Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Neurocomputing 470 (2022) 11–28

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Digital twins based on bidirectional LSTM and GAN for modelling the COVID-19 pandemic

César Quilodrán-Casas [a,b,*], Vinicius L.S. Silva [b], Rossella Arcucci [a,b], Claire E. Heaney [b], YiKe Guo [a], Christopher C. Pain [a,b]

[a] Data Science Institute, Department of Computing, Imperial College London, UK
[b] Department of Earth Science & Engineering, Imperial College London, UK

ARTICLE INFO

Article history:
Received 11 February 2021
Revised 4 August 2021
Accepted 19 October 2021
Available online 22 October 2021
Communicated by Zidong Wang

Keywords:
Reduced order models
Digital twins
Deep learning
Long short-term memory networks
Generative adversarial networks

ABSTRACT

The outbreak of the coronavirus disease 2019 (COVID-19) has now spread throughout the globe infecting over 150 million people and causing the death of over 3.2 million people. Thus, there is an urgent need to study the dynamics of epidemiological models to gain a better understanding of how such diseases spread. While epidemiological models can be computationally expensive, recent advances in machine learning techniques have given rise to neural networks with the ability to learn and predict complex dynamics at reduced computational costs. Here we introduce two digital twins of a SEIRS model applied to an idealised town. The SEIRS model has been modified to take account of spatial variation and, where possible, the model parameters are based on official virus spreading data from the UK. We compare predictions from one digital twin based on a data-corrected Bidirectional Long Short-Term Memory network with predictions from another digital twin based on a predictive Generative Adversarial Network. The predictions given by these two frameworks are accurate when compared to the original SEIRS model data.

Additionally, these frameworks are data-agnostic and could be applied to towns, idealised or real, in the UK or in other countries. Also, more compartments could be included in the SEIRS model, in order to study more realistic epidemiological behaviour.

1. Introduction

The outbreak of the coronavirus disease 2019 (COVID-19) has now spread throughout the world, infecting over 153 million reported individuals as of May 4th 2021 [1]. Globally, at least 3.2 million deaths have been directly attributed to COVID-19 [1] and this number continues to rise. There is a lack of information and uncertainty about the dynamics of this outbreak, thus, there is an urgent need for research in this field to help with the mitigation of this pandemic [2]. Agent-based models [3–5] and SEIR-type models [6,7] have been widely used to study epidemiological problems. However, when modelling complex scenarios, these models can become computationally expensive, e.g. such models may have many millions of degrees of freedom that must be solved at every time step [8,9]. Also, the time steps may be small to resolve the transport of people around a domain. For instance, in a model of a town, a person in a car or train may travel large distances in just a few minutes [10]. This advection can have limitations in terms of Courant number restrictions [11] based on the spatial resolution, as well as the speed of the transport. Furthermore, these models may have a set of variables for each member of a population. Thus, if a country is modelled with many millions of people, the computational expense of such models becomes an issue and they may even become intractable [12]. This has motivated the current research on digital twins or reduced-order Models (ROMs) for virus modelling. Although ROMs have been developed in fields such as fluid dynamics, they are new for virus modelling. For this new application area, we study a simple test case to try to understand the application of these methods to virus modelling. The prize of an accurate and fast ROM means that it may be readily used, possibly interactively, to explore different control measures, to assimilate data into the models, and to help determine the spatial and future temporal variation of infections. We may need to develop new ROM approaches to meet the demands of this new virus application area and explore the relative merits of existing and new ROM approaches which is the focus of this paper.

In this paper, we compare the performance of two digital twins, also known as non-intrusive reduced-order models (NIROMs). The

* Corresponding author.
E-mail address: cesar.quilodran-casas13@imperial.ac.uk (C. Quilodrán-Casas).

https://doi.org/10.1016/j.neucom.2021.10.043
0925-2312/

first of these uses Principal Component Analysis (PCA) for the dimensionality reduction step and a data-corrected (via optimal interpolation) Bidirectional Long Short-term memory network (BDLSTM) for the prediction or interpolation step. The second NIROM, again, uses PCA for the dimensionality reduction step, but uses a Generative Adversarial Network (GAN) for the prediction step. The test case used for the comparison of these so-called digital twins or NIROMs is the spread of COVID-19 in an idealised town. This is modelled by the SEIRS equations [13,14] that have been extended to take account of spatial variation. The NIROMs are used to approximate future states of the system which are compared against the ground truth.

NIROMs have been used with success in several fields, to speed up computational models without losing the resolution of the original model [15–18] and without the need to make changes to the code of the high-fidelity model. Typically, the first stage in constructing a NIROM is to reduce the dimension of the problem by using compression methods such as PCA [19], autoencoders, or a combination of both [20–22]. Solutions from the original computational model (known as snapshots) are then projected onto the lower-dimensional space, and the resulting snapshot coefficients are interpolated in some way, to approximate the evolution of the model. This interpolation, which approximates unseen states of the model, constitutes the second stage of the NIROM. Originally, classical interpolation methods were used, such as cubic interpolation [23], radial basis functions [24,25] and Kriging [26]. Recently, non-intrusive reduced-order methods (sometimes referred to as model identification methods [27,28] or described by the more general term of digital twins [29–31]) have taken advantage of machine learning techniques, using multi-layer perceptrons [16], cluster analysis [27], LSTMs [28,32–34] and Gaussian Process Regression [35]. In this work, we use PCA (also known as Proper Orthogonal Decomposition) to reduce the dimension of the original system, and for the interpolation or prediction, we compare a data-corrected BDLSTM with a predictive GAN. The LSTM network, originally described in [36], is a special kind of recurrent neural network (RNN) that is stable, powerful enough to be able to model long-range time dependencies [37] and overcomes the vanishing gradient problem [38]. A further development was made to this network inspired by bidirectional RNNs [39], in which sequences of data are processed in both forward and backward directions. The resulting BDLSTMs have been proven to be better [40] than unidirectional ones, as the former can capture the forward and backward temporal dependencies in spatiotemporal data [41], in many fields such as speech recognition [42] and traffic control [40]. Bidirectional LSTMs have also been used in text classification [43], predicting efficient remaining useful life of a system [44], and urban air pollution forecasts [34]. LSTMs are widely recognised as one of the most effective sequential models [45] for times series predictions. We compare the performance of LSTMs with GANs [46] which are known for retaining realism. GANs have shown impressive performance for photo-realistic high-quality images of faces [47,48]; image to image translation [49]; synthetical medical augmentation [50]; cartoon image generation [51], amongst others. The basic idea of GANs is to simultaneously train a discriminator and a generator, where the discriminator aims to distinguish between real samples and generated samples; while the generator tries to fool the discriminator by creating fake samples that are as realistic as possible. The GAN is a generative model and its use in making predictions in time is a recent development [52]. By learning a distribution that fits the training data, the aim is that new samples, taken from the learned distribution formed by the generator, will remain 'realistic' over time and will not diverge.

Although non-intrusive reduced-order modelling has not been applied to epidemiological problems, as far as we are aware, neural networks have been used to model the spread of viruses. Previous studies have used Long Short-term Memory networks for COVID-19 predictions: Modified SEIR predictions of the trend of the epidemic in China [53], general outbreak prediction with machine learning [54], time series forecasting of COVID-19 transmission in Canada [55], and predicting COVID-19 incidence in Iran [56], amongst others. Generative networks have also been used to model aspects of the COVID-19 outbreak, mainly used in image recognition, e.g. chest X-rays [57,58]. Bayesian updating has also been applied to COVID-19 by Wang et al. [59]. Furthermore, [60] used spatio-temporal conditional GANs for estimating the human mobility response to COVID-19.

The advantages of using an LSTM and GAN are that the former is an effective sequential model [45] and the latter can learn the underlying data distribution, reducing the forecast divergence [61]. Whilst LSTM and GAN have been used to study COVID-19, the datasets employed in the aforementioned studies differ from our approach since 1) our approach uses a spatio-temporal dataset, rather than scalar quantities that evolve in time; 2) the dimensionality of the spatio-temporal model is reduced during the compression stage of NIROM. The NIROM transforms the spatio-temporal problem into a multivariate time-series problem.

The novelty of this paper lies in the use of data-corrected forecasts with the state-of-the-art LSTM. This network is used for the prediction step of a NIROM or digital twin, and results of this model are compared with a NIROM based on GAN methods that are novel for prediction in time. In summary, the main novelties and contributions of this paper are:

- The application of reduced-order modelling to virus/epidemiology modelling.
- The application of the novel data-corrected BDLSTM-based NIROM approach. This is the first time that the data-corrected BDLSTM has been incorporated within a NIROM. Using data from the solutions of the extended SEIRS equations, optimal interpolation is included in the prediction-correction cycle of the BDLSTM to stabilise the forecast and to achieve improved accuracy.
- Comparison is made between time-series predictions of two digital twins: one based on the state-of-the-art LSTM and the other based on a GAN, a recent network that is known for its realistic predictions. The GAN can generate time sequences from random noise that are constrained to generate a forecast. Both NIROMs use PCA for the compression step.

The structure of this paper is as follows. Section 2 introduces the classical SEIRS model and the extended SEIRS model, which takes account of spatial variation. The SEIRS model in this paper also includes an additional way of categorising people according to their environment. Section 3 presents the methodology of the two digital twins (based on results from the extended SEIRS model) and explains how the predictions are performed. The results and the discussion of these experiments are presented in Sections 4 and 5. Finally, conclusions and future work are discussed in Section 6.

2. SEIRS model

2.1. Classical SEIRS model

The SEIRS equations that govern virus infection dynamics categorise the population into four compartments: Susceptible, Exposed, Infectious or Recovered. See Fig. 1 for an illustration of the rates that control how a person moves between these compartments. The infection rate, β, controls the rate of spread which

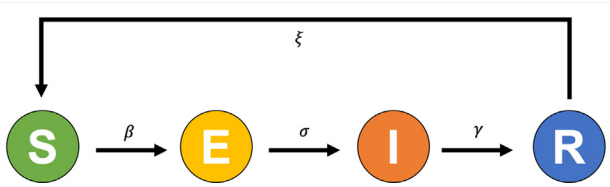


Fig. 1. Key variables and parameters in the SEIRS model representing the compartments Susceptible (S), Exposed (E), Infectious (I), and Recovered (R). Modified from [62].

represents the probability of transmitting disease between a susceptible and an exposed individual (someone who has been infected but is not yet infectious). The incubation rate, σ, is the rate of exposed individuals becoming infectious (the average duration of incubation is $1/\sigma$). The recovery rate, $\gamma = 1/T_D$, is determined by the average duration, T_D, of infection. For the SEIRS model, ξ is the rate at which recovered individuals return to the susceptible state due to loss of immunity.

Vital dynamics can be added to a SEIRS model, by including birth and death rates represented by μ and ν, respectively. To maintain a constant population, one can make the assumption that $\mu = \nu$, however, in the general case, the system of ordinary differential equations can be written:

$$\frac{\partial S}{\partial t} = \mu N - \frac{\beta S I}{N} + \xi R - \nu S, \tag{1a}$$

$$\frac{\partial E}{\partial t} = \frac{\beta S I}{N} - \sigma E - \nu E, \tag{1b}$$

$$\frac{\partial I}{\partial t} = \sigma E - \gamma I - \nu I, \tag{1c}$$

$$\frac{\partial R}{\partial t} = \gamma I - \xi R - \nu R \tag{1d}$$

where $S(t), E(t), I(t)$ and $R(t)$ represent the number of individuals in the susceptible, exposed (infected but not yet infectious), infectious and recovered compartments respectively. At time t, the total number of individuals in the population under consideration is given by $N(t) = S(t) + E(t) + I(t) + R(t)$. If the birth and death rates are the same, N remains constant over time.

2.2. Extended SEIRS model

In this study, the SEIRS model is extended in two ways. First, we introduce diffusion terms to govern how people move throughout the domain, thereby incorporating spatial variation into the model. Second, we associate a group with each person, indicated by the index $h \in \{1, 2, \ldots, \mathscr{H}\}$. This indicates the person has gone to work or school, gone shopping, gone to a park or stayed at home, for example, and transmission rates for each group can be set according to the risk of being in offices, schools, shopping centres, outside, or at home. These modifications to the SEIRS equations result in the following system of equations:

$$\frac{\partial S_h}{\partial t} = \mu_h N_h - \frac{S_h \sum_{h'} (\beta_{h h'} I_{h'})}{N_h} + \xi_h R_h - \nu_h^S S_h - \sum_{h'=1}^{\mathscr{H}} \lambda_{hh'}^S S_{h'} + \nabla \cdot \left(k_h^S \nabla S_h \right), \tag{2a}$$

$$\frac{\partial E_h}{\partial t} = \frac{S_h \sum_{h'} (\beta_{h h'} I_{h'})}{N_h} - \sigma E_h - \nu_h^E E_h - \sum_{h'=1}^{\mathscr{H}} \lambda_{hh'}^E E_{h'} + \nabla \cdot \left(k_h^E \nabla E_h \right), \tag{2b}$$

$$\frac{\partial I_h}{\partial t} = \sigma E_h - \gamma_h I_h - \nu_h^I I_h - \sum_{h'=1}^{\mathscr{H}} \lambda_{hh'}^I I_{h'} + \nabla \cdot \left(k_h^I \nabla I_h \right), \tag{2c}$$

$$\frac{\partial R_h}{\partial t} = \gamma_h I_h - \xi_h R_h - \nu_h^R R_h - \sum_{h'=1}^{\mathscr{H}} \lambda_{hh'}^R R_{h'} + \nabla \cdot \left(k_h^R \nabla R_h \right), \tag{2d}$$

in which the subscript h represents which group an individual is associated with. Instead of having scalar values for each compartment, we now have fields: $S_h(\boldsymbol{\omega}, t), E_h(\boldsymbol{\omega}, t), I_h(\boldsymbol{\omega}, t)$ and $R_h(\boldsymbol{\omega}, t)$, where the people associated with group h for the susceptible, exposed, infectious and recovered compartments, respectively, vary in space, $\boldsymbol{\omega}$, and time, t. The transmission terms $\beta_{hh'}$ govern how the disease is transmitted from people in groups $h' \in \{1, 2, \ldots, \mathscr{H}\}$ to people in group h. The terms involving $\lambda_{hh'}^{(\cdot)}$ are interaction terms that control how people move between the groups describing the various locations/activities for the compartment given in the superscript. These values could, for example, control whether people in the school group move into the home group. When moving from one group to another, the individual remains in the same compartment. Describing the spatial variation, the diffusion coefficients for each compartment are given by $k_h^{(\cdot)}$. The birth rate for a group is μ_h and the death rate is set for each compartment and group, where, for example, ν_h^S is the death rate of group h for the susceptible compartment. The term σ represents the rate at which some of the people in the exposed compartment, E, transfer to the infectious compartment, I. The recovery rate is now:

$$\gamma_h = \frac{1}{T_{D_h}}, \tag{3}$$

in which T_{D_h} are the average durations of infections in infection groups I_h. Therefore the infectious rates become:

$$\beta_{hh} = \gamma_h R0_h, \quad h \in \{1, 2, \ldots \mathscr{H}\}. \tag{4}$$

Here we assume $\beta_{hh'} = 0$ when $h \neq h'$. This assumption means that a person in the Home group cannot infect someone in the Mobile group (as the former will be at home and the latter will be outside of the home), and, likewise, a person in the Mobile group cannot infect someone in the Home group.

