Journal of Computational Science 69 (2023) 102024

Contents lists available at ScienceDirect

Journal of Computational Science

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

Ensemble Kalman filter for GAN-ConvLSTM based long lead-time forecasting

Meiling Cheng [a], Fangxin Fang [a,*], Ionel M. Navon [b], Christopher Pain [a]

[a] Applied Modeling and Computation Group, Department of Earth Science and Engineering, Imperial College London, SW7 2BP, UK
[b] Department of Scientific Computing, Florida State University, Tallahassee, FL, 32306-4120, USA

ARTICLE INFO

Keywords:
Ensemble Kalman filter
Convolutional LSTM
Deep convolutional GAN
Machine learning
Data assimilation

ABSTRACT

Data-driven machine learning techniques have been increasingly utilized for accelerating nonlinear dynamic system prediction. However, machine learning-based models for long lead-time forecasts remain a significant challenge due to the accumulation of uncertainty along the time dimension in online deployment. To tackle this issue, the ensemble Kalman filter (EnKF) has been introduced to machine learning-based long-term forecast models to reduce the uncertainty of long lead-time forecasts of chaotic dynamic systems. Both the deep convolutional generative adversarial network (DCGAN) and convolutional long short term memory (ConvLSTM) are used for learning the complex nonlinear relationships between the past and future states of dynamic systems. Using an iterative Multi-Input Multi-Output (MIMO) algorithm, the two-hybrid forecast models (DCGAN-EnKF and ConvLSTM-EnKF) are able to yield long lead-time forecasts of dynamic states. The performance of the hybrid models has been demonstrated by one-level and two-level Lorenz 96 models. Our results show that the use of EnKF in ConvLSTM and DCGAN models successfully corrects online model errors and significantly improves the real-time forecasting of dynamic systems for a long lead-time.

1. Introduction

Machine learning (ML) has been widely applied in many fields including environment sciences [1], ocean modeling [2,3], engineering and physics [4]. In particular, ML models are extensively used as a data-driven predictive tool to better understand complex and nonlinear physical systems yielding high computational efficiency [5]. However, due to the complex nonlinearity and uncertainties of a nonlinear dynamic system, it is still challenging to perform accurate forecasts of the future evolution of many dynamical systems [6,7]. Therefore, a reliable and efficient technique for long lead-time forecasts of dynamic systems is highly desirable.

Recent progress in ML models has shown how to learn the dynamics of a model from its historical datasets and forecast beyond the training period, to some extent [8–10]. For example, Chen et al. [6] proposed an Auto-Reservoir Neural Network to perform the multi-step-ahead prediction based on a short-term time series. Similarly, da Silva et al. [11] combined a Complete Ensemble Empirical Mode Decomposition (CEEMD) with Stacking-ensemble learning (STACK) to 30-minute-ahead forecasts of wind turbine power generation. Zhang et al. [12] proposed a deep learning algorithm, which integrated a bidirectional gated recurrent unit with attention mechanism (BiAGRU) for 24 h ahead air pollution prediction. Kow et al. [13] carried out 10-step-ahead PM2.5 forecasts for multiple stations by combining a CNN and a Back Propagation Neural Network (BPNN). Zhou et al. [14] proposed a Deep Multi-output LSTM neural network model for 4-step-ahead air quality forecasting in Taipei City, Taiwan. These results demonstrate that ML models possess enormous potential in learning various complicated functions between past and future events and providing accurate multi-step-ahead forecasts. Nevertheless, it can be noticed that due to the instabilities and biases in forecasting subgrid-scale processes, the performance of ML-based models generally decays as the forecasting horizon extends beyond the training period [15,16]. There is a general consensus that data assimilation is a suitable approach to address this issue [17,18].

Data assimilation techniques are used to integrate observations into the state of the dynamic system at a given time in an optimal way, thus reducing the error and uncertainty in numerical modeling [19–22]. In particular, the Kalman filter originally proposed by Kalman [23], is a widely used optimal sequential data assimilation method for linear state–space systems and measurement processes with Gaussian error statistics. Ensemble Kalman filter (EnKF) as a special Kalman filter [24], has been extensively utilized in many fields, such as geosciences [25], hydrology [26,27] and physics [28–31], to correct background prediction based on a combination of heterogeneous ground measurements, experiments and remote sensing [32]. Furthermore, a hybrid EnKF and variational analysis scheme can be used for producing a superior ensemble of forecasts at longer lead times [33].

Recent works have also drawn attention to combining data assimilation with ML models in dynamic systems. Considering the evolution

* Corresponding author.
E-mail address: f.fang@imperial.ac.uk (F. Fang).

https://doi.org/10.1016/j.jocs.2023.102024
Received 23 March 2022; Received in revised form 5 January 2023; Accepted 28 March 2023
Available online 8 April 2023

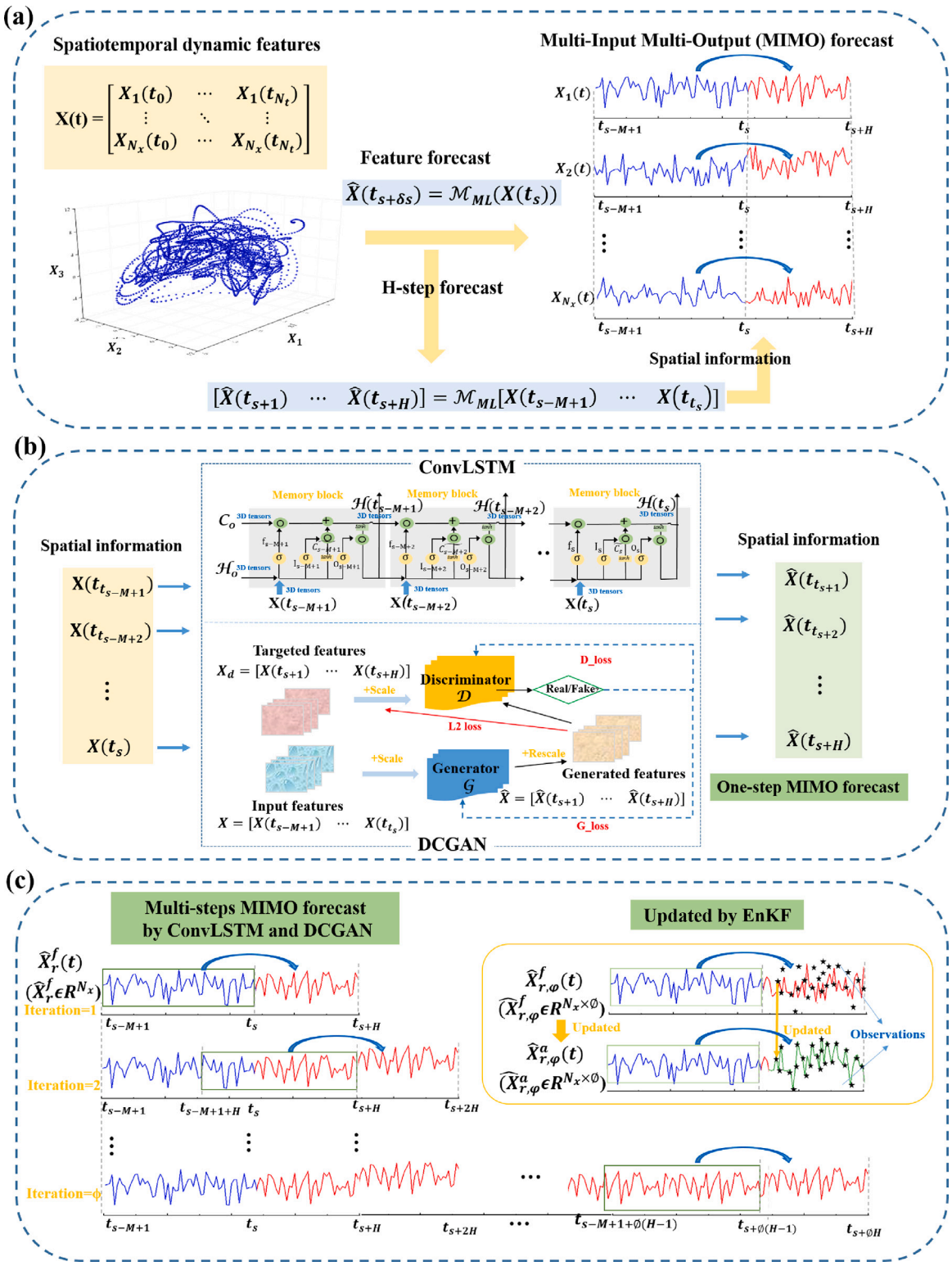


Fig. 1. Illustration of a hybrid model combining ensemble data assimilation and ConvLSTM and DCGAN models for long lead-time forecasting. (a) Spatiotemporal dynamic features are used for Multi-Input Multi-Output (MIMO) forecast. (b) Architectures of the ConvLSTM and DCGAN models for MIMO forecast. (c) Extension of forecasting horizon integrating ConvLSTM and DCGAN with EnKF in a recursively MIMO strategy.

of a dynamic system:

$$\mathcal{X}(t_{s+\delta t}) = \mathcal{M}(\mathcal{X}(t_s)) + \mathcal{E}(t_s), \tag{1}$$

where $\mathcal{M}$ represents the dynamic model, $\mathcal{X}(t_s)$ is the state variable at time t_s, and $\mathcal{E}(t_s)$ is the model noise. Data assimilation methods are used for reducing the error $\mathcal{E}$ by the inclusion of observational data into the dynamic model $\mathcal{M}$, thus improving forecast accuracy (referred to as $\mathcal{M}_{physical} + \mathcal{E}_{DA}$ *mode*). While it is fairly new and limited, a hybrid data assimilation (DA) and ML model can be generally classified as:

(a) $\mathcal{X}(t_{s+\delta t}) = \mathcal{M}_{physical}(\mathcal{X}(t_s)) + \mathcal{E}_{DA+ML}(t_s)$: The dynamic model is physics-informed modeling, where the dynamic processes are

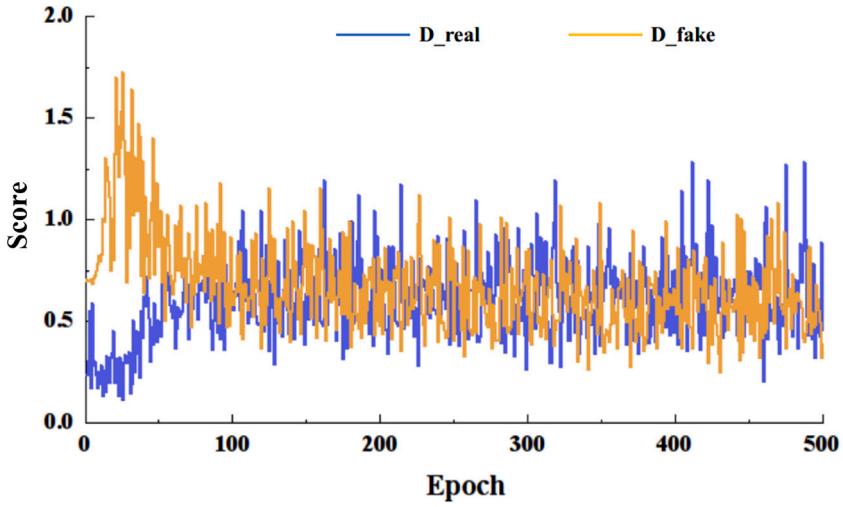


Fig. 2. Discriminator scores between real and fake in the DCGAN model during the training process.

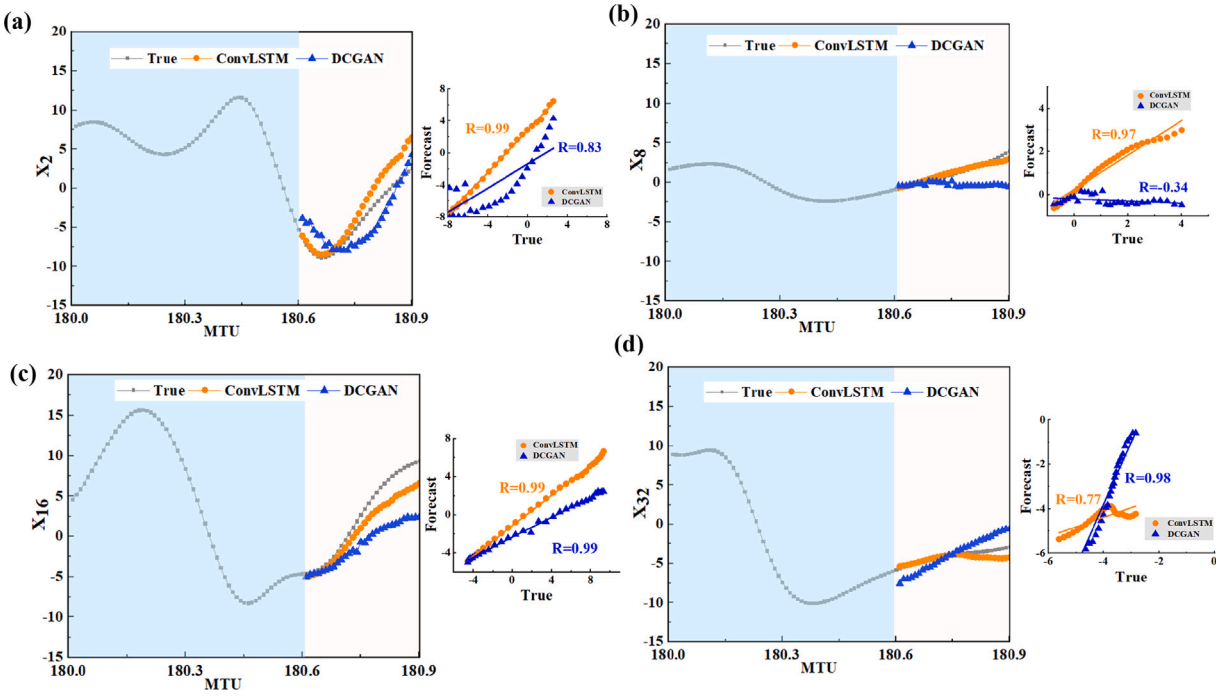


Fig. 3. One step MIMO forecasts of X_2, X_8, X_{16} and X_{32} state variables conducted by the ConvLSTM and DCGAN models, where the length of the known time series (for inputs) $M = 60$ (0.06 MTU) and the predicted time series $H = 30$ (0.03 MTU), i.e., 30-time-step-ahead forecasts.

represented by the governing equations. The physical models can thus be used for complex nonlinear dynamic simulations, but require a high computational cost. ML techniques are used for efficient data assimilation in real-time forecasts. ML models learn the assimilation process (error term $\mathcal{E}$) and improve the predictive accuracy of physical models [18,27,31,34–37].

