







**Reservoir computing on epidemic spreading: A case study on COVID-19 cases**Subrata Ghosh <sup>1</sup>, Abhishek Senapati <sup>2,3</sup>, Arindam Mishra <sup>4</sup>, Joydev Chattopadhyay,<sup>2</sup> Syamal K. Dana <sup>4</sup>,  
Chittaranjan Hens <sup>1,\*</sup> and Dibakar Ghosh <sup>1</sup><sup>1</sup>*Physics and Applied Mathematics Unit, Indian Statistical Institute, 203 B. T. Road, Kolkata 700108, India*<sup>2</sup>*Agricultural and Ecological Research Unit, Indian Statistical Institute, 203 B. T. Road, Kolkata 700108, India*<sup>3</sup>*Center for Advanced Systems Understanding (CASUS), Goerlitz, Germany*<sup>4</sup>*Department of Mathematics, Jadavpur University, Kolkata 700032, India*

(Received 14 December 2020; accepted 23 June 2021; published 16 July 2021)

A reservoir computing based echo state network (ESN) is used here for the purpose of predicting the spread of a disease. The current infection trends of a disease in some targeted locations are efficiently captured by the ESN when it is fed with the infection data for other locations. The performance of the ESN is first tested with synthetic data generated by numerical simulations of independent uncoupled patches, each governed by the classical susceptible-infected-recovery model for a choice of distributed infection parameters. From a large pool of synthetic data, the ESN predicts the current trend of infection in 5% patches by exploiting the uncorrelated infection trend of 95% patches. The prediction remains consistent for most of the patches for approximately 4 to 5 weeks. The machine's performance is further tested with real data on the current COVID-19 pandemic collected for different countries. We show that our proposed scheme is able to predict the trend of the disease for up to 3 weeks for some targeted locations. An important point is that no detailed information on the epidemiological rate parameters is needed; the success of the machine rather depends on the history of the disease progress represented by the time-evolving data sets of a large number of locations. Finally, we apply a modified version of our proposed scheme for the purpose of future forecasting.

DOI: [10.1103/PhysRevE.104.014308](https://doi.org/10.1103/PhysRevE.104.014308)**I. INTRODUCTION**

The impact of the unprecedented pandemic COVID-19 is widespread, practically collapsing all human activities around the world. A severe crisis has arisen in the public health systems and economy everywhere. Under this extreme condition, various agencies, government and non-government, are looking for ways and means to stop the spread of the virus and to develop a health support system appropriate for mitigating this disaster. Predicting the number of infected cases is challenging, although it is the most important task for understanding the gravity of spread and to keep preparing the public health system for innumerable large demands [1–4].

An accurate prediction methodology may enable policy-makers to deter the spread of the pandemic by designing and implementing effective disease control strategies [5–13]. A wide range of models are being developed by this time, borrowing ideas from statistical physics and epidemiology, to understand the trend of disease progression for the purpose of prediction. Data-driven techniques such as machine learning and artificial-intelligence tools are applied to forecast the future trend of COVID-19-infected cases [14,15]. For instance, an exponential smoothing model can forecast [3] confirmed COVID-19-infected cases. The recurrent neural network approach has been used [16] to predict the early trend of COVID-19 in China by training the machine from

SARS data of the year 2003. Recently, Li *et al.* [17] considered the spatiotemporal information on infection where susceptible-infected-recovered (SIR) dynamics (constructing differential equations) is adjusted with a recurrent neural network to forecast the temporal data with limited resources. Many other approaches such as deep learning using a long-short-term-memory network [18,19], support vector machine [20,21], hybrid autoregressive moving average model [20,22], neural network [23], supervised XGBoost classifier [19], or random forest algorithm [24] have been utilized to predict the infection trend as well as the mortality and severity of patient conditions. However, these prediction-based techniques heavily depend on several structural parameters as well as intrinsic components of the machine itself. The successful forecasting by machine learning is also deterred by the limited availability of temporal data. The key question we raise here is whether there is any possibility of predicting the infection trend of a disease, in general, in targeted locations by feeding infection data on the disease available from other locations in different countries? We accept the constraint that detailed information on the basic reproduction number and the force of infection of the locations may not be available.

We attempt to address this issue in a simple way using reservoir computing, i.e., the echo state network (ESN). The ESN is a modified version of the recurrent neural network that easily avoids the training-related challenges and tunes the output layer only to mimic the target data at the time of a training procedure. The ESN has been used extensively to predict complex signals ranging from chaotic time series

\*chittaranjanhens@gmail.com

to stock-price data [25–32], and currently, it has been shown that it can easily capture the critical onset of generalized synchronization [33–36] and detect collective bursting in neuron populations [37]. Therefore, the ESN showed encouraging records of handling multiple inputs of temporal data and the ability to trace the correlation between them [34,37]. Motivated by this fact, we utilize the strength of the ESN to develop a strategy for predicting the spread of any infectious disease from the available collection of a multitude of infection data on the same disease.

First, we check the efficiency of the ESN for a large collection of synthetic epidemic data generated from the classical SIR model. Finally, the prediction capability of the ESN is carefully investigated with available incidence data on COVID-19 from a large number of locations around the world, with the aim of identifying the real outbreak scenario in other targeted locations. The machine works successfully to predict the spread of the disease to the extent of 2 weeks and little more. The ESN is thus shown to be an effective tool for data-driven future prediction of any infectious disease, in general. Note that a future prediction from the previous data (in each location) is not the sole objective of this work. A nonmonotonic trend of a real data set always resists the forecasting of the future trend. Being aware of this drawback, we adopt an alternative formalism: whether a machine (here the ESN) can capture the trend of infection of target locations by utilizing the infection trend of other locations at the same time. As a result, this alternative formalism (with some adjustment) can truly forecast the future trend of infection.

## II. DESCRIPTION OF THE ECHO STATE NETWORK

In this study, a standard leaky tanh network is considered as the ESN. The dynamics of each reservoir node is governed by the following recursive relation [25]:

$$\mathbf{r}(t+1) = (1 - \alpha)\mathbf{r}(t) + \alpha \tanh(\mathbf{W}_{\text{res}}\mathbf{r}(t) + \mathbf{W}_{\text{in}}\mathbf{s}(t)). \quad (1)$$

Here  $\mathbf{r}(t)$  is the  $N_{\text{res}}$ -dimensional vector denoting the state of the reservoir nodes at time instant  $t$  and  $\mathbf{s}(t)$  is the  $M$ -dimensional input vector. The matrices  $\mathbf{W}_{\text{res}}$  (dimension:  $N_{\text{res}} \times N_{\text{res}}$ ) and  $\mathbf{W}_{\text{in}}$  (dimension:  $N_{\text{res}} \times M$ ) represent the weights of the internal connection of reservoir nodes and weights of the input, respectively. The parameter  $\alpha$  is the leakage constant, which can take values between 0 and 1. It is to be noted that the tanh function is operated elementwise. We take  $\alpha = 0.5$  and  $N_{\text{res}} = 1000$  throughout our simulations. The reservoir weight matrix  $\mathbf{W}_{\text{res}}$  is constructed by drawing random numbers uniformly over the interval  $(-1, 1)$  and the spectral radius of the matrix  $\mathbf{W}_{\text{res}}$  is rescaled to less than unity. Matrix  $\mathbf{W}_{\text{in}}$  containing input weights is also generated by randomly chosen elements from the interval  $(-1, 1)$ . Note that a constant bias,  $b = 1$  can be added in the input vector  $\mathbf{s}(t)$ .