An eigenvalue problem can be formed by placing an eigenvalue, λ_0, in front of the terms σE_h in Eqs. (2b) and (2c), and by setting all four time derivatives to zero in Equations (2). In addition, this term will need to be linearised. To model the beginning of the virus outbreak, a possible way of linearising is shown here:

$$\frac{S_h^g \sum_{h'} (\beta_{h h'} I_{h'})}{N_h} \approx \sum_{h'} (\beta_{h h'} I_{h'}), \quad \forall h \in \{1, 2, \ldots, \mathscr{H}\}. \tag{5}$$

The eigenvalue is equivalent to the reciprocal of $R0$, that is $R0 = \frac{1}{\lambda_0}$.

We remark that the system of Equations (2) is similar to the neutron transport equations and comment that codes written to solve nuclear engineering problems could be reapplied to virus modelling without much modification.

2.3. Extended SEIRS model for two groups

As said in the introduction, the area of reduced-order modelling is new to virus modelling, so we choose a simple test case to try to understand the application of these methods to virus modelling. In this paper, we restrict ourselves to the specific case where there are two possible and distinct groups in addition to the SEIRS compartments. The groups comprise people who remain at home ('Home', H), and others who are mobile and can move to riskier surroundings ('Mobile', M). The index representing the group, h,

has therefore two values: $h \in \{H, M\}$. For this case, the transmission terms between Home and Mobile must be zero, so $\beta_{HM} = 0$ and $\beta_{MH} = 0$. This is because an individual at Home cannot infect someone in the Mobile group and vice versa as they will not be near one another. We wish interaction terms $\lambda^{(\cdot)}_{hh'}$, which control how people move from Home to Mobile groups and vice versa, to be such that conservation is obeyed. In other words, the number of people leaving the Home group (for a given compartment) must equal the number of people entering the Mobile group (for that compartment). On inspection of Eq. (2a), for group $h = H$, we can see that people moving between the Home and Mobile groups in the susceptible compartment will be $-\lambda^S_{HH} S_H - \lambda^S_{HM} S_M$. From Eq. (2a), for group $h = M$, people moving between the Home and Mobile groups in the susceptible compartment is given by the terms $-\lambda^S_{MM} S_M - \lambda^S_{MH} S_H$. To enforce that the number of people leaving S_H is equal to the number of people joining S_M, the interaction coefficients can be set as follows:

$$\lambda^S_{HH} = -\lambda^S_{MH}, \quad \lambda^S_{MM} = -\lambda^S_{HM} \quad \text{and} \quad \lambda^S_{HH} = \lambda^S_{MM}. \tag{6}$$

Suppose $\lambda^S_{HH} =: \tilde{\lambda}^S$, then we can say that the number of people leaving S_M (joining if $\tilde{\lambda}^S < 0$) is $\tilde{\lambda}^S(S_H - S_M)$ and the number of people joining S_H (leaving if $\tilde{\lambda}^S < 0$) is $\tilde{\lambda}^S(S_H - S_M)$. Similar relationships hold for the other three compartments, i.e. replace the superscript S in Eqs. (6) with E, I and R in turn. See Fig. 2 for an illustration of how people move between compartments and groups in this extended SEIRS model. Rădulescu [7] uses a similar approach to model a small college-town which has seven locations (medical centre, shops, university campus, schools, parks, bars and churches) all with appropriate transmission rates.

The spatial variation is discretised on a regular grid of $N_X \times N_Y \times N_Z$ control volume cells. The point equations can be recovered by choosing $N_X = N_Y = N_Z = 1$. We use a 5 point stencil and second-order differencing of the diffusion operator, as well as backward Euler time stepping. We iterate within a time step, using Picard iteration, until convergence of all nonlinear terms and evaluate these nonlinear terms at the future time level. To solve the linear system of equations we use Forward Backward Gauss-Seidel (FBGS) for each variable in turn, and once convergence has been achieved, Block FBGS is used to obtain overall convergence of the eight linear solutions. This simple solver is sufficient to solve the relatively small problems presented here.

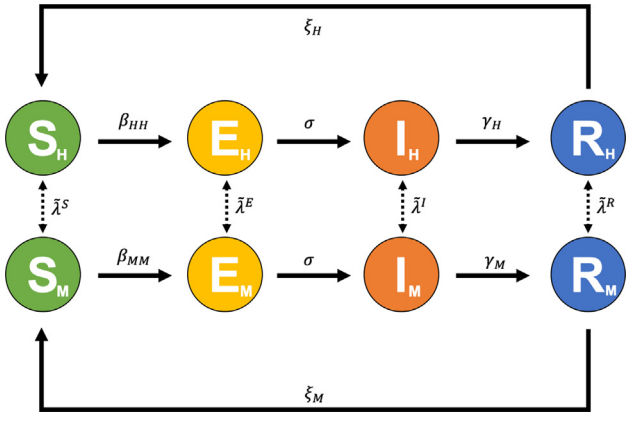


Fig. 2. Movement between compartments Susceptible (S), Exposed (E), Infectious (I) and Recovered (R), and groups Home (H) and Mobile (M) for the extended SEIRS model. The spatial variation is not represented here, just movement between compartments and groups. The movement between home and mobile groups is defined by $\tilde{\lambda}^{(\cdot)}$.

The parameters $\beta_{hh'}, \sigma, \gamma_h$ and ξ_h, were chosen based on parameters observed in the UK, similar to Nadler et al. [63] who also estimated the parameters from data, this time for the SIR equations. According to the [64], the incubation period is between 1 and 14 days, with a median of 5 days. Here, an incubation rate of 4.5 days is used, which is within the range of observed COVID-19 incubation periods in the UK. The SEIRS model presented here is flexible, however, meaning that it could be applied to other regions with different parameters. More details about the configuration of the SEIRS model and the results obtained are given in Section 4.1.

3. Methods

In this section the two NIROMs or digital twins are described in detail. Non-intrusive reduced-order modelling contains two steps: dimensionality reduction and prediction. Both models use PCA for the dimensionality reduction step as is common in reduced-order models. For the prediction step, one NIROM uses a Bidirectional LSTM, the other uses a GAN. Both methods require an inverse PCA transform to obtain predicted solutions from the NIROMs of the extended SEIRS model. This section concentrates on explaining use of the Bidirectional LSTM and the GAN for making predictions in time.

3.1. Bidirectional long short-term memory networks

The LSTM network comprises three gates: input ($\mathbf{i}_{\mathbf{t_k}}$), forget ($\mathbf{f}_{\mathbf{t_k}}$), and output ($\mathbf{o}_{\mathbf{t_k}}$); a block input, a single cell $\mathbf{c}_{\mathbf{t_k}}$, and an output activation function. This network is recurrently connected back to the input and the three gates. Due to the gated structured and the forget state, the LSTM is an effective and scalable model that can deal with long-term dependencies [36]. The vector equations for a LSTM layer are:

$$\begin{aligned}
\mathbf{i}_{t_k} &= \phi\left(\mathbf{W}_{xi}\mathbf{x}_{\mathbf{t_k}} + \mathbf{W}_{Hi}\mathbf{H}_{t_{k-1}} + \mathbf{b}_i\right) \\
\mathbf{f}_{t_k} &= \phi\left(\mathbf{W}_{xf}\mathbf{x}_{\mathbf{t_k}} + \mathbf{W}_{Hf}\mathbf{H}_{t_{k-1}} + \mathbf{b}_f\right) \\
\mathbf{o}_{t_k} &= \phi\left(\mathbf{W}_{xo}\mathbf{x}_{\mathbf{t_k}} + \mathbf{W}_{Ho}\mathbf{H}_{t_{k-1}} + \mathbf{b}_o\right) \\
\mathbf{c}_{t_k} &= \mathbf{f}_{t_k} \circ \mathbf{c}_{t_{k-1}} + \mathbf{i}_{t_k} \circ \tanh\left(\mathbf{W}_{xc}\mathbf{x}_{\mathbf{t_k}} + \mathbf{W}_{Hc}\mathbf{H}_{t_{k-1}} + \mathbf{b}_c\right) \\
\mathbf{H}_{t_k} &= \mathbf{o}_{t_k} \circ \tanh\left(\mathbf{c}_{t_k}\right)
\end{aligned} \tag{7}$$

where ϕ is the sigmoid function, $\mathbf{W}$ are the weights, $\mathbf{b}_{i,f,o,c}$ are the biases for the input, forget, output gate and the cell, respectively, $\mathbf{x}_{\mathbf{t_k}}$ is the layer input, $\mathbf{H}_{\mathbf{t_k}}$ is the layer output and $\circ$ denotes the entry-wise multiplication of two vectors.

The idea of BDLSTMs comes from bidirectional RNN [39], in which sequences of data are processed in both forward and backward directions with two separate hidden layers. BDLSTMs connect the two hidden layers to the same output layer. The forward layer output sequence is iteratively calculated using inputs in a forward sequence, $\overrightarrow{\mathbf{H}}_{\mathbf{t_k}}$, from time t_{k-n} to t_{k-1}, and the backward layer output sequence, $\overleftarrow{\mathbf{H}}_{\mathbf{t_k}}$, is calculated using the reversed inputs from t_{k-1} to t_{k-n}. The layer outputs of both sequences are calculated by using the equations in (7). The BDLSTM layer generates an output vector $\mathbf{u}_{\mathbf{t_k}}$:

$$\mathbf{u}_{\mathbf{t_k}} = \psi\left(\overrightarrow{\mathbf{H}}_{\mathbf{t_k}}, \overleftarrow{\mathbf{H}}_{\mathbf{t_k}}\right) \tag{8}$$

where ψ is a concatenating function that combines the two output sequences.

3.1.1. Prediction with BDLSTM

The prediction workflow of the BDLSTM + BLUE-based NIROM is presented in Fig. 3 and includes the dimensionality reduction step

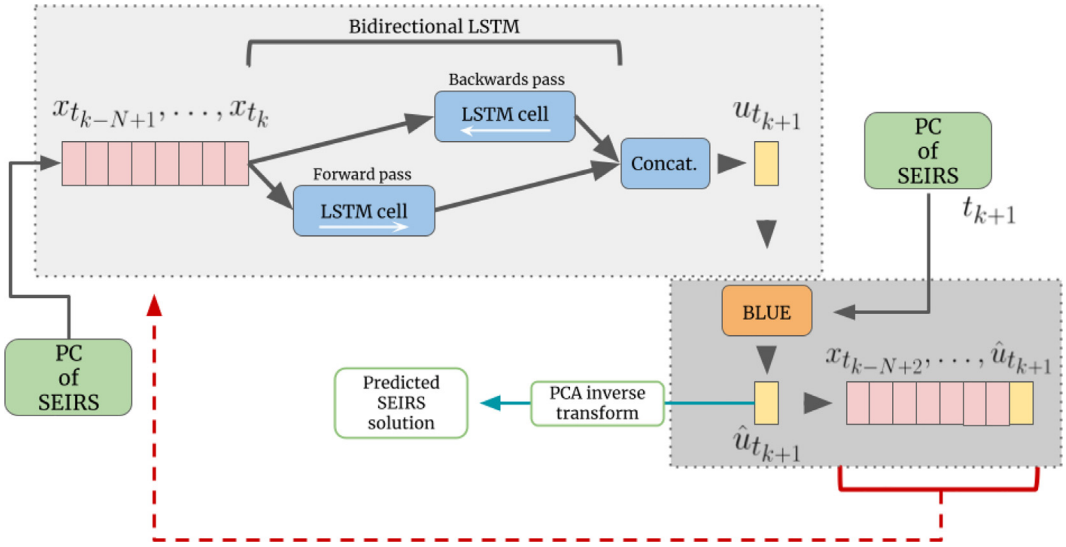


Fig. 3. BDLSTM + BLUE-based NIROM for a sequence of two time levels. Top-left: off-line bidirectional LSTM network. Bottom-right: data-correction of the prediction. The Best Linear Unbiased Estimation (BLUE) is used to data-correct the prediction of the network. One time level corresponds to 10 time-steps of the original SEIRS solution.

with PCA as well as the BDLSTM. While LSTMs are known for producing time-series predictions, the workflow introduces a data-corrected step using the Best Linear Unbiased Estimator (BLUE). This step improves the accuracy of those predictions. The BDLSTM network f^{BDLSTM} is a function trained off-line to predict t_{k+1} given the previous N time-levels from the latent vector $\mathbf{x}$:

$$f^{BDLSTM} : \mathbf{x}_{t_{k-N+1}}, \ldots, \mathbf{x}_{\mathbf{t_k}} \rightarrow \tilde{\mathbf{u}}_{t_{k+1}}. \tag{9}$$

Once the network is able to predict the solution $\tilde{\mathbf{u}}_{t_{k+1}}$, this is joined to the solutions at $u_{t_{k-N}}, u_{t_{k-N+1}}, \ldots, u_{t_k}$, to create $\mathbf{u}_p$. The prediction vector $\mathbf{u}_p$ is then optimised online using BLUE:

$$\hat{\mathbf{u}}_p = \bar{\mathbf{u}}_p + \mathbf{C}_{\mathbf{u}_p\mathbf{v}}\mathbf{C}^{-1}(\mathbf{v} - \bar{\mathbf{v}}) \tag{10}$$

where $\hat{\mathbf{u}}_p$ is the data-corrected prediction, $\bar{\mathbf{u}}_p$ is the mean of the vector $\mathbf{u}_p$ over time, $\mathbf{v}$ and $\bar{\mathbf{v}}$ are the observations and mean of the observations over time, respectively, $\mathbf{C}_{\mathbf{u}_p\mathbf{v}}$ is the covariance between $\mathbf{u}_p$ and observations $\mathbf{v}$, and $\mathbf{C}$ is the covariance of the observations. The first entry of $\mathbf{u}_p$ is dropped and the new vector is used to make a prediction of t_{k+2}. This is an iterative process. Thus, the data-corrected BDLSTM is defined by:

$$f^{BDLSTM+BLUE} : \mathbf{x}_{t_{k-N+1}}, \ldots, \mathbf{x}_{\mathbf{t_k}} \rightarrow \hat{\mathbf{u}}_p \tag{11}$$

In the prediction with the BDLSTM workflow, before performing a PCA on the original dataset, we normalised the values of each compartment by their corresponding means and standard deviation. This step was not done for the predictive GAN.

