t solves which information to output based on the cell state. These gates work together to store and learn long and short-term sequence-related information. The memory cell C is an accumulator of the state information. Update of old cell state C_{t-1} into the new cell state C_t is computed using equation (6). The calculation of both the new candidate values $\hat{C}$ of the memory cell and output of current LSTM block h_t uses hyperbolic tangent function as in equations (5) and (7). The two states, cell state, and hidden state are being transmitted to the next cell for every time step. Weights and biases are obtained by minimizing a cost function during the training. LSTM neural networks consist of a set of connected LSTM cells.

LSTM models usually have better performance when the data are stationary. In order to check if a time series is stationary, the Augmented Dickey-Fuller test was conducted. For each series, we found that the p-value is less than 0.05. Therefore, the time series are stationary and easier to generalize [42].

During the training procedure, LSTM cell weights are tuned iteratively, starting from random weights. The main idea of this process is to cycle through all sequences in the training set a certain number of times, where each cycle is called one epoch. The number of training examples utilized per iteration is called batch size. A training step or an epoch can be divided into many iterations based on the batch size.

The most common loss function is used, i.e., the mean squared error. Squaring the error has the advantage of always resulting in a positive error and penalizes the learner more when the error is bigger. Getting smaller values of the loss function means that the prediction of our model is improving. The Adam optimization algorithm is used to minimize the loss function error. The default values from the deep learning library were used (learning rate = 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$) [43].

2.4 Training and evaluating time series models

To validate the performance of the time series forecasting models, the data is split into train-test sets, $70\% - 30\%$ for network training and test, respectively. During the training, observed data is the input, and

during the test, the performance is evaluated. Therefore, our train set consists of 185 of 265 records. The input is the incidence of Dengue cases and three meteorological variables (incidence, average temperature, average atmospheric pressure, and rainfall). To describe the input vector associated with each city, consider a vector associated X_{ti} corresponding to a specific week ti, where the vector $X_{ti} = [Icd_{ti}, T^a_{ti}, Pr^a_{ti}, R^w_{ti}]$. Then the input of the model is a concatenated vector $X = [X_{t1}, X_{t2}, X_{t3}, ..., X_{t185}]$ which has the information of the 185 weeks. Figure 3 shows the LSTM architecture considered in this work.

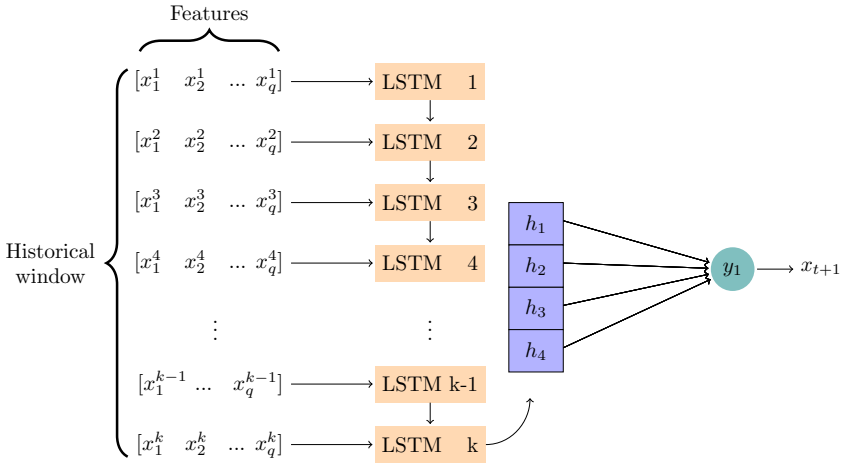


Figure 3: LSTM architecture showing the hidden state and the historical windows.

3 Time Series Clustering Analysis

Three main approaches to time series clustering are encountered in the literature (see, for instance, [35]): (i) distance-based, directly with distances on raw data points; (ii) feature-based, indirectly with features extracted from the raw data; (iii) model-based, indirectly with models built from the raw data. The performance of clustering approaches (i) and (ii) depend greatly on the particular distance metric used because the time series may have noise, different dynamics, different scales, etc.

According to the literature, there are three primary methods for clustering time series (for an example, see [35]): The first method uses distances between raw data points directly; the second uses characteristics taken from the raw data in an indirect manner, and the third method uses models created from the raw data in an indirect manner. Because the time series may contain noise, different dynamics, different scales, etc., the performance of the first two clustering procedures strongly depends on the chosen distance metric.

The primary challenges are a) figuring out the number of clusters, b) defining the metrics of similarity, and c) if feature-based clustering is used, figuring out which features are the most important ones. This is because there is no prior knowledge to categorize the time series.

