Development of an early warning system for emerging pandemics based on deep learning

Guillem Samper Argelagués

June 30, 2025

Abstract

This project explores the use of deep learning to build an early warning system for emerging pandemics. It focuses on predicting COVID-19 case trends using time-series models (LSTM and RNN) and evaluates whether these models can simulate the impact of public health policies on the evolution of the pandemic. Multiple architectures were tested, including encoder-decoder and autoregressive designs, using different kinds of policies as features. The models successfully predicted new cases using past case data alone, achieving strong performance metrics. However, they failed to incorporate policy features meaningfully. As a result, the goal of simulating alternative scenarios where different kinds of policies are implemented could not be fulfilled. The findings suggest that case evolution is predictable without contextual data, but this limits the potential for other use cases.

Keywords: pandemic modeling, time series forecasting, LSTM, RNN, public health policy

1 Introducction - Context of the Project

Researching the possible ways to approach the issue of the development of an early warning system for emerging pandemics based on deep learning, I found a particularly interesting area of study. During the COVID-19 pandemic, a large volume of data was generated regarding the evolution of the pandemic: cases, deaths, vaccinations, hospital occupancy, among others. All this data is publicly accessible, making it suitable for the development of deep learning models. These models can contribute to improving the management of future pandemics. From a health crisis that has had so many negative consequences, it is essential to extract as many lessons as possible. Thus, in case we encounter a similar situation again, the response would be more efficient and better prepared. This work aims to contribute in this direction, with the objective of optimizing the forecasting and response mechanisms for new pandemics. The specific objectives of this work are as follows:

- Analyse various state-of-the-art methodologies to identify the most suitable one for solving the problem.
- Train a deep learning model capable of predicting the number of positive cases and the level of hospital occupancy.
- Establish the relationship between the level of pandemic impact and the various preventive measures applied by different countries.

The following report starts with an overview of the state of the art in pandemic forecasting, followed by a detailed description of the data sources used and the preprocessing steps applied. It then outlines the different methodologies implemented throughout the project, analyses the results obtained, and concludes with a discussion of the main findings and proposed future lines of investigation.

2 State of the art

The review of existing literature on the topic focuses on data-driven solutions. Traditional approaches rely on mathematical modelling to predict the effects of pandemics. One of these models is the Susceptible-Infected-Removed (SIR) model, which offers a theoretical framework to analyse the spread of an epidemic within a community. Successful work in this area has been done, showing that it is possible to explain the spread of a pandemic using these kinds of mathematical model[1].

However, as noted in [2], SIR models have limitations, typically exhibiting only three dynamics: a single epidemic wave, convergence to an endemic state, or periodic waves.

• Contact E-mail: guillem.sampera@autonoma.cat
• Supervised by: Ernest Valveny Llobet (Departament de Ciències de la Computació) and Victor Jose Yuste Mateos (Departament de Bioquimica i Biologia Molecular)
• Academic Year 2024/25

The standard SIR model is even more restricted, always following the same qualitative behaviour. Given the complex real-world COVID-19 patterns, incorporating global dynamics is essential. The cited study extends the SIR model to address these limitations [2]. This study also considers three possibilities in the SIR model for COVID-19, susceptible individuals from a local community can travel in and out of their community without any exposure to COVID-19, they can be exposed to COVID-19 while travelling and develop symptoms after they return to their community, or they can be diagnosed with COVID-19 during their travelling and return to their community after recovery.

In the field of deep learning, pandemic prediction has been modelled as a time-series problem. As a result, architectures such as Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks are commonly used to address this issue, as demonstrated in [3] and [4] where the main focus was to improve the prediction of COVID-19 cases. In other studies [5], the effects of preventive measures are examined to complete the data given to the Deep Learning models.

The findings support the direction of this project. As the goal was to use deep learning methods the works reviewed on time series modelling with RNNs and LSTMs are the most relevant. Moreover, the inclusion of contextual information, such as public health policies, has been explored in order to improve the predicting capabilities. This motivates the two main objectives of this project: not only to predict case trends using deep learning techniques, but also to investigate whether these models can simulate the potential impact of different policy decisions.

3 Data

The first step of this project was to find reliable data sources. The objective was to gather a wide range of features from as many countries as possible. Using diverse data would result in a model more capable of generalizing. All the data used in this project comes from two sources.

From the first data source, information was obtained on the progression of the pandemic itself, including variables such as the number of cases, deaths, and hospital occupancy and also the progression in the vaccination process, which is considered policy data. This data is structured temporally and organized by country.

The second, which contains information about the public policies taken by the governments during the evolution of the epidemic, is taken from the Oxford COVID-19 Government Response Tracker [6]. It covers three years of public policy settings for more than 180 countries reporting different types of indicators such as school closures, travel restrictions, mask mandates and vaccination policies. The raw data can be found in their GitHub repository [7].