Next, we consider time-series data on  $N$  patches, among which the data on  $M$  patches are fed into the machine, and the remaining  $N - M$  patches, whose time signals are to be predicted by the ESN, are targeted. A fraction of the data points (when  $t = 0, 1, \dots, t_r$ ) from each of the infected signals is used for training purposes [see the upper left, light-red box in Fig. 1]. At first, the target is to identify the infection of the rest of the patches ( $N - M$ ) by the ESN during the training or

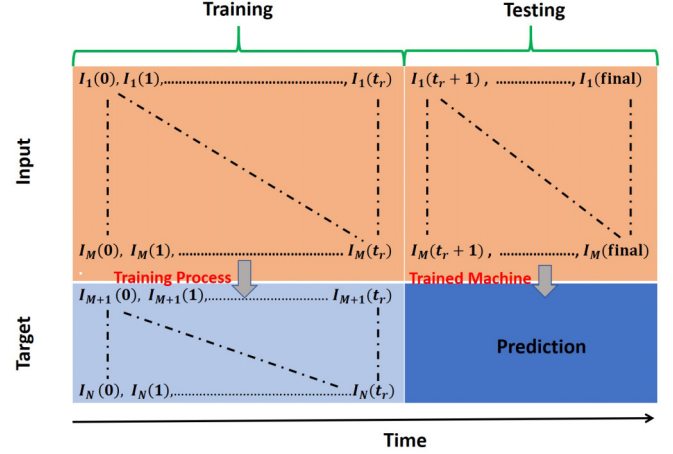


FIG. 1. Training and testing scheme using the echo state network. Upper panel (light-red boxes): Input parts. Lower left panel (light blue box): Data set of the output data. Lower right panel (dark-blue box): Predicted-testing data.

learning process (lower left, light-blue box in Fig. 1). Once the machine is trained, input from  $M$  patches with the rest of the data points  $(t_r + 1, \dots, t_{\text{final}})$  is fed into the machine (upper right, light-red box in Fig. 1) to predict the infection in the  $N - M$  patches (lower right, dark-blue box in Fig. 1).

At each time  $t$ , the input vector  $\mathbf{s}(t)$  will have  $M$  number of elements:  $[\mathcal{I}_1(t), \mathcal{I}_2(t), \dots, \mathcal{I}_M(t)]^T$ . At time  $t$ , the contribution of the input weight matrix in the dynamics of the reservoir [see Eq. (1)] can be written as follows:

$$\begin{bmatrix} \mathbf{W}_{\text{in}}(1, 1) & \cdots & \mathbf{W}_{\text{in}}(1, M) \\ \mathbf{W}_{\text{in}}(2, 1) & \cdots & \mathbf{W}_{\text{in}}(2, M) \\ \vdots & \vdots & \vdots \\ \mathbf{W}_{\text{in}}(N_{\text{res}}, 1) & \cdots & \mathbf{W}_{\text{in}}(N_{\text{res}}, M) \end{bmatrix} \times \begin{bmatrix} \mathcal{I}_1(t) \\ \mathcal{I}_2(t) \\ \vdots \\ \mathcal{I}_M(t) \end{bmatrix}.$$

In the training process, at each time instant  $t$ , the reservoir state  $\mathbf{r}(t)$  and input  $\mathbf{s}(t)$  are accumulated in  $\mathbf{X}(t) = [\mathbf{s}(t); \mathbf{r}(t)]$ . The output relation can be written in vector form as

$$\mathbf{Y} = \mathbf{W}_{\text{out}}\mathbf{X}. \quad (2)$$

Here,  $\mathbf{Y}$  is a matrix of dimension  $(N - M) \times K$ , where  $K$  is the length of the time signal. Matrix  $\mathbf{X}$  having dimension  $(N_{\text{res}} + M) \times K$  looks like

$$\begin{bmatrix} \mathcal{I}_1(t_1) & \mathcal{I}_1(t_2) & \cdots & \mathcal{I}_1(t_K) \\ \mathcal{I}_2(t_1) & \mathcal{I}_2(t_2) & \cdots & \mathcal{I}_2(t_K) \\ \vdots & \vdots & \vdots & \vdots \\ \mathcal{I}_M(t_1) & \mathcal{I}_M(t_2) & \cdots & \mathcal{I}_M(t_K) \\ r(1, 1) & r(1, 2) & \cdots & r(1, K) \\ r(2, 1) & r(2, 2) & \cdots & r(2, K) \\ \vdots & \vdots & \vdots & \vdots \\ r(N_{\text{res}}, 1) & r(N_{\text{res}}, 2) & \cdots & r(N_{\text{res}}, K) \end{bmatrix}.$$

Matrix  $\mathbf{W}_{\text{out}}$  can be determined by the Ridge regression method as

$$\mathbf{W}_{\text{out}} = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I})^{-1}, \quad (3)$$

where  $\lambda$  is the regularization factor that avoids overfitting.  $\mathbf{Y}$  is the time-series data for the targeted patches and  $\mathbf{I}$  is the

identity matrix of dimension  $(N_{\text{res}} + M) \times (N_{\text{res}} + M)$ . Note that when  $\lambda = 0$ , Eq. (3) reduces to the least-squares method.

We consider  $N$  patches, in which data from  $M$  number of patches are fed into the machine for training purposes. At time  $t$ , the dimension of the output vector of the targeted patches will be  $(N - M) \times 1$ . Thus the output matrix ( $t \in [t_{r+1}, t_f]$ ) can be written as (Fig. 1)

$$\mathbf{y}(t) = \begin{bmatrix} \mathcal{I}_{M+1}(t) \\ \mathcal{I}_{M+2}(t) \\ \vdots \\ \mathcal{I}_N(t) \end{bmatrix}.$$