Table 1: Silhouette, Calinski Harabasz and Dunn scores,values for clustering methods. Values in bold indicate the best ones.

Clustering method	**Metric**	**Silhouette score**	**Calinski Harabasz score**	**Dunn score**
k-means	Euclidean distance	0.5077	288.3678	0.5508
	Correlation	**0.5901**	**1132.5455**	0.0014
	Spearman correlation	**0.5901**	242.2947	0.0000
	Dynamic time warping	0.5094	247.6501	**0.8168**
Hierarchical clustering	Euclidean distance	0.5439	271.5737	0.6298
	Correlation	**0.5887**	239.3821	0.0000
	Spearman Correlation	0.5629	**974.6371**	**7.6011**
	Dynamic time warping	0.4954	238.0885	0.7177
DBScan	Euclidean distance	-0.1883	3.1317	**0.3163**
	Correlation	**0.5114**	**82.6522**	0.0000
	Spearman correlation	0.4799	6.6679	0.0000
	Dynamic time warping	-0.2707	6.9671	0.0103

We conducted multiple experiments evaluating the performance of the main clustering algorithms taking into account raw-based, and feature-based approaches [35] to determine which methodology would be the most suitable for our case study. The Elbow Method is employed to choose how many clusters to consider. Additionally, a silhouette, Calinski Harabasz, and Dunn scores are used to analyze the quality of the clusters.

For the raw-based approach, k-means, hierarchical and dbscan clustering are implemented. To the evaluation of the performance of the aforementioned clustering methods, a set of distance metrics are considered. The considered metrics are the Euclidean distance, point-wise correlation, Spearman correlation, and dynamic time warping. For feature-based clustering, following [44], a set of seventeen features are considered, i.e., the Mean, Variance, First order of autocorrelation, Strength of trend, Strength of linearity, Curvature, Seasonality, Strength of trough, Number of peaks, Spectral entropy, Lumpiness, Level of shift using the rolling window, Variance change, Flat spots using discretization, Number of crossing points, Kullback-Leiber score and the Index of the maximum KLscore.

Silhouette score, Calinski Harabasz score, and Dunn score are used to analyze the separation distance between the resulting clusters. These scores are especially useful if there is no a priori knowledge of what is the true label for each object, which is the most common situation in real applications. In this paper, we consider that for a pair of clusters A and B, the silhouette score $s(i)$ is computed as [45]:

$$s(i) = \frac{b(i) - a(i)}{\max(b(i), a(i))}. \tag{8}$$

where $i \in A$, and $a(i)$ is the mean distance associated to the point i to all the other points in the cluster A. Similarly, $b(i)$ is the mean distance associated with the point i to all the points of the cluster B.

To evaluate the quality of each cluster encountered in terms of cohesion, for each cluster A, consider its associated $a(i)$ as a metric between the point i and the corresponding cluster A, i.e. how well the point i is assigned to the cluster A. In this case, the smaller the value of $a(i)$, the better the assignment.

To evaluate how the clusters are well defined in terms of separation one from each other, we consider a cluster B such as $i \notin B$, and such that the distance between $i \in A$ and the set B is the closest amongst all other encountered clusters. The expression (8) provides a metric between the considered clusters A and B. In this case, the smaller the value of $s(i)$, the more the proximity of the clusters A and B. From expression (8), notice that $s(i)$ lies in the range of $[-1, 1]$.

The Caliniski-Harabasz score for K number of clusters in a data set $D = [d_1, d_2, d_3, ..., d_N]$ is defined as:

$$CH = \left[\frac{\sum_{k=1}^{K} n_k \|c_k - c\|^2}{K-1}\right] / \left[\frac{\sum_{k=1}^{K}\sum_{i=1}^{n_K} \|d_i - c_K\|^2}{N-K}\right] \tag{9}$$

where, n_k and c_k are the number of points and centroid of the k^{th} group respectively, c is the global centroid, N is the total number of data points.

A higher Caliniski-Harabasz score index value means that the clusters are dense and well separated, although there is no acceptable cut-off value [46].