The features extracted from these two data sources can be divided into two groups, the first, which are features about the level of pandemic impact itself like hospitalizations, deaths or new cases. Only the new cases are used for the scope of this project. The other type of features are called contextual features, they give information about the actions that public institutions took during the evolution of the pandemic. It includes policies such as school closures mask mandates or income support, along with data on the progression of vaccinations.

During the project, when referring to policy features it means the 4 indices which are the combination of different indicators.

There are the three categories of indicators:

- **Containment and Closure**: Eight individual indicators about the closing of School, workplace and public transport, restriction on internation travel, gathering size, public events and internal movement, and stay at home requirements.
- **Economic Response**: Two individual indicators, income support and debt/contract relief for households.
- **Health Systems**: Six individual indicators about public information campaigns, testing policies, contact tracing, facial coverings, vaccination policies and the protection of elderly people.

And these indicators are combined into 4 indices:

- **Government response index**: Combines all the the mentioned indicators.
- **Containment and health index**: All indicators except the economic support ones.
- **Stringency index**: All indicators on containment and closure and the public information campaigns from health systems.
- **Economic support index**: Combines indicators about the level of income support from public organizations and the debt relief for private households.

Each index has a normalised value between 0 and 100 and the indicators have discrete values that can range between 0 to 2, to 3 or to 4, depending on each indicator.

3.1 Data pre-processing

This data didn't come ready for training out of the box. As more than one data source was used, the first step was to filter out countries that had incomplete data in any of the two sources. Then, remove data points in the beginning and end of the time which is covered, which represents the start and finish of the pandemic, for which there are no reported cases. It was considered that these data points were not meaningful for the model to learn.

The other major preprocessing step came after attempting to implement the initial methods, which will be explained in the following section. Initially there was a data point for each day, reporting the new cases and the level of policies implemented at that point. At first, the goal was to forecast cases on a daily basis, this approach proved to be ineffective due to the inconsistency and quality of the data. The reporting of cases varied depending on the country, and in general, cases were not reported every day. This led to a large number of entries being zero, making it impossible for any model to yield meaningful predictions. For instance, in the data from Spain, 85.84% of the daily entries were zero. A graphical representation of the evolution of the cases can be seen in Figure 1.

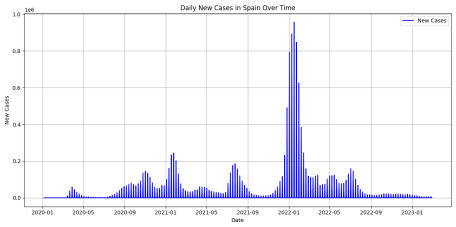

Fig. 1: Line plot of the normalized number of daily reported COVID-19 cases in Spain, from January 2020 to January 2023. It can be appreciated that the line keeps dropping to zero.

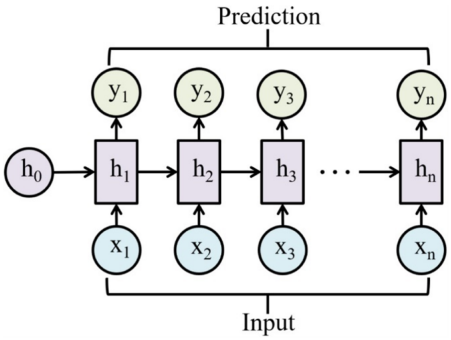

Fig. 2: Basic RNN architecture [8]

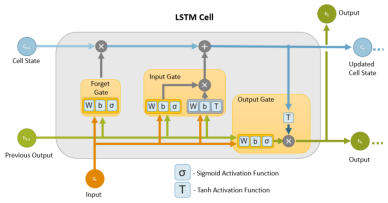

Fig. 3: Architecture of a LSTM cell [10]

4 METHODOLOGY

The methodology developed in this project can be divided into two lines of work, each attempting to fulfil each the project objectives. On the one hand, it focuses on predicting the number of COVID-19 cases using deep learning methods such as Recurrent Neural Networks (RNNs) and Long Short Term Memory (LSTM) networks. These models were trained on time series data which contained features about the evolution of the number of new cases and the different policies implemented by governments, as explained in Section 3. On the other hand, it explores the possibility that these models can explain how the policy variables can influence the evolution of new cases. With the objective of evaluating whether or not these models can serve as tools to simulate how changes in implemented policies might affect the number of new cases during a pandemic. The following sections describe in detail the methods and adaptations employed for each of these two tasks.

4.1 Predicting cases

The following two methods (RNNs and LSTM) were used to try to predict the evolution of the number of new cases.

