Universidad Nacional de Asunción
Facultad Politécnica

Time Series Clustering and Data Augmentation Techniques to Improve the Forecast of Dengue Cases in Paraguay with Deep Learning

Juan Vicente Bogado Machuca

Orientadores: D.Sc. Christian Schaerer
D.Sc. Diego Stalder

Tesis presentada a la Facultad Politécnica, Universidad Nacional de Asunción, como requisito para la obtención del Grado de Máster en Ciencias de la Computación.

San Lorenzo
2020

To my father José De Jesús, in loving memory.
To my wonderful mother Digna; and to my brothers
Natalia, Cintia, José Luis, and Antonella.

Acknowledgements

I would like to express my sincere gratitude to my advisors: D. Sc. Christian Schaerer, for the accompaniment and for his valuable and constructive advises during the planning and development of this work; and D. Sc. Diego Stalder, with his guidance and encouragement, he has contributed significantly to my formation as a researcher.

Thanks to M. Sc. Santiago Gómez, for supporting and reviewing my work.

A special thanks to Mg. Héctor Estigarribia who always supported me and for strongly recommending me to pursue this master's degree.

Thanks to all people at NIDTEC for the excellent working and learning environment, especially to my classmates.

Thanks to FEEI-PROCIENCIA-CONACyT for the scholarship.

Time series clustering and data augmentation techniques to improve the forecast of Dengue cases in Paraguay with deep learning[1]

Author: Juan Vicente Bogado Machuca
Advisors: Christian Schaerer, D.Sc.
Diego Stalder, D.Sc.

ABSTRACT

Dengue fever is a public health problem and accurate forecasts can help governments to take the best preventive actions. As the volume of data provided continuously increases, machine learning and deep learning (DL) models have become an attractive approach. However, it is difficult to perform accurate predictions in areas with fewer cases. In this work, traditional approaches such as LARS LASSO Regression (LR), Random Forest (RF), Support Vector Regression (SVR) vs DL models based on Long Short-Term Memory (LSTM) are compared, considering weekly Dengue incidence and climate, in 217 cities in Paraguay.

Several cities may present heterogeneous behaviors and poor accuracy, to mitigate this problem, two approaches are proposed: clustering and data augmentation. First, clustering analysis between time series was performed, based on silhouette scores for measuring how well observations are clustered. Results indicate that hierarchical clustering combined with correlation is the most appropriate approach. Then several LSTM models are compared on subgroups of similar time series. Second, several data augmentation techniques were applied, and the synthetic time series obtained was used as input to train models, the results indicate that the synthetic series obtained with Bayesian estimation technique are the one that improved the performance of the model.

The Root Mean Square Error (RMSE) confirms that the LSTM clustered models improve the accuracy in $19.48 \pm 18.80\%$ and LSTM with Bayesian based data augmentation improves $16.86 \pm 16.57\%$. The main contribution of this work are two techniques that can improve the performance of time series models by combining information from similar time-series and weather data.

[1] An expanded summary in Spanish can be found in Appendix A

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

INTRODUCTION

Dengue fever is a mosquito-borne viral disease with a higher incidence in urban areas, Dengue is transmitted to humans mainly by the *Aedes aegypti* mosquito acting as a vector. Symptoms include fever, headaches, joint and muscle pain, and nausea [10]. The disease could cause from mild fever to severe conditions of Dengue hemorrhagic fever and shock syndrome. Worldwide, it is estimated that more than 50,000,000 infections occur each year, including 500,000 hospitalizations for Dengue hemorrhagic fever [28]. Over the last decade, there has been a dramatic increase of Dengue infections in South American countries such as Colombia, Ecuador, Paraguay, Peru, Venezuela, and Brazil. It is also known that Dengue has an endemic characteristic, and this is why it is considered a public health problem in tropical and subtropical regions [58].

In Paraguay, after the first Dengue epidemic in the period 1989-1990, no outbreaks were reported for a decade, until a second big outbreak in 2007. From 2009, a constant circulation is 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 and four serotypes are registered (DEN 1, DEN 2, DEN 3, DEN 4), any of these serotypes being able to produce the disease [59], Dengue has a high incidence in the country, as seen in Figure 1.1. Among all cases reported by the Ministry of Public Health and Social Welfare (MSPBS, by its Spanish acronym) in this period, more than one hundred deaths were identified. These numbers reveal that Dengue imposes a high economic and social burden on health care systems, affecting the public health system, households, and society in general; people with underlying diseases and pregnant women are the most susceptible to complications. The average cost incurred by each patient was 5,793,544 Gs. (1053.53 USD) [57].

Currently, the fight against Dengue is based on adequate clinical and laboratory care, epidemiological surveillance, and educational campaigns with vector control programs as a basic strategy to mitigate the spread of Dengue. However, the results have not been successful and in the absence of a more effective strategy *e. g.* with the introduction of an effective vaccine, this disease will continue to produce a considerable economic and social burden. The proper application of control measures depends on the management of the beginning of the disease season. As the seasons vary over the years, accurate forecasts can be critical tools in the fight against the disease. In the absence of treatment able to control Dengue fever outbreaks, accurate and early forecasts of Dengue might minimize the problem and help the government

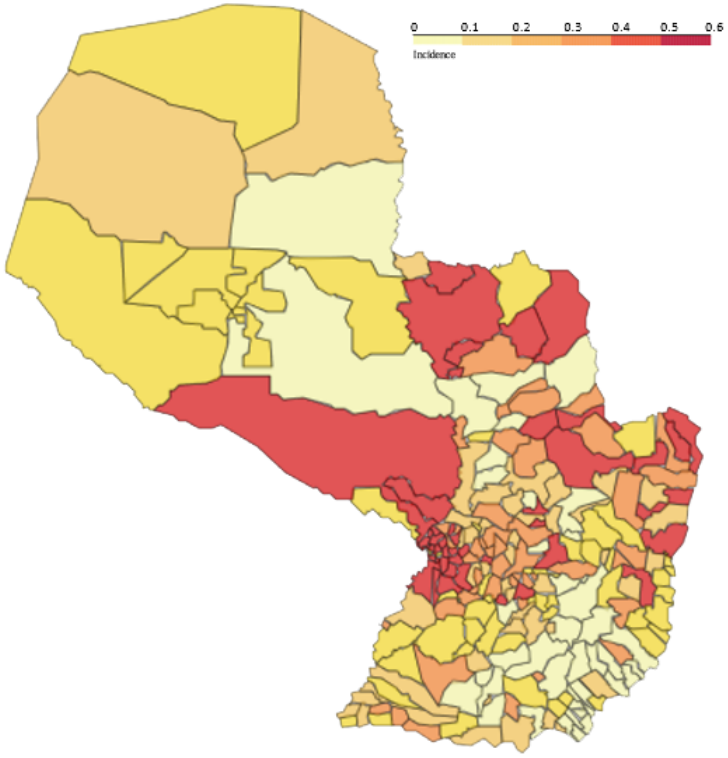


Figure 1.1: Map of Paraguay indicating the maximum incidence on each city from year 2009 to 2013.

to implement effective control measures [2].

Modeling epidemic models requires multiple unknown variables such as the population, vector population, and their dynamics. In addition, parameters such as the reproduction rate of the vector are affected by meteorological variables *e.g.*, rainfall, temperature, etc. So, to include them in the models, feature selection techniques are usually applied. These techniques use multivariate metrics to select which set of variables could be more informative to the model [67]. However, the relationship between Dengue incidence and meteorological data is highly complex and cannot be easily inferred [62, 65, 36]. Additionally, some data from health care providers can arrive delayed, incomplete, or underestimated to the reporting system. Most of the compartmental models *e.g.*, SIR Model, are restricted to fit and characterize the data only for one epidemic outbreak. This is why data-driven approaches based on machine learning and deep learning have become competitive alternatives to traditional models by considering the incidence of a disease as a time series forecasting problem [35, 71, 47, 46].

Understanding the behavior of the disease is a complex combination of epidemiological and environmental factors, and is a difficult task for classical methods to make predictions. In this context, models based on deep learning have proven to have better or the same result than statistical models [54], in addition to allowing more external variables to be handled in a simpler way. Deep learning approaches, specifically LSTM (Long Short Term Memory) proposed by Hochreiter *et al.* [30], have proven that they are able to outperform state-of-the-art models and have been used to forecast influenza trends successfully [42, 76, 80]. However, to achieve opti-

mal results with deep learning models, a large amount of data is necessary and the lack of long-term data affects the performance of these models, producing overfitting.

This work investigates which model performs better predictions in the case of Dengue fever epidemics, considering traditional machine learning vs deep learning models. When the best candidate has been selected, two strategies well known in the literature to improve model prediction are considered, *i.e.*, clustering and data augmentation.

Applying clustering techniques to time series is not a new procedure [40], but it has recently been considered to improve the performance of deep learning models [52, 50]. However, the effectiveness of the application of these techniques has not been measured and when it has been done, it has been based solely on the final performance of the model without evaluating the clustering techniques [4]. This work seeks to carry out a detailed analysis of time series clustering to determine which is the most appropriate technique to group time series of Dengue cases, and then evaluate its contribution to the performance of the deep learning models.

Data augmentation is a widely used technique in the image processing area [63], it consists of applying small transformations to the images such as adding noise or rotating them, without affecting the main characteristics of the image. These techniques, inspired by those applied in the image area, have already been adapted to time series [56], without evaluating whether there is a benefit when applied to a model. Another form of data augmentation for epidemiological series is to fit the observed data with mathematical models [68], agent-based models [76], or Bayesian data augmentation [20]. This work proposes how to apply Bayesian data augmentation to a time series, exploring the epidemiological component by applying a combination of mathematical models (SIR model) and Bayesian inference to artificially augment the data and this result is compared with the classic data augmentation techniques.

The contribution of the clustering in the forecasting of Dengue cases is the identification of geographical areas of the behavior of the disease and the reduction of the size of the models necessary to cover the country level. The contribution of the Bayesian data augmentation technique in epidemiological mathematical models is to provide complete information on the observed epidemiological events. In addition, both techniques can be used to avoid overfitting in the models [70].

1.1 Objectives

1.1.1 General Objective

1. Propose strategies to improve the accuracy of Dengue fever models based on deep learning by applying time series clustering and data augmentation

1.1.2 Specific Objectives

1. Evaluate traditional machine and deep learning models in order to select a benchmark model to forecast Dengue incidence

2. Analyze which time series clustering methods can be used to simplify the Dengue forecasting models.

3. Propose a new Bayesian data augmentation approach based on synthetic data generated by a compartmental model.

4. Evaluate traditional time series data augmentation against the proposed Bayesian approach.

5. Quantify the improvement of clustering-based and data-augmentation-based methods.

The work is organized as follows, Chapter 2, presents the techniques used for the forecast; in Chapter 3 the fundamental bases and experimental results of the first approach, clustering, are presented; Chapter 4 presents the foundations and experimental results of the second approach, data augmentation; Chapter 5 shows the discussion of the experimental results obtained and; finally, Chapter 6 presents the conclusions and future work.

Chapter 2

MODELING DENGUE FEVER

In this chapter, traditional and contemporaneous Dengue fever models are presented. Dengue fever is considered a public health problem and mathematical models help to characterize outbreaks and make decisions. In the literature, deterministic and statistical models have been developed to predict the incidence of the disease as a function of time. Some of them are based on epidemiological and entomological studies, and allow to include meteorological factors, vector population density, or even social media data.

2.1 Compartmental Models

Among all the mathematical methods for modeling epidemics, the most basic are the compartmental models. These categories of models usually assume that the individuals in the population pass through several states (compartments) over time. The model can be a set of ordinary differential equations, although can also run with a stochastic framework. They can also reproduce the spatio temporal patterns when they consider agents *e. g.*, humans, vectors, etc. [9]. The following subsection introduces one of the simplest compartmental models *e.g.*, SIR model.

2.1.1 SIR Model

SIR is the acronym for Susceptible-Infected-Recovered, which are the compartment labels or states considered in this model and are defined as follows:

- *Susceptible population (S).* Individuals without immunity to the disease and, therefore, can become infected when exposed to the infectious agent.
- *Infected population (I).* Infected individuals and who can spread the infection to susceptible individuals when they have contact depending on the disease considered.
- *Recovered population (R).* Individuals who have immunity to the disease and do not affect others when they come into contact. The deceased individuals are also in this group.

This model captures the disease dynamics by defining only two parameters: the transmission rate β and the recovery rate γ.

At one given time, each individual can have only one state and can change its state as shown in Figure 2.1. So, a susceptible individual can be infected ($S \rightarrow I$), then can recover ($I \rightarrow R$) or die. Deaths are not considered in this work. Therefore, the sum of the individuals should give the total population: $N = S + I + R$. The total population (N), remains constant because births and deaths in the population are not considered given the short period that the outbreak lasts.

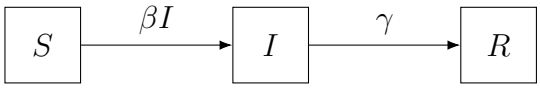


Figure 2.1: SIR model states. Adapted from [68].

Let t be the variable that indicates time and $S(t)$, $I(t)$ and $R(t)$ the number of people in each group at time t. For simplicity, the explicit representation of the time where dropped. The set of differential equations which can model the temporal variations of the fraction of people in different populations are:

$$\frac{dS}{dt} = -\beta SI \tag{2.1}$$

$$\frac{dI}{dt} = \beta SI - \gamma I \tag{2.2}$$

$$\frac{dR}{dt} = \gamma I, \tag{2.3}$$

where β is the transmission rate, *i.e.*, the average number of contacts between Susceptibles and Infectious that lead to the infection of the Susceptible, per Susceptible and per Infectious. The recovery rate γ, can be obtained as the inverse of the typical period (*e.g.*, in days) that an Infected person remains infectious. Additionally, a set of initial conditions for each state must be defined ie S_0, I_0, R_0. Therefore, for a given transmission and recovery rates, the SIR model predicts on scenario as in Figure 2.2.

2.1.2 Other Compartmental Models

When more data is available, other variables must be considered to improve the predictions, the SIR model can be extended adding new states such as:

- *Asymptomatic population (A).* Individuals who were in contact with an infected and who can infect but did not develop symptoms of the disease.
- *Exposed population (E).* Individuals in the incubation phase, who are infected but cannot yet infect.
- *Deceased population (D).* Individuals who die from the disease and are not included in the R compartment.
- *Maternally derived immunity (M).* Infants with immunity inherited from their mother.

Therefore, new states introduce new models such as SIS, SEIR, MSIR, SEIAR, SEIS, SIRD, SEIRD, MSEIR, MSEIRS [9]. New equations also require the introduction of new parameters, which can be:

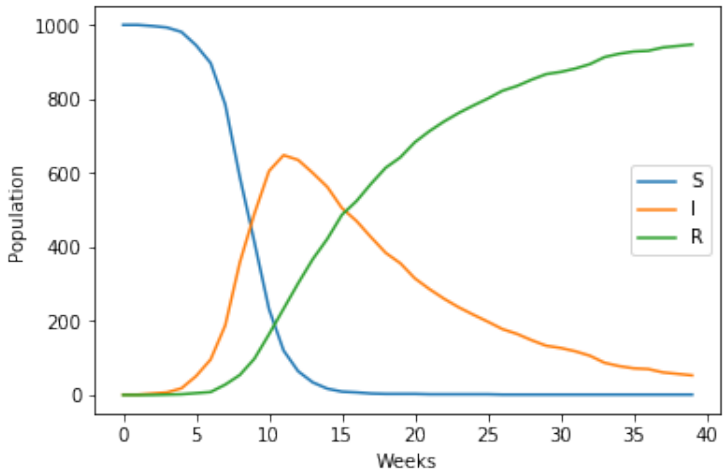


Figure 2.2: SIR model with values $S_0 = 999$, $I_0 = 1$, $R_0 = 0$, $N = 1,000$, $\beta = 0.001$ and $\gamma = 0.1$.

- μ: Death rate.
- B: Birth rate.
- σ: Temporary immunity rate.
- $\frac{1}{\epsilon}$: Incubation rate.

These models can be used to predict how a disease spreads, the total number of infected and the duration of one epidemic outbreak. In addition the aforementioned models can be used to estimate various epidemiological parameters such as the reproductive number. However, such models cannot predict the time for the next outbreak. This limitation can be overcome by changing the formulation of the problem to a time series forecasting problem. To this end, statistical or machine learning models can combine the information of more than one epidemiological season. So, they can predict when the next outbreak may start.

2.2 Time Series: Data Analysis and Forecasting

Observational data organized sequentially according to their time of occurrence are called a *time series*. Several areas like medicine, weather forecasting, economics, and astronomy have a large amount of data collected and there is a need for methodologies to analyze and forecast variables using a combination of the past and other correlated variables. In the case of Dengue fever epidemics, a time series model can be used to predict the number of Dengue cases in the following weeks or when the next outbreak can occur using historical data, see Figure 2.3. Other variables such as temperature or humidity which affects the transmission rate can be also included in the model.

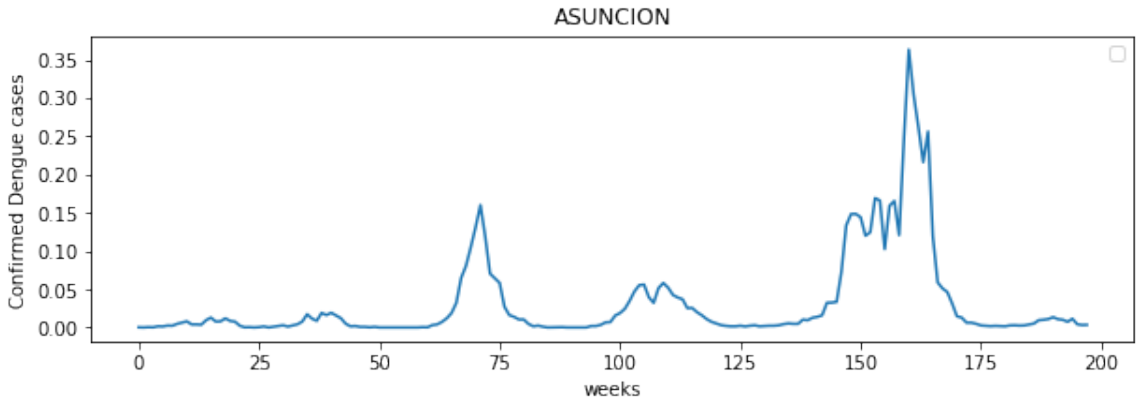


Figure 2.3: Time series that shows weekly incidence of Dengue cases in the five-year period (2009 - 2013) in Asunción, a city in Paraguay.

A time series is a sequence of N observations, equidistant and ordered chronologically through time [45]. A univariate time series can be represented as follows:

$$Y = \{y_1, y_2, y_3, ..., y_N\}, \tag{2.4}$$

where y_t is the observation at point t $(1 \leq t \leq N)$ of the time series, and N is the number of observations or length of the series. However, time series may arise in distinct ways, sampled from a continuous series (*e.g.*, temperature measured at hourly intervals) or accumulated over a period of time (*e.g.*, reported Dengue cases in a week). Real problems usually requires the analysis of multivariate series which can be organized in a matrix of order $N \times M$:

$$Y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1M} \\ y_{21} & y_{22} & \cdots & y_{2M} \\ \vdots & \vdots & \vdots & \vdots \\ y_{N1} & y_{N2} & \cdots & y_{NM} \end{bmatrix}$$

where N is the length of each observation and M is the number of observed variables. The observed variables can be categorical, numeric, or Boolean. Given these multivariate data, it is desirable to model not only the relationships between the series but also between the series, as well as analyze the temporal interdependence that the series have with each other.

Times series variations can be characterized in four categories of components [14]:

1. *Trend:* indicates whether the values of the series are increasing or decreasing during a long period of time (*e.g.*, a year).

2. *Seasonal variation:* variations that are observed regularly every certain period of time (*e.g.*, less than a year, such as weekly, monthly, or quarterly).

3. *Cyclic Variations:* variations which occur over a span of more than one year are the cyclic variations.

4. *Random or Irregular movements:* fluctuations which are unpredictable, unforeseen or uncontrollable.

Another important characterization from a statistical point of view is the stationarity [77]. This statistical property assumes that the process generating the time series does not change over time (*e.g.*, mean, variance, autocorrelation, etc. are all constant over time). However, if the time series is not stationary, a difference can be applied to obtain stationarity. For example the first difference is the series of changes from one period to the next:

$$y'_t = y_t - y_{t1}. \tag{2.5}$$

There are advanced tools and techniques to analyze time series from different areas such as statistics, machine learning and deep learning. One common approach is a statistical time series analysis because it assumes that the observations can be characterized as a stochastic process *i.e.*, a sequence of random variables, ordered and equidistant chronologically, referred to one (univariate or scalar process) or several (multivariate or vector process) characteristics of an observable unit at different times [45].

2.2.1 Statistical Models: ARIMA, SARIMA

In time-series data, a point near in time tends to be strongly correlated to another time. The simplest approach predicts the current value of the time series as a linear combination of its previous values and the current residual, this method is called Autoregressive (AR). There is a set of traditional statistical models that combine this approach with a moving average scheme, (*e.g.*, Autoregressive Moving Average (ARMA), Autoregressive Integrated Moving Average (ARIMA), and Seasonal Autoregressive Integrated Moving Average (SARIMA)) [7].

An autoregressive model (AR) is typically represented as follows:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t, \tag{2.6}$$

where $y_{t-1}, y_{t-2}, ..., y_{t-p}$ are the past values of the series, c is a constant, $\phi_1, \phi_2, ..., \phi_p$ are the autoregressive model parameters, p is the auto regression order and the error ε_t. If $p = 1$ the model will consider only $t - 1$ values to fit predict t.

A moving average (MA) model, instead of using past values for the regression uses errors from past forecasts, a moving average model can be written as:

$$y_t = c + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t, \tag{2.7}$$

where $\theta_1, \theta_2, ..., \theta_q$ are the model parameters of the moving average, q is the order (*e.g.*, a model of order $q = 3$ takes the 3 moving window of three averages) and ε_t are the error terms. The error terms are generally assumed to be independent, identically distributed variables sampled from a normal distribution with zero mean.

Finally, the differentiation with the autoregressive, the autoregression, and moving average models are combined, the ARIMA full model can be written as follows:

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t, \tag{2.8}$$

where y'_t is the differenced series (it may have been differenced more than once, *i.e.*, d). Therefore the ARIMA model has three parameters (p, d, q):

- *p:* order of autorregresive part;

- *d:* degree of differencing;
- *q:* order of the moving average part.

Where there are the following special cases:

- White noise, ARIMA(0,0,0).
- Random walk, ARIMA(0,1,0) with no constant.
- Random walk with drift, ARIMA(0,1,0) with a constant.
- Autoregression, ARIMA(p,0,0).
- Moving average, ARIMA(0,0,q).

To predict the seasonal component, a Seasonal ARIMA (SARIMA) model can be used. This model introduces additional parameters as follows:

$$SARIMA \quad \underbrace{(p,d,q)}_{\text{non-seasonal part of the model}} \quad \underbrace{(P,D,Q)_m}_{\text{seasonal part of the model}} \quad ,$$

where m is the number of observations per year. The uppercase parameters of the seasonal part are similar to the lowercase ones that we have already seen, only that they take the results of the non-seasonal part.

The models derived from artificial intelligence techniques, *i.e.*, machine and deep learning, have been shown to be able to handle multivariate time series effectively. Although statistical methods are an important basis in forecasting time series, as machine learning methods give better or equal results to statistical methods [54], this work focuses on machine learning methods.

2.2.2 Machine and Deep Learning Models

Artificial intelligence is a broad research area that includes techniques that emulate human thinking, such as pattern recognition, image classification, voice recognition, language analysis, among others [25]. Machine learning (ML) is a subset of artificial intelligence where there is a set of methods to deal with time series forecasting as a supervised learning problem. Deep learning (DL) is a branch of ML which takes advantage of the large amount of data available to model complex and nonlinear relationships [5]. The hierarchical relationship between ML and DL is illustrated in Figure 2.4.

Both areas, DL, and ML have algorithms to forecast time series. However, they use different approaches, *i.e.*, ML has a set of algorithms such as regression models, Support Vector Regression, Random Forest, while DL has a set of specialized recurrent neural network architectures that works like the biological neural connections and the memory of the human brain *e.g.*, Long-Short Term Memory (LSTM), Gated recurrent units (GRUs), etc [39, 22]. Each model has its own characteristics and depends on the type of problem faced. In the literature it has been shown that deep learning models have reached the state of the art in the prediction of cases of influenza [79, 81, 82], which is also an endemic disease. To translate these results into the forecast of Dengue cases, the characteristics of the problem must be analyzed.

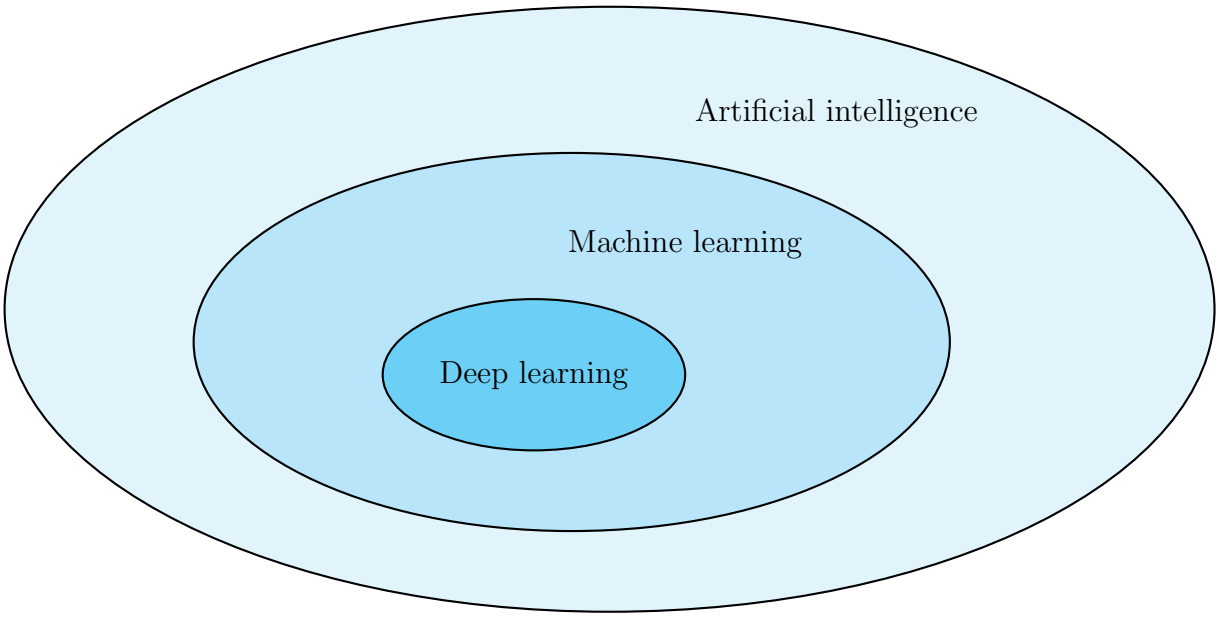


Figure 2.4: Representation of the hierarchical relationship between machine and deep learning techniques.

2.2.2.1 Support Vector Regression (SVR)

Support vector machine (SVM) analysis is a popular ML tool for classification and regression, identified in 1992 [74]. The SVM for regression, called SVR, is a supervised learning approach. Lets suppose that the input data a set of independent variables $(x_1, \cdots, x_l)$ and the output another set of dependent variables $(y_1, \cdots, y_l)$ *e.g.*, Dengue incidence. The method of SVR can solve regression problems such as time series [13]. Smola *et al.* [66] formulate the problem as a convex optimization problem, the objective is to minimize the coefficients, specifically, the l2-norm of the coefficient vector w. The error term is handled in constraints, where the absolute error is set less than or equal to a specified margin, called the maximum error ϵ. Also for any value that falls outside of ϵ, its deviation from the margin can be denoted as ξ, this value is also is wanted to minimize. Then, the SVR is trained by solving:

$$\begin{aligned} \min_{w} \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{\ell}(\xi + \xi_i^*) \\ \text{s.t.} \quad & y_i - \langle w, x_i\rangle - b \leq \epsilon + \xi_i \\ & \langle w, x_i\rangle + b - y_i \leq \epsilon + \xi_i^* \\ & \xi_i, \xi_i^* \qquad\qquad \geq 0 \end{aligned} \tag{2.9}$$

where $\xi = (\xi_1, \xi_1^*, \ldots, \xi_\ell, \xi_\ell^*)$ is a slack variable, $C > 0$ is a penalty parameter and ϵ is a free parameter that serves as a threshold. The inner product plus the intercept, $\langle w, x_i\rangle + b$ is the prediction. All predictions must be within a $\pm$ ϵ range. SVR uses the following parameters:

- *Kernel.* The kernel helps to find a hyperplane when the dimension of the data increases, allowing to move to a higher dimensional space. Some regression problems cannot be adequately described using a linear model. In those cases, the dual Lagrange formulation makes it possible to extend the previously de-

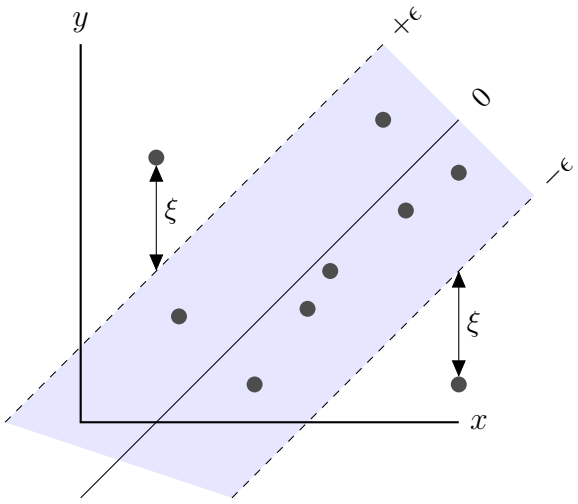


Figure 2.5: Scheme of the SVR, adapted from [66].

scribed technique to non-linear functions. Obtain a nonlinear SVR regression model by replacing the scalar product $\langle x_1, x_2 \rangle$ with a nonlinear kernel function $G(x_1, x_2) =< \phi(x_1), \phi(x_2) >$, where $\phi(x)$ is a transformation that assigns x to a high-dimensional space.

- *Hyperplane.* It is the line (or plane in higher dimensions) that separates the data sets in classes, in SVR, it is the space on which the forecasts will be made.
- *Decision Boundary.* It can be seen as a boundary space that can be seen as a threshold.

The objective is to find a function that minimizes errors, basically SVR looks for the values that are within the boundary decisions, *i.e.*, accept the values that are within $\Delta\epsilon$ of the hyperplane reference as shown in Figure 2.5.

2.2.2.2 Least Angle Regression with LARS LASSO (LR)

In machine learning algorithms, the learning process consists of finding the coefficients (model) by minimizing a cost function. Least Absolute Shrinkage and Selection Operator (LASSO) regularization consists of adding a penalty to the cost function. This penalty produces simpler models that generalize better. Least Angle Regression finds the attribute which is most highly correlated to the target value. Efron *et al.* [18] proposed a variation of the Least Angle Regression (LARS) algorithm with LASSO regularization, which they called *LASSO regression*, this algorithm is described in 1.

Suppose that a cost function J is defined considering the mean squared error as follow:

$$J = \frac{1}{M} \sum_{i=1}^{M} (y_i - \hat{y}_i)^2, \tag{2.10}$$

where M is the number of observations, and y_i and $\hat{y}_i$ are the observed and estimated values at ith observation with n predictors $x_{1i}, x_{2i}, x_{3i}, \ldots, x_{ni}$ respectively.

A LASSO regularization coefficient C can be added to measure of model complexity. It can be written as follow:

$$C = \frac{1}{N}\sum_{j=1}^{N}|w_j|, \tag{2.11}$$

where N is the number of coefficients of the model and w the coefficient vector. This regularization can be introduced to the cost function J by adding a constant α as follow:

$$J = \frac{1}{M}\sum_{i=1}^{M}(y_i - \hat{y}_i)^2 + \alpha\frac{1}{N}\sum_{j=1}^{N}|w_j|, \tag{2.12}$$

where α indicates how important is the regularization, *i.e.*, the simplicity in relation to its performance. This regularization works better with simple models, it also has the ability to discard predictors if the coefficients goes to 0.

Instead of giving a vector result, the LARS solution consists of a curve denoting the solution for each value of the LASSO of the parameter vector.

Algorithm 1: LARS

start with all variables equal to 0 in the model
do
  find a predictor x_i most correlate with the target y_i. The variable most correlated is the one that makes the smallest angle with the target via LASSO, hence the name
  move in the direction of this variable until other variable x_{i+1} it is equally correlated
  keep moving in a direction such that the rest remain equally correlated with x_i and x_{i+1}, until some variable x_{i+2} it also correlates with the residual
  if *coefficient hits* 0 **then**
    drops its variable
  end
while *maximum of variables not reached*

2.2.2.3 Random Forest (RF)

Given a set of data, algorithms are made based on diagrams of logical conditions, very similar to flowcharts, which are used to represent and categorize a series of conditions that happen successively, to solve a problem, these algorithms are called decision trees. In 2001, Breiman describes the use in regressions of the *Random Forest* [11] proposed by Ho [29]. Random Forest is a tree-based algorithm, that basically consists of combining different predictions from several decision trees [41]. The figure 2.6 shows the structure of random forest, the trees are seen to run in parallel.

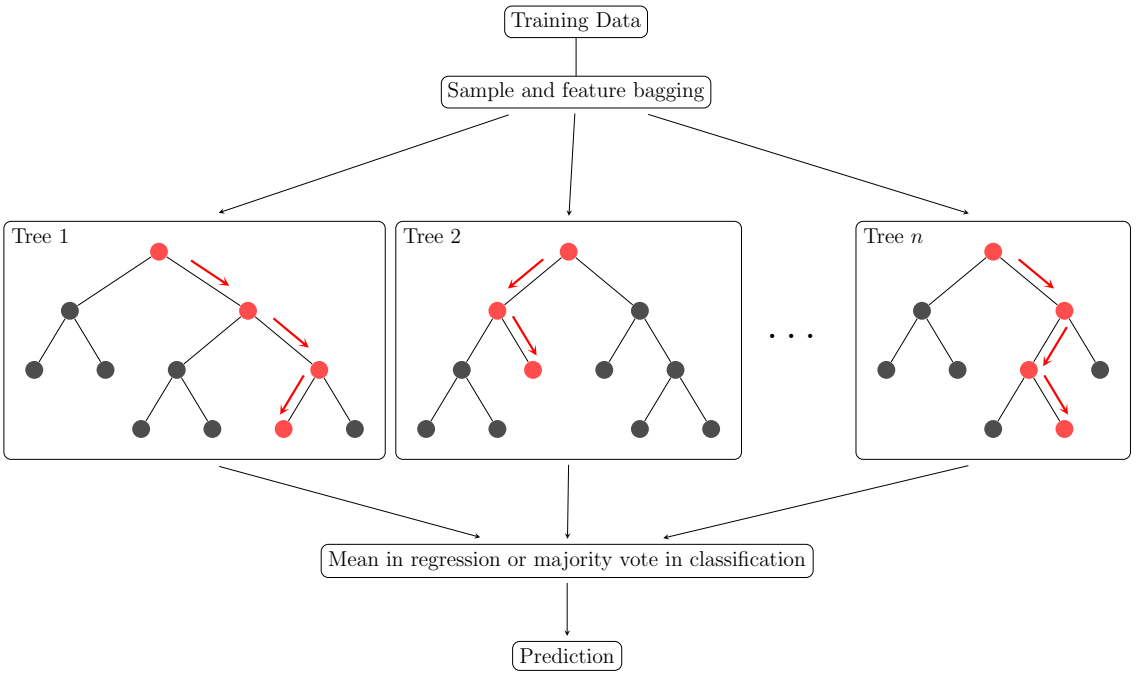


Figure 2.6: Structure of Random Forest

Random forest works like this, first k samples are chosen from the training set, then a decision tree associated with those k samples is generated. This procedure is repeated for the n decision trees that are used, once the result of each tree is had and the mean of the results is found, this is taken as the output of the random forest. It is worth mentioning that Random forest is also a classification technique and that instead of returning the mean of the results obtained, the result with the majority of votes is returned.

2.2.2.4 Long-short Term Memory (LSTM)

Recurrent neural networks (RNN) are a type of network that integrates feedback cells, which allows the network to maintain information from a certain amount of training periods, this type of network has good results in the area of classification of imaging and object detection [60].