Dunn score also called the Dunn index, is defined as

$$U = \min_{1\le i\le c}\left\{\min_{1\le j\le c, j\neq i}\left\{\frac{\delta(X_i, X_j)}{\max_{1\le k\le c}\{\Delta(X_k)\}}\right\}\right\} \tag{10}$$

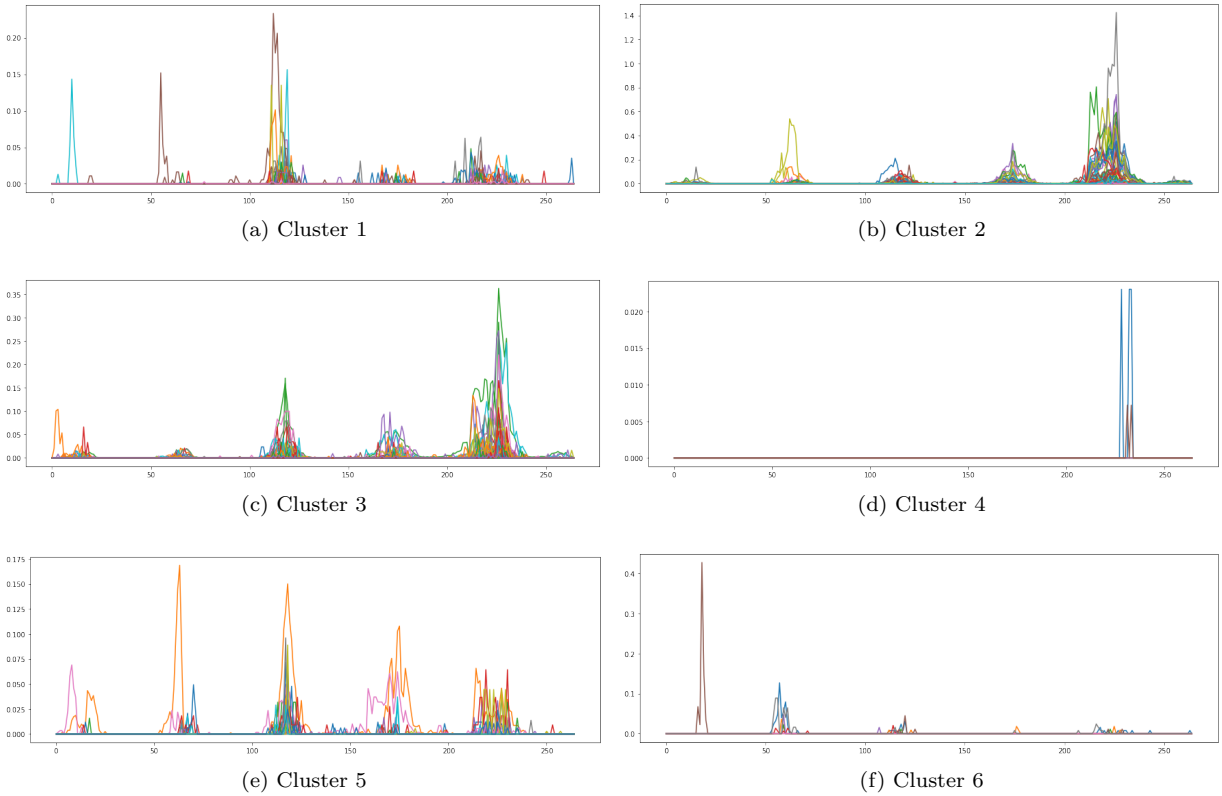
where $\delta(X_i, X_j)$ is the intercluster distance and $\Delta(X_k)$ is the intracluster distance.

Table 1 presents the Silhouette, Calinski-Harabasz, and Dunn scores in order to compare the raw-based and feature-based clustering. The best results are obtained more consistently with all metrics with the feature-based methods using Spearman correlation. K-means and Hierarchical clustering present similar clustering scores. However, based on [28] results, the method we selected was hierarchical clustering. Distance-based approach recognized a greater number of districts as outliers while getting a number of seven clusters. Feature-based approach allocated the data in four clusters and recognized fewer districts as outliers.

Acknowledging the feature-based approach leads to better performance on clustering, further experiments were focused on this method following the proposal techniques from [29]. To this end, we obtain global features from a time series to summarize, describe, and group them.

3.1 Grouping by incidence to evaluate the forecasting

We divide the time series of the cities in three equal-sized groups according to their population. This is because less populated areas usually have fewer cases. The first group (denoted as Group 1, which corresponds to the most populated cities) is composed by taking the population from the 66rd to 100rd



(a) Cluster 1

(b) Cluster 2

(c) Cluster 3

(d) Cluster 4

(e) Cluster 5

(f) Cluster 6

Figure 4: Clusters formed. The similarity between the members of each cluster can be appreciated.

percentile, the second group (denoted as Group 2, which corresponds to intermediate populated cities) from the 33rd to 66rd percentile, and the last group (denoted as Group 3, corresponds to less populated cities) from the 0rd to 33rd percentile. We present time series of five cities randomly selected from each group. The selected cities are: San Lorenzo, Capiatá, Caaguazú, Areguá, Salto del Guairá, Choré, Juan León Mallorquín, Santa Rosa del Aguaray, Quiindy, Eusebio Ayala, Encarnación, San Pedro del Ycuamandijú, Capitán Miranda, Yhú, and Santa Rita.