4.1.1 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a type of neural network architecture which, unlike traditional feed-forward neural networks that pass information through the network without cycles, the RNN contains cycles that allow information to be passed back into the network, as can be seen in Figure 2. At each time step t, the RNN takes an input vector $\mathbf{x_t}$, and updates its hidden state, $\mathbf{h_t}$, using the following equation:

$$h_t = \sigma_h(W_{xh}x_t + W_{hh}h_{t-1} + b_h \qquad (1)$$

where $\mathbf{W}_{xh}$ is the weight matrix between the input and hidden layer, $\mathbf{W}_{hh}$ is the weight matrix for the recurrent connection, $\mathbf{b}_h$ is the bias vector, and σ_h is the activation function. The output at each time step, t, is given by the following:

$$\mathbf{y}_t = \sigma_y(\mathbf{W}_{hy}\,\mathbf{h}_t + \mathbf{b}_y) \qquad (2)$$

where $\mathbf{W}_{hy}$ is the weight matrix between the hidden and output layers, $\mathbf{b}_y$ is the bias vector, and σ_y is the activation function for the output layer.[8]

This enables them to take into account not just the current input but also the previous ones. [9]. This feature allows them to maintain a memory of previous inputs, which makes them ideal for a time-series problem like the one here, where the context and the order of data points are crucial [8].

4.1.2 LSTM

Long Short-Term Memory (LSTM) networks are a variant of recurrent neural networks (RNNs) designed to address their limitations in capturing long-term dependencies, which are caused by the vanishing gradient problem. In theory, the hidden state h_t in each RNN cell can keep track of the information for an arbitrary amount of past time steps. However, as the hidden state h_t is updated at each time step, makes it difficult to capture long-term dependencies. Neural networks are trained by backpropagation of an error measure and adjust the weights of the matrices and biases such that the error is reduced for future iterations. When a network becomes too deep, gradients tend to explode or vanish. This problem causes the RNNs to struggle to learn long-term dependencies in data [10].

LSTMs improve the memory capacity of the RNNs by introducing gates into the memory cells, which helps handle the vanishing gradients problem. The architecture is similar to an RNN, but the recurrent cells are replaced with LSTM cells. These cells also use the previous output recursively, but additionally, they keep track of an internal cell state $\mathbf{c_t}$, which is a vector that handles the long-term memory. This means that the LSTM has access to short-term memory via the hidden state $\mathbf{h_t}$, and long-term memory via the cell state $\mathbf{c_t}$ [10].

As shown in Figure 3, the content of the cell state is updated through operations by gates inside the LSTM cell. These gates are the input, output, and forget gate, along with the activation function, are used to model LSTMs and learn the behaviour of temporal correlations. [11]

4.1.3 Implementation details

For the two methods explained in Section 4.1.1 and Section 4.1.2 the training was done with full teacher forcing. Different sets for testing were used, which will be explained in more detail in Section 5.2. The parameters for the results presented were obtained by doing a hyperparameter search. As explained in Section 3.1 the predicted cases represent weekly cases.

4.2 Simulations

The objective of this part of the project was to establish a relationship between the level of pandemic impact and the various preventive measures applied by the governments of the different countries.

If any of the following methods prove to be effective, they could be used as a tool to run simulations by modifying the input data to observe how this affects the predicted number of new cases. Then, several interesting questions could be asked and proved: how would the curve of cases change if these other policies were implemented instead of those that were actually implemented? or which policies should be implemented in order to minimize the number of cases? These questions proved to be challenging to answer, so a number of different methods were tried so as to get a good result.

Change policy data

Here, the trained models that predict the number of new cases explained in Section 4.1.1 and Section 4.1.2 are reused, but with one important difference: the input data corresponding to government policies is manually changed. To illustrate, a random country from the test set is selected, and its policy-related features are radically modified for example, by setting all the policy indicators to zero. This altered input is then passed through the model to examine the sensitivity of the predictions to changes in these contextual inputs. While this is a radical example, it is useful to assess the response of the model to changes in policy inputs and to explore the potential of these models for running simulations on the impact of the policies.

Distinguishing features and autoregressive training

In this method, a distinction is introduced between the features of the actual level of pandemic impact, in this case the number of new cases, and the contextual features, which are the policies that were implemented by each government. Initially the input just has data about the level of pandemic impact, this is encoded via an LSTM layer, the contextual features are added to this encoded representation and all this goes through a fully connected layer that returns the final output prediction. Here, the same model architecture as the one used to predict cases is used, but introducing a distinction on how the different types of features are used and processed by the model.

This design is thought to be particularly useful in scenarios like this one, where the policy indicators are contextual features which are meant to condition the prediction rather than being modelled.

Additionally, as the objective of these changes is to be able to run simulations on custom data, the training procedure was also adapted. Instead of training with full teacher forcing, where all the predictions are generated with the ground truth, an autoregressive approach was introduced. The method used was exponential decay, where, as training progresses it is more probable that it feeds to the model its own past predictions rather than the ground truth. The last epochs of the training are always done with full autoregression. This approach is designed to obtain a model capable of providing meaningful forecasts based on its own predictions.

See Figure 4 for a visual representation of these changes.