3.2. Generative adversarial network

Proposed by Goodfellow et al. [46], Generative Adversarial Networks (GANs), are unsupervised learning algorithms capable of learning dense representations of the input data and are intended to be used as a generative model, i.e. they are capable of learning the distribution underlying the training dataset and able to generate new samples from this distribution. The training of the GAN is based on a game theory scenario in which the generator network G must compete against an adversary. The generator network G directly produces time-sequences from a random distribution as input (latent vector $\mathbf{z}$):

$$G : \mathbf{z} \sim \mathcal{N}(0, I_L) \rightarrow \mathbf{y}_{GAN} \in \mathbb{R}^{N \times M} \tag{12}$$

where $\mathbf{y}_{GAN}$ is an array of N time sequences with M dimensions, L is the size of the latent vector, and I_L is an identity matrix of size L. The discriminator network D attempts to distinguish between samples drawn from the training data and samples drawn from the generator, considered as fake. The output of the discriminator $D(\mathbf{y})$ represents the probability that a sample came from the data rather than a "fake" sample from the generator, and the vector $\mathbf{y}$ represents "real" samples of the principal components from the ROM. The output of the generator $G(\mathbf{z})$ is a sample from the distribution learned from the dataset. Eqs. (13) and (14) show the loss functions of the discriminator and generator, respectively:

$$L_D = -\mathbb{E}_{y \sim p_{data}(y)}[\log(D(\mathbf{y}))] - \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(\mathbf{z})))] \tag{13}$$

$$L_G = -\mathbb{E}_{z \sim p_z(z)}[\log(D(G(\mathbf{z})))] \tag{14}$$

In this work, the generator is trained using Eq. (14). During the training process the latent space $\mathbf{z}$ is generated as a Gaussian random noise, as in Eq. (12). The discriminator is trained using Eq. (13). The loss function of the discriminator takes as inputs: a time sequence of compressed states from the extended SEIRS simulation ("real" sample), and a time sequence of compressed states generated by the generator ("fake" sample). After training, the discriminator can be discarded since only the generator is used during the prediction process.

3.2.1. Predictions with GAN

To make predictions in time using a GAN, an algorithm named Predictive GAN [52] is used. The network is trained to generate data at a sequence of N time levels from $t_{k-N+1}, \ldots, t_k$ no matter at which point in time k is. In other words, the network will generate data that represents the dynamics of N consecutive time levels. Following that, given known solutions from time levels t_{k-N+1} to t_{k-1}, the input of the generator $\mathbf{z}$ can be optimised to produce solutions at time levels t_{k-N+1} to t_k. Hence the new prediction is the solution at time t_k. To predict the next time level, having known solutions at t_{k-N+2} to t_{k-1} and the newly predicted solution at time t_k, we can predict the solution at time level t_{k+1}. The process repeats until predictions have been obtained for all the desired

time levels. Fig. 4 illustrates how the GAN-based NIROM works, including the PCA step and the Predictive GAN algorithm.

In this work, for a GAN that has been trained to predict N time levels, $G(\mathbf{z}_{\mathbf{t_k}})$ is defined as

$$G(\mathbf{z}_{\mathbf{t_k}}) = \begin{bmatrix} \left(\tilde{\mathbf{x}}_{t_{k-N+1}}\right)^T \\ \left(\tilde{\mathbf{x}}_{t_{k-N+2}}\right)^T \\ \vdots \\ \left(\tilde{\mathbf{x}}_{t_k}\right)^T \end{bmatrix}, \tag{15}$$

where $\left(\tilde{\mathbf{x}}_{t_k}\right)^T = \left[\tilde{x}^1_{t_k}, \tilde{x}^2_{t_k}, \cdots, \tilde{x}^M_{t_k}\right]$ and it represents a predicted low dimension space of the extended SEIRS model states at time level t_k. M is the number of principal components used in the ROM, and $\tilde{x}^i_{t_k}$ represents the predicted ith principal component at time level t_k.

Considering we have solutions at time levels from t_{k-N+1} to t_{k-1}, denoted by $\left\{\mathbf{x}_{t_j}\right\}_{j=k-N+1}^{k-1}$, then to predict the solution at time level t_k we perform an optimisation defined as:

$$\begin{aligned} \mathbf{z}_{t_k} &= \arg\min_{\mathbf{z}_{\mathbf{t_k}}} \mathscr{L}\left(\mathbf{z}_{t_k}\right), \\ \mathscr{L}\left(\mathbf{z}_{t_k}\right) &= \sum_{j=k-N+1}^{k-1} \left(\mathbf{x}_{t_j} - \tilde{\mathbf{x}}_{t_j}\right)^T \mathbf{W}_\alpha \left(\mathbf{x}_{t_j} - \tilde{\mathbf{x}}_{t_j}\right), \end{aligned} \tag{16}$$

where $\mathbf{W}_\alpha$ is a square matrix of size M whose diagonal values are equal to the principal components weights, all other entries being zero. It is worth noticing that only the time levels from t_{k-N+1} to t_{k-1} are taken into account in the functional which controls the optimisation of $\mathbf{z}_{t_k}$. The newly predicted time level t_k is added to the known solutions $\mathbf{x}_{t_k} = \tilde{\mathbf{x}}_{t_k}$, and the converged latent variables $\mathbf{z}_{t_k}$ are used to initialise the latent variables at the next optimisation to predict time level t_{k+1}. The process repeats until all time levels are predicted. It is worth mentioning that the gradient of Eq. (16) can be calculated by automatic differentiation [65–67]. In other words, the error generated by the loss function is backpropagated in Eq. (16) through the generator.

Finally, the predictive GAN function is defined by:

$$f^{PredictiveGAN} : \mathbf{x}_{t_{k-N+1}}, \ldots, \mathbf{x}_{t_{k-1}} \rightarrow \tilde{\mathbf{x}}_{t_{k-N+1}}, \ldots, \tilde{\mathbf{x}}_{t_k} \tag{17}$$

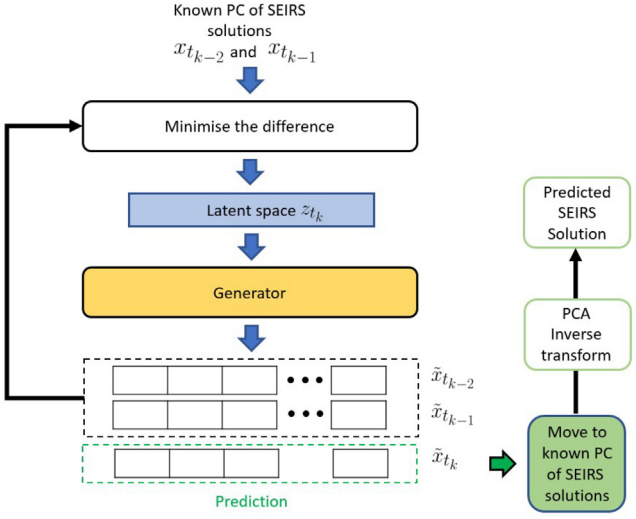


Fig. 4. Workflow of GAN-based NIROM for a sequence of three time levels ($N = 3$). Adapted from Silva et al. [52].

Both $f^{BDLSTM+BLUE}$ and $f^{PredictiveGAN}$ are iterative processes that represent the forecast functions from the BDLSTM + BLUE method and the Predictive GAN method, respectively. The digital twins, namely, a BDLSTM-based NIROM, a BDLSTM + BLUE-based NIROM and a GAN-based NIROM, are then obtained by applying the inverse transform of the PCA to the solutions provided by f^{BDLSTM}, $f^{BDLSTM+BLUE}$ and $f^{PredictiveGAN}$, respectively.

4. Results

The following section presents the test case, the parameters used in the extended SEIRS model (shown in Section 2.3), and the predictions of the two digital twin models of the spread of the COVID-19 infection for this idealised scenario explained in Section 3.1 and 3.2, respectively. The models are general, however, and could be applied to mode complex scenarios. The first digital twin is based on a bidirectional LSTM and the second is based on a predictive GAN model. Both systems were implemented using TensorFlow [68] and the Keras wrapper [69] in Python.

4.1. Test case

The domain of the test case occupies an area measuring 100 km by 100 km and is subdivided into 25 regions as shown in Fig. 5. Those labelled as 1 are regions into which people do not travel and the region labelled as 2 is where homes are located. People in the home group remain at home in region 2, and people in the mobile group can travel anywhere in regions labelled 2 or 3. Within this domain, the extended SEIRS equations will model the movement of people around the domain as well as determining which compartment and group the people are in at any given time. People can be in one of four compartments: Susceptible, Exposed, Infectious or Recovered, and for each of these, people can either be at Home or Mobile. To model the spatial variation, diffusion is used as the transport process.

Now we must set the coefficients for the extended SEIRS model. For regions 2 and 3 the diffusion coefficients have the same value:

$$k_h^c = \begin{cases} \frac{2.5L^2}{T_{\text{one day}}} & \text{for the transient case} \\ 0.05\frac{2.5L^2}{T_{\text{one day}}} & \text{for the steady} - \text{state eigenvalue problem} \end{cases} \quad \begin{aligned} &\forall h \in \{H, M\}, \\ &\forall c \in \{S, E, I, R\}, \end{aligned} \tag{18}$$

in which L is a typical length scale. Here, L is taken as the length of the domain, i.e. 100 km. For region 1, all diffusion coefficients are zero, thus no people will move into this region, see Fig. 5.

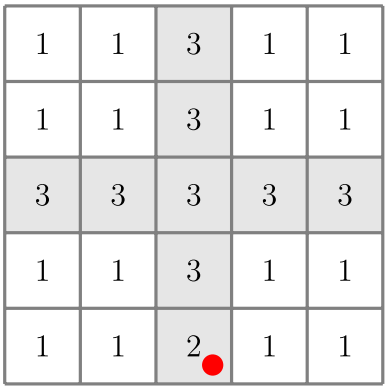


Fig. 5. Cross-shaped area in a domain of 100 km × 100 km. The grey regions represent where people can travel. The red dot indicates a location at which comparison will be made between the two digital twins based on BDLSTM and GAN.

$R0_h$, $h \in \{H, M\}$ are the average number of people in group h a person within group h infects while in that group. In this example, $R0_H = 0.2$ for people at Home ($h = H$), and $R0_M = 10$ for Mobile people ($h = M$).

If one solves an eigenvalue problem, using these values of $R0_h$, starting from an initial uninfected population, then the resulting overall $R0$ is $R0 = 7.27$. That is one person at the infectious stage of the virus can infect on average 7.27 other people. The death rate is assumed to equal the birth rate, given by:

$$\mu = \frac{1}{(60 \times 365 \times T_{\text{one day}})} = \nu, \tag{19}$$

where the average age at death is taken to be 60 years and $T_{\text{one day}}$ is the number of seconds in one day. The rate at which recovered individuals return to the susceptible state due to loss of immunity for both Home and Mobile groups is defined as:

$$\xi_h = \frac{1}{(365 \times T_{\text{one day}})}. \tag{20}$$

The interaction terms or intergroup transfer terms, $\lambda^{(\cdot)}_{hh'}$, govern how people in a particular compartment move from the home to the mobile group, or vice versa. The aim is that most people will move from home to mobile group in the morning, travel to locations in regions 2 or 3 and return home later on in the day. To achieve this, the values $\lambda^{(\cdot)}_{hh'}$ depend on other parameters, as now described. Night and day is defined through the variable:

$$R_{DAY} = 0.5 \sin\left(\frac{2\pi t}{T_{\text{one day}}}\right) + 0.5, \tag{21}$$

in which t is time into the simulation. For region 2 (see Fig. 5):

$$N_{H\,aim} = 1000(1 - R_{DAY}) + 1000, \quad N_{M\,aim} = 0, \quad \Lambda_{H,H} = \frac{1000}{T_{\text{one day}}}. \tag{22}$$

$N_{H\,aim}$ and $N_{M\,aim}$ can be thought of as the total number of people that we aim to have in the H and M groups in region 2 (i.e. where there are homes). This results in a pressure to move people from their homes during the day and back into them during the night time when they return home. Thus, $\Lambda_{H,H}$ is set in such a way as to move people out of their homes on time scale of $\frac{1}{1000}$ of a day. For all other regions:

$$N_{H\,aim} = 0, \quad N_{M\,aim} = 0, \quad \Lambda_{H,H} = 0. \tag{23}$$

For time dependent problems, a forcing term is defined as:

$$\mathscr{S}_{H2M} = 0.5 + 0.5\,\text{sgn}(F), \tag{24}$$

where

$$F = \frac{N_H - N_{H\,aim}}{\max\{\epsilon, N_H, N_{H\,aim}\}}, \tag{25}$$

in which $\text{sgn}(F) = 1$ if $F \geqslant 0$, otherwise $\text{sgn}(F) = -1$. With this definition of $\mathscr{S}_{H2M}$, in Eq. (24), for time-dependent problems, we can define the intergroup transfer terms as follows:

$$\lambda^S_{H,H} = 0.01\Lambda_{H,H}\,\mathscr{S}_{H2M}F; \qquad \lambda^E_{H,H} = \lambda^I_{H,H} = \lambda^R_{H,H} = \lambda^S_{H,H}, \tag{26}$$

$$\lambda^S_{M,M} = -\Lambda_{H,H}(1 - \mathscr{S}_{H2M})F; \qquad \lambda^E_{M,M} = \lambda^I_{M,M} = \lambda^R_{M,M} = \lambda^S_{M,M}, \tag{27}$$

$$\lambda^S_{H,M} = \Lambda_{H,H}(1 - \mathscr{S}_{H2M})F; \qquad \lambda^E_{H,M} = \lambda^I_{H,M} = \lambda^R_{H,M} = \lambda^S_{H,M}, \tag{28}$$

$$\lambda^S_{M,H} = -0.01\Lambda_{H,H}\mathscr{S}_{H2M}\,F; \qquad \lambda^E_{M,H} = \lambda^I_{M,H} = \lambda^R_{M,H} = \lambda^S_{M,H}. \tag{29}$$

For eigenvalue problems, the parameters are defined as follows:

$$r_{ratio} = 25.65; \tag{30}$$
$$r_{switch} = \begin{cases} 1 & \text{in region 1} \\ 0 & \text{elsewhere.} \end{cases}$$

The parameter r_{switch} switches on the home location in the equations below:

$$\Lambda_{H,H} = \frac{r_{switch}}{T_{\text{one day}}}; \quad \Lambda_{M,M} = 10000\frac{(1 - r_{switch})}{T_{\text{one day}}}. \tag{31}$$

The intergroup transfer coefficients are set to be

$$\lambda^S_{H,H} = \frac{1}{\epsilon}; \quad \lambda^R_{H,H} = \lambda^S_{M,M} = \lambda^R_{M,M} = \lambda^S_{H,H}, \tag{32}$$
$$\lambda^E_{H,H} = \lambda^I_{H,H} = \Lambda_{H,H} + \Lambda_{M,M}, \tag{33}$$
$$\lambda^E_{M,M} = \lambda^I_{M,M} = \Lambda_{H,H} r_{ratio}. \tag{34}$$
$$\lambda^S_{H,M} = \lambda^E_{H,M} = \lambda^I_{H,M} = \lambda^R_{H,M} = -\Lambda_{H,H} r_{ratio}, \tag{35}$$
$$\lambda^S_{M,H} = \lambda^E_{M,H} = \lambda^I_{M,H} = \lambda^R_{M,H} = -\Lambda_{H,H}. \tag{36}$$

This defines all the parameters required for the extended SEIRS model.