4 Forecasting in groups

This research aims to enhance a deep-learning neural network's capacity to forecast dengue cases through clustering. Clustering can be done in four different ways: (i) by taking into account each city individually; (ii) by taking into account the administrative division of the nation into departments (Paraguay has 17 departments or states, each of which includes several neighboring cities); (iii) by combining all the series of each city; and (iv) by forming groups using the best clustering technique. Based on this, the approaches that are taken into consideration are *City*, which is trained using only data from the city (for this approach, we have 217 models), *Department*, which is trained using data from the department (for this approach, we have 17 models), *Cluster*, which is trained using data from each cluster encountered as described in §3 (six models are implemented with this approach), and *Country*, which is trained using all the data (one model is implemented with this approach).

Algorithm 4: Input for models

Data: time series
Result: inputs for models
foreach *timeSeries* **do**
  $TS \leftarrow timeSeries$
  $city \leftarrow TS$
  $departament \leftarrow$ all-time series that belongs to the same department as TS
  $cluster \leftarrow$ all-time series that belongs to the same cluster as TS
  $country \leftarrow$ all-time series
end

The algorithm presented in Algorithm 4 illustrates the procedure of the function that produces inputs for each model. Note that the input for *country* will be the same for all cities. All models are evaluated at the city level to verify for generalization in the forecasts. The summarized workflow of the model selection can be seen in Figure 1.

The models *City*, *Department*, *Cluster*, and *Country* were trained with data from the city, the department, the cluster to which it belongs, and all the data from the country, respectively. The test set consists of the first thirty-five weeks of the year 2013. The metrics calculated to evaluate the forecast of each model are the RMSE, the standard deviation of the RMSE, and the maximum percentual errors. The group and selected cities were described in §3.1.

To measure the performance of each model, we use the root mean squared error ($RMSE$), the Standard Deviation of Percentual Error ($STDERR$), and Maximum Percentual Error ($MAXERR$), which are used to measure the distance between the predicted and observed values. In this paper, we consider the following definition for the root-mean-squared error.

The root mean squared error ($RMSE$) is defined as follows

$$RMSE := \sqrt{\frac{1}{n}\sum_{i=0}^{n}(Y_t - \hat{Y}_t)^2}, \tag{11}$$

The Standard Deviation of Percentual Error is defined as follows

$$STDERR := \sqrt{\frac{\sum(X-u)^2}{n}} \tag{12}$$

where u is the mean of data, and X is defined as

$$X := \frac{(Y_t - \hat{Y}_t)}{Y}100 \tag{13}$$

where The Maximum Percentual Error is defined as

$$MAXERR := \max\left\{\frac{(Y_t - \hat{Y}_t)}{Y}100\right\} \tag{14}$$

in equations11 to 14 Y_t is the Dengue incidence observed for time t, and $\hat{Y}_t$ is the incidence predicted by the model for a time t, and n is the size of the set.

Figure 5, 6 and 7 present the metrics values for the test set and each city of the groups. Figure 5 shows that model *Cluster* has the best results for each city of Group 1, followed by the *City* and *Department* models. However, in Figure 6 and Figure 7 we can see that *Department* and *City* performs better than the *Cluster* model.

The *Cluster* model presents the lowest RMSE in most cities compared to *City*, *Department*, and *Country* in Group 1. Figure 8 shows that in the test case, the *Cluster* model performs better even for cities from Groups 2 and 3.

The typical maximum incidence, or *Icd*, is around 0.33. All models perform well in cities with incidence rates close to the mean. Only in Group 2 (intermediate incidence cities) the *Cluster* model barely outperforms other models. But in cities from Group 1, such as Capiatá, where RMSE improves 17.2% in comparison to the benchmark performing model, the *Cluster* model performs significantly better. The *City* and *Department* models frequently fail almost entirely in models with incidences that are well outside the average, with the *Cluster* model performing up to 62.5% percent better. The *Country* model consistently comes in second place in experiments; nevertheless, when it surpasses the *Cluster* model, the average improvement is only 8.3%, which is negligible in comparison to the other results.