### III. PREDICTION FOR SYNTHETIC DATA

The classical SIR model is used to numerically generate a large set of independent synthetic time-series data (say  $i = 1, 2, \dots, N$ ) on infection for different sets of disease transmission rates and the initial fraction of the infected population. The disease spreads into the patches or locations, where the SIR dynamics of the  $j$ th isolated location is captured by a set of three coupled equations:

$$\dot{\mathcal{S}}_j(t) = -\beta_j \mathcal{S}_j(t) \mathcal{I}_j(t), \quad (4)$$

$$\dot{\mathcal{I}}_j(t) = \beta_j \mathcal{S}_j(t) \mathcal{I}_j(t) - \gamma_j \mathcal{I}_j(t), \quad (5)$$

$$\dot{\mathcal{R}}_j(t) = \gamma_j \mathcal{I}_j(t). \quad (6)$$

Based on the health conditions, the population of the  $j$ th location is categorized into three compartments: susceptible ( $\mathcal{S}_j$ ), infected ( $\mathcal{I}_j$ ), and recovered ( $\mathcal{R}_j$ ). The parameters  $\beta_j$  and  $\gamma_j$  denote the rate of disease transmission and recovery rate, respectively. We fix the recovery rate at  $\gamma_1 = \gamma_2 = \dots = \gamma_N = 1/14 \text{ day}^{-1}$  for this study. We generate a set of  $N$  independent synthetic data series by random choices of  $\beta_j$  from the uniform distribution  $\mathcal{U}(0, 0.25)$ . The initial infections  $[\mathcal{I}_j(0)]$  are also taken from  $\mathcal{U}(10^{-7}, 10^{-4})$  and  $\mathcal{R}_j(0) = 0$ , and  $\mathcal{S}_j(0) = 1 - \mathcal{I}_j(0) - \mathcal{R}_j(0)$ . The choice of  $\beta_j$  is based on available data and country-level estimation of the basic reproduction number for COVID-19 [38], which varies from 0 to 3.5. Model (4)–(6) is integrated for a time interval  $[0, 300]$  with a time step 0.01 using the RK4 routine. Therefore, each synthetic data set contains 30 000 data points (300 days). Since a variation in the disease transmission rate ( $\beta_j$ ) and an initial fraction of the infected population  $[\mathcal{I}_j(0)]$  lead to diversity in peak sizes as well as the time duration for reaching the peak of infection, we treat the independent synthetic data sets as collected infection data for different regions or countries where an outbreak of the same disease takes place.

To explain our scheme more clearly, we have drawn randomly selected infected signals ( $\mathcal{I}_j$ ) in Fig. 2(a) (red lines). Due to the distribution of the disease transmission rate and initial state (initial fraction of the infected population), the time to reach a peak of infection varies from one isolated patch to other patches as shown in a number of the time domain plots of  $\mathcal{I}$  (red lines). For comparison, we have drawn vertical lines at a fixed time  $t = t_r$  in each of the red signals [Fig. 2(a)]. The topmost signal is infected earlier and reaches the zero state before time  $t_r$ . The second signal from the top reaches its peak at  $t = t_r$ . The third one reaches the infection peak

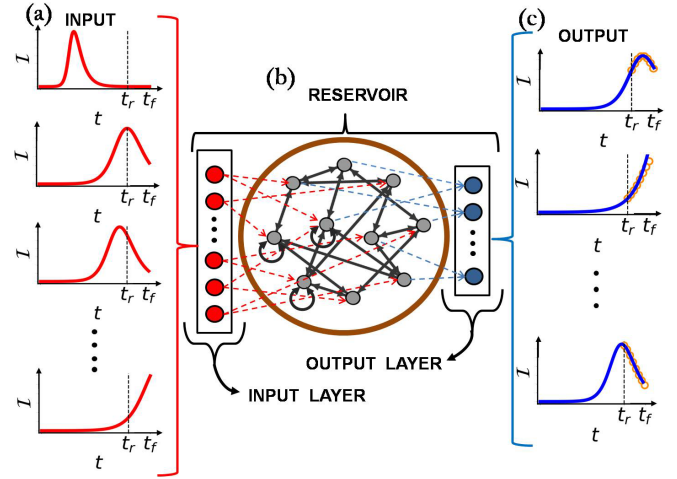


FIG. 2. Schematic of the ESN and signal variability. (a) Infection data inputs (red signal) fed into the machine. The dashed vertical line (at  $t = t_r$ ) signifies a time limit of data point inputs to the machine. (b) ESN structure: input layer, reservoir, and output layer. The weights of the input and the reservoir once selected are kept fixed throughout the training and testing procedure. (c) Data output of targeted locations or patches (blue signals). Left parts of the dashed vertical lines ( $t \leq t_r$ ) are closely mapped with the machine-generated signals at the time of training. Right parts of the vertical lines are predicted data (red circles) from the machine at the time of testing the ESN.

earlier than  $t = t_r$ . The bottom signal is gradually increasing and yet to reach the peak at time  $t_r$ . The sequence of data until  $t = t_r$  for each red signal is fed into the ESN [Fig. 2(b)] for training purposes. Note that the ESN has three components: (a) an input layer, which captures the input data; (b) a reservoir network that associates the input data with its nodes generally in a nonlinear way; and (c) an output layer, which generates the desired or targeted data. In our proposed scheme, the ESN output layer is controlled in such a way that it closely maps the output signal (blue lines) up to the time  $t = t_r$  [Fig. 2(c)]. Noticeably, these segments of the output signals (left of the dashed vertical lines; blue curves) are not similar to each other: the upper one does not reach the peak value, whereas the lowermost signal just crosses the peak before  $t = t_r$ . Once the training process is over, all the components of the ESN are kept fixed and a further stream of data at the input layer  $[t > t_r$ ; right part of the dashed vertical line; Fig. 2(a)] is passed into the ESN to predict the target signals beyond time  $t = t_r$ . The predicted sequences are shown by red circles at outputs, almost perfectly matching the targets (synthetic data in blue lines). The ESN shows a strong ability to predict the targeted data for almost all the data streams. Thus we claim here: *Feeding a wide variety of independent signals [for random choices of  $\beta_j$  and  $\mathcal{I}_j(0)$ ] into the ESN enables it to be well trained. The ESN does not require precise information on  $\beta_j$  or  $\gamma_j$ .*

For detailed clarification, we consider  $N = 1000$  independent time signals of infected data ( $\mathcal{I}$ ) among which  $M = 950$  time series (95% of the whole data set of all equal size) are used for training purpose. We target the remaining  $N - M = 50$  patches (5% of the whole data set) to be predicted by this

