



## SELECCIONES MATEMÁTICAS






Universidad Nacional de Trujillo

ISSN: 2411-1783 (Online)

2021; Vol. 8(1): 12-26.



### Forecasting SARS-CoV-2 in the peruvian regions: a deep learning approach using temporal convolutional neural networks

Luis Aguilar I. , Miguel Ibáñez-Reluz , Juan C. Z. Aguilar , Elí W. Zavaleta-Aguilar  and L. Antonio Aguilar 

Received, Feb. 22, 2020

Accepted, Apr. 30, 2020



#### How to cite this article:

Aguilar L. et al. *Forecasting SARS-CoV-2 in the peruvian regions: a deep learning approach using temporal convolutional neural networks*. *Selecciones Matemáticas*. 2021;8(1):12–26. <http://dx.doi.org/10.17268/sele.mat.2021.01.02>

#### Abstract

*The SARS-CoV-2 pandemic had taken the world by surprise since its discovery on December 2019, causing major losses worldwide. In this work, a deep learning model was developed to predict and forecast the daily SARS-CoV-2 cases on the Peruvian regions. The data used belongs to the open covid-19 data set, sourced by the Health Ministry of Peru (MINSA). The data set includes the periods from March 03, 2020 to March 16, 2021. A holdout approach was used, creating a training and validation data splits. Using the validation set, a temporal convolution neural network (TCN) composed by five layers was developed. The model was design to predict a mean tendency alongside with a prediction interval. To find the best hyper parameter configuration, a Bayesian approach was applied over the validation set. The TCN model was trained using the optimal configuration. Once trained, the model was able to predict the different SARS-CoV-2 trends present in the regions. Next, a forecast was performed beyond the available data, using a window of 15 days ahead (March 17 to March 31, 2021) for each region. Forecast results suggested a continued trend for all the regions, except Lima. The model performance was evaluated using the MAE, MAD, MSLE and RMSLE metrics on the test period, showing training to validation metrics improvements of 14.534, 3.123, 0.042, 0.047 respectively.*

**Keywords** . Deep Learning, forecasting, SARS-CoV-2, temporal convolutional neural networks, time series data.

**1. Introduction.** SARS-CoV-2 is a novel virus, which was first reported in December 2019, in Wuhan city, Hubei province, China. It belongs to a family of seven known coronavirus strains, which cause severe acute respiratory syndrome [1]. The initial symptoms are similar to the flu, with presence of fever during three to four days. Unlike the flu, it can cause major damages to different organs including the lungs. Furthermore, the existence of comorbidities such as chronic pulmonary disease and diabetes mellitus could result in severe complications [2]. SARS-CoV-2 also presents a high infection rate [3]. SARS-CoV-2 can be transmitted between people through direct, indirect or close contact. The primary transmission is contained in the secretions or secretion droplets of an infected patient. These are released into the air

\*Department of Mathematics, National University of Piura, Urb. Miraflores s/n, Castilla Apartado Postal 295, Piura, Perú. (laguilari@unp.edu.pe).

†Medicine Faculty, Cesar Vallejo University, Av. Victor Larco 1770, Trujillo, Perú. (mibanezr@ucvvirtual.edu.pe).

‡Department of Mathematics and Statistics, Universidade Federal de São João del-Rei C.P. 110, CEP 36301-160, São João del-Rei, MG, Brazil. (jaguilar@ufsj.edu.br).

§São Paulo State University (Unesp), Campus of Itapeva Rua Geraldo Alckmin 519, 18409-010 Itapeva, SP, Brazil. (eli.zavaleta@unesp.br).

¶Artificial Intelligent Research, KapAITech Research Group, Condominio Sol de Chan-Chan, Trujillo, Perú. (antonio@kapaitech.com).

when the patient coughs, sneezes, speaks, sings or make any activity which liberate those particles. People wearing no protection which are in close contact with those particles can be infected with SARS-CoV-2, since the particles can enter the body through the mouth, nose or eyes. Symptoms can appear between 1 to 14 days after the first contact. At this point it is recommended to isolate the patient, since it can spread the infection without any visible symptom (asymptomatic). SARS-CoV-2 can be classified in four clinical phases: asymptomatic, mild, moderate and severe. Acute respiratory infections (ARI's) are present in all cases (except the asymptomatic). An ARI is a clinical state which can present the following symptoms: cough, throat pain, congestion (nasal sinuses or lungs), fever, fatigue, body aches; these symptoms can affect the upper or lower respiratory system.

An asymptomatic case presents no visible symptoms during the infection. A mild case includes any patient which develop any symptom presented in ARI's. For a moderate case, it is observed a respiratory insufficiency, which is manifested with a respiratory rate greater than 22. Patients also experiment disorientation, confusion, hypotension. In this state there is a high risk of developing pneumonia, also the lymphocyte count is less than 1000 cell/pL. In a severe case the respiratory insufficiency condition is critical, the PaFi index is less than 300, meanwhile the PaCO<sub>2</sub> and PaO<sub>2</sub> levels are below 32mm/Hg and 60mmHg respectively. The systolic blood pressure is below 100mm/Hg, having a risk of septic shock (PAM < 65 mmHg). An increase in serum lactate is present (> 2 mOsm/L). Since the difficulty in breathing is critical, artificial respirators are used [4]. Patients in a severe case, need to be treated with extreme care.

In Peru, the clinic behavior of SARS-CoV-2 had been very similar with the observed in other countries. Patients with comorbidities and over 60 years old are considered to be more in danger. The first Peruvian SARS-CoV-2 case was reported on March 03, 2020 in Lima, two months after the first known case on Wuhan, China. In order to slow down the infection, on March 16, 2020, the government declared a national sanitary emergency state. The Peruvian government adopted a set of policies, such as: quarantine, close of borders, travel restrictions (local and international), social distancing and closing schools and universities. These policies were later accompanied with economic incentives, which were present in the form of bonuses and the early withdrawal of funds from private pension systems [5].

In August 13, 2020, the healthcare capacity was heavily affected. This was evidenced nationwide in the full use of intensive care units (ICU), artificial respirators and oxygen tanks. In the following four months a slow decrease in cases and fatalities were observed. However, at the beginning of January 2021, a sudden increase in cases was present nationwide, been Lima among the most affected regions. This sudden increase could be the result of the festivities and other social activities which were present at the end of December 2020 [6].

In South America, until March 21, 2021, Peru was the country with more cumulative deceases per million. It also ranked fifth in infected cases by million behind other countries such as: Brazil, Argentina, Chile and Colombia. Regarding vaccination, until March 20, 2021, 478 925 vaccines were applied with Sinopharm and 217 000 with Pfizer, all those applied to the elderly (> 85 years) and National Police and Armed Forces personnel [7]. Perú is far from ideal on vaccinations, since in South America it is in the 7th position and representing, proportionally to the population, approximately 4 % of those vaccinated in Chile, which is leading this measure followed by Uruguay [8]. Factors such as the new Brazilian variant [9], labor informality alongside with a fragile healthcare system may be contributing to worsen the situation.