Figure 5: Forecasting Metrics for cities en in Group 1

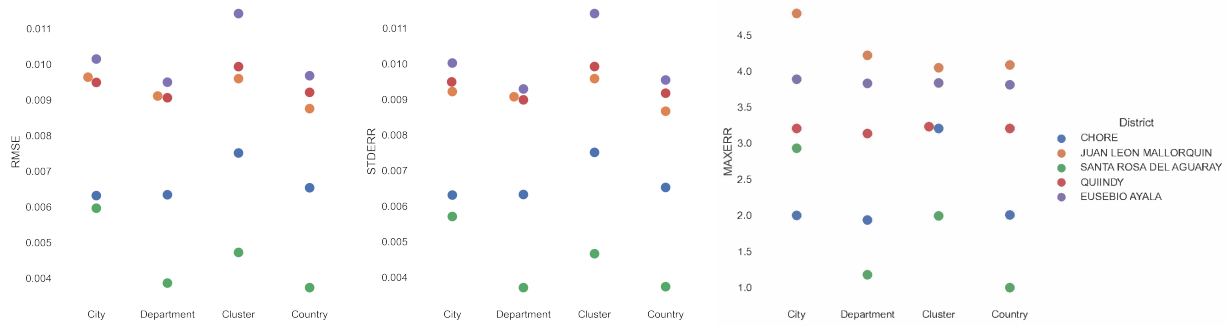


Figure 6: Forecasting Metrics for cities en in Group 2

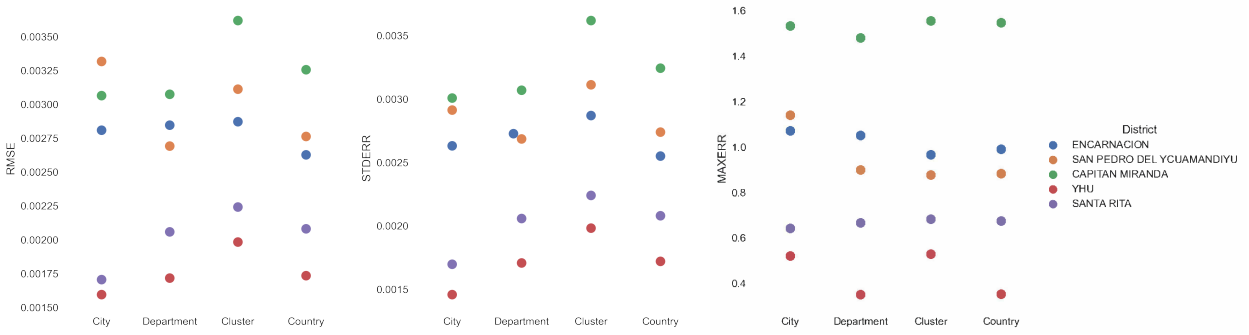


Figure 7: Forecasting Metrics for cities en in Group 3

Peaks in cities with incidence *Icd* levels below the national average are understated by the *Country* model. The city of Encarnación, which has the worst result for this model with a 10.3% below average, is a good example of this. When the model is trained using all the cities (217) in the database, there are many cities with low incidence, which results in lower forecasts, the underestimation may be caused by this.

City model performs very similarly to *Cluster*, but fails in high incidence cities. The discrepancy between the best model and the *City* model in the situations of Areguá (difference of 16.5%) and San Lorenzo (difference of 62.5%) makes this very evident. Low-incidence cities typically do not keep track of cases in the early years of an epidemic, which could be one reason for this phenomenon. This indicates a lack of data for the model, which prevents it from understanding how outbreaks behave.

The *Department* model performs the worst overall. This should not come as a huge surprise because the occurrence of dengue cases is unrelated to the political-territorial arrangement.

In general, the *Cluster* model significantly improves the forecast, especially in cities from Group 1 (the cities with a high incidence of Dengue Fever). As far as the amount of computational work needed to cover the entire country with *Cluster* it is necessary to train six models, for *Country* only one but with a large amount of data which is quite computationally expensive, with *City* 217 models should be trained, which also represents a lot of work. Hence, the *Cluster* presents the best trade-off between computational cost and performance when the country needs to be analyzed.

