Reservoir computing on epidemic spreading: A case study on COVID-19 cases

Ghosh, S.; Senapati, A.; Mishra, A.; Chattopadhyay, J.; Dana, S. K.; Hens, C.; Ghosh, D.;

Originally published:

July 2021

Physical Review E 104(2021), 014308

DOI: https://doi.org/10.1103/PhysRevE.104.014308

Reservoir computing on epidemic spreading: A case study on COVID-19 cases

Subrata Ghosh,[1] Abhishek Senapati,[2,3] Arindam Mishra,[4] Joydev Chattopadhyay,[2] Syamal K. Dana,[4] Chittaranjan Hens,[1,*] and Dibakar Ghosh[1]

[1] *Physics and Applied Mathematics Unit, Indian Statistical Institute, 203 B. T. Road, Kolkata 700108, India*
[2] *Agricultural and Ecological Research Unit, Indian Statistical Institute, 203 B. T. Road, Kolkata 700108, India*
[3] *Center for Advanced Systems Understanding (CASUS), Goerlitz, Germany*
[4] *Department of Mathematics, Jadavpur University, Kolkata 700032, India*

(Dated: September 15, 2022)

A reservoir computing/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 by the infection data of 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, 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 approximately for 4 to 5 weeks. The machine performance is further tested with real data of the current COVID-19 pandemic collected for different countries. We show that our proposed scheme is able to predict the trend of the disease up to 3 weeks for some targeted locations. An important point to note that no detailed information of 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.

I. INTRODUCTION

The impact of the unprecedented pandemic COVID-19 is widespread practically collapsing all human activities around the world. A severe crisis arises in the public health systems and economy everywhere. In this extreme condition, various agencies, government and non-government, are looking for ways and means to stop spreading 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 spreading and to keep preparing the public health system to innumerably large demands [1–4].

An accurate prediction methodology may enable the policy makers to deter the spreading 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, exponential smoothing model can forecast [3] the COVID-19 confirmed 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 of infection where susceptible-infected-recovered (SIR) dynamics (constructing differential equations) is adjusted with recurrent neural network to forecast the temporal data with limited resources. Many other approaches such as deep learning using long-short-term-memory network (LSTM) [18, 19], support vector machine [20, 21], hybrid autoregressive moving average model [20, 22], neural network [23], supervised XGBoost classifier [19], 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 the several structural parameters as well as intrinsic components of the machine itself. The successful forecasting by machine learning is also deterred by limited availability of temporal data. The key question we raise here whether there is any possibility of predicting the infection trend of a disease, in general, in targeted locations by feeding infection data of the disease available from other locations around different countries? We accept the constraint that detail information on 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 the reservoir computing i.e., the echo-state network (ESN). 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. ESN has been used extensively to predict complex signals ranging from chaotic time series to stock-price data [25–32] and currently, it has been shown that it can easily capture critical onset of generalized synchronization [33–36]

* Corresponding author: chittaranjanhens@gmail.com

and detect collective bursting in neuron populations [37]. Therefore, ESN showed encouraging records of handling multiple inputs of temporal data and, ability to trace the correlation between them [34, 37]. Motivated by this fact, we have utilized the strength of the ESN to develop a strategy for predicting the spread of any infectious disease from the available collection of multitude of infection data of the same disease.

At first we check the efficiency of the ESN for a large collection of synthetic epidemic data generated from classical SIR model. Finally, the prediction capability of the ESN is carefully investigated with available incidence data of COVID-19 from large number of locations around the world with an aim to identify the real outbreak scenario in other targeted locations. The machine works successfully to predict spreading of the disease to the extent of two weeks and little more. ESN is thus shown as 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 non-monotonic trend of 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 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.

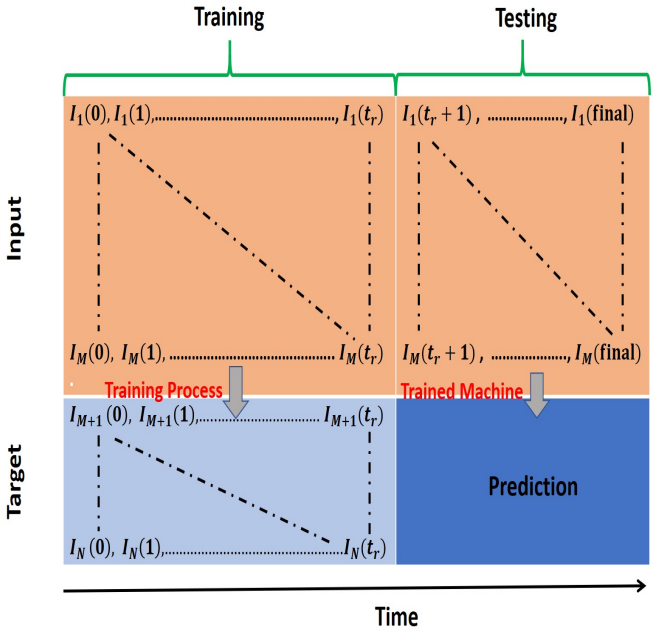

FIG. 1: Training and testing scheme using echo state network. Upper panel (light red boxes) are the input parts. Lower panel (light blue parts) in the left represents the data set of the output data. Right corner panel (deep blue box) is the predicted/testing data.