Estimate SARS-CoV-2 cases trend is a crucial task, since it can help in build and adapt better policies faster. However, this task is a challenging one, since SARS-CoV-2 cases vary in nature. As such different approaches have been proposed. For example, parametric models such as SIR, SEIR, SIRD, and its variants were used on the pandemic early stage for better understand the spreading dynamics of SARS-CoV-2. Such models were ideal, since they were able to operate with relatively small data sets. A SIR model is composed by a system of differential equations which models the relationship between three epidemiological parameters: susceptible (S), infectious (I) and removed (R) cases [10]. In [11] a SIR model was used to analyze the evolution of cases in six worldwide representative countries. The results shown the tendencies in cases per each country alongside with their estimated peaks. After analyzing the results, it was concluded that the use of adequate restrictions alongside with strong policies will have a positive effect in decreasing the infection rates.

A SEIR model, on the other hand, includes one additional parameter: exposed (E), that accounts the people exposition during a pandemic event. In [12] a SEIR model was applied on the Hubei province in China. In order to find a set of suitable parameters, the PSO algorithm was used. This results in a good accuracy for the model. Also, if seasonality and stochastic infections were present, they could generate chaos in the system. The study also shown that the system behavior can change in the presence of a different set of parameters.

Unlike the SIR and SEIR models, which treat the recovered and deceased cases as removed (R), a SIRD model treats them independently. This allows a SIRD model to include the susceptible (S), infectious (I), recovered (R) and deceased (D) cases. [13] implement a SIRD model using data for the Hubei province in

China. The model was applied to two scenarios. The first used the provided data, whereas the second added an estimate of  $x_{20}$  to the number of infected (I) and  $x_{40}$  to the number of recovered (R) cases. The second scenario was created to consider the possibility of cases which could not be reported. Analysis of the first scenario, shown that quarantine policies have an impact in reducing the cumulative cases. Whereas in the second one, the case fatality ratio was around 0.15% of the total population. All estimations were performed using a three week ahead window.

As the pandemic progressed, more data became available. In particular, the efforts provided by the Johns Hopkins University with a global data set repository [14], which is continuously updated, giving support for developing models more quickly. Particularly the application of deep learning models was possible, since they require substantial amount of data. Also, using the information of previous models like SIR and its variants, forecasting deep learning models focused on one or more of the epidemiological parameters: susceptible (S), infectious (I), recovered (R) and deceased (D) cases.

There are different approaches which can be used to forecast SARS-CoV-2 cases. SARS-CoV-2 cases are a phenomenon which is affected by several factors including: social behaviors, quarantine policies, economic status, etc. Among these factors, a time dependency is present. This dependency is reflected in the changes in infections over time. This means, this phenomenon is continuously changing. In this context, a time series approach can be used.

Once the data is treated as a time series, a wide variety of recurrent models can be applied. Recurrent neural networks (RNN) such as long short term memory (LSTM) [15] and its variants are able to include a memory mechanism, which can learn time dependency relationships. Such models and its variants have shown promising results. In [16] a LSTM model was used to forecast SARS-CoV-2 cases in Canada. A data set provided by the Canadian health authority and the COVID-19 data repository [14] were used. An 80%-20% split strategy was implemented for train and test respectively. The RMSE metric was used to evaluate the model performance. Once trained, the LSTM model was able to estimate a peak in cases within 2 weeks.

Although recurrent models are preferred for time series data, other models could also be used, outperforming recurrent ones [17]. For forecasting the daily, confirmed and recovered SARS-CoV-2 cases in Italy, Spain, France, China, USA and Australia; the variational auto encoder (VAE) model obtained better results than four recurrent models (RNN, GRU, LSTM and Bi-LSTM). This was reflected on the VAE scores reported by the RMSE, MAE, MAPE, EV and RMSLE metrics. VAE's are networks which belong to the category of generative models. These models are able to learn an approximation to the data distribution. Once trained, they are able to draw new data points from the learned approximation. Deep VAE's are usually composed by convolutional layers.

Models such as convolutional neural networks (CNN), which are usually applied to image data, have also been used to forecast SARS-CoV-2 cases. An example of such application is presented in [18], where a CNN model was applied to forecasting daily, cumulative, recoveries and deceased SARS-CoV-2 cases. Also, the CNN model was able to forecast the hospitalizations cases (with and without artificial ventilation). The study was executed on France at regional and national levels. The data was collected from diverse sources. The results shown a good performance in forecasting, which was evaluated using the MAE, RMSE, and R2 metrics. Besides the CNN model, the study in [18] also used a variation called temporal convolutional neural networks (TCN). A TCN model is composed by dilated casual convolutional layers. These layers implement different mechanism which allow a TCN to process time series data more naturally. The TCN model was trained with a quantile loss, allowing the model to estimate prediction intervals with a multi output setting. The TCN was evaluated using the same metrics as the CNN. However, the TCN obtained better results than the CNN model.

Another important factor for forecasting is the desired horizon (short or long). [20] presented a multi-head attention, LSTM and CNN models which were developed to predict the SARS-CoV-2 confirmed cases. This study was applied to a short and long horizon using two data sets. These data sets contained records of confirmed cases in several countries including Peru and Brazil. Results on the CNN model in the short horizon were superior than the LSTM model according the SMAPE, MAPE and RMSE metrics. For long horizon, the CNN model was superior in Peru and Brazil.

Lastly, comparing CNN's models against diverse architectures such as GRU, LSTM and MLP, CNN's ones showed better results for forecasting cumulative SARS-CoV-2 cases in seven chinese cities [19]. This study was conducted at an early stage in the pandemic. The forecasting was performed with one day ahead using the previous five days of total and new cases as information. This information involve: confirmed, recovered and deceased cases. The models were evaluated using the MAE and RMSE metrics. The CNN model reports the best scores in forecasting for both metrics in the seven cities.

As seen, each study has focused on one or more components of SARS-CoV-2 cases (daily, regional, cumulative, etc.). As such, this work aims to build a single model, which is capable of predict and forecast the daily cases for the 25 different Peruvian regions using a window of 15 days ahead. To address this task

a TCN model, composed by casual dilations and residual connections is presented. A Bayesian approach was used to estimate the best hyper parameters. The structure of the following sections are as follows: Section 2 describes the theory used to build the TCN model. Section 3 shown the proposed methodology. Section 4 shows the proposed TCN model. Section 5 presents the prediction and forecasting results, which are evaluated using the MAE, MAD, RMSE and RMSLE metrics. Finally, in Section 6 are presented the study conclusions.

**2. Temporal Convolutional Neural Networks (TCN).** TCN are models which apply the convolution operation throughout time. TCN are composed by convolutions (casual and dilated) and residual connections. TCN’s are able to process sequences, from which a set of features are extracted with the use of a kernel.

In a casual convolution, the kernel is applied (convolve) maintaining the natural order of the input sequence. This process is illustrated in Figure 2.1a, where instead of applying the convolution directly over the sequence, the kernel is moved from left to right. This process forces the model to only relay on past data to make its predictions, this also prevents any data leaking from the future. A casual convolution is generally composed by a 1D convolution. However, a casual convolution is unable to maintain a long horizon history, which is essential in order to capture past patterns. Therefore, another variation called dilated convolutions are applied. Dilated convolution relay on dilations to increase the receptive field [21]. This effect can be observed in Figure 2.1b, where an increasing dilation allows next layers to expand the input information.

