The predictive skill of neural network models for the large-scale dynamics of the multi-level Lorenz '96 systems

Seoleun Shin

[1]Department of Oceanography, Chonnam National University, Gwangju, Korea

Key Points:

- The predictive skill of a neural network (NN) model for the prediction of Lorenz '96 dynamics is examined.
- NN models are competent in representing highly nonlinear dynamics even without resolving smaller-scale processes.
- The performance can be enhanced by some approaches in an attempt to increase the diversity of a whole ensemble.

Corresponding author: Seoleun Shin, seoleuns@gmail.com

Abstract

The predictive skill of a neural network model for the prediction of the highly nonlinear Lorenz '96 dynamics is examined and a way to improve the skill is investigated. We train neural networks with pairs of a large-scale variable and its tendency generated by numerical integrations of full-level Lorenz '96 equations. The Neural Network (NN) models are then used to estimate the tendency given state of the variable which is then updated without resolving or parameterizing smaller-scale processes. We also apply ensemble data assimilation to the predicted background states and examine to which degree NN models capture the dynamics in a long-term prediction-analysis cycle. It has been found that NN models are skillful in estimating the tendencies for dynamics with quasi-periodic characteristics. Moreover, they have strong potentials in predicting even more chaotic waves when an external forcing has been increased. We have examined if the performance of NN models can be enhanced by using ensemble frameworks in the context of machine learning or training an NN with a dataset generated by ensemble simulations of full-level Lorenz '96 equations. In these approaches, prediction-analysis cycles run stably for long periods and NN models are skillful in representing the large-scale dynamics. However, NN models can face difficulties in capturing extreme events occurring rarely and whose predictability is very low. An interesting aspect of the results is that efforts need to be focused in finding an effective way to increase the diversity of a whole ensemble and improve the skill in such situations.

Plain Language Summary

We have examined the performance of a neural network model for the prediction of a conceptual dynamic system of complex weather and climate called "Lorenz '96 model". It has been found that neural network models are competent in representing slowly varying dynamics which has quasi-periodic characteristics. This can be done without solving the full-level equation set but by training the neural network with information about the states of the first-level variables and their tendencies. However, it is especially challenging for neural network models to predict rarely occurring extreme waves which are highly unpredictable. Their predictive skill can be improved by using some approaches tested in an attempt to use a combination of neural network models or an effective training strategy. Upon results from this study, we suggest that efforts need to be focused in finding methods to increase the diversity of a whole combination of neural network models.

1 Introduction

Machine learning approaches have drawn intensive attentions from diverse research communities including Numerical Weather Prediction (NWP). For example, the early applications of Neural Network (hereafter, NN) model to prediction and data analysis in the research area of meteorology and oceanography are reviewed and discussed about some of associated difficulties in Hsieh and B. (1998). They also proposed a link between the NN model to variational data assimilation and outlooked a hybrid of NN and dynamical models. For aerodynamic problems, White et al. (2019) introduced a new neural network architecture called "a cluster network" model that performs better than previous model-free methods, in attempts to save time for the prediction of computation intensive problems while maintaining sufficient accuracy. Recently, Scher (2018) measured the predictive skill of a Convolutional Neural Network (CNN) model trained on a simple general circulation model and showed that it could predict model states several time steps ahead. They further suggested that the NN model could produce similar climate statistics to that generated by the global circulation model. Weyn et al. (2019) developed weather prediction models using deep CNN trained on past weather data and used them to predict 500-hPa geopotential height. Their best performing CNN has captured the climatology, annual variability of 500-hPa heights, and predicted realistic atmospheric states at lead times of 14 days although this model did not outperform an operational weather model. Meanwhile, Brajard et al. (2020) built a data-driven surrogate CNN model to emulate the 40-variable Lorenz model (Lorenz & Emanuel, 1998) from partial and noisy observations. Their method is based on an iterative algorithm in such way that at each iteration, an Ensemble Kalman Filter (EnKF) step alternates with a machine learning step that learns the underlying dynamics of the analysis from the data assimilation. In their sensitivity experiments, forecast skills decrease smoothly with increased observational noise as long as the fraction of observations exceeds a half of the model domain. They argued that the success of this newly suggested approach might at least partially rely on the autonomous characteristics of the Lorenz '96 system but expect that this method can be extended to a system of slow variations.

Recently, Scher and Messori (2021) have tested some ensemble generation methods in the context of machine learning such as a "network retraining ensemble" which is generated by starting with different initial seeds for the random number generators in the initialization of the network weights. An ensemble framework can be also generated by retraining the network by using different sets of features or input variables (Borovkova & Tsiamas, 2019). Scher and Messori (2021) have found that their ensemble approaches enable the system perform better than an "unperturbed neural network forecasting system". Among the four ensemble approaches used in their study, the retraining ensemble performed best. The machine learning ensemble models have drawn attentions in diverse research areas (e.g., Eslami E. & Lops, 2019, and others) and demand for them would grow. Studies on the use of neural networks further include postprocessing of ensemble weather forecasts (e.g., Rasp & Lerch, 2018, and others), and it is suggested that neural network models can be more efficient and perform better than benchmark postprocessing techniques.