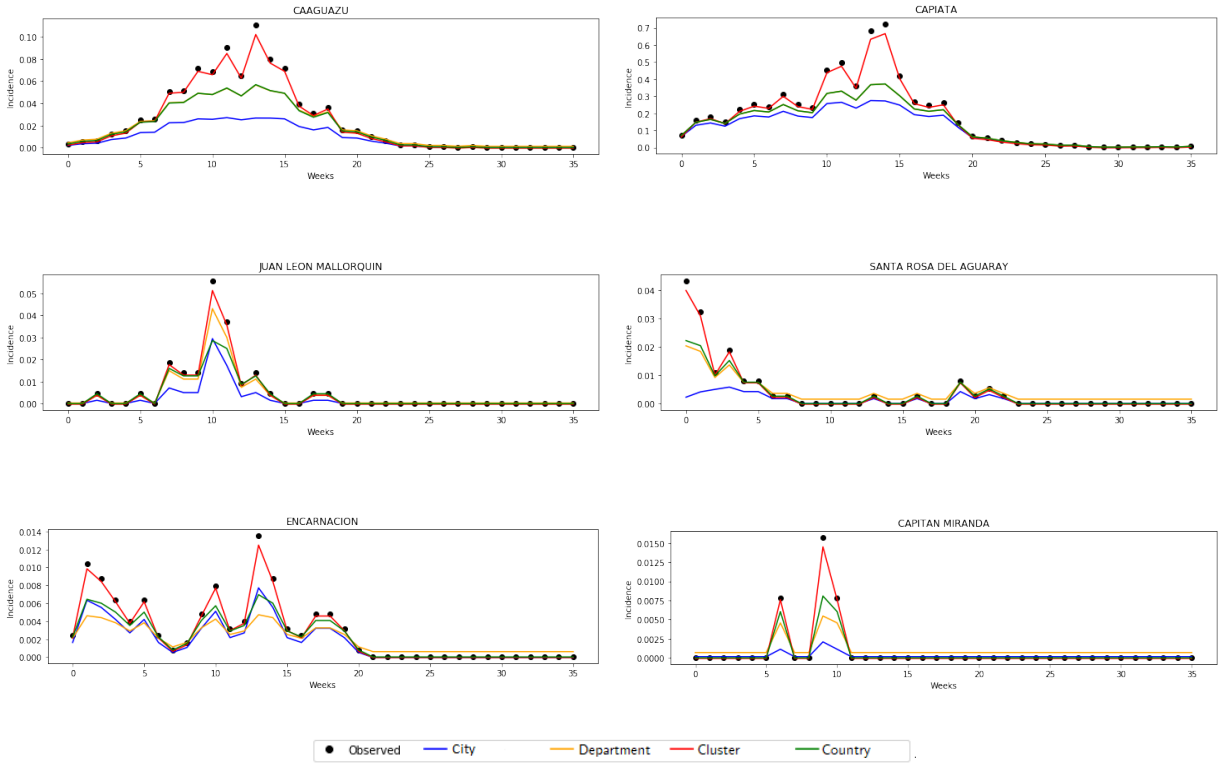


Figure 8: Prediction of the incidence of Dengue *Icd* in Caaguazú and Capiatá from Group 1; Juan León Mallorquin and Santa Rosa from Group 2; and Encarnación and Capitán Miranda from Group 3. Comparison of the *City*, *Department*, *Cluster*, and *Country* models with a prediction of the first 35 weeks of the year 2013.

5 Conclusion

A forecasting method will be a useful tool for guiding the efforts of the health surveillance system ahead of a Dengue regional outbreak, helping to prevent the possibility of its evolution into an epidemic situation. However, due to the lack of (or incomplete) datasets in several regions, forecasting methods may fail in their predictions, affecting the effectiveness of actions taken.

In this work, we propose the grouping of data to improve the performance of the models. Several models were evaluated with different grouping approaches. We tested clustering techniques to correct the lack of data for some cities. Our results indicate that an LSTM model, in combination with a hierarchical clustering algorithm, improves forecast accuracy.

With data clustering, we were able to improve the performance of the models. Also we were able to decrease the number of models needed to make a national prediction at the city level, since a model fitted with a cluster can be used to make predictions in each city that belongs to that cluster. We believe this approach can be applied to wide geographic regions and other mosquito-borne diseases. In future works, we will aim to optimize the clustering of the time series by designing more experiments and also combining data augmentation techniques with clustering.

Acknowledgment

Authors acknowledge the financial support given by PINV15-706 COMIDENCO and FEEI-PROCIENCIA-CONACYT project POSG17-62. JVB, DHS and CES acknowledges the FEEI-PROCIENCIA-CONACYT-PRONII. The authors acknowledge the Ministerio de Salud Pública y Bienestar Social (MSPBS) for the data and fruitful discussions on topics of this work.

References

[1] F. Brauer, C. Castillo-Chavez, and Z. Feng, “Dengue fever and the Zika virus,” in *Mathematical Models in Epidemiology*, Springer, 2019, pp. 409–425.

[2] S. A. Kularatne, "Dengue fever," *BMJ*, vol. 351, 2015.

[3] S. H. Waterman and D. J. Gubler, "Dengue fever," *Clinics in dermatology*, vol. 7, no. 1, pp. 117–122, 1989.