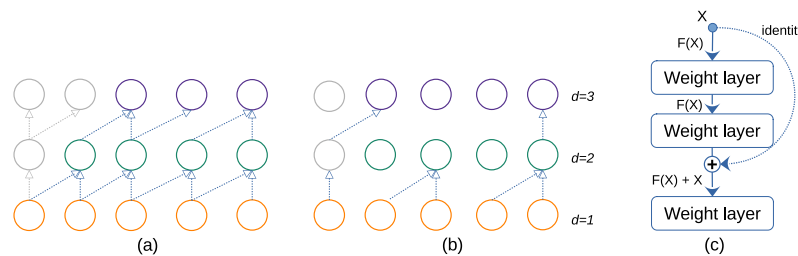


Figure 2.1: Three mechanisms in a TCN: (a) casual convolutions, (b) dilated convolutions and (c) skip residual connections.

Formally for an input sequence  $s = \{x_0, \dots, x_T\}$ , where  $T$  denotes the sequence length; a casual dilated convolution  $\mathcal{F}(s)$ , is defined as:

$$(2.1) \quad \mathcal{F}(s) = (x *_{d} f)(s) = \sum_{i=1}^k f(i) * x_{s-d*i},$$

where the dilation factor  $d$  express the amount of variation over the receptive field. Next, a set of kernels  $f$  of size  $k$  are convolved over the sequence elements in  $s$ , such that  $x_{s-d*i}$  constrain the sequence to only considered past information.

The last component of a TCN is a skip residual connection. This mechanism was introduced by [22] in the Residual Network architecture. Skip residual connections create a shortcut in the information flow (gradient). The residual is skipped over a certain number of layers. Then it is combined in the main information flow. This operation allows to create deeper models and deal with the gradient vanish problem. Figure 2.1c., shows a vanilla residual connection scheme.

**3. Methods.**

**3.1. Feature Engineering.** In its raw format, the open covid–19 data set [23] contained 9 features and 137336 samples which were registered from March 03, 2020 to March 16, 2021 (accessed on 20.03.2021). The daily SARS-CoV-2 cases were represented as observations at the lowest administrative division (district), without a column name. Therefore, a new feature named cases was created, allowing to represent the daily SARS-CoV-2 cases per each district. Also, missing values were imputed accordingly. Next, a total of 5 features (Table 3.1) were selected.

From the 5 features described in Table 3.1 a set of 9 new ones were created. First, using the date feature a total of 6 features were designed to capture time dependencies which are present in the dynamics of SARS-CoV-2 cases (Table 3.2).

Features	Description
Date	Day information, in the format: year–month–day.
Region	Region name.
Province	Province name.
District	District name.
Cases	Total cases per day.

Table 3.1: An initial subset of 5 selected features from the raw open covid–19 data set [23].

Features	Description
Month	Actual month, values from 1 to 12.
Day of year	Ongoing day in the year, values from 1 to 365.
Week	Week day, which values correspond to 0 for Monday and 6 to Sunday.
Weekday	Week of the year, values from 1 to 52.
Quarter	Current quarter.
Month day	Day of the current month.

Table 3.2: Engineering time dependent features to represent the dynamics of daily SARS-CoV-2 cases.

Features	Description
$W_5$	A five-day window of previous cases.
$I_p$	Number of infected provinces in a region.
$I_d$	Number of infected districts in a region.

Table 3.3: Engineering five day feature alongside provinces and districts statistics.

Next, from the cases, province and district features a total of 3 new features:  $W_5$ ,  $I_p$  and  $I_d$  were created respectively. These features represented a window of 5 past days containing information about SARS-CoV-2 cases alongside with statistics about the infection in provinces and districts (Table 3.3).

Finally, since the aim of this study was focused on regional level, data from Table 3.1, alongside with the features from Table 3.2 and Table 3.3 were represented per each region (included the constitutional province of Callao). This new data set was used by the TCN model for validation and training, which contained a total of 10 features (Table 3.4).

Task	Data source	Total features
Forecasting and Modeling	Table 3.1 (region name), Table 3.2 and Table 3.3	10

Table 3.4: Final data set built from previous features. This data set was used by the TCN model for validation and training.

**3.2. Data preparation.** The combine set of features described in Table 3.4 (forecasting and modeling task) were used to train the TCN model. First, the daily cases in each region were arranged into a fix sequence  $W_5$  of length 5 (previous days); where the six day was used as target (Figure 3.1a). To obtain a fix sequence, all samples were padding with zeros to match the length of the longest sequence (first registered case) in Lima on March 06, 2020. The length of the sequence was determined empirically, by a trade-off



between the available data and the window length. Missing values in  $W_5$  were filled with its corresponding monthly mean per each region. Finally,  $W_5$  was sorted by date, allowing to have a  $k$  unique instance of each region at each time step, where  $k$  represent the total regions (Figure 3.1b).

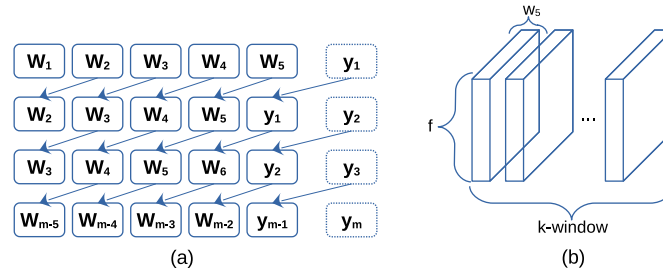


Figure 3.1: A sequence window  $W_5$  containing previous five days are represented in (a). This arrangement was combined with nine features, resulting in a 3D input volume (b). This volume was used as training data by the model.

Since the initial feature size  $W_5$  was limited, it was combined with the remaining 9 features from Table 3.4 into a 3D input tensor as shown in Figure 3.1b. This allowed to set a horizon of five days in the current bath size using:  $k * 5$ . Also, all features in the tensor were scaled individually, such that the maximal absolute value of each one was set to 1.0. After completing this process, the data was divided into two data sets: validation and training. Then, each set was further divided into two partitions: train and test. The train partition was used for model learning, whereas the test one for model evaluation. This process, conditioned the model to not relay on leaked information from any partition.

Partition	Features	Samples	Data range	Days
Train	10	8650	Mar. 06, 2020 – Feb. 14, 2021	346
Test	10	375	Feb. 15, 2021 – Mar. 01, 2021	15

Table 3.5: Validation data set partition used to build the TCN. Values presented in the training data set were not leaked to avoid any overfitting.

Partition	Features	Samples	Data range	Days
Train	10	9025	Mar. 06, 2020 – Mar. 01, 2021	361
Test	10	375	Mar. 02, 2021 – Mar. 16, 2021	15

Table 3.6: Training data set partition used to train the TCN model.

The validation data set was used to build the model. This included the design of the optimal network architecture. Also, the validation set served as an initial search point to find the best hyper-parameter configuration. This was achieved through the use of a Bayesian optimization approach. As such, the model learn from the train partition, meanwhile it was evaluated against the test one. Table 3.5 described the partitions in detail.

On the other hand, the training data set was used to create the final model. Thus, the model was fitted on the train partition using the best hyper-parameter configuration found on the validation data set. Then, the model was evaluated using the test partition. Table 3.6 shows the partition details used.

