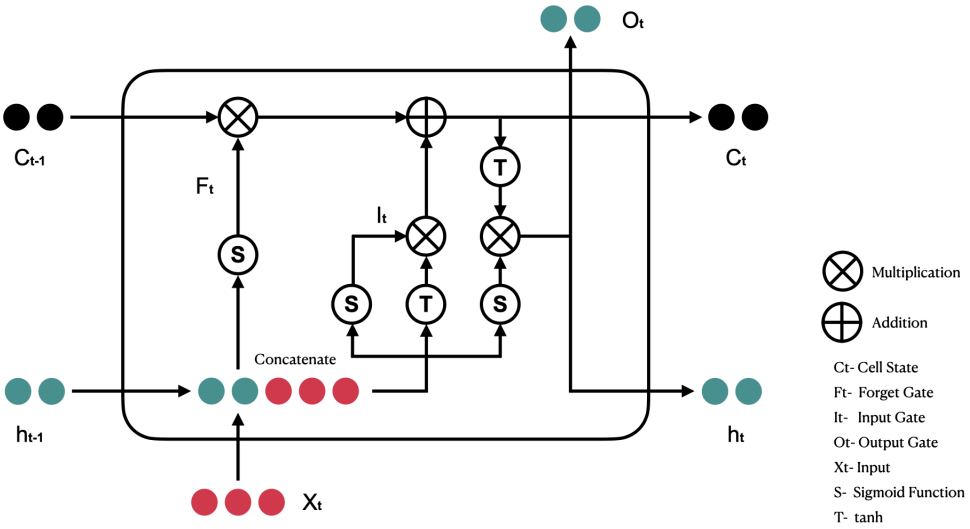
Supplimentary material

Structure of neural network. We detail the structure of LSTM regression network and the cell gate below.

Supplementary Figure 1. This diagram illustrates the architecture of a simple LSTM network for regression. The network starts with a sequence input layer followed by an LSTM layer. The network ends with a fully connected layer and a regression output layer.



Supplementary Figure 2. Structure of a LSTM cell that describes the cell state c_t (green circles, dim =2) and hidden state h_t (red circles, dim =2) and various gates[1].

Training

In this section, we provide details on how we fit the models discussed in Section Methods, and introduce the experiment setups for performance evaluation.

Data Preprocessing and Preparation

Smoothing. Recall that we denote Y_t as the number of COVID-19 cases at time t. We adopt the standard smoothing techniques below to smooth out the irregular roughness between time steps:

$$\bar{Y}_t := \frac{1}{\tau}\left(\sum_{i=t-\tau+1}^{t} Y_i\right),$$

where τ is the smoothing lag number. In our study, we choose $\tau = 7$ by exploiting the day of week effect: the number of observations each date is influenced by which day of a week it is. For example, the observations of COVID-19 case detection are empirically higher on Fridays, which might be attributed to the fact that people have time to do the testing on Fridays. The day of week effect has long been observed in healthcare and the stock market. To avoid the day of week effect, we smoothed the data by using the average of observations from 6 previous dates and the current date. We will use the smoothed data as the ground truth throughout the study.

Experiment setup for single trial

In reality, we often aim at making reliable and timely predictions based on an appropriate length of data. Furthermore, it is well-known that the different variants of COVID-19 could result in different transmission mechanisms, leading the joint distribution of confirmed cases to shift from time to time. Therefore, it is not advised to train the model using all historical data since pandemic. Therefore, we cut the time series into different pieces and split each piece into training set and test set. We then apply the model training and testing on each piece, which we call a trial. For each trial, we do a three-step data preprocessing:

Differencing: Statistical methods such as AR have guaranteed performance on stationary time series. However, the raw data are typically not stationary. Differencing can help us stabilize the mean of a time series by removing changes in the level of a time series, and therefore eliminating (or reducing) trend and seasonality. This can make the differenced time series stationary. Specifically, we keep differencing data until stationarity is achieved by standard tests such as Augmented Dickey Fuller Unit Root Test (ADF)[2] or Kwiatkowski Phillips Schmidt Shin Test (KPSS)[3]. We found a consistent stationarity among different trials with only one differencing operation.