(b) $\mathcal{X}(t_{s+\delta t}) = \mathcal{M}_{physical+parameterized_{DA/ML}}(\mathcal{X}(t_s)) + \mathcal{E}(t_s)$: Data assimilation and ML techniques can be used for parameterization in physical models [32,38,39]. For example, Pawar and San [32] used a set of neural network architectures to learn the correlation between resolved flow variables and the parameterizations of unresolved flow dynamics, where data assimilation was used to correct the error in the physical model.

(c) $\mathcal{X}(t_{s+\delta t}) = \mathcal{M}_{ML+DA}(\mathcal{X}(t_s)) + \mathcal{E}(t_s)$: Instead of physical models, a ML model is used as a forward surrogate dynamic model $\mathcal{M}$, where the computational cost is significantly reduced in comparison to physical models. Here data assimilation techniques are used for training the ML models. For example, the data was assimilated for training the parameters (weights) within the network's hidden layers [18,40,41]. ML models (e.g. autoencoders) can be used to exploit the latent structure while data assimilation is employed for correcting the model errors in the latent space. Note that in their work, data assimilation techniques were used only during the training period. As discussed above, the issue of decay of predictive accuracy in ML modeling persists for the long lead-time forecast.

(d) $\mathcal{X}(t_{s+\delta t}) = \mathcal{M}_{ML}(\mathcal{X}(t_s)) + \mathcal{E}_{DA+ML}(t_s)$: ML models are used for both rapid forecasting (M_{ML}) and efficient data assimilation correcting the error term $\mathcal{E}$ from ML models as the forecasting horizon extends beyond the training period, thus improving predictive accuracy. In current works, e.g. Gottwald and Reich [42],

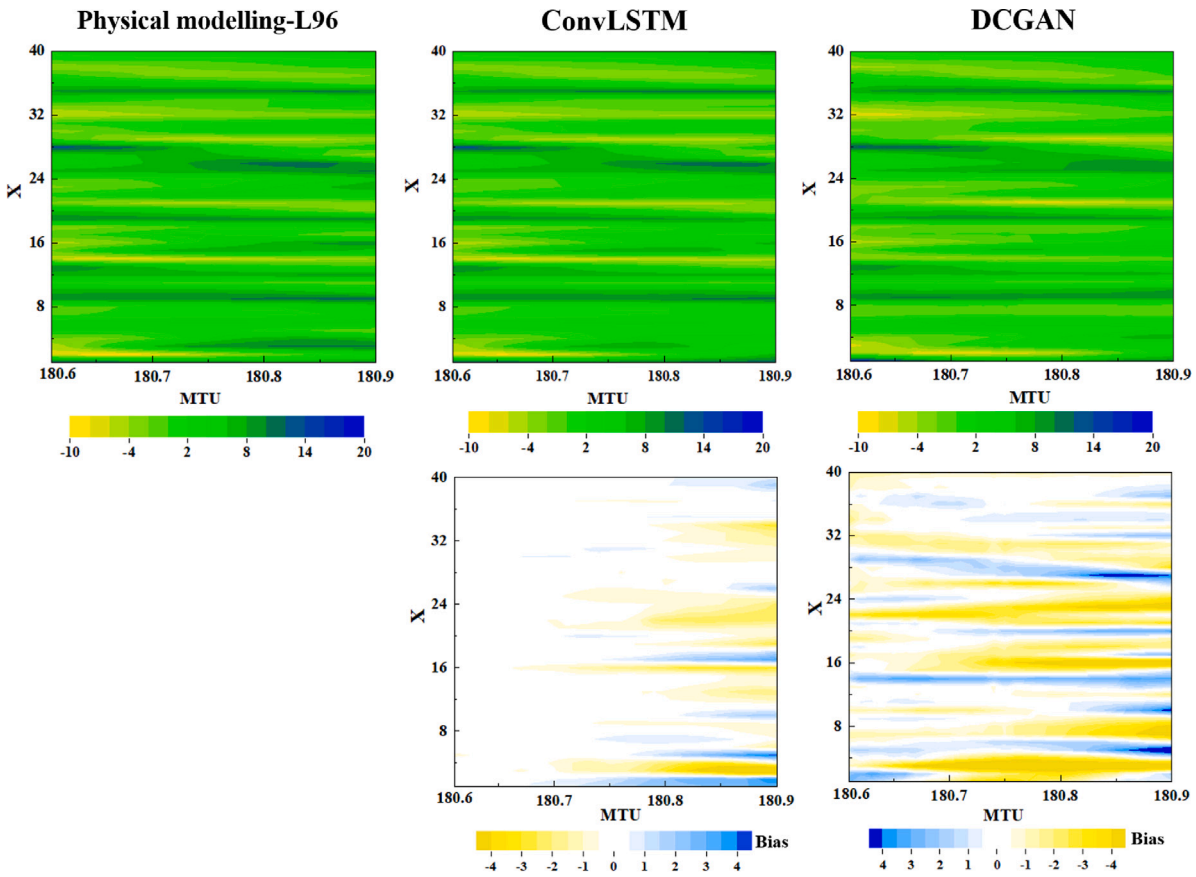


Fig. 4. Comparison of the trajectories (top panel) and residuals (bottom panel) of all state variables between the physical modeling-L96, the ConvLSTM and DCGAN models for the lead-times 180.6 – 180.9 MTU, where the results from the physical modeling-L96 are considered as 'references'.

data assimilation techniques were used for updating the outputs in ML modeling just once, not for long-term forecasting. Our aim in this work is to introduce data assimilation into operational ML-based real-time forecasting.

Generally speaking, the hybrid DA and ML frameworks have immense potential in physical sciences [43]. Despite these advanced trends, most of the existing hybrid DA and ML frameworks either rely on the physical models as a forward model with a high computation burden or use ML models with a limited forecasting capability, to perform short-term forecasts. Thus, they are not applicable to efficiently carry out long lead-time forecasts of nonlinear dynamic systems. To the best of our knowledge, the topic of real-time ML-based operational frameworks in fluid mechanics remains relatively unexplored.

In this work, we propose a hybrid data assimilation and ML-based long lead-time forecast framework ($\mathcal{M}_{ML}+\mathcal{E}_{DA+ML}$ *mode*). The convolutional LSTM (ConvLSTM, Xingjian et al. [44]) and deep convolutional GAN (DCGAN, Goodfellow et al. [45]), acting as the dynamic surrogate model $\mathcal{M}_{ML}$, are used for efficient calculation of ensembles as well as long term forecasts. The EnKF will be introduced for reducing the model error $\mathcal{E}_{DA+ML}$. The main contributions of the present work can be summarized as follows:

- A Multi-Input Multi-Output (MIMO) forecasting architecture is built in the DCGAN-ConvLSTM framework, which can make multi-step forecasts of temporal/dynamic information or prediction of any target variable.
- ConvLSTM and DCGAN both consist of convolutional LSTM layers, which can learn long-term dependency by LSTM and extract time-invariant features using convolutional layers. Thus, the two algorithms are powerful enough to efficiently and accurately exploit the intrinsic dynamics of the high-dimensional dynamic system.
- A recursively MIMO forecasting strategy is developed to perform long lead-time forecasts of any target variable.
- Importantly, the EnKF-based DCGAN-ConvLSTM framework enables real-time assimilation of observations into the ML-based forecast models, which allows us to minimize model errors and determine uncertainties, thus improving the accuracy of forecasts.

This paper is organized as follows. In Section 2, the details of an EnKF are briefly introduced, followed by the ConvLSTM model and DCGAN models in Section 3. The EnKF-based DCGAN-ConvLSTM forecasting architecture for long lead-time forecasting is presented in detail in Section 4. Detailed comparisons between the above-mentioned models appear in Section 5. Finally, the summary and conclusions are provided in Section 6.

2. Ensemble Kalman filter

EnKF is a Monte Carlo implementation of the Bayesian update problem, which uses ensembles of stochastic realizations (state vectors) for approximating the states of a dynamic system. The state matrix consists of all the state vectors in the ensemble as:

$$\mathcal{X}(t) = (\mathcal{X}_1(t), \dots, \mathcal{X}_k(t), \dots, \mathcal{X}_{N_e}(t)) \in \mathbb{R}^{N_x \times N_e}, \tag{2}$$

where $\mathcal{X}_k(t)$ is the state vector of the dynamic system, the subscript k is the number of the state vector in the ensemble, the subscript N_x represents the length of the state vector, the subscript N_e is the size of the ensemble.

EnKF-based data assimilation consists of two steps, prediction and update. In the prediction step, according to Eq. (1), the forecast model

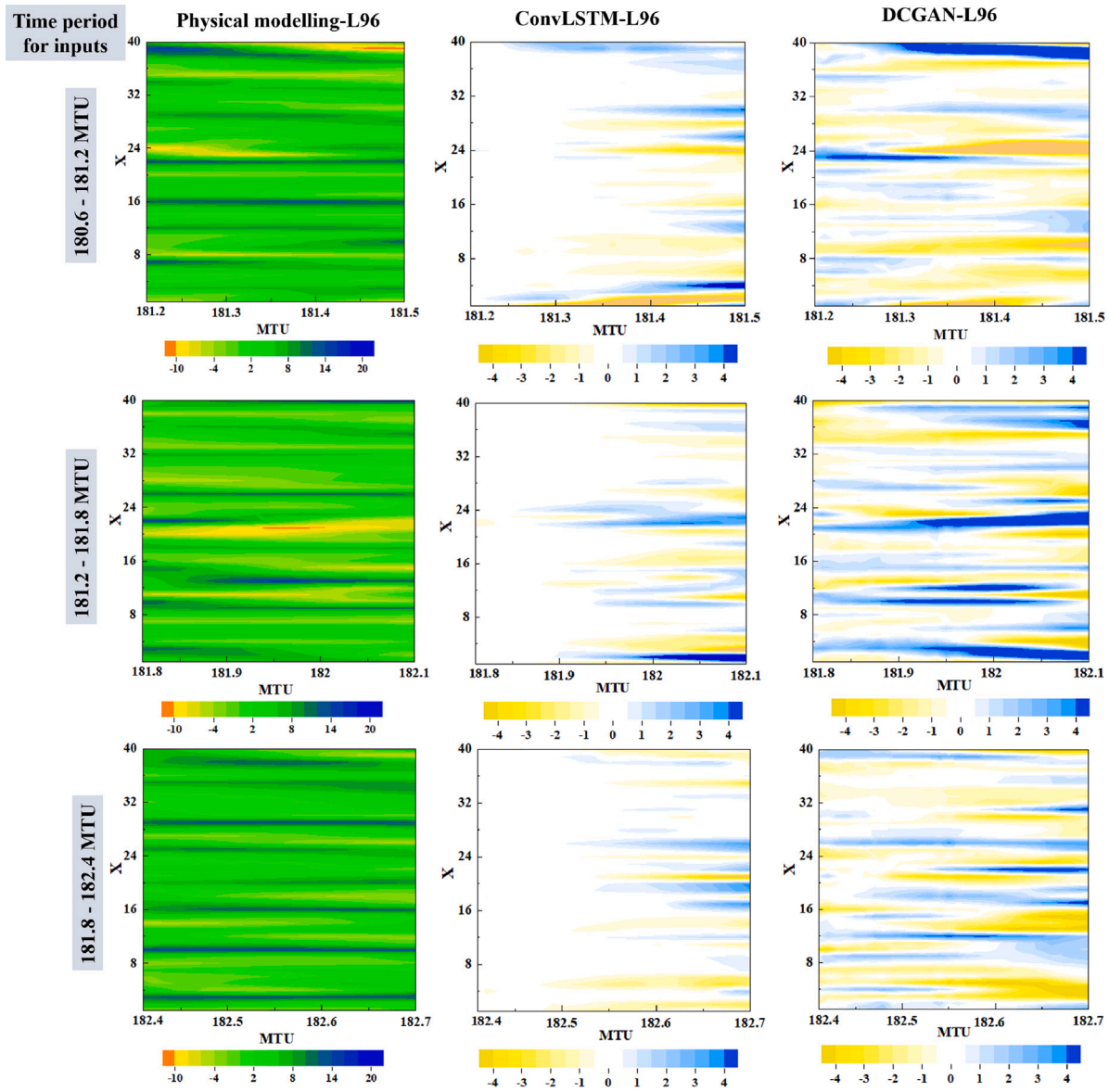


Fig. 5. Residuals of all states from the ConvLSTM and DCGAN models in comparison to those from the physical modeling-L96. Left column: full state trajectory from the physical modeling-L96; the difference of results between the ML model and physical modeling-L96: middle column-ConvLSTM and right column-DCGAN, where the data at the last-60-step time instances (shown in the left) as inputs while the corresponding forecast periods are 181.2 – 181.5 MTU, 181.8 – 182.1 MTU, and 182.4 – 182.7 MTU.