**4. Proposed Model.** A TCN model was developed to predict and forecast the daily SARS-CoV-2 cases in the 25 Peruvian regions. This model, was build using the validation data set described in Table 3.5. During this process a set of optimal hyper-parameters were found. Then, the final model was build using the training data set described in Table 3.6. The model architecture was designed entirely with dilated casual convolutions (DC-Conv) layers, which applied convolutions according to Eq 2.1. These layers showed

better validation set performance in contrast with other architectures, such as LSMT or pure CNN's. In order to normalize the variance between layers, a batch normalization layer  $\mathcal{BN}$  [24] was applied using the following equation:

$$(4.1) \quad \mathcal{BN}(x) = \frac{x - E[x]}{\sqrt{Var[x] + \epsilon}} * \gamma + \beta$$

Where  $x$  represents the outputs from a DC-Conv layer, meanwhile  $\gamma$  and  $\beta$  are learnable parameters. The value  $\epsilon$  was set to  $1 \times 10^{-5}$  to avoid possible computation overflows. Also, during training, the batch layer keeps running estimates of its mean and variance, which are then used for normalization during inference. Next, as an activation function between DC-Conv layers, the Rectified Linear Unit (ReLU) [25] was selected. This function allowed to normalize the outputs according with:

$$(4.2) \quad ReLU(x) = \max(0, x)$$

Where  $x$  represent the output layer. Finally, a dropout layer [26], which acted as a regularization during training was implemented. This layer allowed to randomly turn off some elements in the in the inputs according with a probability  $p$ , which was sampled from a Bernoulli distribution. Then, the outputs are scaled as follows:

$$(4.3) \quad Dropout(x) = \frac{1}{1 - p}$$

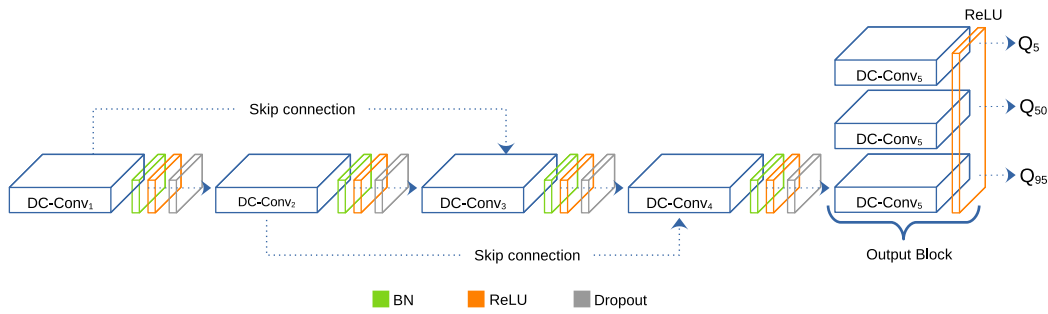


Figure 4.1: Proposed forecasting Temporal Convolutional Neural Network (TCN) model. Each blue block denotes a casual dilated convolution. A sequence of batch normalization (green), relu activation (orange) and dropout (gray) is applied after each block. Finally, each output layer learns a prediction interval.

Due to the randomness present in Eq 4.3, this layer was deactivated when performing predictions. Using these components a TCN model was designed. This model was composed by a set of five DC-Conv layers (Eq 2.1), which are followed by a batch normalization layer (Eq 4.1). As an activation function ReLU (Eq 4.2) was used. In order to regularize the model, a dropout layer (Eq 4.3) was applied at the end of each DC-Conv layer, except the output layer. The output block was composed by three independent DC-Conv layers. In order to avoid negative predictions, a ReLU activation was applied in the output. Also a set of two residual connections were implemented according with the scheme in Figure 2.1c. Figure 4.1 show the complete architecture. Each configuration of the components of the DC-Conv layers are detailed in Table 4.1.

Casual convolutions were achieved when removing the extra padding present in the output of the blocks. The amount of padding was determined as follows:  $(k - 1) * d$ , where  $k$  and  $d$ , represent the kernel and dilatation size. As observed in Figure 4.1, at the end of each output block a single prediction was computed. Then, using a quantile loss, each block was trained to learn a prediction interval. Concretely, each output block was specialized in learn the 5%, 50% and 95% quantile, which correspond to Q1, Q2, and Q3 respectively. During training, each quantile  $q$  was computed using Eq. (4.4):

$$(4.4) \quad \mathcal{L}(\xi_i|q) = \begin{cases} q\xi_i, & \xi_i \geq 0 \\ (q - 1)\xi_i, & \xi_i < 0 \end{cases}$$

Block	Input ch.	Output ch.	Kernel size	Padding	Dilation	Dropout ratio
DC-conv <sub>1</sub>	5	32	3x3	8	4	0.407
DC-conv <sub>2</sub>	32	32	3x3	16	8	0.581
DC-conv <sub>3</sub>	32	32	3x3	16	8	0.478
DC-conv <sub>4</sub>	32	32	3x3	20	10	0.354
DC-conv <sub>5</sub>	32	1	2x2	5	14	–

Table 4.1: Detailed information about the internal configuration of each DC-Conv layers.

Where  $q \in [0, 1]$  and  $\xi_i$  is defined as the difference between the real values  $y$  and predictions  $\hat{y}$  for a mini batch  $b$ , such that:  $\xi_i = y_i - \hat{y}_i$ . Finally, the total quantile error is averaged during training over the total samples  $m$  using:

$$(4.5) \quad \mathcal{L}(y, \hat{y}|q) = \frac{1}{m} \sum_{i=1}^m \mathcal{L}(\xi_i|q)$$

The loss defined in Eq 4.5 was optimized using the mini batch stochastic gradient descent with momentum (SGD) [27] optimizer with the hyper-parameter configuration found in the validation set. The hyper-parameters were obtained through Bayesian optimization using the BoTorch [28] library. Table 4.2 describe the optimal configuration found, which was used to built the final model on the training data set. The cycle learning [29] strategy was used during training, since it helped the model to better converge. The hyper-parameters were initialized using Xavier [30] with a normal distribution.

Hyper-parameter	Value
Base learning rate	0.0057
Maximum learning rate	0.0067
Momentum	0.832
Weight decay	5.43E-06
Optimizer	SGD
Cycle learning step size	2000
Training epochs	5781
Batch size	125
Initializer	Xavier
Shuffle during training	Yes

Table 4.2: Optimal hyper-parameter configuration obtained with Bayesian optimization on the validation data set.

Because the disperse magnitude of the target variable  $y$ , it was normalized during training with the natural logarithmic  $\log(y + 1)$ . Then, at inference, the inverse operation  $e(\hat{y} + 1)$  was applied over the mean prediction  $\hat{y}$ . The TCN model was build to predict and forecast using a multi step strategy of one day ahead. This allow one to estimate beyond the available data feeding the continuous  $\hat{y}_{t+1}$  as  $W_5$  features, which were combined with the other features described in Table 3.4.

Once the model was trained, a set of metric were applied to evaluate the model. These metrics used the partitions described by Tables 3.5 and 3.6 respectively. The train partitions were used to determined the learning capabilities of the model, whereas the test partitions were used to measure the model generalization to new unseen data, which offer an estimated error for the forecasting as well.