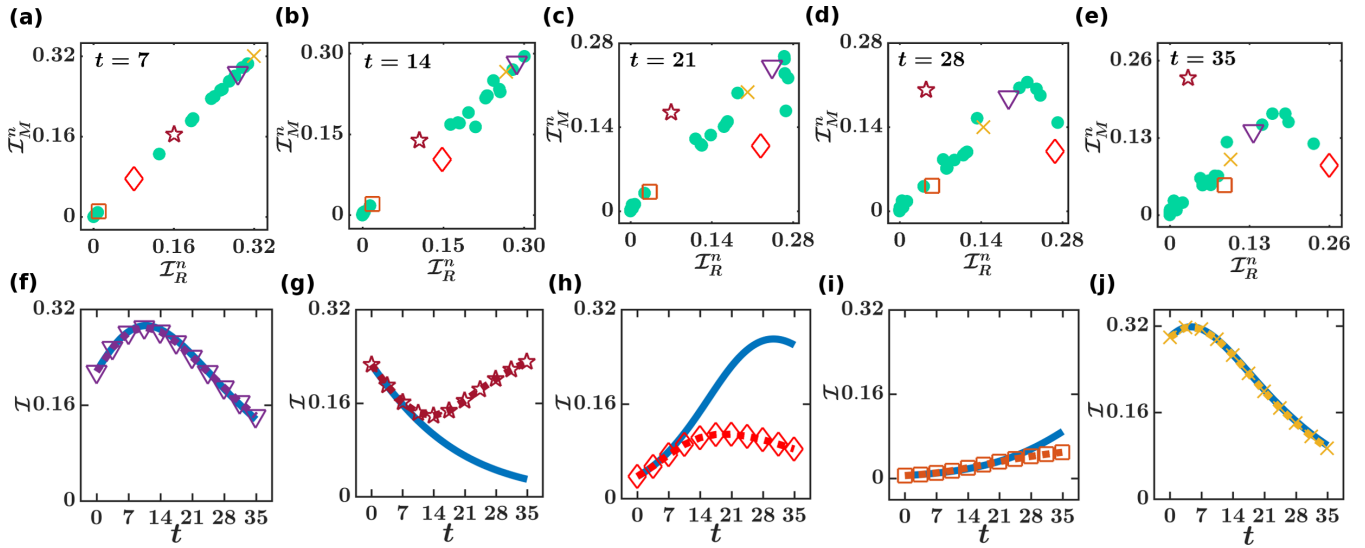


FIG. 3. (a)–(e) Snapshots of synthetically generated data versus machine-based data; ( $\mathcal{I}_R^n$ ) vs ( $\mathcal{I}_M^n$ ) plot for data taken at time points  $t = 7, 14, 21, 28,$  and  $35,$  respectively. Results of 50 patches are presented here. Five randomly chosen nodes are marked with squares, diamonds, pentagrams, triangles, and  $\times$ 's. At  $t = 7,$   $\mathcal{I}_M^n$  and  $\mathcal{I}_R^n$  are correlated, as they lie on the diagonal line, signifying that the machine can predict the real data efficiently. (b) At  $t = 14,$  two data points (diamond and pentagram) deviate slightly from their original counterparts. The error increases for a longer duration ( $t = 21, 28, 35$ ) of forecasting as shown in (c)–(e). However, most of the patches (green circles) lie on the diagonal line. (f)–(j) The infection trend of five randomly chosen nodes are shown for 5 weeks. Data generated by simulation of the SIR model are shown by thick blue lines. The machine-generated data closely predict infection in some of the patches (marked by triangles,  $\times$ 's, and squares) up to 30–35 days. For two patches (marked by pentagrams and diamonds), the machine-generated data deviate after 10 days.

95% data set through the ESN at the time of testing. A data set of  $t_r = 100$  days, i.e., 10 000 data points, is used for training purposes. After the ESN is trained (when the output layer is properly tuned), we predict the infection for the next 35 days (3500 data points) for the remaining 50 time-series data. The synthetic time series is obtained by integrating Eq. (4) for the  $j$ th location as designated by  $\mathcal{I}_R^n$ , whereas the ESN predicted data for the same are denoted  $\mathcal{I}_M^n$ . Figure 3(a) describes the correlation between  $\mathcal{I}_R^n$  and  $\mathcal{I}_M^n$  ( $n = 951, 952, \dots, 1000$ ) for all the patches at time  $t = 7$  (data during training are not shown here). All the patches (represented by filled green circles and another five markers for five patches) lie on the diagonal line, signifying excellent accuracy of prediction of the trained ESN. Five randomly identified patches are shown by five markers (triangle, pentagram, diamond, square, and  $\times$  markers). The corresponding signals are shown in Figs. 3(f)–3(j), where the true synthetic data [generated from Eq. (4)] are plotted with thick blue lines. Noticeably, the signal data for each patch closely match the true data at  $t = 7,$  confirming that the ESN predicts the trend of all patches with a higher accuracy. Next, we have checked  $\mathcal{I}_R^n$  and  $\mathcal{I}_M^n$  data at  $t =$  day 14 as shown in Fig. 3(b). Most of the patches (green circles) still lie on the diagonal line, confirming the prediction ability of the ESN, however, few patches (diamonds and pentagrams) deviate a little from the diagonal line, which is further confirmed in Figs. 3(g) and 3(h), where the predicted and the true signals start to deviate from each other after  $t \sim 14$  days. The more we increase the time of prediction, the larger the deviation that occurs for these two particular cases [see the positions of pentagram and diamond markers in Figs. 3(c)–3(e)]. Three particular patches (triangles, squares, and  $\times$ 's) are predicted with a higher accuracy, as they almost remain on the diagonal

line at  $t = 21, 28,$  and  $35.$  The related continuous time signals for the three patches are shown in Figs. 3(f), 3(i), and 3(j), respectively. A large fraction of green patches moves along the diagonal lines, ensuring the higher prediction ability of the ESN. Noticeably, the ESN can efficiently predict the signal during an increasing trend [cf. Fig. 3(h) for 10 days and Fig. 3(i) for 30 days]. Also, it can capture the decreasing trend [Fig. 3(g) for 14 days] and predict both for 35 days [Figs. 3(f) and 3(j)]. Thus, the nonmonotonicity of the infection trend can be captured by the ESN with a higher accuracy. It is noteworthy that the proposed approach works well if we increase the number of target locations up to 10%–20% (we have checked, but the results are not shown here). It was shown that under suitable conditions, an ESN of size  $N$  can memorize the previous inputs of size  $N$  [25]. Also, for complex systems (e.g., chaotic signals), the ESN has the ability to predict over a short time scale which is actually longer than the Lyapunov time scale [26]. In our example cases, signals are not chaotic, however, the epidemic curves are sensitive to the initial states (initial infection), leading to different outcomes [39]. On the other hand, the intrinsic epidemiological parameters of patches are not identical. Therefore, the times to attain the maximum of infection and peak of infection will vary from node to node. Thus the accuracy of prediction may fail after a certain time. From our numerical simulation, it is clear, for model-generated data, that all are accurately predicted up to 2 weeks. After that, due to the limitation of the memory capacity of the reservoir, time signals of certain nodes are poorly captured and machine-generated data behave abruptly such as in Fig. 3(g).

Next we try to validate our scheme using a COVID-19-infected data set. We have already confirmed that the ESN