is performed on each ensemble member independently:

$$\mathcal{X}_k^f(t_s) = \mathcal{M}(\mathcal{X}_k^b(t_{s-1})) + \mathcal{E}_k(t_{s-1}), \tag{3}$$

where $\mathcal{M}$ is a forecast model, representing the dynamic system presented in Section 1, $\mathcal{X}_k^b(t_{s-1}) \in \mathbb{R}^{N_x}$ is the background vectors of all model states for the kth ensemble member, and $\mathcal{E}_k(t_s)$ denotes the white noises for the forecast model at time t_s. The introduction of noises $\mathcal{E}_k(t_s)$ enables EnKF to correctly propagate the uncertainty engaged in the system, which avoids a divergence of the systems due to a low variance with observation in propagation process [46]. The ensemble mean of forecasting vectors of all model states at time t_s is defined as:

$$\overline{\mathcal{X}}_k^f(t_s) = \frac{1}{N_e}\sum_{k=1}^{N_e}\mathcal{X}_k^f(t_s). \tag{4}$$

An estimate of the model error covariance P^f is then computed as:

$$P^f(t_s) = \frac{1}{N_e-1}\sum_{k=1}^{N_e}[\mathcal{X}_k^f(t_s) - \overline{\mathcal{X}}^f(t_s)][\mathcal{X}_k^f(t_s) - \overline{\mathcal{X}}^f(t_s)]^T. \tag{5}$$

In the update step, given the observation $\mathcal{Z}(t_s)$ each ensemble is then updated by:

$$\mathcal{X}_k^a(t_s) = \mathcal{X}_k^f(t_s) + K(t_s)[\mathcal{Z}(t_s) - \mathcal{H}\mathcal{X}_k^f(t_s)], \tag{6}$$

where $\mathcal{H}$ is the operator that converts the model states to observation space [26], and the Kalman gain is calculated by:

$$K(t_s) = (\rho_s \circ P^f(t_s))\mathcal{H}^T[\mathcal{H}(\rho_s \circ P^f(t_s))\mathcal{H}^T + \mathcal{R}(t_s)]^{-1}, \tag{7}$$

where $\mathcal{R} = \sigma^o\mathcal{I}$ is the observation error covariance matrix, and $\circ$ denotes a Schur product (an element-by-element multiplication). Here, the localization techniques are applied to ensemble assimilations [47–50]. The operation ρ_s is the correlation coefficient based on distance (the localization radius which is space and time-dependent) so that remote observations are removed. The (posterior) error covariance matrix $P^a(t_s)$ is estimated by:

$$P^a(t_s) = [1 - K(t_s)\mathcal{H}]P^f(t_s). \tag{8}$$

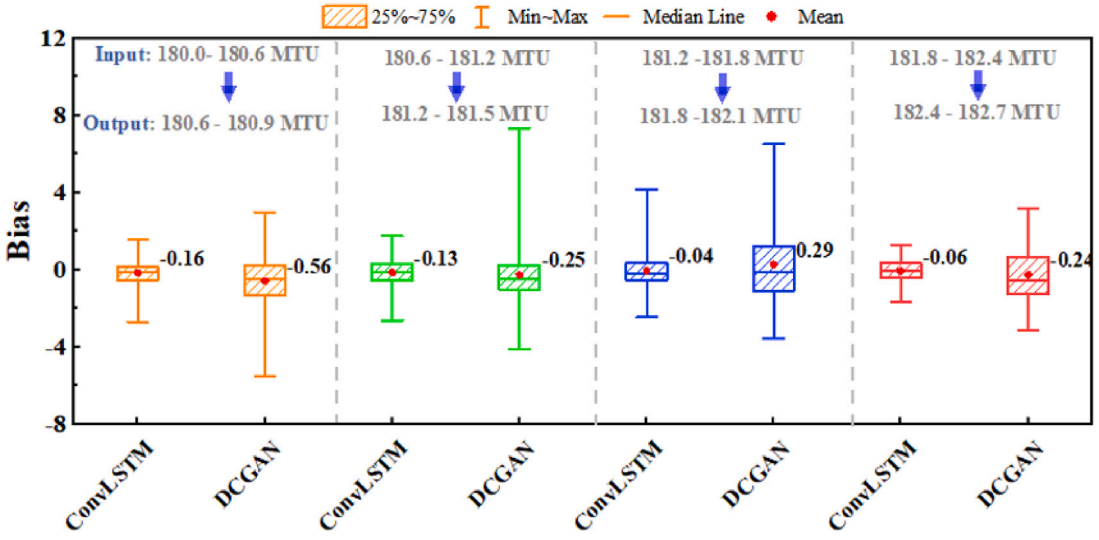


Fig. 6. The RMSE of predicted all states from the ConvLSTM and DCGAN models for the lead-time periods, 180.6 – 180.9 MTU, 181.2 – 181.5 MTU, 181.8 – 182.1 MTU, and 182.4 – 182.7 MTU.

3. Machine learning methods

3.1. ConvLSTM

The ConvLSTM model combines the advantages of CNN and LSTM, which can learn long-term dependency and extract time-invariant features [44].

The ConvLSTM architecture is similar to traditional LSTM architecture, consisting of a series of memory blocks. The key to the memory block is the cell state and three gates. The cell state C_s is the path of information transmission, which enables information to be transmitted in sequence. The gates are responsible for updating or discarding historical information. The cell contains three gates, which are input gate I_s, forget gate f_s, and output gate O_s. The input gate I_s processes the new data, the forget gate f_s is used to eliminate redundant or useless data and finally the output gate O_s processes the input data with the cell state. All the inputs, cell states, hidden states, and gates in the ConvLSTM are 3D tensors.

The following equations characterize the ConvLSTM model as:

$$f_s = \sigma(W_{xf} * \mathcal{X}_s + W_{hf} * h_{s-1} + W_{cf} \circ C_{s-1} + b_f), \tag{9}$$

$$I_s = \sigma(W_{xi} * \mathcal{X}_s + W_{hi} * h_{s-1} + W_{ci} \circ C_{s-1} + b_i), \tag{10}$$

$$\widetilde{C}_s = \tanh(W_{xc} * \mathcal{X}_s + W_{hc} * h_{s-1} + b_c), \tag{11}$$

$$C_s = f_s \circ C_{s-1} + I_s \circ \widetilde{C}_s, \tag{12}$$

$$O_s = \sigma(W_{xo} * \mathcal{X}_s + W_{ho} * h_{s-1} + W_{co} \circ C_s + b_o), \tag{13}$$

$$h_s = O_s \circ \tanh(C_s), \tag{14}$$

where W_i, W_o, W_f and W_c represent the weights for each gate, b_i, b_o, b_f and b_c are the bias terms, σ denotes the logistic sigmoid function, *tanh* is the hyperbolic tangent function, C_s represents the cell state, $\widetilde{C}_s$ is the updated cell state, $\mathcal{X}_s$ and h_s are the cell input and output respectively. In addition, the operator '$*$' stays instead of the vector multiplication. '$*$' is the convolution operation and '$\circ$' denotes the Hadamard product.

3.2. DCGAN

The DCGAN consists of two modules, a generator $\mathcal{G}$ and a discriminator D [45]. The discriminator D, is fed with samples from "real" data $\mathcal{X}_d$ and "fake" data (the reconstructed samples) $\widehat{\mathcal{X}}$ created by the generator $\mathcal{G}$: $\widehat{\mathcal{X}} = \mathcal{G}(\mathcal{X})$. The discriminator D tries to distinguish the $\widehat{\mathcal{X}}$ from $\mathcal{X}_d$ by making a binary decision. In contrast, the generator $\mathcal{G}$ aims at fooling the discriminator D into labeling $\widehat{\mathcal{X}}$ as a sample $\mathcal{X}_d$. Mathematically, this translates into the following minimization–maximization loss function $\mathcal{L}_{DCGAN}$:

$$\min_{\mathcal{G}} \max_{D} E_{\mathcal{X}_d \sim p_{data}(\mathcal{X}_d)}[\log D(\mathcal{X}_d)] + E_{\mathcal{X} \sim p_{\mathcal{X}}(\mathcal{X})}[\log(1 - D(\mathcal{G}(\mathcal{X})))], \tag{15}$$

where $p_{\mathcal{X}}(\mathcal{X})$ is a prior distribution for the input dataset $\mathcal{X}$, and $p_{data}(\mathcal{X}_d)$ is the corresponding probability data distribution for the targeted outputs $\mathcal{X}_d$.

Here an additional loss function, $L2$ function, has been introduced in the generator $\mathcal{G}$:

$$\mathcal{L}_G = E_{\mathcal{X} \sim p_{\mathcal{X}}(\mathcal{X})}[\log(1 - D(\mathcal{G}(\mathcal{X})))] + \lambda E_{\mathcal{X}_d \sim p_{data}(\mathcal{X}_d)} \|\mathcal{G}(\mathcal{X}) - \mathcal{X}_d\|_2^2, \tag{16}$$

where the hyperparameter λ is the $L2$ objective weight.

The final objective of the proposed DCGAN is:

$$\min_{\mathcal{G}} \max_{D} \mathcal{L}_{DCGAN} + \lambda E_{\mathcal{X}_d \sim p_{data}(\mathcal{X}_d)} \|\mathcal{G}(\mathcal{X}) - \mathcal{X}_d\|_2^2. \tag{17}$$

4. Long lead-time forecasting using hybrid machine learning models and EnKF

In this study, we aim at developing a ML-based model for the long lead-time forecast. In current ML models, it is still challenging to keep the predictive accuracy beyond the training period, not to mention long lead-time forecasts. Here a hybrid ML-based EnKF (ML-EnKF) framework is introduced to reduce the cumulative error in a long-term forecast. The main procedures are described below:

(a) The DCGAN and ConvLSTM models are first trained for efficient nonlinear dynamical prediction (Section 4.1, $\mathcal{M}_{ML}$ *mode*);
(b) The trained DCGAN and ConvLSTM models are then used for long lead-time forecasts in a recursive way, where the last ensemble multiple-time step-ahead forecasts/outputs are used as new inputs for the next time step-ahead ensemble forecasts (in Section 4.2, $\mathcal{M}_{ML}$ *mode*);
(c) The EnKF data assimilation approach is further incorporated into the DCGAN and ConvLSTM models for reducing forecast uncertainty of dynamic features at future time instants (in Section 4.3, $\mathcal{M}_{ML} + \mathcal{E}_{DA+ML}$ *mode*).

4.1. Multi-Input Multi-Output (MIMO) forecast using ConvLSTM and DCGAN

In this sub-section, the DCGAN and ConvLSTM models are trained for representing the physical dynamics shown in Eq. (3) ($\mathcal{M}_{ML}$ *mode*).