II. DESCRIPTION OF ECHO-STATE NETWORK (ESN)

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

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

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

Next we consider time series data of N patches, among which the data of M patches are fed into the machine and the remaining $N - M$ patches are targeted whose time signals are to be predicted by ESN. A fraction of data points (when $t = 0, 1, ..., t_r$) from each of the infected signals is used for training purpose (see upper left panel (light red) illustrated in a scheme in Fig. 1). At first target is to identify the infection of the rest of the patches $(N - M)$ by the ESN during the training or learning process (lower left panel, light blue in Fig. 1). Once the machine is trained, input from M patches with rest of the data points $(t_r + 1, ..., t_{\mathrm{final}})$ are fed into the machine (upper right panel, light red in Fig. 1) to predict the infection in the $N - M$ patches (lower right panel, deep blue in Fig. 1).

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

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

In the training process, at each time instant t, the reservoir state $\mathbf{r}(\mathrm{t})$ and input $\mathbf{s}(\mathrm{t})$ are accumulated in $\mathbf{X}(\mathrm{t}) = [1; \mathbf{s}(\mathrm{t}); \mathbf{r}(\mathrm{t})]$. The output relation can be written in vector form as:

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

Here, $\mathbf{Y}$ is a matrix of dimension $(N-M)\times K$, where K is the length of the time signal. The matrix X having dimension $(N_{\rm res}+M+1)\times K$ look like:

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \mathcal{I}(1,1) & \mathcal{I}(1,2) & \cdots & \mathcal{I}(1,K) \\ \mathcal{I}(2,1) & \mathcal{I}(2,2) & \cdots & \mathcal{I}(2,K) \\ \vdots & \vdots & \vdots & \vdots \\ \mathcal{I}(M,1) & \mathcal{I}(M,2) & \cdots & \mathcal{I}(M,K) \\ r(1,1) & r(1,2) & \cdots & r(1,K) \\ r(2,1) & r(2,2) & \cdots & r(2,K) \\ \vdots & \vdots & \vdots & \vdots \\ r(N_{\rm res},1) & r(N_{\rm res},2) & \cdots & r(N_{\rm res},K) \end{bmatrix}.$$

The matrix $\mathbf{W}_{\rm out}$ can be determined by Ridge regression method as follows:

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

where λ is the regularization factor that avoids over fitting. $\mathbf{Y}$ is the time series data of the targeted patches and I is identity matrix of dimension $(N_{\rm res}+M+1)\times(N_{\rm res}+M+1)$. Note that when $\lambda=0$, Eq. 3 reduces to least-square method.

We consider N patches, in which M number of patches are fed into the machine for training purpose. 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 ON SYNTHETIC DATA

The classical SIR model is used to numerically generate a large set of independent synthetic time series data (say $i=1,2,...,N$) on infection for different sets of disease transmission rates and initial fraction of 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 3-coupled equations:

$$\dot{\mathcal{S}}_j(t) = -\beta_j\mathcal{S}_j(t)\mathcal{I}_j(t), \tag{4}$$
$$\dot{\mathcal{I}}_j(t) = \beta_j\mathcal{S}_j(t)\mathcal{I}_j(t)-\gamma_j\mathcal{I}_j(t), \tag{5}$$
$$\dot{\mathcal{R}}_j(t) = \gamma_j\mathcal{I}_j(t). \tag{6}$$

Based on 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 β_j and γ_j denote the rate of disease transmission and recovery rate, respectively. We fix the recovery rate at $\gamma_1=\gamma_2=...=\gamma_N=1/14$ day^{-1} for this study. We generate a set of N independent synthetic data series by random choices of β_j from 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 β_j is based on available data and country level estimation of basic reproduction number for COVID-19 [38] that varies from 0 to 3.5. The model (4) is integrated for a time interval $[0,300]$ with a time step 0.01 using the RK4 routine. Therefore, each synthetic data set contains 30000 data points (300 days). Since a variation in disease transmission rate (β_j) and initial fraction of 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.

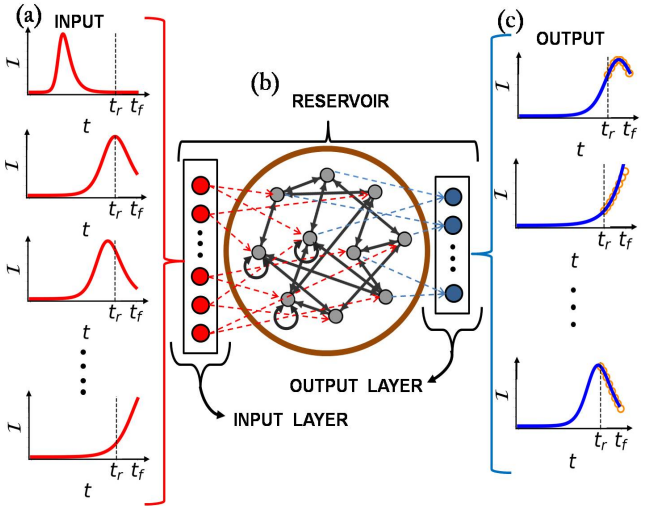

FIG. 2: Schematic representation of the ESN and signal variability. (a) Infection data inputs (red signal) fed into the machine. Dashed vertical line (at $t=t_r$) signifies a time limit of data point inputs to the machine. (b) ESN structure: input layer, reservoir and output layer. The weights of input 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 ($t\le t_r$) 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.