We are thus modelling the daily cycle of night and day for the transient calculations, in which there is a pressure for mobile people to go to their homes at night, and there will be many people leaving their homes during the day moving to the mobile group. For region 2, the average ratio of the number of people at home to the number of people that are mobile from the transient calculations during the first 10 days of the simulation is used to form the ratio r_{ratio}. This ratio is then used in the steady-state eigenvalue calculations to enforce consistency with the transient calculations. However, acknowledging the difference in the steady-state and time-dependent diffusion terms we scale the former by a factor of 0.05 as shown in Eq. (18) above. The coefficient $\frac{1}{\epsilon}$, where $\epsilon = 10^{-10}$, was added onto the diagonal of all the S and R equations (as shown above) to effectively set their values to approximately zero as they play no role in the eigenvalue calculations. This enables only minor modifications to be made to the transient code, to give the eigenvalue problem.

The domain of the numerical simulation is divided into a regular mesh of 10×10 cells. As there are four compartments and two groups in this problem, there will be eight variables for each cell in the mesh per time step, which gives a total number of 800 variables per time step. The total time of the transient simulation is 3888×10^3 seconds, or 45.75 days, with a time step of $\Delta t = 1000$ seconds resulting in 3880 time levels. Each control volume is assumed to have 2000 people in the home region cells and all other fields are set to zero, so only susceptible people are non-zero at home initially. This is with the exception that we assume that 0.1% of people at home has been exposed to the virus and will thus develop an infection.

The S, E, I, R fields for people at home and mobile are shown in Fig. 7 for the default transient configuration over 45 days. The daily cycle might, for instance, start at about 6 am (e.g. $t = 0$), say, where people start to leave their homes. People have started to leave their homes, become mobile and start to diffuse through the domain. This continues towards the end of the day where they have moved further away from their homes. However, at midnight they make their way back to their homes and thus, with a relatively small spread of the virus near the homes. Notice that at this time level, a small percentage of the population is exposed, infectious or recovered, and the rest is susceptible to S. We see the daily cycle of people moving from their homes to becoming mobile and we also see the gradual increase in the number of people in the exposed, infectious and recovered compartments for both mobile

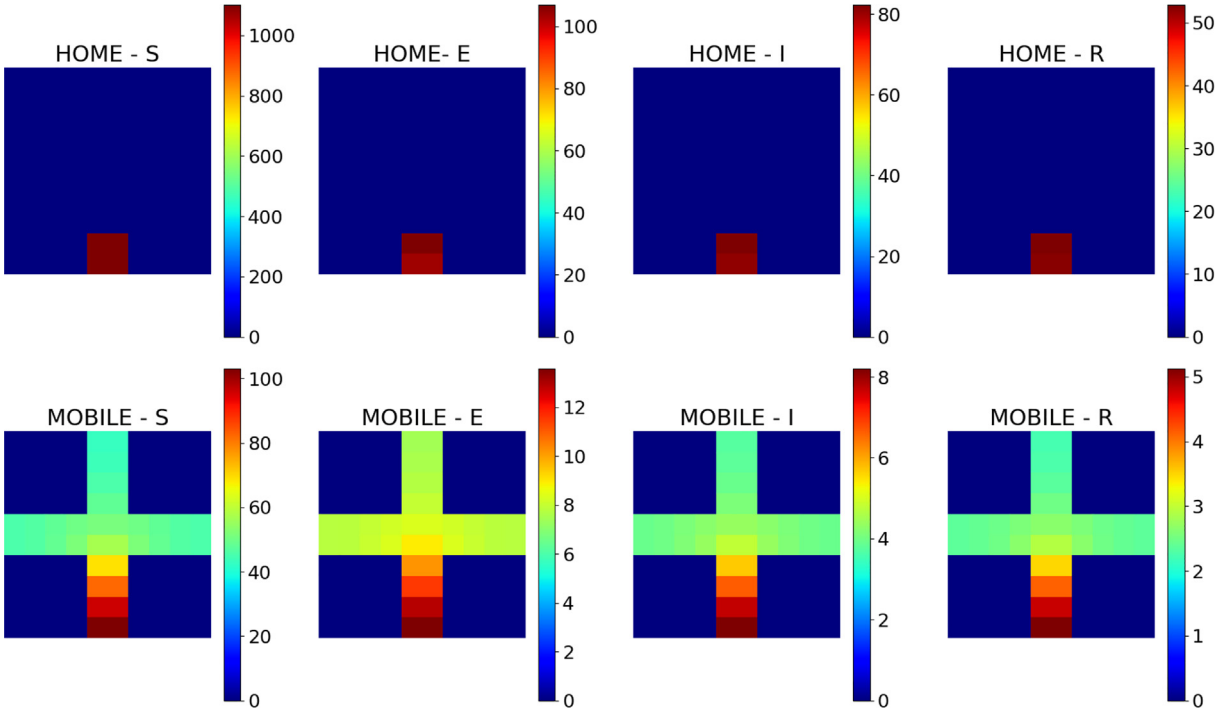


Fig. 6. Spatial variation of the test case domain after 2×10^6 seconds for the Home (top) and Mobile groups (bottom) and the S, E, I and R compartments (left to right).

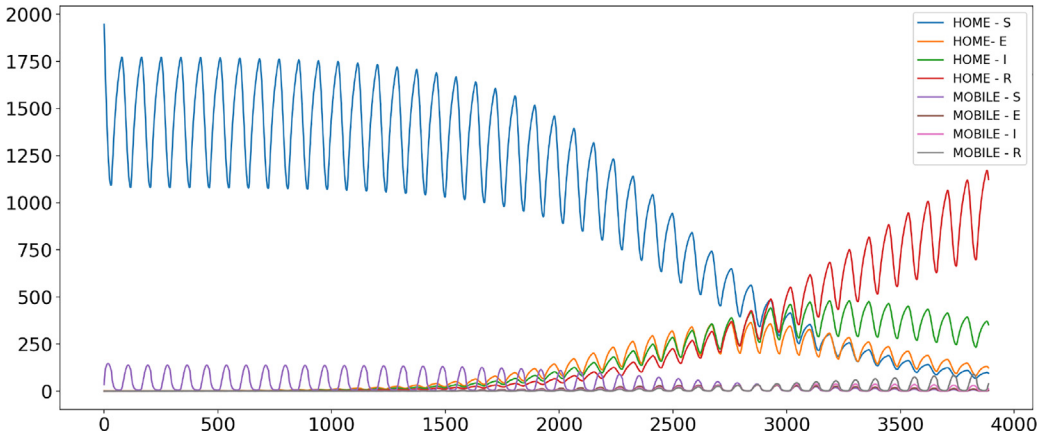


Fig. 7. Total number of people in each compartment and group versus time.

and home groups. Notice that the number of exposed and infectious people increases rapidly in this simulation and then starts to decrease because the number of susceptible people decreases. That is, recovered people gradually increases and they are immune. Fig. 6.

4.2. Dimensionality reduction

For the first step of the non-intrusive reduced-order models, a principal component analysis (PCA) is applied to the solutions of the extended SEIRS equations to obtain a low-dimensional space in which the BDLSTM and predictive GAN will operate. In total there are solutions of the SEIRS equations at 3880 time levels, each with 800 degrees of freedom (100 points in space in 4 compartments and 2 groups). After applying PCA, the first 15 principal components were retained, which represent $> 99.9\%$ of the variance. Both neural networks sample data every 10 time-steps from the PCs and both have a time window of 9 time levels, as this configuration roughly represents a cycle (one day) of the original extended SEIRS simulation. Thus, for training, both neural networks have access to 3800 samples, see Table 1.

The main goal of both the BDLSTM-based NIROMs and the GAN-based NIROM is to be able to act as digital twins or reduced-order models for the extended SEIRS model, producing predictions in a much faster time than is required to solve the extended SEIRS model itself (assuming the latter is sufficiently demanding). Once the principal components have been determined, the second step of the NIROMs is to learn the evolution of the principal components and to predict future states of the system as described in the following sections. Fig. 8.

4.3. Bidirectional long short-term memory network

The network f^{BDLSTM} is trained using the principal components at 8 time levels $t_{k-7}, t_{k-6}, \ldots, t_k$ to generate the principal components at the next time level t_{k+1}, 9 time-levels in total. The time interval

Table 1
Each training sample consists of PCs at 9 time levels taken at intervals of 10 time-steps of the SEIRS solutions. The subscripts represent time levels associated with both the SEIRS equations and the PCs. The BDLSTM is trained using 90% of the available data, reserving the remaining 10% for testing.

First sample	t_0	t_{10}	t_{20}	t_{30}	t_{40}	t_{50}	t_{60}	t_{70}	t_{80}
Second sample	t_{10}	t_{20}	t_{30}	t_{40}	t_{50}	t_{60}	t_{70}	t_{80}	t_{90}
⋮	⋮								⋮
Final Sample BDLSTM	t_{3340}	t_{3350}	t_{3360}	t_{3370}	t_{3380}	t_{3390}	t_{3400}	t_{3410}	t_{3420}
Final sample GAN	t_{3800}	t_{3810}	t_{3820}	t_{3830}	t_{3840}	t_{3850}	t_{3860}	t_{3870}	t_{3880}

between these time levels is equivalent to 10 time steps of the extended SEIRS simulation results. The network is trained using 90% of the available data, reserving the remaining 10% for testing. Fig. 9 depicts the prediction of one time-step, at a single point of the domain, using data from the original simulation, once f^{BDLSTM} is trained. This is a validation that the model can make accurate predictions on both the training data and the test data. The BDLSTM-based NIROM is obtained by the inverse transform of the PCA coefficients predicted by f^{BDLSTM}.

The BDLSTM architecture is based on Cui et al. [40] and f^{BDLSTM} was trained for 500 epochs using a grid search of hyperparameters

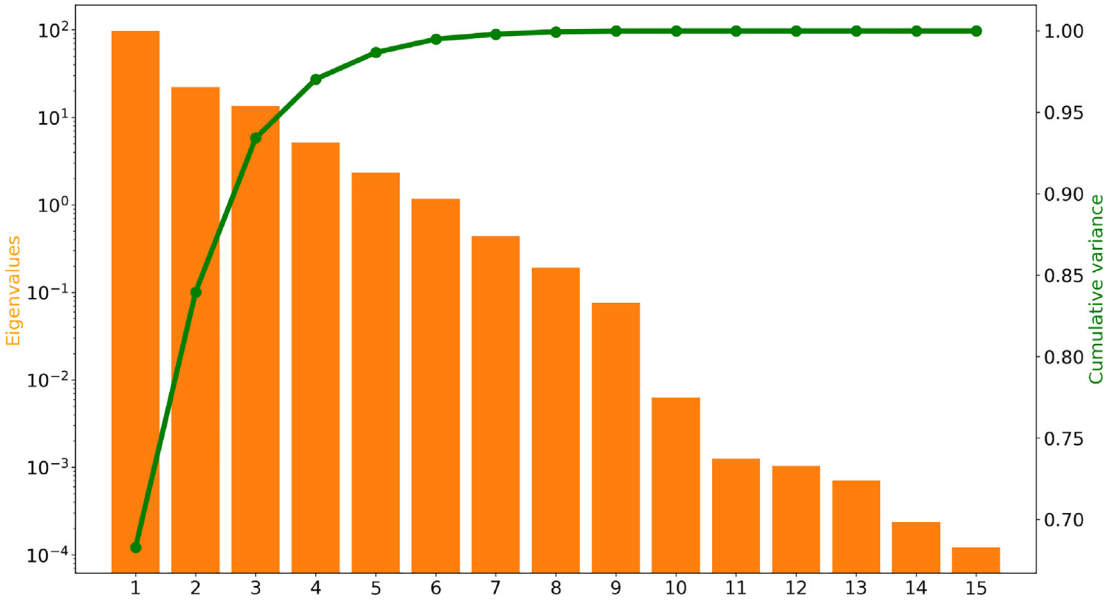


Fig. 8. Eigenvalues (left) and normalised cumulative sum of the variance (right) of the first 15 components.

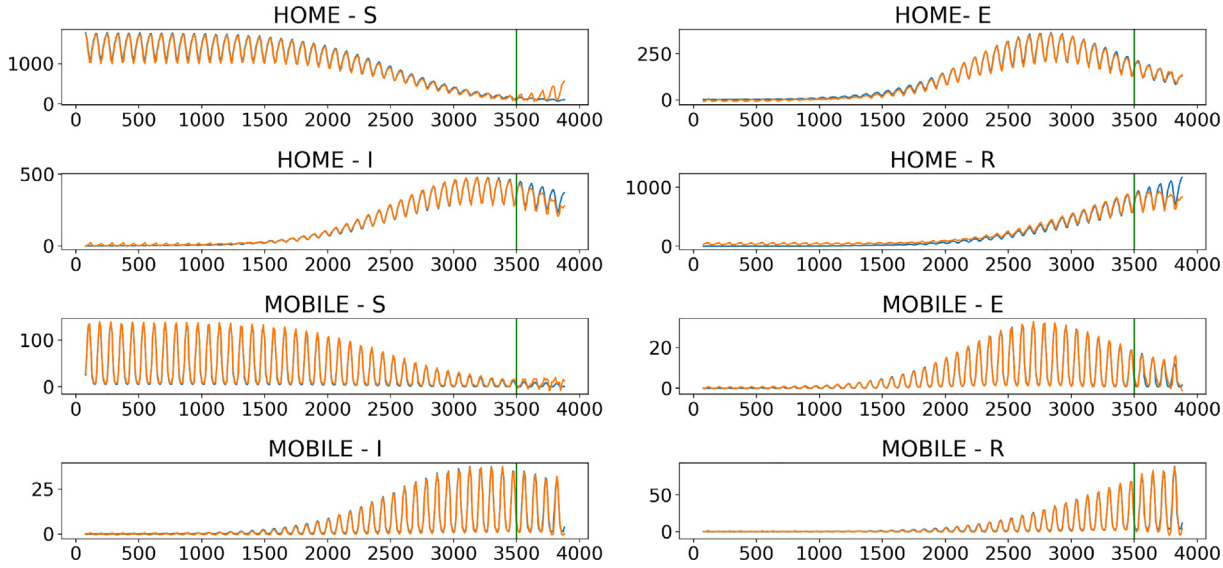


Fig. 9. The BDLSTM-based NIROM prediction (orange) over time of the outcomes of the infection (in number of people) in one point (marked as a red circle in Fig. 5) of the mesh starting at time step 0. The predictions are off-line, not data-corrected and have a sliding window of 8 time-steps and use the data from the original dataset (blue) to predict the next one. The green line shows the start of the test data.