Dueben and Bauer (2018) examined whether models based on machine learning can compete with conventional physical models in the simulation of large-scale flows of the three-level Lorenz '96 system (Thornes et al., 2017) and global weather forecasts. Upon some success with NN models for short-term forecasts, they discussed that NN models would face challenging issues such as complex interactions between coupled components of the earth system. This perspective has drawn our attention to the predictive skill of NN models for a highly-nonlinear dynamic system and finding a way to improve their performance. In connection with the studies by Dueben and Bauer (2018), we investigate closely the accuracy of predicted large scale variables of multi-scale Lorenz '96 systems without parameterizing or resolving smaller scale processes explicitly. Also, ensemble data assimilation is applied for the variable to examine to which extent the NN models can identify the evolution of complex dynamics in a long-term prediction-analysis cycle. In this study, it has been found that the accuracy of NN model predictions is influenced by the dynamical characteristics of the Lorenz '96 system, and it is challenging for the NN model to capture highly unpredictable transitions correctly. For the purpose of finding a way to improve the predictive skill of NN models in such situations, we use data sets generated by ensemble simulations of full equation sets for the training of NN models and additionally test an "ensemble" (Scher & Messori, 2021) frameworks.

A neural network ensemble is generated most commonly following a two-stage design process (Sharkey, 1996): 1) first generating individual networks; 2) and then combining the output of several networks that solve the same task. For an extension of it, Loyola R. (2006) suggest a three-stage design process for neural network ensembles: 1) first generating multiple training data sets different from the original data set; 2) training networks individually using the data sets, one for each network; 3) and then selecting a subset of the more suitable networks for a combination. In our initial pursuit, we adopt simple approaches each from these two types of NN ensembles for testing highly-nonlinear cases. Also, we suggest a way of NN training by providing information about diverse state transitions obtained from ensemble simulations of full-level equation sets. It has been found that NN models have potentials in predicting states of large-scale variables without having to parameterize or resolving smaller-scale processes explicitly so that they can be applicable for an individual multi-scale or a coupled system. However, it has been also observed that NN models have difficulties in estimating correct tendency for dynamical states which undergoes rapid changes.

We provide detailed explanations of the two- and three-level Lorenz '96 systems and parameters chosen for the tests in the following section. Tests of two different external forces in the Lorenz system are introduced and dynamic features are also illustrated. Then, the NN model generated for this study is briefly presented and the information of training/validation data sets are given. Also, we summarize some concise details of data assimilation process based on Ensemble Kalman Filter (EnKF). In section 3, we compare the accuracy of predictions by NN models and numerical solvers for the large-scale flow dynamics, and evaluate the performance of NN models when ensemble data assimilation is applied cyclically. Also, results from the test of machine learning ensemble are given

in details. We discuss the skill of NN models in representing large-scale dynamics for multi-scale systems and summarize the performance of the NN models dependent on the dynamical properties of a target system in the final section.

2 Experimental set-up

2.1 The Lorenz '96 system

We examine the performance of a neural network model for the two-level Lorenz '96 system (Lorenz, 1996) first and then test the three-level Lorenz '96 equation suggested in Thornes et al. (2017). The three-level equation is an extended version of the two-level one by adding one more level to further subdivide multi-scale interactions. Accordingly, the NN model updates only states of large-scale variables without solving one or two smaller-scale processes of each Lorenz '96 system, respectively. We describe these two multi-level systems by adopting the same notations used in Lorenz (1996) and Thornes et al. (2017). The two-level model consists of the two governing equations coupled to represent a larger scale variable "x" and a smaller scale variable "y":

$$\begin{aligned} \frac{dx_k}{dt} &= x_{k-1}(x_{k+1} - x_{k-2}) - x_k + F - \frac{hc}{b}\sum_{j=1}^{J} y_{j,k} \,, \\ \frac{dy_{j,k}}{dt} &= -cb\, y_{j+1,k}\,(y_{j+2,k} - y_{j-1,k}) - c\, y_{j,k} + \frac{hc}{b} x_k \,. \end{aligned} \tag{1}$$

The indices j and k range from 1 up to J and K that represent the dimension of x and y, respectively. Likewise, the three-level model is composed of three governing equations with the variables, a large-scale "x", medium-scale "y" and small-scale "z", and the index i is used to for the variable z and it ranges from 1 up to I:

$$\begin{aligned} \frac{dx_k}{dt} &= x_{k-1}(x_{k+1} - x_{k-2}) - x_k + F - \frac{hc}{b}\sum_{j=1}^{J} y_{j,k} \,, \\ \frac{dy_{j,k}}{dt} &= -cb\, y_{j+1,k}\,(y_{j+2,k} - y_{j-1,k}) - c\, y_{j,k} + \frac{hc}{b} x_k - \frac{he}{d}\sum_{i=1}^{I} z_{i,j,k} \,, \\ \frac{dz_{i,j,k}}{dt} &= -ed\, z_{i-1,j,k}\,(z_{i+1,j,k} - z_{i-2,j,k}) - g_z e\, z_{i,j,k} + \frac{he}{d} y_{j,k} \,. \end{aligned} \tag{2}$$

Table 1 shows the list of the parameters chosen following Thornes et al. (2017) and Dueben and Bauer (2018). Among the parameters in table 1, F is the large-scale forcing term that can modulate the behaviors of the Lorenz system. It is set to be first 8.0 and then raised to 20.0 to examine the predictive skill of NN models for a system having different dynamical characteristics (Lorenz, 1996). For simplicity, the other parameters are left unchanged that are tunable to control the frequency, amplitude of oscillations, and the strength of coupling between variables that represent different scales. The degrees of the freedom given to the two-level model are $I = 10$, $J = 10$, and to the three level are $I = 8$, $J = 8$, $K = 8$ as chosen in Thornes et al. (2017) and Dueben and Bauer (2018).