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

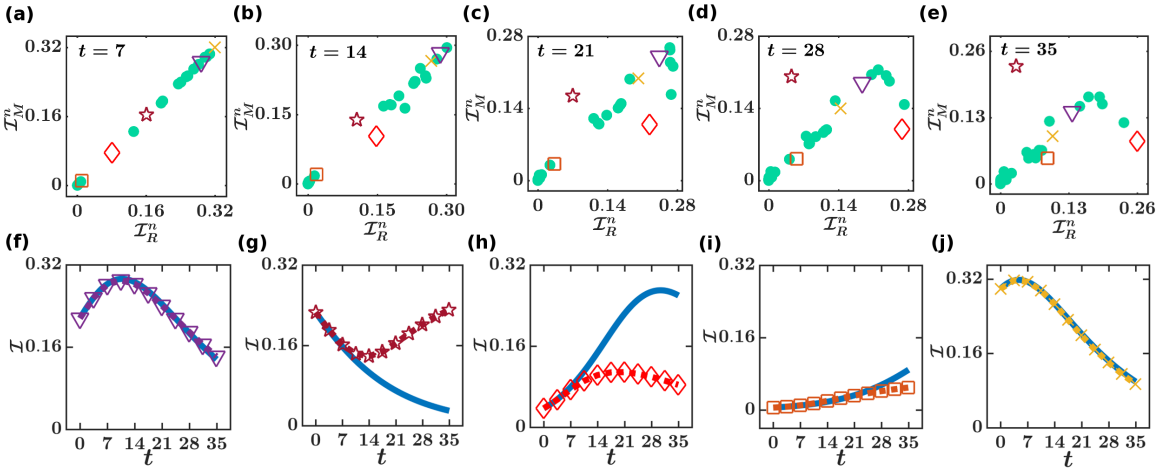

FIG. 3: (a)-(e) Snapshot of synthetically generated data against machine based data, $(\mathcal{I}_R^n)$ vs. $(\mathcal{I}_M^n)$ plot for data taken at time points $t = 7, 14, 21, 28,$ and 35, respectively. Results of 50 patches are presented here. 5 randomly chosen nodes are marked by square, diamond, pentagram, triangle and cross marks. At $t = 7$, $\mathcal{I}_M^n$ and $\mathcal{I}_R^n$ are correlated as they lie on the diagonal line, signifying the machine can predict the real data efficiently. (b) At $t = 14$, two data points (diamond and pentagram) are slightly deviated from their original counter part. 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 5 randomly chosen nodes are shown for five weeks. Data generated by simulation of the SIR model is shown with thick lines (blue line). The machine generated data closely predicts infection in some of the patches (marked by triangle, cross and square) upto 30-35 days. For two patches (marked by pentagram and diamond), the machine generated data are deviated after 10 days.

each of the red signals (Fig. 2(a)). The top-most signal is infected earlier and reaches the zero state before the time t_r. The second signal below from the top reaches its peak at $t = t_r$. The third one reaches the infection peak earlier than $t = t_r$. The last one at the bottom is gradually increasing and yet to reach the peak at time t_r. The sequence of data until $t = t_r$ of each red signal is fed into the ESN (Fig. 2(b)) for training purposes. Note that the ESN has three components, (a) input layer, which captures the input data, (b) reservoir network that associates the input data to its nodes generally in a nonlinear way, (c) 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, the part of the output signals (left of the dashed vertical lines; in blue curves) are not similar to each other: the upper one does not reach at 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)) are passed into the ESN to predict the target signals beyond the time $t = t_r$. The predicted sequences are shown in circles (red circles) at outputs almost in perfect matching with the targets (blue lines). 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 β_j and $\mathcal{I}_j(0)$) into ESN enables it to be well trained. ESN does not require precise information of β_j or γ_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 95% data set through ESN at the time of testing. A data set of $t_r = 100$ days, i.e., 10000 data points is used for the training purpose. After the ESN is trained (when the output layer is properly tuned), we predict the infection for next 35 days (3500 data points) for the remaining 50 time series data. The synthetic time series is obtained by integrating Eqn. 4 for the j^{th} location as designated by $\mathcal{I}_R^n$ whereas the ESN predicted data for the same is denoted by $\mathcal{I}_M^n$. Figure 3(a) describes the correlation between $\mathcal{I}_R^n$ and $\mathcal{I}_M^n$ $(n = 951, 952, ..., 1000)$ for all the patches at time $t = 7$ (data during training is not shown here). All the patches (represented by filled green circles and other five markers for 5 patches) lie on the diagonal line signifying an excellent accuracy of prediction of the trained ESN. Five randomly identified patches are shown by five markers (triangle, pentagram, diamond, square, and cross markers). The corresponding signals are shown in Fig. 3(f-j), where the true synthetic data (generated from Eqn. (4)) are plotted with thick lines (blue line). Noticeably, signal data of each patch closely matches with the true data at $t = 7$ confirming that ESN predicts the trend of all patches with higher accuracy. Next