Then, each region was evaluated independently using the test partition (Table 3.6) which include data points from March 02, 2021 to March 16, 2021. The metrics are defined in terms of the mean predictions  $\hat{y}_m$  and real data points  $y_m$ , such that  $m$  represents the total samples presented in each partition (train or test). These metrics are the mean absolute error (MAE), median absolute error (MAD), mean squared logarithmic error (MSLE) and root mean squared logarithmic (RMSLE). The following equations describe each one in more detail:



$$(4.8) \quad MSLE = \frac{1}{m} \sum_{i=1}^m [\log(\hat{y}_i + 1) - \log(y_i + 1)]^2$$

$$(4.9) \quad RMSLE = \sqrt{\frac{1}{m} \sum_{i=1}^m [\log(\hat{y}_i + 1) - \log(y_i + 1)]^2}$$

Also, to measure the prediction intervals robustness, an estimate of the percentage of real data points  $y$  which are captured by the lower ( $L$ ) and upper ( $U$ ) interval area is used, such that:  $L \leq y \leq U$

$$(4.10) \quad P(y, L, U) = \frac{1}{m} \sum_{i=1}^m u_i, \quad u(y) = \begin{cases} 0 & \text{if } L \geq y \geq U \\ 1 & \text{if } L \leq y \leq U \end{cases}$$

## 5. Results.

**5.1. Prediction.** Predictions were computed using the training data set described in Table 3.6. The train partition was used for model learning, meanwhile the test one for generalization on new unseen data. Date periods included March 06, 2021 to March 01, 2021 for the train partition and March 02, 2021 to March 16, 2021 for test. Results were reported for the 25 peruvian regions.

The TCN model computed a mean prediction  $\hat{y}$ , alongside with a prediction interval area. Blue points represent the actual daily cases reported by MINSA in each region. The orange line represent the mean prediction  $\hat{y}$  for the daily SARS-CoV-2 cases. Meanwhile, the light blue areas denote the prediction intervals. This means that, for a certain data point  $y$ , the model is able to estimate its mean tendency ( $\hat{y}$ ) alongside with a prediction interval area, giving the model more flexibility to address the problem of uncertainty presented in the dynamics of the daily SARS-CoV-2 cases. Figure 5.1 shown the mean predictions  $\hat{y}$  alongside the prediction intervals generated by the model for the train and test partitions.

As observed in Figure 5.1, the diverse daily SARS-CoV-2 tendencies were well captured by the model in all regions. The predictions  $\hat{y}$  allowed the model to draw a mean estimation of the daily SARS-CoV-2 tendency in each region. Whereas the prediction intervals captured the variation in tendencies. The model is able to categorized some data points as outliers. This can be observed in the regions of Huanuco, Madre de Dios, Cajamarca, Puno, Apurimac and Ucayali, which have anomalies with higher points. On the other hand, in Lima the prediction interval area indicates a bigger number of cases than the reported. This could suggest that the tendency in Lima is higher than the actual reported.

**5.2. Forecasting.** Forecasting was applied from March 17, 2021 to March 31, 2021, with a total window of 15 days. This period was not present in the original data set (accessed on 20.03.2021), which only included from March 06, 2020 to March 16, 2021. Forecasting tendencies were automatically adjusted by the TCN model per each region, despite the variations in SARS-CoV-2 cases. Daily SARS-CoV-2 tendencies were captured by the mean predictions  $\hat{y}$  while the interval prediction areas shown the possible variations. Observing the mean predictions  $\hat{y}$ , all the regions suggest a continuous trend until March 31, 2021, with the exception of Lima. Although for Lima, the mean tendency  $\hat{y}$  seems to slightly decrease, the interval areas shown a wide number of cases. In fact, this decrease tendency have been observed before for almost all regions. However, instead of a continuous decrease in cases, it was almost followed by a considerable or sudden increment. On the other hand, according with the prediction interval areas, almost all regions will be subjected to possible peaks until the end of March 2021. Figure 5.2 shown these results for all regions.

To obtain an estimate over the forecast behavior during the 15 day window, an average over the maximum, mean and minimum predictions were computed. These averages correspond with the upper, mean and lower intervals obtained by the model. As observed, according with the maximum average, the top five regions are: Lima (8026), Callo (394), Cusco (393), Ancash (367) and La Libertad (364). Mean and minimum tendencies across regions present slightly variations with some identical values due to numeric rounding. Observing the variations in regions, it is noted how the average tendencies could drastically change in such a short window (15 days). This dynamic tendency was present in all regions. Figure 5.3 show the detail averages for all regions.

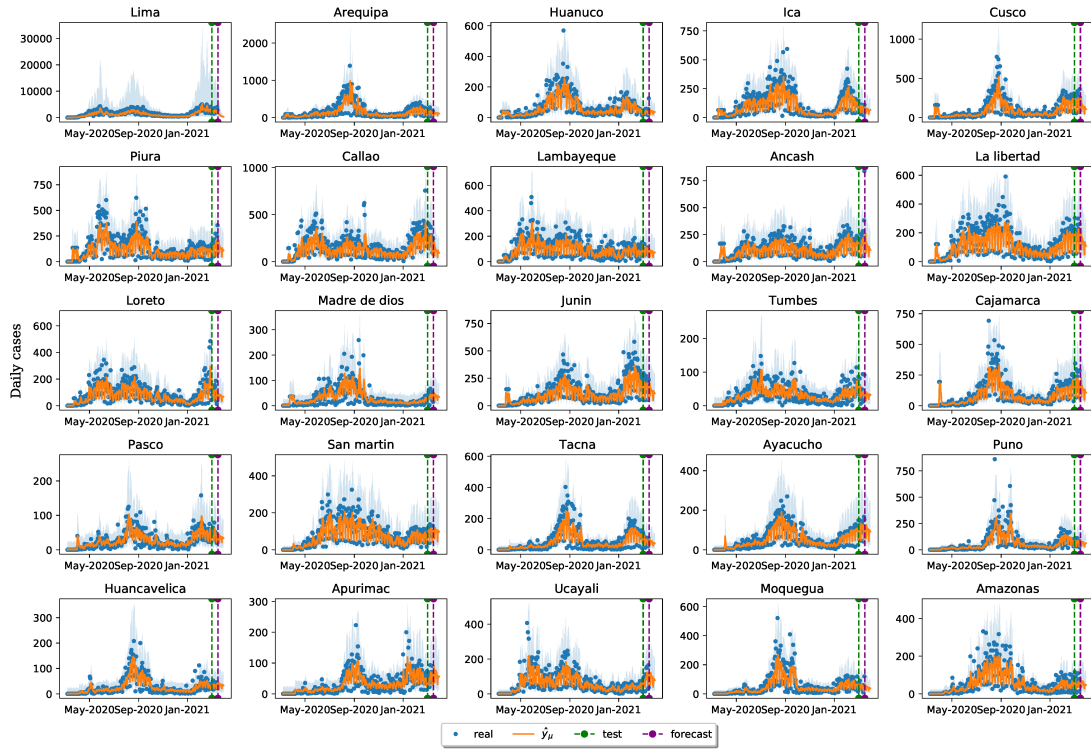


Figure 5.1: Daily SARS-CoV-2 predictions from March 03, 2020 to March 31, 2021. The orange lines represent the mean predictions  $\hat{y}_t$ , while the light blue area shown the 95% prediction interval. The green dashed line indicates the beginning of the test (March 02, 2021 to March 16, 2021). Meanwhile, the purple dashed line denotes the start of forecasting using a 15 day ahead window (March 17, 2021 to March 31, 2021).

