Simultaneous Time Series Forecasting on the World's COVID-19 Daily Vaccinations

[Extended Abstract]

Konstantinos Georgiou
The University of Tennessee, Knoxville
Knoxville, TN, United States
kgeorgio@vols.utk.edu

Michela Taufer
The University of Tennessee, Knoxville
Knoxville, TN, United States
taufer@utk.edu

ABSTRACT

The global distribution of the COVID-19 vaccines is one of the most challenging tasks the modern health industry has ever faced. Predicting the number of a country's daily vaccinations can be vital for its protecting its economy and adapting its policy response. We present a pipeline that trains on all countries' historical vaccination data and attempts to predict their next ten days of daily vaccinations by utilizing their underlying relationships. We use Encoder-Decoder Long Short-Term Memory Networks with walk-forward validation and evaluate the results using mean, per-country, and per-date RMSE.

CCS Concepts

•Computing methodologies → Neural networks; •Applied computing → Health care information systems;

Keywords

Multivariate Time Series; Time Series Forecasting; Recurrent Neural Networks; Long-Short Term Memory Networks; COVID -19

1. MOTIVATION

The COVID-19 pandemic is one of the most impactful events of recent human history with profound effects on the health of billions and the economies of almost all of the world's countries. The imperative need for the rapid development and global distribution of the COVID-19 vaccines generated one of the most complex tasks of modern public health history. The urgency and significance of this task have attracted the interest of a large portion of the scientific community as there are many insights that can be obtained from these data.

One insight that can be found very useful, is the prediction of a country's future daily vaccinations. By doing so, we could identify potential future reductions of the rate of some countries' vaccinations, which can give their governments the opportunity to control possible complications.

2. CONTRIBUTIONS

Many attempts have been made to extract useful insights from these data, but most of them are comparative analyses between two or more countries' historical vaccination records, and some of them attempt to predict the number of future vaccinations of one country at a time. We present a method that utilizes all countries' historical vaccination data in order to identify causal relationships between them and make simultaneous predictions on their next ten days of daily vaccinations. We do heavy preprocessing to fix the null values of the dataset, we then use an external dataset to recalculate the per hundred people values of the vaccination columns, we train an Encoder-Decoder Long Short-Term Memory Network, and finally evaluate the predictions using the average, per-country, and per-date RMSE.

The rest of the paper is organized as follows: Section 3 describes the dataset, preprocessing methodology, and evaluation techniques used in this study; Section 4 describes and discusses the results; Section 5 summarizes our findings and proposes future improvements.

3. METHODOLOGY

In this section, we describe the datasets used and the framework we developed to predict the countries' future daily vaccinations.

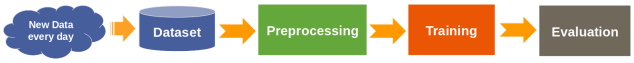

Figure 1: General Pipeline.

3.1 Dataset

The main data source used, which is the COVID-19 World Vaccination Progress dataset, contains information about the daily and total vaccinations of 193 different countries over 135 different dates. The data are being collected almost daily and of writing this paper, the dataset has 14230 rows and 15 different features. The main features used for the training are the the daily vaccinations, the total vaccinations, the number of people vaccinated, the number of people fully vaccinated, and their respective per hundred citizen values as well as the country id and the date. We made predictions on the daily vaccination per hundred citizen values.

Additionally, an auxiliary dataset was used, namely DataBank - World Development Indicators. This data source contains numerous useful static features about the world's countries, such as health expenditure per gdp and hospital beds per 1000. In our analysis. the country population

Absolute Columns	Percentage Columns
Daily Vaccinations	Daily Vaccinations per 100
People Vacc.	People Vacc. per 100
People Fully Vacc.	People Fully Vacc. per 100
Total Vaccinations	Total Vaccinations per 100

Table 1: The columns we kept before starting the preprocessing steps.

was used to recalculate the per hundred citizen values of the columns we had to infer their null values.

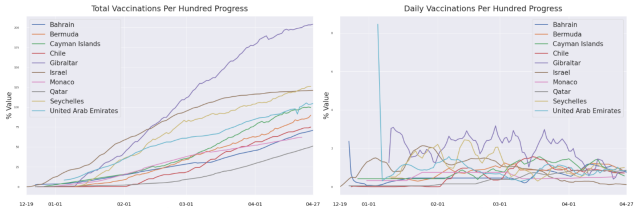

Figure 2: Top 10 Countries in terms of total (left) and daily (right) vaccinations per hundred citizens.