we have checked $\mathcal{I}_R^n$ and $\mathcal{I}_M^n$ data at $t = 14^{th}$ day as shown in Fig. 3(b). Most of the patches (green circles) still lie on the diagonal line confirming the prediction ability of ESN, however, few patches (diamond and pentagram) are little deviated from the diagonal line, which is further confirmed from the Fig. 3(g-h) where the predicted and the true signals start to deviate to each other after $t \sim 14$ days. The more we increase the time of prediction, the larger a deviation occurs for these two particular cases (see the position of pentagram and diamond markers in Fig. 3(c-e)). Three particular patches (triangle, square and cross) are predicted with higher accuracy as they almost remain on the diagonal line at $t = 21, 28, 35$. 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 move along the diagonal lines ensuring the higher prediction ability of 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 non-monotonicity of the infection trend can be captured by ESN with higher accuracy. Noteworthy that the proposed approach works well if we increase the number of target locations up to $10\% - 20\%$ (we have checked, but results are not shown here). It was shown that under suitable conditions, ESN of size N can memorize the previous inputs of size N[25]. Also for complex systems (e.g chaotic signals), the ESN has ability to predict in a short time scale which is actually greater than the Lyapunov time scale [26]. In our example cases, signals are not chaotic, however, the epidemic curves are sensitive to initial states (initial infection) leading to different outcomes [39]. On the other hand, the intrinsic epidemiological parameters of patches are not identical. Therefore, time 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 two weeks. After that, due to the limitation of memory capacity of the reservoir, time signal of certain nodes are poorly captured and machine generated data behaves abruptly in Fig. 3 (g).

Next we try to validate our scheme using COVID-19 infected data set. We have already confirmed that the ESN easily predicts the data of targeted locations by exploiting the infectious data of other locations (at the same time). In the next section, we re-investigate the efficiency of ESN for COVID-19 cases. It needs a special mention that our scheme does not require any specific knowledge of the reproduction number of each location, duration of intervention (lock down) and impact of mobility within the locations.

IV. PREDICTION ON REAL DATA

To check the feasibility of prediction by ESN in a real outbreak scenario, we consider time-series data sets of 189 locations consisting of daily new cases of COVID-19 [40], [https://covid19.who.int/]. We have used daily infected data for all the locations/patches for 279 days (from 22 January to 26 October 2020). For training purpose, we consider the infection data for 257 days (22 January to 4 October) of each location. We decompose the entire set into two groups. Infected data of 179 locations are fed into the ESN at the time of training and predict the current infection trend of other 10 locations. The output weights are tuned in such a way that it 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 1000×1000, and fixed the leaking rate at $\alpha = 0.5$ and hence, the input matrix size is 1000×179 and the output matrix is 1179×10. We predict the infection trend for 22 days extending from 5^{th} October to 26^{th} October 2020.

To pre-process the data, we have used the savgol filter (python package). We consider all provinces of China and all states of USA, Australia, France and India. We have ignored data of some locations, which are not severely affected by the disease; data are removed if the cumulative infection is lower than $\sim 10^4$. For the prediction purpose, we have randomly picked 5 states of India (Rajasthan, Telengana, Tripura, Uttar Pradesh, and West Bengal) and 5 other countries (Croatia, Finland, Germany, Israel, and Italy). Thus ESN can predict the cases of infection in most of the targeted locations for 3 weeks as shown in Fig. 4 (a-j) with real data (blue lines) and machine generated data (red dashed lines with circle). Note that daily infection has significantly increased for Germany, Italy and Croatia as the disease reappears there. The shaded grey regions (from October 5 to October 26) demarcate the predicted regimes for each location. For a better clarity, we have also shown the predicted data (for the shaded regions) separately in zoomed versions (second and fourth rows of the Fig. 4). Our proposition can efficiently determine the increasing or the decreasing trend of infection in the targeted locations. Interestingly, for Finland (Fig. 4(b)), ESN captures both the trends: initially increasing and decreasing at later time. Thus ESN performs well for most of the locations randomly chosen 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 at most for two weeks (not shown here). A dynamical modeling of COVID-19 data demands a large set of information about effective reproduction number (infection rate may change non-monotonically), mobility through transportation network and detailed description of large number of compartments (variables). Our proposition overcomes this drawback and depends only on available multi-dimensional data set. We expect a