Long-short term memory is a RNN specifically designed to forecast time series, thanks to its memory cells which preserves long and short dependencies [30]. These 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, a LSTM network uses 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. Figure 2.7 shows an LSTM cell architecture.

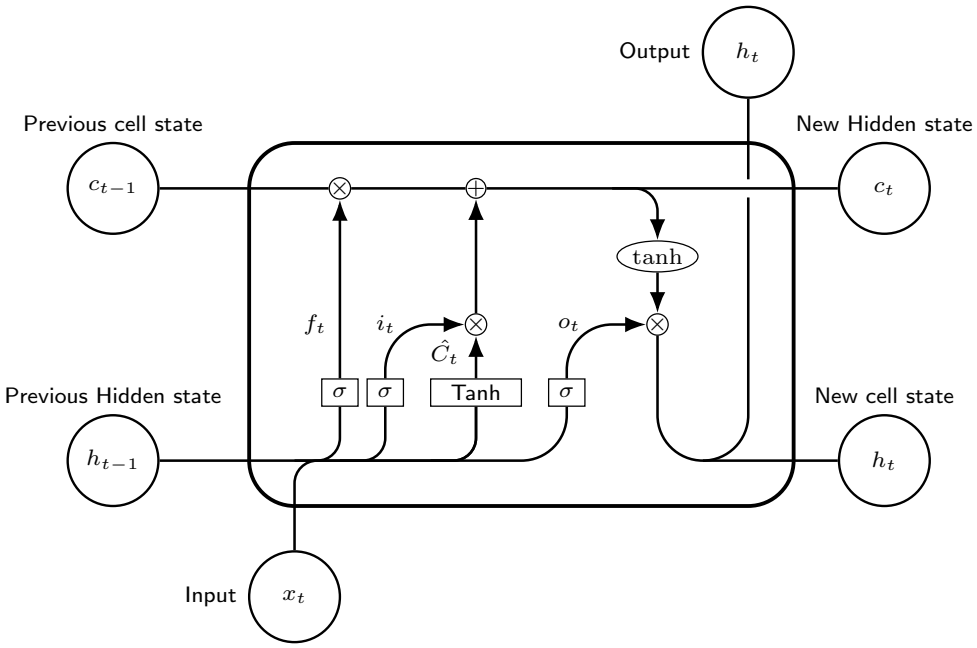


Figure 2.7: LSTM cell architecture showing the forget, input, and output gates and, the hidden state. Adapted from [53].

States of LSTM cells are computed as follows [6]:

$$i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i), \tag{2.13}$$
$$f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f), \tag{2.14}$$
$$o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o), \tag{2.15}$$
$$\hat{C}_t = \tanh(W_C h_{t-1} + U_C x_t + b_C), \tag{2.16}$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \hat{C}_t, \tag{2.17}$$
$$h_t = o_t \odot \tanh(C_t), \tag{2.18}$$

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 (2.13)-(2.15) 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 as 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 (2.17). The calculation of new candidate values $\hat{C}$ of memory cell and output of current LSTM block h_t uses hyperbolic tangent function as in equations

(2.16) and (2.18). 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. An LSTM neural network consists of a set of connected LSTM cells.

During the training procedure LSTM cells 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 most common loss function is used, *i.e.*, the mean squared error. The smaller value of the loss function means that the prediction of our model is improving. To minimize the error of the loss function an optimization algorithm is used, *i.e.*, Adam algorithm.

2.2.3 Training time series models

The models mentioned above (SVR, LASSO LARS, RF, LSTM) have shown to have good performance with time series and are traditionally used for these problems, however each technique has its particularity and must be taken into account, such as:

- In RF, a large number of trees are necessary to get stable estimates of variable importance and proximity [41].
- LR is a good option if is suspected that some predictors of the model are unnecessary, since they can be discarded if they reach the value 0 [18].
- SVR is not recommended for large or extremely noisy datasets, since the parameter ϵ will be difficult to choose [72].
- SVR is effective in cases where the number of dimensions is greater than the number of samples.
- When there are highly correlated features, LR may randomly select one of them of part of them [18].
- The number of trees necessary for good performance grows with the number of predictors in RF [41].
- LSTM requires large amounts of data to avoid overfitting [70].

To perform a regression in time series it must be taken into account that time series have the particularity that they are autocorrelated, *i.e.*, an observation x_t is correlated with an observation x_{t-h} where t is the index of an observation and h is an integer such that $h \leq t$. Therefore, the historical data of the series should be used to forecast subsequent data of the same series, in addition to the meteorological variables that are considered. This is done through sliding windows, where one part of the series is taken to forecast the next part in the same series. Figure 2.8 describes the sliding window procedure.

To validate the performance of the time series forecasting models, the data is split in train-test sets, for network training and test respectively. During the training, observed data is the input and, during the test, the performance is evaluated. For this problem the input is the incidence of Dengue cases and some meteorological variables whose selection process is described in 2.4.

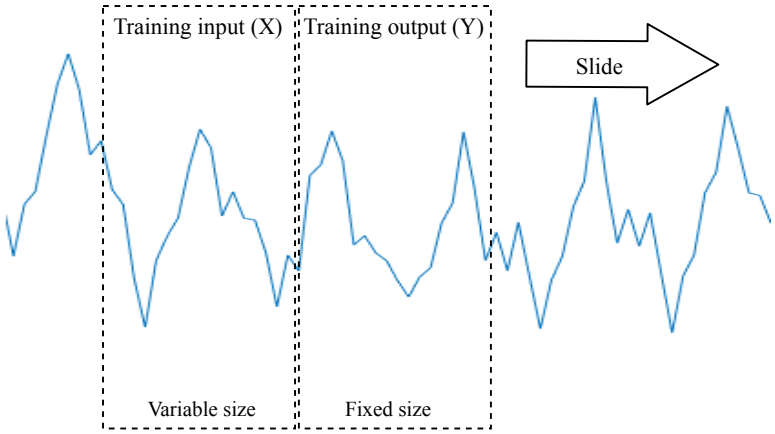


Figure 2.8: Time Series split for model validation, adapted from [6].

2.3 Dengue Fever Cases Forecast Evaluation

In order to evaluate each model predictions, a way to measure the performance of each model must be defined.

There are several metrics that are traditionally used to measure time series forecasting error, these are: Mean Absolute Error (MAE), Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE) and Symmetric Mean Absolute Percentage Error (SMAPE). Since forecast performance is evaluated for time series with the same scale and the data preprocessing procedures were performed, it is reasonable to choose MAE, MSE or RMSE according to Shcherbakov *et al.* [61]. In this work, the $RMSE$ was used to measure the performance of each model because it considers the distance between the predicted and observed values. In this work, the following definition for the root mean squared error is considered.

Definition 1. *The root mean squared error ($RMSE$) is defined as follows*

$$RMSE := \sqrt{\frac{1}{n}\sum_{i=0}^{n}(Y_t - \hat{Y}_t)^2}, \tag{2.19}$$

where 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.

2.4 Dengue Cases in Paraguay

Dengue fever cases (c) from January 2009 to December 2013, organized in 217 cities of 17 states in Paraguay, and population (p) of each city, was provided by the COMIDENCO project [27]. The COMIDENCO dataset has data individually, each record corresponds to a case and has the status of the patient, hospital, city, department, whether or not they traveled and blood count data. COMIDENCO team curated data to develop epidemiological models. The meteorological data were obtained from weather stations that are distributed throughout the country

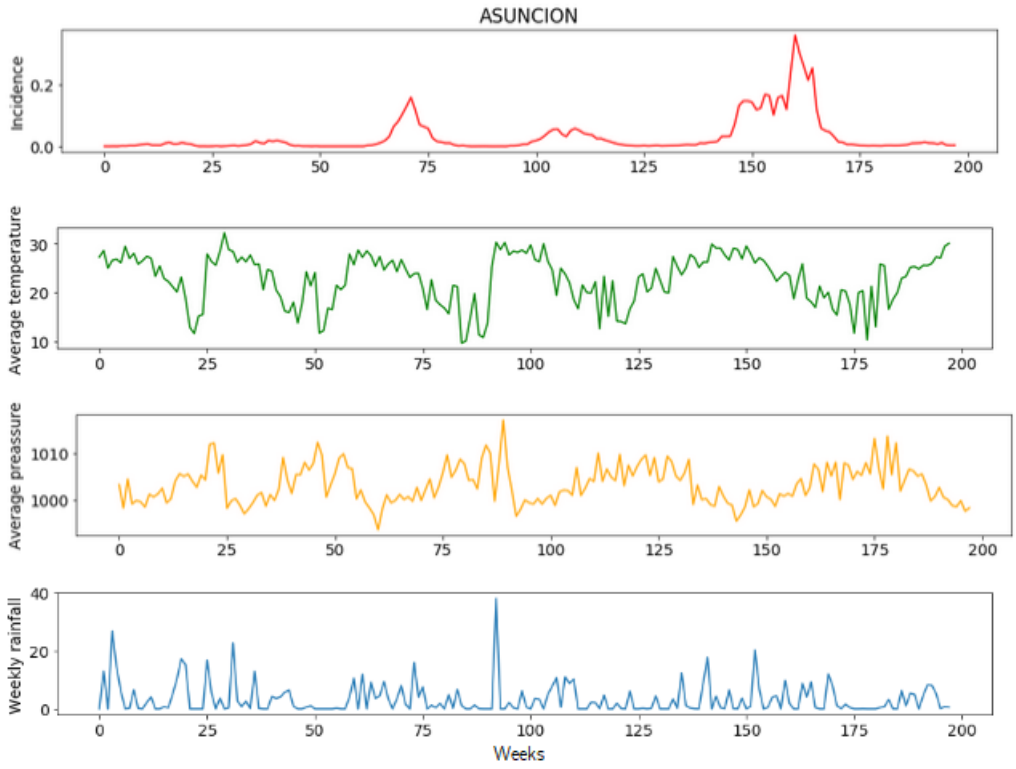


Figure 2.9: Time series available for each city in the country, in this sample: Asunción. Values are Incidence (Icd), Average Temperature (T^a), Average Atmospheric Pressure (Pr^a) and Weekly Rainfall (R^w)

[17]. Meteorological data included daily reports of minimum, average and maximum temperature, minimum, average and maximum atmospheric pressure, rainfall, maximum, average and minimum wind speed and cloudiness.

2.4.1 Sample Selection

As there are 217 cities, a representative sample of the cities was made. The cities time series where divided in three equal-sized groups according to their population, because cities with less population usually have fewer cases. The first group (denoted as Group 1, corresponds to most populated cities) is composed by taking the population from the 66th to 100th percentile, the second group (denoted as Group 2, corresponds to intermediate populated cities) from the 33th to 66th percentile, and the last group (denoted as Group 3, corresponds to less populated cities) from the 0th to 33th percentile. Time series of five cities from each group are randomly selected. The selected cities where:

1. Group 1
   (a) San Lorenzo
   (b) Capiatá
   (c) Caaguazú
   (d) Areguá

(e) Salto del Guairá

2. Group 2

(a) Choré
(b) Juan León Mallorquín
(c) Santa Rosa del Aguaray
(d) Quiindy
(e) Eusebio Ayala

3. Group 3

(a) Encarnación
(b) San Pedro del Ycuamandiyú
(c) Capitán Miranda
(d) Yhú
(e) Santa Rita.

These samples will be used for all experiments here in after.

2.4.2 Data Preprocessing

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.

Definition 2. *The incidence of Dengue fever in a city, denoted as Icd, is given by*

$$Icd := 100\frac{c}{p}, \tag{2.20}$$

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

Incidence is the number of new cases of a disease that occurs over a specific period of time, such as a week. Incidence shows the probability that a person in a certain population is affected by that disease. As it is a percentage measure, it was decided to work with the incidence rather than the number of cases, as a normalization.

According to [50], the features selected are mean temperature, atmospheric pressure, and rainfall. To obtain an effective value for each group, the features are combined weighting them by their corresponding city population.

Finally, there are four variables used in our model, the incidence and three meteorological variables. Each time series has 265 records, since there are 265 weeks in the period 2009-2013. The details of these variables can be seen in Table 2.1, an excerpt of the time series can be seen in Figure 2.9.

Table 2.1: Feature parameters for the LSTM model

Parameter	**Symbol**	**Unit**
Weekly incidence of Dengue cases	Icd	%
Average temperature	T^a	C^o
Average atmospheric pressure	Pr^a	mmHg
Weekly rainfall	R^w	mml

2.5 Benchmark Model Selection

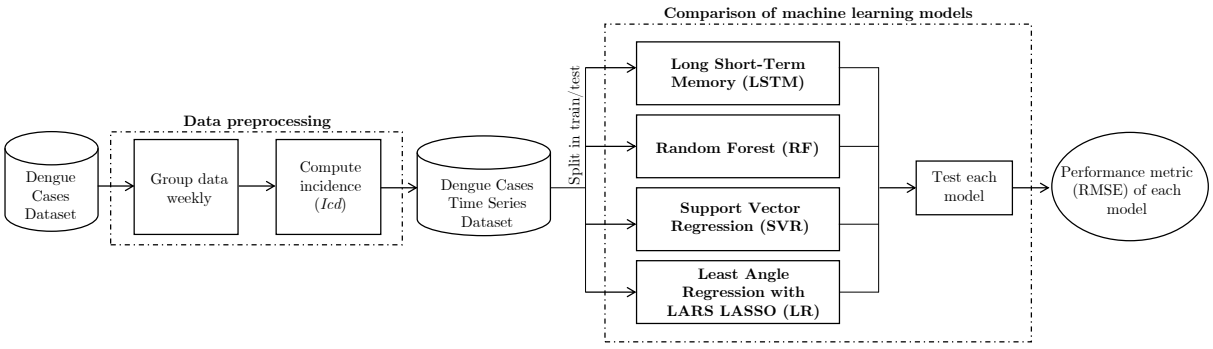


Figure 2.10: Summarized workflow of the benchmark model selection process. The data is grouped weekly and the Dengue cases incidence (Icd) is computed as described in (2.20), then, the data is trained and tested with machine learning techniques (LSTM, RF, SVR and LR) to determine which one has the best performance in order to be selected as benchmark model.

In order to select a machine learning model as a benchmark for the experiments to be performed, this section presents a comparison of a LSTM model with traditional machine learning models known for their ability to forecast time series data. The comparison ranks each approach according to certain performance criteria *e.g.*, RMSE. Random Forest (RF), LARS LASSO Regression (LR), Support Vector Regression (SVR), and LSTM are used. Figure 2.10 shows the workflow of the selection process.

The parameters were defined as follows: for RF number of trees = 1,000 [32], for LR $\alpha = 1,662e-6$ was used [38], for SVR we use $\epsilon = 0.2$ and Radial Basis Function (RBF) kernel [31] and for LSTM 1 layer with 8 neurons, the default values from the deep learning library were used (learning rate = 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$) [64].

Table 2.2 shows that in all Groups (most, intermediate, and less populated cities), the LSTM model have the lowest RMSE on 9 of 15 cities. RF seems to be the second model with good results. While, LR presents better results only in 3 cities of 15. This is why the LSTM model is selected as the benchmark model.

When the data is stationary, LSTM models usually have better performance. In order to check if all time series are stationary, the Augmented Dickey-Fuller test was conducted (see Appendix B). For each series, the p-value is less 0.05, therefore, the time series are stationary and easier to generalize [37]. To validate the performance of LSTM model, the data was splitted in train-test sets, $70\% - 30\%$ of of each time series that was used for network training and test respectively. Therefore, the train set consists in 185 of 265 records. To describe the input vector associated to 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.

Table 2.2: Comparison of models performance using the RMSE score. Values in bold indicate the best values.

Group	City	LSTM	RF	LR	SVR
Group 1	San Lorenzo	0.1360	0.1424	**0.0565**	0.1477
	Capiatá	0.1334	**0.0219**	0.0993	0.2318
	Caaguazú	**0.0266**	0.0386	0.0395	0.0381
	Areguá	**0.1201**	0.2204	0.2274	0.2261
	Salto del Guairá	**0.0259**	0.0309	0.0273	0.0342
Group 2	Chore	0.0075	**0.0065**	0.0071	0.0183
	Juan León Mallorquin	**0.0096**	0.0107	0.0124	0.0115
	Santa Rosa del Aguaray	0.0060	**0.0055**	0.0068	0.0101
	Quiindy	**0.0095**	0.0113	0.0090	0.0236
	Eusebio Ayala	**0.0101**	0.0105	0.0117	0.0151
Group 3	Encarnación	**0.0028**	0.0035	0.0044	0.0037
	San Pedro Del Ycuamandijú	**0.0033**	**0.0033**	**0.0033**	**0.0033**
	Capitán Miranda	**0.0031**	0.0031	0.0033	0.0043
	Yhú	0.0017	**0.0015**	0.0016	0.0025
	Santa Rita	0.0021	**0.0017**	**0.0017**	0.0081
	Average RMSE	**0.0332**	0.0342	0.0340	0.0519

2.6 Discussion

Fitting models to the time series of many cities is a challenging task because some cities display low and high incidences, showing heterogeneous behavior. That is why it is difficult to find a model that generalizes all cities, and particularly for machine learning techniques, the length of the series may not be enough for the models to fit correctly. As each model has its particularity, the available data must be analyzed. Particularly in the problem of forecasting Dengue fever cases, two main drawbacks are identified:

1. *Multi-spatial forecasting.* Models are needed that can generalize a geographic space, *i.e.*, a country and its cities, or that can be generalized to a large group of cities.

2. *Lack of data.* Since time series are closely related to time, it is difficult or sometimes impossible to collect data on past epidemics. Furthermore, the Dengue fever epidemic is relatively recent, as explained in Section 1, and the time series are not very long. Lack of data leads to a model overfitting.

The first drawback can be addressed by grouping the data, searching for those that behave in a similar way, that option must be analyzed, unsupervised clustering techniques must be studied, and the most suitable for time series determined. This approach will be addressed in Chapter 3.

The second drawback, if it is not possible to collect more data, can be addressed by generating synthetic data that represent the observed data, for that, a model must first be found that represents the behavior of the series. In the case of deep learning networks, they are less prone to overfitting the larger the database [70]. A common procedure in the image area is to increase data by generating new images from

the existing ones and applying small transformations to them, such as cropping, rotating, changing the contrast or adding noise [75]. A similar approach can be applied to time series, and also applying statistical techniques such as Bayesian inference that allow the generation of new series that maintain their characteristics. This approach will be addressed in the Chapter 4.

Chapter 3

TIME SERIES CLUSTERING

Clustering is an unsupervised machine learning task aimed to classify in groups a big amount of data when there is not 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 [40].

3.1 Clustering Algorithms

Clustering techniques are classified according to the way they perform the partitions, thus having centroid-based, connectivity-based, and density-based, the most representative being the k-means algorithm, hierarchical clustering and DBScan respectively [40].

K-means [3] 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 2 shows the k-means partitioning process.

Algorithm 2: 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* [49], clusters are generated as the name implies, hierarchically. 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 3 shows the hierarchical clustering process.

Algorithm 3: 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 group. 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 3.1.

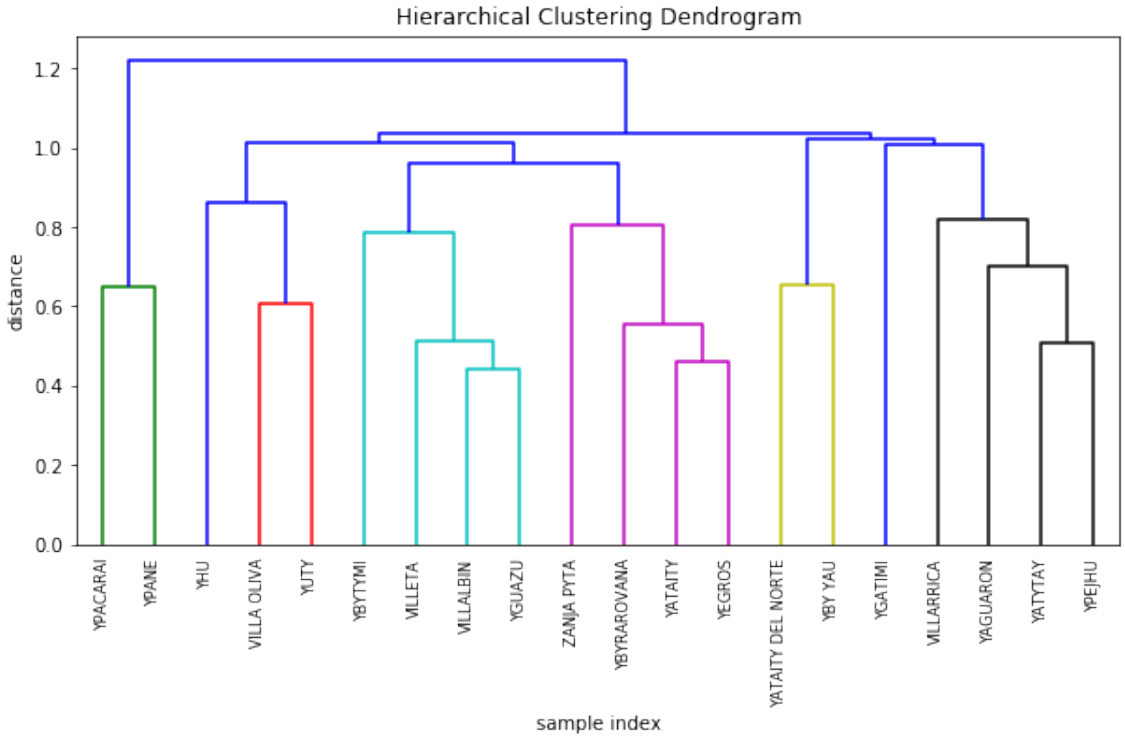


Figure 3.1: Dendrogram formed by applying hierarchical clustering to a sample of 20 cities from the COMIDENCO dataset [27], five groups formed can be observed, the lines represent the distances between elements and the lines of the same color represent those that are in the same cluster.

In *DBScan* [19], 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 4 describes the DBScan.

Algorithm 4: 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.

3.1.1 Number of Clusters

Unless the number of clusters required is known in advance, determining the optimal number of clusters (k) is a complex task. This is a frequent problem in data clustering, since it is an input parameter that is needed for some clustering algorithms, and there is no certain answer, however, there are techniques that help to infer the optimal number of groups, such as:

- *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, which would look like the "elbow" of the graph, the calculated error is the within-cluster sum of squares (WSS), *i.e.*, the sum of the squared deviations from each observation and the cluster centroid. Algorithm 5 shows

the procedure to find the optimal number of clusters using the elbow method.

Algorithm 5: Elbow method

Data: time series to cluster, upper limit of cluster size (n)
Result: optimal number of clusters
for $i = 1$ **to** n **do**
    compute the clustering algorithm with i clusters and calculate the WWS
end
plot the curve of WSS according to the number of clusters k
select the value that is on the curve (elbow) on the graph as the appropriate number of groups

The reason the elbow of the plotted curve is considered as the optimal number of clusters is because from the elbow the cost of the clusters does not contribute much, then the groupings after the elbow do not have a significant improvement in the separation of the components. Figure 3.2 shows a plot of the elbow method.

- *Silhouette score.* Measures how well an observation is clustered by estimating the mean distance between clusters. The silhouette value is a measure of how similar an object is to its own cluster compared to other clusters. It ranges from -1 to +1, where a high value indicates that the object is well matched with its own cluster and poorly matched with neighboring clusters. If most of the objects have a high value, then the cluster configuration is appropriate. If many points have a low or negative value, then the cluster configuration may have too many or too few clusters. The silhouette can be calculated with any distance metric. Algorithm 6 shows the procedure to find the optimal number of clusters using the silhouette score.

Algorithm 6: Silhouette score

Data: clusters, upper limit of cluster size (n)
Result: optimal number of clusters
for $i = 1$ **to** n **do**
    compute the clustering algorithm with i clusters and calculate the silhouette score
end
plot the curve of silhouette score according to the number of clusters k
select the maximum value on the curve of the graph as the appropriate number of groups

It is also used to measure the quality of clustering, and it is the metric used in the experiments of this work, the details are described in Section 3.3, the formula that details it is shown in equation 3.4.

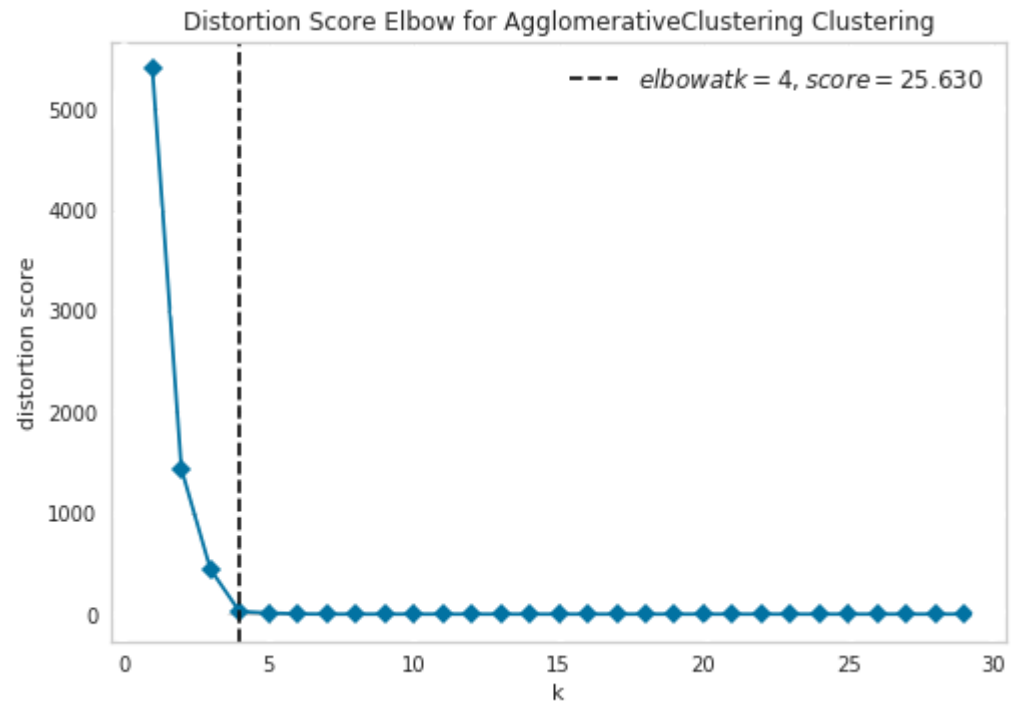


Figure 3.2: Elbow method plot for InfoDengue dataset [16]. It is seen that the selected value is the "elbow" of the curve, in this case 4. The number of clusters ranged from 2 to 30.

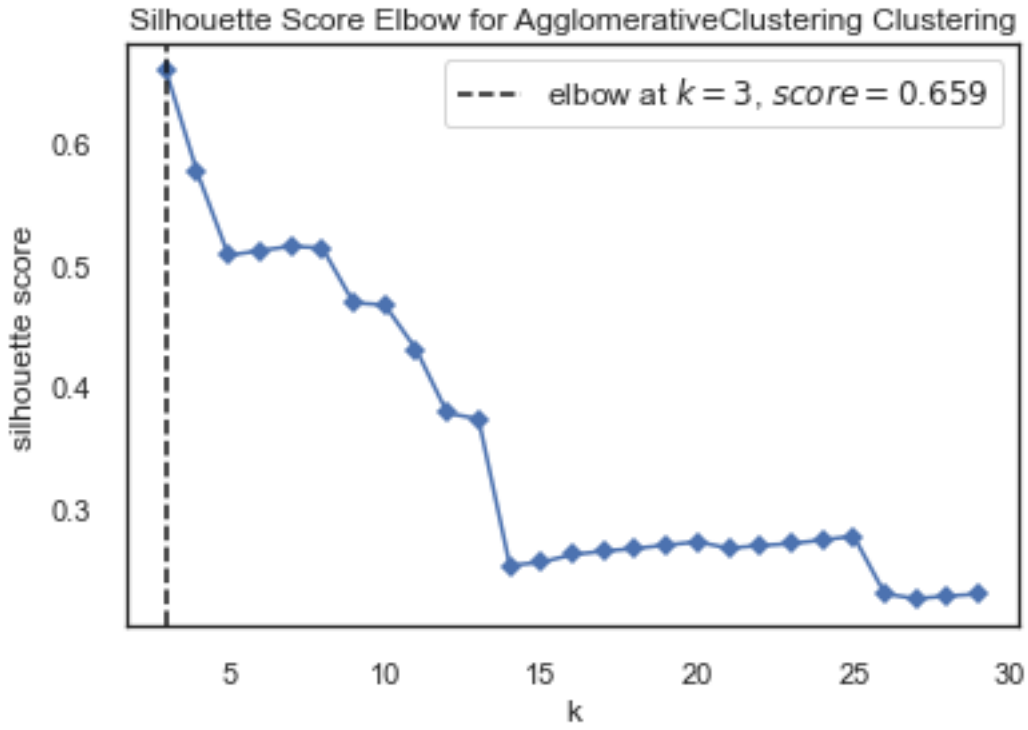


Figure 3.3: Silhouette score plot for InfoDengue dataset [16]. The maximum value of the silhouette score indicates the optimal number of clusters, in this case 3. The number of clusters ranged from 2 to 30.

Once the number of clusters to be formed is selected, the way to measure the distance between the elements to be grouped must be chosen, for this dissimilarity metric must be determined.

3.1.2 Dissimilarity Metrics

The distance is used to determine how close a pair of observations are, the closest observations are more similar and therefore may belong to the same cluster. The metrics used to measure the distance between two observations are called *dissimilarity metrics*. All clustering algorithms use dissimilarity metrics. Among all the metrics to measure dissimilarity for time series, Euclidean distance and dynamic time warping are the most cited in the literature according to Giusti *et al.* [26] , and correlation-based measures such as Spearman correlation are also used [34]. All this metrics are known for their ability to measure time series, details are presented below

- *Euclidean.* The distance between a pairwise y and $\hat{y}$ is defined as

$$d(y, \hat{y}) = \sqrt{\sum_{t=1}^{N} (y_t - \hat{y}_t)^2}, \tag{3.1}$$

  where N is the length of the time series, and y_t and $\hat{y}_t$ are the t-th element of time series Y and $\hat{Y}$, respectively. With this metric, distances are compared in timesteps at the same time location. Y and $\hat{Y}$ must be have the same length.

- *Correlation.* Is defined as

$$d(y, \hat{y}) = 1 - \frac{(y - \bar{y}) \cdot (\hat{y} - \bar{\hat{y}})}{||(y - \bar{y})||_2 ||(\hat{y} - \bar{\hat{y}})||_2}, \tag{3.2}$$

  where $\bar{y}$ is the mean of the elements of time series y, $\bar{\hat{y}}$ is the mean of the elements of time series $\hat{y}$, and $y \cdot \hat{y}$ is the dot product of y and $\hat{y}$.

- *Spearman correlation.* can be seen as a square of Euclidean distance between two rank vectors, which is written as follows

$$d(y, \hat{y}) = 1 - \frac{6 \sum_i rank(y_i, \hat{y}_i)^2}{N(N^2 - 1)} \tag{3.3}$$

  To calculate the Spearman rank correlation, each data value is replaced by their rank if the data in each vector would be ordered by their value. The rank is the interval between the maximum value and the minimum value in a pairwise. Then $rank(y_i)$ and $rank(\hat{y}_i)$ are the ranks of time series y and $\hat{y}$ respectively. N is the length of the time series.

- *Dynamic time warping.* It is an algorithm designed to measure the difference between two time series but not necessarily by measuring each point at the

same timestep. Algorithm 7 describes how the Dynamic time warping works.

Algorithm 7: Dynamic time warping

Data: time series pairwise (y and $\hat{y}$)
Result: distance between time series
divide the series into n equal points
for $i = 1$ **to** n **do**
    compute the euclidean distance, as in equation 3.1 between the i point in the y series and every point in the $\hat{y}$ series
    store the minimum distance calculated
    for $i = 1$ **to** n **do**
        compute the Euclidean distance, as in equation 3.1 between the i point in the $\hat{y}$ series and every point in the y series
        store the minimum distance calculated
    end
end
add up all the minimum distances that were stored and this is the measure of similarity between the two series

When time series are too long or when they have missing data, they tend to become intractable for clustering algorithms with certain dissimilarity metrics, one approach that addresses these problems is feature-based clustering.

3.2 Feature-based Algorithms

Feature-based clustering approach involves using the most significant features from each time series and performing clustering based on those features. To obtain features, Feature selection and feature extraction can be used. Feature selection aims to reduce the number of feature sets available to a subset of relevant features that minimize redundancy and increase the relevance of the features. The objective of feature extraction is to obtain characteristics from the series themselves that are relevant. Although Feature selection has been indicated as the most appropriate for clustering [1], it must be had a set of observations of characteristics that are known to be related. In the context of Dengue cases, this information is not available. Therefore, feature extraction has become the simplest technique to perform feature-based clustering.

Nanopoulus *et al.* [51] have proposed to extract some basic statistical characteristics of the time series, obtaining robust results clustering, Mörchen *et al.* [48] has proposed to use the Discrete Wavelet Transform and the Discrete Fourier Transform and obtained an improvement in terms of computational cost, and Hyndman *et al.* [33] have made a compilation of the most relevant features to extract for time series, the features proposed by Hyndman *et al.* are:

1. Mean of time series.
2. Variance in time series.
3. First order of autocorrelation.
4. Trend. Is the value of the trend component of STL decomposition [15].

5. Linearity. Augmented Dickey-Fuller test for linearity result.
6. Curvature. Computed based on the coefficients of an orthogonal quadratic regression.
7. Seasons. Is the value of the trend component of STL decomposition [15].
8. Number of peaks. Also called spikes, are notoriously high values in the series, above of the mean.
9. Spectral entropy.
10. Changing variance in remainder. Divide the series in blocks and the variances of each block are computed and the variance of the variances across blocks measures is the changing variance in remainder.
11. Level of shift using rolling window. Is the maximum difference in mean between consecutive blocks.
12. Variance change. Is the maximum difference in variance between consecutive blocks.
13. Flat spots using discretization. Are calculated by dividing the time series into ten equal-sized intervals, and computing the maximum length within any single interval.
14. Number of crossing points. It is the number of points that cross the mean line.
15. Kullback-Leiber score. Is the maximum difference in Kullback-Leiber divergence using kernel density estimation between consecutive blocks.
16. Index of the maximun Kullback-Leiber score.

This work uses Hyndman *et al.* proposed features followed by a clustering process. After this, the quality of the groups formed must be evaluated, for that purpose there are evaluation metrics.

3.3 Evaluation Metrics

There are two ways to validate clustering with internal techniques, also called internal or unsupervised validation and external or supervised validation.

The internal validation is done on the results of the cluster, without having access to other external information, *i.e.*, the true labels of data. They are based on cohesion and separation measures. Cohesion evaluates how closely the elements of the same cluster are to each other, while separation measures quantify the level of separation between clusters.

External validation techniques are based on external information, such as the labels of the training data. They are related to supervised learning, these techniques are based on comparing the expected values with those obtained.

Due to the nature of the problem of clustering time series of Dengue data, the techniques to be used to evaluate the quality of clustering were internal validation metrics, such as Silhouette Score or Calinski Harabasz index. The Calinski-Harabasz

index is generally higher for convex clusters, like those obtained through DBSCAN [44]. So it is not suitable for measuring results between density-based techniques and hierarchical or partition-based techniques. Silhouette score is used to analyze the separation distance between the resulting clusters. It is especially useful if there is no prior knowledge of what is the true label for each object, which is the most common situation in real applications. In this work, a pair of clusters A and B are considered, the silhouette score $s(i)$ is computed as [52]:

$$s(i) = \frac{b(i) - a(i)}{\max(b(i), a(i))}. \tag{3.4}$$

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 mean distance associated to 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, smaller the value of $a(i)$, 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 (3.4) provides a metric between the considered clusters A and B. In this case, smaller the value of $s(i)$, more the proximity of the clusters A and B. From expression (3.4), notice that $s(i)$ lies in the range of $[-1, 1]$.

3.4 Clustering of Dengue Cases Time Series

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 amount of necessary adjustments per city.

2. For deep learning models, the greater the amount of data, the overfitting is avoided and therefore the network performance improves.