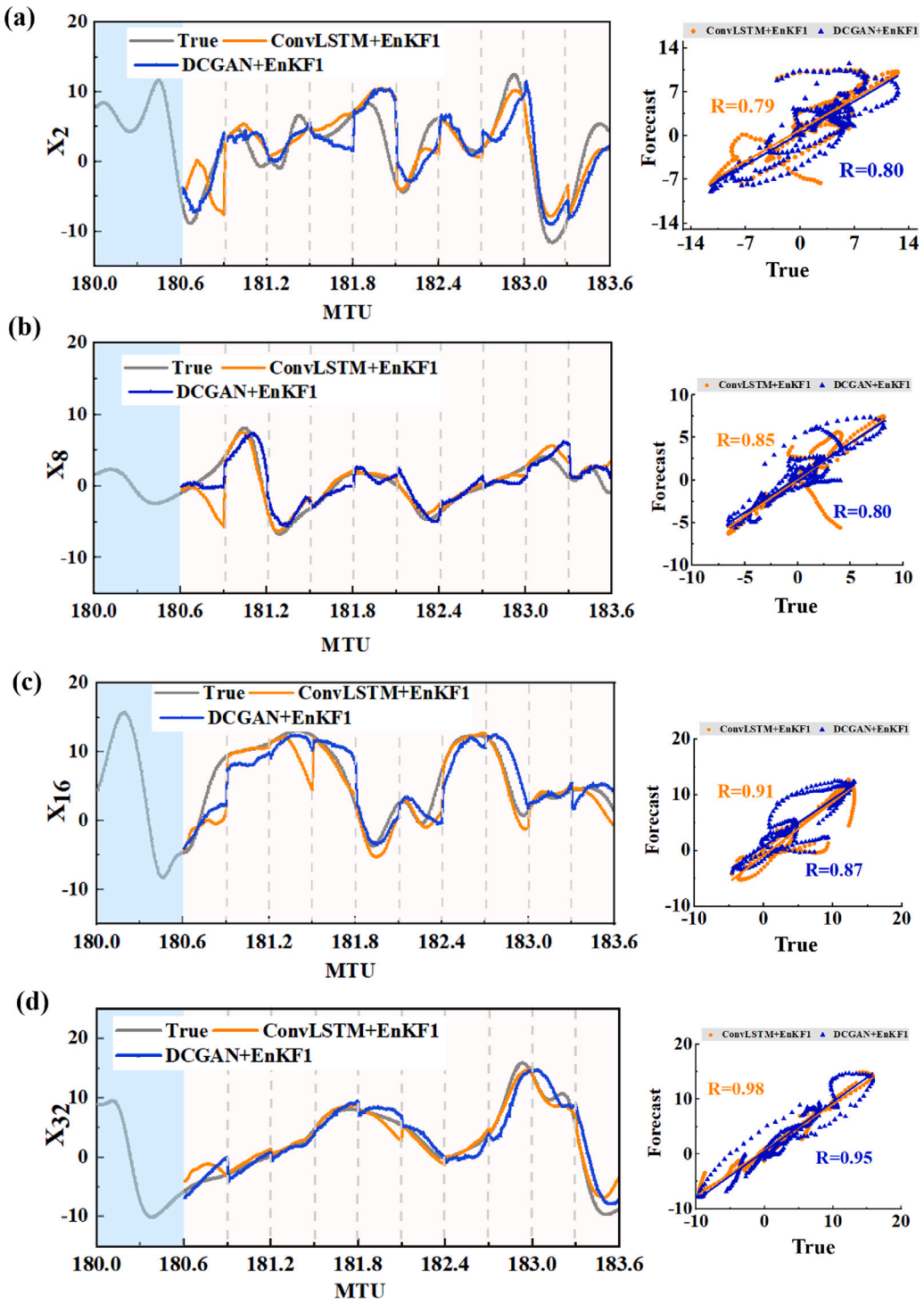


Fig. 7. The trajectories of X_2, X_8, X_{16} and X_{32} predicted using the ConvLSTM-EnKF1 and DCGAN-EnKF1, where the length of the known series is $M = 60$ (0.06 MTU) and the predicted series is $\Phi \times H = 300$ (3 MTU), i.e., 300-steps-ahead forecasts.

These ML models have replaced the physic models ($\mathcal{M}_{physical}$) as a forward surrogate model. For long lead-time forecasting, we adopt a Multi-Input Multi-Output (MIMO) strategy [51] as shown in Fig. 1(a). The MIMO forecasting is an estimation of future time series:

$$[\hat{\mathcal{X}}(t_{s+1}), \ldots, \hat{\mathcal{X}}(t_{s+H})] = \mathcal{M}_{ML}[\mathcal{X}(t_{s-M+1}), \ldots, \mathcal{X}(t_s)], \tag{18}$$

where $\mathcal{M}_{ML}$ is a ML forecasting model, $\hat{\mathcal{X}}$ represents the state variables, the inputs are the state values from the previous time steps $(t_{s-M+1}, \ldots, t_s)$, M is the length of time lag while the outputs are the forecasts at the lead-time steps $((t_{s+1}, \ldots, t_{s+H}) \in (t_0, t_{N_t}))$, H denotes the total predictive time steps. Here the ConvLSTM and DCGAN models acting as forecasting model $\mathcal{M}_{ML}$ are used for learning the input–output relationship.

Given N_w pairs of multi-inputs and multi-outputs $(\mathcal{X}_I, \mathcal{X}_O) \in (\mathcal{I}_{tr}, \mathcal{O}_{tr})$ used for ConvLSTM and DCGAN training purpose, the training inputs $\mathcal{X}_I$ are :

$$\mathcal{I}_{tr} : \mathcal{X}_I \in \mathbb{R}^{N_x \times M \times N_w}, \tag{19}$$

while the corresponding output datasets $\mathcal{X}_O$ are:

$$\mathcal{O}_{tr} : \mathcal{X}_O \in \mathbb{R}^{N_x \times H \times N_w}. \tag{20}$$

In this study, the ConvLSTM forecasting model consists of one input layer, two convolutional LSTM layers and two convolutional layers. The rectified linear unit (Leaky ReLU) activation function is employed between the layers. With respect to the DCGAN forecasting model, the generator $\mathcal{G}$ has one input layer, one convolutional LSTM layer and two convolutional layers. The discriminator $\mathcal{D}$ uses three hidden convolutional layers, each using a densely connected layer. The output layer is densely connected to the final hidden layer with a sigmoid activation function. Specifically, both ConvLSTM and DCGAN models are trained using the Adam optimizer [52]. Adam stands for adaptive moment estimation, a method of calculating current gradients using prior gradients. The Adam optimization approach is a stochastic gradient descent extension that has lately gained traction in computer vision and natural language processing applications. The weights in the networks are updated after computing the combined loss of each batch.

4.2. Long lead-time forecasts using ConvLSTM and DCGAN

In this sub-section, the long lead-time forecasts are obtained from the trained ConvLSTM and DCGAN in a recursively MIMO strategy ($\mathcal{M}_{ML}$ *mode*). Once the ConvLSTM and DCGAN training processes are completed, one can find the relationship between the input–output pairs $(\mathcal{X}_I, \mathcal{X}_O)$. By using the trained ConvLSTM and DCGAN models as shown in Fig. 1(b), the H-step predictive solutions beyond the training period $[t_0, t_{N_{tr}}]$ for any input data $\mathcal{X}_I \in (\mathcal{I}/\mathcal{I}_{tr})$ can be obtained in Eq. (18).

A recursively MIMO forecasting strategy [16] is here developed to make long lead-time forecasts, as shown in Fig. 1(c). The forecasts are given in Algorithm 1: where $\hat{\mathcal{M}}_{ML}$ represents the trained forecasting model, $\phi(\phi \in (1, \boldsymbol{\Phi}))$ is the length of steps using the MIMO strategy.

Algorithm 1 Multi-step MIMO forecasts using ConvLSTM and DCGAN.

(1) One MIMO step forecasting process.
- Set the length of time lags M and forecasting time steps H.
- Obtain the optimal parameters in the training process.
- Sample M steps datasets from the initial time t_s during the test period $(t_{N_{tr}}, t_{N_t})$.
- Make H-step ensemble forecasts in one MIMO step strategy by ConvLSTM and DCGAN models.

(2) Long lead-time forecast process.
- Select the iteration length $\boldsymbol{\Phi}$.

for $\phi = 1$ to $\boldsymbol{\Phi}$ **do**
  if $\phi = 1$ **then**
    • The next multi-steps-ahead is obtained by the trained model $\hat{\mathcal{M}}_{ML}$ as:
    $$\hat{\mathcal{M}}_{ML}[\mathcal{X}(t_{s-M+1}), \dots, \mathcal{X}(t_s)].$$
  else if $(H \times \phi) \le M$ **then**
    • The next multi-steps-ahead is obtained by the trained model $\hat{\mathcal{M}}_{ML}$ as:
    $$\hat{\mathcal{M}}_{ML}[\mathcal{X}(t_{s-M-2+H(\phi-1)}), \dots, \mathcal{X}(t_s), \hat{\mathcal{X}}(t_{s+1}), \dots, \hat{\mathcal{X}}(t_{s+H(\phi-1)})].$$
  else
    • The next multi-steps-ahead is obtained by the model $\hat{\mathcal{M}}_{ML}$ as:
    $$\hat{\mathcal{M}}_{ML}[\hat{\mathcal{X}}(t_{s-M+1+H(\phi-1)}), \dots, \hat{\mathcal{X}}(t_{s+H(\phi-1)})].$$
  end if
end for
- Obtain the long length-$\boldsymbol{\Phi} \times H$ multi-steps of forecasts $\hat{\mathcal{X}}$.

4.3. Extension of forecasting horizon integrating ConvLSTM and DCGAN with EnKF

In the sections above, the DCGAN and ConvLSTM models have been developed for efficient forecasts. However, the performance of the trained ML forecasting models decays as the predictive horizon $\boldsymbol{\Phi}$ extends beyond a few time steps. To reduce the cumulative error in long lead-time forecasts, we introduce here the data assimilation technique (the EnKF approach) to the ConvLSTM and DCGAN models in the online forecasting process ($\mathcal{M}_{ML} + \mathcal{E}_{DA+ML}$ *mode*), as shown in Fig. 1(c).

As stated in Section 2, the EnKF is a sequential ensemble data assimilation technique. Given the background state $\mathcal{X}^b$ ($\mathcal{X}^b = [\mathcal{X}(t_{s-M+1}), \dots, \mathcal{X}(t_s)]$), the initial ensembles at time t_s ($t_s \in (t_{N_{tr}}, t_{N_t})$) are generated by:

$$\mathcal{X}^b_k(t_s) = \mathcal{X}^b(t_s) + \epsilon, \tag{21}$$

where ϵ is randomly obtained from a Gaussian distribution $\mathcal{N}(\mathcal{X}^b, \sigma^2)$, $k \in (1, N_e)$ and N_e is the size of the ensemble.

The forecast matrix in Eq. (3) from trained ML-based models at time t_s is defined by

$$\hat{\mathcal{X}}^f_k(t_s) = [\hat{\mathcal{X}}^f_{k,1}(t_s), \dots, \hat{\mathcal{X}}^f_{k,s}(t_s), \dots, \hat{\mathcal{X}}^f{}_{k,N_x}(t_s)] \in \mathbb{R}^{N_x \times N_e}. \tag{22}$$

Given the observations $\mathcal{Z}$ and the forecast ensemble $\hat{\mathcal{X}}^f_k$, the forecast results $\hat{\mathcal{X}}^a_k$ can be updated by following the steps shown in Algorithm 2:

Algorithm 2 Multi-step MIMO forecasts using the hybrid ConvLSTM-EnKF and DCGAN-EnKF.

Set the iteration length $\boldsymbol{\Phi}$.
Set the length of the time lag M and forecasting time step H in one MIMO step.
Set the ensemble sample size N_e.
(1) Inputs:
- Generate the ensembles $\mathcal{X}^b$ at the initial forecast time t_s.
- Set observations $\mathcal{Z}$.
- Set $\hat{\mathcal{X}}^f(t_s) = \mathcal{X}^b(t_s)$

for $\phi = 1$ to $\boldsymbol{\Phi}$ **do**
  (2) Prediction step:
  • Get the dynamic features from the trained models in Eq. (3): $\hat{\mathcal{X}}^f(t_{N_\phi}) = \hat{\mathcal{M}}_{ML}(\hat{\mathcal{X}}^f(t_{N_{\phi-1}}))$.
  • Obtain the model error covariance P^f in Eq. (5).
  for $i = 1$ to H **do**
    (3) Update step:
    • Calculate the Kalman gain $K(t_i)$ in Eq. (7).
    • Update the ensemble of states $\hat{\mathcal{X}}^a(t_i)$ in Eq. (6).
    • Update the error covariance matrix $P^a(t_i)$ in Eq. (8).
  end for
end for
- Obtain the updated length-$\boldsymbol{\Phi} \times H$ multi-steps of forecasts $\hat{\mathcal{X}}^a$.

5. Numerical examples

In this section, our two-hybrid ConvLSTM-EnKF and DCGAN-EnKF models are tested with one-level and two-level variants of the Lorenz 96 models, which have been commonly used as a prototypical test case in data assimilation [10,31,36,53].