3.2 Preprocessing

We are performing three distinct preprocessing steps on our dataset. Before starting, we select to keep nine out of the fifteen columns and we set select one of them (the date) as our index. We keep the following columns:

3.2.1 *Fixing the Missing Values*

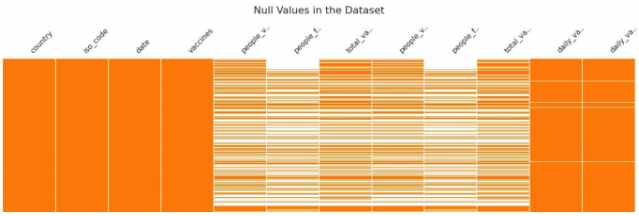

Figure 3: Null values in the dataset's features. The white gaps represent the null values.

In the preprocessing phase of the pipeline, we first fix the null values of the daily vaccinations (daily), total vaccinations (total), people vaccinated (vacc), and people fully vaccinated (f_vacc) columns. To do so, we define and use the following relationships:

$$daily_{current} = total_{current} - total_{previous} \quad (1)$$
$$total_{current} = vacc_{current} + f\_vacc_{current} \quad (2)$$
$$total_{current} = total_{previous} + daily_{current} \quad (3)$$
$$total_{current} = total_{next} + daily_{next} \quad (4)$$
$$vacc_{current} = total_{current} - f\_vacc_{current} \quad (5)$$
$$f\_vacc_{current} = total_{current} - vacc_{current} \quad (6)$$

Using the above equations repeatedly until the number of missing values does not change and then inferring the rest of the values by averaging the next and the previous values, we are able to infer the 100% of the missing values. We should note that the above operations are being performed separately for each country's records.

To recalculate the per hundred citizen values of the above columns, we need the population of each country. We first load an external dataset that has this information and preprocess it. After ensuring that the missing countries from the external data do not have many records in ours, we use the population of each country and the previous fixed columns to recreate their per hundred citizen values.

3.2.2 *Cleaning and Normalizing*

Because we are going to make predictions based on the underlying relationships between the countries, it's import that we don't keep dates where we don't have records for many countries yet. For that reason, we identify the number of countries with valid records one month prior to the latest record, and we drop the most recent dates whose number of countries with valid records is less than 90% of that number.

As all the features we are going to use for the training are numerical, we had to scale them all before feeding them to the network. After trying scaling them in the [0, 1] range, we saw that the accuracy was dropping compared to larger ranges, indicating that the very low values of some features were causing the network to lose important information due to its float precision limitations. During hyper-parameter tuning, we found that the optimal range was [0, 1000000]. The normalization was performed on a per-country basis.

3.2.3 *Reshaping the Dataset*

Next, for each column column we have, we create a new instance of it for each country present in the dataset. This way, we transform our data so that the number of rows is equal to the number of dates, and the number of columns is equal to the number of countries times the number of columns. This way, we are able to feed the daily records of all countries simultaneously for each time-step and thus give it the opportunity to find correlations between them. This operation transformed the dataset from a shape of 11882 rows and 10 columns to a shape of 126 rows and 1409 columns. The 10th and column before reshaping and the 1409th after reshaping is the date. The columns of the dataset after reshaping it look like this:

Country 1 Columns	..	Country N Columns
Daily Vacc. Country_1	..	Daily Vacc. Country_N
Vacc. Country_1	..	Vacc. Country_N
Fully Vacc. Country_1	..	Fully Vacc. Country_N
Total Vacc. Country_1	..	Total Vacc. Country_N
% Daily Vacc. Country_1	..	% Daily Vacc. Country_N
% Vacc. Country_1	..	% Vacc. Country_N
% F. Vacc. Country_1	..	% F. Vacc. Country_1
% Total Vacc. Country_1	..	% Total Vacc. Country_N

Table 2: The columns after the preprocessing steps. There are 1409 columns in total (original columns times number of Countries).

The last preprocessing step of our methodology includes another reshaping of the dataset, which this time is the grouping of rows/dates in 10-day windows. We have 126 days of data entries in total, so we create 12 10-day groups and drop the last 6 days of it. We do this because we are

going to train our network using Walk-Forward Validation.

3.3 Training

Using Keras and Tensorflow, we created a sequential model with two Long Short-Term Memory layers and two Dense Layers. The input shape of the network is (time window size, # features) = (10, 1408) because we will train in 10-day time steps and we have one less feature because we are not including the date column during training. The first and the second layer have 800 units each, while the Dense layers have 400 and 1408 (the number of features), respectively. The implementation of the Walk-Forward Validation is being handled by the TimeDistributed layer. The model is shown in Figure 4.

We use the Adam optimizer and the mean squared error as a loss function. Before training, we split the dataset into a train (80% - nine ten-day groups) and a test (20% - three ten-day groups) set. After training, we use the trained model and generate a prediction dataset with equal size as the test set.