Since there is no clue as to which class the series should belong to, there are several uncertainties when performing clustering, such as: determining the number of clusters, defining the metrics of dissimilarity, and if they are feature-based, determine which are the most relevant features. There are diverse options proposed to deal with these uncertainties, and they depend on the approach of clustering that is performed.

Liao [40], distinguish three main approaches to time series clustering:

1. Distance-based, directly with distances on raw data points.

2. Feature-based, indirectly with features extracted from the raw data.

3. Model-based, indirectly with models built from the raw data.

The performance of Distance-based clustering approaches depends greatly on the particular distance metric used because the time series may have noise, different dynamics, different scales, etc. Feature-based performance depends on the correct selection of the features that describe them. Those Model-based depend on the selection of the model or the correct construction of the model. As there is no model that describes the time series of Dengue cases, for this work, only the distance-based and feature-based approaches were used.

When is required to model Dengue cases in several cities, it can be seen that some have similar behaviors, this is why it is proposed to group them. To apply time series clustering, definitions of several parameters are required, but these vary according to each problem, therefore prior experiments must be carried out. This work propose to perform clustering of time series of Dengue cases, for that the Distance-based and Feature-based approaches were tested, since there is no model that can be assumed to generate the time series of Dengue cases, the Model-based approach is not used. The number of clusters is a necessary input parameter for some algorithms, using the elbow method, the number of groups to be formed is calculated. The most appropriate dissimilarity measures are determined through experiments, combining the metrics with different clustering techniques. Finally, to decide which one obtains the best results, the results are validated with an internal evaluation metric (3.4), these clustered results will also be compared with data grouped according to the political division of the country. A summarized workflow of this proposal is shown in Figure 3.4.

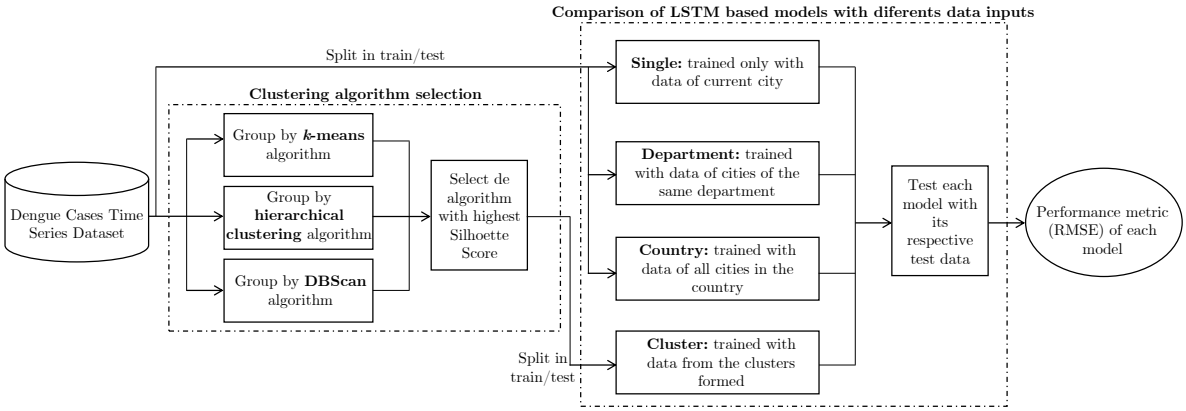


Figure 3.4: Summarized workflow of the experiments for time series clustering. All Dengue cases time series are cluster with different algorithms (k-means, hierarchical clustering and DBScan) then, the algorithm with best silhouette score is selected to be used to generate the input for the *Cluster* model and its performance is evaluated against *Single*, *Department* and *Counry* models. All models are LSTM based as described in 2.5.

3.5 Experimental Results

A previous study to analyze which technique would be the most appropriate for this case study was carried out. For this reason, several experiments comparing the considered clustering techniques [40] performance were run, considering raw-based

and feature-based approach, using the elbow method to determine the number of clusters to form, applying different metrics for clustering and evaluating them with silhouette score. The clustering methods applied were k-means, hierarchical, and Dbscan. For the raw-based, a set of distance metrics were considered[1]. In the case of feature-based clustering, a set of features were also considered according to Hyndman et. al [33] (See Table 3.1).

Table 3.1: Features extracted from a time series

Feature	**Description**
Mean	Mean
Var	Variance
ACF1	First order of autocorrelation
Trend	Strength of trend
Linearity	Strength of linearity
Curvature	Strength of curvature
Season	Strength of seasonality
Peak	Number of peaks
Entropy	Spectral entropy
Lumpiness	Changing variance in remainder
Lshif	Level of shift using rolling window
Vchange	Variance change
Fspots	Flat spots using discretization
Cpoints	Number of crossing points
Klscore	Kullback-Leiber score
ChangeIdx	Index of the maximum KLscore

Table 3.2 presents the Silhouette scores (See equation 3.4) in order to compare the raw-based and feature-based clustering. The first and second column scores indicate that the best results are obtained with the feature-based methods with Spearman correlation. K-means and Hierarchical clustering present similar scores. However, based on [50] results, the method selected was hierarchical clustering. The expanded results of clustering can be seen in Appendix C. Figure 3.5 shows that the groups are not necessarily geographically close.

This work seeks to improve the performance of a deep learning neural network to forecast Dengue cases through clustering. Clustering is carried out in four ways:

1. Considering each city of the country individually.
2. Considering administrative division of the country in departments. (Paraguay has 17 departments or states which group several neighboring cities).
3. Grouping all the series of each city together.
4. Forming groups adopting the best clustering technique.

Based on this, the models considered were *Single* that was trained with data only from the city (for this approach there are 217 models), *Department* was trained with data of the department (for this approach there are 17 models), *Cluster* that used

[1]euclidean distance, correlation, spearman correlation, dynamic time warping

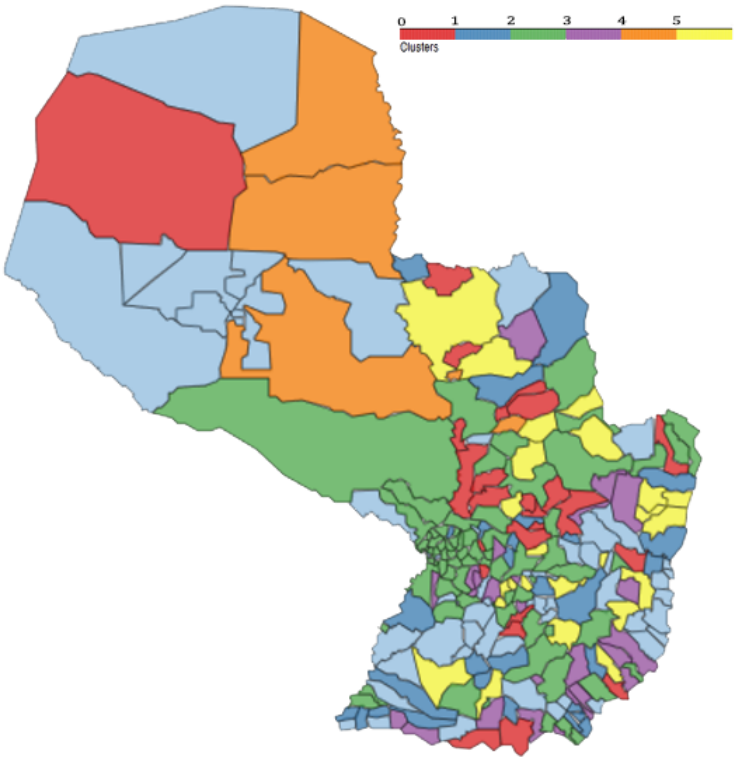


Figure 3.5: Map that represents the Hierarchical clustering with Spearman correlation distance of the cities of Paraguay by color codes. Cities of the same color belong to the same cluster. It can be seen that the clusters are not necessarily geographically contiguous.

Table 3.2: Silhouette score values for clustering methods. Values in bold indicate the best ones on each row.

Clustering method	**Metric**	**Silhouette score**	
		Raw Data	Feature based
K-means	Euclidean distance	0.7059	**0.7118**
	Correlation	**0.1936**	0.0048
	Spearman correlation	0.4805	**0.9953**
	Dynamic time warping	**0.7489**	-0.3017
Hierarchical clustering	Euclidean distance	**0.8387**	0.6871
	Correlation	**0.0161**	0.0016
	Spearman correlation	0.4059	**0.9954**
	Dynamic time warping	**0.7489**	0.6042
DBScan	Euclidean distance	**0.8141**	0.7327
	Correlation	**0.4543**	0.0010
	Spearman correlation	0.0054	**0.9826**
	Dynamic time warping	0.0523	**0.6447**

data from each cluster formed (for this approach there are 6 models), and *Country*

that was adjusted with all data from the country (for this approach there is 1 model). All these models were evaluated at the city level to check their generalization in the forecasts.

Table 3.3: Comparison of each clustered LSTM model using the RMSE. Values in bold are the best ones.

Group	**City**	*Single*	*Department*	*Cluster*	*Country*
Group 1	San Lorenzo	0.1360	0.0689	**0.0510**	0.0689
	Capiatá	0.1334	0.1102	**0.0940**	0.1102
	Caaguazú	0.0266	0.0165	**0.0135**	0.0165
	Areguá	0.1201	0.1091	**0.1003**	0.1091
	Salto del Guairá	0.0259	0.0211	**0.0186**	0.0203
Group 2	Chore	0.0075	0.0065	**0.0063**	**0.0063**
	Juan León Mallorquin	0.0096	0.0091	0.0096	**0.0088**
	Santa Rosa del Aguaray	0.0060	0.0047	**0.0037**	0.0039
	Quiindy	0.0095	**0.0091**	0.0099	0.0092
	Eusebio Ayala	0.0101	0.0114	**0.0095**	0.0097
Group 3	Encarnación	0.0028	0.0028	**0.0026**	0.0029
	San Pedro Del Ycuamandijú	0.0033	0.0031	**0.0027**	0.0028
	Capitán Miranda	**0.0031**	0.0033	**0.0031**	0.0033
	Yhú	0.0017	0.0020	**0.0016**	0.0017
	Santa Rita	0.0021	0.0021	**0.0017**	0.0022
Average RMSE		0.0332	0.0253	**0.0226**	0.0257

The incidence indicates that the largest number of cases are concentrated around the capital and the northeast border cities. The models *Single*, *Department*, *Cluster* and *Country* were trained with data from the city, the department, the cluster to which it belongs, and all data from the country respectively. The *Cluster* model is the one with the lowest RMSE in most cities compared to *Single*, *Department*, and *Country*. Table 3.3 shows the RMSE for the predictions of these cities for the first thirty-five weeks of the year 2013. Dengue incidence predictions for the first 35 weeks of 2013 are shown in Figures 3.6, 3.7 and 3.8.

The average maximum incidence rate is ≈ 0.33. In cities with an incidence rate close to the mean, all models behave similarly well. Only in Group 2, intermediate cities, *Cluster* model narrowly surpassed others models, as seen in Figure 3.7. However, the *Cluster* model has much better performance in cities from Group 1 (see Figure 3.6) and 3 (see Figure 3.8), as is the case of the city of Capiatá, where RMSE improves 14.7% compared to the best performing model, see Table 3.3. The difference is also better in cities with very low incidence, as in the case of Encarnación the improvement in RMSE is 8.7%, in all cities with low incidence the improvement is similar. In the models with incidences far from the average, the *Single* and *Department* models tend to fail almost completely, being improved by up to 62.5% by the *Cluster* model. The *Country* model remains second in most experiments, the times in which it exceeds the *Cluster* model the average improvement is 5%, which is not very significant compared to the rest. In addition, the *Cluster* model outperforms the non-LSTM models that performed better on the benchmark test (see Table 2.2).

The *Country* model underestimates peaks in cities with incidence lower than

average. This can be seen in the case of the city of Encarnación where they have the worst performance being 10.3% worse. This underestimation can happen due to the fact that there are numerous cities with low incidence, yielding lower predictions when the model is trained with all cities (217) of the database.

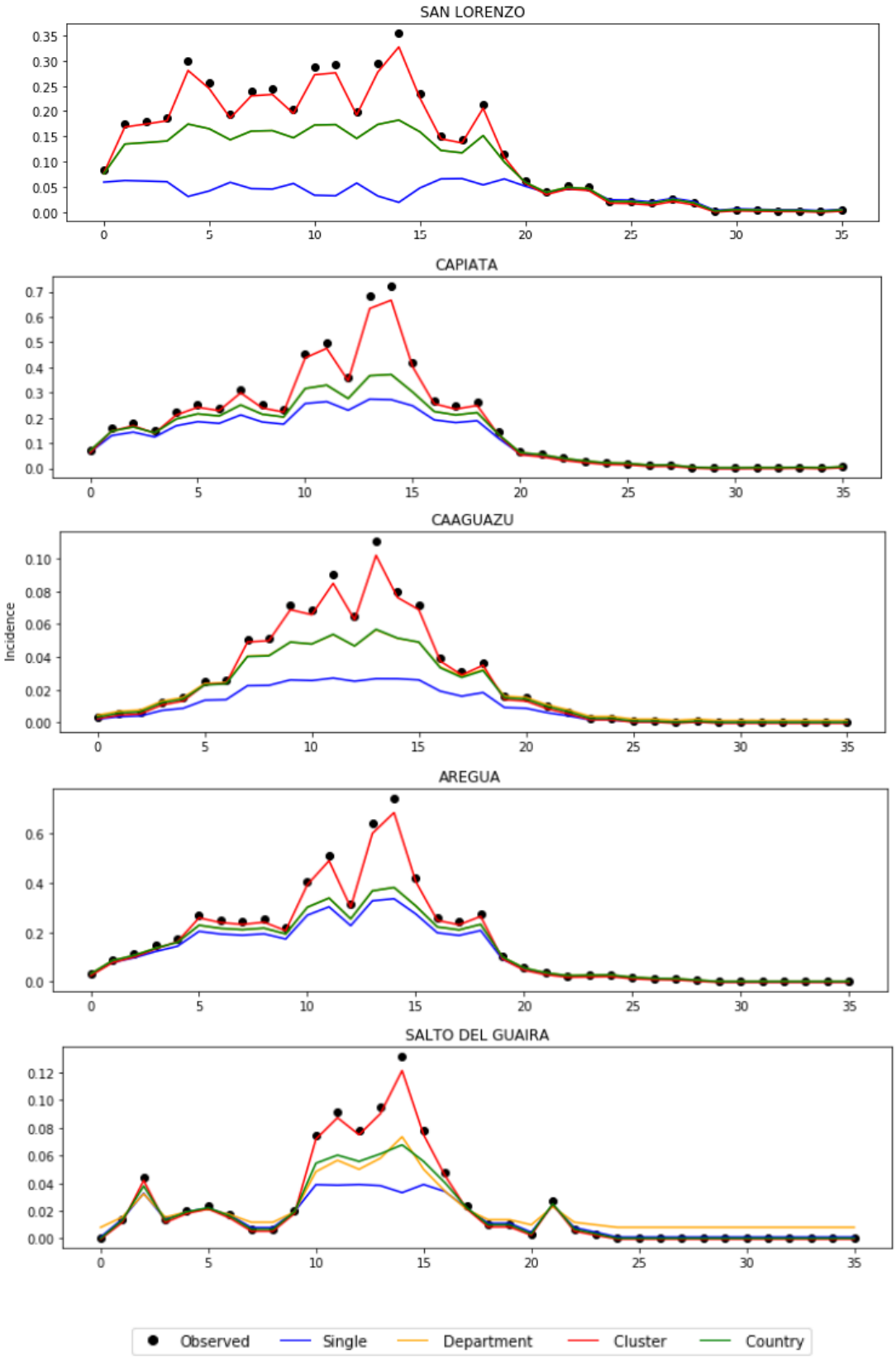


Figure 3.6: Prediction of the incidence of Dengue in the cities of group 1 (San Lorenzo, Capiatá, Caaguazú, Areguá and Salto del Guairá). Comparison of the *Single*, *Department*, *Cluster* and *Country* models with a prediction of the first 35 weeks of the year 2013.

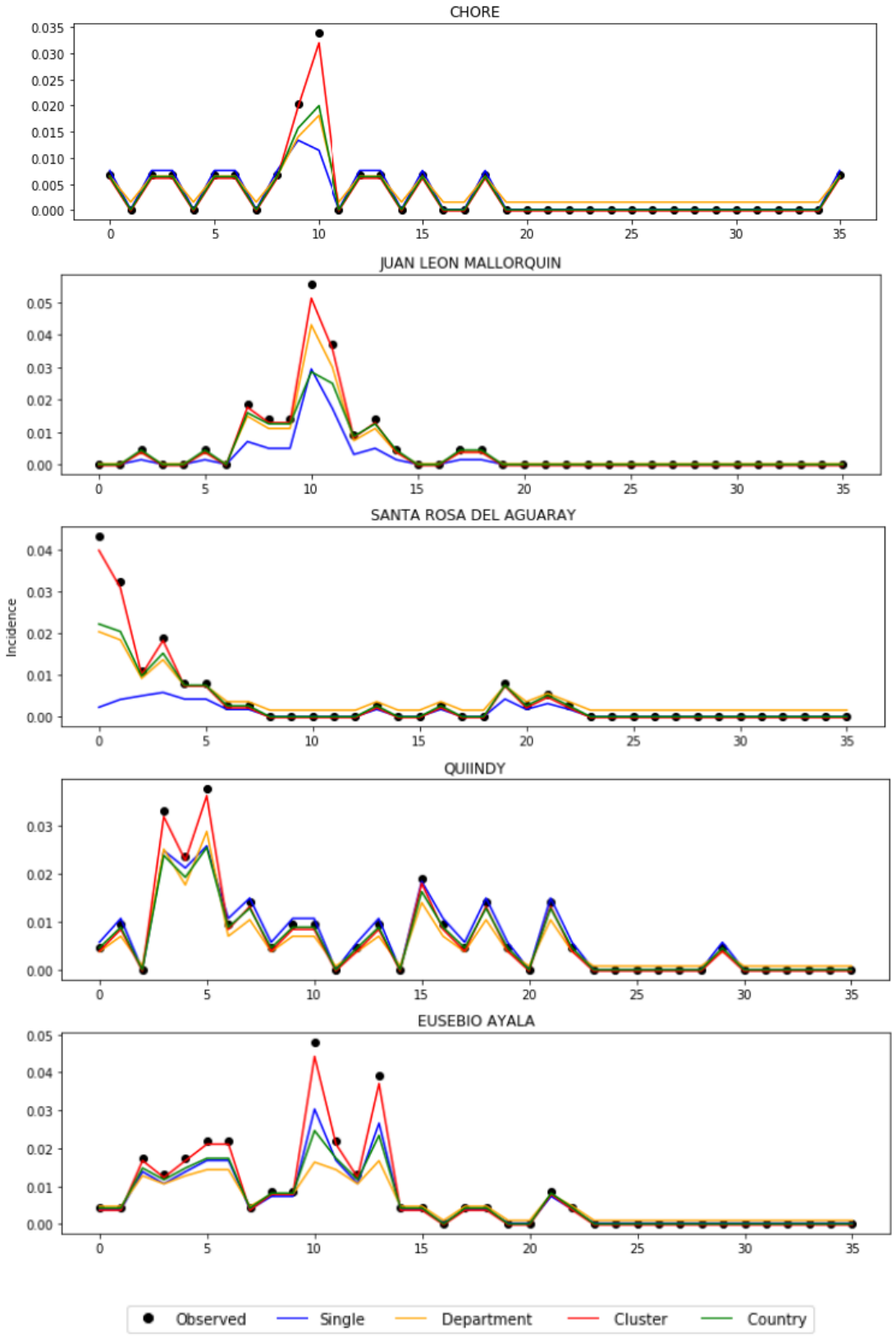


Figure 3.7: Prediction of the incidence of Dengue in the cities of group 2 (Choré, Juan León Mallorquín, Santa Rosa del Aguaray, Quiindy and Eusebio Ayala). Comparison of the *Single*, *Department*, *Cluster* and *Country* models with a prediction of the first 35 weeks of the year 2013.

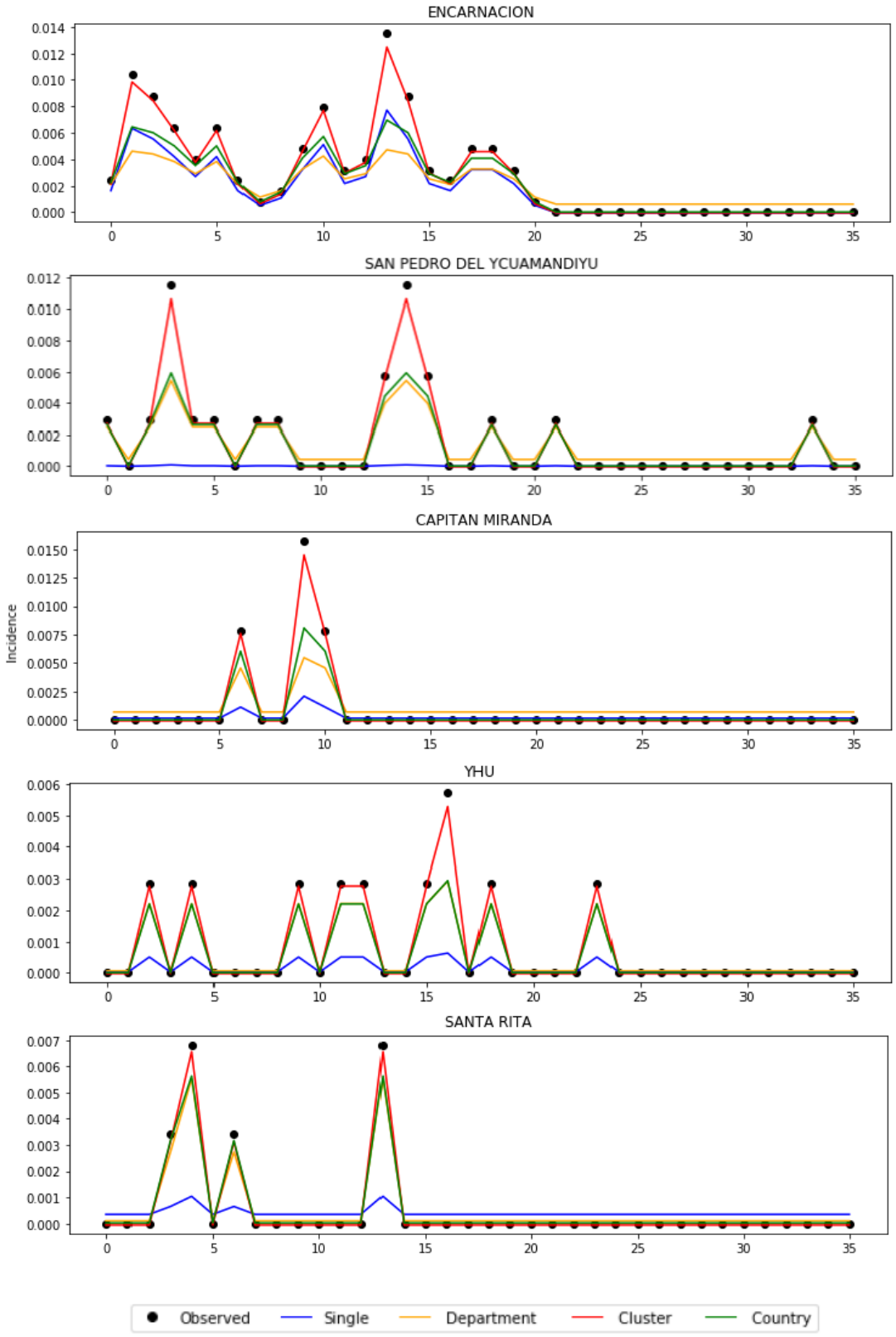


Figure 3.8: Prediction of the incidence of Dengue in the cities of group 3 (Encarnación, San Pedro del Ycuamandijú, Capitán Miranda, Yhú and Santa Rita). Comparison of the *Single*, *Department*, *Cluster* and *Country* models with a prediction of the first 35 weeks of the year 2013.

Single model performs very similarly to *Country*, but fails in high incidence cities. This can be clearly seen in the difference between the best model and himself in these cases: Areguá 16.5% and San Lorenzo 62.5%. A possible cause for this phenomenon is that low incidence cities usually do not record cases in the early years of the epidemic. This means a lack of data for the model, and therefore, it is not able to learn the behavior of the outbreaks.

The *Department* model is the one with the worst overall performance, his is not too surprising as political-territorial organization is not related to the incidence of Dengue cases.

In general, the models ranked according to how low RMSE they have are first *Cluster*, second *Country*, third *Single* and finally *Department*. To cover the entire country with *Cluster* it is necessary to train 6 models, for *Country* only one but with a large amount of data which is quite computationally expensive and with *Single* 217 models are necessary, which represents a lot of work. Hence, when the country needs to be analyzed, the *Cluster* presents the best trade-off between computational cost and performance.

Chapter 4

TIME SERIES DATA AUGMENTATION

This chapter discusses classical data augmentation techniques applied to machine learning, also other more sophisticated ones such as Bayesian inference. The basic idea of data augmentation is to generate a synthetic data set that covers the unexplored input space while maintaining the correct labels [78]. In this way, it is sought to reduce overfitting using the synthetic data when training a model.

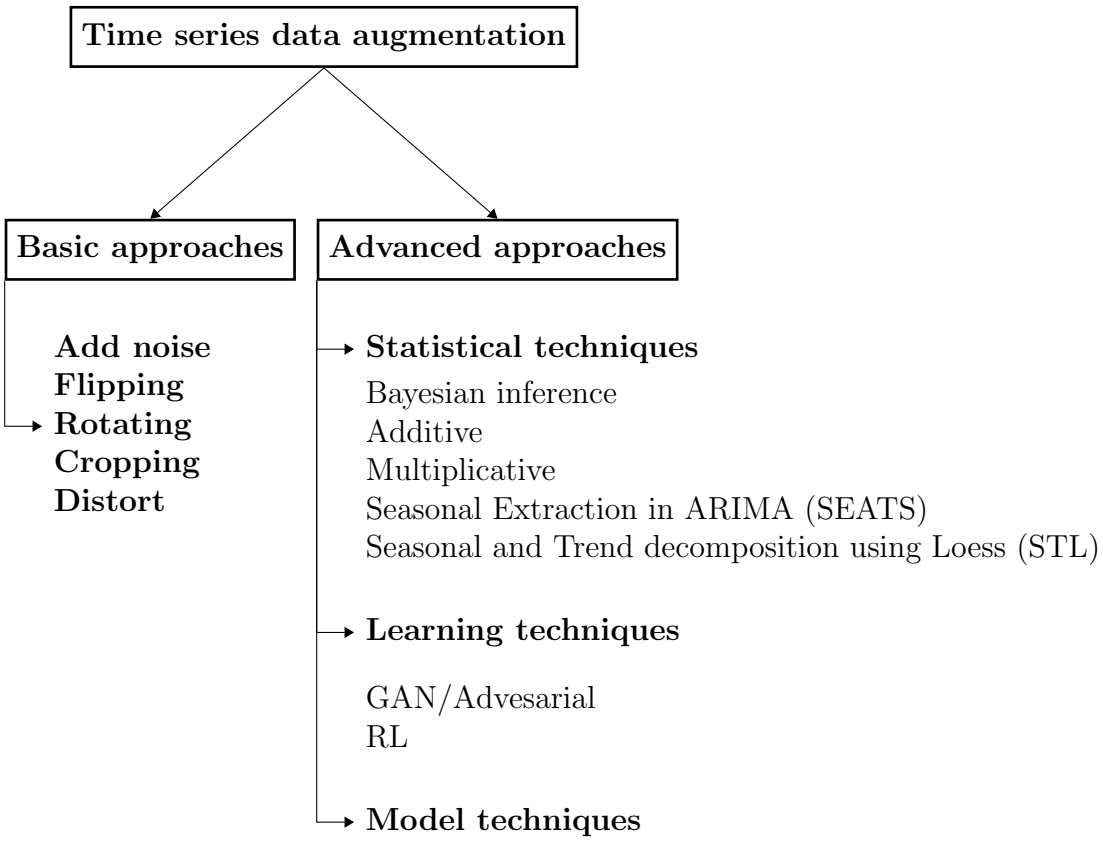


Figure 4.1: Taxonomy of data augmetation techniques for time series. Adapted from [78].

Synthetic data for data augmentation can be generated in several ways, Wen [78] proposes a taxonomy for the classification of data augmentation techniques for time series, as seen in Figure 4.1. The techniques are classified as Basic and Advanced. All the techniques of the basic approach will be analyzed. For the advanced techniques, since the learning techniques are based on deep learning models, and the objective of this work is to improve a deep learning model, this technique will not be analyzed because it is redundant. For the techniques with models, it is necessary to know a model and the parameters that generate the time series to apply variations, in this case this information is not available. Of the statistical techniques, only Bayesian inference allows obtaining a distribution of parameters associated with the observations to generate synthetic data. The other statistical techniques (additive, multiplicative, Seasonal Extraction in ARIMA (SEATS) and Seasonal and Trend decomposition using Loess (STL)) are based on decomposing the series into its components, such as trend or seasonality, these techniques allow obtaining up to four decomposition of each one, which is not significant compared to the others techniques.

Then, from the basic approach all techniques will be presented, and from the advanced approach Bayesian inference will be presented. All the approaches will be compared to each other in an experiment to determine which is the best performer. Figure 4.2 shows a summary diagram of this proposal.

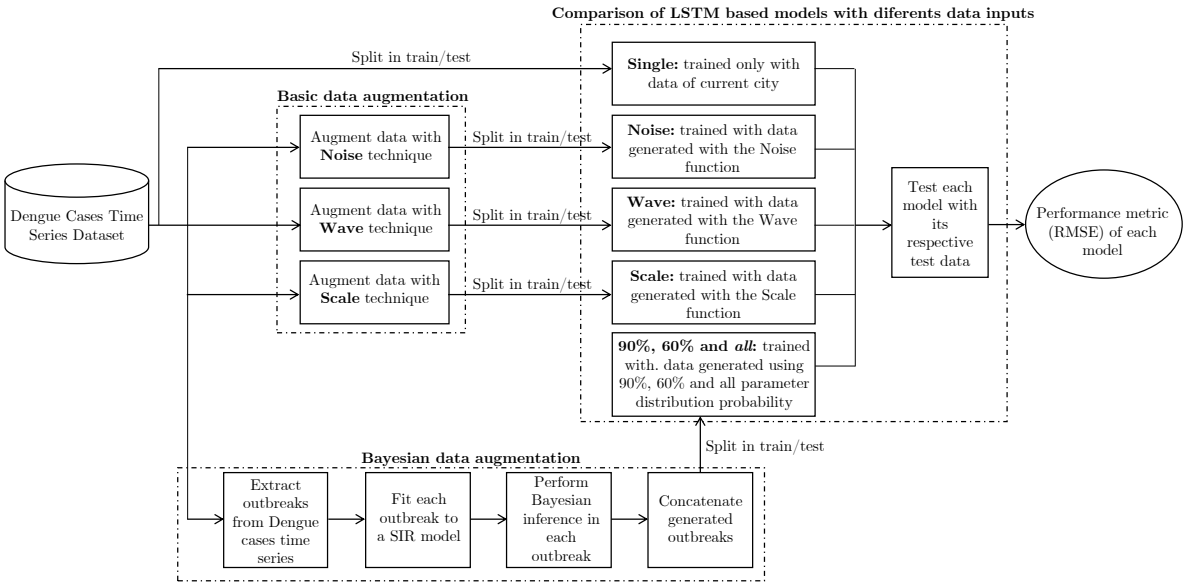
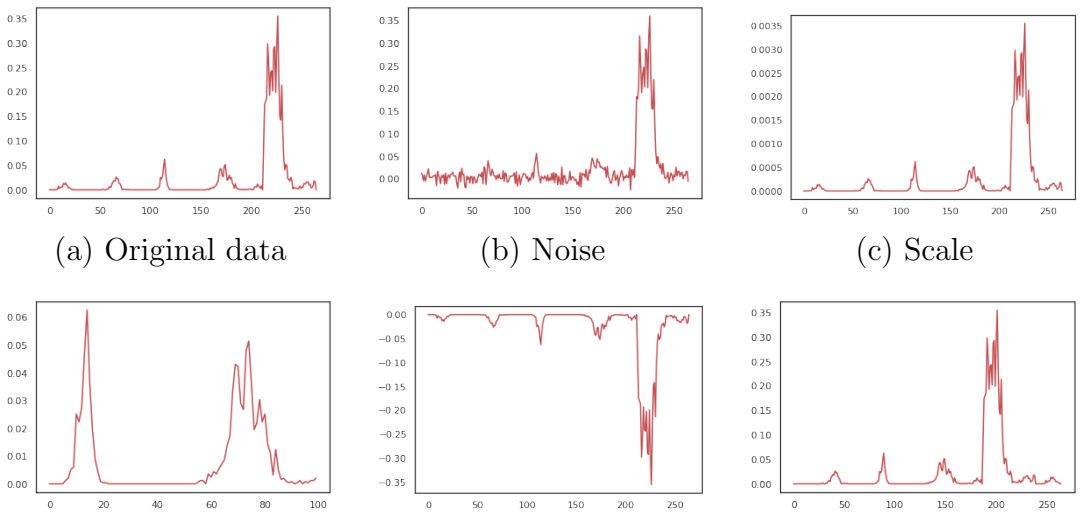


Figure 4.2: Summarized workflow of the proposal for data augmentation techniques. New time series is generated to be used as input for the LSTM based models *Noise*, *Wave*, *Scale* (using basic data augmentation approaches), 90%, 60% and *all* (using Bayesian data augmentation approach) in order to compare them against each other to determine which one has the best performance.



(a) Original data (b) Noise (c) Scale

(d) Crop (e) Flip (f) Wave

Figure 4.3: Time series data augmentation techniques. Adapted from [56].

4.1 Basic Approaches

The basic approach to data augmentation is to apply small changes from the original series to generate the synthetic ones. Recently, Rashid *et al.* [56] applied data augmentation with methods similar to those applied in image processing. But some techniques such as color variation cannot be adapted to time series since the concept of color does not exist in this type of data, then not all techniques applied to images can be applied to time series due to the differences between the data types. In order to apply any of those techniques to time series, it is wanted that the series does not lose the characteristics that define it.

For this work, techniques based on adding noise, shifting the observations and scaling the series will be used. Since flip and crop directly change the seasons of the observed series, the effects of these techniques can be seen in Figure 4.17. It is necessary to clarify that the work of Rashid et al. addresses a problem of classification of time series, while this work addresses a regression problem, specifically of Dengue cases in Paraguay. However, Rashid *et al.* did not evaluate whether or not the applied techniques produce a benefit for their problem. This is why in this work these techniques for data augmentation will be considered individually to measure their efficiency (if any) in the model.

This work seeks to evaluate the best data augmentation technique, these techniques will be compared with each other, in order to determine which one has the best performance. The best performing technique will then be compared to the Bayesian Data Augmentation approach.

4.1.1 Experimental Results

These techniques seek to generate series similar to those observed but without changing their trend, seasonality, or autocorrelation. The three techniques used for this experiment are:

1. *Noise.* New time series are generated by adding white noise to them. White

noise is a stochastic process where its variables are not correlated, a white noise signal has zero mean which is generated taking random values from a normal distribution with $\mu = 0$ and $\sigma = 1$, then this value is added to the original observation, thus the series with noise is obtained. Recall that μ is mean and σ is standard deviation. Figure 4.4 shows the effect of applying noise to the observed time series.

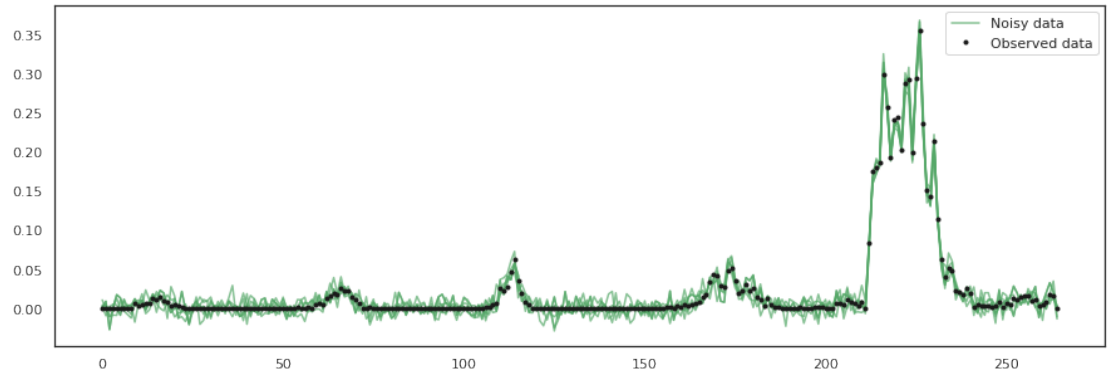


Figure 4.4: Time series observed with noisy series. For illustrative purposes only five noisy series are shown.