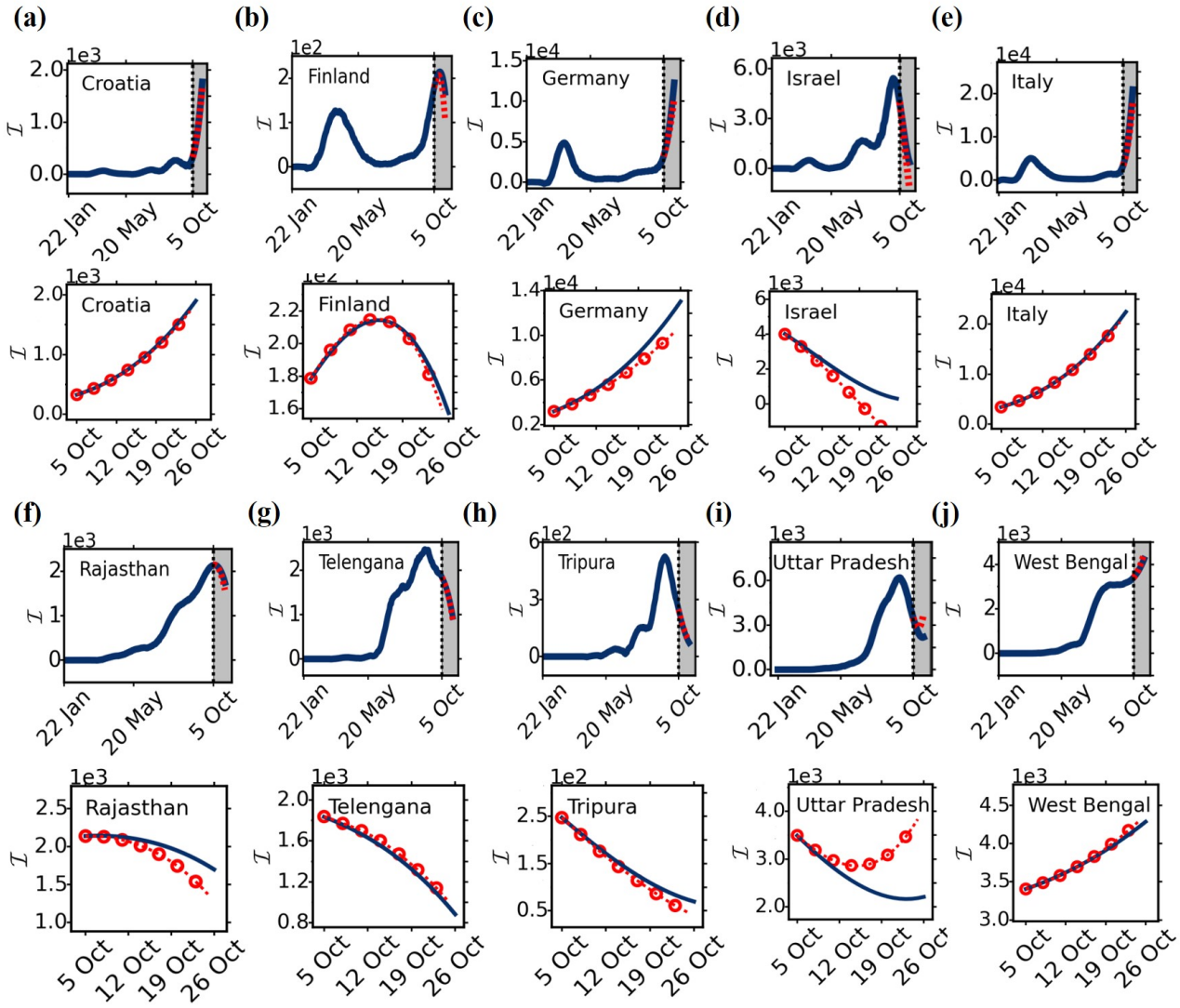

FIG. 4: Prediction of COVID-19 data of 10 randomly chosen locations from world data sets on COVID-19: (a) Croatia: infection sharply increases (blue line) at the time of prediction (October 5 to October 26, shaded region). Machine closely predicts (red circles) the trend (blue line) in the shaded region. A zoomed version of the shaded region is presented in an immediate lower panel (second row). Almost similar scenarios 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. 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 with real data.

higher resolution of 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 ESN can truly capture the future trend of infection data at least upto 10-14 days. We elaborate 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/traced the current data of selected locations by utilizing the data set of other locations (see the scheme in Fig. 1). As we have claimed, usage of the large pool of de-synchronized infection data series (all input data sets are independent and uncorrelated) in the input of ESN makes it easier to predict the trend of infection of randomly selected locations. Against the same backdrop,

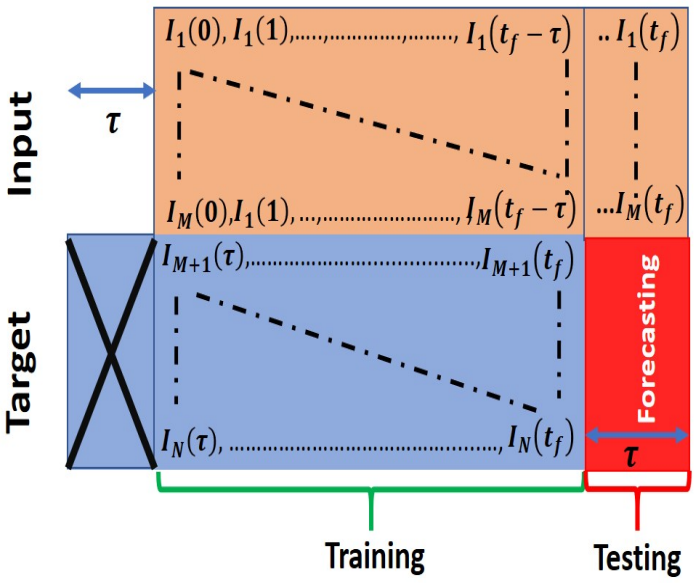

FIG. 5: Generalized approach for future forecasting of selected locations. This scheme enables to predict the future trend of infection of the target locations for a duration τ time unit.

here we aim to estimate the future trend of infection of the above mentioned target locations for a certain duration of time. Note that the initial growth rate for this type of the infection is slow (follows power law [41]) compared to the growth rate in later time. Thus we assume, ignoring initial data (of targeted locations) for few days will not affect the overall activities of ESN. Thus, we hypothesize that short amount of time-shifted of the input data can lead us to forecast the future trend (for short term $\sim$ 2 weeks) of infection of the target locations. To do this we use the following steps:

Spatial and temopral Decomposition. We collect the same data set from N locations. The data was saved from $t = 0$ to $t = t_f$ We decompose the data set into two parts: Input (M locations) and Target ($N - M$ locations,$N >> (N - M)$). We shift each of the inputs with τ ($\tau << t_f$) time unit, i.e., the input will be added to the machine at $t = \tau$ (light red rectangular regime of Fig. 5). As a consequence, we remove the initial trend of target locations upto τ time unit (blue rectangular regime with cross mark). We will continue this learning process until the entire target data is utilized for training purpose, 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 $t = t_f$.