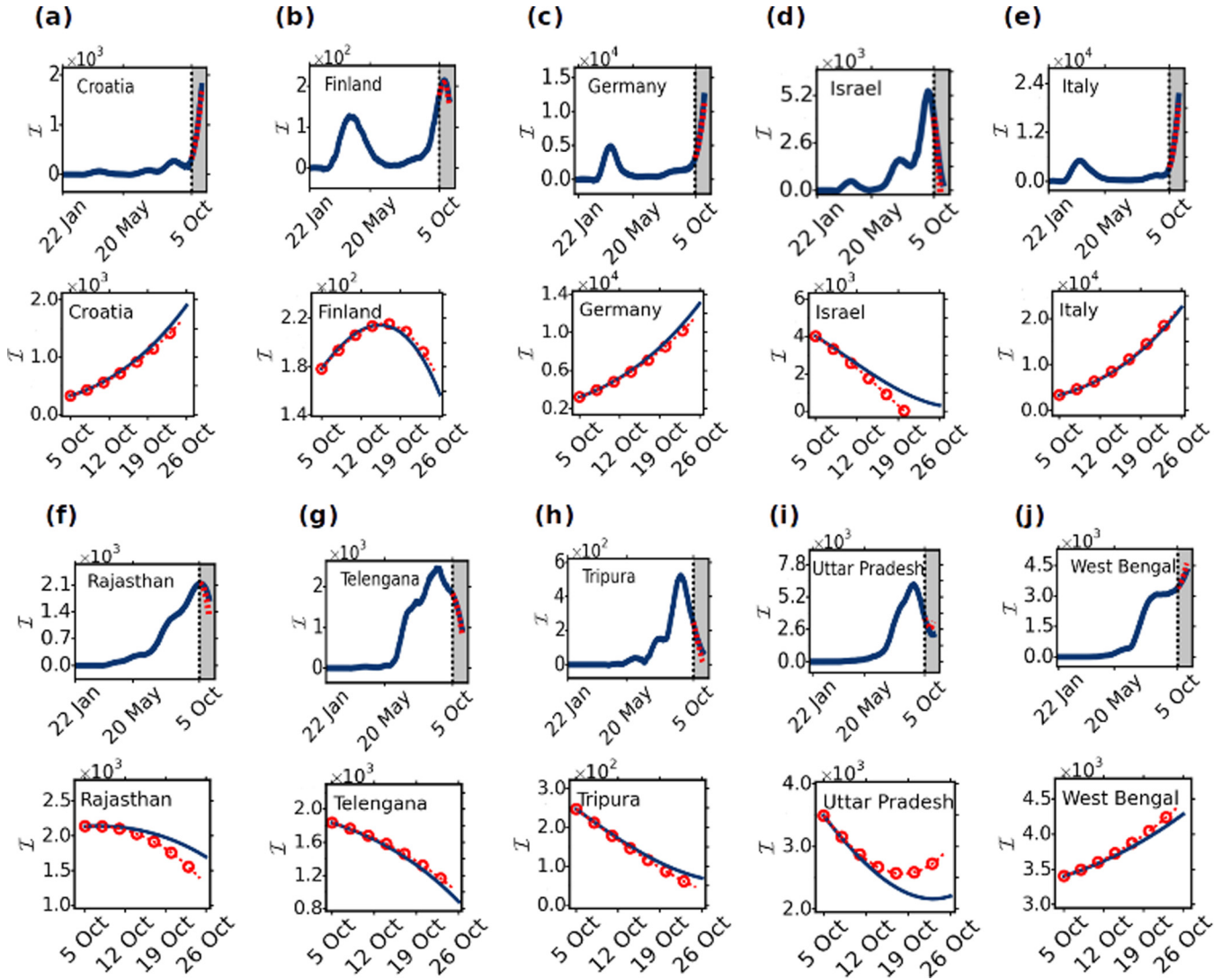


FIG. 4. Prediction of COVID-19 data of 10 randomly chosen locations from world COVID-19 data sets. (a) Croatia: infection sharply increases (blue line) at the time of prediction (5 October to 26 October; shaded region). The machine closely predicts (red circles) the trend (blue line) in the shaded region. A zoomed-in version of the shaded region is presented directly below (second row). Similar scenarios are observed for (b) Germany, (e) Italy, and (j) West Bengal. The decreasing trends of (d) Israel, (f) Rajasthan, (g) Telengana, and (h) Tripura are also well captured by the ESN. The ESN also predicts the trend in (i) Uttar Pradesh, but for a shorter time. Interestingly, the machine prediction of the increasing and decreasing trend of infection in (b) Finland is closely matched by the real data.

easily predicts the data on targeted locations by exploiting the infectious data on other locations (at the same time). In the next section, we reinvestigate the efficiency of the ESN for COVID-19 cases. It requires special mention that our scheme does not require any specific knowledge of the reproduction number of each location, duration of intervention (lockdown), or impact of mobility within the locations.

**IV. PREDICTION ON REAL DATA**

To check the feasibility of prediction by the ESN in a real outbreak scenario, we consider time-series data sets of 189 locations consisting of daily new cases of COVID-19 [40,41].

We have used daily infected data for all the locations or patches for 279 days (from 22 January to 26 October 2020). For training purposes, we consider the infection data for 257 days (22 January to 4 October) at each location. We decom-

pose the entire set into two groups. Infected data from 179 locations are fed into the ESN at the time of training and ESN predicts the current infection trend of 10 locations. The output weights are tuned in such a way that they can capture the infected cases for the 10 patches at the time of training. Once the training is over, we use infection data for 22 days of 179 locations to predict the infection trend of 10 locations. We have considered the size of the reservoir  $1000 \times 1000$  and fixed the leaking rate at  $\alpha = 0.5$ , and hence, the input matrix size is  $1000 \times 179$ . We predict the infection trend for 22 days extending from 5 October to 26 October 2020.

To preprocess the data, we have used the savgol filter (Python package) [42]. We consider all provinces in China and all states in the United States, Australia, France, and India. We have ignored data for some locations, which are not severely affected by the disease; data are removed if the cumulative infection is lower than  $\sim 10^4$ . For predictive purposes, we have

randomly picked five states in India (Rajasthan, Telengana, Tripura, Uttar Pradesh, and West Bengal) and five other countries (Croatia, Finland, Germany, Israel, and Italy). Thus the ESN can predict the cases of infection in most of the targeted locations for 3 weeks as shown in Figs. 4(a)–4(j) with real data (blue lines) and machine-generated data (dashed red lines with circles). Note that the daily infection has significantly increased for Germany, Italy, and Croatia as the disease reappears there. The gray shaded regions (from 5 October to 26 October) demarcate the predicted regimes for each location. For better clarity, we have also shown the predicted data (for the shaded regions) separately in zoomed-in versions (second and fourth rows in Fig. 4). Our proposition can efficiently determine an increasing or decreasing trend of infection in the targeted locations. Interestingly, for Finland [Fig. 4(b)], the ESN captures both the trends: initially increasing and later decreasing. Thus the ESN performs well for most of the randomly chosen locations from a large pool of infected data sets and predicts 5% of the entire data set using the 95% data set. We have checked that 20% patches can be predicted by our scheme for at most 2 weeks (not shown here). Dynamical modeling of COVID-19 data demands a large set of information including the effective reproduction number (the infection rate may change nonmonotonically), mobility through the transportation network, and a detailed description of a large number of compartments (variables). Our proposition overcomes this drawback and depends only on an available multidimensional data set. We expect that a higher resolution of the data set will enable the ESN to capture the infection trend of a larger number of target locations more accurately and to enhance the duration of prediction. Apart from the current prediction of targeted locations, we confirm with a revised scheme that the ESN can truly capture the future trend of infection data up to at least 10–14 days. We elaborate on this scheme in the next section.