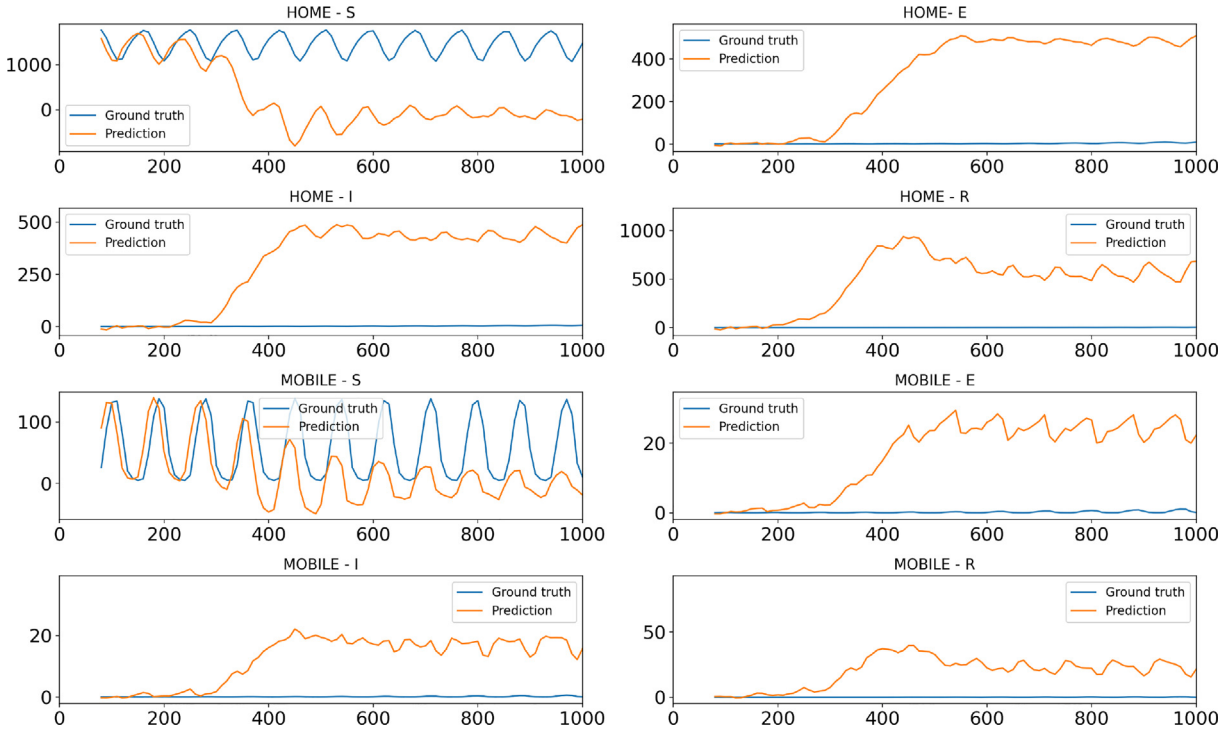


Fig. 10. The BDLSTM-based NIROM prediction, over time, of the outcomes of the infection (in number of people) in one point (marked as a red circle in Fig. 5) without any data-correction from time-step 0. The predictions from BDLSTM-based NIROM act iteratively like an input for the prediction of the following time-step.

including hidden nodes in the LSTM layer, batch sizes, and dropouts.

Without including data-correction (Fig. 10), the predictions of the BDLSTM-based NIROM start diverging after $\sim$ 30 iterations. This means that BDLSTM-based NIROM does not diverge greatly from the original dataset before $\sim$ 30 cycles of input-output, without external information. Therefore, the prediction by f^{BDLSTM} needs to be data-corrected to align with the dynamics of the principal components of the extended SEIRS solution.

The data-corrected prediction by the BDLSTM (BDLSTM + BLUE-based NIROM) starting from time step 90 (9×10^4 seconds), is shown in Fig. 11a. Each cycle in the curves corresponds roughly to a period of one day. Fig. 11b depicts the data-corrected prediction every 10 time-steps starting from time-step 2000 of the simulation (2×10^6 seconds). Comparable results are obtained at other points of the mesh. In both cases, the BDLSTM + BLUE-based NIROM struggles at predicting the Susceptible compartments in both Home and Mobile groups. The BDLSTM + BLUE-based NIROM performs poorly at predicting the initial values in both cases starting from the beginning of the dataset and from $t = 2000$ (2×10^6 seconds).

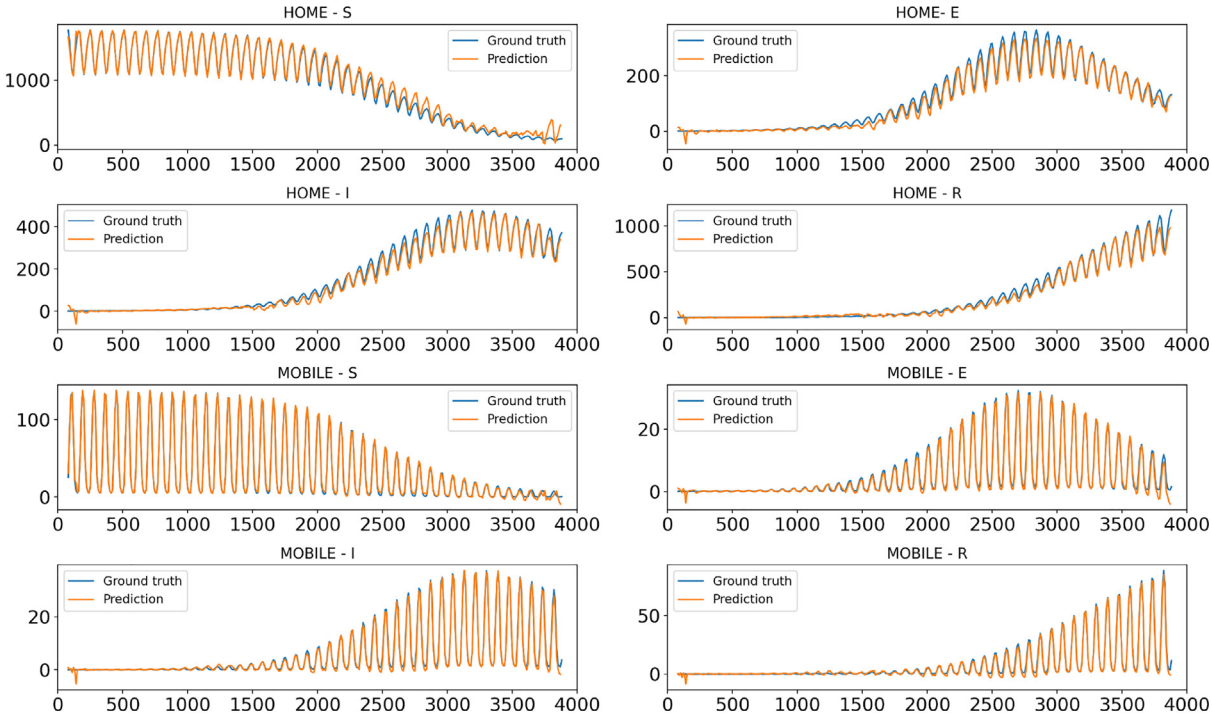
4.4. Prediction using GAN

A GAN-based NIROM, is applied to the spatial variation of COVID-19 infection, to make predictions based on training using data from the numerical simulation. The generator and discriminator are trained using a sequence of 9 time levels of the principal components of the extended SERIS model with a time interval of 10 time steps between them. The first 8 time levels are used in the optimisation process, described in Section 3.2.1, and the last time level is used in the prediction. The network is trained using all time steps of the numerical simulation.

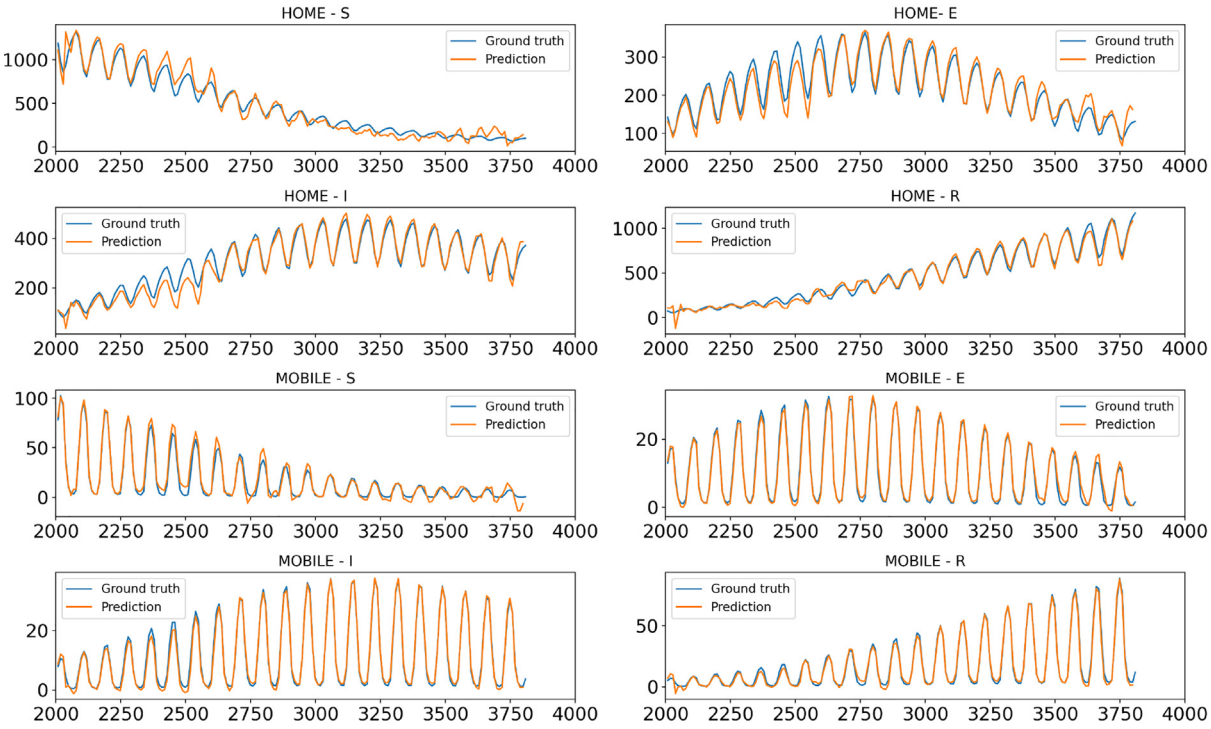
The GAN architecture is based on DCGAN [70]. The generator and discriminator are trained for 55,000 epochs. The 9 time levels are given to the networks as a two-dimensional array with nine rows and fifteen columns. Each row represents a time level and each column is a principal component from PCA (a low dimensional representation of the simulation states). Although it is not an image, it can be represented as one. The DCGAN can take advantage of the time dependency of the two-dimensional array (the image), as the simulation states for the first time level is in the first row, for the second time level is in the second row, and so forth. During the optimisation process in each iteration of $f^{PredictiveGAN}$, the singular values from the SVD are used as weights in the Eq. (16).

The prediction in $f^{PredictiveGAN}$ is performed by starting with 8 time levels from the numerical simulation and using the generator to predict the ninth. During the next iteration, the last prediction is used in the optimisation process and this is repeated until the end of the simulation. It is worth mentioning that after 8 iterations the $f^{PredictiveGAN}$ works only with data from the predictions. Data from the numerical simulation is used only for the starting points. The GAN-based NIROM is then obtained by applying the inverse transform of the PCA to the predictions given by $f^{PredictiveGAN}$.

Fig. 12a shows the prediction over time of the GAN-based NIROM at one point of the mesh (bottom-right corner of region 2 shown in Fig. 5). Each cycle in the curves corresponds to a period of one day. The process is repeated this time with the simulation starting at time step 2×10^3 (2×10^6 seconds). The result over time for one point of the mesh (bottom-right corner of region 2) is presented in Fig. 12b. Comparable results regarding the error in the prediction are obtained at other points of the mesh, therefore we do not present them here. We observe from Fig. 12 that the GAN-based NIROM can reasonably predict the outcomes of the numerical model.



(a) Starting at 9×10^4 seconds.

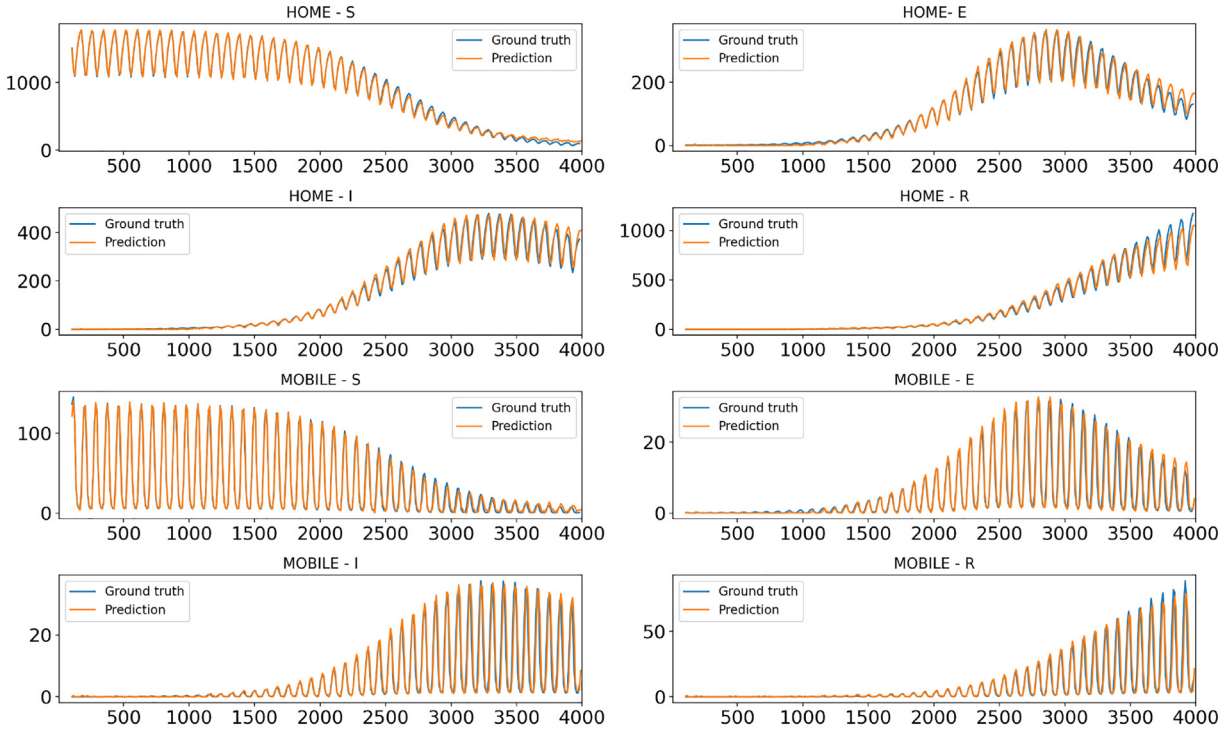


(b) Starting at 2×10^6 seconds.