Table 1: List of parameters for the Lorenz '96 equation set

Notation	Definition	Value
H	Strength of the coupling between spatial scales	1.0
B	Relative magnitudes of the coupled variables x and y	10.0
C	Relative evolution speed of the coupled variable x and y	10.0
F	Large-scale forcing applied to the system	8.0 or 20.0
D	Relative magnitudes of the coupled variables y and z	10.0
E	Relative evolution speed of the coupled variables y and z	10.0
g_z	Damping parameter	1.0

The multi-level Lorenz '96 equations are integrated by the fourth-order Runge-Kutta method (e.g., Ott et al., 2004, and others), and the numerical solutions of x are assumed to represent the true states of the large-scale flow. In this way, the data for training and validation of neural network models are obtained by integrating these discretized multi-level equation sets. The temporal discretization is $\Delta t = 0.005$ Model Time Units (MTUs) for the integration as in Dueben and Bauer (2018). The lag autocorrelation and dynamical characteristics of the simulated system is shown in Fig. 1a for the case $F = 8$. The correlation draws closely to zeros near 0.37 MTU, which corresponds to about 2 days (Lorenz, 1996). Figure 1b shows the phase space of x_1 and x_2 trajectories among the ten x variables, illustrating a quasi-periodic characteristics of the system. Dynamical systems with diurnal or seasonal cycles have a strong quasi-periodicity (Sterk & van Kekem, 2017). Trajectory alternates one or the other periodic courses. Meanwhile, the lag autocorrelation given $F = 20$ is about 0.27 MTU (1.3 days) (Fig. 1c). With the increased forcing, the system shows highly chaotic characteristics as illustrated by the the phase space trajectories of x_1 and x_2. During the time integration of the equation, x_k^n and $\Delta x_k^n = x_k^{n+1}$ - x_k^n are saved at every 1 MTU for the training of NN models so that 2,010,000 sets of this pair are prepared. Namely, the input data for the training are pairs of the largest scale variable and its tendency sampled in such way that temporal correlation of each pair is sufficiently low (Dueben & Bauer, 2018). Eventually, the predictive skill of a numerical solver and NN models are evaluated by measuring the deviation from the true state of x when they solve the evolution of only large-scale variable x. It means that the numerical solver integrates only the first-level equation for x while the y-dependent term in the equation is omitted. Meanwhile, NN models estimate the tendency of the large-scale variable only to update its state at every time step.

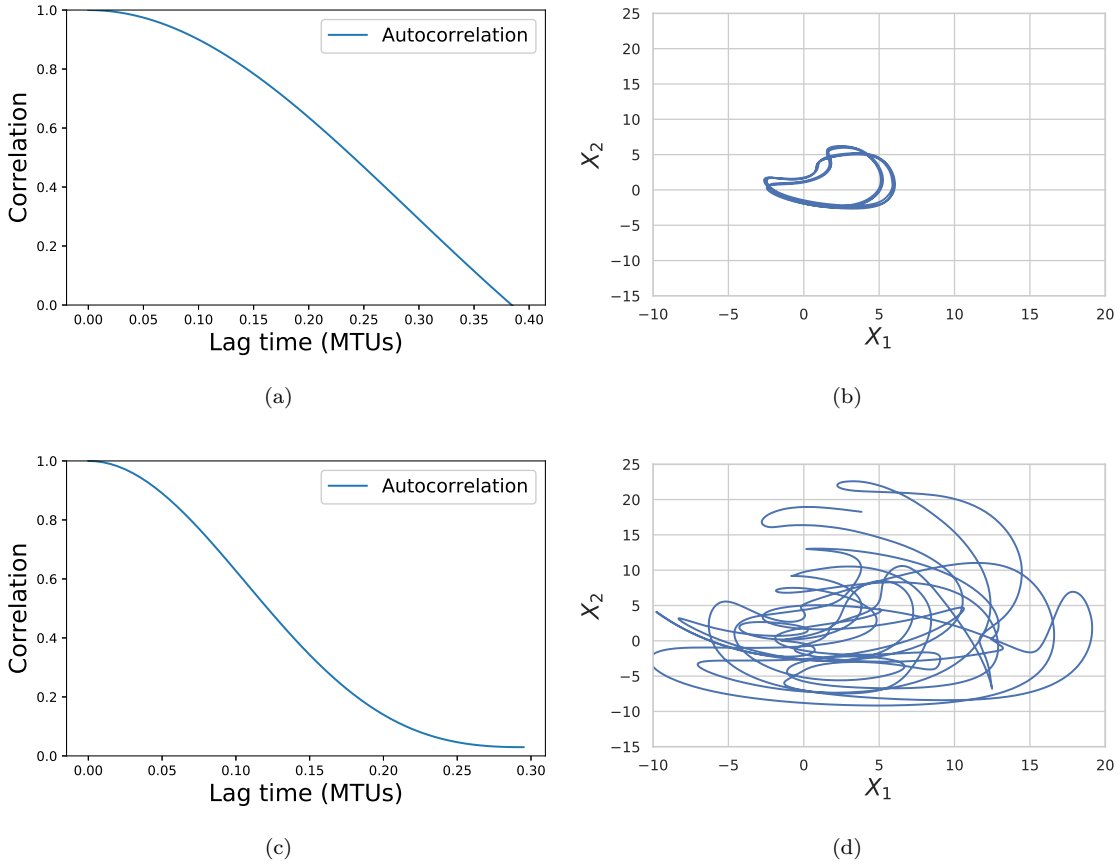


Figure 1: Autocorrelation in time and the phase space of x_1 and x_2 trajectories of the two-level Lorenz' 96 system given $F = 8$; (a) and (b), and $F = 20$; (c) and (d).