[4] M. G. Guzman, S. B. Halstead, H. Artsob, *et al.*, "Dengue: A continuing global threat," *Nature Reviews Microbiology*, vol. 8, no. 12supp, S7, 2010.

[5] D. Romero, J. Olivero, R. Real, and J. C. Guerrero, "Applying fuzzy logic to assess the biogeographical risk of dengue in South America," *Parasites & vectors*, vol. 12, no. 1, p. 428, 2019.

[6] Ministerio de Salud Pública y Bienestar Social Dirección General de Vigilancia de la Salud. Enfermedades transmitidas por vectores, *Boletín epidemiológico. 30:11*, 2013. [Online]. Available: `http://vigisalud.gov.py/files/boletines/SE51_2013_Boletin.pdf`.

[7] C. Codeco, F. Coelho, O. Cruz, S. Oliveira, T. Castro, and L. Bastos, "Infodengue: A nowcasting system for the surveillance of arboviruses in Brazil," *Revue d'Épidémiologie et de Santé Publique*, vol. 66, S386, 2018.

[8] P. E. Pérez-Estigarribia, P.-A. Bliman, and C. E. Schaerer, "A class of fast-slow models for adaptive resistance evolution," *Theoretical Population Biology*, vol. 135, pp. 32–48, 2020. DOI: `https://doi.org/10.1016/j.tpb.2020.07.003`.

[9] T. F. M. De Lima, R. M. Lana, T. G. de Senna Carneiro, *et al.*, "Dengueme: A tool for the modeling and simulation of dengue spatiotemporal dynamics," *International journal of environmental research and public health*, vol. 13, no. 9, p. 920, 2016.

[10] M. Andraud, N. Hens, C. Marais, and P. Beutels, "Dynamic epidemiological models for Dengue transmission: A systematic review of structural approaches," *Plos one*, vol. 7, no. 11, 2012.

[11] M. G. Martinez, D. H. Stalder, C. E. Schaerer, and J. V. Bogado, "Feature selection within time series clustering," *Proceedings of the 3rd South American International Industrial Engineering and Operations Management Conference*, 2022.

[12] R. Arias-Michel, M. García-Torres, C. E. Schaerer, and F. Divina, "Feature selection using approximate multivariate markov blankets," in *Hybrid Artificial Intelligent Systems*, F. Martínez-Álvarez, A. Troncoso, H. Quintián, and E. Corchado, Eds., Cham: Springer International Publishing, 2016, pp. 114–125, ISBN: 978-3-319-32034-2.

[13] S. Gómez-Guerrero, G. Sosa-Cabrera, M. García-Torres, I. Ortiz-Samudio, and C. E. Schaerer, "Multivariate symmetrical uncertainty as a measure for interaction in categorical patterned datasets," in *the Entropy 2021: The Scientific Tool of the 21st Century*, MDPI: Basel, Switzerland, 2021, pp. 114–125. DOI: `10.3390/Entropy2021-09826`.

[14] G. Sosa-Cabrera, M. García-Torres, S. Gómez-Guerrero, C. E. Schaerer, and F. Divina, "A multivariate approach to the symmetrical uncertainty measure: Application to feature selection problem," *Information Sciences*, vol. 494, pp. 1–20, 2019.

[15] Y. Shi, X. Liu, S.-Y. Kok, *et al.*, "Three-month real-time dengue forecast models: An early warning system for outbreak alerts and policy decision support in Singapore," *Environmental health perspectives*, vol. 124, no. 9, pp. 1369–1375, 2016.

[16] F. D. Silva, A. M. d. Santos, R. d. G. C. F. Corrêa, and A. d. J. M. Caldas, "Temporal relationship between rainfall, temperature and occurrence of dengue cases in São Luís, Maranhão, Brazil," *Ciencia & saude coletiva*, vol. 21, pp. 641–646, 2016.

[17] M. C. Ramírez-Soto, J. V. B. Machuca, D. H. Stalder, D. Champin, M. G. Mártinez-Fernández, and C. E. Schaerer, "Sir-si model with a gaussian transmission rate: Understanding the dynamics of dengue outbreaks in lima, peru," *Plos one*, vol. 18, no. 4, e0284263, 2023.

[18] M. A. Johansson, N. G. Reich, A. Hota, J. S. Brownstein, and M. Santillana, "Evaluating the performance of infectious disease forecasts: A comparison of climate-driven and seasonal dengue forecasts for Mexico," *Scientific reports*, vol. 6, p. 33 707, 2016.