**V. FUTURE FORECASTING: A PROPOSITION**

One may ask whether the proposed method can be used to capture the future trend of infection. Till now, we have predicted or traced the current data of selected locations by utilizing the data set for other locations at the same time (see scheme in Fig. 1). As we have claimed, usage of a large pool of desynchronized infection data series (all input data sets are independent and uncorrelated) in the input of the ESN makes it easier to predict the trend of infection of randomly selected other locations. With the same spirit, here we aim to predict the trend of infection of the above-mentioned target locations for a future duration of time. Note that the initial growth rate for this type of infection is slow (follows a power law [43]) compared to the growth rate at later times. Thus we assume that ignoring initial data (of targeted locations) for a few days will not affect the overall performance of the ESN. Thus, we hypothesize that a short duration time-shifting of the input data can lead us to forecast the future trend (for a short term, ~2 weeks) of infection of the target locations. To do this we use the following steps:

(1) *Spatial and temporal decomposition.* We collect the same data set from  $N$  locations. The data were saved from  $t = 0$  to  $t = t_f$ . We decompose the data set into two parts: input

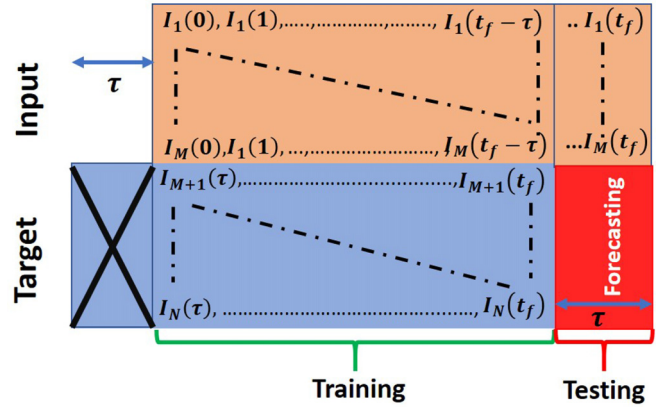


FIG. 5. Generalized approach to future forecasting of selected locations. This scheme enables prediction of the future trend of infection of the target locations for a duration  $\tau$  time unit.

( $M$  locations) and target [ $N - M$  locations,  $N \gg (N - M)$ ]. We shift each of the inputs with a  $\tau$  ( $\tau \ll t_f$ ) time unit, i.e., the input will be added to the machine at  $t = \tau$  (light-red rectangular regime in Fig. 5, top). As a consequence, we remove the initial trend of target locations up to the  $\tau$  time unit (rectangular blue regime with  $\times$  in Fig. 5, bottom right). We continue this learning process until all of the target data are utilized for training purposes, i.e., it will end at  $t = t_f$ . Therefore  $t_f - \tau$  input data points will be used to train the machine such that it can capture the  $M$ -dimensional target data from  $t = \tau$  to  $t = t_f$ .

(2) *Forecasting using the testing procedure.* Now we can use the trained machine to forecast the target data from  $t_f$  to  $t_f + \tau$  (dark-red regime; Fig. 5, bottom right) from the input data extending from  $t_f - \tau + 1$  to  $t_f$  (light-red regime; Fig. 5, top right). The green brace below the light-blue matrix (bottom left) represents the training time and the red brace (bottom right) signifies the future forecasting of the target locations.

**Forecasting future trends from COVID-19 data**

To validate our forecasting scheme, we have used COVID-19 data [40,41] preprocessed for 465 days: from 22 January 2020 to 30 April 2021). We decompose the entire set into two groups. Infected data from 241 locations are fed into the ESN at the time of training. We target to forecast the infection trend of 10 other locations. To forecast 14 days in the future, we have discarded the initial 14 days from the targeted data. We have trained the machine by utilizing the COVID-19 data from 22 January 2020 to 16 April 2021 (total, 451 days) to track the target data from 4 February 2020 to 30 April 2021 (total, 451 days). After training is finished, we forecast 14 days of data on targeted locations from 1 May 2021 to 14 May 2021. Note that we have data in hand until 30 April 2021. However, we can forecast for 14 more days, from 1 May to 14 May. The machine-generated data are represented by gray shading (Fig. 6) for 100 realizations. In each realization, the reservoir weights are randomly changed. The blue line is the average of these 100 realizations. Our machine-generated prediction reflects that in most of the states in India [Figs. 6(f)–6(j)],

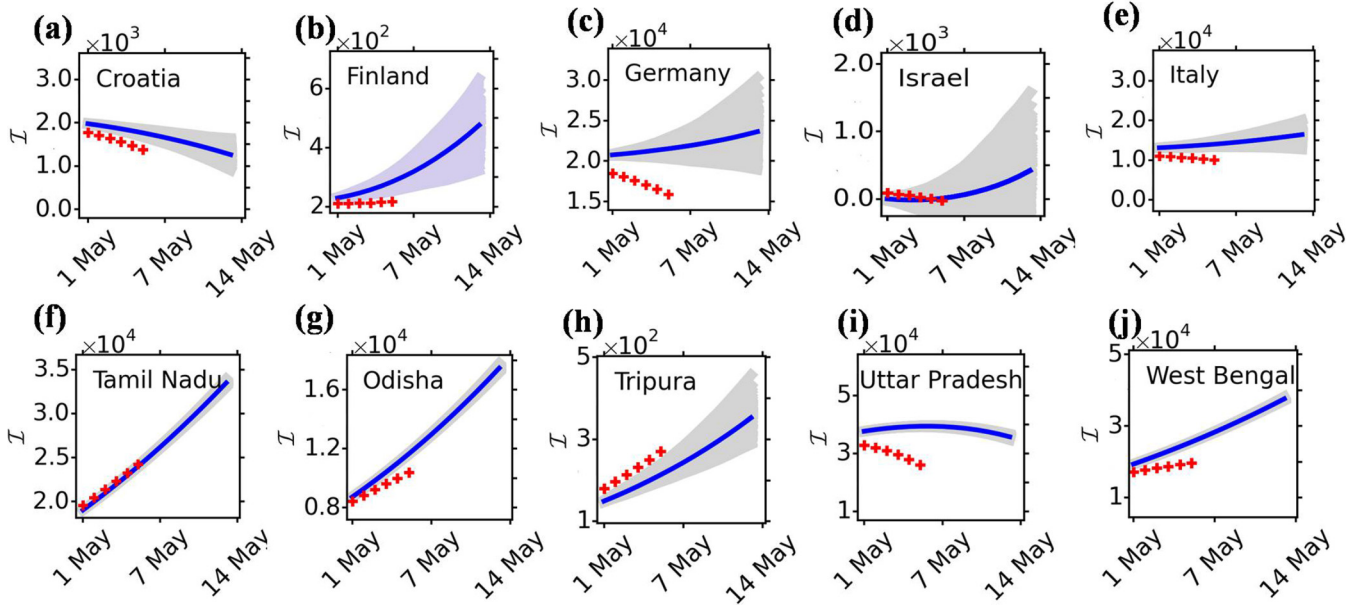


FIG. 6. Forecasting the future trend of infection of 10 selected locations from 1 May 2021 to 14 May 2021. The gray region in each figure represents the machine-generated data for 100 realizations. The blue line shows the average of each gray region. Red crosses show the real data for the period 1 May 2021 to 6 May 2021.

the daily infection will increase, except in Uttar Pradesh [Fig. 6(i)]. The real data for each location in India are shown by red crosses (from 1 May to 6 May 2021). Clearly, the machine-generated data closely follow the original trend of infection. Trends in Italy and Germany are weakly captured [Figs. 6(c) and 6(e)] by the machine, whereas the slow increase (decrease) in Israel (Croatia) is closely matched by the real data. The upper bound of  $\tau$  (initial delay in data input) by increasing (or decreasing) the number of targeted locations is a real question that demands further investigation in future.