2.2 The NN model

"Artificial Neural Networks" consist of many connected neurons that can produce a sequence of activations. Neurons get activated through weighted connections from other neurons activated earlier. In this way, "learning" is about finding the "weights" in order to make NNs eventually exhibit "desired behavior" (Schmidhuber, 2014). We use the Keras Python library Chollet (2015) to construct a sequential neural network which is a linear stack of layers with neurons placed on input, hidden, and output layers. The input layer is the first layer that feeds on information from input data, while results from neural networks are sent off from the output layer. There are hidden layers in between. The information about the input is given to the first layer with specifications of the dimension of the input data. The connections of neurons in the network designate assigned weights. A positive weight reflects an excitatory connection, while negative values inhibitory connections. Then a bias term is added to the total weighted sum of inputs to shift the activation function that controls the output. We use the same design options for the generation of NN models as done in Dueben and Bauer (2018). For the activation function, we use the hyperbolic tangents while a stochastic gradient descent is used for optimization processes. During the training phase

of the neural network, the weights and biases within the network are optimized by reducing a given loss function. We use the mean absolute error as for the loss function. The optimization algorithm repeats propagations and weight updates. When an input vector is fed on the network, it is propagated forward through the network, layer by layer, until it reaches the output layer. The output of the network is then compared to the desired output and the loss function is computed. The resulting error value is calculated for each of the neurons in the output layer. The error values are then propagated from the output back through the network in such a way to minimize the loss function. The software package developed by Hatfield (2019) has provided a basis for the computational tool for the investigation of the multi-scale Lorenz 96 systems in this study.

Figure 2 illustrates the processes of feeding data into the neural network and following sequences. Each neuron (node) in a layer is sequentially connected to a neuron in other layer forming a multilayer neural network. As briefly stated in the previous section, 2,010,000 sets of x_k^n and Δx_k^n = x_k^{n+1} - x_k^n are prepared for the training of NN models. Then, the constructed NN model is used to make predictions of the tendency of the large-scale variables x to update the state of x of the next time step. In this study, we try only the so called "global" approach suggested in Dueben and Bauer (2018), in which all x_k variables (k =1, $\cdots$, K) are used to predict a tendency of a single variable x_k, instead of using variables surrounding that grid point. Dueben and Bauer (2018) used the third-order Adams–Bashforth explicit time-stepping scheme by iteratively using NN model tendency outputs to reduce the error associated with temporal discretization. However, we directly use the tendency (Δx_k^m) output from NN models to update the states of x_k at the next time step ($x_k^{m+1} = x_k^m + \Delta x_k^m$) to uniformly use the same order of time stepping as in the tests for which data assimilation is applied. Some more details of the data assimilation procedure are given in the next subsection. It might need to be reminded again that the other smaller scale variables (y or z) of the Lorenz system are neither used for training nor resolved in the NN.

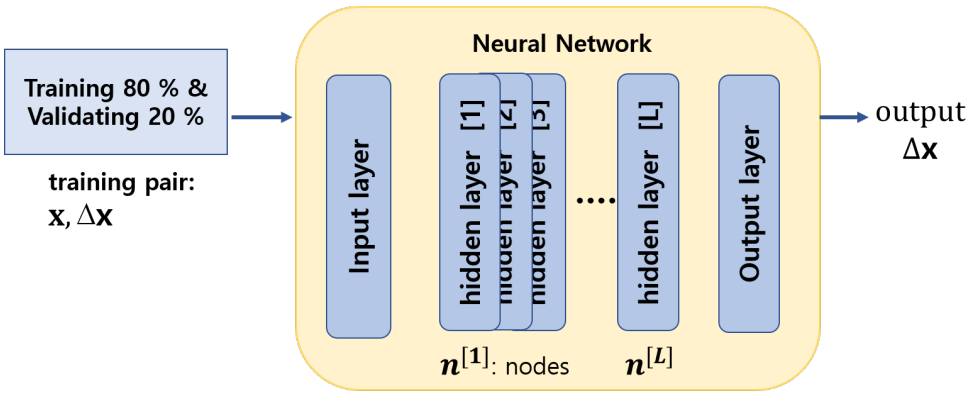


Figure 2: Schematic diagram for the training of neural network models

Since the solution of the Lorenz 96 equation has relatively short-term autocorrelation given the parameters in this study (Fig. 1a and 1c) and the input data are 1 MTUs apart, we did not select a

random splitting of training and verification data sets (Scher & Messori, 2019). As frequently done in machine learning practices, 80 % of the numerically generated data is used for training and 20 % for verification (Fig. 2). All input states of the large-scale variable x_k have been normalized in such way that the value ranges between -1.0 and 1.0. The updated value using the output tendencies is then inverted again to produce new physical state x_k. We have examined the performance sensitivity of NN models to the number of epochs, nodes, hidden layers, by simply changing their numbers. It is observed that the performance is little affected or even gets slightly worse if the number of hidden layers, nodes, and epochs exceeds 32, 100, 100, respectively for the dynamical system given parameters. In this study, we present results from tests all with the NN model combination of 32, 100, 100 except one test with the combination 4, 100, 100 for the experiment with $F = 8.0$ and $F = 20$ for the two-level Lorenz '96 equation.