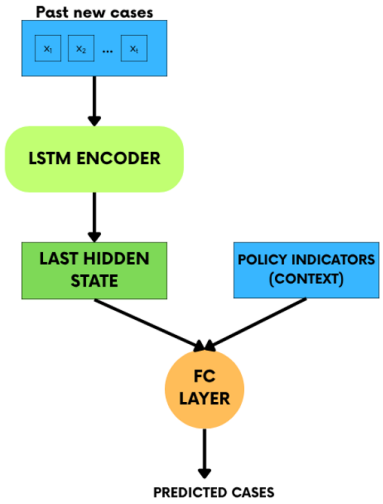

Fig. 4: Model architecture for single-step prediction using LSTM and policy features. The past new cases are processed by the LSTM encoder, and the resulting hidden state is concatenated with the policy indicators. This combined representation is passed through a fully connected layer to produce the final prediction.

Sequence to sequence LSTM

To simplify the challenge of generating simulations using autoregression, the following method approaches the problem from a different perspective. Rather than simulating a sequence of predictions step by step,which can lead to error accumulation over time, this method proposes predicting a fixed number of future steps based on a given window of past observations. This change reduces the complexity of the problem and aims to make it more stable. A practical example of this approach for a pandemic: if data has already been collected up to a certain point, it can be used to forecast the progression of cases over the coming weeks.

The architecture of the LSTM also changes from the previous methods. It is an encoder-decoder architecture. The number of past time steps and the amount of forecasted time steps has to be set at training time. The trained model will only be capable of running under those two parameters at simulation time.

Once again, the features representing the level of pandemic impact are differentiated from the contextual feature. The first type of features is passed through the encoder and then, before it goes through the decoder, the contextual features are added. The idea behind this method is to encode the past time steps to a hidden dimension which can be set, and then add information about the context before return-

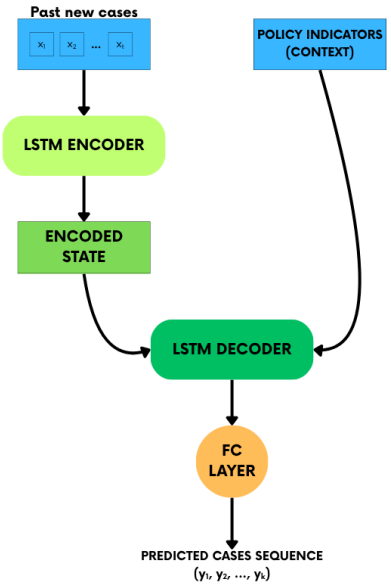

Fig. 5: Diagram of the encoder–decoder LSTM architecture. The past values of new cases are encoded through the LSTM encoder, producing a hidden representation that, together with the future policy indicators, is passed to the LSTM decoder. The decoder outputs a sequence of predicted case values through a fully connected layer.

ing a final prediction. The dimension of the output from the LSTM will be the number of time steps that we want to forecast and the input dimension the number of past time steps used as data for the prediction.

See Figure 5 for a visual representation of this architecture.

5 Experiments and Results

5.1 Metrics

In order to illustrate the results of the different methods and compare them, three main metrics are used: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and the coefficient of Determination (R^2).

Mean Absolute Error (MAE)

The MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It is calculated using Equation 3

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3}$$

Where $\mathbf{y_i}$ is the actual value, $\hat{\mathbf{y}}_\mathbf{i}$ is the predicted value, and $\mathbf{n}$ is the number of observations.

This metric represents the average absolute difference between the predicted and actual values in the same units as the target variable.

Root Mean Squared Error

The RMSE also measures the average magnitude of the predictions error, but it gives higher weight to larger errors as it can bee seen in its formula in Equation 4.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4}$$

Because it squares the errors before averaging, RMSE penalizes larger errors more severely than MAE. This makes it a valuable metric when it is important to avoid significant deviations from the actual values. In this context, it serves to assess how well the model performs in capturing rapid changes or peaks in the number of cases.

Coefficient of Determination R^2

The R^2 score quantifies how well the predicted values approximate the actual data. It is defined as in Equation 5.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{5}$$

Where $\bar{y}_i$ is the mean of the actual values. The R^2 value ranges from 0 to 1, where 1 indicates perfect prediction, and values closer to 0 indicate that the model fails to explain the variability in the data.

This metric was the reference metric to evaluate the models. It is specially meaningful in this context because the objective is not only to minimize prediction errors but to make sure that the model can capture the variability of the epidemic curves. A high R^2 indicates that the model can explain a large portion of the variance in the actual data, meaning it can reproduce the dynamics of the evolution of cases.

In addition, R^2 is especially useful because it always takes values between 0 and 1, and it is independent to the scale or units of the data. this makes it ideal for comparing the results between countries, even if the number of cases is very different. This also makes it ideal for comparing the performance of the different methods and architectures use during the development of this project.

5.2 Predicting cases

5.2.1 LSTM

As the data was split country-wise, two types of training were performed: a classical train-test split, where 80% of the countries were used as the training set and the remaining 20% as the test set, and a leave-one-out validation. This second approach intended to identify which types of case evolution curves posed particular challenges for the model to predict accurately and to highlight possible data quality issues.