Forecasting using testing procedure. Now we can use the trained machine to forecast the target data from t_f to $t_f + \tau$ (deep red regime) from the input data starting $t_f - \tau + 1$ to t_f (light red part in the right side). The green brace below the light blue matrix represents the training time and the red brace signifies the future forecasting of the target locations.

A. Forecasting future trend from COVID-19 data

To validate our modified scheme, we have The raw data [40], [https://covid19.who.int/] is preprocessed 465 days: (from 22 January 2020 to 30 April October, 2021). We decompose the entire set into two groups. Infected data of 241 locations are fed into the ESN at the time of training. We target to forecast of the infection trend of 10 locations. To forecast 14 days in future, we have discarded initial 14 days from the targeted data. We have trained the machine by utilizing the COVID-19 data from 22st January, 2021 to 17th April, 2020 (total 452 days) to track the target data from February 5, 2020 to April 30, 2021 (total 438 days). After training is finished, we forecast 14 days data of targeted locations from 1st may, 2021 to 14th may, 2021. Note that, we have data in hand until 30th April, 2021. However we can forecast for 14 days more from May 1 to May 14. The machine generated data is marked with grey dots (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 most of the states (Fig. 6 (f)-(j)) in India, the daily infection will increase except Uttar Pradesh (Fig. 6 i). The real data of each location in India is shown with red markers(from May 1 to May 6, 2021) which are closely matched with machine generated data. Trends in Italy and Germany are are weakly captured (Fig. 6 c,e) by the machine generated data where as for Israel slowly increasing (d). The upper bound of τ 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 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 we called them as target locations.

The proposed technique does not largely depend on the intrinsic parameters of the ESN. In the literature, there exist several phenomenological models [42–44] for predicting the trend of infection. However, these models have limitations for prediction due to intrinsic uncertainties in system parameters. For instance, the well known Gompertz function cannot capture the trend of the second wave [45–47] of infection whereas it can efficiently predict the initial daily infection. Also, suitable choices of parameters of Gompertz function

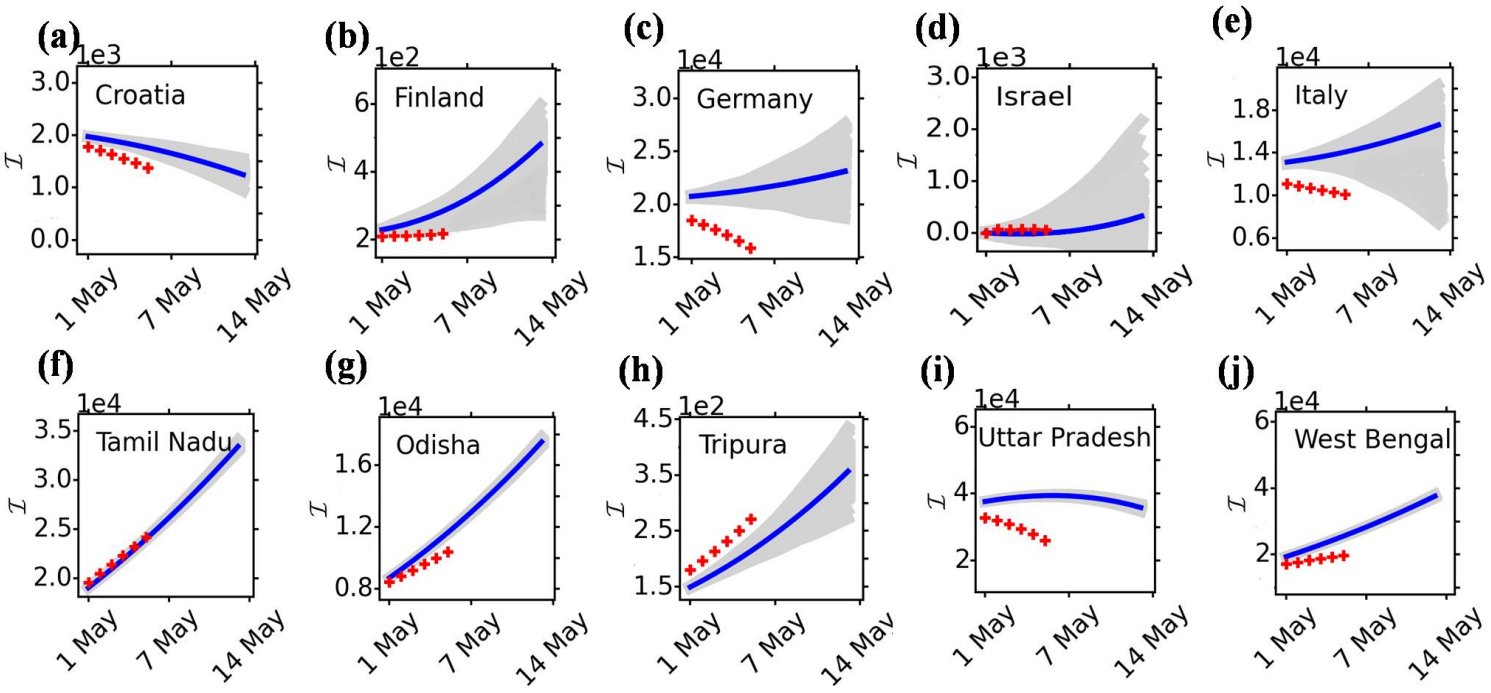

FIG. 6: Forecasting future trend of infection of 10 selected locations from May 1, 2021 to May 14, 2021. The grey patch in each figure is the machine generated data for 100 realizations. The blue line is the average of each grey patch. The red markers are the real data for the period 1st May, to 6th May, 2021.