2.3 Data assimilation

We are especially interested in the use of ensemble predictions from a NN prediction system as background states for data assimilation. The term 'background' means that the predicted states of each model prior to a data assimilation process. To examine this question, we use the Ensemble Kalman Filter (EnKF) method with perturbed observations with a simplified setting. The observation is generated by adding perturbations to true states with the standard deviation of 1.0 which is often used in data assimilation studies (e.g., Ott et al., 2004, and others). At first, EnKF is applied at every time-step for the variable x_k at every grid point. This is intended to examine the ability of models in updating right tendencies for the dynamic system when the state is corrected by EnKF every time step. After this test, data assimilation is applied for every 10 time step, which is approximately corresponding to 6 hour interval. A multiplicative inflation with a constant factor 1.1 is multiplied to the background ensemble states. The size of ensemble is chosen to be 10, the same number as given degrees of freedom (the number of grid is 10) for the two-level system. We use the simply same number of ensemble member for the three-level tests as well, where the number of grid for the large-scale variable is 8. The initial ensemble states are sampled from the outputs from a long-term spin-up simulation of multi-level equations so that each member is the state at a randomly chosen time step. The accuracy of analysis is evaluated by comparing with the assumed true states which are obtained by numerically integrating the full-level equations for 2000 time steps starting from the final state of the spin-up simulation mentioned above. We use the Root Mean Square Difference (RMSD) from the true state to measure the accuracy of ensemble mean of model predictions or analysis.

3 Results and Discussions

3.1 The performance of NN models

At first, we compare the results from the tests using the numerical solver and the NN models with 4 hidden layers having 100 neurons on each for the two-level Lorenz '96 system with the

forcing $F = 8$. Data assimilation is processed at every time step, which means that corrections on the forecasts are made at every time step prior to the production of a new prediction. We expect that accuracy of analysis becomes equivalent to each other eventually in both numerical and NN model tests. Figure 3 shows the time series of ensemble spread and RMSD for all x_k. Also, RMSD and ensemble spread averaged over the time period between 2 and 10 MTUs is denoted by 'RMSD' and 'SPRD' respectively on the upper left part of each panel. We have excluded the values before 2 MTU as a spin-up period of data assimilation. The RMSD and SPRD in the test with the NN model with 4 hidden layers (Fig. 3b) are equivalent to those values in the test with the numerical solver (Fig. 3a). This means that NN model with this configuration is as skillful as the numerical model for the dynamical system showing a quasi-periodic behaviors.

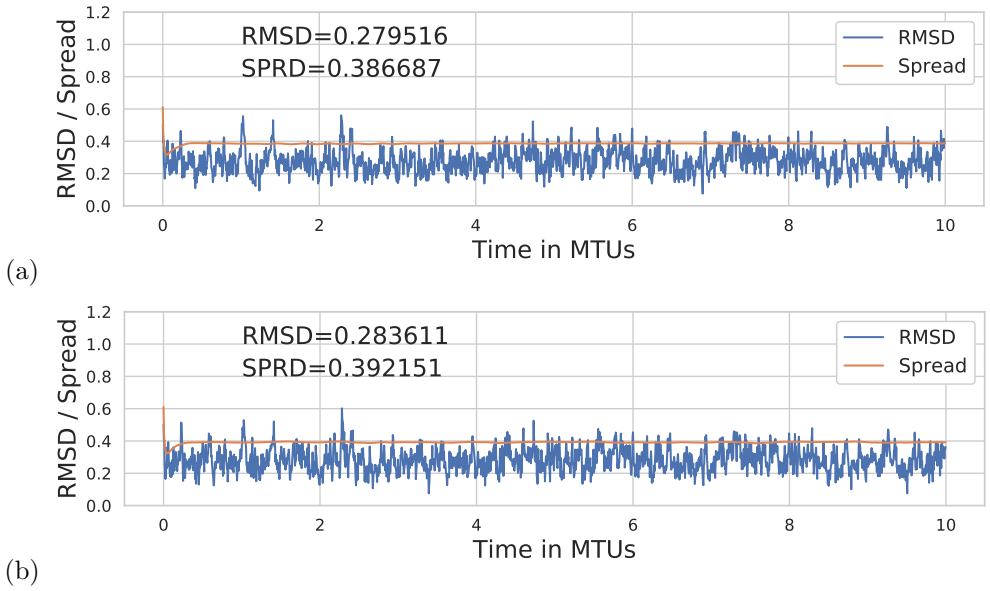


Figure 3: Time series of RMSD and ensemble spread in tests with (a) a numerical model and (b) an NN model with 4 hidden layers, for the two-level equation system with F=8.

Interestingly, the situation changes significantly when EnKF is applied to at every 10th step, i.e., every about 6 hour as often done in numerical weather prediction systems. Since observation is available at every grid point, the data assimilation works no better than a simple insertion of observation if a time-averaged RMSD is larger than the observation error. Figure 4a shows that the time-series of RMSD is below observation error 1.0 but much higher than the ensemble spread in the test with the numerical model. Meanwhile, the analysis RMSD is smaller than the value in the test with the numerical solver as well as the given observation error in the test using the NN model with 4 hidden layers (Fig. 4b). This result indicates that the the NN model is competent in predicting tendencies of large-scale flow even without resolving smaller scale evolutions explicitly. It may suggest that the NN model has potential to be an alternative tool in the prediction of large-scale geophysical flows with quasi-periodic behaviors with a lower computational complexity in that sub-grid processes are not required to be treated in predictions.