**VI. CONCLUSION**

We have proposed a machine learning-based mechanism for efficient prediction of COVID-19 infection. A modified version of the neural network (ESN) has been used to predict new infections in randomly chosen locations. Available data from a large number of locations are utilized to train the machine such that it can map the infection trend of other locations, which we call target locations.

The proposed technique does not depend largely on the intrinsic parameters of the ESN. In the literature, there exist several phenomenological models [44–46] for predicting the trend of infection. However, these models have limitations for prediction due to intrinsic uncertainties in the system parameters. For instance, the well-known Gompertz function cannot capture the trend of the second wave [47–49] of infection, whereas it can efficiently predict the initial daily infection. Also, suitable choices of parameters of the Gompertz function immediately before the prediction are necessary (please see the Appendix for a detailed investigation of the Gompertz function). In our model-free machine learning scheme this restriction is relaxed, as the ESN can successfully trace the second wave of specific locations [see Figs. 4(b)–4(d) and Fig. 6]. Forecasting is really a challenging task, however, we

have proposed a second scheme using a data-shifting technique during the training process that shows promising results for future forecasting. We expect that our proposition might be useful for diverse sets of spatiotemporal data, ranging from physiological to multivariate climate data. In the same manner, we can use other types of recurrent neural networks for prediction of infection trends of certain locations that we intend to try in future.

**ACKNOWLEDGMENTS**

C.H. and S.G. were supported by the INSPIRE-Faculty grant (Code IFA17-PH193). J.C. was supported by the Technology Innovation Hub on Data Science, Big Data Analytics and Data Curation (Grant No. NMICPS/006/MD/2020-21; 16 October 2020).

**APPENDIX: PREDICTION THROUGH THE GOMPERTZ CURVE**

In the literature, there are many models and mechanisms available for data fitting which enable us to estimate suitable parameters for short-term forecasts as well as the uncertainty in forecasting. For instance, the generalized Richard model, logistic growth model, subepidemic wave model [44,45], flexible growth model curve [46], and Gompertz curve [48,49] have been widely used for forecasting infection data. We use the following Gompertz function [47] for capturing the daily infection:

$$I(t) = aKe^{-\ln(\frac{K}{N_0})e^{-at}} \left( \ln\left(\frac{K}{N_0}\right)e^{-at} \right), \quad (A1)$$

where the parameter  $K$  is the saturating value of the infected cases,  $N_0$  is the initial infection, and  $a$  represents the

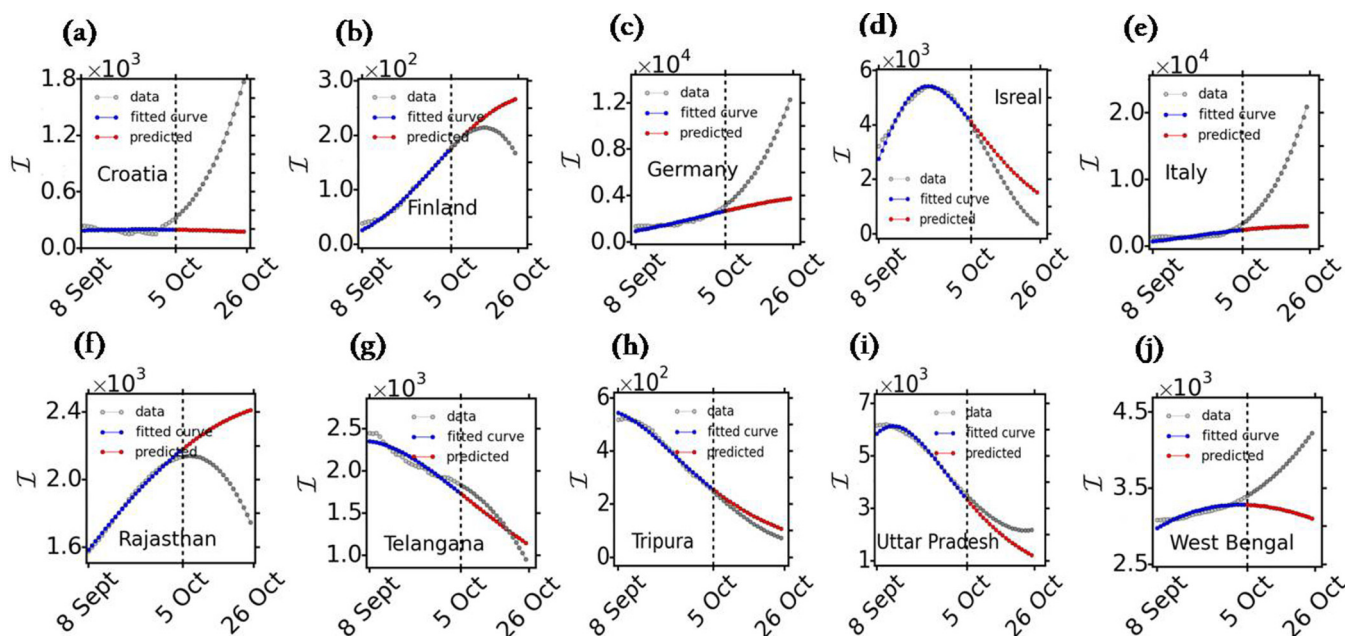


FIG. 7. Forecasting through the Gompertz curve in 10 randomly chosen regions. Four weeks of data are used to standardize the intrinsic parameters of the Gompertz function. Twenty-two days were forecast, from 5 October to 26 October 2020.

decreasing trend of the initial exponential growth. Now we estimate the parameters  $a$  and  $K$  to predict the infection pattern from 5 October to 26 October 2020 (22 days). Here we take daily infection data for 10 countries or regions for 4 weeks, from 8 September to 5 October 2020, and fit it with the Gompertz curve to obtain the best-fitted parameters.

We can see that the Gompertz curve is able to provide good predictions for certain locations (Fig. 7; Israel, Uttar Pradesh, Telangana, Tripura, and Finland) and weakly predicts the infection trend in Croatia, Germany, West Bengal, Rajasthan, and Italy. However, machine-generated data perform well in most of the cases (see Fig. 4 for comparison).