2. *Wave.* This approach shift the values by a factor of i steps, can be expressed as: $y_t = y_{t+i}\ \forall y_t \in Y$, where Y is the time series. For this experiment $i \in [-5, 5]$. Figure 4.5 shows the effect of wave.

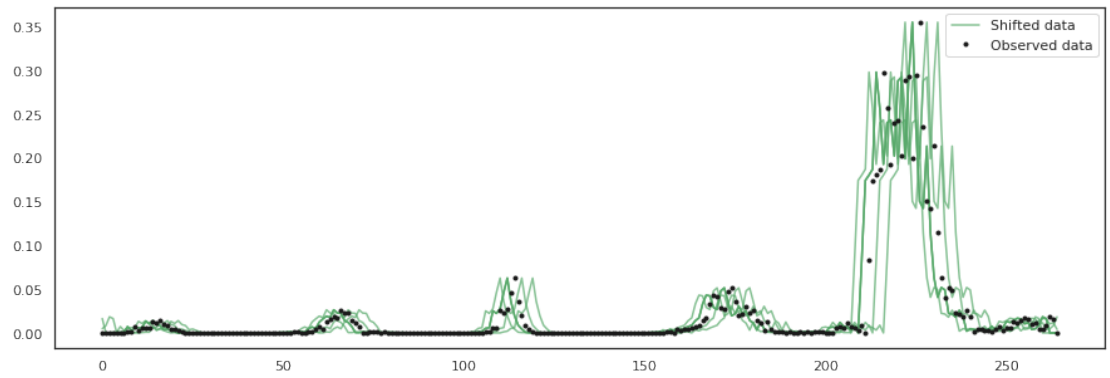


Figure 4.5: Time series observed with series resulting from the wave function. For illustrative purposes only five noisy series are shown.

3. *Scale.* This function consists of using a factor k in the time series, can be expressed as: $y_t = y_t * k\ \forall y \in Y$, where Y is the observed time series. For this experiment $k \in [-0.5, 0.5]$. Figure 4.6 shows the effect of scale.

Table 4.1: Comparison of each LSTM model using RMSE. Values in bold are the best ones.

Group	**City**	*Single*	*Noise*	*Wave*	*Scaled*
	San Lorenzo	0.1360	0.1242	**0.0503**	0.0675
	Capiatá	0.1334	0.2598	0.1264	**0.1218**
Group 1	Caaguazú	0.0266	0.0412	**0.0223**	0.0225
	Areguá	0.1201	0.2623	0.1364	**0.1191**
	Salto del Guairá	0.0259	0.0482	0.0252	**0.0225**
	Choré	0.0075	0.0191	**0.0062**	0.0063
	Juan León Mallorquin	0.0096	0.0229	**0.0089**	0.0090
Group 2	Santa Rosa del Aguaray	0.0060	0.0182	**0.0036**	**0.0036**
	Quiindy	0.0095	0.0220	**0.0087**	**0.0087**
	Eusebio Ayala	**0.0101**	0.0200	0.0112	0.0102
	Encarnación	**0.0028**	0.0053	0.0029	0.0029
	San Pedro del Ycuamandijú	0.0033	0.0040	**0.0027**	**0.0027**
Group 3	Capitán Miranda	0.0031	0.0074	**0.0030**	0.0031
	Yhú	0.0017	0.0025	**0.0015**	**0.0015**
	Santa Rita	0.0021	0.0077	**0.0017**	**0.0017**
Average RMSE		0.0332	0.0577	0.0274	**0.0269**

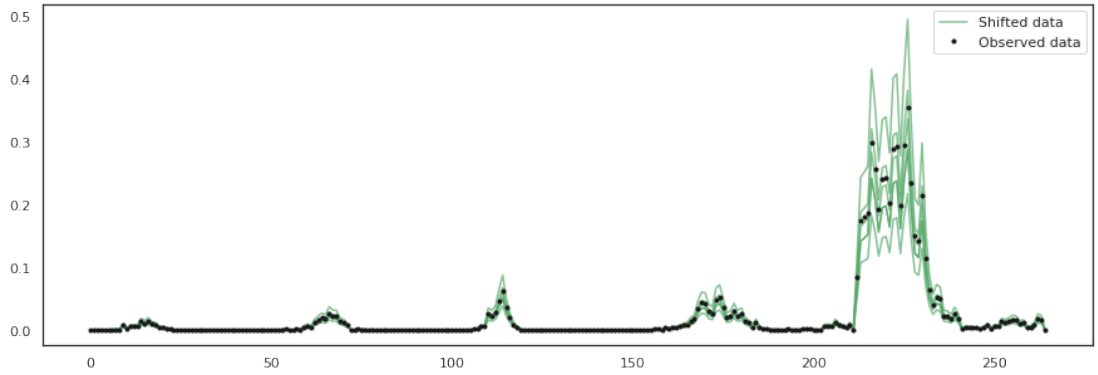


Figure 4.6: Time series observed with series resulting from the scale function. For illustrative purposes only five noisy series are shown.

The *Single*, *Noise*, *Wave* and *Scale* models were trained with directly observed data, with data created by adding noise to the observations, with data created by moving the observations on the x-axis and with data created by multiplying the observations by a scalar respectively. Table 4.1 shows that the *Wave* model is the one with the lowest RMSE error in more cities, but the *Scale* model is the one with the best average error. This is related to the standard deviation since the *Scale* model exceeds or matches the *Wave* model as the tests are done in cities with lower incidence.

The *Single* model outperforms the other metrics in certain cases, such as in the city of Eusebio Ayala or Encarnaciín (see Figures 4.8 and 4.9), however its performance does not present much difference, especially in cities with low incidence (see Figure 4.9). It is important to note that the *Single* model was trained without additional data, so it represents the result of not using data augmentation techniques.

However, the advantage of the *Single* model over the one immediately after it is only 9.3% in the best case, when it is not the best, the best model beats the *Single* model by 63.1%.

The *Wave* model performs particularly well in the cities with the highest incidence, in group 1, as seen in Figure 4.7. However, as it is tested in cities with lower incidence, its performance dramatically decreases, in Figure 4.8 it can be seen how that is the worst model of group 2 (cities with medium incidence). The relationship between the incidence of cities and the performance of the network may be related to the form of the series, cities with high incidence tend to have high and well-defined peaks, while cities with low incidence do not have defined peaks or they do not have any peak in the early years.

The *Wave* model may have the ability to capture the characteristics of the series, however it does not seem to have the ability to capture the trend of the series. This data augmentation technique can be useful for well-marked series with constant trends.

The *Scale* model is the one that remains constant and is the best, albeit by little, in most cities. The generalization capabilities of the *Scale* model do not seem to be better than those of the *Wave* model, but the *Scale* model is closer to the peaks, it is this characteristic that positions it as the best. Although no model has been able to reach the real peaks, at least in this experiment.

The *Noise* model is the one with the worst performance, in most cases it is not capable of inferring any variation in the series. Although there are cases where it is observed that it has a certain capacity to detect where the peaks will be, as seen in the city of San Pedro del Ycuamandijú (Figure 4.9). These results indicate that noise is not well handled by the LSTM model, and that for input it may be to suggest removing noise from the series.

Sorting the models based on their performance, the first place is occupied by the *Scale* model followed closely by the *Wave* model, the *Single* model is third and the *Noise* model is by far the worst model. The benefit of using these data augmentation techniques is very low compared to previous experiments, but these techniques have the advantage of being computationally inexpensive and do not require other requirements to be used.

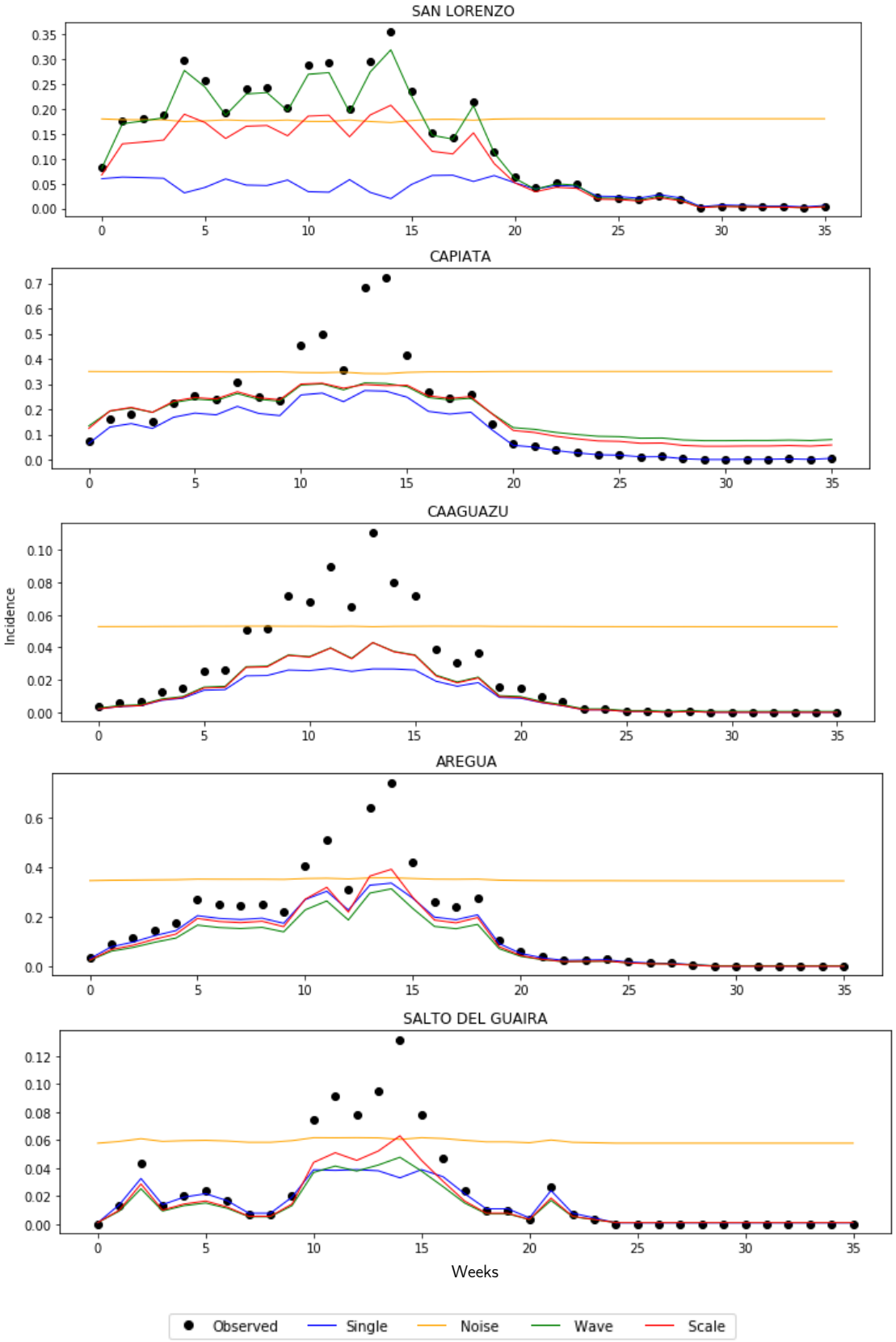


Figure 4.7: Prediction of the incidence of Dengue in the cities of group 1 (San Lorenzo, Capiatá, Caaguazú, Areguá and Salto del Guairá). Comparison of the *single*, *Noise*, *Wave* and *Scaled* models with a prediction of the first 35 weeks of the year 2013.

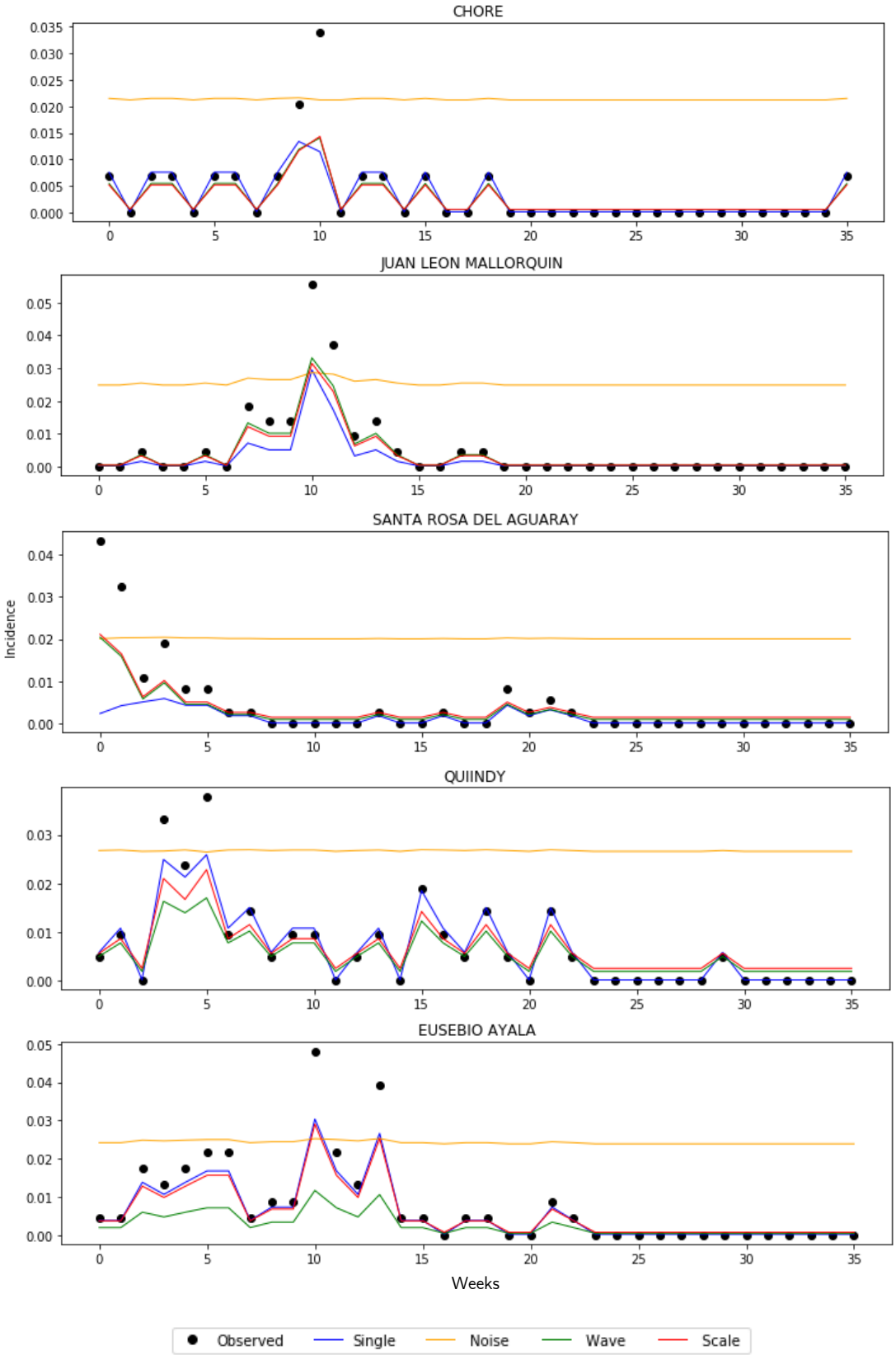


Figure 4.8: Prediction of the incidence of Dengue in the cities of group 2 (Choré, Juan León Mallorquín, Santa Rosa del Aguaray, Quiindy and Eusebio Ayala). Comparison of the *single*, *Noise*, *Wave* and *Scaled* models with a prediction of the first 35 weeks of the year 2013.

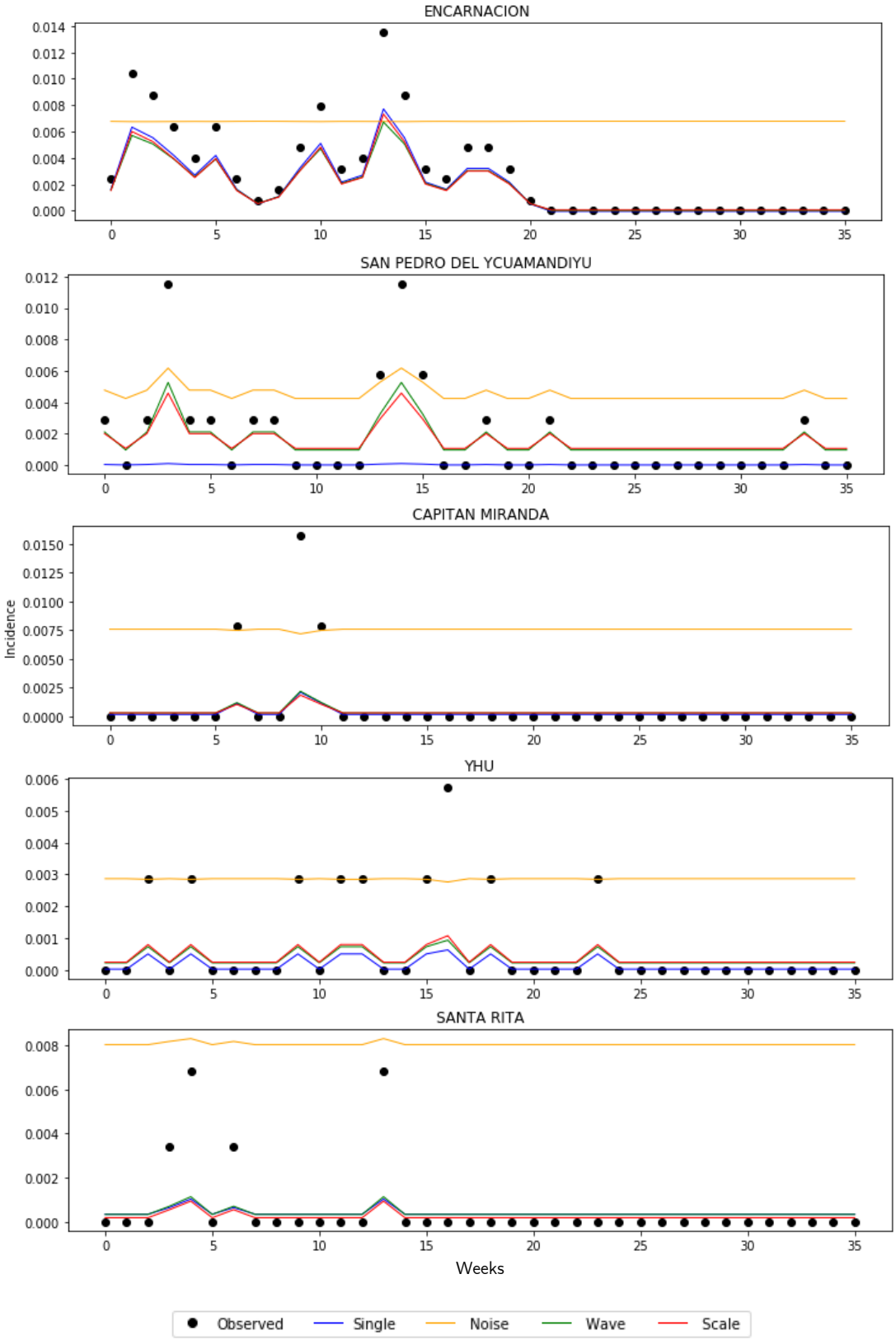


Figure 4.9: Prediction of the incidence of Dengue in the cities of group 3 (Encarnación, San Pedro del Ycuamandijú, Capitán Miranda, Yhú and Santa Rita). Comparison of the *single*, *Noise*, *Wave* and *Scaled* models with a prediction of the first 35 weeks of the year 2013.

4.2 Bayesian Inference

Bayesian inference is a statistical inference technique based on Bayes' theorem. Bayes' theorem is closely related to the concept of conditional probability [8]. The conditional probability of an event A is the probability that the event occurs knowing that an event B has already occurred. This probability is written $P(A \mid B)$, which means probability of A given B. In the case where event B has no effect on the probability of event A, the conditional probability of the event A is simply the probability of event A, *i. e.*, $P(A)$. From this definition, the conditional probability is described as

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}, \tag{4.1}$$

where $P(A \cap B)$ is the probability that A and B occur at the same time, and $P(B)$ is the probability of B occurring.

The term Bayes' theorem is in honor of Reverend Thomas Bayes, and is also referred as Bayes law [69]. This theorem shows the conditional probability or posterior probability, or simply *posterior* of an event A after B is observed in terms of the prior probability of A, prior probability of B and the conditional probability of B given A. Bayes' theorem is defined as follows

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}, \tag{4.2}$$

where $P(A)$ is the probability of A occurring, $P(B)$ is the probability of B occurring, $P(B \mid A)$ is the probability of B given A and $P(A \mid B)$ is the probability of A given B. Bayes' theorem relies on incorporating prior probability $P(A)$ distributions in order to generate posterior probabilities $P(A \mid B)$, to show how true a hypothesis is, based on evidence. This approach can be represented as follows

$$P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)}, \tag{4.3}$$

where H stands for hypothesis and E for evidence, $P(H \mid E)$ is the so-called posterior distribution probability, or simply *posterior*, $P(E \mid H)$ is the likelihood - which will be detailed later -, $P(H)$ is the prior probability distribution, or simply *prior*, represents the information available on the hypothesis regardless of any previous experiment, $P(E)$ is the marginal probability density of the data in all possible hypotheses, it is equal to: $\int_H P(E \mid H)P(H)\, dH$. Equation 4.3 indicates that the probability can vary as more evidence is added, therefore this process can be made iterative and thus find the posterior probabilities.

In this context, *Bayesian inference* is a method of statistical inference that uses Bayes' theorem to update the probability of a hypothesis as more evidence is available. Bayesian models have in common the assignment of probability as a measure of belief of a hypothesis (prior), so inference is a process of readjustment of measures of belief when new evidence are known. Bayesian inference looks at the evidence and calculates an estimated value based on the prior assigned to the hypothesis. This implies that having more data available can obtain more conclusive results. Moreover, a model can be considered as the hypothesis to be tested, this leads to the definition of Bayesian Inference for parameter estimation.

4.2.1 Bayesian Inference for parameter estimation

With the Bayesian formulation you can solve problems of parameter estimation, model selection, and hypothesis testing, in this work it will be used for parameter estimation. The basis of the Bayesian inference comes from the Bayes theorem, to apply it to parameter estimation of models, the following modification is made to equation 4.2:

$$P(\Theta \mid y) = \frac{P(y \mid \Theta)P(\Theta)}{P(y)}, \tag{4.4}$$

where y are observations and Θ is a set of parameters for a model, then the following components are defined:

- $P(\Theta)$ is the set of prior distributions of parameter set Θ before y is observed.
- $P(y|\Theta)$ is the likelihood of y given a model.
- $P(\Theta|y)$ is the full posterior distribution, of parameter set Θ that expresses uncertainty about parameter set Θ after considering both the prior and data into account.
- $P(y)$ is defined as $\int P(y \mid \Theta)P(\Theta)\, d\Theta$.

Since there are usually multiple parameters, Θ represents a set of j parameters that can be considered like this

$$\Theta = \theta_1, \theta_2, ..., \theta_j, \tag{4.5}$$

With this approach, Bayesian inference can be used to apply it to a mathematical model for Dengue cases, *e.g.*, SIR Model, and obtain a probability distribution of its parameters. Like any Bayesian approach, the main components are the prior and the likelihood.

4.2.1.1 Prior Distribution

From the above data, the *prior* distribution is one of the main concepts in the Bayes theorem and therefore in Bayesian inference, priors are basically a probability distribution associated with the quantity Θ before any observation is available [55]. A probability distribution is a function that assigns a random variable a probability of occurring , as exemplified in Figure 4.10. A prior can be determined from past evidence, such as previous experiments. Prior probability distributions have usually belonged to one of two categories: informative priors and uninformative priors.

- *Informative priors.* When previous information about the model parameters is available, it is added as a prior, this can come from previous experiments or from the literature. In this way the estimation does not start from scratch and is closer to the expected solution. However, in most cases this information is not available, so uninformative priors are used.
- *Uninformative priors.* Is a class of prior in which the objective is to minimize the amount of subjective information content, and using a prior that is determined only by the model and the observed data. Uninformative priors also provide information to the model, only it makes the inference start more scratch.

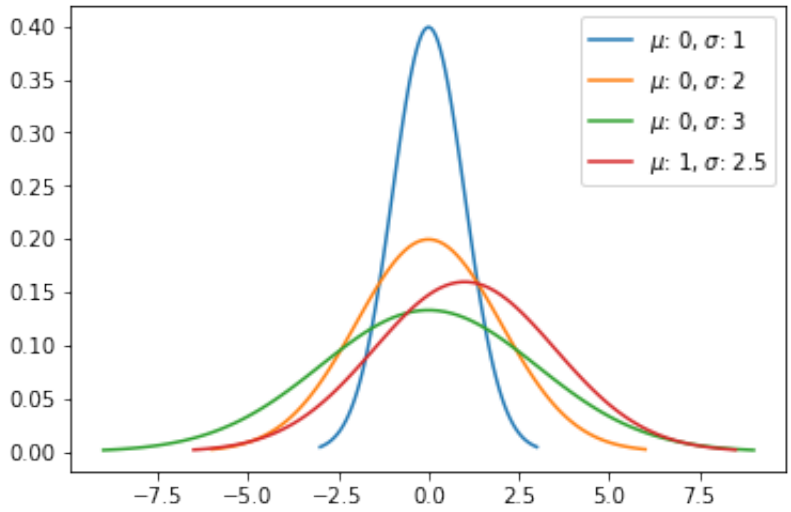


Figure 4.10: Normal distributions with different values for the mean (μ) and standard deviation (σ) parameters.

Applied in Bayes' theorem, the prior is multiplied by the likelihood function and then normalized to estimate the posterior probability distribution.

4.2.1.2 Likelihood and Posterior Distribution

The likelihood function, measures how well a statistical model fits a sample of data for given values of the unknown parameters. Likelihood represents the available information provided by the observations. Is defined as:

$$P(y \mid \Theta) = \prod_{i}^{n} P(y_i \mid \Theta), \tag{4.6}$$

where y_i is each sample of an observation, in this case, in a time series. The effect of the data y on the posterior distribution $P(\Theta \mid y)$ is obtained through the probability $P(y \mid \Theta)$.

In this way, the Bayesian inference based on models is carried out on the set of parameters Θ of a certain model, for cases of Dengue it can be the SIR model, then $\Theta = \gamma, \beta$ according to the equation 2.1, and the data y are the cases observed in a certain period of time, the likelihood $P(y|\Theta)$ is defined by the probability of each parameter of Θ, and the prior $P(\Theta)$ is the distribution that is estimated the parameter set Θ has before taking the observations into account. Finally the posterior $P(\Theta|y)$ is the estimated probability distribution of the parameter set Θ according to the observed data. Finally, with the posterior distribution, each possible value for the set of parameters Θ has an associated probability and it can be verified how likely it is to occur in the context of the observed data. As the result of Bayesian inference is a distribution for the parameter set Θ, data augmentation for a deep learning model can be done by generating several simulations with the parameters that sample from Θ, as shown in figure 4.11.

What is done here is the estimation of unknown parameters of models from observations that are assumed to come from that model, this is called an inverse problem, in health areas such as epidemiology it is called parameter estimation

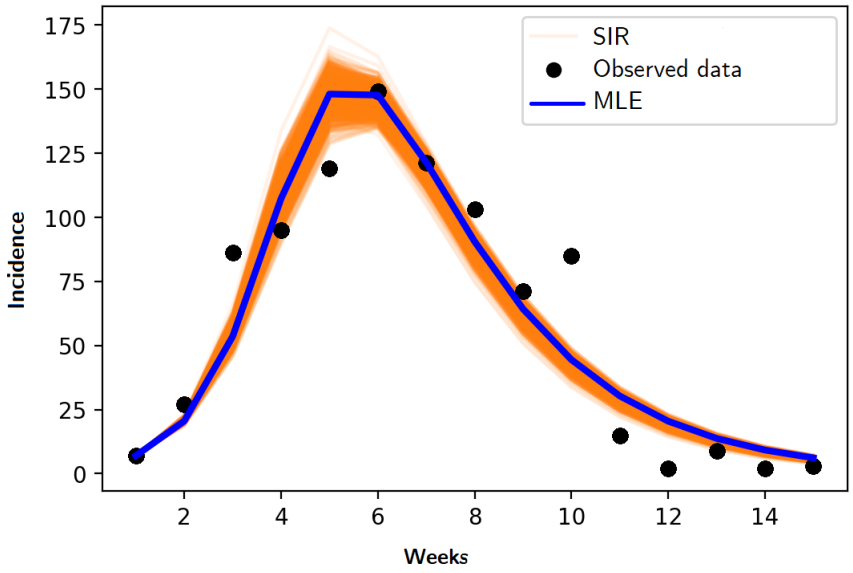


Figure 4.11: Sample of fictitious data that mimics an epidemiological outbreak of Dengue cases, the maximum value of the likelihood (MLE) and 1,000 simulations generated with samples of the distributions of the set of parameters, in this case γ and β from the SIR model.

or model calibration [73]. The main goal of parameter estimation is to find the parameters for a model in such a way that the model output matches the observed data as closely as possible. The estimation process usually consists of iteratively varying the parameters of the model until the result fits the observed data. The parameter estimation of the model can be seen as an optimization problem whose objective is the best possible parameter configuration and look for techniques that minimize the error, *e. g.*, Least Squares Minimization. If the model is integrable, an analytical solution can be found. If not, you can use a sampling algorithm like *Markov chain Monte Carlo* can be used to sample and thus estimate the full posterior distribution of parameters given priors and observed data [24].

4.2.2 Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) combines Monte Carlo simulations and Markov chains and, basically, is a computer–driven sampling method. Monte Carlo simulations attempt to estimate a parameter by repeatedly generating random numbers.

Monte Carlo simulations seek to estimate a parameter by repeatedly generating random numbers. Monte Carlo assumes that samples can be drawn in the domain of the target, and as those samples fall into the target, the probability density function of these points can be calculated. Suppose that a two-dimensional plane is had where the points form a letter E, a function that characterizes those points is difficult to find, but taking several random points in the plane and selecting those that fall within the points that form the letter E, we can obtain the probability density function of the parameters. That is the idea behind Monte Carlo. Indeed, the more points are used, the better the result.

A Markov Chain is a discrete-time stochastic process where the current value

is dependent on the value of the prior variable. This means that the current value of the string depends only on the previous value in the chain. Mathematically, let $\{X_0, X_1, ...\}$ a stochastic process then $(X_t)_{t\geq 0}$ is a Markov chain defined as:

$$P(X_t \mid X_0, X_1, ..., X_{t-1}) = P(X_t \mid X_{t-1}) \tag{4.7}$$

A random walk is an example of a Markov chain. The random walk is a mathematical formalization of the path that results from making successive random steps. For example, on a number line, a random walk starts at a point X_0, then randomly takes a step to a position X_1 randomly taking $+1$ or -1 steps from its current position. In a random walk, the next position depends only on the current position, that is the behavior of a Markov chain.

Back in the context of distribution functions, MCMC is a combination of Monte Carlo sampling and Markov chains. MCMC chooses a random parameter value to be considered. The simulation will continue to generate random values (Monte Carlo), but according to an algorithm that determines if it can be considered a good value for the parameter. It is possible to compare each generated value with another and determine which is more explanatory by calculating the probability that it represents the data. If a randomly generated parameter value is better than the previous one, it is added to the string of parameter values with some associated probability (Markov chain).

The algorithm that determines if it can be considered a good value for the parameter is called Metropolis-Hastings, there are other algorithms, but for this work the algorithm used is Metropolis-Hastings. This algorithm is based on a Markov chain that generates a candidate for parameter. The algorithm attempt to determine if the candidate is in the correct trajectory, by accept or reject a parameter candidate in the chain [23]. This algorithm first draws a candidate C of the distribution $Q(C; X_t)$ at position X_t, then the candidate is accepted with probability

$$min\left(1, \frac{P(C \mid y)}{P(X_t \mid y)} \frac{Q(X_t; C)}{Q(C; X_t)}\right). \tag{4.8}$$

where $P(C \mid y)$ is the probability of the candidate given the data observed, $P(X_t \mid y)$ is the probability of the position X_t given the data observed, the transition distribution $Q(C; X_t)$ is a a distribution designed to be easy to sample, a common parameterization of $Q(Y; X(t))$ is a multivariate Gaussian distribution centered on $X(t)$ [21]. Algorithm 8, adapted form [21] represents the iterative step of Metropolis-Hastings.

Algorithm 8: Step of Metropolis-Hastings

draw a candidate $C \sim Q(Y; X(t))$
$q \leftarrow \frac{P(C \mid y)}{P(X_t \mid y)} \frac{Q(X_t; C)}{Q(C; X_t)}$
$r \leftarrow R \sim [0, 1]$
if $r \leq q$ **then**
  $X_{t+1} \leftarrow C$
else
  $X_{t+1} \leftarrow X_t$
end

This means that if candidate C is not accepted, the position X_t is repeated in the chain. In this way, the Metropolis-Hastings algorithm works together with MCMC.

In summary, an inference can be made on the parameters of a model from Bayes' theorem, iteratively obtaining samples using MCMC and evaluating the probability of each sample. This technique allows obtaining a distribution of parameters for the model, called posterior.

4.2.3 Bayesian Inference on Epidemic Models

Bayesian inference allows obtaining a distribution of parameters of a model, this distribution is called posterior. By taking samples from this distribution, simulations similar to the observations can be obtained. In this work, the model used to characterize Dengue outbreaks is the SIR model.

The SIR model, as defined in equation (2.1), provides information on the situation of Susceptible, Infected and Recovered individuals, in the context of Dengue cases. Only information on Infected individuals (observed data) will be used, figure 4.12 emphasizes the curve of infected in a simulation.

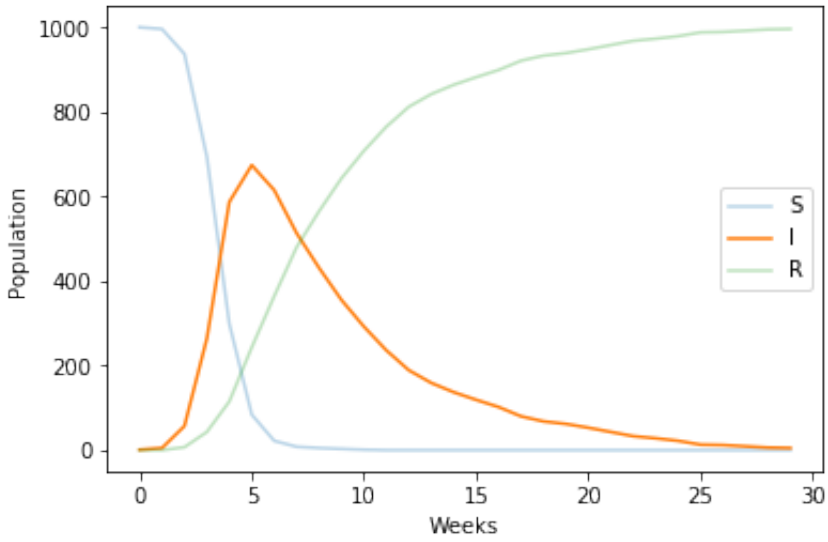


Figure 4.12: SIR model with initial values $S_0 = 999$, $I_0 = 1$, $R_0 = 0$, $N = 1,000$, $\beta = 0.002$ and $\gamma = 0.2$. The curve of infected individuals (I) is highlighted.

Recalling Model-based Bayesian inference concept (Section 4.2.5), it is possible to infer the distribution of parameters to generate data similar to the observations. With this distribution of parameters, it is possible to perform simulations and obtain data similar to those observed. The central idea of this experiment is to use these simulations to improve the performance of an LSTM network, since the more data the better the network performance [70].

4.2.4 Multi-season SIR model

The SIR model results in a single curve, which represents a single epidemic outbreak, and the observations in the form of a time series are entered as input into the LSTM network as a vector $Y = [y_1, y_2, ..., y_t]$ with t weekly observations, as shown in figure 4.13.