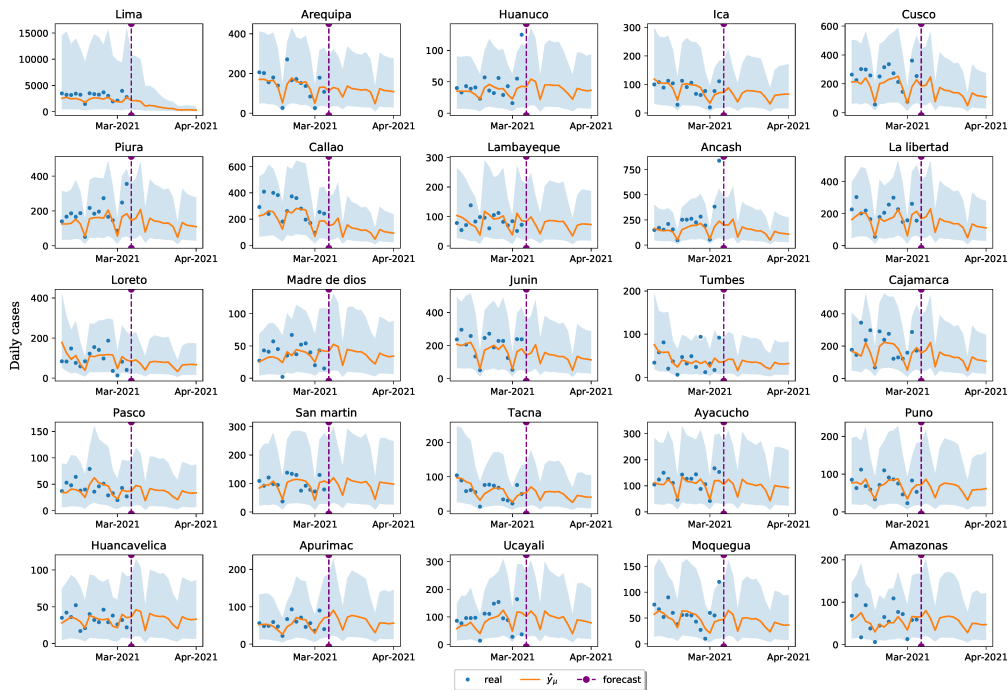


Figure 5.2: A more detailed view of the daily SARS-CoV-2 forecast tendency per each regions. The orange line indicates the mean prediction  $\hat{y}_t$ . Meanwhile the light blue area shown the 95% prediction interval. Forecasting was estimated from March 17, 2021 to March 31, 2021 (purple dashed line).

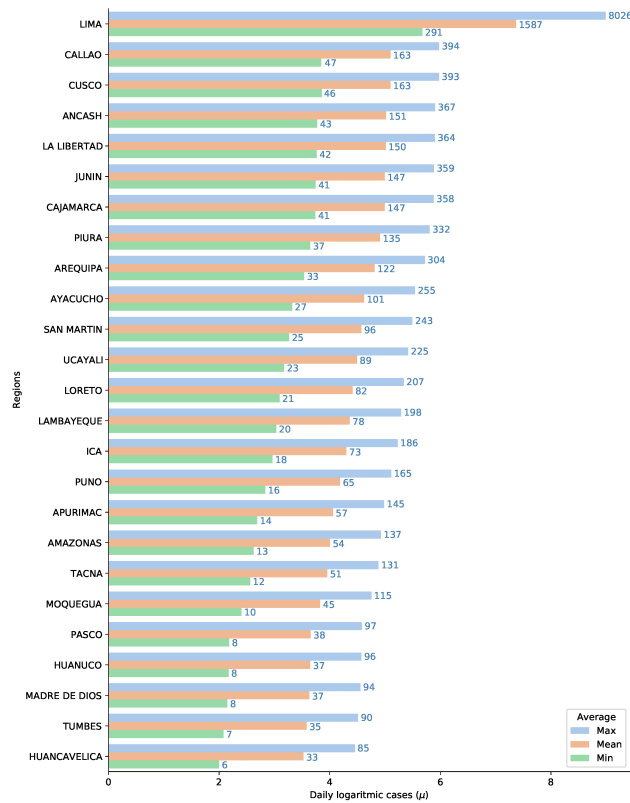


Figure 5.3: Computed averages per each region over the 15 day forecasting window (March 17, 2021 to March 31, 2021). In orange are shown the average estimated  $\hat{y}$ , whereas the blue and green bars represent the average maximum and minimum cases respectively.

**5.3. Evaluation Metrics.** In order to evaluate the model performance, an evaluation strategy was implemented. This strategy evaluated the mean and interval predictions estimated by the model. First MAE, MSLE, RMSLE and MAD were used to measure the mean predictions  $\hat{y}$  (orange lines in Figure 5.1 and Figure 5.2). MAE and MAD allowed to measure how far away are the estimations from the mean. Meanwhile, MSLE measured the model underestimation and RMSLE its robustness against outliers. Next to address the uncertainty presented in the dynamics of SARS-CoV-2 Eq. 4.10 was used. This allowed us to estimate the percentage of real data  $y$  which were captured by the model (area between the lower and upper intervals).

The evaluation strategy was applied to the training and validation data sets. Results on the train partitions (training and validation) reported how well the model learn, whereas the test partitions measured the model generalization to new unseen data. This allowed us to obtaining a global error estimation. Next, the metrics were applied per each region considering the test partition period from March 02, 2021 – March 16, 2021.

First, it was analyzed the model learning capabilities. According with the results reported on Table 5.1 the final model shown an improved from the validation set for MAE of 2.185, MAD report a slightly decrease of 0.287, whereas MSLE and RMSLE improved 0.021 and 0.017 respectively. Results from the train partition are as an estimate to the expected errors in the predictions reported in Figure 5.1 which were computed from March 06, 2020 to March 01, 2021.

Next, evaluating the model generalization ability to new unseen data, Table 5.1 shown an improvement in all metrics as follows: MAE (14.534), MAD (3.123), MSLE (0.042), RMSLE (0.047). Furthermore, results from the test partition are an estimate to the expected errors in the prediction and forecasting reported in Figure 5.1 and Figure 5.2 respectively. From the results on Table 5.1, for the train and test partitions (training data set), one can conclude that the model have learning correctly and is able to generalize to new unseen data.

In order to evaluate the model at regional level, the metrics were computed per each region using the test partition from the training data set described in Table 3.6. Results at this level shown variations per each region. As such, the regions which obtained the lowest metrics were: Huancavelica (MAE: 7.740, MAD: 7.045) and Ayacucho (MSLE: 0.038, RMSLE: 0.195). Meanwhile the highest errors were reported for

Data set	Data partition							
	Train				Test			
	MAE	MAD	MSLE	RMLSE	MAE	MAD	MSLE	RMSLE
Validation	50.298	13.986	0.422	0.650	74.015	22.026	0.222	0.471
Training	48.113	14.274	0.401	0.633	59.482	18.903	0.179	0.423
Improvement	<b>↑2.185</b>	<b>↓0.287</b>	<b>0.021</b>	<b>↑0.017</b>	<b>↑14.534</b>	<b>↑3.123</b>	<b>↑0.042</b>	<b>↑0.047</b>

Table 5.1: Evaluation metrics for training and validation sets. In bold are shown the improvements from the final TCN model.