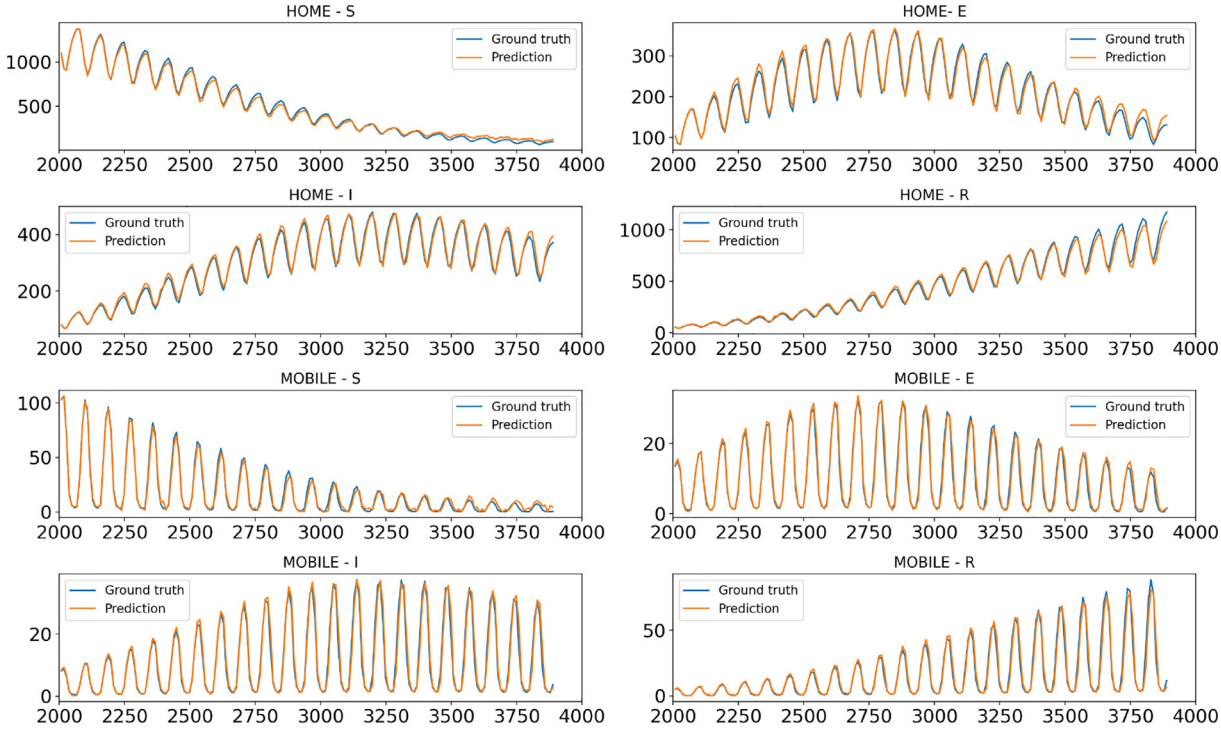
Fig. 11. BDLSTM + BLUE-based NIROM prediction (in number of people) at one point (marked as a red circle in Fig. 5) of the domain over time starting from different time levels.

4.5. Comparison between BDLSTM and predictive GAN

Formatted as Jupyter notebooks, the codes for both digital twins presented in this paper are publicly available at https://github.com/c-quilo/SEIR-BDLSTM (for the BDLSTM) and https://github.com/viluiz/gan/tree/master/PredGAN (for the GAN). The dependencies of the codes are Python (version 3.7), Numpy (version 1.18.5), Keras (version 2.4.3) and TensorFlow (version 2.4.0).



(a) Starting at 9×10^4 seconds.



(b) Starting at 2×10^6 seconds.

Fig. 12. GAN-based NIROM prediction (in number of people) at one point (marked as a red circle in Fig. 5) of the domain over time starting from different time levels.

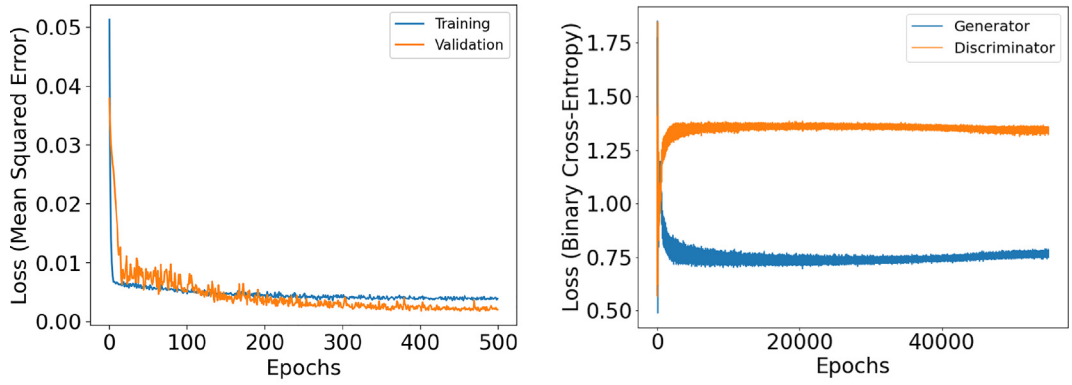
Table 2
Hyperparameters used for the data-corrected bidirectional LSTM and the predictive GAN. (†Time distributed dense output layer with a sigmoid activation function, ‡negative slope coefficient, ††Adam with Nesterov momentum.)

	BDLSTM	GAN
Epochs	500	55,000
Batch size	32	256
Hidden nodes	64	n/a
Latent space size	n/a	100
Batch normalisation	–	✓(generator)
Layer normalisation	✓	–
Dropout	0.5	0.3 (discriminator)
Activation function	sigmoid†	LeakyReLU (0.3‡)
Loss function	Mean Square Error	Binary cross entropy
Optimiser	Nadam††	Adam
Learning rate	0.001	0.001
β_1	0.9	0.9
β_2	0.999	0.999
ϵ	10^{-7}	n/a

The final hyperparameters used in the Bidirectional Long Short-Term Memory and predictive GAN networks are given in Table 2.

The training losses of both networks, BDLSTM and GAN, are depicted in Fig. 13.

Fig. 14 presents a comparison over a short period of time (100 time-levels) including BDLSTM-based NIROM, the BDLSTM + BLUE-based NIROM, and GAN-based NIROM. Over this time period, the BDLSTM + BLUE-based NIROM and GAN-based NIROM show better prediction accuracy than the BDLSTM-based NIROM. The GAN-based NIROM slightly outperforms the BDLSTM + BLUE-based NIROM for compartments S, E, I and R of the Home group and for compartment S of the Mobile group. For the remaining compartments and groups, the BDLSTM + BLUE-based NIROM and GAN-based NIROM both yield predictions that are extremely close to the original extended SEIRS model (the ground truth). The BDLSTM benefits greatly from the data-correction with the BLUE estimator. However, it needs constant



(a) BDLSTM (b) Predictive GAN

Fig. 13. Training losses of f^{BDLSTM} (mean squared error), and the generator G and discriminator D used in $f^{PredictiveGAN}$ (binary cross-entropy).

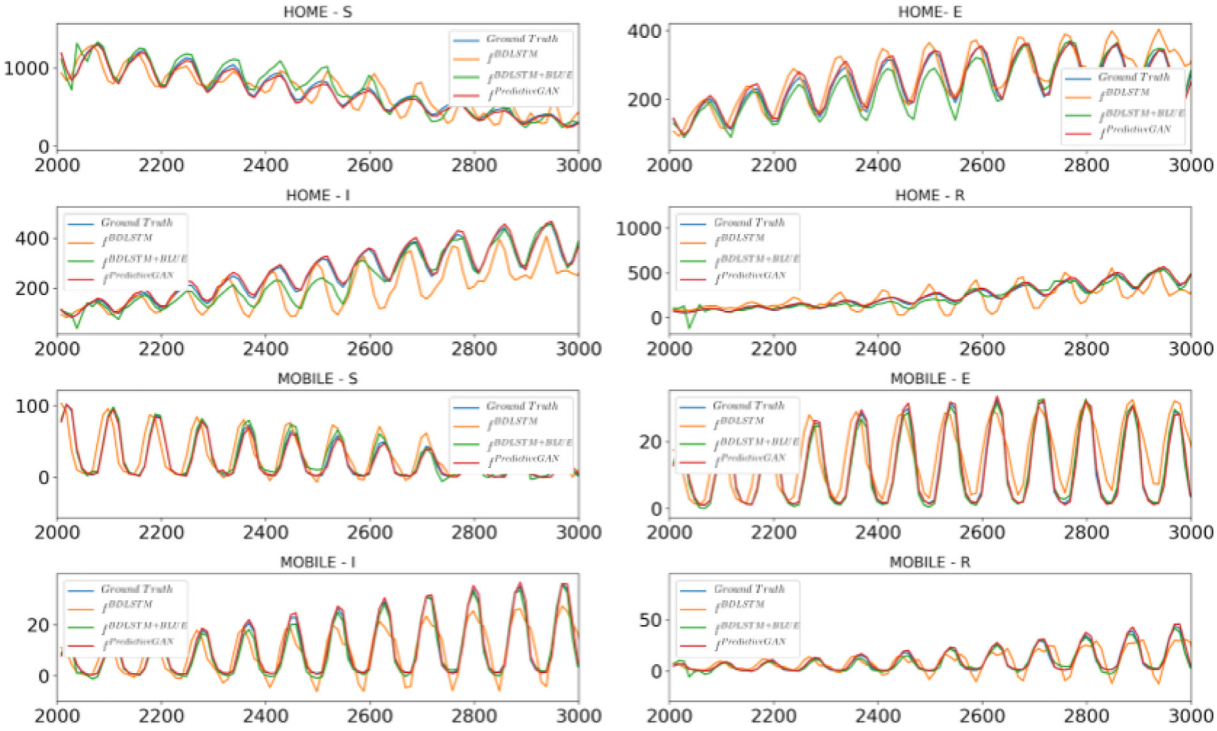
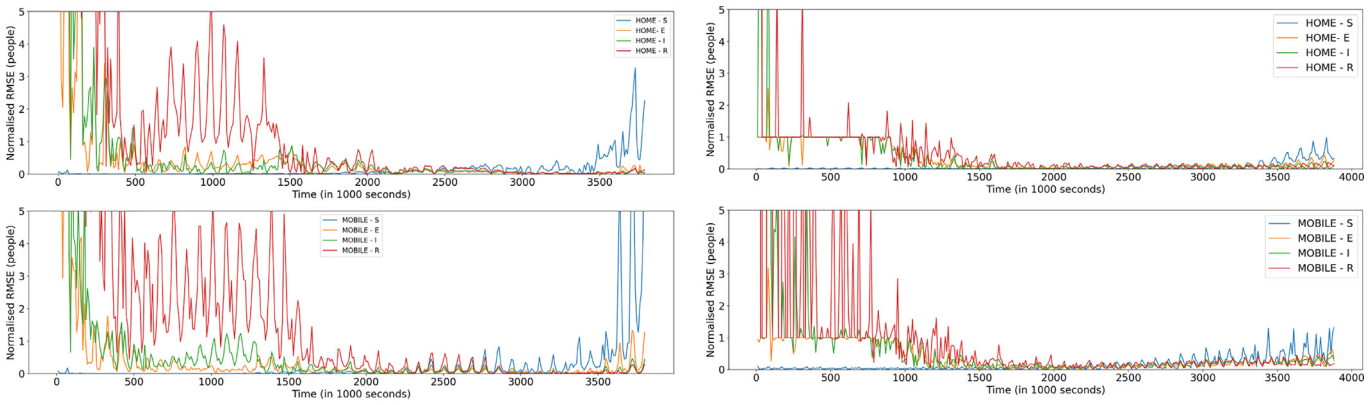


Fig. 14. Comparison of forecasts (in number of people) produced by three methods: BDLSTM-based NIROM (orange), BDLSTM + BLUE-based NIROM (green), and GAN-based NIROM (red), over time to the ground truth (blue). The forecast starts from t = 2000 (2×10^6 seconds) of the SEIRS model solution.



(a) BDLSTM+BLUE-based NIROM (b) GAN-based NIROM

Fig. 15. Time-series of the Normalised root mean squared error of the predictions for the Home (top) and Mobile (bottom) compartments. Left: BDLSTM + BLUE-based NIROM, Right: GAN-based NIROM.

input from the model solution data to correct its trajectory. While the GAN-based NIROM replicates the dynamics of the SEIRS model solution well, just with the input of 8 time levels at the start. Thus, the GAN-based NIROM does not constantly look at the extended SEIRS model solution data.

Fig. 15 shows the normalised root mean squared error (NRMSE) over time for both digital twins. The mean was calculated using only the active regions of each compartment and group (Fig. 5), i. e the Home group is only considered in region 2, while the Mobile group is considered across the entire active region (all regions but 1). For this simulation, we start the prediction at time step 90 (9×10^{40} seconds).

The RMSE at time level k is defined as the following:

$$RMSE^k = \frac{\|\mathbf{u}^k - \mathbf{v}^k\|_2}{\sqrt{m}} \tag{37}$$

where k is the time level, $\mathbf{u}^k \in \mathbb{R}^m$ are the predictions for a particular compartment and group, based on BDLSTM + BLUE-based NIROM or GAN-based NIROM at time level k (having mapped the output of the network back to the control-volume grid), $\mathbf{v}^k \in \mathbb{R}^m$ is the data from the extended SEIRS model solutions at time level k, m is the number of active control volumes per compartment and group, and $\|\|_2$ represents the Euclidean norm. A RMSE value is computed for the eight combinations of compartments and groups. The normalised RMSE at time level k is defined by:

$$NRMSE^k = \frac{\|\mathbf{u}^k - \mathbf{v}^k\|_2}{\|\mathbf{v}^k\|_2}. \tag{38}$$

In the prediction of the Home compartments using the BDLSTM + BLUE-based NIROM prediction, it is worth noting that there is a decreasing trend of the Home – Recovered and Home – Infectious people, while the number of people in Home – Susceptible increases towards the end of the dataset surpassing the normalised RMSE of the other compartments and groups. The predictions by GAN-based NIROM on the Home group present similar behaviour over time. However, the decreasing trends are more rapid and the increased error of the Home – Susceptible compartment is smaller towards the end of the dataset.