immediate before the prediction are necessary (please see the appendix Sec. A for detailed investigation through Gompertz function). In our model-free machine learning scheme this restriction is relaxed as ESN can successfully trace the second wave of specific locations (See Fig. 4 (b-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 of future forecasting. We expect our proposition might be useful for diverse set of spatiotemporal data ranging from physiological to multivariate climate data. In a same manner, we can use other types of recurrent neural networks for prediction of infection trend of certain locations that we intend to try in future.

Codes to reproduce the results presented here are freely accessible at Reservoir-computing-on-epidemic-spreading.

Acknowledgments: C.H. and S.G. are supported by the INSPIRE-Faculty grant (code: IFA17-PH193). J.C. is supported by Technology Innovation Hub on Data Science, Big Data Analytics and Data Curation (Grant No: NMICPS/006/MD/2020-21, dt. 16/10/2020).

VII. APPENDIX

A. Prediction through Gompertz Curve

In literature, there are so many models and mechanism available to data fitting which enables us to estimate suitable parameters for short term forecasts as well as its uncertainty in forecasting. For instance, generalized Richard model, logistic growth model, sub-epidemic wave model [42, 43] flexible growth model curve[44] and Gompertz curve [46, 47] have been thoroughly used for forecasting infection data. We use the following Gompertz function [45] for capturing the daily infection ($\mathcal{I}(t)$)

$$\mathcal{I}(t) = aKe^{-\ln(\frac{K}{N_0})e^{-at}}(\ln(\frac{K}{N_0})e^{-at}), \qquad (7)$$

where the parameter K is the saturating value of the infected cases, N_0 is the initial infection, and a represents the decreasing trend of the initial exponential growth. Now we estimate the parameters a and K to predict the infection pattern from 5th October to 26th October, 2020 (22 days). Here we take daily infection data of 10 countries/regions for 4 weeks from 8th Sept to October 5, 2020 and fit it with Gompertz curve (GC), to obtain the best fitted parameters. We can see that Gompertz curve is able to provide good prediction for certain locations (Fig. 7, Israel, Uttarpradesh, Telengana, Tripura, and Finalnd) and fails to predict the infection trend in

Croatia, Germany, West Bengal, Rajasthan, and Italy. However, machine generated data performs well for most of the cases (see Fig. 4 for comparison).

[1] M. Perc, N. Gorišek Miksić, M. Slavinec, and A. Stožer, Frontiers in Physics **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.*, The Lancet Infectious Diseases **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.*, Nature Reviews Physics **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.*, The Lancet Global Health **8**, e488 (2020).
[8] T. Colbourn, The Lancet Public Health **5**, e236 (2020).
[9] S. Ghosh, A. Senapati, J. Chattopadhyay, C. Hens, and D. Ghosh, arXiv preprint arXiv:2010.07649 (2020).
[10] D. L. Heymann and N. Shindo, The Lancet **395**, 542 (2020).
[11] J. Tsai and M. Wilson, The Lancet Public Health **5**, e186 (2020).
[12] A. A. AlMomani and E. Bollt, arXiv preprint arXiv:2004.08897 (2020).
[13] L. Gallo, M. Frasca, V. Latora, and G. Russo, arXiv preprint arXiv:2012.00443 (2020).
[14] S. Lalmuanawma, J. Hussain, and L. Chhakchhuak, Chaos, Solitons & Fractals **139**, 110059 (2020).
[15] A. Senapati, S. Rana, T. Das, and J. Chattopadhyay, Journal of Theoretical Biology **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.*, Journal of Thoracic Disease **12**, 165 (2020).
[17] Z. Li, Y. Zheng, J. Xin, and G. Zhou, arXiv preprint arXiv:2007.10929 (2020).
[18] A. Fokas, N. Dikaios, and G. Kastis, Journal of the Royal Society 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.*, Nature Machine Intelligence **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. Wieczorek, J. Siłka, 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.*, Nutrition, Metabolism and Cardiovascular Diseases **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, Physical Review Letters **120**, 024102 (2018).
[27] R. S. Zimmermann and U. Parlitz, Chaos: An Interdisciplinary Journal of Nonlinear Science **28**, 043118 (2018).
[28] J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott, Chaos: An Interdisciplinary Journal of Nonlinear Science **27**, 121102 (2017).
[29] Z. Lu, B. R. Hunt, and E. Ott, Chaos: An Interdisciplinary Journal of Nonlinear Science **28**, 061104 (2018).
[30] X. Lin, Z. Yang, and Y. Song, Expert Systems with Applications **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, Information Processing Letters **95**, 521 (2005).
[33] T. Weng, H. Yang, C. Gu, J. Zhang, and M. Small, Physical Review E **99**, 042203 (2019).
[34] T. Lymburn, D. M. Walker, M. Small, and T. Jüngling, Chaos: An Interdisciplinary Journal of Nonlinear Science **29**, 093133 (2019).
[35] X. Chen, T. Weng, H. Yang, C. Gu, J. Zhang, and M. Small, Physical Review E **102**, 033314 (2020).
[36] A. Panday, W. S. Lee, S. Dutta, and S. Jalan, Chaos: An Interdisciplinary Journal of Nonlinear Science **31**, 031106 (2021).
[37] S. Saha, A. Mishra, S. Ghosh, S. K. Dana, and C. Hens, Physical Review Research **2**, 033338 (2020).
[38] J. Hilton and M. J. Keeling, PLoS Computational Biology **16**, 1 (2020).
[39] M. Castro, S. Ares, J. A. Cuesta, and S. Manrubia, Proceedings of the National Academy of Sciences **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.*, The Lancet Infectious Diseases **20**, 534 (2020).
[41] B. F. Maier and D. Brockmann, Science **368**, 742 (2020).
[42] A. Smirnova and G. Chowell, Infectious Disease Modelling **2**, 268 (2017).
[43] K. Roosa, Y. Lee, R. Luo, A. Kirpich, R. Rothenberg, J. M. Hyman, P. Yan, and G. Chowell, Journal of clinical medicine **9**, 596 (2020).
[44] C. F. Tovissodé, B. E. Lokonon, and R. Glèlè Kakaï, Plos One **15**, e0240578 (2020).
[45] M. Català, S. Alonso, E. Alvarez-Lacalle, D. López, P.-J. Cardona, and C. Prats, PLoS Computational Biology **16**, e1008431 (2020).
[46] Á. Berihuete, M. Sánchez-Sánchez, and A. Suárez-Llorens, Mathematics **9**, 228 (2021).
[47] A. Ohnishi, Y. Namekawa, and T. Fukui, Progress of Theoretical and Experimental Physics **2020**, 123J01 (2020).