Lima (MAE: 719.096, MAD: 691.845) and Tumbes (MSLE: 0.531, RMSLE: 0.729). However it is noted that, despite these large errors, the majority of regions fall far below as noted in Table 5.2. This means, the model was able to learn individual trends presented in each region.

REGION	MAE	MAD	MSLE	RMSLE
LIMA	719.096	691.845	0.098	0.313
AREQUIPA	28.092	24.026	0.104	0.323
HUANUCO	14.159	9.848	0.139	0.372
ICA	16.776	15.016	0.077	0.277
CUSCO	59.207	50.106	0.092	0.304
PIURA	50.232	39.245	0.111	0.334
CALLAO	72.981	67.914	0.119	0.345
LAMBAYEQUE	31.721	21.268	0.236	0.486
ANCASH	87.521	30.341	0.192	0.438
LA LIBERTAD	48.217	52.57	0.133	0.365
LORETO	42.876	42.825	0.376	0.613
MADRE DE DIOS	12.782	9.715	0.452	0.673
JUNIN	41.463	30.606	0.059	0.242
TUMBES	23.217	17.837	0.531	0.729
CAJAMARCA	63.542	54.592	0.234	0.483
PASCO	11.381	7.253	0.099	0.315
SAN MARTIN	20.532	22.367	0.066	0.256
TACNA	14.181	8.717	0.13	0.361
AYACUCHO	18.657	10.017	0.038	0.195
PUNO	14.112	12.692	0.059	0.243
HUANCAVELICA	7.74	7.045	0.083	0.289
APURIMAC	14.559	11.067	0.092	0.303
UCAYALI	31.769	28.281	0.249	0.499
MOQUEGUA	19.074	13.179	0.263	0.513
AMAZONAS	23.153	20.014	0.45	0.671

Table 5.2: MAE, MAD, MSLE and RMSLE evaluation metrics per each region. Results were computed using the test partition from the training data set, with a window of 15 days (Mar. 02, 2021 – Mar. 16, 2021).

Finally, the prediction intervals were evaluated per each region according with the metric defined in Eq. 4.10. Table 5.3 shown the percentage of real data points  $y$  which were captured by the model. This evaluation considered the data from the train and test partitions from Table 3.6. Results indicated that the prediction interval area produced by the TCN model was able to capture the 96.1% for train and 97.3% for test respectively. This shown the model robustness when addressing variations in the SARS-CoV-2 cases among regions.

REGION	$P_i - Train(\%)$	$P_i - Test(\%)$
LIMA	100	100
AREQUIPA	94.737	100
HUANUCO	96.122	93.333
ICA	96.953	100
CUSCO	92.798	100
PIURA	94.46	100
CALLAO	94.46	100
LAMBAYEQUE	95.014	100
ANCASH	95.845	93.333
LA LIBERTAD	95.845	100
LORETO	95.291	100
MADRE DE DIOS	95.845	93.333
JUNIN	95.845	100
TUMBES	96.953	86.667
CAJAMARCA	97.23	86.667
PASCO	96.399	100
SAN MARTIN	98.061	100
TACNA	96.399	100
AYACUCHO	98.892	100
PUNO	96.122	100
HUANCAVELICA	96.676	100
APURIMAC	96.676	100
UCAYALI	95.291	100
MOQUEGUA	96.122	86.667
AMAZONAS	94.737	93.333
AVERAGE	96.11	97.333

Table 5.3: Percentage of real data points  $y$  which are captured by the TCN model. Results are shown per each region according with the train and test partitions (training data set).

**6. Conclusions.** A deep learning model to predict and forecast the daily SARS-CoV-2 cases in the Peruvian regions was proposed. As such, a TCN model was trained using the open data set provided by the Health Ministry of Peru (MINSA). The study comprehend data from March 06, 2020 to March 16, 2021 and was able to predict and forecast using a window of 15 days ahead.

The model was trained with a total of 10 features, which were engineering to better capture the daily dynamics of SARS-CoV-2 cases per each region. Since a TCN model was used, the features were combined into a 3D-volume, which was used to feed the model.

To obtain a robust model, the TCN was optimized using a Bayesian approach, which was computed on the validation set. This process was designed to avoid any data leaking from future points into the model hyper-parameters estimation. Empirical evidence on the validation data set shown that a five-layer model was more suitable to learn the daily SARS-CoV-2 tendencies across regions.

Predictions using the train and test partition described in Table 3.6 shown that the model was able to capture the diverse tendencies presented across the regions. To do so, the model used its mean prediction  $\hat{y}$  to model the main tendencies. Also, in order to capture variations, the model used its prediction intervals. Forecasting results shown a constant tendency with indications of peaks until the end of the forecasting window (March 17, to March 31, 2021) for all regions.

Results from the average tendencies (minimum, mean and maximum) shown that the five regions with the highest peaks during the forecasting were: Lima (8026) followed by Callo (394), Cusco (393), Ancash (367) and La Libertad (364).

According with the reported results in the training and validation sets (Table 5.1), the model was successfully able to learn from the train partition. Concretely, the model shown an improvement in all metrics, with the exception of MAD, which presented a slightly decrease of 0.287. On the other hand, the model generalization capabilities shown an improvement in all metrics for the test partition.

Results on each regions shown variations per each metric, where Huancavelica (MAE: 7.740, MAD: 7.045) and Ayacucho (MSLE: 0.038, RMSLE) obtained the lowest results. Meanwhile Lima (MAE: 719.096, MAD: 691.845) and Tumbes (MSLE: 0.531, RMSLE: 0.729) shown the highest ones.

Prediction intervals shown that the model was able to capture the real data points  $y$  in the train partition with an average of 96.11% and 97.33% for the test, showing a robust estimated for the possible variations.

The presented model can be used by any region as a tool to evaluate the dynamic tendencies in the daily SARS-CoV-2 cases. Moreover, the model can be applied as part of the decision making of policies. As such, it is recommended to continuously training the model, especially when more data became available. This will assure more accurate estimates.

Finally, the presented model did not address any information of the ongoing vaccine campaign on Peru. At the moment of finish this study, people with high risk were under vaccination. Also the new SARS-CoV-2 P.1 variant (20J/501Y.V3) detected in Lima was not consider into this study, due to its apparition in the last 3 days of data. In order to include those variance, a new model with update data must be developed.