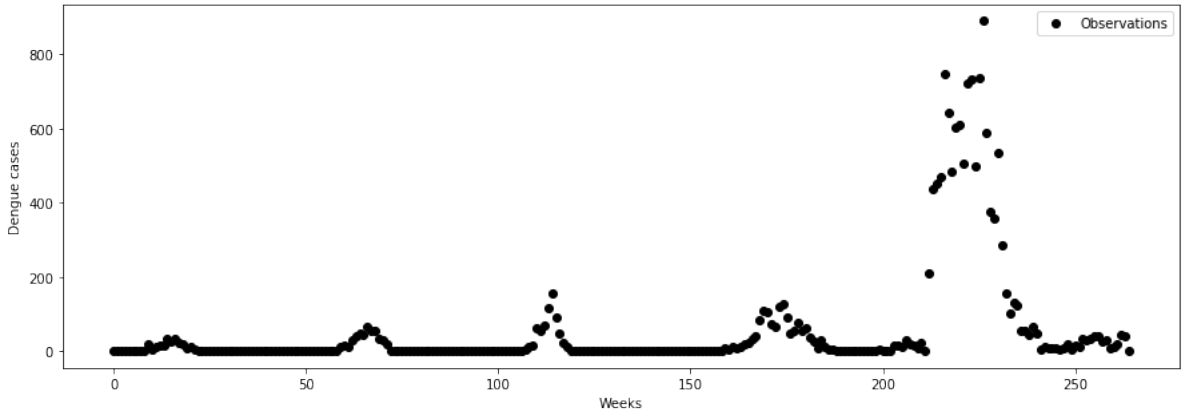


Figure 4.13: Observations of Dengue cases, in this sample: San Lorenzo city, the data correspond to observations from 2009 to 2013. It can be seen that there are five outbreaks.

An outbreak is the significant increase in cases in relation to the values usually observed. When outbreaks occur seasonally, the disease causing the outbreaks is said to be endemic, such as Dengue in Paraguay. In this way, each peak within the time series represents an outbreak. In order to carry out the inference using the SIR model for time series, in this work, a multi-season SIR model is proposed, to apply this model first each peak of the time series must be extracted. The peaks of each outbreak are found with algorithm 9.

Algorithm 9: Find peaks

Data: time series, *range*
Result: peaks
peaks=[ ]
TS= time series vector
t = time series length
$diff$ = TS[0] - TS[1]
for $i = 1$ **to** t **do**
  for $j = i$ **to** $range - 1$ **do**
    if $diff^2 < 0$ **then**
      add TSr[j+1] to peaks
    end
    $diff$ = TSr[j] - TSr[j+1]
  end
  peak=max(peaks)
end

Basically it is to find the local maximum in the time series, as an outbreak lasts approximately 40 weeks [43], that is the value of *range* that was used in the algorithm. From 2009 to 2013, there are five peaks or less depending on the city. Once the peaks have been identified, the next step is to determine the beginning and end of the outbreak. This is called finding the width of the outbreak. The

algorithm 10 shows the process to find the width of an outbreak.

Algorithm 10: Find width of a single outbreak

Data: time series peaks
Result: peaks width start and end index
i=peak position
TS= time series vector
t = time series length
while *TS[i]* $\neq 0$ **do**
    start=TS[i] i=i-1
end
j=peak position
while *TS[j]* $\neq 0$ **do**
    end=TS[j] j=j+1
end

Figure 4.14 shows the peaks and the width of each outbreak as a result of the algorithms 9 and 10, the observations are represented with a solid line for illustrative purposes. This algorithm was used to find the peaks of each city.

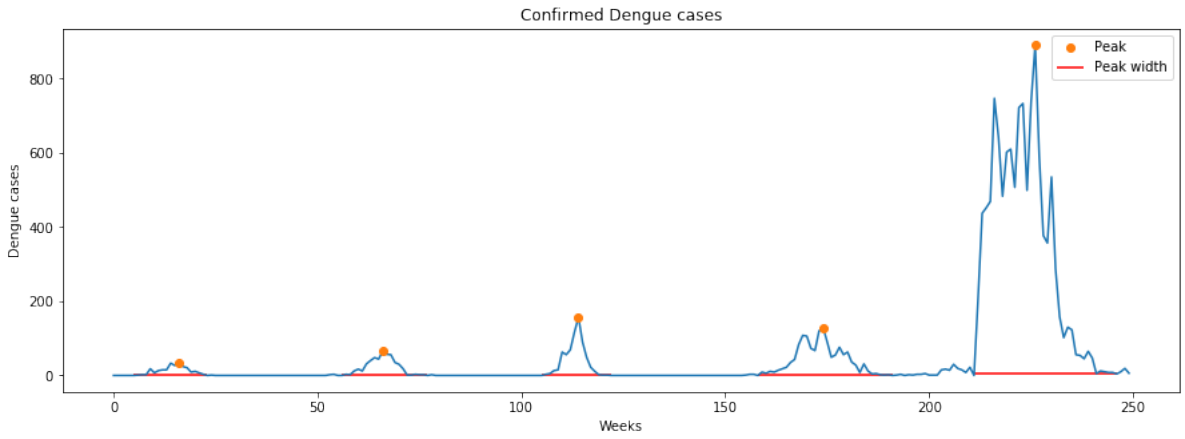


Figure 4.14: Data from the city of San Lorenzo indicating the peaks found and their width.

Then, for each city, the process of finding peaks is carried out and their widths once we have the observations in the form of individual outbreaks (see figure 4.15) can be adjusted to a SIR model.

Thus, a modified version of the SIR model can be defined for time series that works by seasons, this model will be called `SeasonalSIR`. `SeasonalSIR` receives a time series, finds the outbreaks and the beginning and end of each one, for each outbreak, the traditional SIR model is executed, each outbreak is considered a season, a season with active cases is followed by another without cases, to handle this, the `SeasonalSIR` model artificially drops the β value to zero. With an infection rate of $\beta \leq 0$, an outbreak does not occur. In this way, the `SeasonalSIR` model can return a multiseasonal Dengue cases time series. Algorithm 12 shows how the `SeasonalSIR` function works, *start* and *end* are vectors that have the beginning and end of each season or outbreak respectively, *seasons* is a vector with an index for each season of the series, β is a vector that contains the values of β_s for each season

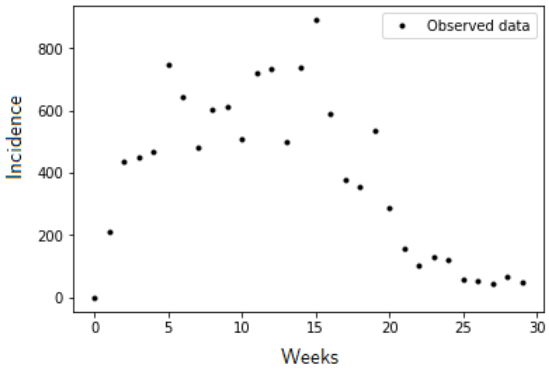


Figure 4.15: A single outbreak taken from the series of observations of the city of San Lorenzo.

Table 4.2: RMSE of different MLE functions and observations

Function	**RMSE**
Poisson	$2,025.2399$
Normal	$2,614.0839$
Normal with moving average	$6,333.2479$

and γ is a vector with the γ_s values for each season, $SIR()$ runs the usual SIR model according to the equation 2.1. Vectors *starts*,*ends* and *peaks* are obtained from each time series to be adjusted. The vectors β and γ are the vectors to be obtained by performing the Bayesian inference. The length of each vector is equal to the number of seasons the series has.

Function `SeasonalSIR`$(starts, ends, season, \beta, \gamma)$`:`
  foreach *season* **do**
    $t = ends - starts$
    $TS \longleftarrow SIR(\beta, \gamma, t)$
  end
  return TS
End Function

Algorithm 11: SeasonalSIR function

To perform Bayesian inference on data, it is necessary, in addition to the data, the likelihood function and the prior. In this case, there is no previous information that helps to decide which is the likelihood function to use and the distribution of the prior. Therefore, this is information must be assumed, to help decide what likelihood function to use, a small experiment has been carried out.

The gamma, normal and normal likelihood functions were tested with moving average and a maximum estimate of likelihood (MLE) was found. The estimate that is closest to the observed data is considered the most appropriate. The difference between the MLE and the observations is measured with RMSE (2.19). Table 4.2 shows the results of this experiment. Figure 4.16 shows a graphical comparison. With this, everything necessary to perform the Bayesian inference is had.

All the previously proposed techniques (add noise, scale, shift, bayesian data augmentation) will be applied to the Dengue database, before being trained with a machine learning model, then the performance of each one will be evaluated and it will be compared if there are improvements against the model without augmented

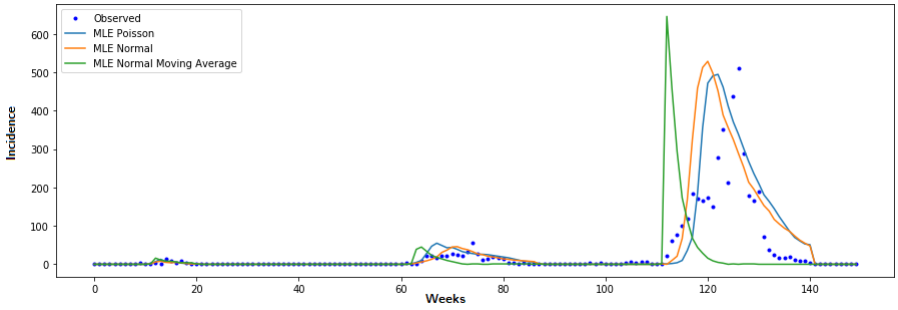


Figure 4.16: Comparison of MLE values according to different likelihood functions and observations.

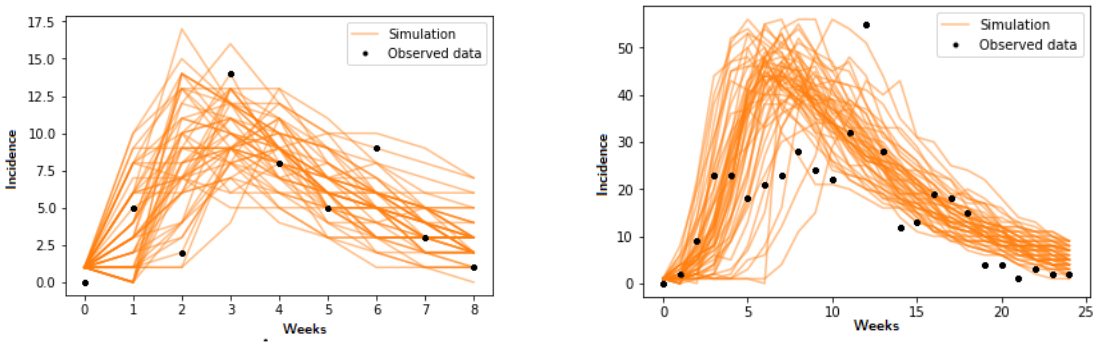
data.

4.2.5 Bayesian Data Augmentation

Using the function `SeasonalSIR` (described in 12) with the likelihood and the prior, the MCMC method can be applied to find the posterior. Each value of the posterior obtained has an associated probability, so for example those with $\geq 90\%$ probability belong to the 90% credibility interval, in this experiment, the samples are varied between credibility intervals to find the optimal. The following steps are carried out for each city that belongs to the experiment:

1. Find the peaks and their respective widths.
2. Run MCMC for `SeasonalSIR model`.
3. Take 100 samples that belong to the 90% credibility interval of the posterior one and generate 100 series from them.
4. Take 100 samples that belong to the 60% credibility interval of the posterior one and generate 100 series from them.
5. Take 100 random samples from the posterior and generate 100 series from them.
6. Train the LSTM model in 3 different groups, 90% (with the samples 90% credibility interval), 60% (with samples of 60% credibility interval) and *all* (with the random samples).
7. Compare the results of 90%, 60% and *all* using RMSE.
8. Perform a binary search to find the percentage that represents the interval with the best results.

The MCMC took 100,000 samples per variable, and the posteriors were obtained for each case. This is how the data was generated to feed the LSTM models. Figure 4.17 show the simulations along with the observations. All these models were evaluated at the city level to check their generalization in the forecasts.



(a) (b)

Figure 4.17: (a) and (b) are details of observations and simulations generated from the samples taken from the posteriors. Only 50 simulations were plotted and the series was cropped into single outbreaks for illustrative purposes.

4.2.6 Experimental results

The experiments were done on each time series individually. The models tested were: *all* with 100 simulations generated with random samples from the distribution, 90% with 100 simulation from samples corresponding to the 90% credibility interval, 60% with 100 simulation from samples corresponding to the 60% credibility interval, and *single* than the time series without added elements. Table 4.3 shows the RMSE for the predictions of these cities for the first thirty-five weeks of the year 2013. The results show that there is no significant difference between the models trained with the simulations. The graphical representation of the models results is seen below (See figures 4.18,4.19 and 4.20).

When analyzing the results in depth with more decimals, it was observed that there is a difference at $1e-4$ level between models, the plan was to perform a search between the confidence intervals to find the one with the lowest error, *i.e.*, the one with the best performance, however the difference between the samples between intervals was not as sensitive as expected. Therefore, this search was discarded and all Bayesian based models were considered as having the same performance.

Regarding the goal of improving the performance of an LSTM model by aggregating data, all models achieved that goal. In group 1 the improvement is up to 57.3%, in groups 2 and 3 this improvement drops to 10.4%. In addition, the 90%, 60% and *all* models outperforms the non-LSTM models that performed better on the benchmark test (see Table 2.2). It can be concluded that adding simulations generated from Bayesian inference improves the performance of an LSTM network. It is important to note that this technique tends to overestimate the points. In summary, when ranking the models as those with the best results *all*, 90% and 60% are tied, the *single* model is the one with the worst performance. An important observation is that these models outperform those that were better than LSTM in the benchmark model selection section 2.5.

Table 4.3: Comparison of each LSTM model using RMSE. Values in bold are the best ones.

Group	**City**	*Single*	90%	60%	*all*
	San Lorenzo	0.1360	**0.0580**	**0.0580**	**0.0580**
	Capiatá	0.1334	**0.1046**	**0.1046**	**0.1046**
Group 1	Caaguazú	0.0266	**0.0152**	**0.0152**	**0.0152**
	Areguá	0.1201	**0.1112**	**0.1112**	**0.1112**
	Salto del Guairá	0.0259	**0.0202**	**0.0202**	**0.0202**
	Choré	0.0063	**0.0061**	**0.0061**	**0.0061**
	Juan León Mallorquin	0.0096	**0.0094**	**0.0094**	**0.0094**
Group 2	Santa Rosa del Aguaray	0.0060	**0.0058**	**0.0058**	**0.0058**
	Quiindy	0.0095	**0.0094**	**0.0094**	**0.0094**
	Eusebio Ayala	0.0101	**0.0090**	**0.0090**	**0.0090**
	Encarnación	0.0028	**0.0021**	**0.0021**	**0.0021**
	San Pedro del Ycuamandijú	0.0033	**0.0032**	**0.0032**	**0.0032**
Group 3	Capitán Miranda	0.0031	**0.0030**	**0.0030**	**0.0030**
	Yhú	0.0017	**0.0016**	**0.0016**	**0.0016**
	Santa Rita	0.0021	**0.0015**	**0.0015**	**0.0015**
Average RMSE		0.0332	**0.0240**	**0.0240**	**0.0240**

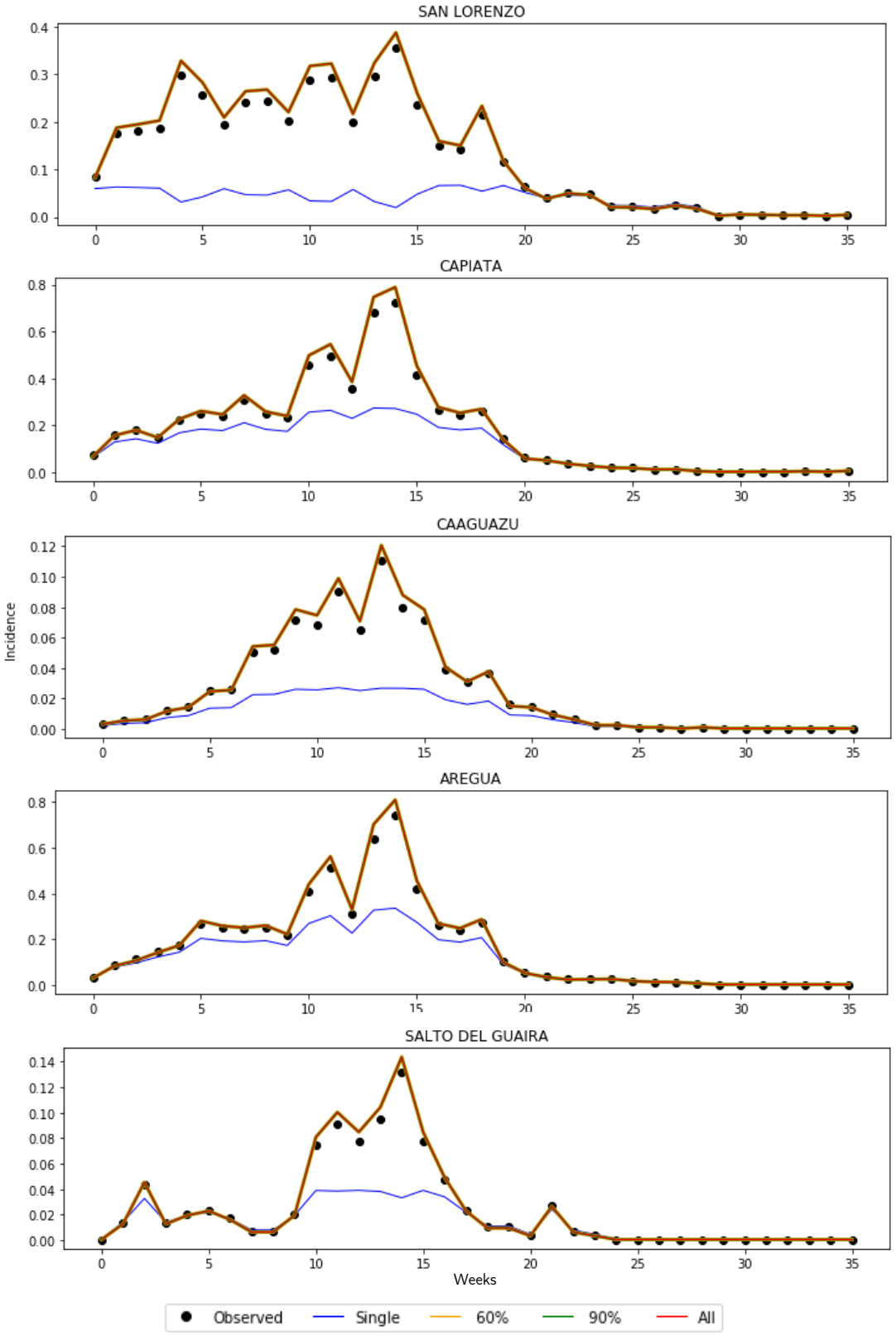


Figure 4.18: Prediction of the incidence of Dengue in the cities of group 1 (San Lorenzo, Capiatá, Caaguazú, Areguá and Salto del Guairá). Comparison of the *single*, 90%, 60% and *all* models with a prediction of the first 35 weeks of the year 2013.

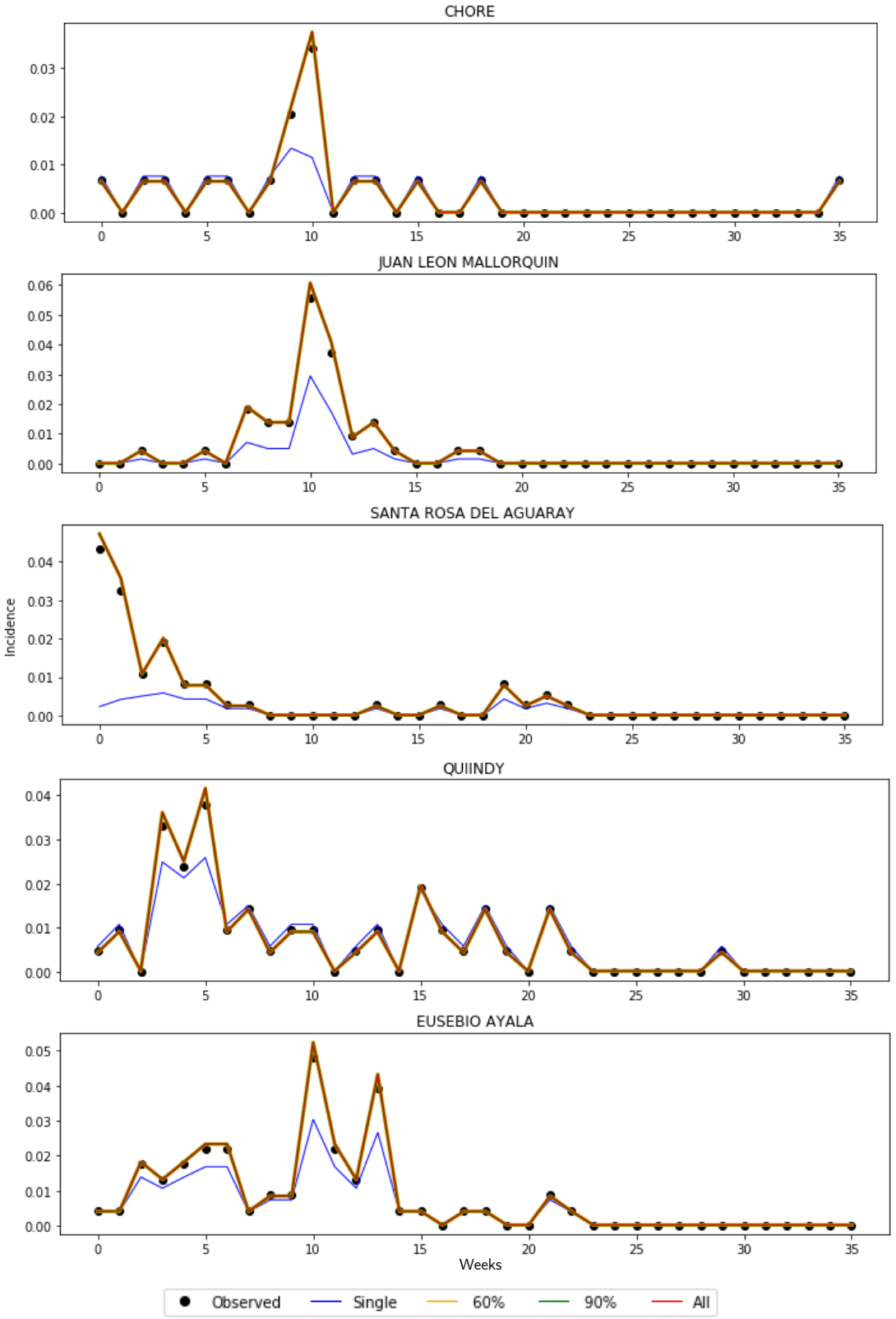


Figure 4.19: Prediction of the incidence of Dengue in the cities of group 2 (Choré, Juan León Mallorquín, Santa Rosa del Aguaray, Quiindy and Eusebio Ayala). Comparison of the *single*, 90%, 60% and *all* models with a prediction of the first 35 weeks of the year 2013.

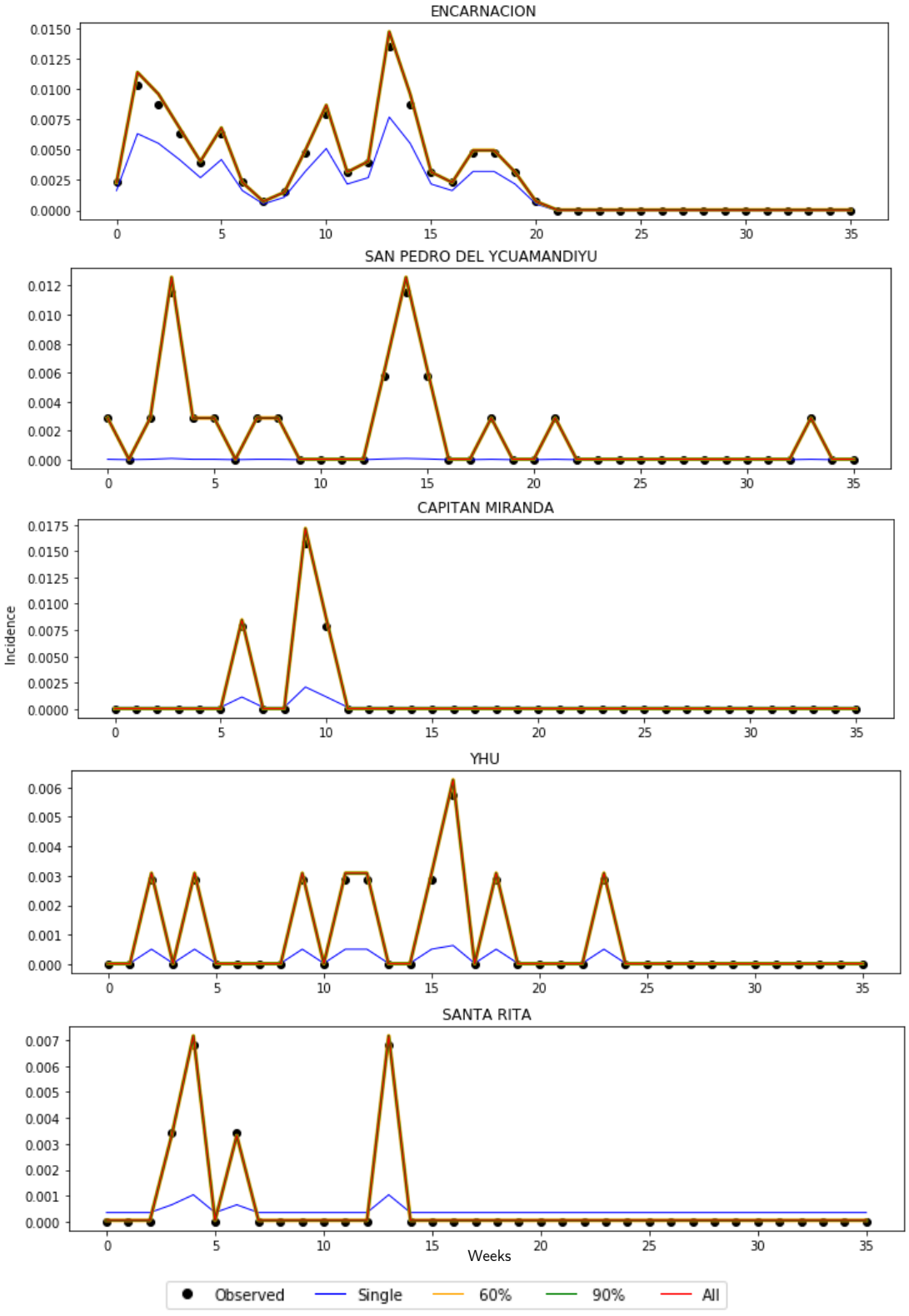


Figure 4.20: Prediction of the incidence of Dengue in the cities of group 3 (Encarnación, San Pedro del Ycuamandijú, Capitán Miranda, Yhú and Santa Rita). Comparison of the *single*, 90%, 60% and *all* models with a prediction of the first 35 weeks of the year 2013.

4.3 Basic Approaches vs. Bayesian Data Augmentation Experimental Results

Among the techniques of the basic approach, *Scale* was the one that obtained the best result. From the techniques of the advanced approach, the Bayesian inference model is had, as the results of these experiments are very similar, one was chosen and it was called *Bayesian*. Both models are compared below as shown in Table 4.4.

Table 4.4: Comparison of each LSTM model using RMSE. Values in bold are the best ones.

Group	**City**	*Bayesian*	*Scale*
Group 1	San Lorenzo	**0.0520**	0.0675
	Capiatá	**0.0956**	0.1218
	Caaguazú	**0.0138**	0.0225
	Areguá	**0.1021**	0.1191
	Salto del Guairá	**0.0188**	0.0225
Group 2	Choré	**0.0063**	0.0077
	Juan León Mallorquín	**0.0090**	0.0098
	Santa Rosa del Aguaray	**0.0036**	0.0051
	Quiindy	**0.0087**	0.0101
	Eusebio Ayala	**0.0102**	0.0117
Group 3	Encarnación	**0.0029**	0.0029
	San Pedro del Ycuamandijú	**0.0027**	0.0032
	Capitán Miranda	**0.0031**	0.0037
	Yhú	**0.0015**	0.0020
	Santa Rita	**0.0017**	0.0023
Average RMSE		**0.0221**	0.0429

The *Bayesian* model outperforms the *Scale* model in all the cities sampled. Therefore, it can be said that among the data augmentation techniques, the most effective is *Bayesian*, Figure 4.21 shows the performance comparison between the models.

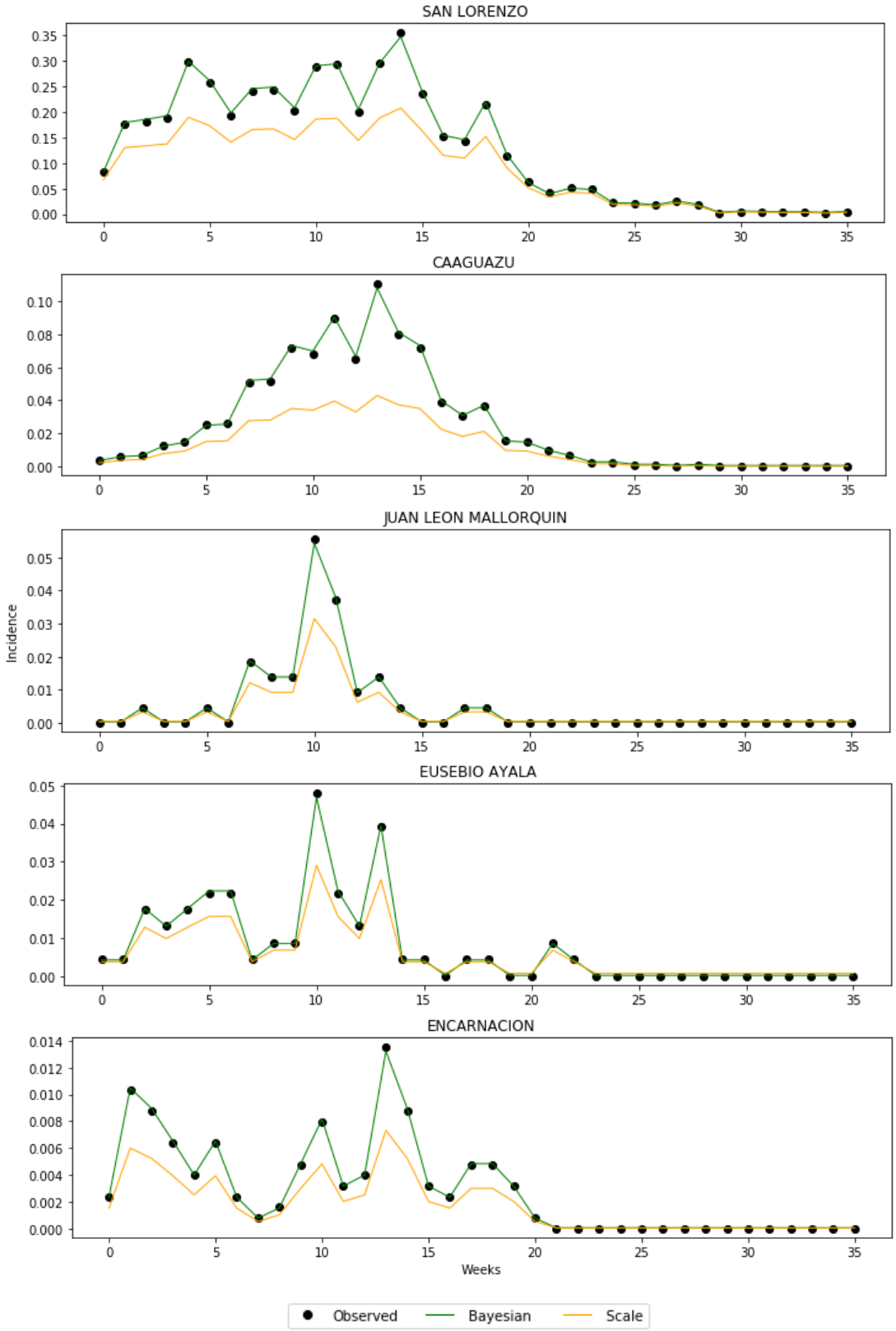


Figure 4.21: Prediction of the incidence of Dengue in the cities of group 3 (Encarnación, San Pedro del Ycuamandijú, Capitán Miranda, Yhú and Santa Rita). Comparison of the *Bayesian* and *Scale* models with a prediction of the first 35 weeks of the year 2013.

Chapter 5

TIME SERIES CLUSTERING VS. BAYESIAN DATA AUGMENTATION

The comparison of the results of the previous experiments considering the limitations of each technique, they are:

1. The difference in the data generated. With the clustering technique, each cluster has four time series per city (incidence, average temperature, average atmospheric pressure, and weekly rainfall). The Bayesian inference technique requires one model for each series, so only samples of the Dengue cases series were generated.

2. The way the models were trained. Models were trained in on-the-fly mode, *i. e.*, for each iteration (epoch) of the model training, a different sample was passed to it. This technique could not be applied to the clustering technique, since the average number of elements in a cluster is 36.17, which means that only ≈ 36 epochs can be made, when the others models were trained with 100 epochs.

So this comparison does not seek to determine which of all is the best, seeks to determine in which case each is recommended. Two experiments have been performed comparing different clustering techniques and different sampling criteria for Bayesian inference models. From these experiments, the best for each case are *Cluster* and *Bayesian* respectively. Figure 5.1 shows the performance comparison between these with each other and with the *Single* model, which represents the model without aggregated data.

When there are different time series of contiguous and delimited geographical locations, the first step should be to group them according to the results of the grouping experiment (see Section 3.5) the best way is to group them using clustering techniques, in this case of hierarchical clustering. If clustering cannot be performed, the best option is to characterize the series using a model and perform Bayesian inference as described in Section 4.2.5. Clustering techniques and Bayesian inference techniques have shown to considerably improve the performance of the models, reaching the expected values in several cases.

The *Bayesian* model allows to improve the performance of the network without the need for more additional data, unlike the *Cluster* model where more series are

needed to group them, therefore it is a good technique in case there is only one time series, this is an advantage when there is a significant lack of data. However, *Cluster* model in addition to improving the performance of the network, helps to reduce the dimensionality of the problem, since it represents a smaller number of models to train.

The *Cluster* model represents the increase in the complexity of the model, by having a multidimensional input, the input is the incidence of Dengue cases and three meteorological variables (incidence, average temperature, average atmospheric pressure, rainfall). So the vector TS that represents a city is defined as:

$$TS = \begin{bmatrix} Icd_1 & T_1^a & Pr_1^a & R_1^w \\ Icd_2 & T_2^a & Pr_1^a & R_2^w \\ \vdots & \vdots & \vdots & \vdots \\ Icd_{185} & T_{185}^a & Pr_{185}^a & R_{185}^w \end{bmatrix}$$

where Icd is the incidence, T^a is the average temperature, Pr^a is the average atmospheric pressure and R^w is the weekly rainfall, values range from 1 to 185 since each series has 265 records, and training is done on the 70% of the series, leaving the remaining 30% for validation. Then the input for each model m is defined as:

$$input_m = \begin{bmatrix} TS_1 \\ TS_2 \\ \vdots \\ TS_d \end{bmatrix}$$

where d is the dimension of each cluster. Once the input is entered into the model, it is trained with 100 epochs (The training process is detailed in Section 2.5).

On the other hand, the Bayesian model has as input TS_{input}, the matrix:

$$TS_{input} = \begin{bmatrix} Icd_{11} & Icd_{21} & \cdots & Icd_{e1} & T_1^a & Pr_1^a & R_1^w \\ Icd_{12} & Icd_{22} & \cdots & Icd_{e2} & T_2^a & Pr_1^a & R_2^w \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ Icd_{1185} & Icd_{2185} & \cdots & Icd_{e185} & T_{185}^a & Pr_{185}^a & R_{185}^w \end{bmatrix}$$

where e is the number of epochs the model was trained, where e is the number of epochs in which the model is trained. This input varies in each epoch, so that in epoch e, the input TS_{input} is:

$$TS_{input} = \begin{bmatrix} Icd_{e1} & T_1^a & Pr_1^a & R_1^w \\ Icd_{e2} & T_2^a & Pr_1^a & R_2^w \\ \vdots & \vdots & \vdots & \vdots \\ Icd_{e185} & T_{185}^a & Pr_{185}^a & R_{185}^w \end{bmatrix}$$

this indicates that for each training, a different sample of incidence is used. This training method (on-the-fly) is mostly used when looking to explore other information in addition to the data that you have when training a model [12]. However, due to the limitations of the amount of data, this method could only be used in the *Bayesian* model since the necessary number of samples can be obtained from the parameter distribution.