The leave-one-out training strategy revealed a considerable variation in model performance across countries, indicating that some were consistently easier or harder to predict than others. Here, two countries are used as examples, Iceland and Latvia. Iceland was among the worst-performing countries, while Latvia was among the best. As shown in Figure 6, the evolution of the case curve in Iceland is not smooth, which suggests that the issue lies not in the model itself but inconsistent data reporting in that country. On the other hand, see Figure 7 for the results corresponding to Latvia. As the ground truth shows a more gradual and consistent progression for the number of cases, the model predictions are significantly more accurate.

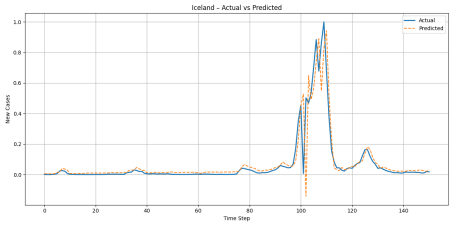

Fig. 6: Normalized results for Iceland using leave-one-out training strategy

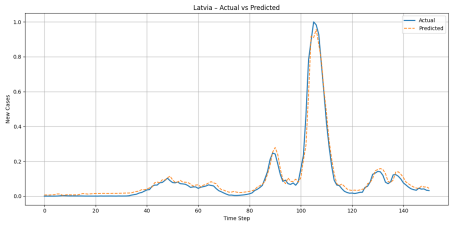

Fig. 7: Normalized results for Latvia using leave-one-out training strategy

For additional context on the variability in model performance, see the metrics for Ireland and Latvia in Table 1.

The average values of the metrics across the 44 countries in the dataset using an LSTM are shown in Table 2.

The results for the test set, representing 20% of the countries can be found in Table 3 in comparison with the RNN metrics that come next.

5.2.2 RNN

Even though the literature suggested that LSTMs outperform RNNs, the results of the experiments reveal that this problem could be similarly solved using an RNN, even showing slightly better metrics. This suggests that this problem is easy to predict and that the long time predictions are not as relevant as initially thought. For the same set of countries in the test set, see the metrics for the RNN in Table 3 in comparison with the LSTM metrics.

See also the visual representation of the prediction for Latvia in Figure 8.

5.3 Simulations

5.3.1 Impact of Policy Features on Model Predictions

This part of the project was to evaluate how a radical modification to the policy features would affect the model performance. The naive intuition is that if these changes have a meaningful effect on how the model predicts the evolution of cases, this would mean that just by tweaking these variables it could be seen how they affect the evolution of

Country	MAE	RMSE	R^2
Iceland	0.084	0.031	0.768
Latvia	0.030	0.019	0.973

TABLE 1: Results for Iceland and Latvia using the leave-one-out strategy.

Metric	Value
MAE	0.025
RMSE	0.047
R^2	0.927

TABLE 2: Average results of leave one out training for each country

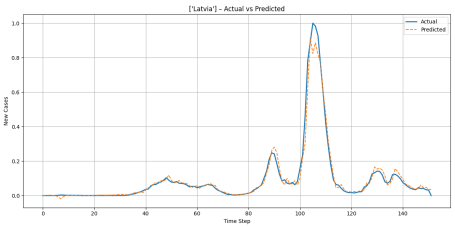

Fig. 8: Normalized results for Latvia, using a RNN

the pandemic. In order to see if this works, the same model used for predicting cases was used for predicting a single country for which all the policy variables were set to 0. To compare how the model predictions differ, see in Figure 7 how the model predicts the cases for Latvia in comparison with Figure 9. It can be observed that there isn't a big difference, just in the peak of cases between the time steps 100 and 120, now the prediction undershoots the number of cases, but apart from that, no significant changes can be perceived. This suggests that simply modifying the input data without changing the training process is insufficient to evaluate the effect of policies in the final predictions. It also reveals a problem in doing simulations based on the assumption that the variables of the policy indicators play a significant role in the final prediction.

For additional context, the original values for the policy features from Latvia can be seen in Figure 10.

5.3.2 Distinguishing Policy and Case Data in Autoregressive Training

The results shown in this section aim to evaluate if this architecture can yield correct forecasts when run on real, unaltered data, and whether it is suitable for running simulation experiments by manually modifying the input variables for the policy features. The idea was to test the model's ability to correctly predict the number of new cases using ground truth data and then evaluate whether modifying the contextual features influences these predictions.

During training, the results were encouraging. After extensive hyperparameter tuning, it was possible to train a model whose validation loss remained relatively stable, even as the level of teacher forcing was reduced and the degree of autoregression increased. However, it is impor-

Model	MAE	RMSE	R^2
LSTM	0.019	0.039	0.947
RNN	0.019	0.039	0.949

TABLE 3: Test set results for predicting COVID-19 cases using LSTM and RNN models.