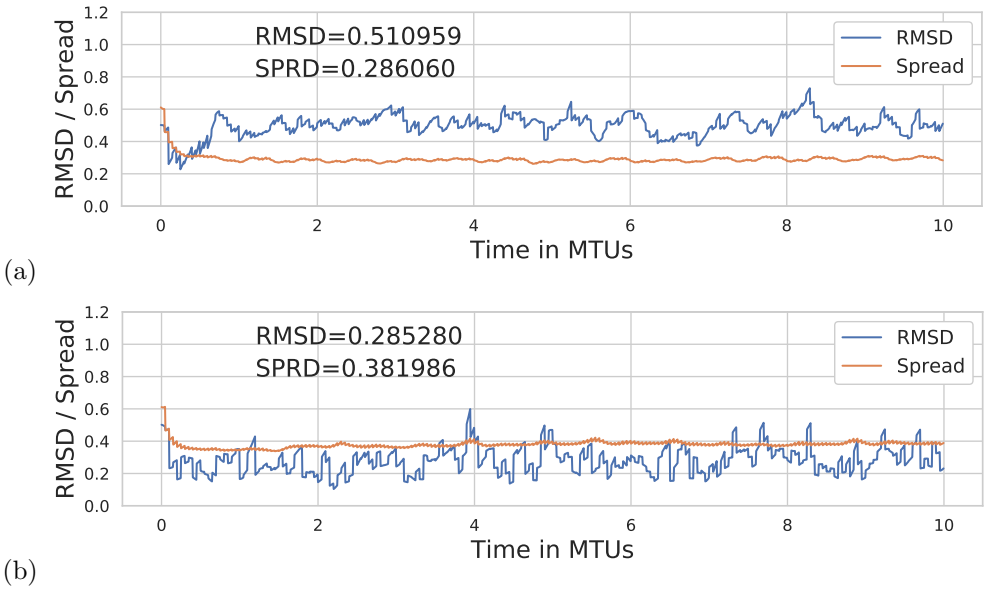


Figure 4: The same as figure 3, but EnKF is applied at every 10th time step.

Now, we compare the results from the tests for the system with the forcing $F = 20$. Differently from the situation in $F = 8$, the performance of the NN model with 4 hidden layers is slightly worse than that of the numerical counterpart according to the overall value of RMSD (Fig. 5a and 5b). In an attempt to improve the skill of the NN model, we first increase the number of hidden layers. Here, we present the result from the test with the NN model having 32 hidden layers. The increase of layers has led to an improvement in the performance and the peaks of large errors have subsided (Fig. 5c). Further increases in the number of layers or neurons (beyond 32 and 100, respectively and in combination) are little effective in a significant decrease of errors any more. From a number of tests with different combinations, we have learned that the improvement through such a simple change in the neural network might be limited and may not bring out fundamental changes in the predictive skill of NN models, at least for highly nonlinear problems. Therefore, we rather pay more attention to an alternative training strategy and network ensemble combinations than tuning of optimal NN configurations.

First, we have examined the increase of training data for NN model. One way was simply increase the integration time for true states, and we found that there is no significant gain in the skill of NN models. Next, we have prepared a training data set including outputs from ensemble integrations starting from perturbed initial states so that the size of training data became the number of ensemble member multiplied by the 2,010,000 pairs of x_k^n and Δx_k^n. It is found that the larger the ensemble size, the better the performance of NN models is. We present results from the test with the NN model trained by a data set including 30 members of ensemble integrations. Besides this approach, we have tested two additional ensemble approaches in the machine learning context. One is the training of neural networks with a single data set, but by starting different

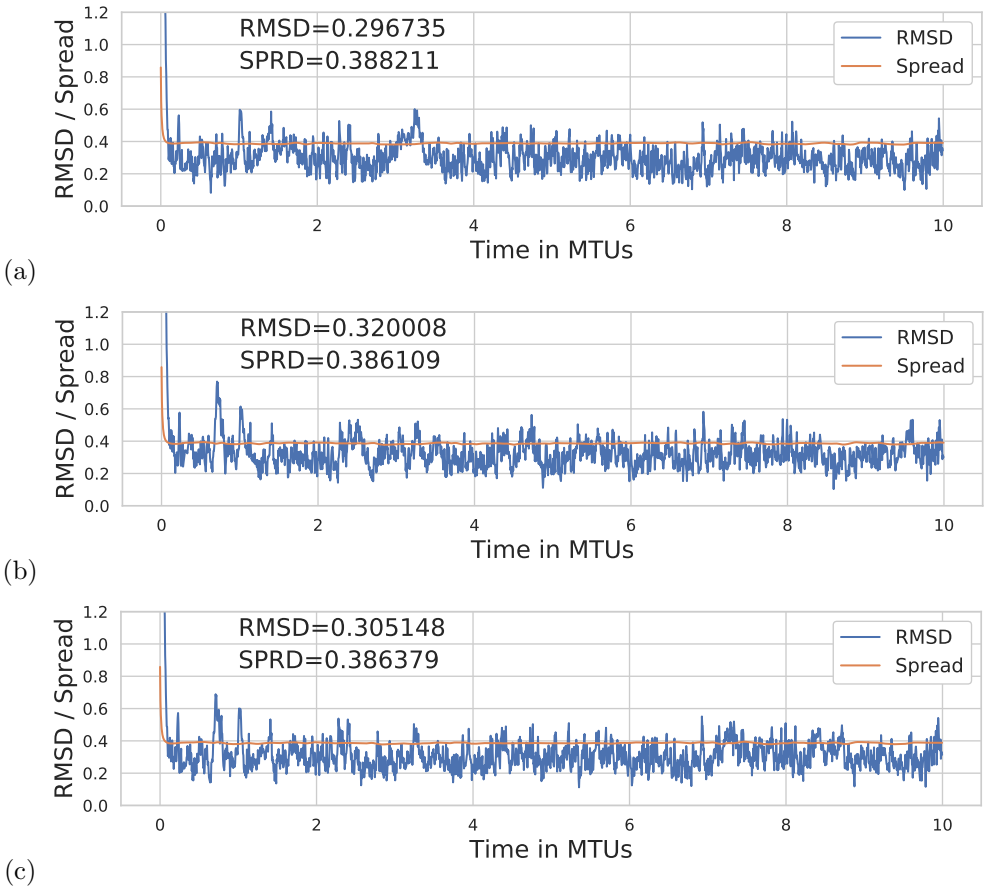


Figure 5: Time series of RMSD and SPRD in tests with (a) a numerical model, (b) an NN model with 4 hidden layers, (c) and with 32 hidden layers for the two-level system with F=20.