Although both models showed improvements in the performance of the models (See table 5.1), the most significant improvements are in group 1 for both cases. Group 1 corresponds to the cities with high incidence, these cities in general have more defined peaks. In groups 2 and 3, the peaks are not well defined or do not exist, especially in the early stages of the observations, which indicates that at the beginning of the epidemic, no cases were registered in the cities belonging to these groups. Both techniques rely on the observations to improve the models, as the series of group 1 are well defined, they are the ones with the best results.

The fact that the percentage of improvement is lower in the other cases does not mean that it is not enough, this is seen in Figure 5.1, in the cities of Juan León Mallorquín and Eusebio Ayala (Group 2) and the city of Encarnación (Group 3).

The *Cluster* model was not always the best in all cases, unlike the *Bayesian* model. This can be seen in cities where it has 0% of improvement, *i.e.*, it was not the best model.

Table 5.1: Analysis of the observed improvement percentages of the *Cluster* and *Bayesian* models. Details of this calculation can be seen in Appendix D

		Improvement (%)	
Group	**City**	*Cluster*	*Bayesian*
Group 1	San Lorenzo	62.5000	57.3529
	Capiatá	29.5352	21.5892
	Caaguazú	49.2481	42.8571
	Areguá	16.4863	7.4105
	Salto del Guairá	28.1853	22.0077
Group 2	Choré	16.0000	18.6667
	Juan León Mallorquín	0.0000	2.0833
	Santa Rosa del Aguaray	38.3333	3.3333
	Quiindy	0.0000	1.0526
	Eusebio Ayala	5.9406	10.8911
Group 3	Encarnación	7.1429	25.0000
	San Pedro del Ycuamandijú	18.1818	3.0303
	Capitán Miranda	0.0000	3.2258
	Yhú	5.8824	5.8824
	Santa Rita	19.0476	28.5714
Average improvement		19.48 ± 18.80	16.86 ± 16.57

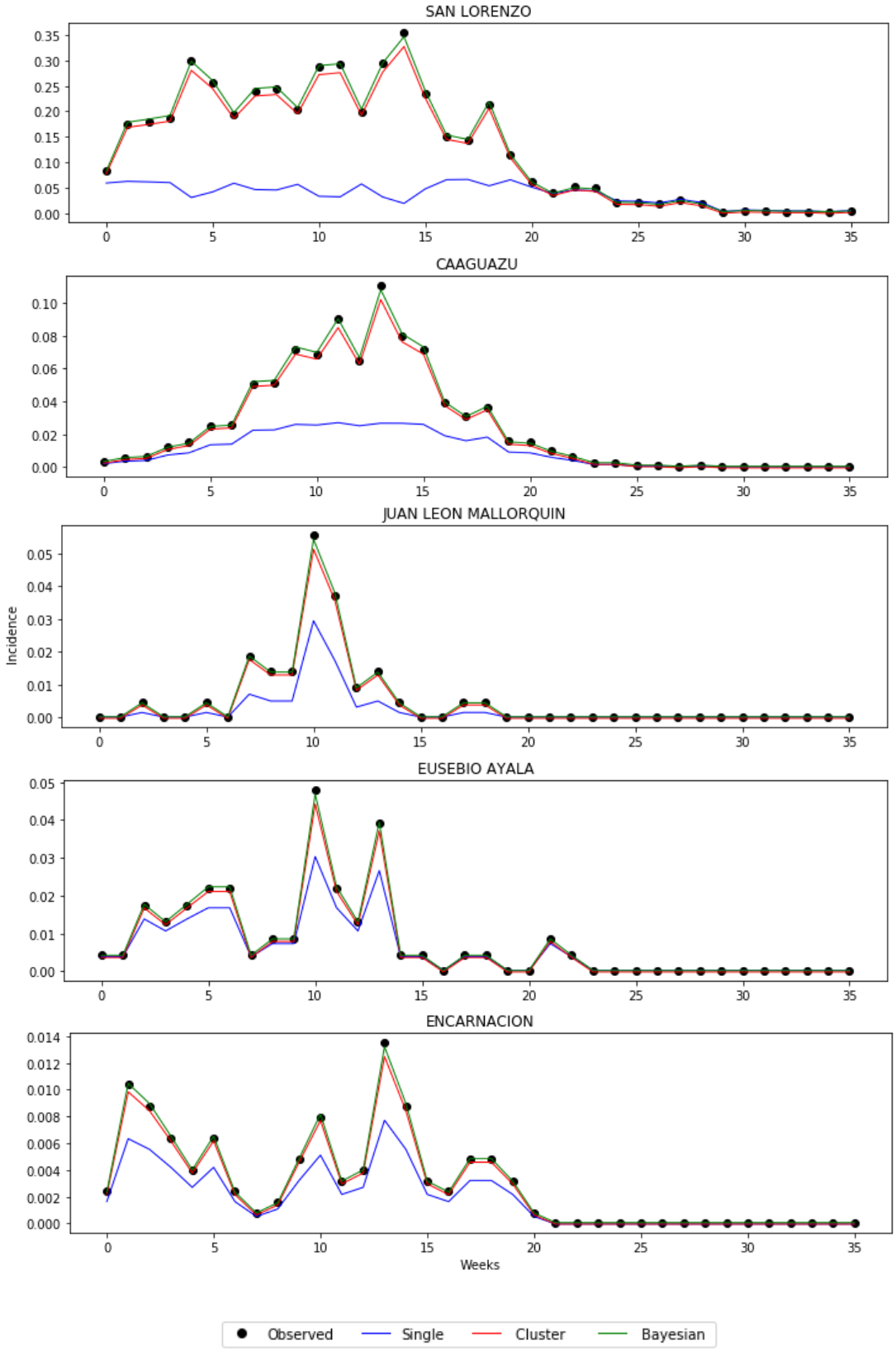


Figure 5.1: Comparison of the best results obtained in each experiment in a sample of cities (San Lorenzo and Caaguazú from group 1, Juan León Mallorquín and Eusebio Ayala from group 2 and Encarnación from group 3). Comparison of the models with a prediction of the first 35 weeks of the year 2013.

Chapter 6

CONCLUSIONS AND FUTURE WORKS

6.1 Conclusions

In this work, time series clustering and data augmentation techniques have been tested to improve the performance of a deep learning model to forecasting Dengue fever cases in Paraguay.

For representation, 5 random cities belonging to each group were selected (Group 1: high population, Group 2: medium population, Group 3: low population). The experiments were carried out in these cities.

In order to carry out the experiments, the performance of techniques traditionally used for forecasting time series (SVR, Random Forest, LARS LASSO , LSTM) was compared, with the LSTM deep learning model having the best performance. Therefore, LSTM was selected as the benchmark model, *i.e.*, the reference model for the other experiments. Using the LSTM model, it was sought to improve its performance. In the experiments, the benchmark model is called *Single*.

Then the LSTM models were trained in different ways according to each approach:

1. Times series clustering. This approach sought to improve network performance by training the LSTM model in series groups, thus having the following models:
   (a) *Department*. The LSTM model was trained with the series grouped according to the department to which they belong according to the political division of the country.
   (b) *Country*. The LSTM model was trained with all the series in the country.
   (c) *Cluster*. The LSTM model was trained according to the clustering resulting from clustering techniques. At this point, a previous study was carried out to determine which is the most appropriate clustering technique. The number of clusters to be formed was determined with the elbow method and several clustering techniques (k-means, Hierarchical, DBscan) were tested, each with a set of distance metrics (Euclidean, Correlation, Pearson's Correlation, Dynamic time warping) and the results were evaluated using silhouette score. The best results were obtained using hierarchical clustering and correlation, with this technique and metric the algorithms were formed to train this model.

Among these models, the one with the best overall performance was the *Cluster* model. The improvement is especially observed in groups 1 and 3.

2. Data augmentation. This approach seeks to improve network performance by increasing the amount of data by generating synthetic data from the observed data. Models were formed using a basic approach and Bayesian inference

   (a) Basic approaches. This approach consists of applying small transformations to the observed data to generate new ones, the models used were:

      i. *Noise*. This LSTM model was trained with random noise variations added to the observations.
      ii. *Wave*. This LSTM model was trained with random variations from the shifted series.
      iii. *Scale*. This LSTM model was trained with variations of the series multiplied by a random scalar.

   Among these models, the one that had the best average performance was *Scale*, however its performance is much worse than the *Cluster* model.

   (b) Bayesian inference. The *Bayesian* LSTM model was trained with the simulations obtained from the distribution of parameters obtained from the proposed model. The model is a modified version of the SIR model adapted for various seasons.

   The *Bayesian* model far outperforms *Scale*, the best of the basic approach, making *Bayesian* model the best among data augmentation techniques.

Both models (*Cluster* and *Bayesian*) have been shown to significantly improve the performance of a deep learning time series forecasting model. *Cluster* model is not always the best, especially in cities from groups 2 and 3, but is the model with the most significant improvement. *Bayesian* is always the best model in all tests, but tends to overestimate the cases. In problems in which a model must be adjusted to an observation, one problem is overfitting, what happens when the model cannot be generalized because it does not have enough information. So these techniques can be considered as regularization methods to avoid overfitting.

6.2 Future works

Based on the results obtained, some future works that have been identified are presented below.

- Apply these techniques in a time series classification problem.
- Combine clustering and Bayesian inference techniques.
- Optimize the clustering of the time series by designing more experiments.
- Use other modern machine learning models for forecasting time series (Gated Recurrent Unit, Bidirectional recurrent neural networks, Deep transformer).
- Apply these approaches with other endemic and vector-borne diseases.

Bibliography

[1] Salem Alelyani, Jiliang Tang, and Huan Liu. "Feature selection for clustering: a review." In: *Data clustering: algorithms and applications* 29.1 (2013).

[2] Antonio Arbo. "Dengue: heavy burden for the public health of Paraguay". In: *Revista del Instituto de Medicina Tropical* 14.1 (2019), pp. 1–2.

[3] David Arthur and Sergei Vassilvitskii. *k-means++: The advantages of careful seeding*. Tech. rep. Stanford, 2006.

[4] Kasun Bandara, Christoph Bergmeir, and Slawek Smyl. "Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach". In: *Expert Systems with Applications* 140 (2020), p. 112896.

[5] Stefano A Bini. "Artificial intelligence, machine learning, deep learning, and cognitive computing: what do these terms mean and how will they impact health care?" In: *The Journal of arthroplasty* 33.8 (2018), pp. 2358–2361.

[6] Salah Bouktif et al. "Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches". In: *Energies* 11.7 (2018), p. 1636.

[7] George EP Box and David A Pierce. "Distribution of residual autocorrelations in autoregressive-integrated moving average time series models". In: *Journal of the American statistical Association* 65.332 (1970), pp. 1509–1526.

[8] George EP Box and George C Tiao. *Bayesian inference in statistical analysis*. Vol. 40. John Wiley & Sons, 2011.

[9] Fred Brauer. "Compartmental models in epidemiology". In: *Mathematical epidemiology*. Springer, 2008, pp. 19–79.

[10] Fred Brauer, Carlos Castillo-Chavez, and Zhilan Feng. "Dengue fever and the zika virus". In: *Mathematical Models in Epidemiology*. Springer, 2019, pp. 409–425.

[11] Leo Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[12] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[13] Chih-Chung Chang and Chih-Jen Lin. "LIBSVM: A library for support vector machines". In: *ACM transactions on intelligent systems and technology (TIST)* 2.3 (2011), pp. 1–27.

[14] Chris Chatfield. *Time-series forecasting*. CRC press, 2000.

[15] Robert B Cleveland et al. "STL: A seasonal-trend decomposition". In: *Journal of official statistics* 6.1 (1990), pp. 3–73.

[16] Claudia Codeco et al. "InfoDengue: a nowcasting system for the surveillance of dengue fever transmission". In: *BioRxiv* (2016), p. 046193.

[17] Dirección Nacional de Aeronáutica Civil. *Dirección de Meteorología e Hidrología*. Aug. 2015. URL: `https://www.meteorologia.gov.py/`.

[18] Bradley Efron et al. "Least angle regression". In: *The Annals of statistics* 32.2 (2004), pp. 407–499.

[19] Martin Ester et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." In: *Kdd*. Vol. 96. 34. 1996, pp. 226–231.

[20] Jonathan Fintzi et al. "Efficient data augmentation for fitting stochastic epidemic models to prevalence data". In: *Journal of Computational and Graphical Statistics* 26.4 (2017), pp. 918–929.

[21] Daniel Foreman-Mackey et al. "emcee: the MCMC hammer". In: *Publications of the Astronomical Society of the Pacific* 125.925 (2013), p. 306.

[22] Rui Fu, Zuo Zhang, and Li Li. "Using LSTM and GRU neural network methods for traffic flow prediction". In: *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*. IEEE. 2016, pp. 324–328.

[23] Dani Gamerman and Hedibert F Lopes. *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*. CRC Press, 2006.

[24] Andrew Gelman et al. *Bayesian data analysis*. CRC press, 2013.

[25] Michael R Genesereth and Nils J Nilsson. *Logical foundations of artificial intelligence*. Morgan Kaufmann, 2012.

[26] Rafael Giusti and Gustavo EAPA Batista. "An empirical comparison of dissimilarity measures for time series classification". In: *2013 Brazilian Conference on Intelligent Systems*. IEEE. 2013, pp. 82–88.

[27] Santiago Gómez et al. "Construcción de un modelo de incidencia de dengue aplicado a comunidades de Paraguay". In: *Segundo Encuentro de investigadores*. Sociedad Científica del Paraguay. 2017.

[28] Maria G. Guzman et al. "Dengue: a continuing global threat". In: *Nature Reviews Microbiology* 8.12supp (2010), S7.

[29] Tin Kam Ho. "Random decision forests". In: *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE. 1995, pp. 278–282.

[30] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[31] Wei-Chiang Hong et al. "Forecasting urban traffic flow by SVR with continuous ACO". In: *Applied Mathematical Modelling* 35.3 (2011), pp. 1282–1291.

[32] Juan Huo, Tingting Shi, and Jing Chang. "Comparison of Random Forest and SVM for electrical short-term load forecast with different data sources". In: *2016 7th IEEE International conference on software engineering and service science (ICSESS)*. IEEE. 2016, pp. 1077–1080.

[33] Rob J Hyndman, Earo Wang, and Nikolay Laptev. "Large-scale unusual time series detection". In: *2015 IEEE international conference on data mining workshop (ICDMW)*. IEEE. 2015, pp. 1616–1619.

[34] Félix Iglesias and Wolfgang Kastner. "Analysis of similarity measures in times series clustering for the discovery of building energy patterns". In: *Energies* 6.2 (2013), pp. 579–597.

[35] Shancheng Jiang et al. "Combining Deep Neural Networks and classical time series regression models for forecasting patient flows in Hong Kong". In: *IEEE Access* 7 (2019), pp. 118965–118974.

[36] Michael A Johansson et al. "Evaluating the performance of infectious disease forecasts: A comparison of climate-driven and seasonal dengue forecasts for Mexico". In: *Scientific reports* 6 (2016), p. 33707.

[37] Perktold Josef, Seabold Skipper, and Taylor Jonathan. *statsmodels.tsa.stattools.adfuller*. `https://www.statsmodels.org/devel/generated/statsmodels.tsa.stattools.adfuller.html`. 2013.

[38] Tarjei Kristiansen. "Forecasting Nord Pool day-ahead prices with Python". In: *The Python Papers* 12.1 (2018).

[39] Sumit Kumar et al. "Energy load forecasting using deep learning approach-LSTM and GRU in spark cluster". In: *2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)*. IEEE. 2018, pp. 1–4.

[40] T Warren Liao. "Clustering of time series data—a survey". In: *Pattern recognition* 38.11 (2005), pp. 1857–1874.

[41] Andy Liaw, Matthew Wiener, et al. "Classification and regression by randomForest". In: *R news* 2.3 (2002), pp. 18–22.

[42] Liyuan Liu et al. "Lstm recurrent neural networks for influenza trends prediction". In: *International Symposium on Bioinformatics Research and Applications*. Springer. 2018, pp. 259–264.

[43] Rachel Lowe et al. "Climate services for health: predicting the evolution of the 2016 dengue season in Machala, Ecuador". In: *The lancet Planetary health* 1.4 (2017), e142–e151.

[44] Ujjwal Maulik and Sanghamitra Bandyopadhyay. "Performance evaluation of some clustering algorithms and validity indices". In: *IEEE Transactions on pattern analysis and machine intelligence* 24.12 (2002), pp. 1650–1654.

[45] José Alberto Mauricio. "Análisis de series temporales". In: *Universidad Complutence de Madrid* (2007).

[46] Jorge D Mello-Román et al. "Predictive Models for the Medical Diagnosis of Dengue: A Case Study in Paraguay". In: *Computational and mathematical methods in medicine* 2019 (2019).

[47] Seonwoo Min, Byunghan Lee, and Sungroh Yoon. "Deep learning in bioinformatics". In: *Briefings in bioinformatics* 18.5 (2017), pp. 851–869.

[48] Fabian Mörchen. *Time series feature extraction for data mining using DWT and DFT*. 2003.

[49] Fionn Murtagh and Pedro Contreras. "Algorithms for hierarchical clustering: an overview". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2.1 (2012), pp. 86–97.

[50] Elisa Mussumeci and Flávio Codeço Coelho. "Large-scale multivariate forecasting models for Dengue-LSTM versus random forest regression". In: *Spatial and Spatio-temporal Epidemiology* 35 (2020), p. 100372.

[51] Alex Nanopoulos, Rob Alcock, and Yannis Manolopoulos. "Feature-based classification of time-series data". In: *International Journal of Computer Research* 10.3 (2001), pp. 49–61.

[52] Godwin Ogbuabor and FN Ugwoke. "Clustering algorithm for a healthcare dataset using silhouette score value". In: *International Journal of Computer Science & Information Technology* 10.2 (2018), pp. 27–37.

[53] Christopher Olah. *Understanding LSTM Networks*. Aug. 2015. URL: `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`.

[54] Antonio Rafael Sabino Parmezan, Vinicius MA Souza, and Gustavo EAPA Batista. "Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model". In: *Information Sciences* 484 (2019), pp. 302–337.

[55] John S Ramberg et al. "A probability distribution and its uses in fitting data". In: *Technometrics* 21.2 (1979), pp. 201–214.

[56] Khandakar M Rashid and Joseph Louis. "Times-series data augmentation and deep learning for construction equipment activity recognition". In: *Advanced Engineering Informatics* 42 (2019), p. 100944.

[57] Aldo Ismael Rodrıguez-Castro, Jorge Rolón, and Carlos Miguel Rıos-González. "Dengue internation costs in a third level hospital of attention of Paraguay 2017". In: *Revista del Instituto de Medicina Tropical* 14.1 (2019), pp. 14–20.

[58] David Romero et al. "Applying fuzzy logic to assess the biogeographical risk of dengue in South America". In: *Parasites & vectors* 12.1 (2019), p. 428.

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

[60] Jürgen Schmidhuber. "Deep learning in neural networks: An overview". In: *Neural networks* 61 (2015), pp. 85–117.

[61] Maxim Vladimirovich Shcherbakov et al. "A survey of forecast error measures". In: *World Applied Sciences Journal* 24.24 (2013), pp. 171–176.

[62] Yuan Shi et al. "Three-month real-time dengue forecast models: an early warning system for outbreak alerts and policy decision support in Singapore". In: *Environmental health perspectives* 124.9 (2016), pp. 1369–1375.

[63] Connor Shorten and Taghi M Khoshgoftaar. "A survey on image data augmentation for deep learning". In: *Journal of Big Data* 6.1 (2019), p. 60.

[64] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. "A comparison of ARIMA and LSTM in forecasting time series". In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2018, pp. 1394–1401.

[65] Fabrıcio Drummond Silva et al. "Temporal relationship between rainfall, temperature and occurrence of dengue cases in São Luus, Maranhão, Brazil". In: *Ciencia & saude coletiva* 21 (2016), pp. 641–646.

[66] Alex J Smola and Bernhard Schölkopf. "A tutorial on support vector regression". In: *Statistics and computing* 14.3 (2004), pp. 199–222.

[67] Gustavo Sosa-Cabrera et al. "A multivariate approach to the symmetrical uncertainty measure: Application to feature selection problem". In: *Information Sciences* 494 (2019), pp. 1–20.

[68] Angel Stanoev, Daniel Trpevski, and Ljupco Kocarev. "Modeling the spread of multiple concurrent contagions on networks". In: *PloS one* 9.6 (2014), e95669.

[69] Stephen M Stigler. "Who discovered Bayes's theorem?" In: *The American Statistician* 37.4a (1983), pp. 290–296.

[70] Chen Sun et al. "Revisiting unreasonable effectiveness of data in deep learning era". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 843–852.

[71] Dongdong Sun, Minghui Wang, and Ao Li. "A multimodal deep neural network for human breast cancer prognosis prediction by integrating multi-dimensional data". In: *IEEE/ACM transactions on computational biology and bioinformatics* 16.3 (2018), pp. 841–850.

[72] B Üstün et al. "Determination of optimal support vector regression parameters by genetic algorithms and simplex optimization". In: *Analytica Chimica Acta* 544.1-2 (2005), pp. 292–305.

[73] Tazio Vanni, Rosa Legood, and Richard G White. "Calibration of disease simulation model using an engineering approach". In: *Value in Health* 13.1 (2010), pp. 157–157.

[74] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[75] Jason Wang and Luis Perez. "The effectiveness of data augmentation in image classification using deep learning". In: *Convolutional Neural Networks Vis. Recognit* (2017).

[76] Lijing Wang, Jiangzhuo Chen, and Madhav Marathe. "TDEFSI: theory-guided deep learning-based epidemic forecasting with synthetic information". In: *ACM Transactions on Spatial Algorithms and Systems (TSAS)* 6.3 (2020), pp. 1–39.

[77] William WS Wei. "Time series analysis". In: *The Oxford Handbook of Quantitative Methods in Psychology: Vol. 2*. 2006.

[78] Qingsong Wen et al. "Time Series Data Augmentation for Deep Learning: A Survey". In: *arXiv preprint arXiv:2002.12478* (2020).

[79] Guikai Xi et al. "A deep residual network integrating spatial-temporal properties to predict influenza trends at an intra-urban scale". In: *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*. 2018, pp. 19–28.

[80] Jiucheng Xu et al. "Forecast of Dengue Cases in 20 Chinese Cities Based on the Deep Learning Method". In: *International Journal of Environmental Research and Public Health* 17.2 (2020), p. 453.

[81] Chao-Tung Yang et al. "Influenza-like illness prediction using a long short-term memory deep learning model with multiple open data sources". In: *The Journal of Supercomputing* (2020), pp. 1–27.

[82] Jie Zhang and Kazumitsu Nawata. "A comparative study on predicting influenza outbreaks". In: *Bioscience trends* (2017).

Appendix A

Expanded summary in Spanish

Mejora del pronóstico de casos de Dengue en Paraguay con aprendizaje profundo mediante agrupación de series temporales y técnicas de aumento de datos

A.1 Introducción

El Dengue es una enfermedad viral transmitida por mosquitos que se transmite al ser humano principalmente por el mosquito *Aedes aegypti* como vector, con mayor incidencia en las zonas urbanas.Durante la última década, ha habido un aumento dramático de las infecciones por dengue en países de América del Sur como Colombia, Ecuador, Paraguay, Perú, Venezuela y Brasil. También se sabe que el dengue tiene una característica endémica, por lo que se considera un problema de salud pública en las regiones tropicales y subtropicales.

En Paraguay, luego de la primera epidemia de dengue en el período 1989-1990, no se reportaron brotes durante una década, hasta un segundo gran brote en 2007. A partir de 2009 se observa una circulación constante, reportando entre los años 2009 a 2015 un aumento sostenido de casos y una tercera gran epidemia en 2013, año en el que se observaron 153.793 casos notificados.

Actualmente, la lucha contra el Dengue se basa en una adecuada atención clínica y de laboratorio, vigilancia epidemiológica y campañas educativas con programas de control de vectores como estrategia básica para mitigar la propagación del Dengue. Sin embargo, no se ha tenido éxito y en ausencia de una estrategia más eficaz, por ejemplo, la introducción de una vacuna efectiva, esta enfermedad seguirá produciendo una considerable carga económica y social. La correcta aplicación de las medidas de control dependen del manejo del inicio de la temporada de dengue. Dado que los brotes varían a lo largo de los años, los pronósticos precisos pueden ser herramientas fundamentales en la lucha contra la enfermedad.

La mayoría de los modelos compartimentales, como el Modelo SIR, están restringidos a caracterizar los datos sólo para un brote epidémico y no tiene en cuenta otras variables como las climáticas. Esta es la razón por la que los enfoques basados en el aprendizaje automático y el aprendizaje profundo se han convertido en alternativas competitivas a los modelos tradicionales al considerar la incidencia de una enfermedad como un problema de predicción de series temporales. Compren-

der el comportamiento de la enfermedad es una compleja combinación de factores epidemiológicos y ambientales, y es una tarea difícil para los métodos de regresión clásicos. En este contexto, los modelos basados en el aprendizaje profundo han demostrado tener mejores o iguales resultados que los modelos estadísticos, además de permitir manejar más variables externas de una manera relativamente más fácil. Enfoques de aprendizaje profundo, en concreto LSTM (Long Short Term Memory) propuesto por Hochreiteret ha demostrado que pueden superar los modelos de la literatura y se ha utilizado para pronosticar con éxito las tendencias de la influenza. Sin embargo, para lograr resultados óptimos con modelos de aprendizaje profundo, se necesita una gran cantidad de datos y la falta de datos a largo plazo afecta el rendimiento de estos modelos produciendo un sobreajuste.

Este trabajo investiga qué modelo realiza mejores predicciones en el caso de las epidemias de Dengue, considerando los modelos de aprendizaje automático tradicionales frente a los de aprendizaje profundo. Una vez seleccionado el mejor candidato, se consideran dos estrategias bien conocidas en la literatura para mejorar la predicción del modelo, es decir, agrupación y el aumento de datos. El componente ambiental dr se explora agrupando datos para entrenamiento basado en su similitud y se explora el componente epidemiológico al aplicar una combinación de modelos matemáticos (modelo SIR) e inferencia Bayesiana para aumentar artificialmente los datos. La contribución de la agrupación es la identificación por áreas geográficas del comportamiento de la enfermedad y la reducción del tamaño de los modelos necesarios para cubrir el país. La contribución de las técnicas de aumento de datos bayesianos en modelos matemáticos epidemiológicos es proporcionar información completa sobre los eventos epidemiológicos observados. Además, ambas técnicas pueden utilizarse como regularizadores para evitar un sobreajuste en los modelos.

A.2 Objetivos

A.2.1 Objetivo general

1. Proponer estrategias para mejorar la precisión de los modelos de Dengue basadas en el aprendizaje profundo mediante la aplicación de agrupaciones de series de tiempo y aumento de datos.

A.2.2 Objetivos específicos

1. Evaluar los modelos tradicionales de aprendizaje profundo y de máquina para seleccionar un modelo de referencia para pronosticar la incidencia del dengue.
2. Analizar qué métodos de agrupación de series de tiempo se pueden utilizar para simplificar los modelos de pronóstico de dengue.
3. Proponer un nuevo enfoque de aumento de datos bayesiano basado en datos sintéticos generados por un modelo compartimental.
4. Evaluar el aumento de datos de series de tiempo tradicionales frente al enfoque bayesiano propuesto.
5. Cuantificar la mejora de los métodos basados en agrupamiento y en aumento de datos.

A.3 Propuestas

A.3.1 Agrupamiento de series temporales

La agrupación en clústeres es una tarea de aprendizaje automático no supervisada que tiene como objetivo clasificar en grupos una gran cantidad de datos cuando no hay conocimiento previo sobre grupos reales. Las particiones en grupos se hacen de tal manera que los elementos de un grupo sean lo más similares posible entre sí.

Dado que no hay ninguna pista sobre a qué clases debe pertenecer la serie, existen varias incertidumbres al agrupar, como determinar el número de grupos, definir las métricas de disimilitud y, si están basadas en características, determinar cuáles son las más relevantes. Hay varias opciones propuestas para hacer frente a estas incertidumbres, y dependen del enfoque de agrupamiento que se realice. Se utilizaron dos enfoques principales para el agrupamiento de series de tiempo:

1. Basado en distancia, directamente con distancias en puntos de datos sin procesar.

2. Basado en características, indirectamente con características extraídas de los datos sin procesar.

Las técnicas de agrupación también se clasifican según la forma en que realizan las particiones, teniendo así las basadas en centroide, las basadas en conectividad y las basadas en densidad, siendo las más representativas el algoritmo k-means, la agrupación jerárquica y DBScan respectivamente.

A menos que se conozca de antemano el número de grupos necesarios, determinar el número óptimo de grupos (k) es una tarea compleja. Este es un problema frecuente en la agrupación de datos, ya que es un parámetro de entrada que se necesita para algunos algoritmos de agrupación, y no hay una respuesta segura, sin embargo, existen técnicas que ayudan a inferir el número óptimo de grupos, tales como: el método del codo y la evaluación de silueta. En este trabajo se utilizó la evaluación de silueta.

Aunque todos los algoritmos de agrupamiento necesitan calcular la distancia, el enfoque basado en la distancia toma las distancias entre cada par de datos brutos, es decir, las distancias se miden directamente entre series de tiempo. La distancia se utiliza para determinar qué tan cerca están un par de observaciones, las observaciones más cercanas son más similares y, por lo tanto, pueden pertenecer al mismo grupo. El enfoque de agrupación en clústeres basado en características implica el uso de las características más importantes de cada serie temporal y la realización de agrupaciones en función de esas características.

Cuando lo que se quiere es modelar casos de Dengue en varias ciudades, se puede ver que algunas tienen comportamientos similares, por eso se propone agruparlos. Para aplicar la agrupación de series de tiempo, se requieren definiciones de varios parámetros, pero estos varían según cada problema, por lo que se realizaron experimentos previos. Entonces, en este trabajo se propuso realizar el agrupamiento de series de tiempo de casos de dengue, para ello se probaron los enfoques basados en la distancia y basados en características, ya que no existe un modelo que se pueda asumir que genere la serie temporal de casos de dengue. El número de grupos es

un parámetro de entrada necesario para algunos algoritmos, utilizando el método del codo se calcula el número de grupos a formar. Las medidas de disimilitud más adecuadas se determinan mediante experimentos, combinando las métricas con diferentes técnicas de agrupamiento. Finalmente, para decidir cuál obtuvo los mejores resultados, los resultados se validan con una métrica de evaluación interna (evaluación de silueta). Para utilizar este enfoque se necesitan varias series de tiempo para poder agruparlas, en caso de que estas series de datos no estén disponibles, se pueden utilizar otras técnicas para generar datos sintéticos basados en observaciones individuales, como las derivadas de la inferencia estadística.

A.3.2 Aumento de datos

La idea básica del aumento de datos es generar un conjunto de datos sintéticos que cubra el espacio de entrada inexplorado manteniendo las etiquetas correctas. De esta forma, se busca reducir el sobreajuste utilizando los datos sintéticos como regularizador al entrenar un modelo.

Este trabajo busca evaluar la mejor técnica de aumento de datos, las técnicas tradicionales serán comparadas con otras técnicas más avanzadas basadas en estadísticas como la inferencia Bayesiana.

A.3.2.1 Tradicional

El enfoque tradicional para el aumento de datos es aplicar pequeños cambios de la serie original para generar los sintéticos. Recientemente, se ha aplicado el aumento de datos con métodos similares a los aplicados en el procesamiento de imágenes. Pero no todas las técnicas aplicadas a imágenes se pueden aplicar a series de tiempo debido a las diferencias entre los tipos de datos.

Para este trabajo se utilizarán técnicas basadas en agregar ruido, desplazar las observaciones y escalar la serie. Estas técnicas se describen a continuación:

1. *Ruido.* Desde la serie original, otras se generan agregándoles ruido blanco. El ruido blanco es un proceso estocástico donde sus variables no están correlacionadas, una señal de ruido blanco tiene media cero. Luego, se genera una señal de ruido blanco tomando valores aleatorios de una distribución normal con $\mu = 0$ y $\sigma = 1$, luego este valor se suma a la observación original, así se obtiene la serie con ruido. Recuerde que μ es la media y σ es la desviación estándar.

2. *Desplazar.* Esta función translada los valores por un factor de i pasos, es decir: $y_t = y_{t+i}\ \forall y_t \in Y$, donde Y es la serie de tiempo. Para estos experimentos $i \in [-5, 5]$.

3. *Escalar.* Esta función consiste en usar un factor k en la serie de tiempo, es decir: $y_t = y_t * k\ \forall y \in Y$, donde Y es el observado series de tiempo. Para estos experimentos $k \in [-0.5, 0.5]$.

Este trabajo busca evaluar la mejor técnica de aumento de datos, estas técnicas serán comparadas con otras técnicas más avanzadas basadas en criterios estadísticos (aumento de datos Bayesiano).

A.3.2.2 Bayesiano

La inferencia bayesiana es una técnica de inferencia estadística basada en el teorema de Bayes. El teorema de Bayes está estrechamente relacionado con el concepto de probabilidad condicional. La probabilidad condicional de un evento A es la probabilidad de que el evento ocurra sabiendo que ya ocurrió un evento B. Esta probabilidad se escribe $P(A \mid B)$, lo que significa probabilidad de A dado B. En el caso de que el evento B no tenga efecto sobre la probabilidad del evento A, la probabilidad condicional del evento A es simplemente la probabilidad del evento A, es decir, $P(A)$. A partir de esta definición, la probabilidad condicional se describe como

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}, \tag{A.1}$$

donde $P(A \cap B)$ es la probabilidad de que A y B ocurran al mismo tiempo, y $P(B)$ es la probabilidad de que ocurra B. El término teorema de Bayes es en honor al reverendo Thomas Bayes, y también se le conoce como ley de Bayes. Este teorema muestra la probabilidad condicional o probabilidad posterior, o simplemente *posterior* de un evento A después de que B se observe en términos de la probabilidad previa de A, la probabilidad previa de B y la probabilidad condicional de B dado A. El teorema de Bayes se define de la siguiente manera

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}, \tag{A.2}$$

donde $P(A)$ es la probabilidad de que ocurra A, $P(B)$ es la probabilidad de que ocurra B, $P(B \mid A)$ es la probabilidad de que ocurra B dado A y $P(A \mid B)$ es la probabilidad de A dado B. El teorema de Bayes se basa en incorporar distribuciones de probabilidad previa $P(A)$ para generar probabilidades posteriores $P(A \mid B)$, para mostrar cuán verdadera es una hipótesis, según la evidencia. La base de la inferencia bayesiana proviene del teorema de Bayes, para aplicarlo a modelos se hace la siguiente modificación a la ecuación anterior:

$$P(\Theta \mid y) = \frac{P(y \mid \Theta)P(\Theta)}{P(y)}, \tag{A.3}$$

donde y son observaciones y Θ es un conjunto de parámetros para un modelo, entonces se definen los siguientes componentes:

- $P(\Theta)$ es el conjunto de distribuciones previas del conjunto de parámetros Θ antes de que se observe y.
- $P(y|\Theta)$ es la probabilidad de y bajo un modelo.
- $P(\Theta|y)$ es la distribución posterior completa del conjunto de parámetros Θ que expresa incertidumbre sobre el conjunto de parámetros Θ después de tener en cuenta tanto el anterior como los datos.
- $P(y)$ se define como $\int P(y \mid \Theta)P(\Theta)\, d\Theta$.

Dado que generalmente hay múltiples parámetros, Θ representa un conjunto de j parámetros que se pueden considerar así

$$\Theta = \theta_1, \theta_2, ..., \theta_j, \tag{A.4}$$

Con este enfoque, la inferencia bayesiana se puede utilizar para aplicarla a un modelo de pronóstico para casos de dengue, por ejemplo, el SIR, y obtener una distribución de probabilidad de sus parámetros.