5.1. One-level Lorenz 96

The one-level Lorenz 96 model is defined on a periodic one-dimensional domain by the following set of ordinary differential equations:

$$\frac{d\mathcal{X}_r}{dt} = (\mathcal{X}_{r+1} - \mathcal{X}_{r-2})\mathcal{X}_{r-1} - \mathcal{X}_r + F \tag{23}$$

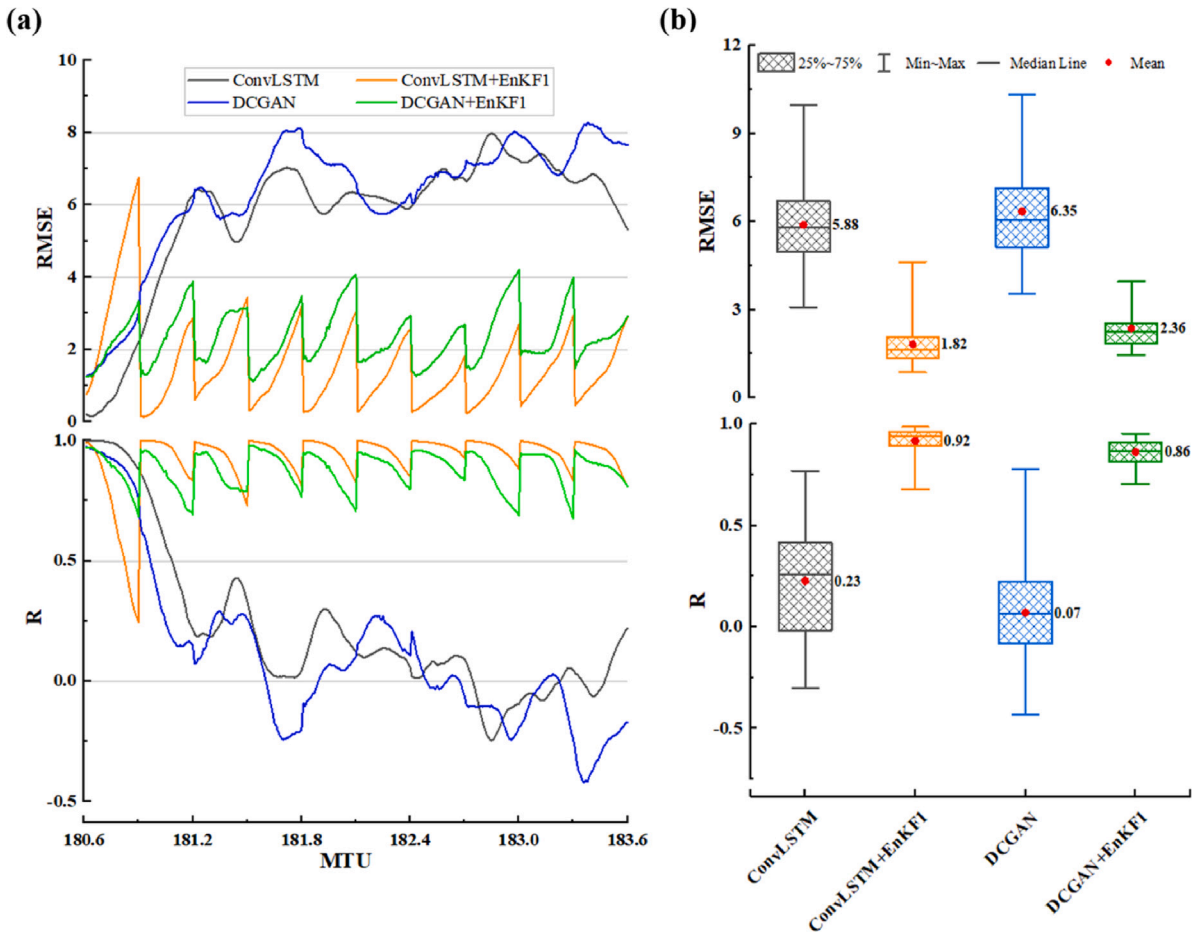


Fig. 8. The RMSEs and correlation coefficients of the ConvLSTM, DCGAN, ConvLSTM-EnKF1, and DCGAN-EnKF1 models during the forecasting period of 180.6 – 183.6 MTU, where (a) the spatial-averaged RMSEs and (b) the temporal-averaged RMSEs. Note that EnKF is not introduced in the ConvLSTM and DCGAN models while EnKF is used for updating the inputs in ConvLSTM-EnKF1 and DCGAN-EnKF1.

where $\mathcal{X}_r$ is the scalar state variable measured along a circle of constant latitude of the earth divided into N_x equal-sized sectors, $r \in [0, N_x]$, $\mathcal{X}_{N_x} = \mathcal{X}_0$, $\mathcal{X}_{N_x-1} = \mathcal{X}_1$, $\mathcal{X}_{N_x-2} = \mathcal{X}_2$, $N_x = 40$. The external forcing $F = 12$ is chosen as a strongly supercritical value to render the system sufficiently chaotic.

5.1.1. Model setup

A long "truth" run of the physical Lorenz 96 model (referred to as physical modeling-L96) is performed to generate the datasets for training and validating the ConvLSTM-EnKF and DCGAN-EnKF models. The "truth" run is integrated for 200 model time units (MTUs) using a time step δt= 0.01 MTU. Outputs from the first 180 MTUs are used for training (90%) and validation (10%), and the remaining 20 MTUs are used for testing. Therefore, we assess the performance of the trained models for temporal integration during the period $[180, 200]$ MTUs. The training process of the DCGAN, consisting of a generator and a discriminator, is more complex than ConvLSTM. Fig. 2 shows the discriminator scores on real (blue) and fake (orange predicted values) variable values vary during the DCGAN training. It can be noticed that the DCGAN model stabilizes between epochs 150 and 500. After epoch=150, the discriminator scores for real and fake samples remain stable and are slightly beyond or around 0.5 with increased epochs.

The performances of two trained ML models developed in this study are assessed using the root mean squared error (RMSE), and correlation coefficient (R) which are defined as follows:

$$RMSE = \sqrt{\frac{\sum_{j=1}^{N} (\hat{\mathcal{X}}_j - \mathcal{X}_{d,j})^2}{N}}, \tag{24}$$

$$R = \frac{\sum_{j=1}^{N} (\hat{\mathcal{X}}_j - \overline{\hat{\mathcal{X}}})(\mathcal{X}_{d,j} - \overline{\mathcal{X}}_d)}{\sqrt{\sum_{j=1}^{N} (\hat{\mathcal{X}}_j - \overline{\hat{\mathcal{X}}})^2} \sqrt{\sum_{j=1}^{N} (\mathcal{X}_{d,j} - \overline{\mathcal{X}}_d)^2}}, \tag{25}$$

where the subscript j represents the pairing of N targeted states $\mathcal{X}_d$ and forecasts $\hat{\mathcal{X}}$ by points ($[0, N_x]$) and predictive period ($[t_{N_{tr}}, t_{N_t}]$), and the overbars signify means over points and/or time.

5.1.2. One step MIMO forecast

Fig. 3 illustrates a series of forecasts from the trained ConvLSTM and DCGAN models in one-step MIMO modeling. Here we use the last 60-time-step variable values as known inputs and obtain a 30-time-step-ahead forecast of all state variables ($N_x = 40$). The "true" results or references are computed using the physical modeling-L96. It can be seen that the ConvLSTM model performs better than the DCGAN model in one-step MIMO forecasting where the correlation coefficients of ConvLSTM results are higher than 0.97 except for X_{32}. This can be further confirmed in Fig. 4, which compares the forecasts of all state variables from both the trained ML models during the lead-times (180.6 – 180.9 MTUs). It can be noticed that the residuals (bottom panel) in Fig. 4 of the ConvLSTM are smaller than those of the DCGAN model. In general, the predictive results from the ConvLSTM model have achieved a better match with the ground true values at the lead-times (180.6 – 180.9 MTUs) than that of the DCGAN model. The main reason for this is that the loss function in the GAN mainly measures the distance between the probability distributions unlike ConvLSTM where the loss function measures the misfit between the predicted and

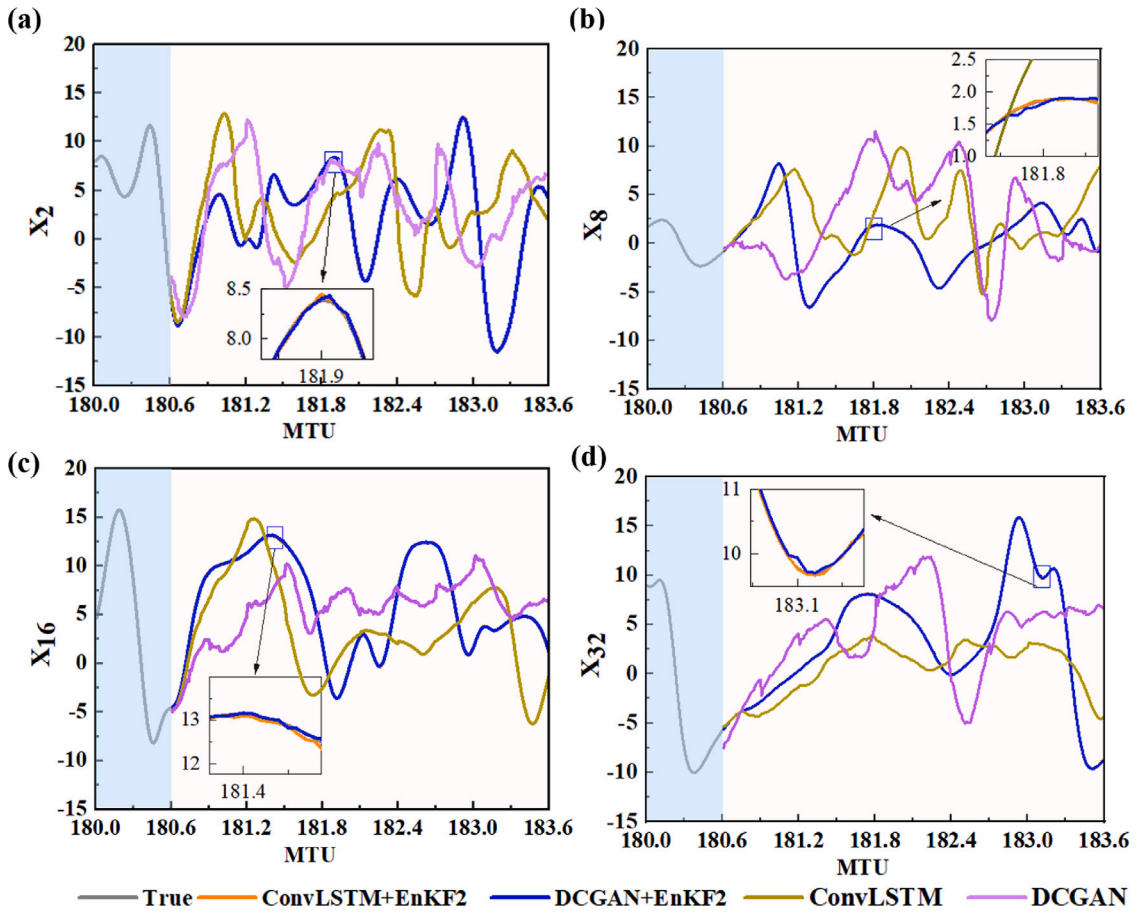


Fig. 9. Same as Fig. 7 except for using the ConvLSTM-EnKF2 and GAN-EnKF2 models, where both inputs and outputs obtained from the trained ML models are updated using EnKF.

true values. Therefore, the trained DCGAN helps us to analyze, forecast and compute the probability of variable values, based the past datasets with respect to changing time. The capability of the DCGAN in real-time prediction varies from one case to another. In the study case of multiple time-step predictions in physical modeling-L96, compared to the DCGAN, the ConvLSTM is more powerful in forecasting.

The accuracy of results from the ConvLSTM and DCGAN models has further been estimated during different predictive periods (shown in Figs. 5 and 6), where the references are computed using the physical modeling-L96. In Fig. 5, we can see that the residuals from both the ConvLSTM and DCGAN models are mostly smaller during these predictive periods. Fig. 6 shows the RMSEs of results from both the ConvLSTM and DCGAN models for four predictive periods. The box plots depict the median, the 25th and 75th percentiles and the minimum and maximum computed over four predictive periods. The bias of mean values is less than -0.16 and -0.56 for the ConvLSTM and DCGAN models respectively at different predictive periods. Again, the ConvLSTM model attains a closer agreement with the results from the physical modeling-L96 than that of the DCGAN model. The ConvLSTM model can better detect the underlying functional mapping through pattern extraction and long-term dependency learning.

5.1.3. Long lead-time forecasts with ML-EnKF

In this section, we further demonstrate the forecasting performance of the ConvLSTM and DCGAN models combined with EnKF in multi-step-MIMO forecasting. The EnKF is introduced for improving the accuracy of long-term forecasting. The forecast period is $[180.6, 183.6]$ MTU. In MIMO forecast (in Eq. (18)), the total forecasting time steps in one iteration/step are set to $H = 30$ while the length of previous time steps for inputs is $M = 60$ (where the time step size is 0.01 MTU). The iteration number Φ of multi-step MIMO is 10, i.e. the long-term forecast period (300-steps-ahead) is $10 \times (0.01 \times 30) = 3$ MTUs. The initial conditions $\mathcal{X}^b$ are the state variable values from 180 MTU (in Algorithm 2). Given a perturbation (randomly drawn from a Gaussian distribution with a zero mean) to the initial variable $\mathcal{X}^b$ (in Eq. (21)), the initial ensembles are generated, where the ensemble sample size is 50 ($N_e = 50$). Each ensemble member is integrated in time using the trained ConvLSTM and DCGAN models.

In ML-EnKF models, the forecasts at the lead-time steps (i.e., $\hat{\mathcal{X}}^f(t_{s+1}, \ldots, t_{s+H})$ in Algorithm 2) are obtained from the trained ConvLSTM and DCGAN models. For the next-MIMO-step forecasting, the inputs are the forecasts $\hat{\mathcal{X}}^f(t_s)$ at the previous time steps $(t_{s-M+1}, \ldots, t_s)$, which are updated using the EnKF whenever an ongoing sequence of observations are available. In comparison to the true values (from physical modeling-L96), the performance of the ConvLSTM-EnKF and DCGAN-EnKF are evaluated through:

(a) *the forecast state variables* during the lead-time period using the ConvLSTM and DCGAN, where the inputs are the updated forecast results using EnKF (hereafter referred to ConvLSTM-EnKF1 and DCGAN-EnKF1);
(b) *the updated forecast results* during the lead-time period, where both outputs and inputs in the ConvLSTM and DCGAN models are updated using EnKF (hereafter referred to ConvLSTM-EnKF2 and DCGAN-EnKF2). At this step, the outputs (forecast states) from ConvLSTM-EnKF1 and DCGAN-EnKF1 are further updated using EnKF.