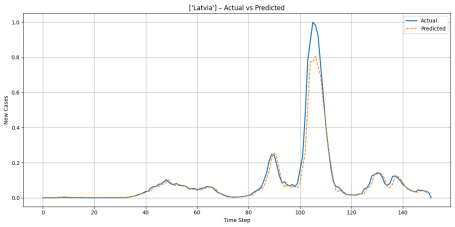

Fig. 9: Normalized results for Latvia, the variables on policies were manually set to zero.

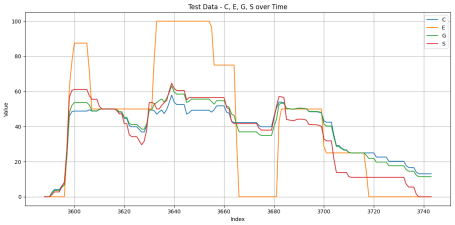

Fig. 10: Line plot for the 4 variables of policies from Latvia

tant to note that even the best performing solution exhibits a considerable amount of noise. For context, refer to Figure 11 and for the same experiment see Figure 12 for the evolution of training versus test loss as the teacher forcing probability decreases for every epoch.

Also, the metrics on the test set were good, as shown in Table 4.

The intuition is that this model is robust enough to perform simulations with full teacher forcing. But as it can be seen in Figure 11, this model is not able to run fully autoregressive simulations, the results fail to produce meaningful predictions. The model was trained using an exponential decay in the teacher forcing ratio, moving from full teacher forcing to fully autoregressive training as noted in Figure 12. It performs well when evaluated in the standard way, using teacher forcing to deliver predictions on the test set, but struggles during simulation when running fully autoregressive simulations. The main issue is error accumulation: because the model no longer sees the ground truth at each step, small prediction errors quickly build up. As the simulation continues, it keeps feeding its own (wrong) previous outputs as inputs, which makes the predictions move further away from the actual values, resulting in very poor performance.

Even though the model achieved acceptable performance when using teacher forcing, it failed to produce reliable predictions under fully autoregressive conditions. This makes it unfit to fulfil the objective of using this model for simulations since it requires the ability to run on its own outputs.

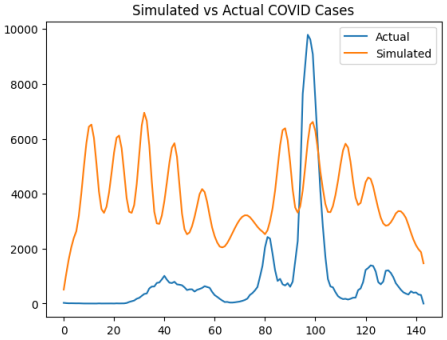

Fig. 11: Predicted versus simulated results for Latvia

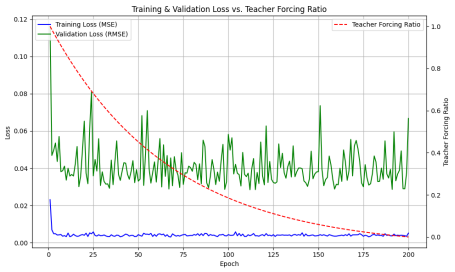

Fig. 12: Training vs validation loss, LSTM using exponential decay on the level of teacher forcing used in each epoch. Red line represents the evolution of the teacher forcing probability, the green line the validation loss (RMSE) and the blue line the training loss.

Metric	Value
MAE	0.060
RMSE	0.067
R^2	0.865

TABLE 4: Results with Latvia as the test set, for exponential decay on the teacher forcing

5.3.3 LSTM sequence to sequence

The results presented here are the best obtained with this method. Different sets of features have been used to improve model performance. Starting from the minimal amount of features, which are the four indexes that consist of an aggregation of indicators, to adding information about the level of vaccinations and the sixteen individual indicators. As the results in Section 5.4 suggest, the model works better with the disaggregated indicators, the results presented in this section are obtained using them in their disaggregated form.

This model was not easy to train out of the box, it required quite a significant amount of time in hyperparameter optimization. In the end, the model which trained the best was a very small one, otherwise it quickly overfitted. The architecture of the best model consisted in a two layer encoder-decoder LSTM with a hidden dimension of just 3 cells.

The resulting metrics were not satisfactory, as can be seen in Table 5.

Let's see a couple of figures to illustrate why this model is not fit to make reliable predictions. Figure 13, shows the predictions of the first 200 time steps present in the test set.

Metric	Value
MAE	0.069
RMSE	0.014
R^2	0.597

TABLE 5: Results with Latvia as the test set, for exponential decay on the teacher forcing

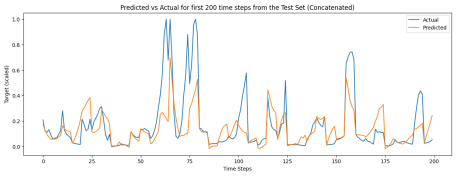

Fig. 13: Normalized results for Spain, training with only the four policy indicators as features.