La inferencia bayesiana permite obtener una distribución de parámetros de un modelo, esta distribución se denomina posterior. Al tomar muestras de esta distribución, se pueden obtener simulaciones similares a las observaciones. En este trabajo, el modelo utilizado para caracterizar los brotes de dengue es el modelo SIR.

El modelo SIR, proporciona información sobre la situación de los individuos susceptibles, infectados y recuperados, en el contexto de los casos de dengue. En este trabajo, sólo se utilizará información sobre individuos infectados.

Recordando el concepto de inferencia bayesiana basada en modelos, es posible inferir la distribución de parámetros para generar datos similares a las observaciones. Con esta distribución de parámetros, es posible realizar simulaciones y obtener datos similares a los observados. La idea central de este experimento es utilizar estas simulaciones para mejorar el rendimiento de una red LSTM, ya que cuantos más datos, mejor será el rendimiento de la red.

Sin embargo, el modelo SIR da como resultado una sola curva, que representa un solo brote epidémico, y las observaciones en forma de serie de tiempo se ingresan como entrada en la red LSTM como un vector $Y = [y_1, y_2, ..., y_t]$ con t observaciones semanales. Para realizar la inferencia mediante el modelo SIR, se debe extraer cada pico de la serie temporal.

Un brote es el aumento significativo de casos en relación con los valores habitualmente observados. Cuando los brotes ocurren estacionalmente, se dice que la enfermedad que los causa es endémica, como el dengue en Paraguay. De esta manera, cada pico dentro de la serie temporal representa un brote.

Básicamente se trata de encontrar el máximo local en la serie de tiempo, ya que un brote dura aproximadamente 40 semanas, ese es el valor de *range* que se utilizó en el algoritmo. De 2009 a 2013, hay cinco picos o menos según la ciudad. Una vez que se han identificado los picos, el siguiente paso es determinar el comienzo y el final del brote. A esto se le llama encontrar el ancho del brote.

Luego, para cada ciudad, se lleva a cabo el proceso de búsqueda de picos y sus anchos una vez que tenemos las observaciones en forma de brotes individuales se pueden ajustar a un modelo SIR.

Así, se puede definir una versión modificada del modelo SIR para series de tiempo que trabajen por temporadas, este modelo se denominará `SeasonalSIR`. `SeasonalSIR` recibe una serie temporal, encuentra los brotes y el inicio y final de cada uno, para cada brote se ejecuta el modelo SIR tradicional, cada brote se considera una temporada, una temporada con casos activos le sigue otra sin casos , para manejar esto, el modelo `SeasonalSIR` baja artificialmente el valor β a cero. Con una tasa de infección de $\beta \leq 0$, no se produce un brote. De esta manera, el modelo `SeasonalSIR` puede devolver una serie de tiempo de casos de dengue multiestacional. El algoritmo 12 muestra cómo funciona la función `+SeasonalSIR+,`*start* y *end* son vectores que tienen el comienzo y el final de cada temporada o brote respectivamente, *seasons* es un vector con un índice para cada temporada de la serie, β es un vector que contiene los valores de β_s para cada temporada y γ es un vector con los valores γ_s para cada temporada, $SIR()$ ejecuta el modelo SIR habitual. Los vectores *inicio*, fin y *picos* se obtienen de cada serie de tiempo que se va a ajustar. Los vectores β y γ son los vectores que se obtendrán al realizar la inferencia

bayesiana. La longitud de cada vector es igual al número de temporadas que tiene la serie.

Function `SeasonalSIR`($inicio, fin, temporada, \beta, \gamma$)**:**
  foreach *temporada* **do**
    $t = inicio - fin$
    $TS \longleftarrow SIR(\beta, \gamma, t)$
  end
  return TS
End Function

Algorithm 12: SeasonalSIR

Todas las técnicas propuestas anteriormente (agregar ruido, escala, desplazamiento, aumento de datos bayesianos) serán aplicadas a la base de datos de dengue, antes de ser entrenadas con un modelo de aprendizaje automático, luego se evaluará el desempeño de cada una y se comparará si existen mejoras en relación al modelo sin datos aumentados.

A.4 Experimentos

En este trabajo, se han probado técnicas de agrupamiento de series de tiempo y aumento de datos para mejorar el rendimiento de un modelo de aprendizaje profundo para pronosticar los casos de dengue en Paraguay.

Para la representación, se seleccionaron 5 ciudades aleatorias pertenecientes a cada grupo (Grupo 1: alta incidencia, Grupo 2: incidencia media, Grupo 3: baja incidencia). Los experimentos se llevaron a cabo en las ciudades seleccionadas.

Para la realización de los experimentos se comparó el desempeño de las técnicas tradicionalmente utilizadas para la predicción de series de tiempo (SVR, Random Forest, LARS LASSO, LSTM), con el modelo de aprendizaje profundo LSTM de mejor desempeño. Por lo tanto, se seleccionó LSTM como modelo de referencia, es decir, modelo de referencia para los demás experimentos. Utilizando el modelo LSTM, se buscó mejorar su desempeño. En los experimentos, el modelo de referencia se denota como *Single*.

Luego, los modelos LSTM se entrenaron de diferentes maneras según cada enfoque:

1. Agrupación de series de tiempos. Este enfoque buscaba mejorar el rendimiento de la red entrenando el modelo LSTM en grupos en serie, teniendo así los siguientes modelos:

   (a) *Departamento*. El modelo LSTM se entrenó con las series agrupadas según el departamento al que pertenecen según la división política del país.

   (b) *País*. El modelo LSTM se entrenó con todas las series del país.

   (c) *Cluster*. El modelo LSTM se entrenó de acuerdo con la agrupación resultante de las técnicas de agrupación. En este punto, se realizó un estudio previo para determinar cuál es la técnica de clustering más adecuada. El número de clusters a formar se determinó con el método de Elbow y se probaron varias técnicas de clustering (k-means, jerárquico, DBscan), cada una con un conjunto de métricas de distancia (Euclidean,

Correlation, Pearson Correlation, Dynamic time warping) y los resultados evaluados utilizando la puntuación de silueta. Los mejores resultados se obtuvieron utilizando agrupamiento jerárquico y correlación, con esta técnica y métrica se formaron los algoritmos para entrenar este modelo.

Entre estos modelos, el que tuvo el mejor rendimiento general fue el modelo *Cluster*. La mejora se observa especialmente en los grupos 1 y 3.

2. Aumento de datos. Este enfoque busca mejorar el rendimiento de la red aumentando la cantidad de datos generando datos sintéticos a partir de los datos observados. Los modelos se formaron utilizando un enfoque básico e inferencia bayesiana

   (a) Enfoques básicos. Este enfoque consiste en aplicar pequeñas transformaciones a los datos observados para generar nuevos, los modelos utilizados fueron:

      i. *Ruido*. Este modelo LSTM se entrenó con variaciones de ruido aleatorias agregadas a las observaciones.
      ii. *Dezplazar*. Este modelo LSTM se entrenó con variaciones aleatorias de la serie desplazada.
      iii. *Escala*. Este modelo LSTM se entrenó con variaciones de la serie multiplicadas por un escalar aleatorio.

   Entre estos modelos, el que tuvo el mejor desempeño promedio fue *Scale*, sin embargo su desempeño es mucho peor que el modelo *Cluster*.

   (b) Inferencia bayesiana. El modelo *Bayesiano* LSTM se entrenó con las simulaciones obtenidas de la distribución de parámetros obtenidos del modelo propuesto. El modelo es una versión modificada del modelo SIR adaptado para varias temporadas.

   El modelo *Bayesian* supera bastante a *Scale*, el mejor del enfoque básico, lo que hace que el modelo *Bayesian* sea el mejor entre las técnicas de aumento de datos.

Se ha demostrado que ambos modelos (*Cluster* y *Bayesian*) mejoran significativamente el rendimiento de un modelo de pronóstico de series de tiempo de aprendizaje profundo. El modelo *Cluster* no siempre es el mejor, especialmente en las ciudades de los grupos 2 y 3, pero es el modelo con la mejora más significativa. *Bayesian* es siempre el mejor modelo en todas las pruebas, pero tiende a sobreestimar los casos. En problemas en los que un modelo debe ajustarse a una observación, puede ocurrir el sobreajuste, lo que sucede cuando el modelo no se puede generalizar porque no tiene suficiente información. Por tanto, estas técnicas pueden considerarse métodos de regularización para evitar el sobreajuste.

A.5 Conclusiones y trabajos futuros

A.5.1 Conclusiones

El error cuadrático medio (RMSE) confirma que los modelos agrupados LSTM mejoran la precisión en $19.48 \pm 18.80\%$ y LSTM con aumento de datos basado en

Bayesiano mejora $16.86 \pm 16.57\%$. La principal contribución de este trabajo son dos técnicas que pueden mejorar el rendimiento de los modelos de series de tiempo al combinar información de series de tiempo y datos meteorológicos similares.

A.5.2 Trabajos futuros

En base a los resultados obtenidos, a continuación se presentan algunos trabajos futuros que se han identificado.

- Aplicar estas técnicas en un problema de clasificación de series de tiempo.
- Combina técnicas de clúster e inferencia bayesiana.
- Optimizar la agrupación de las series de tiempo diseñando más experimentos.
- Utilizar la inferencia bayesiana para obtener la estimación del número de casos reales en otras enfermedades.
- Utilizar otros modelos modernos de aprendizaje automático para pronosticar series de tiempo (Gated Recurrent Unit, Bidirectional recurrent neural networks, Deep transformer).
- Aplicar estos enfoques con otras enfermedades endémicas y transmitidas por vectores.

Appendix B

Dickey-Fuller test to check stationarity

Before applying any transformation must be determined if a series is stationary or not,a stationary time series is a stochastic process whose probability distribution at a fixed time instant or a fixed position is the same for all time instants or positions. Consequently, parameters such as mean and variance, if they exist, do not vary over time or position. To check stationarity, Augmented Dickey–Fuller test (ADF test) was used. The ADF test is a statistical significance test. That is, there is a hypothesis testing involved with a null and alternative hypothesis and as a result a test statistic is computed and p-values get reported. From the statistic test, an inference can be made as to whether a given time series is stationary or not. Unit root is a characteristic of a time series that makes it nonstationary. ADF test belongs to the unit root test. Technically , a unit root is said to exist in a time series of value of $\alpha = 1$ in the below equation.

$$y_t = \alpha y_{t-1} + \beta X_e + \epsilon \tag{B.1}$$

where y_t is the value of the time series at time t and X_e is an exogenous variable. The null hypothesis is $\alpha = 0$. The presence of a unit root means the time series is non-stationary. ADF test basically consists of performing the unit root test throughout the entire series. It is considered that with a p-value ≤ 0.05 the series is stationary [37]. Only 14 cities with non-stationary series where found, these cities were: Benjamin Aceval, Buena Vista, Chaco, General Aquino, Limoy Pueblo, Pozo Colorado, R. I. 3 Corrales, Roque González De Santa Cruz, San Pedro, San Rafael Del Paraná, San Roque González De Santacruz, Vaquería, Ybytymi, Ypane. To make predictions more effectively in these cities, a differentiation (as mentioned in chapter 2) must be performed first.

Table B.1 shows the detailed results of the ADF test.

Table B.1: Detailed results of the ADF test.

City	p**-value**
1RO DE MARZO	0.0002
25 DE DICIEMBRE	0.0000
3 DE FEBRERO	0.0000
ABAI	0.0074
ACAHAY	0.0000

Table B.1: Detailed results of the ADF test.

City	p**-value**
ALBERDI	0.0080
ALTO VERA	0.0000
ALTOS	0.0004
ANTEQUERA	0.0000
AREGUA	0.0026
ARROYOS Y ESTEROS	0.0009
ASUNCION	0.0037
ATYRA	0.0014
AYOLAS	0.0001
AZOTEY	0.0000
BAHIA NEGRA	0.0000
BELEN	0.0000
BELLA VISTA	0.0020
BENJAMIN ACEVAL	0.0901
BORJA	0.0069
BUENA VISTA	0.0832
CAACUPE	0.0004
CAAGUAZU	0.0234
CAAZAPA	0.0002
CABALLERO ALVAREZ	0.0024
CAMBYRETA	0.0033
CAPIATA	0.0140
CAPIIBARY	0.0092
CAPITAN BADO	0.0000
CAPITAN MEZA	0.0000
CAPITAN MIRANDA	0.0000
CARAGUATAY	0.0000
CARAPEGUA	0.0008
CARAYAO	0.0020
CARLOS ANTONIO LOPEZ	0.0000
CARMELO PERALTA	0.0047
CARMEN DEL PARANA	0.0000
CECILIO BAEZ	0.0000
CERRITO	0.0402
CHACO	0.2990
CHORE	0.0000
COLONIA FRAM	0.0000
COLONIA INDEPENDENCIA	0.0000
CONCEPCION	0.0000
CORONEL BOGADO	0.0042
CORONEL MARTINEZ	0.0004
CORONEL OVIEDO	0.0001
CORPUS CHRISTI	0.0023
CURUGUATY	0.0007
DESMOCHADOS	0.0000
DR BOTRELL	0.0000

Table B.1: Detailed results of the ADF test.

City	p**-value**
DR. JUAN MANUEL FRUTOS	0.0072
EDELIRA	0.0002
EMBOSCADA	0.0036
ENCARNACION	0.0123
ESCOBAR	0.0000
EUGENIO A GARAY	0.0019
EUSEBIO AYALA	0.0009
FASSARDI	0.0000
FELIX PEREZ CARDOZO	0.0000
FERNANDO DE LA MORA	0.0064
FILADELFIA	0.0069
FUERTE OLIMPO	0.0000
GENERAL AQUINO	0.1280
GENERAL ARTIGAS	0.0000
GENERAL BERNARDINO CABALLERO	0.0001
GENERAL BRUGUEZ	0.0053
GENERAL DELGADO	0.0000
GENERAL DIAZ	0.0000
GENERAL MORINIGO	0.0125
GENERAL RESQUIN	0.0000
GUARAMBARE	0.0021
GUAYAIBI	0.0000
GUAZUCUA	0.0000
HERNANDARIAS	0.0001
HOHENAU	0.0000
HORQUETA	0.0043
HUMAITA	0.0000
ISLA PUCU	0.0000
ISLA UMBU	0.0000
ITA	0.0172
ITACURUBI DE LA CORDILLERA	0.0007
ITACURUBI DEL ROSARIO	0.0000
ITAKYRY	0.0000
ITANARA	0.0000
ITAPE	0.0318
ITAPUA POTY	0.0086
ITAUGUA	0.0059
ITURBE	0.0002
J A SALDIVAR	0.0015
JESUS	0.0000
JOSE DOMINGO OCAMPOS	0.0000
JUAN DE MENA	0.0000
JUAN E. OLEARY	0.0000
JUAN EULOGIO ESTIGARRIBIA	0.0001
JUAN LEON MALLORQUIN	0.0003
KATUETE	0.0093

Table B.1: Detailed results of the ADF test.

City	p**-value**
LA PALOMA	0.0006
LA PASTORA	0.0020
LA VICTORIA	0.0166
LAMBARE	0.0230
LAURELES	0.0000
LEANDRO OVIEDO	0.0000
LIMA	0.0000
LIMOY PUEBLO	0.2990
LIMPIO	0.0092
LOMA GRANDE	0.0009
LOMA PLATA	0.0155
LORETO	0.0000
LUQUE	0.0086
MACIEL	0.0000
MARIANO ROQUE ALONSO	0.0305
MAURICIO JOSE TROCHE	0.0000
MBARACAYU	0.0000
MBOCAYATY	0.0000
MBOCAYATY DEL YHAGUY	0.0000
MCAL. ESTIGARRIBIA	0.0067
MCAL. FRANCISCO SOLANO LOPEZ	0.0000
MINGA GUAZU	0.0003
MINGA PORA	0.0002
MOISES BERTONI	0.0000
NANAWA	0.0000
NARANJAL	0.0000
NATALICIO TALAVERA	0.0000
NATALIO	0.0000
NUEVA ALBORADA	0.0000
NUEVA COLOMBIA	0.0003
NUEVA ESPERANZA	0.0000
NUEVA GERMANIA	0.0000
NUEVA ITALIA	0.0156
NUEVA LONDRES	0.0000
OBLIGADO	0.0001
PARAGUARI	0.0000
PASO YOBAI	0.0000
PEDRO JUAN CABALLERO	0.0009
PILAR	0.0364
PIRAPO	0.0000
PIRAYU	0.0028
PIRIBEBUY	0.0001
POZO COLORADO	0.4970
PUERTO FALCON	0.0023
PUERTO PINASCO	0.0000
QUIINDY	0.0000

Table B.1: Detailed results of the ADF test.

City	p**-value**
R I 3 CORRALES	0.8594
RAUL ARSENIO OVIEDO	0.0000
REPATRIACION	0.0053
ROQUE GONZALEZ DE SANTA CRUZ	0.0904
SALTO DEL GUAIRA	0.0000
SAN ALBERTO	0.0001
SAN ANTONIO	0.0044
SAN BERNARDINO	0.0000
SAN CARLOS	0.0203
SAN COSME Y DAMIAN	0.0000
SAN ESTANISLAO	0.0012
SAN IGNACIO	0.0089
SAN JOAQUIN	0.0000
SAN JOSE DE LOS ARROYOS	0.0001
SAN JOSE OBRERO	0.0000
SAN JUAN BAUTISTA	0.0034
SAN JUAN DEL PARANA	0.0083
SAN JUAN NEPOMUCENO	0.0002
SAN LAZARO	0.0000
SAN LORENZO	0.0490
SAN MIGUEL	0.0000
SAN PATRICIO	0.0000
SAN PEDRO	0.2376
SAN PEDRO DEL PARANA	0.0069
SAN PEDRO DEL YCUAMANDIYU	0.0484
SAN RAFAEL DEL PARANA	0.0703
SAN ROQUE GONZALEZ DE SANTACRUZ	0.2042
SAN SALVADOR	0.0004
SANTA ELENA	0.0002
SANTA MARIA	0.0000
SANTA RITA	0.0020
SANTA ROSA	0.0000
SANTA ROSA DEL AGUARAY	0.0040
SANTA ROSA DEL MBUTUY	0.0009
SANTA ROSA DEL MONDAY	0.0000
SANTIAGO	0.0000
SAPUCAI	0.0133
SIMON BOLIVAR	0.0002
TACUARAS	0.0064
TACUATI	0.0003
TAVAI	0.0005
TAVAPY	0.0021
TEBICUARY	0.0000
TEBICUARYMI	0.0000
TEMBIAPORA	0.0000
TOBATI	0.0000

Table B.1: Detailed results of the ADF test.

City	***p*-value**
TOMAS ROMERO PEREIRA	0.0001
TRINIDAD	0.0000
UNION	0.0000
VALENZUELA	0.0000
VAQUERIA	0.0622
VILLA DEL ROSARIO	0.0000
VILLA ELISA	0.0073
VILLA HAYES	0.0197
VILLA OLIVA	0.0000
VILLALBIN	0.0000
VILLARRICA	0.0025
VILLETA	0.0240
YAGUARON	0.0062
YATAITY	0.0015
YATAITY DEL NORTE	0.0000
YATYTAY	0.0000
YBY YAU	0.0000
YBYRAROVANA	0.0001
YBYTYMI	0.0884
YEGROS	0.0000
YGATIMI	0.0000
YGUAZU	0.0000
YHU	0.0091
YPACARAI	0.0000
YPANE	0.0531
YPEJHU	0.0000
YUTY	0.0000
ZANJA PYTA	0.0287

The ADF test can also be done visually, checking that the standard deviation and the mean have a constant trend. Figures B.1 to B.15 show the visual analyzes for the cities sampled in this work.

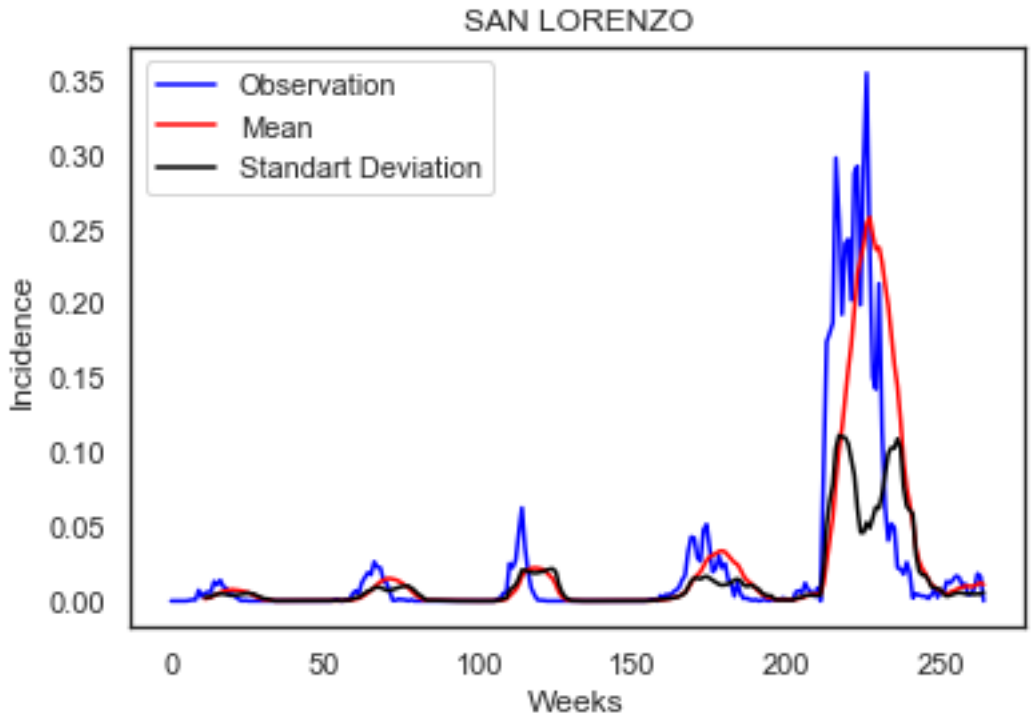


Figure B.1: ADF test for San Lorenzo city.

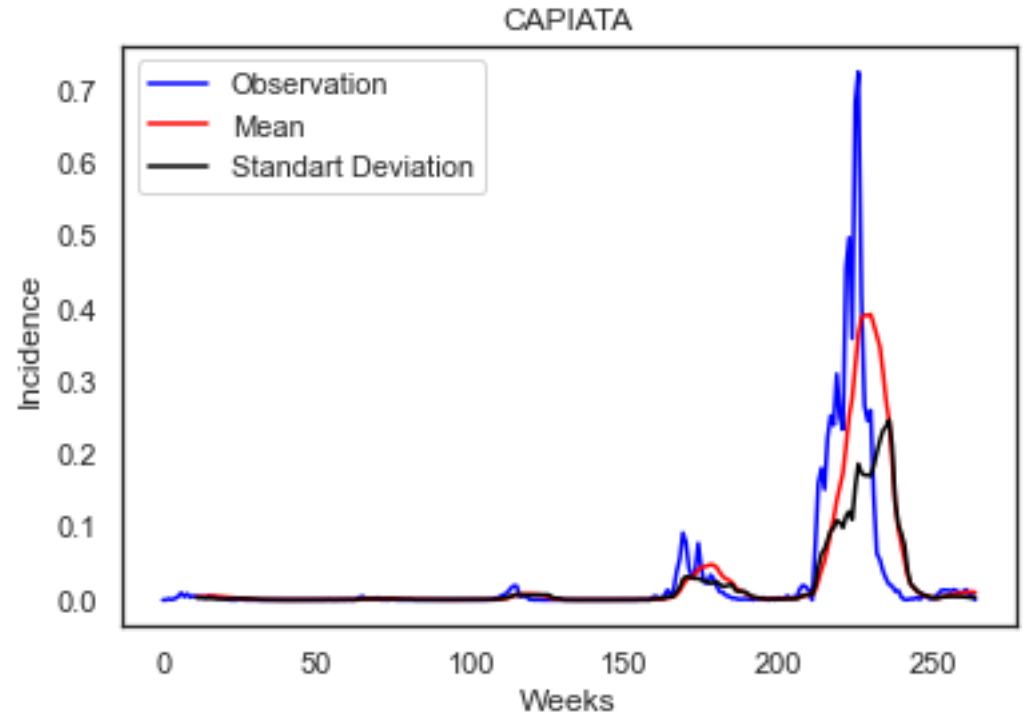


Figure B.2: ADF test for Capiatá city.

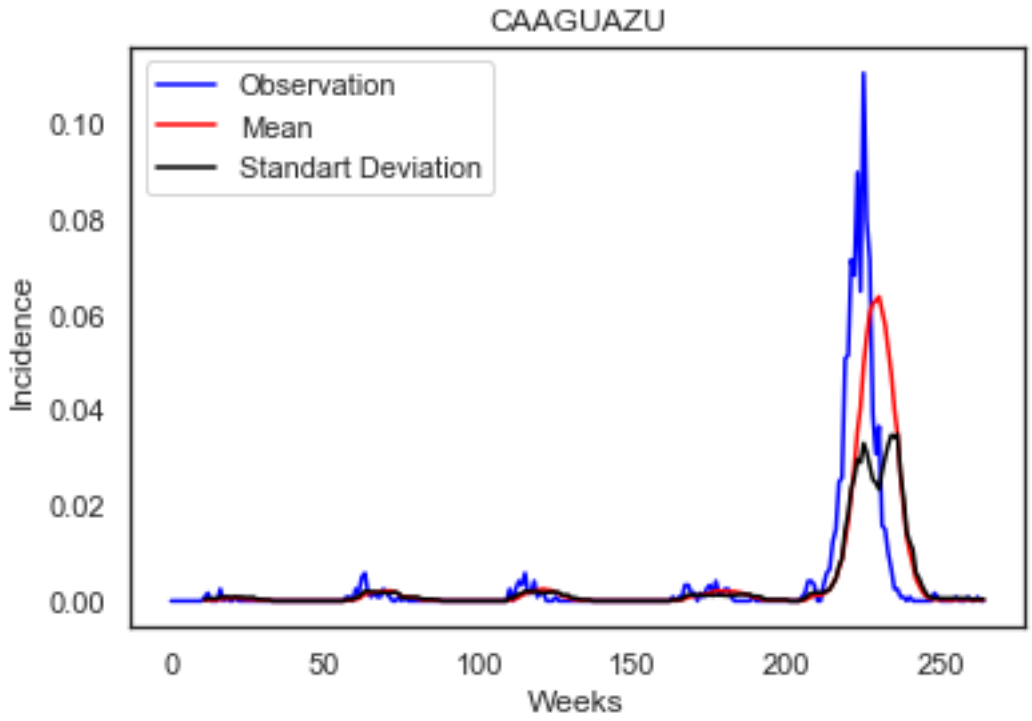


Figure B.3: ADF test for Caaguazú city.

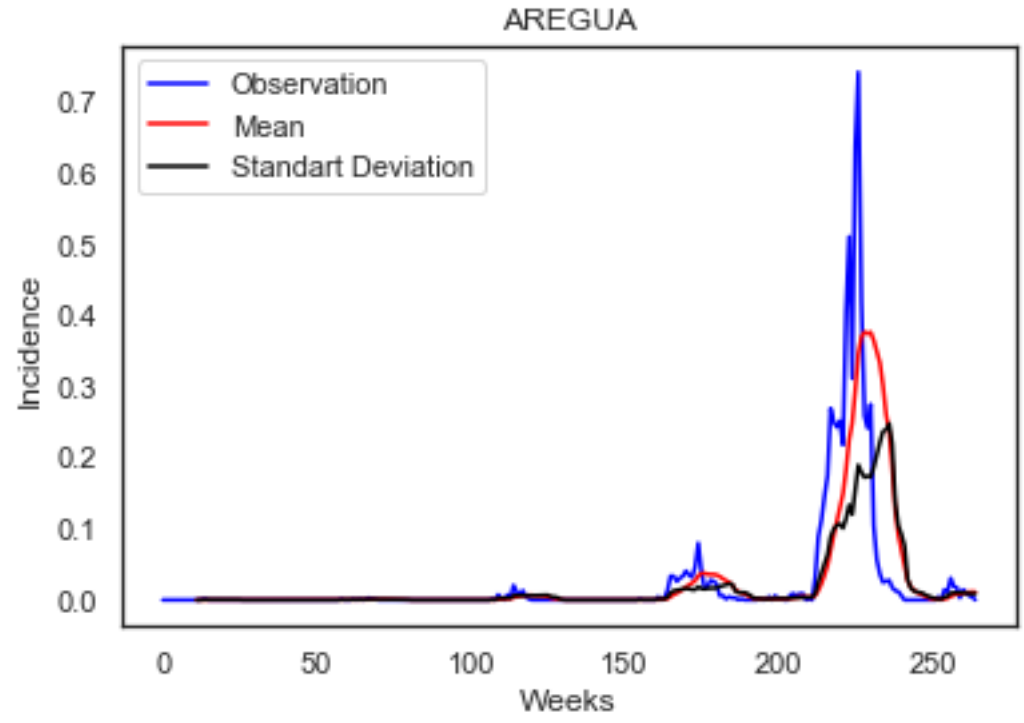


Figure B.4: ADF test for Areguá city.

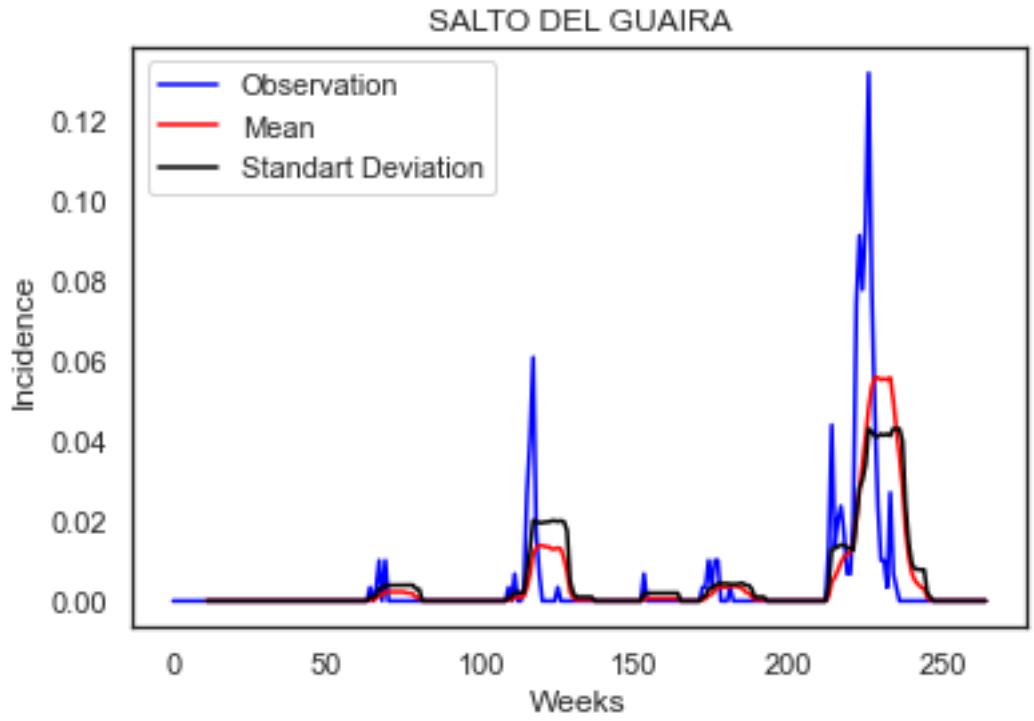


Figure B.5: ADF test for Salto del Guairá city.

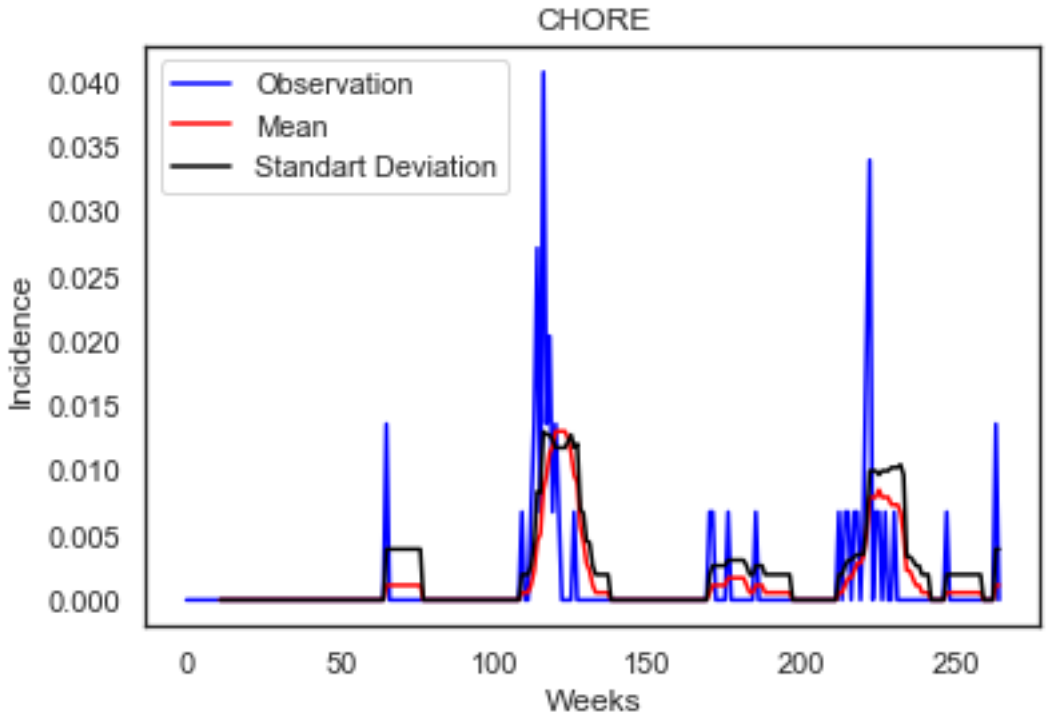


Figure B.6: ADF test for Choré city.

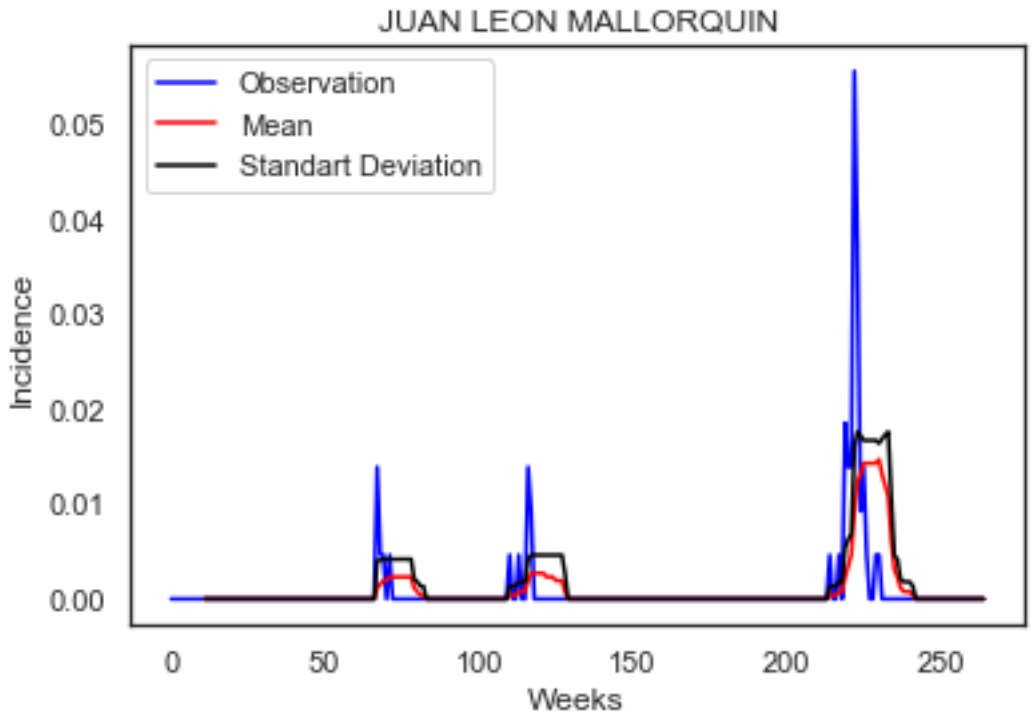


Figure B.7: ADF test for Juan León Mallorquín city.