### ORCID and License

Luis Aguilar I. <https://orcid.org/0000-0003-4272-2848>

Miguel Ibáñez-Reluz <https://orcid.org/0000-0002-0722-4643>

Juan C. Z. Aguilar <https://orcid.org/0000-0002-7000-8089>

Elí W. Zavaleta-Aguilar <https://orcid.org/0000-0003-3129-5975>

L. Antonio Aguilar <https://orcid.org/0000-0002-1555-0748>

This work is licensed under the [Creative Commons - Attribution 4.0 International \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/)

## References

- [1] Bchetnia M, Girard C, Duchaine C, Laprise C. The outbreak of the novel severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2): A review of the current global status. *J Infect Public Health*. 2020 Nov; 13(11):1601-1610. doi: 10.1016/j.jiph.2020.07.011.
- [2] Harrison SL, Fazio-Eynullayeva E, Lane DA, Underhill P, Lip GYH. Comorbidities associated with mortality in 31,461 adults with COVID-19 in the United States: A federated electronic medical record analysis. *PLoS Med*. 2020 Sep 10; 17(9): e1003321. doi: 10.1371/journal.pmed.1003321.
- [3] Giannis D, Zogas IA, Gianni P. Coagulation disorders in coronavirus infected patients: COVID-19, SARS-CoV-1, MERS-CoV and lessons from the past. *J Clin Virol*. 2020 Jun; 127: 104362. doi: 10.1016/j.jcv.2020.104362.
- [4] MINSA. Documento técnico atención y manejo clínico de casos de covid-19 - escenario de transmisión focalizada [Internet]. 2020 [cited 2021 Jan 29]. Available from: <http://www.insnsb.gob.pe/documentos-minsa-covid-19/>
- [5] MINSA. Alerta epidemiológica código: AE-016- 2020 [Internet]. 2020 [cited 2021 Jan 29]. Available from: <https://www.dge.gob.pe/portal/docs/alertas/2020/AE016.pdf>
- [6] ESSALUD. Data COVID-19 - Reporte diario [Internet]. 2020 [cited 2021 Feb 04]. Available from: <https://apps.essalud.gob.pe/data-covid-19/>
- [7] MINSA. Gis Visor Vacunados [Internet]. 2020 [cited 2021 Mar 21]. Available from: <https://gis.minsa.gob.pe/GisVisorVacunados/>
- [8] Roser M, Ritchie H, Ortiz-Ospina E, Hasell J. Coronavirus pandemic (COVID-19) [Internet]. 2020 [cited 2021 Feb 04]. Available from: <https://ourworldindata.org/coronavirus>



- [9] MINSA. El Ministerio de Salud detectó la presencia de la variante brasileña del coronavirus en Loreto, Huánuco y Lima [Internet]. 2021 Feb 04 [cited 2021 Mar 21]. Available: <https://www.gob.pe/institucion/minsa/noticias/341090-el-ministerio-de-salud-detecto-la-presencia-de-la-variante-brasilena-del-coronavirus-en-loreto-huanuco-y-lima>
- [10] Abou-Ismaïl A. Compartmental models of the COVID-19 pandemic for physicians and physician-scientists. *SN Compr Clin Med*. 2020 Jun 4; 2: 852-858. doi: 10.1007/s42399-020-00330-z.
- [11] Cooper I, Mondal A, Antonopoulos CG. A SIR model assumption for the spread of COVID-19 in different communities. *Chaos Soliton Fract*. 2020 Jun 28; 139: 110057. doi: 10.1016/j.chaos.2020.110057.
- [12] He S, Peng Y, Sun K. SEIR modeling of the COVID-19 and its dynamics. *Nonlinear Dyn*. 2020 Jun 18; 101:1667-1680. doi: 10.1007/s11071-020-05743-y.
- [13] Anastassopoulou C, Russo L, Tsakris A, Siettos C. Data-based analysis, modelling and forecasting of the COVID-19 outbreak. *PLoS One*. 2020 Mar 31; 15(3): e0230405. doi: 10.1371/journal.pone.0230405.
- [14] Dong E, Du H, Gardner L. An interactive web-based dashboard to track COVID-19 in real time. *Lancet Infect Dis*. 2020 Feb 19; 20(5): 533-534. doi: 10.1016/S1473-3099(20)30120-1.
- [15] Hochreiter S, Schmidhuber J. Long Short-Term Memory. *Neural Comput*. 1997 Nov; 9(8):1735-1780. doi: 10.1162/neco.1997.9.8.1735.
- [16] Chimmula VKR, Zhang J. Time series forecasting of COVID-19 transmission in Canada using LSTM networks. *Chaos Soliton Fract*. 2020 Jun; 135:109864. doi: 10.1016/j.chaos.2020.109864.
- [17] Zeroual A, Harrou F, Dairi A, Sun Y. Deep learning methods for forecasting COVID-19 time-Series data: A comparative study. *Chaos Soliton Fract*. 2020; 140: 110121. doi: 10.1016/j.chaos.2020.110121.
- [18] Mohimont L, Chemchem A, Alin F, Krajecki M, Steffanel L. Convolutional neural networks and temporal CNNs for covid-19 forecasting in France. *Appl Intell*. 2021 Apr 14: 1-26. doi: 10.1007/s10489-021-02359-6.
- [19] Huang CJ, Chen YH, Ma Y, Kuo PH. Multiple-input deep convolutional neural network model for COVID-19 forecasting in China. *medRxiv*. 2020 Mar 23, Pre print. doi: 10.1101/2020.03.23.20041608.
- [20] Abbasimehr H, Paki R. Prediction of COVID-19 confirmed cases combining deep learning methods and Bayesian optimization. *Chaos Soliton Fract*. 2021; 142:110511. doi: 10.1016/j.chaos.2020.110511.
- [21] Bai S, Kolter JZ, Koltun V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv*. 2018; 19:1803.01271v2.
- [22] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; 2016 Jun 26-30; Las Vegas, NV, USA. IEEE; 2016. p. 770-778. doi: 10.1109/CVPR.2016.90.
- [23] MINSA. Casos positivos por COVID-19 - [Ministerio de Salud - MINSA] [Online]. 2020 [cited 2021 Mar 20]. Available from: <https://www.datosabiertos.gob.pe/dataset/casos-positivos-por-covid-19-ministerio-de-salud-minsa>
- [24] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Lawrence N, Reid M, editors. *Proceedings of the 32nd International Conference on Machine Learning*; 2015 Jul 7-9; Lille, France. PMLR v37; 2015. p. 448-456.
- [25] Nair V, Hinton GE. Rectified linear units improve restricted Boltzmann machines. *Proceedings of the 27th International Conference on International Conference on Machine Learning*. 2010 Jun 21-24; Haifa, Israel. p. 807-814.
- [26] Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv*. 2012 Jul 03 : 1207.0580v1.
- [27] Sutskever I, Martens J, Dahl G, Hinton G. On the importance of initialization and momentum in deep learning. *Proceedings of the 30th International Conference on International Conference on Machine Learning*; 2013 Jun 17-19; Atlanta, Georgia, USA. PMLR v28; 2013. p. 1139-1147.
- [28] Balandat M, Karrer B, Jiang DR, Daulton S, Letham B, Wilson AG, Bakshy E. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. *arXiv*. 2020 Dec 08 : 1910.06403v3.
- [29] Smith LN. Cyclical learning rates for training neural networks. *Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*; 2017 Mar 21-24; Santa Rosa, CA, USA. IEEE; 2017. p. 464-472. doi: 10.1109/WACV.2017.58.
- [30] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. Teh YW, Titterton M, editors. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*; 2010 May 13-15; Chia Laguna Resort, Sardinia, Italy. PMLR v9; 2010. p. 249-256.