initial seeds for the random number generators to produce multiple NN models. The "Glorot normal" initialization procedure contained in the Keras package is used for the weight initialization in this study. This ensemble method is also tested for weather forecasts and showed good predictive skill in (e.g., Scher & Messori, 2021). As introduced in section 1, this is a kind of two-stage design process (Sharkey, 1996). Another ensemble approach examined here is the use of multiple training data sets to generate multiple NN models. We have prepared individual 10 training data sets from the ensemble integrations starting from perturbed initial conditions. This can be a kind of three-stage design process suggested by Loyola R. (2006) but here we do not go on selecting a subset of the generated networks for a combination. Thus, the number of NN models ready for use is 10, which form another multi model ensemble. To identify the characteristics of the error growths in the ensemble predictions, we did not apply data assimilation at first. A brief description and the list of tests are summarized in Table 2. For comparisons, we have examined L_1 norm errors and Continuous Ranked Probability Score (CRPS) of ensemble predictions in each test described in Table 2. The size of ensemble is set to be 10, as in the previous data assimilation test. Here, we show results only up to 1.0 MTU corresponding to about 5 days, since major differences in the predictive skills between models can be indiscernible later (Fig. 6). According to the L_1 norm

Table 2: List of tests and brief descriptions

Test	Model Description
Numeric	Numerical solver for the equation of large-scale variable x
Single	A single neural network with the configuration of 32 hidden layers
Ens.Train	A single network trained with a data set made of ensemble simulations
Multiple NN-I	Networks trained individually by using different data sets from each other
Multiple NN-II	Networks generated individually by varying the initialization of weights

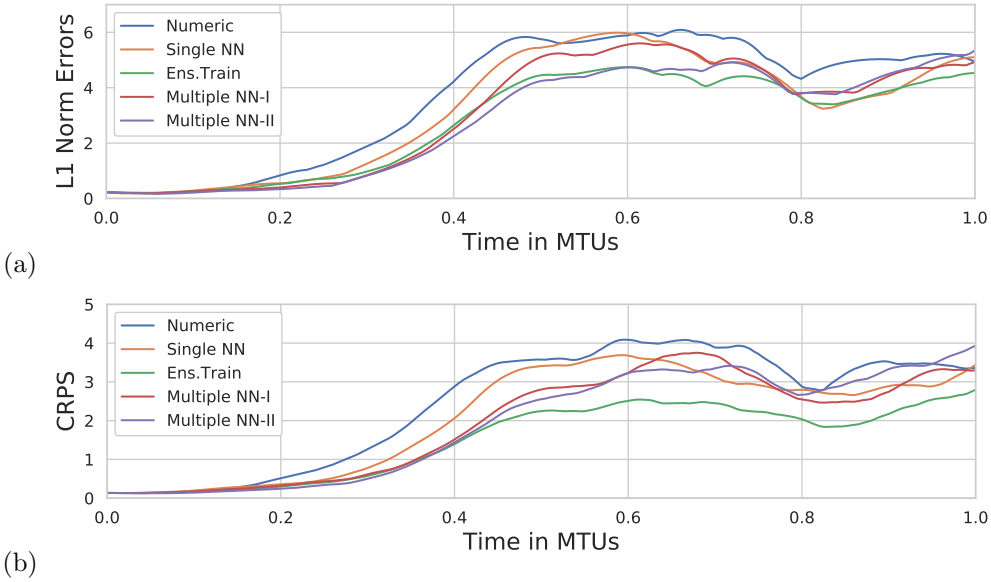


Figure 6: Time series of (a) the L_1 norm of ensemble predictions in tests 'Numeric', 'Single', 'Ens.Train', 'Multiple NN-I', and 'Multiple NN-II' and (b) CRPS in each test.

errors, predictions in Multiple NN-I and Multiple NN-II are better than others during the early stage less than 0.4 MTU (Fig. 6a). Around that time, the prediction error keeps on increasing rapidly in Multiple NN-I while the error remain lowest among predictions until 0.6 MTU. From then on the performances of the two models become similar to each other again. Meanwhile, the prediction error in Ens.Train is slightly larger than those in Multiple NN-I and Multiple NN-II in the early stage, but Ens.Train outperforms after 0.6 MTU and the errors remain smallest most of time. In contrast to this, prediction error in the test Numeric grows fastest but the growth rate diminishes in time so that the level of errors approaches to that of the errors in other tests after about 0.9 MTU. Likewise, the error growth in the test Single shows a similar pattern to this. In a summary, the early growth rates of errors in tests Numeric and Single are distinguishably larger than in other tests but performances of all tests alternate for lead in the later stage after about 0.9 MTU.