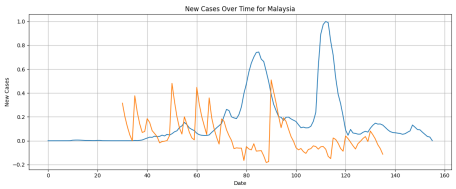

Fig. 14: Normalized results for Spain, training with only the four policy indicators as features.

This sequence to sequence model is unable to model the major peaks in case numbers. But the results are worse when predictions are made iteratively for a test country across all time steps to get a full prediction of all time steps. The resulting prediction performs poorly, as shown in Figure 14.

5.4 Training only with policy data

This is an extra experiment not mentioned in the methodology section (Section 4). As it was important to check the feature importance in a more explicit way, this experiment serves to clarify that point and as a possible justification to the results obtained for the previous methods.

To evaluate whether the LSTM was able to model the evolution of policies to the number of cases, it was trained only using policy data. Only using as features the four variables, which are an aggregation of policy indicators, and as a target the amount of cases. See Figure 15 for the result.

Here several experiments were performed in an attempt to obtain a model with acceptable performance. The feature space was expanded by adding additional variables. As explained in Section 3, the four main policy indicators are an aggregation of different indices. The first thing tried here was to train a model on the sixteen disaggregated features, the result was slightly better, but still not good enough.

The best result, but once again not meaningful was by combining the disaggregated policies with features about the level of vaccination in each country. As shown in Figure 16, the prediction is insufficient.

A similar experiment was conducted but now with only one feature, the new reported cases per week, without any

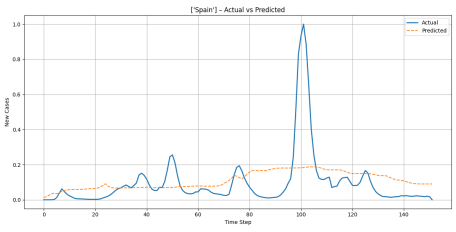

Fig. 15: Normalized results for Spain, training with only the four policy indicators as features.

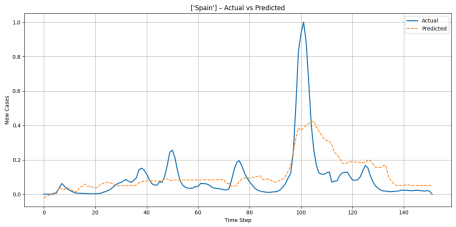

Fig. 16: Normalized results for Spain, training with the sixteen indices and vaccination variables.

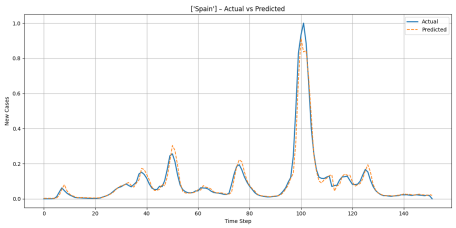

Fig. 17: Normalized results for Spain, training only with the reported new cases per week as a feature.

other feature, as the previous results had suggested, the model is able to get very good metrics on the predictions with just one feature. See Figure 17 with these results and compare it to Figure 16.

What this suggests is that an LSTM is not able to capture the effect of public policy changes on the number of cases. Another possibility is that there may be no meaningful relationship between the two variables, did not significantly influence case trends.

5.5 Results discussion

As suggested by the literature, the task of predicting new COVID-19 casesa relatively straightforward task, both RNNs and LSTM worked as expected once some problems with the input data were solved. Good predictions can be obtained using only the past number of cases for all the different countries even when case curves vary significantly between countries. The fact that both LSTMs and RNNs worked similarly when solving this problem suggests that it doesn't require information from many past steps in order to give a good prediction. This could be tested even further if the data is enriched with daily data instead of weekly data, as was done in this project. It would allow extending the input sequence to include more past time steps. In that case, it would be possible to assess whether RNNs encounter the vanishing or exploding gradient problems in this specific prediction task, as reported in other contexts. The shift made in this project, to go from having a data point for every day to one for every week also simplified the learning problem for the model and made the study of the differences in the performance of LSTM versus RNNs less interesting.

This project encountered a big problem to fulfil one of its objectives, predicting the evolution of the number of cases depending on which policies were to be implemented. All the different methods described in Section 4.2 failed to achieve satisfactory results. The different approaches tried to make the model more dependent on the contextual features but it was not enough. The problem was that these features were not relevant to the model when predicting the

number of cases, it only needed the past number of reported cases to give a good prediction. The results showed in Section 5.4 reveal this in a very clear way, the variables which are not about the past number of cases were practically irrelevant. This fact is significant when generating predictions without relying on ground truth data, which is the problem encountered next.