As shown in Fig. 7, the trajectories of X_2, X_8, X_{16} and X_{32} predicted using the ConvLSTM-EnKF1 and DCGAN-EnKF1 follow the trend of the true ones during the whole lead-times (180.6 – 183.6 MTU). It can be

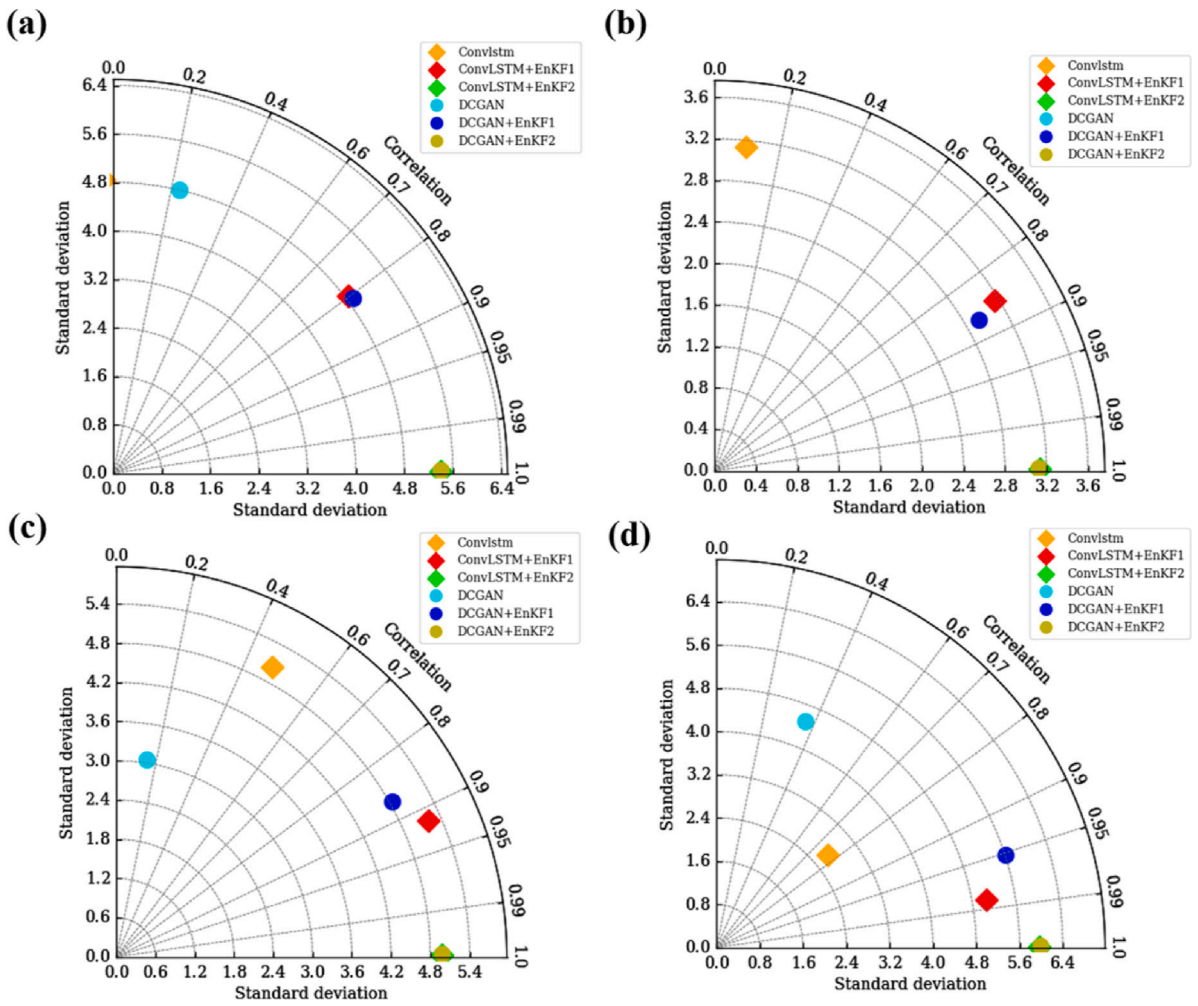


Fig. 10. Tylor diagrams representing correlation coefficient together with the standard deviation difference for proposed ConvLSTM, ConvLSTM+EnKF1, ConvLSTM+EnKF2, DCGAN, DCGAN+EnKF1, DCGAN+EnKF2 for (a) X_2 (b) X_8 (c) X_{16}, and (d) X_{32} state variables during the forecasting period of 180.6 – 183.6 MTU.

observed that a relatively good agreement exists between the forecasts and true values and small differences can be noticed at forecasting steps. In general, the underlying dynamic pattern is captured well by the predictive ConvLSTM-EnKF1 and DCGAN-EnKF1. During the 300-steps-ahead forecasting period, the correlation coefficients of four state variables are between 0.79 and 0.98 for the ConvLSTM and between 0.8 and 0.95 for the DCGAN respectively.

To further demonstrate the forecasting capability of the hybrid models, the error analysis is carried out by the RMSE and correlation coefficient of predicted state variables. As shown in Fig. 8, it can be noticed that the RMSEs of forecasting states are reduced when the inputs for the ConvLSTM and DCGAN models are updated with EnKF. For example, the RMSEs of the ConvLSTM and DCGAN models have a rising trend (with a maximum value of 8) during the predictive period [180.6, 183.6] MTU in Fig. 8(a), while for the ConvLSTM-EnKF1 and DCGAN-EnKF1 the values fluctuate between 0 and 4 except for the initial 30-step forecasts (180.6−180.9 MTU). The correlation coefficients of ConvLSTM-EnKF1 and DCGAN-EnKF1 are around 0.75 – 0.9, while without EnKF, the correlation coefficients are down to 0.2. The introduction of EnKF to the ConvLSTM and DCGAN models has mitigated the cumulated forecast error problem in recursively MIMO forecasting. This can be seen in Fig. 8(b) that the mean RMSE of the ConvLSTM model decreases from 5.88 to 1.82 after the observations are assimilated using EnKF, while that from the DCGAN model declines from 6.35 to 2.36. The mean correlation coefficients of the ConvLSTM-EnKF1 and DCGAN-EnKF1 models are 0.92 and 0.86 respectively.

The accuracy of forecast states during the lead-times can further be enhanced by updating the outputs from the ConvLSTM-EnKF1 and DCGAN-EnKF1 using EnKF. Fig. 9 depicts the forecasts of X_2, X_8, X_{16} and X_{32} from the ConvLSTM-EnKF2 and DCGAN-EnKF2 during the whole lead-time (180.6 – 183.6 MTU). Visually, very little difference between the true values and results from the ConvLSTM-EnKF2 and DCGAN-EnKF2 can be noticed. The results demonstrate that the two-hybrid ConvLSTM-EnKF and DCGAN-EnKF models have successfully captured dynamic features and provide accurate long lead-time forecasts. This can serve as a promising tool by using the hybrid ML-EnKF model to efficiently forecast complex dynamic systems for realistic applications.

A Taylor diagram in Fig. 10 is employed for further assessment of results between the ConvLSTM, ConvLSTM+EnKF1, ConvLSTM+EnKF2, DCGAN, DCGAN+EnKF1, DCGAN+EnKF2 and physical modeling-L96 for X_2, X_8, X_{16}, and X_{32} state variables during the forecasting period of 180.6 – 183.6 MTU. It can be noticed that the results from the hybrid models (i.e., ConvLSTM+EnKF1, ConvLSTM+EnKF2, DCGAN+EnKF1, DCGAN+EnKF2) display a higher correlation coefficient than that of the ConvLSTM and DCGAN models (orange and cyan points), for example, close to 1.0 for the ConvLSTM+EnKF2 and DCGAN+EnKF2 (green and yellow points).

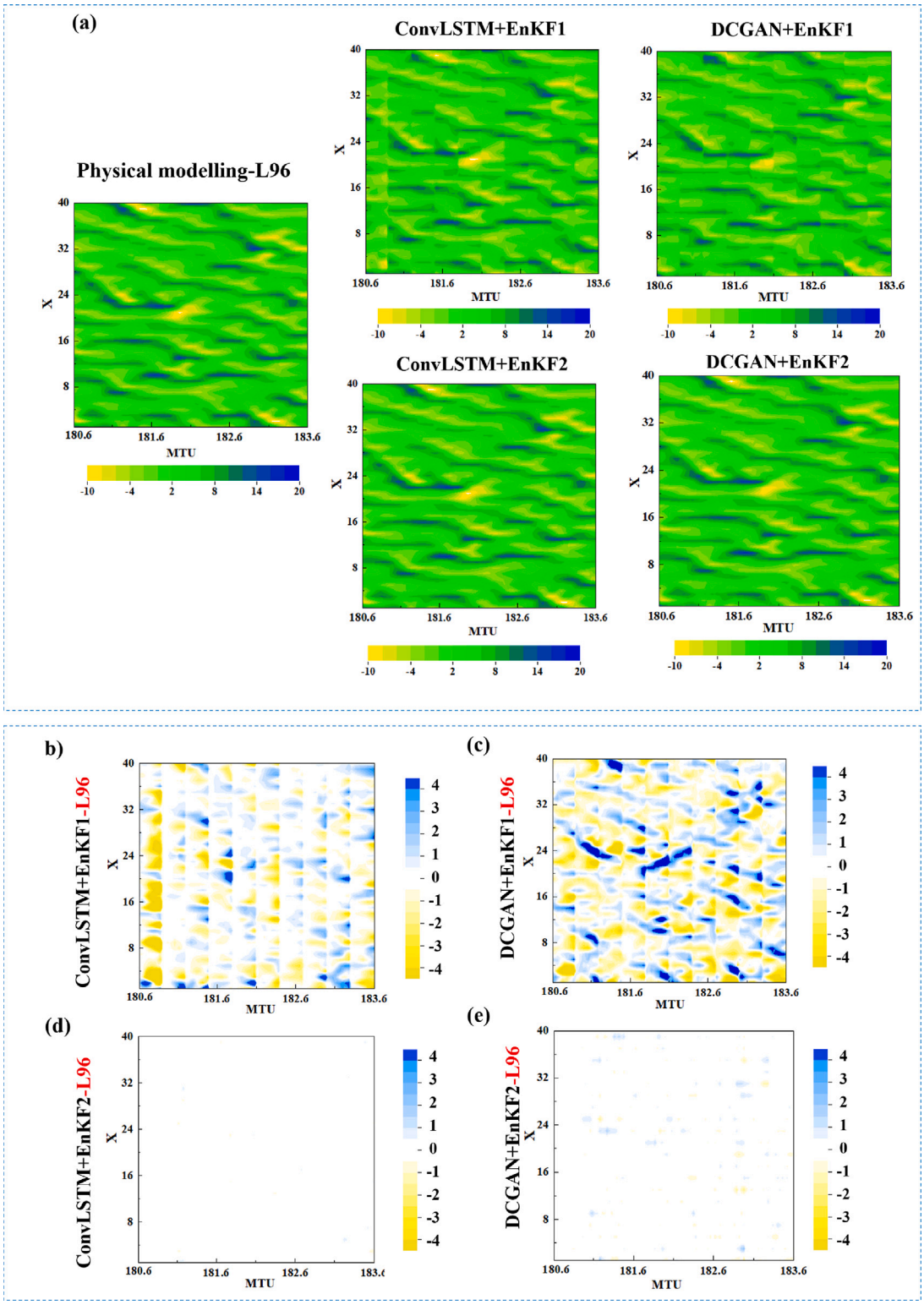


Fig. 11. Comparison of 300-step-ahead forecast trajectories of full states between the ConvLSTM-EnKF1, ConvLSTM-EnKF2, DCGAN-EnKF1 and DCGAN-EnKF2 models with physical modeling-L96 during the forecasting period of 180.6 – 183.6 MTU.

To further compare the performance of the hybrid ML-EnKF models, the trajectories of full states from ConvLSTM-EnKF1, ConvLSTM-EnKF2, DCGAN-EnKF1 and DCGAN-EnKF2 models are presented in Fig. 11(a). It can be seen that the predictive maps from the four models attain a closer agreement to that of the physical modeling-L96. The differences in forecast states between the ML models and physical modeling-L96 are illustrated in Fig. 11(b)–(e). It can be noticed that the differences are quite small over the whole domain, especially for ConvLSTM-EnKF2 and DCGAN-EnKF2 models. The bias of ConvLSTM-EnKF2 and DCGAN-EnKF2 models is hardly visible, which suggests the ConvLSTM-EnKF2 and DCGAN-EnKF2 models are capable to obtain accurate long lead-time forecasts for complex physical dynamics.