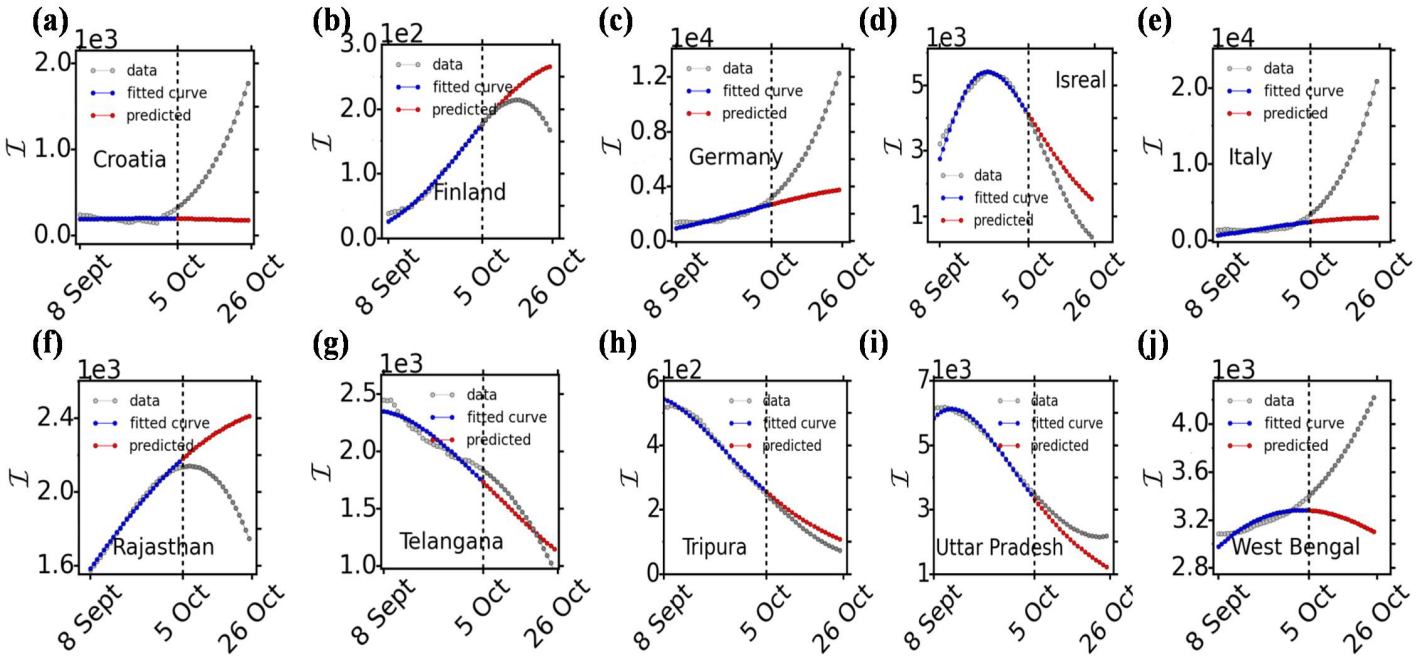

FIG. 7: Forecasting through the Gompertz Curve (GC) in 10 randomly chosen regions. 4 weeks data are used to standardize the intrinsic parameters of Gompertz function. 22 days were forecast from October 5, to October 26, 2020.