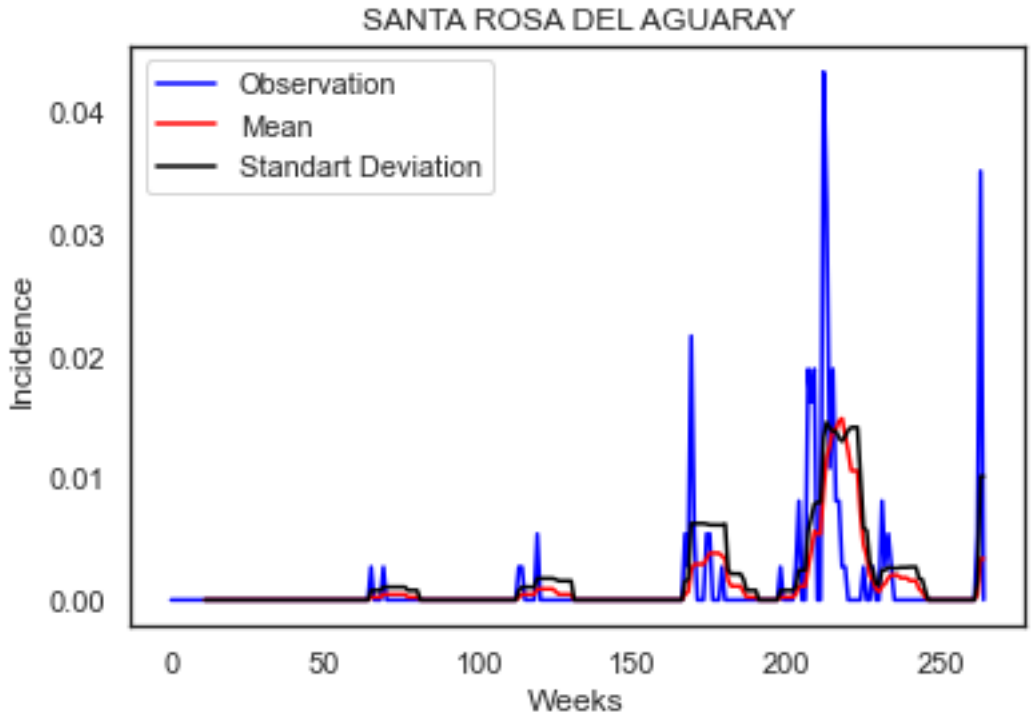


Figure B.8: ADF test for Santa Rosa del Aguaray city.

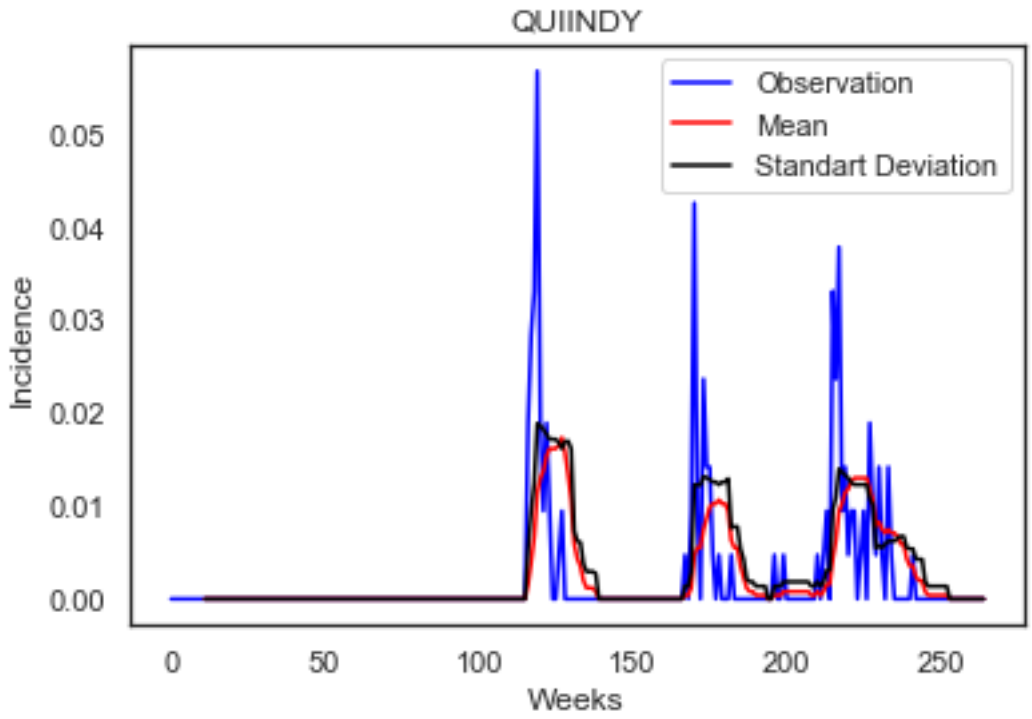


Figure B.9: ADF test for Quiindy city.

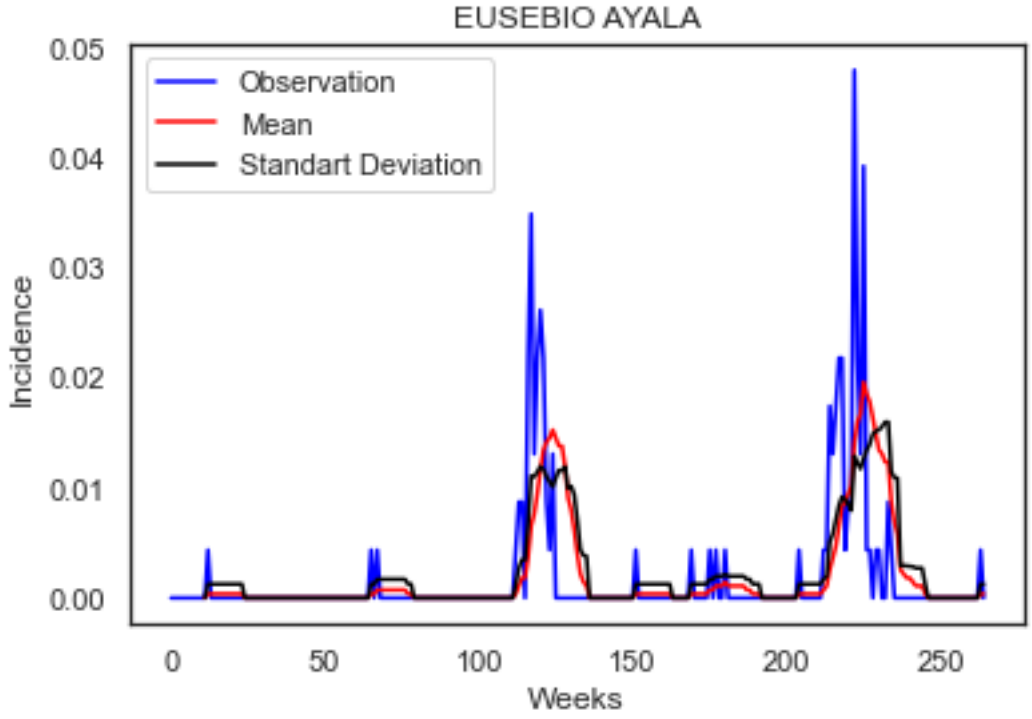


Figure B.10: ADF test for Eusebio Ayala city.

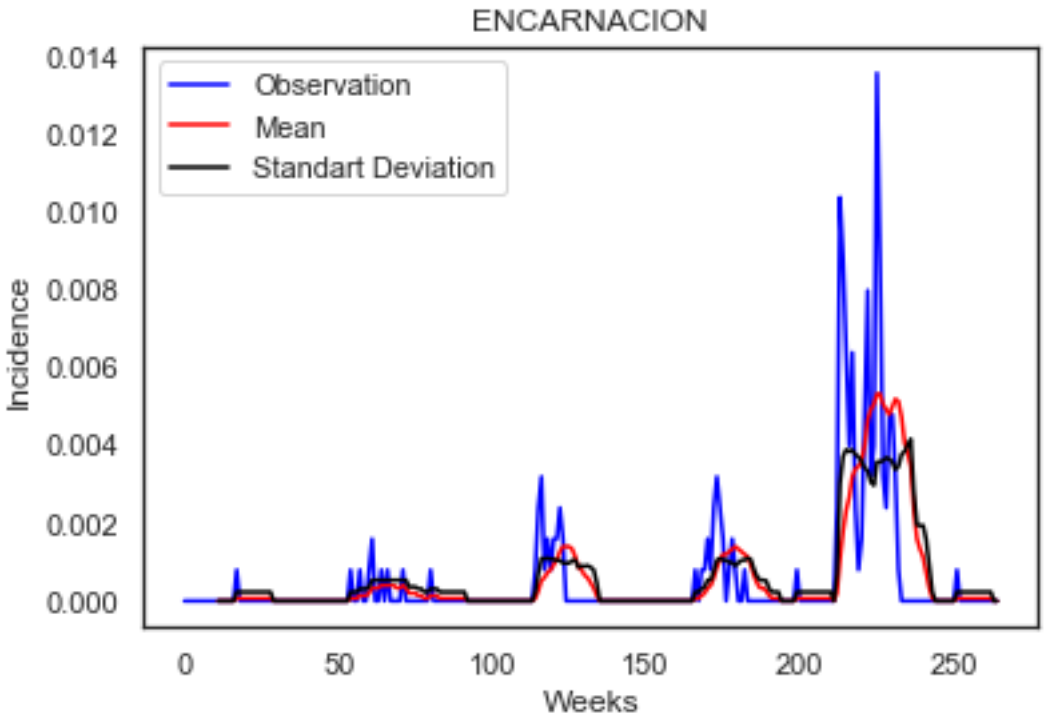


Figure B.11: ADF test for Encarnación city.

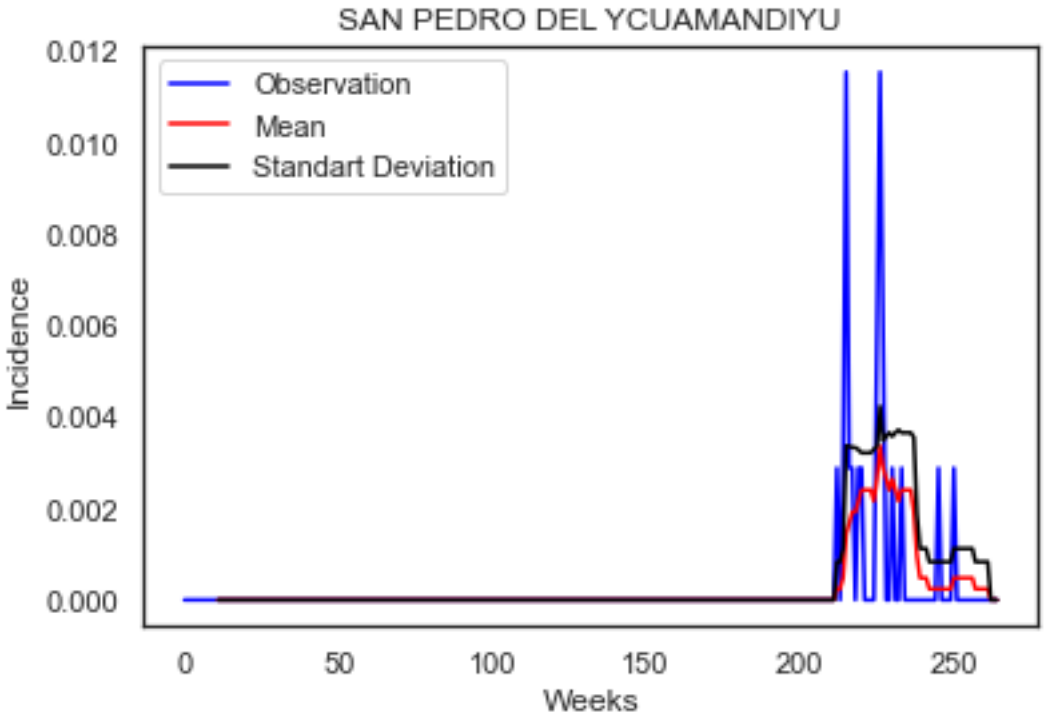


Figure B.12: ADF test for San Pedro del Ycuamandijú city.

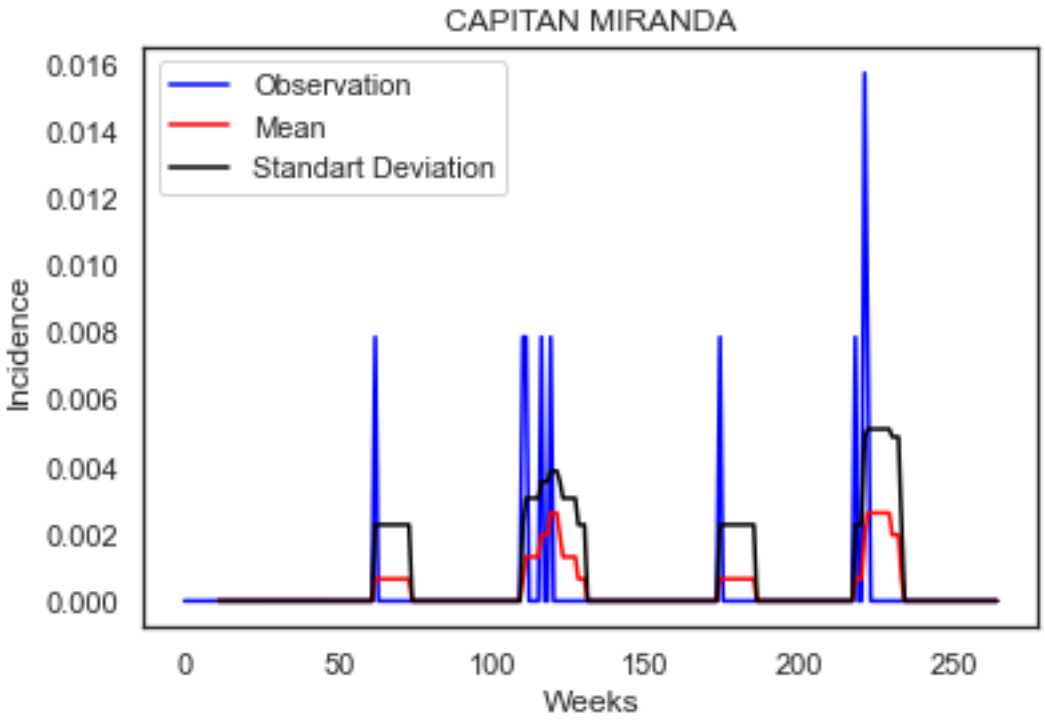


Figure B.13: ADF test for Capitán Miranda city.

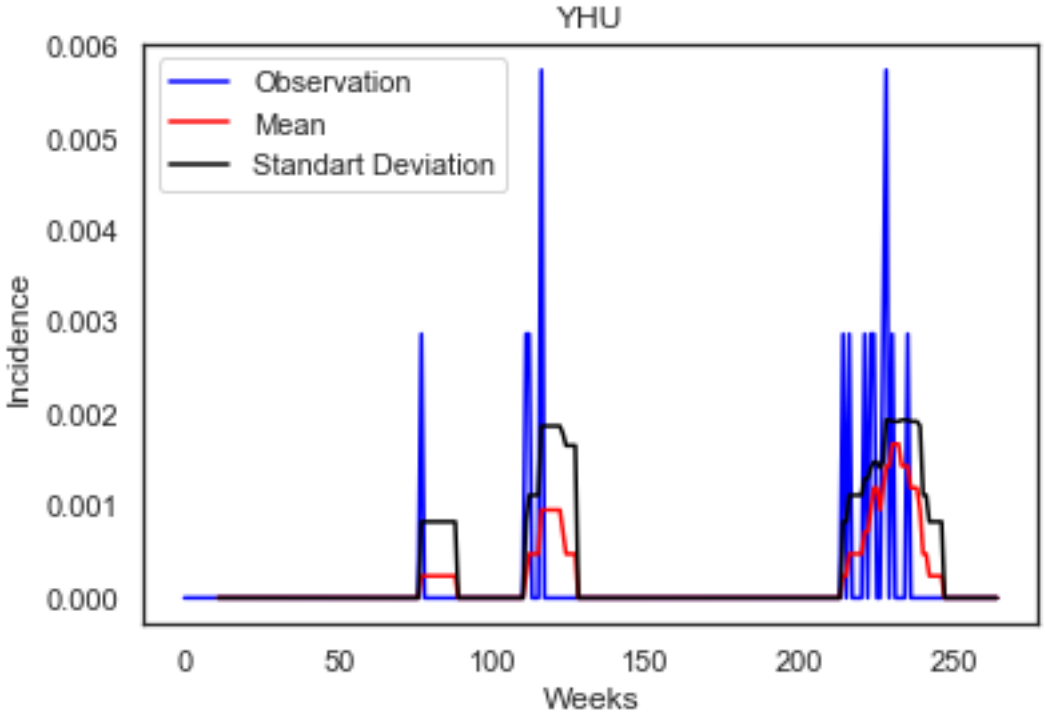


Figure B.14: ADF test for Yhú city.

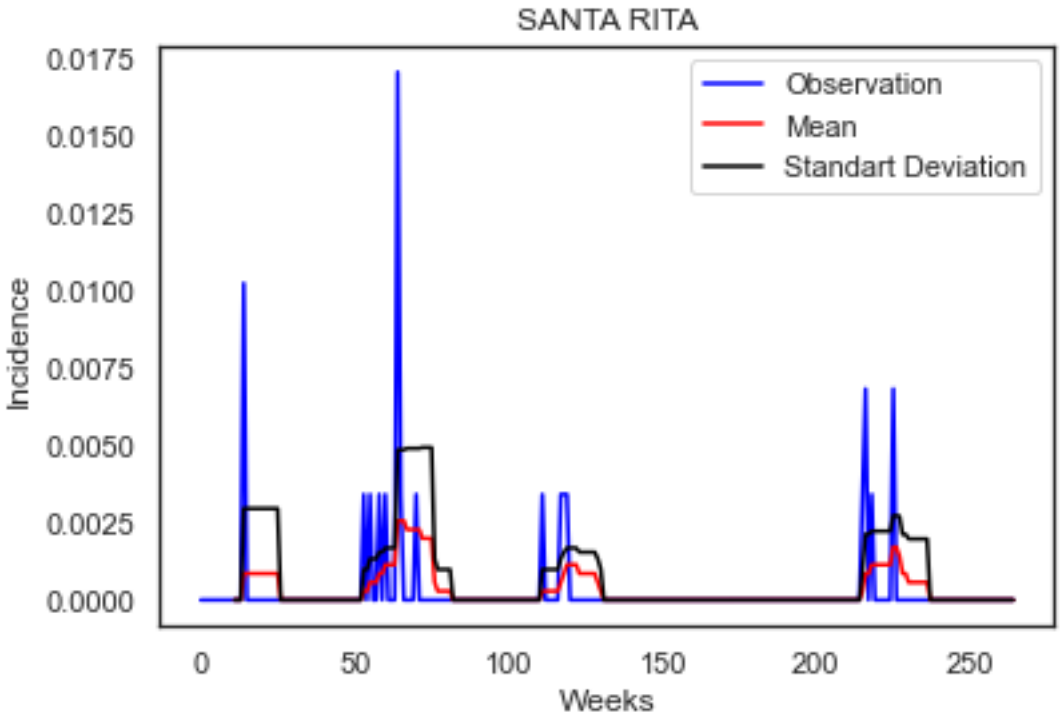


Figure B.15: ADF test for Santa Rita city.

Appendix C

Extended clustering results

This appendix presents the details of the clusters formed with the hierarchical clustering technique and that were used to train the *Cluster* model.

C.1 Clusters elements

The detail of the elements of each cluster can be seen in Table C.1.

Table C.1: Detailed elements of the clusters formed

Cluster	Elements	N° of elements
1	1RO DE MARZO, 25 DE DICIEMBRE, ANTEQUERA, AYOLAS, CABALLERO ALVAREZ, CAPIIBARY, CAPITAN MEZA, CARAYAO, CARLOS ANTONIO LOPEZ, CECILIO BAEZ, EUGENIO A GARAY, ISLA PUCU, ITACURUBI DEL ROSARIO, ITURBE, LIMOY PUEBLO, LORETO, MACIEL, NUEVA GERMANIA, SAN CARLOS, SAN COSME Y DAMIAN, SANTA ROSA DEL AGUARAY, SANTA ROSA DEL MBUTUY, VALENZUELA, VILLA DEL ROSARIO, YATAITY DEL NORTE, YGUAZU, YHU	27

Table C.1: Detailed elements of the clusters formed

Cluster	**Elements**	**N° of elements**
2	3 DE FEBRERO, ABAI, ACAHAY, ALTOS, AREGUA, ARROYOS Y ESTEROS, ATYRA, BAHIA NEGRA, BELLA VISTA, BENJAMIN ACEVAL, CAACUPE, CAAGUAZU, CAAZAPA, CAMBYRETA, CAPIATA, CAPITAN BADO, CAPITAN MIRANDA, CARAGUATAY, CHORE, COLONIA INDEPENDENCIA, DESMOCHADOS, DR BOTRELL, EMBOSCADA, ENCARNACION, EUSEBIO AYALA, FASSARDI, FERNANDO DE LA MORA, GENERAL ARTIGAS, GENERAL DELGADO, GENERAL DIAZ, GUARAMBARE, HERNANDARIAS, HUMAITA, ITA, ITACURUBI DE LA CORDILLERA, ITAPE, ITAPUA POTY, ITAUGUA, J A SALDIVAR, JESUS ,JUAN E. OLEARY, JUAN LEON MALLORQUIN, KATUETE, LA PALOMA, LA PASTORA, LAMBARE, LAURELES, LOMA GRANDE, LOMA PLATA, LUQUE, MARIANO ROQUE ALONSO, MCAL. FRANCISCO SOLANO LOPEZ, MINGA GUAZU, NATALICIO TALAVERA, NUEVA ALBORADA, NUEVA COLOMBIA, NUEVA ESPERANZA, NUEVA LONDRES, PEDRO JUAN CABALLERO, PIRAYU, PIRIBEBUY, QUIINDY, REPATRIACION, SALTO DEL GUAIRA, SAN ANTONIO, SAN BERNARDINO, SAN ESTANISLAO, SAN IGNACIO, SAN JOSE OBRERO, SAN LAZARO, SAN LORENZO, SAN MIGUEL, SAN PEDRO DEL PARANA, SANTA ELENA, SANTA MARIA, SIMON BOLIVAR, TACUARAS, TACUATI, TAVAPY, TEMBIAPORA, TOBATI, TRINIDAD, VILLA ELISA, VILLA HAYES, VILLA OLIVA, VILLETA, YEGROS, YPACARAI, YPANE, ZANJA PYTA	90

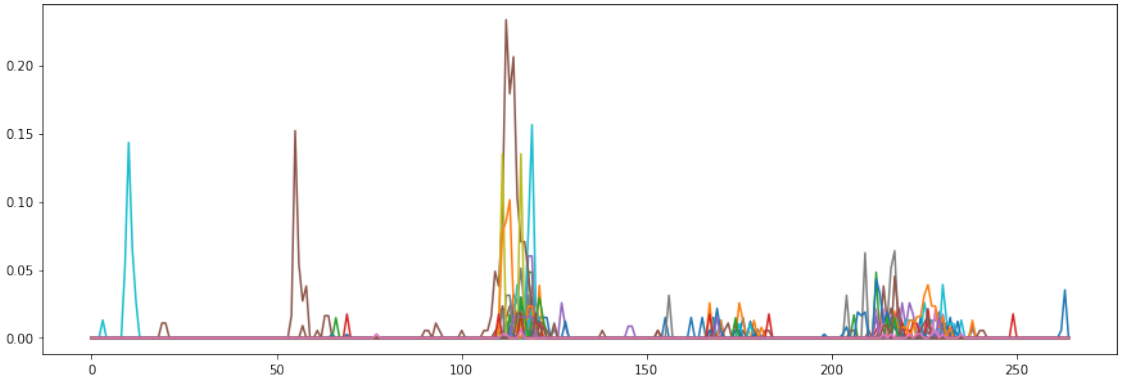
Table C.1: Detailed elements of the clusters formed

Cluster	Elements	N° of elements
3	ALBERDI, ALTO VERA, ASUNCION, AZOTEY, BELEN, BORJA, BUENA VISTA, CARAPEGUA, CARMEN DEL PARANA, CERRITO, CORONEL BOGADO, CORONEL OVIEDO, CORPUS CHRISTI, CURUGUATY, DR. JUAN MANUEL FRUTOS, ESCOBAR, FILADELFIA, FUERTE OLIMPO, GENERAL AQUINO, GENERAL BERNARDINO CABALLERO, GENERAL BRUGUEZ, GENERAL MORINIGO, GUAZUCUA, HOHENAU, ISLA UMBU, ITAKYRY, JUAN DE MENA, LEANDRO OVIEDO, LIMA, LIMPIO, MBOCAYATY DEL YHAGUY, MCAL. ESTIGARRIBIA, NANAWA, NATALIO, NUEVA ITALIA, OBLIGADO, PARAGUARI, PILAR, PIRAPO, PUERTO FALCON, RAUL ARSENIO OVIEDO, ROQUE GONZALEZ DE SANTA CRUZ, SAN JOAQUIN, SAN JOSE DE LOS ARROYOS, SAN JUAN DEL PARANA, SAN PATRICIO, SAN PEDRO DEL YCUAMANDIYU, SAN RAFAEL DEL PARANA, SAN SALVADOR, SANTA RITA, SANTIAGO, SAPUCAI, TAVAI, TEBICUARY, TOMAS ROMERO PEREIRA, VAQUERIA, VILLALBIN, VILLARRICA, YAGUARON, YATAITY, YATYTAY, YBY YAU, YBYTYMI, YPEJHU, YUTY	65
4	CARMELO PERALTA, CHACO, LA VICTORIA, POZO COLORADO, SAN ROQUE GONZALEZ DE SANTACRUZ, YBYRAROVANA	6
5	COLONIA FRAM, CONCEPCION, CORONEL MARTINEZ, EDELIRA, FELIX PEREZ CARDOZO, GUAYAIBI, HORQUETA, ITANARA, JUAN EULOGIO ESTIGARRIBIA, MAURICIO JOSE TROCHE, MBOCAYATY, MOISES BERTONI, PASO YOBAI, SAN JUAN BAUTISTA, SAN JUAN NEPOMUCENO, SAN PEDRO, SANTA ROSA, SANTA ROSA DEL MONDAY, TEBICUARYMI, UNION, YGATIMI	21
6	GENERAL RESQUIN, JOSE DOMINGO OCAMPOS, MBARACAYU, MINGA PORA, NARANJAL, PUERTO PINASCO, R I 3 CORRALES, SAN ALBERTO	8

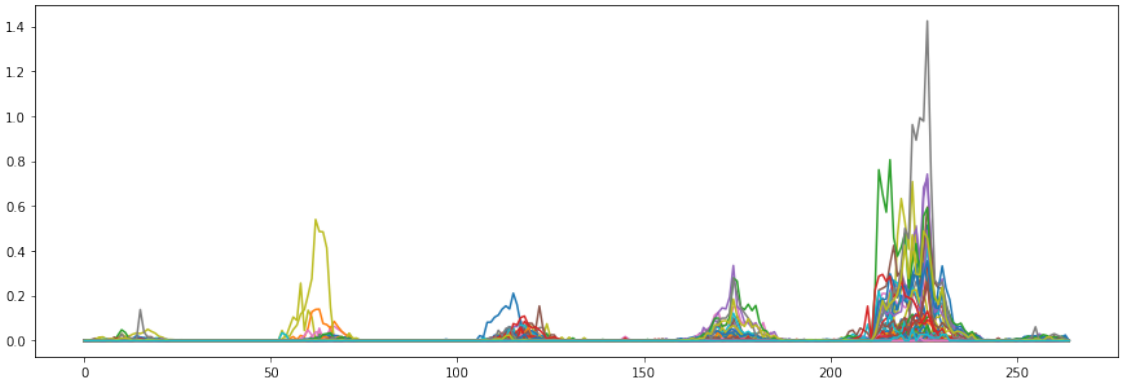
C.2 Visualization of clusters

Table C.2 show the graphic representation of the time series that belong to the same cluster.

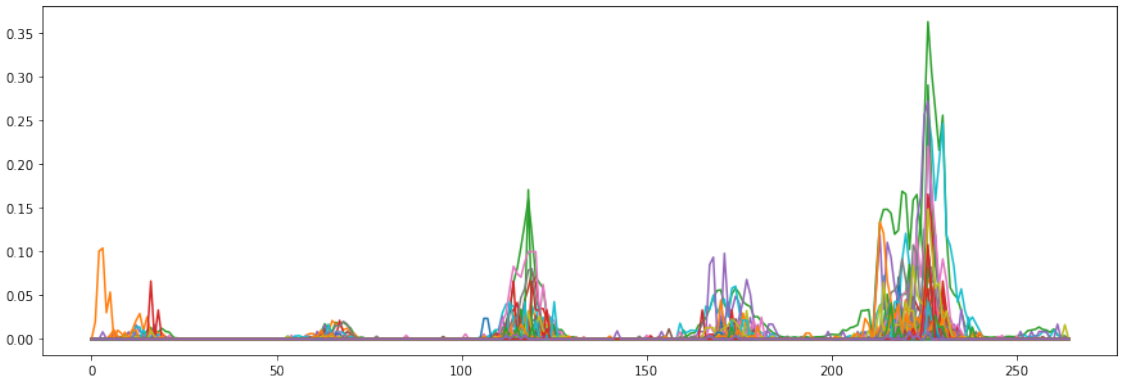
Table C.2: Graphic representation of the time series of each cluster



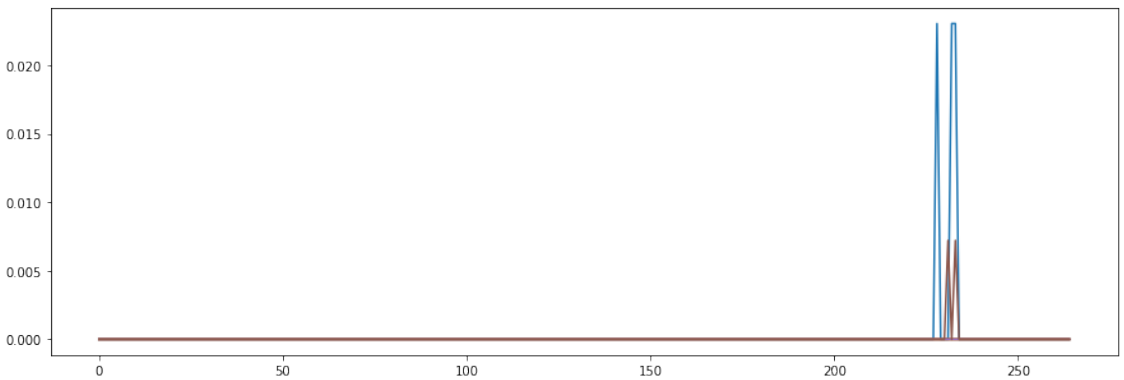
Cluster 1



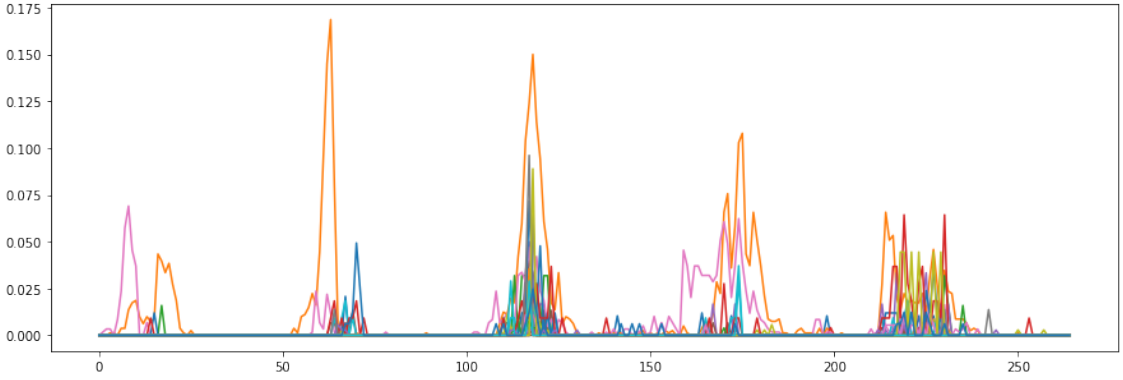
Cluster 2



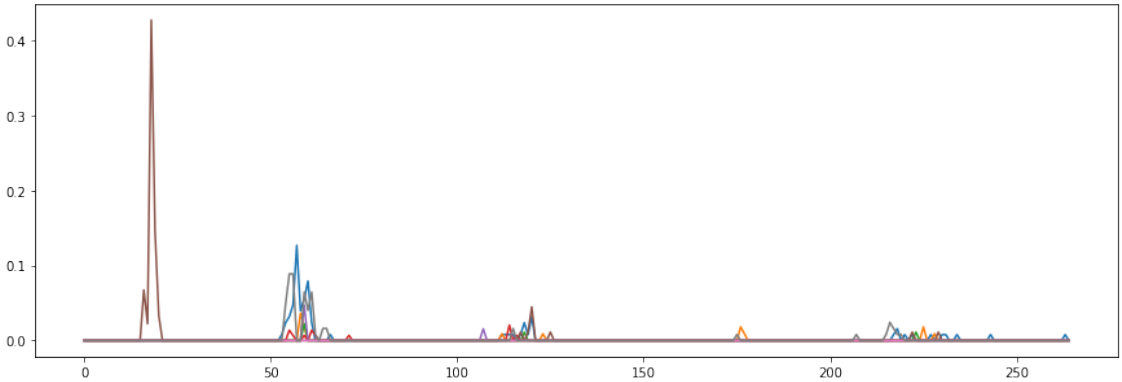
Cluster 3



Cluster 4



Cluster 5



Cluster 6

Appendix D

Details of the improvement comparison between models

This appendix presents the details of the comparisons between the basic model (*Single*) and its comparison with tested models in terms of percentage improvement in RMSE. Table D.1 shows the percentage of improvement of the *Single* model in relation to the *Cluster* model, and Table D.2 shows the percentage of improvement of the *Single* model relative to the *Bayesian* model.

Table D.1: Analysis of the observed improvements of the *Cluster* model

Group	**City**	*Single*	*Cluster*	**Improvement (%)**
Group 1	San Lorenzo	0.1360	0.0510	62.5000
	Capiatá	0.1334	0.0940	29.5352
	Caaguazú	0.0266	0.0135	49.2481
	Areguá	0.1201	0.1003	16.4863
	Salto del Guairá	0.0259	0.0186	28.1853
Group 2	Choré	0.0075	0.0063	16.0000
	Juan León Mallorquín	0.0096	0.0096	0.0000
	Santa Rosa del Aguaray	0.0060	0.0037	38.3333
	Quiindy	0.0095	0.0099	0.0000
	Eusebio Ayala	0.0101	0.0095	5.9406
Group 3	Encarnación	0.0028	0.0026	7.1429
	San Pedro del Ycuamandijú	0.0033	0.0027	18.1818
	Capitán Miranda	0.0031	0.0031	0.0000
	Yhú	0.0017	0.0016	5.8824
	Santa Rita	0.0021	0.0017	19.0476
Average improvement				19.48 ± 18.80

Table D.2: Analysis of the observed improvements of the *Bayesian* model

Group	**City**	*Single*	*Bayesian*	**Improvement** (%)
Group 1	San Lorenzo	0.1360	0.0580	57.3529
	Capiatá	0.1334	0.1046	21.5892
	Caaguazú	0.0266	0.0152	42.8571
	Areguá	0.1201	0.1112	7.4105
	Salto del Guairá	0.0259	0.0202	22.0077
Group 2	Choré	0.0075	0.0061	18.6667
	Juan León Mallorquín	0.0096	0.0094	2.0833
	Santa Rosa del Aguaray	0.0060	0.0058	3.3333
	Quiindy	0.0095	0.0094	1.0526
	Eusebio Ayala	0.0101	0.0090	10.8911
Group 3	Encarnación	0.0028	0.0021	25.0000
	San Pedro del Ycuamandijú	0.0033	0.0032	3.0303
	Capitán Miranda	0.0031	0.0030	3.2258
	Yhú	0.0017	0.0016	5.8824
	Santa Rita	0.0021	0.0015	28.5714
Average improvement				16.86 ± 16.57