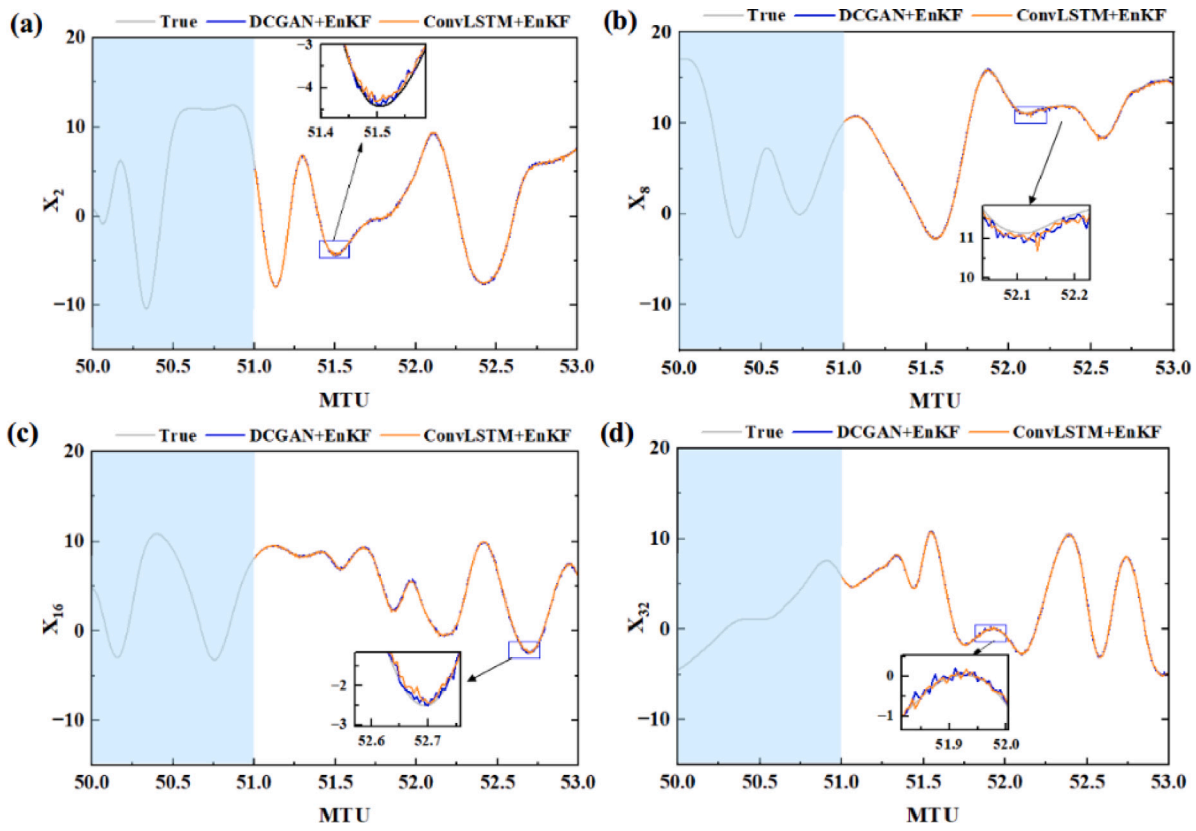


Fig. 12. The trajectories of X_2, X_8, X_{16} and X_{32} predicted using the ConvLSTM-EnKF and DCGAN-EnKF, where the length of the known series is $M = 200$ (1 MTU) and the predicted series is $\Phi \times H = 400$ (2 MTU), i.e., 400-steps-ahead forecasts.

5.2. Two-level Lorenz 96

To further demonstrate the model forecasting capability, a two-level Lorenz 96 model is adopted as a test case. The two-level Lorenz 96 model has two scales of variables, $\mathcal{X}$ and $\mathcal{Y}$ with low and high frequency respectively. Each of $\mathcal{X}_r$ are associated with $\mathcal{Y}_{r,k}$ variables representing unresolved subgrid processes:

$$\frac{d\mathcal{X}_r}{dt} = -(\mathcal{X}_{r+1} - \mathcal{X}_{r-2})\mathcal{X}_{r-1} - \mathcal{X}_r + F - \frac{hc}{b}\sum_{k=1}^{K}\mathcal{Y}_{k,r}, \tag{26}$$

$$\frac{d\mathcal{Y}_{k,r}}{dt} = -cb\mathcal{Y}_{k+1,r}(\mathcal{Y}_{k+2,r} - \mathcal{Y}_{k-1,r}) - c\mathcal{Y}_{k,r} - \frac{hc}{b}\mathcal{X}_r, \tag{27}$$

where the number of $\mathcal{X}$ variables is $N_x = 36$, and the number of $\mathcal{Y}_{k,r}$ variables per $\mathcal{X}_r$ is $K = 10$. The value of the constants is set to $h = 1$, $b = 10$ and $c = 10$. The forcing is set at $F = 20$.

The solutions in the two-level Lorenz 96 are treated as the "true values". The high-frequency $\mathcal{Y}$ variables in Eq. (27) can be calculated using the subgrid parameterization modeling in weather and climate prediction [10]. Our goal is to train the machine learning model for the prediction of the low-frequency variable $\mathcal{X}$. Therefore, the Eq. (18) is rewritten as:

$$[\hat{\mathcal{X}}(t_{s+1}), \ldots, \hat{\mathcal{X}}(t_{s+H})] = \mathcal{M}_{ML}[\mathcal{X}(t_{s-M+1}), \mathcal{U}(t_{s-M+1}), \ldots, \mathcal{X}(t_s), \mathcal{U}(t_s)], \tag{28}$$

where

$$\mathcal{U} = \frac{hc}{b}\sum_{k=1}^{K}\mathcal{Y}_k. \tag{29}$$

The model training and validation period is $[0, 50]$ MTU, and the forecast period is $[51, 53]$ MTU. In MIMO forecast (in Eq. (18)), the total forecasting time steps in one iteration/step are set to $H = 50$ while the length of previous time steps for inputs is $M = 200$ (where the time step size is 0.005 MTU). The iteration number Φ of multi-step MIMO is 8, i.e. the long-term forecast period (400-steps-ahead) is $8\times(0.005\times50) = 2$ MTUs. The initial conditions $\mathcal{X}^b$ are the state variable values from 50 MTU (in Algorithm 2). The ensemble sample size is 50 ($N_e = 50$). Each ensemble member is integrated in time using the trained ConvLSTM model.

As shown in Fig. 12, the trajectories of X_2, X_8, X_{16} and X_{32} predicted using the ConvLSTM-EnKF and DCGAN-EnKF follow the trend of the true ones during the whole lead-times (51 – 53 MTU). It is noticed that the forecasts from both ConvLSTM-EnKF and DCGAN-EnKF have good agreement with true values and very small differences can be observed at forecasting steps.

To further compare the performance of the hybrid ML-EnKF models, the trajectories of full states from ConvLSTM-EnKF and DCGAN-EnKF models are presented in Fig. 13. It can be seen that the predictive maps from both models attain a closer agreement to that of the two-level Lorenz 96 model (physical modeling-L96), and very little difference between the true values and results from the ConvLSTM-EnKF and DCGAN-EnKF can be ascertained visually. According to the bias in Fig. 13, it can be seen that the biases are quite small over the whole domain, especially for the ConvLSTM-EnKF model. The results demonstrate that the ConvLSTM-EnKF and DCGAN-EnKF models are capable to perform accurate long lead-time forecasts for complex dynamic systems. Using ConvLSTM-EnKF and DCGAN-EnKF models for realistic applications will be the subject of our future work.

6. Conclusions

In this work, a hybrid ML-EnKF framework has first been developed for long lead-time forecasts of nonlinear dynamic systems ($\mathcal{M}_{ML} + \mathcal{E}_{DA+ML}$ *mode*). The ConvLSTM and DCGAN have been trained for learning the complex nonlinear relationships between the past and

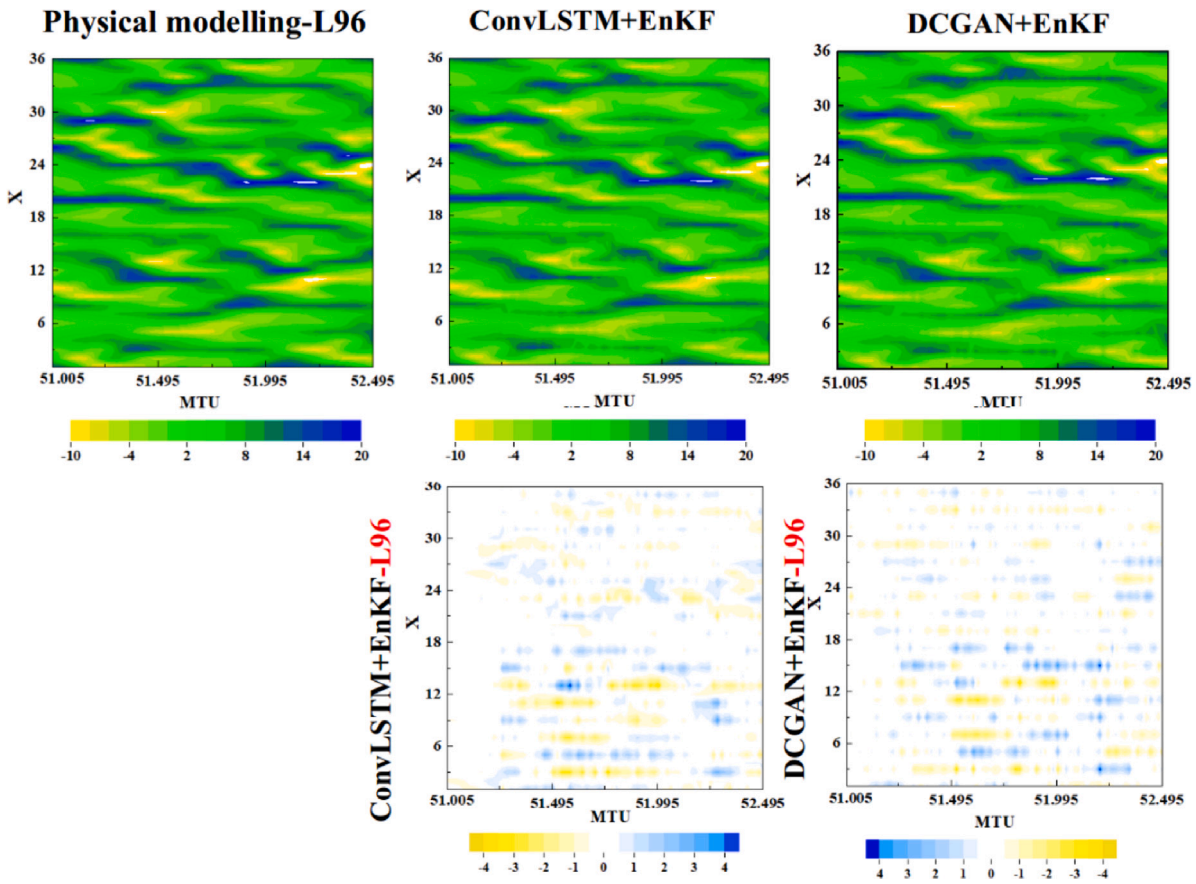


Fig. 13. Comparison of 400-step-ahead forecast trajectories of full states between the ConvLSTM-EnKF and DCGAN-EnKF models with physical modeling-L96 during the forecasting period of 51.005 – 53 MTU.

future states, and then used as the dynamic surrogate model in long-term forecasts based on an iterative MIMO algorithm. During each iteration MIMO forecast stage, the trained ML models are used for multiple time-step-ahead forecasts while the forecast results are updated using EnKF whenever the observations are available during the online forecast process. In comparison to traditional EnKF approaches, the trained ConvLSTM and DCGAN models are used for efficient calculation of ensembles, rather than a physics-informed model which requires a high computational cost.

The performance of the hybrid ConvLSTM-EnKF and DCGAN-EnKF models has been demonstrated by one-level and two-level Lorenz 96 models. To evaluate the model forecasting capability, a detailed accuracy assessment has been performed through the correlation coefficient and RMSE. Our results illustrate that the ConvLSTM model can better capture temporal dynamics in a high-dimensional nonlinear system and predict accurate dynamics than the DCGAN model. In addition, the use of EnKF in ConvLSTM and DCGAN models significantly reduces forecast errors along with lead-times.

Overall, the hybrid ML-EnKF ($\mathcal{M}_{ML} + \mathcal{E}_{DA+ML}$ *mode*) is an efficient and reliable tool for real-time forecasting of dynamic systems. Compared to physical models, the use of ML approaches can reduce the computational cost by several orders of magnitude, thus speeding up forecast and data assimilation processes significantly. It has a promising potential for a wide range of applications, for instance, flooding and real-time air pollution forecasting. We remark that our hybrid forecasting ML-EnKF model of sequentially updating the forecasts with the incoming observations is not restricted to EnKFs. One may instead employ other nonlinear filtering methods such as variational data assimilation approaches. The accuracy and efficiency of the proposed ML-DA forecasting model applied to realistic and complex applications will be the subject of our future work.

CRediT authorship contribution statement

Meiling Cheng: Data curation, Investigation, Writing – original draft. **Fangxin Fang:** Supervision, Methodology, Writing – review & editing, Funding acquisition. **Ionel M. Navon:** Writing – review & editing. **Christopher Pain:** 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.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the China Scholarship Council, China (No. 201806270238) and Engineering and Physical Sciences Research Council (EPSRC), UK (MAGIC (EP/N010221/1), INHALE (EP/T003189/1), MUFFINS (EP/P033148/1) and PREMIERE (EP/T000414/1)) in the UK. The authors are grateful for the support of the Imperial College ICT service. The authors acknowledge the reviewers and Editor for their in depth perspicacious comments that contributed to improving the presentation of this paper.