As the goal was always to produce a practically useful model, one of the objectives was to develop a model robust enough to be used by feeding it its own predictions. The aim was to have .a pipeline where data gathered during a pandemic can be fed to this trained model, and this model generates predictions based on it. Ideally, different kinds of policies could be tested in order to see how the model predicts that it would affect the number of new cases or any other variable that the model was trained for, like the number of hospital beds needed. This idea was quickly ruled out because of what is discussed in the previous paragraph. As modelling the impact of applying different policies turned out to be too ambitious for the scope of this project, focusing instead on predicting the number of new cases based on the model's own past predictions seemed a more feasible objective.

The problem with the different approaches to get a model that could work in an autoregressive way was that the predicted variable was the only important variable for the model. This caused the accumulation of error in the predictions made the output of the model to shift further and further from the ground truth. If the contextual features were more important for the model, this accumulation of error would not be so dramatic as the whole context could help to getting greater precision, even if the predictions were not perfect. The view is that the problem with the models trained with auto regression, although they showed acceptable metrics during training time, they couldn't handle the accumulation of error, as the contextual features didn't help the model to go in the right direction. In a hypothetical model in which these contextual features played an important role, the accumulation of the error in the predicted variable would not be so dramatic. This is why some efforts to get a model which relied solely on the contextual features were made. But as it can be seen in the results, none of these approaches was good enough to explore this idea any further.

6 Conclusion

Since it ultimately turned out to be a more challenging problem to solve than it initially appeared, some methods were applied that were not part of the original plan. This led to the implementation of many different approaches in order to confirm or refute the initial hypothesis, an exhaustive number of experiments was performed but none proved to be really successful, so for the scope of this project, the hypothesis could not be validated, as no model produced usable results. The efforts to make the contextual features more relevant in the final prediction or simplifying the problem, were not enough to get a good final result. The conclusion is that LSTMs are not able to model predictions that can use contextual features to forecast the evolution of the epidemic.

The full source code is available at the GitHub repository: https://github.com/gSamper/covid_prediction.

7 Future lines of investigation

Future research could be built on this work in different directions. Integrate a hybrid model that combines deep learning methods, like the ones explored in this report, with epidemiological approaches such as SIR models. Also, more advanced architectures like Transformers or LSTM networks enhanced with attention mechanisms could be explored. Finally, some form of multitask learning could be applied to predict related outcomes like new cases, hospitalizations and deaths at the same time.

REFERENCES

[1] I. Cooper, A. Mondal, and C. G. Antonopoulos, "A sir model assumption for the spread of covid-19 in different communities," *Chaos, Solitons Fractals*, vol. 139, p. 110057, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960077920304549

[2] H. AlQadi and M. Bani-Yaghoub, "Incorporating global dynamics to improve the accuracy of disease models: Example of a covid-19 sir model," *PLOS ONE*, vol. 17, no. 4, p. e0265815, 2022. [Online]. Available: https://doi.org/10.1371/journal.pone.0265815

[3] S. Shastri, K. Singh, S. Kumar, P. Kour, and V. Mansotra, "Time series forecasting of covid-19 using deep learning models: India-usa comparative case study," *Chaos, Solitons Fractals*, vol. 140, p. 110227, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960077920306238

[4] M. O. Alassafi, M. Jarrah, and R. Alotaibi, "Time series predicting of covid-19 based on deep learning," *Neurocomputing*, vol. 468, pp. 335–344, 2022. [Online]. Available: https://doi.org/10.1016/j.neucom.2021.10.035

[5] R. M. Munshi, M. M. Khayyat, S. Ben Slama, and M. M. Khayyat, "A deep learning-based approach for predicting covid-19 diagnosis," *Heliyon*, vol. 10, no. 7, p. e28031, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405844024040623

[6] T. Hale, N. Angrist, R. Goldszmidt, B. Kira, A. Petherick, T. Phillips, S. Webster, E. Cameron-Blake, L. Hallas, S. Majumdar, and H. Tatlow, "A global panel database of pandemic policies (oxford covid-19 government response tracker)," *Nature Human Behaviour*, 2021. [Online]. Available: https://doi.org/10.1038/s41562-021-01079-8

[7] T. e. a. Hale, "Oxford covid-19 government response tracker github repository," https://github.com/OxCGRT/covid-policy-tracker, 2025, accessed: 2025-05-12.

[8] I. D. Mienye, T. G. Swart, and G. Obaido, "Recurrent neural networks: A comprehensive review of architectures, variants, and applications," *Information*, vol. 15, no. 9, 2024. [Online]. Available: https://www.mdpi.com/2078-2489/15/9/517

[9] R. M. Schmidt, "Recurrent neural networks (rnns): A gentle introduction and overview," 2019. [Online]. Available: https://arxiv.org/abs/1912.05911

[10] C. B. Vennerød, A. Kjærran, and E. S. Bugge, "Long short-term memory rnn," 2021. [Online]. Available: https://arxiv.org/abs/2105.06756

[11] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural Computation*, vol. 31, pp. 1235–1270, 07 2019.