- 
- [1] M. Perc, N. Gorišek Miksić, M. Slavinec, and A. Stožer, *Front. Phys.* **8**, 127 (2020).
- [2] G. Grasselli, A. Pesenti, and M. Cecconi, *JAMA* **323**, 1545 (2020).
- [3] F. Petropoulos and S. Makridakis, *PLoS One* **15**, e0231236 (2020).
- [4] C. Anastassopoulou, L. Russo, A. Tsakris, and C. Siettos, *PLoS One* **15**, e0230405 (2020).
- [5] A. J. Kucharski, T. W. Russell, C. Diamond, Y. Liu, J. Edmunds, S. Funk, R. M. Eggo, F. Sun, M. Jit, J. D. Munday *et al.*, *Lancet Infect. Dis.* **20**, 553 (2020).
- [6] A. Vespignani, H. Tian, C. Dye, J. O. Lloyd-Smith, R. M. Eggo, M. Shrestha, S. V. Scarpino, B. Gutierrez, M. U. Kraemer, J. Wu *et al.*, *Nat. Rev. Phys.* **2**, 279 (2020).
- [7] J. Hellewell, S. Abbott, A. Gimma, N. I. Bosse, C. I. Jarvis, T. W. Russell, J. D. Munday, A. J. Kucharski, W. J. Edmunds, F. Sun *et al.*, *Lancet Global Health* **8**, e488 (2020).
- [8] T. Colbourn, *Lancet Public Health* **5**, e236 (2020).
- [9] S. Ghosh, A. Senapati, J. Chattopadhyay, C. Hens, and D. Ghosh, *Chaos* **31**, 071101 (2021).
- [10] D. L. Heymann and N. Shindo, *Lancet* **395**, 542 (2020).
- [11] J. Tsai and M. Wilson, *Lancet Public Health* **5**, e186 (2020).
- [12] A. A. Al Momani and E. Bollt, *arXiv:2004.08897*.
- [13] L. Gallo, M. Frasca, V. Latora, and G. Russo, *arXiv:2012.00443*.
- [14] S. Lalmuanawma, J. Hussain, and L. Chhakchhuak, *Chaos Solitons Fractals* **139**, 110059 (2020).
- [15] A. Senapati, S. Rana, T. Das, and J. Chattopadhyay, *J. Theor. Biol.* **523**, 110711 (2021).
- [16] Z. Yang, Z. Zeng, K. Wang, S.-S. Wong, W. Liang, M. Zanin, P. Liu, X. Cao, Z. Gao, Z. Mai *et al.*, *J. Thorac. Dis* **12**, 165 (2020).
- [17] Z. Li, Y. Zheng, J. Xin, and G. Zhou, *arXiv:2007.10929*.
- [18] A. Fokas, N. Dikaos, and G. Kastis, *J. R. Soc. Interface* **17**, 20200494 (2020).
- [19] L. Yan, H.-T. Zhang, J. Goncalves, Y. Xiao, M. Wang, Y. Guo, C. Sun, X. Tang, L. Jing, M. Zhang *et al.*, *Nat. Mach. Intell.* **2**, 283 (2020).
- [20] M. H. D. M. Ribeiro, R. G. da Silva, V. C. Mariani, and L. dos Santos Coelho, *Chaos Solitons Fractals* **135**, 109853 (2020).
- [21] S. Singh, K. Parmar, S. Jitendra Singh, J. Kaur, S. Peshoria, and J. Kumar, *Chaos Solitons Fractals* **139**, 110086 (2020).
- [22] T. Chakraborty and I. Ghosh, *Chaos Solitons Fractals* **135**, 109850 (2020).
- [23] M. Wiecek, J. Silka, and M. Woźniak, *Chaos Solitons Fractals* **140**, 110203 (2020).
- [24] A. Di Castelnuovo, M. Bonaccio, S. Costanzo, A. Gialluisi, A. Antinori, N. Berselli, L. Blandi, R. Bruno, R. Cauda, G. Guaraldi *et al.*, *Nutr. Metab. Cardiovasc. Dis.* **30**, 1899 (2020).
- [25] H. Jaeger and H. Haas, *Science* **304**, 78 (2004).



- [26] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, *Phys. Rev. Lett.* **120**, 024102 (2018).
- [27] R. S. Zimmermann and U. Parltitz, *Chaos: Interdisc. J. Nonlin. Sci.* **28**, 043118 (2018).
- [28] J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott, *Chaos: Interdisc. J. Nonlin. Sci.* **27**, 121102 (2017).
- [29] Z. Lu, B. R. Hunt, and E. Ott, *Chaos: Interdisc. J. Nonlin. Sci.* **28**, 061104 (2018).
- [30] X. Lin, Z. Yang, and Y. Song, *Expert Syst. Appl.* **36**, 7313 (2009).
- [31] X. Hinaut and P. F. Dominey, *PLoS One* **8**, e52946 (2013).
- [32] D. Verstraeten, B. Schrauwen, D. Stroobandt, and J. Van Campenhout, *Info. Process. Lett.* **95**, 521 (2005).
- [33] T. Weng, H. Yang, C. Gu, J. Zhang, and M. Small, *Phys. Rev. E* **99**, 042203 (2019).
- [34] T. Lymburn, D. M. Walker, M. Small, and T. Jüngling, *Chaos: Interdisc. J. Nonlin. Sci.* **29**, 093133 (2019).
- [35] X. Chen, T. Weng, H. Yang, C. Gu, J. Zhang, and M. Small, *Phys. Rev. E* **102**, 033314 (2020).
- [36] A. Panday, W. S. Lee, S. Dutta, and S. Jalan, *Chaos: Interdisc. J. Nonlin. Sci.* **31**, 031106 (2021).
- [37] S. Saha, A. Mishra, S. Ghosh, S. K. Dana, and C. Hens, *Phys. Rev. Research* **2**, 033338 (2020).
- [38] J. Hilton and M. J. Keeling, *PLoS Comput. Biol.* **16**, e1008031 (2020).
- [39] M. Castro, S. Ares, J. A. Cuesta, and S. Manrubia, *Proc. Natl. Acad. Sci. USA* **117**, 26190 (2020).
- [40] B. Xu, M. U. Kraemer, B. Gutierrez, S. Mekaru, K. Sewalk, A. Loskill, L. Wang, E. Cohn, S. Hill, A. Zarebski *et al.*, *Lancet Infect. Dis.* **20**, 534 (2020).
- [41] <https://covid19.who.int/>
- [42] Codes to reproduce the results presented here are freely accessible at <https://github.com/subrata-chitta/Reservoir-computing-on-epidemic-spreading>
- [43] B. F. Maier and D. Brockmann, *Science* **368**, 742 (2020).
- [44] A. Smirnova and G. Chowell, *Infect. Dis. Model.* **2**, 268 (2017).
- [45] K. Roosa, Y. Lee, R. Luo, A. Kirpich, R. Rothenberg, J. M. Hyman, P. Yan, and G. Chowell, *J. Clin. Med.* **9**, 596 (2020).
- [46] C. F. Tovissodé, B. E. Lokonon, and R. Glèlè Kakaï, *PLoS One* **15**, e0240578 (2020).
- [47] M. Català, S. Alonso, E. Alvarez-Lacalle, D. López, P.-J. Cardona, and C. Prats, *PLoS Comput. Biol.* **16**, e1008431 (2020).
- [48] Á. Berihuete, M. Sánchez-Sánchez, and A. Suárez-Llorens, *Mathematics* **9**, 228 (2021).
- [49] A. Ohnishi, Y. Namekawa, and T. Fukui, *Prog. Theor. Exp. Phys.* **2020**, 123J01 (2020).