References

[1] W.-P. Tsai, D. Feng, M. Pan, H. Beck, K. Lawson, Y. Yang, J. Liu, C. Shen, From calibration to parameter learning: Harnessing the scaling effects of big data in geoscientific modeling, Nature Commun. 12 (1) (2021) 1–13.

[2] S.C. James, Y. Zhang, F. O'Donncha, A machine learning framework to forecast wave conditions, Coast. Eng. 137 (2018) 1–10.

[3] M. Cheng, F. Fang, I.M. Navon, C. Pain, A real-time flow forecasting with deep convolutional generative adversarial network: Application to flooding event in Denmark, Phys. Fluids 33 (5) (2021) 056602.

[4] J. Fu, D. Xiao, D. Li, H.R. Thomas, C. Li, Stochastic reconstruction of 3D microstructures from 2D cross-sectional images using machine learning-based characterization, Comput. Methods Appl. Mech. Engrg. 390 (2022) 114532.

[5] S.L. Brunton, B.R. Noack, P. Koumoutsakos, Machine learning for fluid mechanics, Annu. Rev. Fluid Mech. 52 (2020) 477–508.

[6] P. Chen, R. Liu, K. Aihara, L. Chen, Autoreservoir computing for multistep ahead prediction based on the spatiotemporal information transformation, Nature Commun. 11 (1) (2020) 1–15.

[7] A. Chattopadhyay, E. Nabizadeh, P. Hassanzadeh, Analog forecasting of extreme-causing weather patterns using deep learning, J. Adv. Modelling Earth Syst. 12 (2) (2020) e2019MS001958.

[8] L.B. Ferreira, F.F. da Cunha, Multi-step ahead forecasting of daily reference evapotranspiration using deep learning, Comput. Electron. Agric. 178 (2020) 105728.

[9] G. Papacharalampous, H. Tyralis, D. Koutsoyiannis, Comparison of stochastic and machine learning methods for multi-step ahead forecasting of hydrological processes, Stoch. Environ. Res. Risk. Assess. 33 (2) (2019) 481–514.

[10] D.J. Gagne, H.M. Christensen, A.C. Subramanian, A.H. Monahan, Machine learning for stochastic parameterization: Generative adversarial networks in the Lorenz'96 model, J. Adv. Modelling Earth Syst. 12 (3) (2020) e2019MS001896.

[11] R.G. da Silva, M.H.D.M. Ribeiro, S.R. Moreno, V.C. Mariani, L. dos Santos Coelho, A novel decomposition-ensemble learning framework for multi-step ahead wind energy forecasting, Energy 216 (2021) 119174.

[12] K. Zhang, J. Thé, G. Xie, H. Yu, Multi-step ahead forecasting of regional air quality using spatial-temporal deep neural networks: a case study of Huaihai Economic Zone, J. Clean. Prod. 277 (2020) 123231.

[13] P.-Y. Kow, Y.-S. Wang, Y. Zhou, I.-F. Kao, M. Issermann, L.-C. Chang, F.J. Chang, Seamless integration of convolutional and back-propagation neural networks for regional multi-step-ahead PM2. 5 forecasting, J. Clean. Prod. 261 (2020) 121285.

[14] Y. Zhou, F.-J. Chang, L.-C. Chang, I.-F. Kao, Y.-S. Wang, Explore a deep learning multi-output neural network for regional multi-step-ahead air quality forecasts, J. Clean. Prod. 209 (2019) 134–145.

[15] L. Xu, N. Chen, Z. Chen, C. Zhang, H. Yu, Spatiotemporal forecasting in earth system science: Methods, uncertainties, predictability and future directions, Earth-Sci. Rev. 222 (2021) 103828.

[16] M. Cheng, F. Fang, T. Kinouchi, I.M. Navon, C. Pain, Long lead-time daily and monthly streamflow forecasting using machine learning methods, J. Hydrol. 590 (2020) 125376.

[17] F. Fang, C. Pain, I.M. Navon, D. Xiao, An efficient goal-based reduced order model approach for targeted adaptive observations, Internat. J. Numer. Methods Fluids 83 (3) (2017) 263–275.

[18] C. Buizza, C.Q. Casas, P. Nadler, J. Mack, S. Marrone, Z. Titus, et al., Data learning: Integrating data assimilation and machine learning, J. Comput. Sci. (2021) 101525.

[19] J. Eyre, Observation bias correction schemes in data assimilation systems: A theoretical study of some of their properties, Q. J. R. Meteorol. Soc. 142 (699) (2016) 2284–2291.

[20] F. Fang, M. Piggott, C. Pain, G. Gorman, A. Goddard, An adaptive mesh adjoint data assimilation method, Ocean Model. 15 (1–2) (2006) 39–55.

[21] I.M. Navon, X. Zou, J. Derber, J. Sela, Variational data assimilation with an adiabatic version of the NMC spectral model, Mon. Weather Rev. 120 (7) (1992) 1433–1446.

[22] Q. Tang, O.S. Schilling, W. Kurtz, P. Brunner, H. Vereecken, H.-J. Hendricks Franssen, Simulating flood-induced riverbed transience using unmanned aerial vehicles, physically based hydrological modeling, and the ensemble Kalman filter, Water Resour. Res. 54 (11) (2018) 9342–9363.

[23] R.E. Kalman, Kalman. a new approach to linear filtering and prediction problems, J. Basic Eng. 82 (1) (1960) 35–45.

[24] G. Evensen, Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, J. Geophys. Res.: Oceans 99 (C5) (1994) 10143–10162.

[25] J. Keller, H.-J.H. Franssen, W. Nowak, Investigating the pilot point ensemble Kalman filter for geostatistical inversion and data assimilation, Adv. Water Resour. 155 (2021) 104010.

[26] M.P. Clark, D.E. Rupp, R.A. Woods, X. Zheng, R.P. Ibbitt, A.G. Slater, J. Schmidt, M.J. Uddstrom, Hydrological data assimilation with the ensemble Kalman filter: Use of streamflow observations to update states in a distributed hydrological model, Adv. Water Resour. 31 (10) (2008) 1309–1324.

[27] J. Bao, L. Li, F. Redoloza, Coupling ensemble smoother and deep learning with generative adversarial networks to deal with non-Gaussianity in flow and transport data assimilation, J. Hydrol. 590 (2020) 125443.

[28] Y. Yang, E.M. Dunham, G. Barnier, M. Almquist, Tsunami wavefield reconstruction and forecasting using the ensemble Kalman filter, Geophys. Res. Lett. 46 (2) (2019) 853–860.

[29] T. Fujita, D.J. Stensrud, D.C. Dowell, Surface data assimilation using an ensemble Kalman filter approach with initial condition and model physics uncertainties, Mon. Weather Rev. 135 (5) (2007) 1846–1868.

[30] C. Sampson, A. Carrassi, A. Aydoğdu, C.K. Jones, Ensemble Kalman filter for nonconservative moving mesh solvers with a joint physics and mesh location update, Q. J. R. Meteorol. Soc. 147 (736) (2021) 1539–1561.

[31] S. Pawar, S.E. Ahmed, O. San, A. Rasheed, I.M. Navon, Long short-term memory embedded nudging schemes for nonlinear data assimilation of geophysical flows, Phys. Fluids 32 (7) (2020) 076606.

[32] S. Pawar, O. San, Data assimilation empowered neural network parametrizations for subgrid processes in geophysical flows, Phys. Rev. Fluids 6 (5) (2021) 050501.

[33] T.M. Hamill, C. Snyder, A hybrid ensemble Kalman filter–3D variational analysis scheme, Mon. Weather Rev. 128 (8) (2000) 2905–2919.

[34] P. Wu, X. Chang, W. Yuan, J. Sun, W. Zhang, R. Arcucci, Y. Guo, Fast data assimilation (FDA): Data assimilation by machine learning for faster optimize model state, J. Comput. Sci. 51 (2021) 101323.

[35] A. Farchi, M. Bocquet, P. Laloyaux, M. Bonavita, Q. Malartic, A comparison of combined data assimilation and machine learning methods for offline and online model error correction, J. Comput. Sci. 55 (2021) 101468.

[36] J. Brajard, A. Carrassi, M. Bocquet, L. Bertino, Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: A case study with the Lorenz 96 model, J. Comput. Sci. 44 (2020) 101171.

[37] A. Wikner, J. Pathak, B.R. Hunt, I. Szunyogh, M. Girvan, E. Ott, Using data assimilation to train a hybrid forecast system that combines machine-learning and knowledge-based components, Chaos 31 (5) (2021) 053114.

[38] J. Brajard, A. Carrassi, M. Bocquet, L. Bertino, Combining data assimilation and machine learning to infer unresolved scale parametrization, Phil. Trans. R. Soc. A 379 (2194) (2021) 20200086.

[39] S. Legler, T. Janjić, Combining data assimilation and machine learning to estimate parameters of a convective-scale model, Q. J. R. Meteorol. Soc. (2021).

[40] M. Peyron, A. Fillion, S. Gürol, V. Marchais, S. Gratton, P. Boudier, G. Goret, Latent space data assimilation by using deep learning, Q. J. R. Meteorol. Soc. 147 (740) (2021) 3759–3777.

[41] X. Kang, A. Kokkinaki, C. Power, P.K. Kitanidis, X. Shi, L. Duan, T. Liu, J. Wu, Integrating deep learning-based data assimilation and hydrogeophysical data for improved monitoring of DNAPL source zones during remediation, J. Hydrol. 601 (2021) 126655.

[42] G.A. Gottwald, S. Reich, Supervised learning from noisy observations: Combining machine-learning techniques with data assimilation, Physica D 423 (2021) 132911.

[43] M.-A. Boucher, J. Quilty, J. Adamowski, Data assimilation for streamflow forecasting using extreme learning machines and multilayer perceptrons, Water Resour. Res. 56 (6) (2020) e2019WR026226.

[44] 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, 2015, pp. 802–810.

[45] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Adv. Neural Inf. Process. Syst. 27 (2014).

[46] G. Burgers, P. Jan van Leeuwen, G. Evensen, Analysis scheme in the ensemble Kalman filter, Mon. Weather Rev. 126 (6) (1998) 1719–1724.

[47] J. Blum, F.-X. Le Dimet, I.M. Navon, Data assimilation for geophysical fluids, in: Handbook of Numerical Analysis, Vol. 14, Elsevier, 2009, pp. 385–441.

[48] S. Falchetti, A. Alvarez, The impact of covariance localization on the performance of an ocean EnKF system assimilating glider data in the Ligurian Sea, J. Mar. Syst. 180 (2018) 76–89.

[49] S.J. Greybush, E. Kalnay, T. Miyoshi, K. Ide, B.R. Hunt, Balance and ensemble Kalman filter localization techniques, Mon. Weather Rev. 139 (2) (2011) 511–522.

[50] R. Petrie, Localization in the ensemble Kalman filter, (MSc Atmosphere), Ocean and Climate University of Reading, 2008.

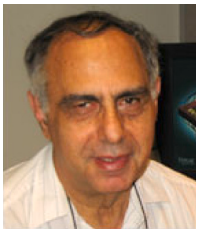
[51] S.B. Taieb, G. Bontempi, A.F. Atiya, A. Sorjamaa, A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition, Expert Syst. Appl. 39 (8) (2012) 7067–7083.

[52] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[53] V. Rao, A. Sandu, A time-parallel approach to strong-constraint four-dimensional variational data assimilation, J. Comput. Phys. 313 (2016) 583–593.

Dr. Meiling Cheng completed her Ph.D. at the department of Earth Science and Engineering at Imperial College London. Using her background in machine learning and fluid dynamics she is working to apply different methods from machine learning to data assimilation techniques for nonlinear fluid flow, atmosphere, flooding predictions and long lead-time forecasting.

Dr. Fangxin Fang is a Senior Research Fellow at the Department of Earth Science and Engineering at Imperial College London. She has conducted and leaded research on the development of cutting-edge machine learning, data assimilation and reduced modeling techniques for improving the predictability of Earth system models (atmosphere, air pollution, radiation modeling, ocean, multiphase flows and environmental issues) and assessment of natural disasters (e.g. flooding) and pollution in coastal/urban environments.



Prof. Ionel M. Navon is a Professor at the Department of Scientific Computing in Florida State University. His research interest includes developing a theory for efficient sensitivity analysis, nonlinear parameter estimation and 4-D variational data assimilation for NASA, NMC/NOAA and FSU-GSM operational 3-D models,working on impact of efficient large-scale minimization algorithms (L-BFGS, Hessian free methods and hybrid methods applied to multidisciplinary problems), as well as developing ensemble aspects related to forecast error covariance, balance constraints, nonlinearity, model errors, computational efficiency, and verification.

Prof. Christopher Paina Professor in the department of Earth Science and Engineering at Imperial College London (ICL), UK. He is also head of the Applied Computation and Modeling Group (AMCG), which is the largest department research group at ICL and comprises of about 70 research active scientists. AMCG specializes in the development and application of innovative and world leading modeling techniques for earth, engineering and biomedical sciences. The group has core research interests in numerical methods for ocean, atmosphere and climate systems, engineering fluids including multiphase flows, neutral particle radiation transport, coupled fluids–solids modeling with discrete element methods, turbulence modeling, inversion methods, imaging, and impact cratering. He was honored by Imperial College London by winning its Research Excellence Award in 2010, in recognition his world-leading research.