There is a very similar behaviour for the predictions of the Mobile groups for both the BDLSTM + BLUE-based and GAN-based NIROMs. There is a decreasing trend for the Mobile – Exposed, Mobile – Infectious and Mobile – Recovered people for both NIROMs. Additionally, the error seen for people in Home – Susceptible increases over time for both models. A summary of the average normalised RMSE over time is shown in Table 3.

In order to compare the skill of the BDLSTM + BLUE-based NIROM and GAN-based NIROM, we look at the spatial skill score (SS):

$$SS = 1 - \frac{RMSE_{f^{BDLSTM+BLUE}}}{RMSE_{f^{PredictiveGAN}}} \tag{39}$$

where $RMSE_{BDLSTM+BLUE-based\ NIROM}$ and $RMSE_{GAN-based\ NIROM}$ are the spatial RMSE averaged over time on each region. The spatial SS is depicted in Fig. 16. If $SS < 0$, the predictive GAN has more skill at predicting that region. Otherwise, if $SS > 0$, the BDLSTM + BLUE-based NIROM is better at predicting that region. While GAN-based NIROM outperforms BDLSTM + BLUE-based NIROM for the prediction of the Home group (compartments S, E, I and R), in general, the data-corrected BDLSTM produces more accurate predictions for the Mobile – Infectious and Mobile – Recovered people.

The execution times with optimisation for both NIROMs are shown in Table 4. These execution times are concerning a set of 9 time-steps. The speed-up for the original simulation is also shown. If optimisation is included, the BDLSTM + BLUE-based NIROM prediction is 2 orders of magnitude faster than GAN-based NIROM.

5. Discussion

These experiments serve as a proof of concept for digital twins or non-intrusive reduced-order models (NIROMs) of SEIRS models.

Table 3
Average normalised RMSE over time for both BDLSTM + BLUE-based NIROM, and GAN-based NIROM over the 4 compartments and 2 groups. The average does not consider the first 50 time-steps as the normalised RMSE is too sensitive during this period.

NIROM	H-S	H-E	H-I	H-R	M-S	M-E	M-I	M-R
BDLSTM + BLUE-based	0.179	0.170	0.164	0.888	0.409	0.176	0.192	0.353
GAN-based	0.078	0.210	0.182	0.264	0.175	0.281	0.287	0.503

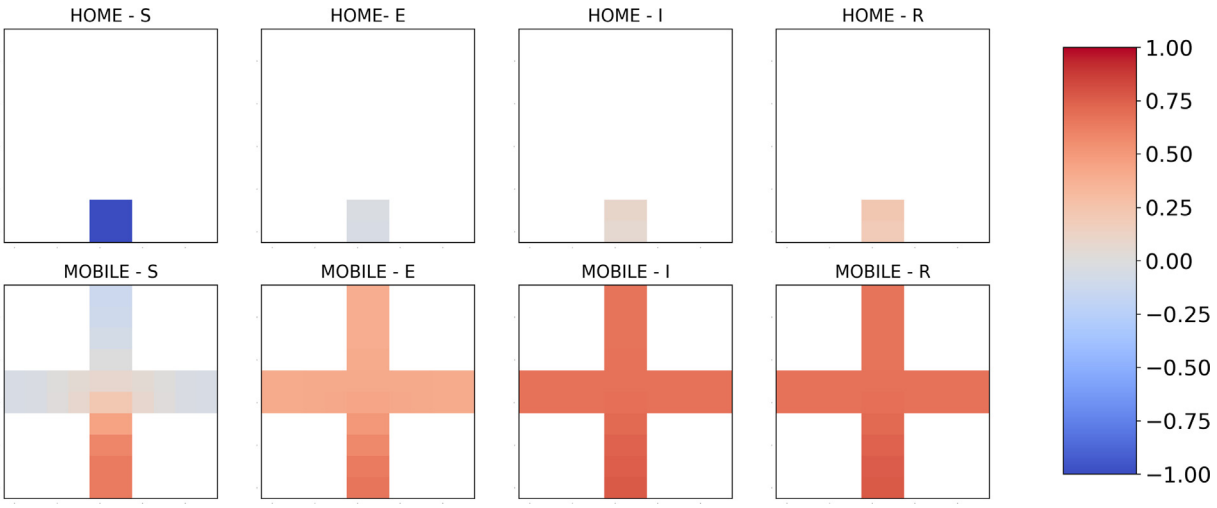


Fig. 16. Spatial skill score over the mesh for all 4 compartments and 2 groups. If the skill score is less than zero, GAN-based NIROM has more skill at predicting that region. Otherwise, if the skill score is greater than 0, the BDLSTM + BLUE-based NIROM is better at predicting that region. The first 50 time-steps were not considered.

Table 4
Execution times with optimisation of a single set of 9 time-steps, and the speed-up of each method with respect to the original simulation. The original simulation does not include an optimisation, thus both speed-up times are with respect to the simulation execution time for 9 time-steps.

NIROM	Execution times (s)	Speed-up (–)	Storage size
SEIRS	0.45	–	7.25 Mb
BDLSTM + BLUE	1.6×10^{-2}	28.12	0.14 Mb
GAN	1.9×10^{0}	0.24	0.14 Mb

In the following we refer to the NIROM based on a BDLSTM as BDLSTM-based NIROM; the NIROM based on the data-corrected BDLSTM as BDLSTM + BLUE-based NIROM and the NIROM based on the predictive GAN as the GAN-based NIROM. The predictions produced by the GAN-based NIROM outperform the BDLSTM + BLUE-based NIROM in the Susceptible compartments for both Home and Mobile groups, while the BDLSTM + BLUE-based NIROM outperforms the GAN-based NIROM for the Exposed, Infectious and Recovered compartments. However, it is important to note that the predictions produced by the BDLSTM + BLUE-based NIROM are corrected using the BLUE optimisation. The predictive GAN also includes optimisation, but is capable of generalising over time just by optimising observational data at the beginning of its prediction.

- The BDLSTM-based NIROM (without data correction) provides fast forecasts which are up to 4 orders of magnitude faster than the simulation. However, it was observed that this model diverges quickly from the model solution when the predicted output is used as an input to predict the following time-step.
- This was fixed by adding a data-correction step, using BLUE (resulting in the BDLSTM + BLUE-based NIROM). The produced forecasts using this method are 2 orders of magnitude faster than the extended SEIRS model solution. However, it has the disadvantage of constantly requiring the extended SEIRS model solution as an input to correct the trajectory of the forecast.
- While the BDLSTM + BLUE-based NIROM outperforms the GAN-based NIROM at producing forecasts of the extended SEIRS model solution, the GAN-based NIROM has the great advantage of not needing a constant stream of data from the extended SEIRS model. The GAN-based NIROM manages to predict the dynamics of the extended SEIRS model accurately with only the input of 8 time-steps at the start of the simulation. These 8 time-steps serve as a constraint to initialise the forecast of the GAN-based NIROM. Additionally, GANs can generate reliable information from random noise, which LSTMs are not designed to do. Nonetheless, the execution times of the GAN-based NIROM are slower than those of the BDLSTM + BLUE-based NIROM by 2 orders of magnitude.
- the GAN-based NIROM has great potential when applied to larger problems. In any case, for a more demanding SEIRS model (with more compartments or with a higher spatial resolution for example), the speed-ups of both digital twins are expected to improve.

Therefore, a combination of both techniques will be valuable in the future for a more accurate prediction that includes information from the time series, using an LSTM, and creating realistic information trained with adversarial networks. Similar efforts in combining LSTM and GAN/adversarial training have been studied for Electrocardiograms [71] and classical music generation [72]. Thus, the prediction of future time-steps will be embedded into the GAN, without requiring further optimisation to make a prediction. This method will reduce execution times, with the caveat that training GANs comes at a higher computational cost. Such a combination of LSTMs and GANs has not yet been applied to model the spread of COVID-19, nor the wider field of epidemiology.

Our choice of using a BDLSTM is supported by previous studies using BDLSTMs for COVID-19 prediction. In Shahid et al. [73], the authors show a comparison of different deep learning methods for forecasting COVID-19 time series data and concluded that a BDLSTM shows robustness and it is an appropriate predictor for this type of data, outperforming a vanilla-LSTM and a Gated Recurrent Unit network. Chatterjee et al. [74] also presents that a BDLSTM is a strong predictive model for forecasting new cases and resulting deaths of COVID-19.

6. Conclusions and future work

In this paper, we have presented two methods for creating digital twins of a SEIRS model which has been extended to be able to model both the spatial and temporal spread of the virus. The digital twins or non-intrusive reduced-order models (NIROMS) were used for predicting the future states of the model comparing the evolution of these experiments to the ground truth. The first NIROM uses Principal Component Analysis (PCA) to reduce the dimension of the

problem followed by a Bidirectional Long Short-term memory network (BDLSTM) to learn the evolution of the reduced variables. The second NIROM also uses PCA but relies on a Predictive Generative Adversarial Network (GAN) to learn the temporal evolution of reduced variables. The prediction produced by the GAN-based NIROM outperforms the predictions by the BDLSTM + BLUE-based NIROM in the Susceptible compartments. Furthermore, GANs can generate reliable information from random noise. These novel approaches: using data-corrected optimisation for LSTMs, and using GANs with an optimisation step show very promising results for time-series prediction.

In summary, this paper proposed a novel Non-Intrusive Reduced Order Model (NIROM) based on a Bidirectional Long Short-Term Memory (BDLSTM) network with a data-correction step derived from BLUE for improved accuracy. A second NIROM is also developed, based on a GAN modified to predict in time. Finally, we compared the two models. The novelty of this paper also relies on that this is the first time that reduced-order modelling techniques have been applied to virus modelling.

Future work involves the combination of LSTM (unidirectional or bidirectional) with a GAN to produce more accurate forecasts that take advantage of the time-series information along with realistic predictions produced by the GAN. Additionally, these frameworks could be applied to larger domains of idealised towns including more compartments to study more realistic epidemiological models.

Glossary

Extended SEIRS	A set of SEIRS equations that have been extended to take into account spatial variation
$f^{BDLSTM+BLUE}$	Predictive model of the principal components of the extended SEIRS model using a bidirectional LSTM with the Best Linear Unbiased Estimator
f^{BDLSTM}	Predictive model of the principal components of the extended SEIRS model using a bidirectional LSTM
$f^{PredictiveGAN}$	Predictive model of the principal components of the extended SEIRS model using a predictive Generative Adversarial Network
BDLSTM + BLUE-based NIROM	Non-intrusive reduced order model of the extended SEIRS model using a bidirectional LSTM with the Best Linear Unbiased Estimator
BDLSTM-based NIROM	Non-intrusive reduced order model of the extended SEIRS model using a bidirectional LSTM
BLUE	Best Linear Unbiased Estimator
GAN-based NIROM	Non-intrusive reduced order model of the extended SEIRS model using a predictive Generative Adversarial Network

CRediT authorship contribution statement

César Quilodrán-Casas: Writing – original draft, Writing – review & editing, Conceptualization, Methodology, Software. **Vinicius L.S. Silva:** Writing – original draft, Writing – review & editing, Conceptualization, Methodology, Software. **Rossella Arcucci:** Supervision, Methodology, Funding acquisition, Writing – original draft, Writing – review & editing. **Claire E. Heaney:** Supervision, Methodology, Writing – original draft, Writing – review & editing. **YiKe Guo:** Supervision. **Christopher C. Pain:** Methodology, Software, Conceptualization, Writing – original & draft, Writing – review & editing, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is supported by the EPSRC grant EP/T003189/1 Health assessment across biological length scales for personal pollution exposure and its mitigation (INHALE), grant EP/N010221/1 Managing Air for Green Inner Cities (MAGIC) consortium, the PREMIERE programme grant (EP/T000414/1), the MUFFINS grant (EP/P033180/1), the RELIANT grant (EP/V036777/1), and by BECAS CHILE, a governmental Chilean scholarship from the Agencia Nacional de Investigación y Desarrollo (ANID). This work has been undertaken, in part, as a contribution to 'Rapid Assistance in Modelling the Pandemic' (RAMP), initiated by the Royal Society. In particular, we would like to acknowledge the useful discussion had within the Environmental and Aerosol Transmission group of RAMP, coordinated by Profs Paul Linden and Christopher Pain.

References

[1] E. Dong, H. Du, L. Gardner, An interactive web-based dashboard to track COVID-19 in real time, Lancet Infect. Diseases 20 (2020) 533–534.
[2] M. Park, A.R. Cook, J.T. Lim, Y. Sun, B.L. Dickens, A systematic review of COVID-19 epidemiology based on current evidence, J. Clin. Med. 9 (2020) 967.
[3] A.H. Auchincloss, A.V. Diez Roux, A new tool for epidemiology: the usefulness of dynamic-agent models in understanding place effects on health, Am. J. Epidemiol. 168 (2008) 1–8.
[4] E. Cuevas, An agent-based model to evaluate the COVID-19 transmission risks in facilities, Comput. Biol. Med. 121 (2020) 103827.
[5] M.S. Shamil, F. Farheen, N. Ibtehaz, I.M. Khan, M.S. Rahman, An Agent-Based Modeling of COVID-19: Validation, Analysis, and Recommendations, Cogn. Comput. (2021) 1–12.
[6] M.Y. Li, J.S. Muldowney, Global stability for the SEIR model in epidemiology, Math. Biosci. 125 (1995) 155–164.
[7] A. Rădulescu, C. Williams, K. Cavanagh, Management strategies in a SEIR-type model of COVID 19 community spread, Scientific Rep. 10 (2020) 21256.
[8] S. Basu, J. Andrews, Complexity in mathematical models of public health policies: a guide for consumers of models, PLoS Med. 10 (2013) e1001540.
[9] K. Rock, S. Brand, J. Moir, M.J. Keeling, Dynamics of infectious diseases, Rep. Prog. Phys. 77 (2014) 026602.
[10] I. Cameron, T. Lyons, J. Kenworthy, Trends in vehicle kilometres of travel in world cities, 1960–1990: underlying drivers and policy responses, Transp. Policy 11 (2004) 287–298.
[11] D. Pavlidis, Z. Xie, J.R. Percival, J.L. Gomes, C.C. Pain, O.K. Matar, Two-and three-phase horizontal slug flow simulations using an interface-capturing compositional approach, Int. J. Multiph. Flow 67 (2014) 85–91.
[12] S. Eubank, Scalable, efficient epidemiological simulation, in: Proceedings of the 2002 ACM symposium on Applied computing, pp. 139–145..
[13] K.L. Cooke, P. Van Den Driessche, Analysis of an SEIRS epidemic model with two delays, J. Math. Biol. 35 (1996) 240–260.
[14] P. Song, Y. Lou, Y. Xiao, A spatial SEIRS reaction-diffusion model in heterogeneous environment, J. Differ. Equ. 267 (2019) 5084–5114.
[15] D. Xiao, F. Fang, A. Buchan, C. Pain, I. Navon, A. Muggeridge, Non-intrusive reduced order modelling of the Navier-Stokes equations, Comput. Methods Appl. Mech. Eng. 293 (2015) 522–541.
[16] J.S. Hesthaven, S. Ubbiali, Non-intrusive reduced order modeling of nonlinear problems using neural networks, J. Comput. Phys. 363 (2018) 55–78.
[17] C.A. Quilodrán Casas, Fast ocean data assimilation and forecasting using a neural-network reduced-space regional ocean model of the north Brazil current (Ph.D. thesis), Imperial College London, 2018..
[18] C. Quilodrán Casas, R. Arcucci, P. Wu, C. Pain, Y.-K. Guo, A reduced order deep data assimilation model, Physica D (2020) 132615.
[19] J. Lever, M. Krzywinski, N. Altman, Points of significance: Principal component analysis, 2017..
[20] C. Quilodrán Casas, R. Arcucci, Y. Guo, Urban air pollution forecasts generated from latent space representations, in: ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations..