Rescale. After differencing the data, we rescale the training data into the $[-1,1]$ with:

$$Y_{\text{train,scaled}} = \frac{Y_{\text{train}} - \mu_{\text{train}}}{Y_{\text{max,train}} - Y_{\text{min,train}}}$$

where μ_{train} denotes the mean of Y_{train}, $Y_{\text{max,train}}$ denotes the maximum of Y_{train}, $Y_{\text{min,train}}$ denotes the minimum of Y_{train}, and $Y_{\text{train,scaled}}$ denotes training data after scaling. We also apply the same re-scaling map to the testing data:

$$Y_{\text{test,scaled}} = \frac{Y_{\text{test}} - \mu_{\text{train}}}{Y_{\text{max,train}} - Y_{\text{min,train}}}$$

Thus the testing data does not affect the selection of our scalar. We train the model on $Y_{\text{train,scaled}}$ and make predictions with $Y_{\text{test,scaled}}$. Finally, we apply the inverse rescaling map to retrieve the original scale, and thus make comparison with the original data.

Reshaping. After differencing and rescaling the data, we transform the data into supervised learning form as shown below in the table 1.

	Input	Output
Row 1	$[Y_{t-n}, \cdots, Y_{t-n+1}]$	Y_{t-n+p}
Row 2	$[Y_{t-n+1}, \cdots, Y_{t-n+2}]$	$Y_{t-n+p+1}$
Row 3	$[Y_{t-n+2}, \cdots, Y_{t-n+3}]$	$Y_{t-n+p+2}$
$\vdots$	$\vdots$	$\vdots$

Supplementary Table 1. The input-output data format. Here p is the lag number.

This step is also conducted on the testing data for all three models and the training data for the LSTM model and the Hybrid model.

We are now ready to conduct the experiment on a single trial. To begin with, each trial has 88 continuous observations. We apply a first order differencing to make the trial data stationary, at the cost of losing one observation. Now we have 87 observations. The first 62 would be used as the training data: notice that our training size is 63, since the 62 differenced values are derived from 63 observations. The remaining 25 values will be used to make a testing data matrix.

We apply the rescaling on the training and test data. Then we reshape the testing data of length 25 to a matrix of size $(18,8)$. For each of the 18 rows, the first 7 values are model inputs, which would return a single predicted value to us: an prediction for the rescaled $Y_t^{(1)}$, where the superscript 1 refers to the first order differencing. Let us denote this prediction with:

$$\widehat{Y}^{(1)}_{t,\text{scaled}} \tag{1}$$

We derive an prediction of the ground truth Y_t from (1). First, we scale (1) back to the original scale by applying the inverse function of rescaling to it. Then (1) becomes an prediction of $Y_t^{(1)}$, say $\hat{Y}_t^{(1)}$. We now retrieve an estimation of the ground truth Y_t with:

$$\hat{Y}_t = \widehat{Y}_t^{(1)} + Y_{t-1} \tag{2}$$

Notice that all observations before Y_t are known. Since we obtain one estimation from each row, we will end up with a list of 18 estimated values. We assess the model by comparing these 18 estimations with the 18 corresponding ground truth values.

Choice of Hyper-parameters

In this section, we detail how we choose the hyper-parameters for each predictive model.

AR: We used lag number 7 for the sake of interpretability, since 7 is the number of days in a week. There could exist more scientific methods to select the lag number. For example, we may check the Bayesian information criterion (BIC). BIC is a class of information criteria to measure the goodness of fit of a statistical model. It builds on the concept of entropy and can weigh the complexity of the estimated model against the goodness of fit of this model to the data. This information helps to assess the model's parameters and how well the model performed.

LSTM: We use batch number 1, epoch number 100, and units 1. The neural network is trained on mean square error, with optimizer adam. The fully connected layer is activated by the default linear function. Since the performance is well enough for our purpose, we do not tune the model further.

Hybrid Model: This neural network is the addition of 2 layers: an AR layer and a LSTM layer, after being weighted by a trainable coefficient between 0 and 1. We use the same set of hyperparameters as we do in the AR model and the LSTM model. As a result, we have some flexibility to tune to make the performance better.

References

1. Karim, R. Animated rnn, lstm and gru (2020).
2. Mushtaq, R. Testing time series data for stationarity. *SSRN* (2011).
3. Shin, Y. & Schmidt, P. The kpss stationarity test as a unit root test. *Econ. Lett.* **38**, 387–392, DOI: https://doi.org/10.1016/0165-1765(92)90023-R (1992).