[19] S. Jiang, R. Xiao, L. Wang, *et al.*, "Combining deep neural networks and classical time series regression models for forecasting patient flows in Hong Kong," *IEEE Access*, vol. 7, pp. 118 965–118 974, 2019.

[20] D. Sun, M. Wang, and A. Li, "A multimodal deep neural network for human breast cancer prognosis prediction by integrating multi-dimensional data," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 16, no. 3, pp. 841–850, 2018.

[21] S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *Briefings in bioinformatics*, vol. 18, no. 5, pp. 851–869, 2017.

[22] J. D. Mello-Román, J. C. Mello-Román, S. Gomez-Guerrero, and M. García-Torres, "Predictive Models for the Medical Diagnosis of Dengue: A Case Study in Paraguay," *Computational and mathematical methods in medicine*, vol. 2019, 2019.

[23] L. Liu, M. Han, Y. Zhou, and Y. Wang, "LSTM recurrent neural networks for influenza trends prediction," in *International Symposium on Bioinformatics Research and Applications*, Springer, 2018, pp. 259–264.

[24] L. Wang, J. Chen, and M. Marathe, "TDEFSI: Theory-guided deep learning-based epidemic forecasting with synthetic information," *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, vol. 6, no. 3, pp. 1–39, 2020.

[25] J. Xu, K. Xu, Z. Li, *et al.*, "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.

[26] P. H. Khotimah, A. F. Rozie, E. Nugraheni, A. Arisal, W. Suwarningsih, and A. Purwarianti, "Deep learning for dengue fever event detection using online news," in *2020 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*, IEEE, 2020, pp. 261–266.

[27] T. Harumy, H. Chan, and G. Sodhy, "Prediction for dengue fever in indonesia using neural network and regression method," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1566, 2020, p. 012 019.

[28] E. Mussumeci and F. C. Coelho, "Large-scale multivariate forecasting models for Dengue-LSTM versus random forest regression," *Spatial and Spatio-temporal Epidemiology*, vol. 35, p. 100 372, 2020.

[29] K. Bandara, C. Bergmeir, and S. Smyl, "Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach," *Expert Systems with Applications*, vol. 140, p. 112 896, 2020.

[30] J. V. Bogado, D. H. Stalder, C. E. Schaerer, and S. Gómez-Guerrero, "Time series clustering to improve dengue cases forecasting with deep learning," in *2021 XLVII Latin American Computing Conference (CLEI)*, 2021, pp. 1–10. DOI: 10.1109/CLEI53233.2021.9640130.

[31] E. G. S. Riveros, S. Gómez-Guerrero, and C. E. Schaerer, "Categorical PCA and Multiple Correlation in the study of the incidence of Dengue fever in communities of Paraguay," *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, vol. 6, no. 2, 2018.

[32] S. Gómez-Guerrero, C. Schaerer, A. Rojas de Arias, J. Mello, and H. Estigarribia, "Construcción de un modelo de incidencia de dengue aplicado a comunidades de Paraguay," in *Segundo Encuentro de investigadores*, Sociedad Científica del Paraguay, 2017. [Online]. Available: http://cimapy.org/en/research/proyects/comidenco-en.

[33] J. V. Bogado, D. Stalder, S. Gómez-Guerrero, and C. E. Schaerer, "Deep learning-based dengue cases forecasting with synthetic data," *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, vol. 7, no. 1, 2020.

[34] Dirección Nacional de Aeronáutica Civil,
, Aug. 2015. [Online]. Available: https://www.meteorologia.gov.py/.

[35] T. W. Liao, "Clustering of time series dataÃ¢ÂÂa survey," *Pattern recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.

[36] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," Stanford, Tech. Rep., 2006.

[37] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: An overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012.

[38] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *Kdd*, vol. 96, 1996, pp. 226–231.

[39] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[40] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[41] S. Bouktif, A. Fiaz, A. Ouni, and M. A. Serhani, "Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches," *Energies*, vol. 11, no. 7, p. 1636, 2018.

[42] P. Josef, S. Skipper, and T. Jonathan, *Statsmodels.tsa.stattools.adfuller*, 2013. [Online]. Available: `https://www.statsmodels.org/devel/generated/statsmodels.tsa.stattools.adfuller.html`.

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

[44] R. J. Hyndman, E. Wang, and N. Laptev, "Large-scale unusual time series detection," in *2015 IEEE international conference on data mining workshop (ICDMW)*, IEEE, 2015, pp. 1616–1619.

[45] G. Ogbuabor and F. Ugwoke, "Clustering algorithm for a healthcare dataset using silhouette score value," *International Journal of Computer Science & Information Technology*, vol. 10, no. 2, pp. 27–37, 2018.

[46] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.