[21] C. Quilodrán-Casas, R. Arcucci, L. Mottet, Y. Guo, C. Pain, Adversarial autoencoders and adversarial LSTM for improved forecasts of urban air pollution simulations, Published as a workshop paper at ICLR 2021 SimDL Workshop (2021)..
[22] T.R.F. Phillips, C.E. Heaney, P.N. Smith, C.C. Pain, An autoencoder-based reduced-order model for eigenvalue problems with application to neutron diffusion, Int. J. Numer. Methods Eng. (accepted doi.org/10.1002/nme.6681) (2021)..
[23] T. Bui-Thanh, M. Damodaran, K. Willcox, Proper orthogonal decomposition extensions for parametric applications in compressible aerodynamics, 21st AIAA Applied Aerodynamics Conference (2003).
[24] P. Breitkopf, I. Lepot, C. Sainvitu, P. Villon, Multi-fidelity POD surrogate-assisted optimization: Concept and aero-design study, Struct. Multidiscip. Optim. 56 (2017) 1387–1412.
[25] D. Xiao, F. Fang, C.E. Heaney, I.M. Navon, C.C. Pain, A domain decomposition method for the non-intrusive reduced order modelling of fluid flow, Comput. Methods Appl. Mech. Eng. 354 (2019) 307–330.
[26] G. Aversano, A. Bellemans, Z. Li, A. Coussement, O. Gicquel, A. Parente, Application of reduced-order models based on PCA & Kriging for the development of digital twins of reacting flow applications, Comput. Chem. Eng. 121 (2019) 422–441.
[27] E. Kaiser, B. Noack, L. Cordier, A. Spohn, M. Segond, M. Abel, G. Daviller, J. Östh, S. Krajnović, R. Niven, Cluster-based reduced-order modelling of a mixing layer, J. Fluid Mech. 754 (2014) 365–414.
[28] Z. Wang, D. Xiao, F. Fang, R. Govindan, C.C. Pain, Y. Guo, Model identification of reduced order fluid dynamics systems using deep learning, Int. J. Numer. Meth. Fluids 86 (2018) 255–268.
[29] A. Rasheed, O. San, T. Kvamsdal, Digital twin: values, challenges and enablers from a modeling perspective, IEEE Access 8 (2020) 21980–22012.
[30] B. Moya, I. Alfaro, D. Gonzalez, F. Chinesta, E. Cueto, Physically sound, self-learning digital twins for sloshing fluids, PLoS One 15 (2020) e0234569.
[31] M. Kapteyn, D. Knezevic, D. Huynh, M. Tran, K. Willcox, Data-driven physics-based digital twins via a library of component-based reduced-order models, Int. J. Numer. Methods Eng. (forthcoming)..
[32] S.E. Ahmed, S.M. Rahman, O. San, A. Rasheed, I.M. Navon, Memory embedded non-intrusive reduced order modeling of non-ergodic flows, Phys. Fluids 31 (2019) 126602.
[33] M. Kherad, M.K. Moayyedi, F. Fotouhi, Reduced order framework for convection dominant and pure diffusive problems based on combination of deep long short-term memory and proper orthogonal decomposition/dynamic mode decomposition methods, Int. J. Numer. Methods Fluids (forthcoming)..
[34] C. Quilodrán-Casas, R. Arcucci, C. Pain, Y. Guo, Adversarially trained LSTMs on reduced order models of urban air pollution simulations, arXiv preprint arXiv:2101.01568 (2021)..
[35] M. Guo, J.S. Hesthaven, Reduced order modeling for nonlinear structural analysis using Gaussian process regression, Comput. Methods Appl. Mech. Eng. 341 (2018) 807–826.
[36] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780.
[37] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-C. Woo, Convolutional LSTM network: A machine learning approach for precipitation nowcasting, in: Advances in neural information processing systems, pp. 802–810..
[38] K. Greff, R.K. Srivastava, J. Koutník, B.R. Steunebrink, J. Schmidhuber, LSTM: a search space odyssey, IEEE Trans. Neural Networks Learn. Syst. 28 (2016) 2222–2232.
[39] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, IEEE Trans. Signal Process. 45 (1997) 2673–2681.
[40] Z. Cui, R. Ke, Z. Pu, Y. Wang, Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction, arXiv preprint arXiv:1801.02143 (2018)..
[41] Z. Cui, R. Ke, Z. Pu, Y. Wang, Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values, Transp. Res. Part C: Emerg. Technol. 118 (2020) 102674.
[42] A. Graves, N. Jaitly, A.-R. Mohamed, Hybrid speech recognition with deep bidirectional LSTM, in: 2013 IEEE workshop on automatic speech recognition and understanding, IEEE, pp. 273–278..
[43] G. Liu, J. Guo, Bidirectional LSTM with attention mechanism and convolutional layer for text classification, Neurocomputing 337 (2019) 325–338.
[44] A. Elsheikh, S. Yacout, M.-S. Ouali, Bidirectional handshaking LSTM for remaining useful life prediction, Neurocomputing 323 (2019) 148–156.
[45] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep learning, vol. 1, MIT Press Cambridge, 2016..
[46] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680..
[47] T. Karras, S. Laine, T. Aila, A Style-Based Generator Architecture for Generative Adversarial Networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2019, pp. 4401–4410.
[48] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, T. Aila, Analyzing and Improving the Image Quality of StyleGAN, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8110–8119..
[49] C. Chu, A. Zhmoginov, M. Sandler, CycleGAN, a Master of Steganography, arXiv preprint arXiv:1712.02950 (2017)..
[50] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, H. Greenspan, GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification, Neurocomputing 321 (2018) 321–331.
[51] Y. Liu, Z. Qin, T. Wan, Z. Luo, Auto-painter: Cartoon image generation from sketch by using conditional Wasserstein generative adversarial networks, Neurocomputing 311 (2018) 78–87.
[52] V.L.S. Silva, C.E. Heaney, Y. Li, C.C. Pain, Data Assimilation Predictive GAN (DA-PredGAN): applied to determine the spread of COVID-19, arXiv preprint arXiv:2105.07729 (2021)..
[53] Z. Yang, Z. Zeng, K. Wang, S.-S. Wong, W. Liang, M. Zanin, P. Liu, X. Cao, Z. Gao, Z. Mai, et al., Modified SEIR and AI prediction of the epidemics trend of COVID-19 in China under public health interventions, J. Thoracic Disease 12 (2020) 165.
[54] S.F. Ardabili, A. Mosavi, P. Ghamisi, F. Ferdinand, A.R. Varkonyi-Koczy, U. Reuter, T. Rabczuk, P.M. Atkinson, COVID-19 Outbreak Prediction with Machine Learning, Available at SSRN 3580188 (2020)..
[55] V.K.R. Chimmula, L. Zhang, Time series forecasting of COVID-19 transmission in Canada using LSTM networks, Chaos Solitons Fractals (2020) 109864.
[56] S.M. Ayyoubzadeh, S.M. Ayyoubzadeh, H. Zahedi, M. Ahmadi, S.R.N. Kalhori, Predicting COVID-19 Incidence Through Analysis of Google Trends Data in Iran: Data Mining and Deep Learning Pilot Study, JMIR Public Health and Surveillance 6 (2020) e18828.
[57] N.E.M. Khalifa, M.H.N. Taha, A.E. Hassanien, S. Elghamrawy, Detection of coronavirus (COVID-19) associated pneumonia based on generative adversarial networks and a fine-tuned deep transfer learning model using chest X-ray dataset, arXiv preprint arXiv:2004.01184 (2020)..
[58] L. Wang, A. Wong, COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images, arXiv preprint arXiv:2003.09871 (2020)..
[59] S. Wang, X. Yang, L. Li, P. Nadler, R. Arcucci, Y. Huang, Z. Teng, Y. Guo, A Bayesian Updating Scheme for Pandemics: Estimating the Infection Dynamics of COVID-19, IEEE Comput. Intell. Mag. 15 (2020) 23–33.
[60] H. Bao, X. Zhou, Y. Zhang, Y. Li, Y. Xie, COVID-GAN: Estimating Human Mobility Responses to COVID-19 Pandemic through Spatio-Temporal Conditional Generative Adversarial Networks, in: Proceedings of the 28th International Conference on Advances in Geographic Information Systems, pp. 273–282..
[61] J. Yoon, D. Jarrett, M. van der Schaar, Time-series generative adversarial networks (2019)..
[62] Institute for Disease Modelling, SEIR and SEIRS models, 2020..
[63] P. Nadler, S. Wang, R. Arcucci, X. Yang, Y. Guo, An epidemiological modelling approach for COVID-19 via data assimilation, Eur. J. Epidemiol. 35 (2020) 749–761.
[64] UK Government, COVID-19: infection prevention and control (IPC), 2020..
[65] R.E. Wengert, A simple automatic derivative evaluation program, Commun. ACM 7 (1964) 463–464.
[66] S. Linnainmaa, Taylor expansion of the accumulated rounding error, BIT Numer. Math. 16 (1976) 146–160.
[67] A.G. Baydin, B.A. Pearlmutter, A.A. Radul, J.M. Siskind, Automatic differentiation in machine learning: a survey, J. Mach. Learn. Res. 18 (2017) 5595–5637.
[68] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. Software available from tensorflow.org..
[69] F. Chollet, et al., Keras. URL: https://keras.io, 2015..
[70] A. Radford, L. Metz, S. Chintala, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, arXiv preprint arXiv:1511.06434 (2015)..
[71] F. Zhu, F. Ye, Y. Fu, Q. Liu, B. Shen, Electrocardiogram generation with a bidirectional LSTM-CNN generative adversarial network, Scientific Rep. 9 (2019) 1–11.
[72] O. Mogren, C-RNN-GAN: Continuous recurrent neural networks with adversarial training, arXiv preprint arXiv:1611.09904 (2016)..
[73] F. Shahid, A. Zameer, M. Muneeb, Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM, Chaos Solitons Fractals 140 (2020) 110212.
[74] A. Chatterjee, M.W. Gerdes, S.G. Martinez, Statistical explorations and univariate timeseries analysis on COVID-19 datasets to understand the trend of disease spreading and death, Sensors 20 (2020) 3089.

Dr. César Quilodrán Casas is a Research Associate at the Data Science Institute, Imperial College London. In 2018, he completed his PhD in data-driven oceanography at the Space and Atmospherics Physics Group at Imperial College London. Currently he works in the Health assessment across biological length scales for personal pollution exposure and its mitigation (INHALE) project and the COVID-19 related Risk Evaluation Fast Intelligent Tool (RELIANT) project, where he forms part of a vast multi-disciplinary team. His research topics include a combination of data assimilation, machine learning and reduced order models.

Vinicius Luiz Santos Silva is a PhD student in the Department of Earth Science and Engineering at Imperial College London. He has been working on machine learning techniques to enhance forward simulators. He is also a Senior Reservoir Engineer at Petrobras, where he has worked in the development, optimisation and data assimilation of carbonate Pre-salt fields. He obtained a BSc in control and automation engineering from Minas Gerais Federal University in 2009, a specialisation in petroleum engineering from Petrobras University in 2011 and an MSc in civil engineering from Rio de Janeiro Federal University in 2016. His areas of interest are machine learning, optimisation, data assimilation, numerical simulation and uncertainty quantification.

Dr Rossella Arcucci is a Research Fellow at Data Science Institute (DSI) and an Advanced Research Fellow/Data Scientist at Imperial College Business School. Being member of the Artificial Intelligence (AI) network at ICL and her main area of expertise is in Data Science with Machine Learning. PhD in Computational and Computer Science in February 2012. At DSI, she created and she is leading the Data Assimilation and Machine Learning (DataLearning) Working Group (https://www.imperial.ac.uk/data-science/research/research-themes/datalearning/). She received the acknowledgement of Marie Sklodowska-Curie fellow from European Commission Research Executive Agency in Brussels on the 27th of November 2017. She is co-investigator of several projects at DSI.

Dr. Claire Heaney is a Research Associate in the Applied Modelling and Computation Group at Imperial College London, currently employed on the EPSRC MUFFINS (EP/P033180/1) and RELIANT (EP/V036777/1) projects. Her research interests lie in the application of model reduction and machine learning to various fields, including single-phase and multi-phase flows. Previous employment experience includes working for Plaxis (geo-engineering software) in the Netherlands as well as research posts at Cardiff and Durham Universities.

Prof. Yi-Ke Guo is the Vice-President (Research and Development) of Hong Kong Baptist University and Professor of Computing Science in the Department of Computing at Imperial College London. He was the founding Director of the Data Science Institute at Imperial College since 2014, which is one of the six Imperial College Global Challenge Institutes. He is a Fellow of the Royal Academy of Engineering (FREng), Member of Academia Europaea (MAE), and Fellow of British Computer Society. Prof.Guo obtained his first-class honours degree in Computing Science from Tsinghua University, China, in 1985 and received his PhD in Computational Logic from Imperial College in 1993. In 1999, he founded InforSense Limited, an Imperial College spin-off software company specialized in big data analysis for life science and medicine. His main research interests lie in the field of machine learning and large-scale data management and has contributed to numerous major research projects. Prof. Guohas published over 250 articles. He had won the "Most Innovative Data Intensive Application Award" at the Supercomputing 2002 conference for Discovery Net, the Bio-IT World "Best Practices Award" for U-BIOPRED in 2014 and the "Best Open Source Software Award" from ACM SIGMM in 2017.

Prof. Christopher Pain leads the Applied Modelling and Computation Group (AMCG) at Imperial College London (ICL). A recipient of the ICL Research Excellence award in 2010, AMCG is one of Imperial's largest research groups comprising approximately 60 scientists. Professor Pain is also the director of the Data Assimilation Lab in the Data Science Institute at ICL and holds visiting professorships at several universities. As well as leading the modelling on several EPSRC-funded projects and consortia, Professor Pain co-leads the Royal Society's Environmental and Aerosol Transmission group, which is part of RAMP (Rapid Assistance in Modelling the Pandemic), which coordinates the voluntary modelling response to COVID-19.