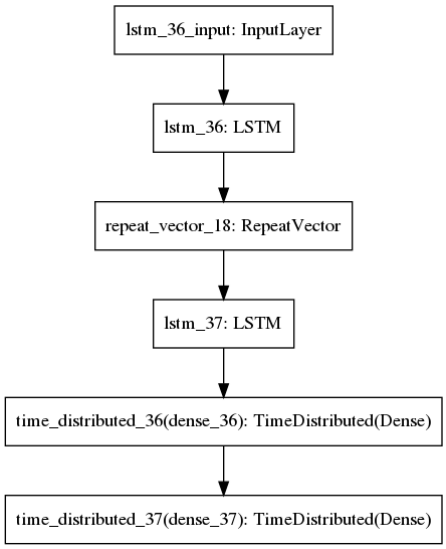

Figure 4: The network we trained our dataset on.

3.4 Evaluation

To evaluate the results, we need to reconstruct the test and prediction sets in the original format. Both sets are three-dimensional, where the first dimension is the number of ten-day windows, the second is the number of days in each window (which is always ten) and the last one is the number of features. First, we flatten the fist and the second dimension, so from the shape of (3, 10, 1408) we end up with a (30, 1408) shape. Then, we reshape the two sets back to the original formatting where we had multiple date entries (one for each country) and only nine columns plus the date column. We end up with a shape of (4598, 10).

We only want to predict the daily vaccinations per hundred citizens column, so we only keep this and the date column. To evaluate the results, we use the root mean square error and we calculate: the total average RMSE, the average RMSE per country (170 values), and the average RMSE per date (30 values).

4. RESULTS

We trained the deep neural network on the first ninety days of the dataset and made predictions on the last thirty days of it. Even though our model output includes all the input features, since we only want to predict the daily vaccinations per hundred citizens, we make our evaluation for that column. By calculating the RMSE between the test set and our predictions for each data entry, we got an average of 0.24617. The average RMSE per date and per country ARE shown in Figures 5 and 6, respectively. Out of the 170 countries, 164 have mean RMSE less than 1.0, 153 less than 0.5, and 73 less than 0.1. The min value is 0.00023 and max is 5.5133. Out of the 30 dates, 25 have mean RMSE less than 0.30, 15 less than 0.25, and 11 less than 0.20. The min value is 0.16928 and the max is 0.3518. Additionally, in Figure 6, we show the true versus predicted values for 12 different countries.

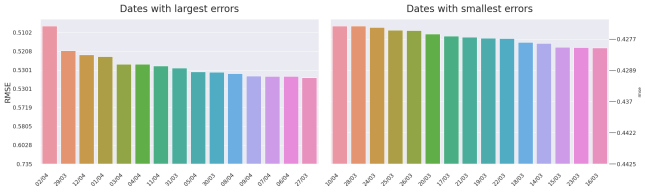

Figure 5: Dates with the largest (left) and smallest (right) errors.

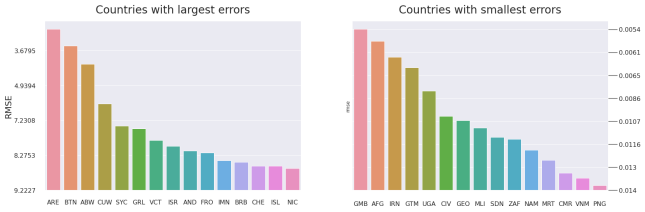

Figure 6: Countries with the largest (left) and smallest (right) errors.

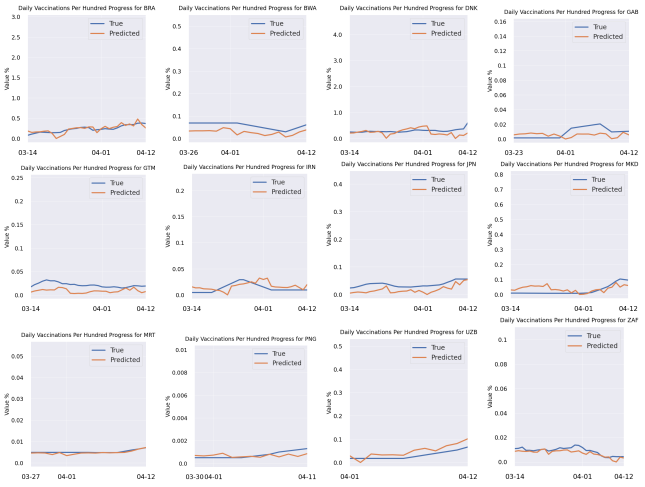

Figure 7: The true versus the predicted daily vaccinations per hundred for 12 different countries.

5. CONCLUSIONS AND FUTURE WORK

Our method showcases what type of preprocessing is necessary to make the COVID-19 vaccination process dataset ready to be used in a time-series analysis pipeline, and a proof-of-concept multivariate network that can train on this dataset with ten-day training windows. We also show that it is possible to predict the daily vaccinations per 100 citizens with satisfactory results using Multivariate Long Short-term Memory Networks for small datasets like the one we used.

Future directions for this work include trying custom loss functions for training, incorporating static features of the countries such as the health expenditure per GDP, and training using one LSTM network per country, and combining their weights using the Functional API of Tensorflow.

6. REFERENCES