The predictive skills of models measured by CRPS (Fig. 6b) are consistent with the accuracy of prediction measured by L_1 error norms in Fig. 6a. A difference is that CRPS is best in the test Ens.Train soon after 0.4 MTUs, while the L_1 error norm is lowest after 0.6 MTUs among the tests. Overall, it is apparent that those multi NN model ensembles (Multiple NN-I and Multiple NN-II here) response less sensitive to initial perturbations than other NN models as well as numerical models in the early stage. Therefore, the early growth of errors are slower than in other tests until the ensemble mean deviate eventually from the truth rapidly afterwards. The predictive skill seems to be best when the training approach in Ens.Train is used during the period presented here. In consideration of these results, we examine how this property affects the accuracy of analysis when data assimilation is applied to the predictions.

Figure 7 shows the time series of analysis errors from the tests Ens.Train, Multiple NN-I, and Multiple NN-II. The overall accuracy of the analysis from the test Ens.Train is better than that in the test Numeric (for a comparison, see Fig. 5a). The other two NN approaches performs equivalently to the result in Numeric (Fig 7b and c). The time-averaged RMSD values in these three tests are also better than the 'Single' NN model test (Fig. 5c).

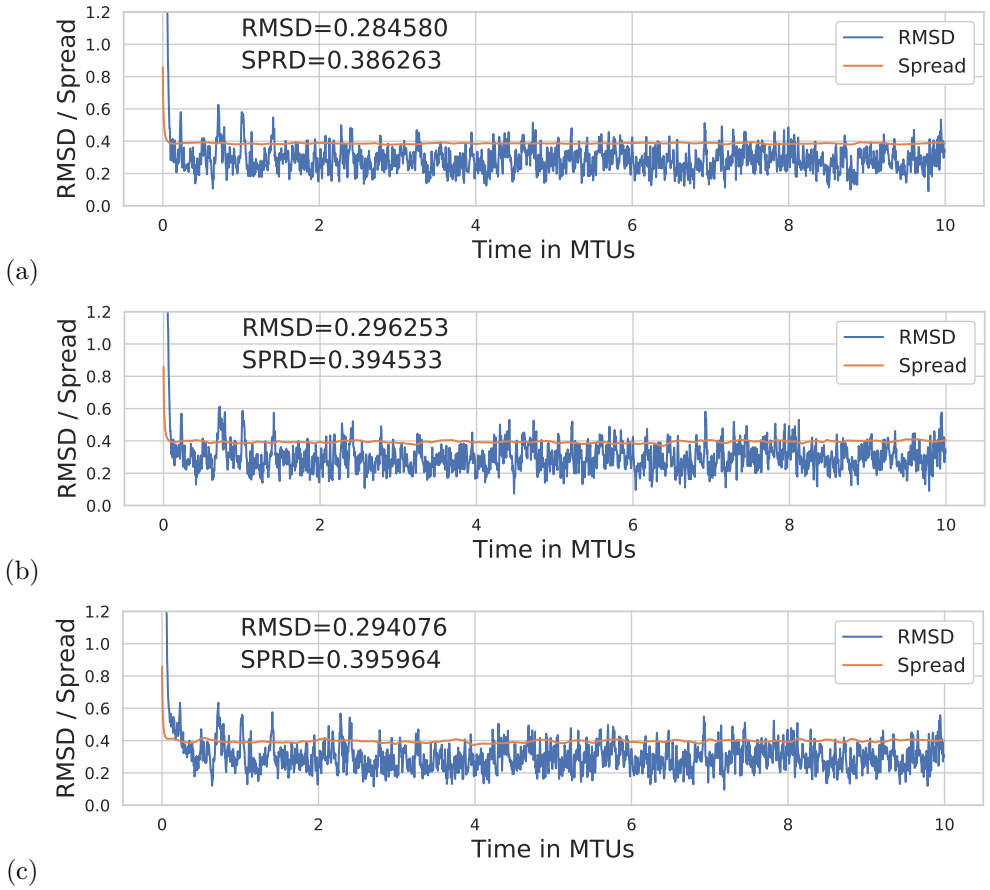


Figure 7: Time series of RMSD and ensemble spread in tests (a) Ens.Train, (b) Multiple NN-I, and (c) Multiple NN-II for the two-level system with F=20.

As done in the test given $F = 8$, we examine the performance of models in a more realistic prediction-analysis cycle, i.e. when EnKF is applied to at every 10th step. Figure 8 shows the time series of RMSD and spread as well as their time-averaged values. In the test with the numerical model, data assimilation plays a role no more than just insertion of observation in the test Numeric (Fig. 8a). The RMSD value is much larger than the given observation error 1.0 most of the time period in this test. It means that the numerical model resolving only the large-scale flow of a multi-scale system cannot capture the evolution of the dynamics any more when the state is corrected through data assimilation only once in 10 time steps. On the other hand, the RMSD value is below 1.0 in a majority of the time period and the analysis errors do not keep growing in the test Single, which shows that the NN model is able to represent the dynamics to some degrees (Fig. 8b). Moreover, errors in the test Ens.Train, Multiple NN-I, and Multiple NN-II are better controlled than in the test Single (Fig. 8c-e). Interestingly, the capability of these NN models for identifying the dynamics is manifested strikingly when EnKF is applied once in 10 time steps. Especially, the accuracy of analysis is best in the test Ens.Train and the error is well controlled below 1.0 for the whole period. The gap between RMSD and SPRD is small compared to the tests Numeric and Single. In the next place, we examine further if the potential of NN models in representing the large-scale dynamics without resolving smaller scale processes also for the three-level system.

3.2 Three-level equations

NN models are trained by data sets including pairs of x and Δx from the numerical simulation of the three-level equation set as described in section 2.2. The predictive skill of NN models is then evaluated for representing large-scale variable x without resolving y and z in the three-level equation set (Eq. 2) when EnKF is applied at every time step first. We have experimented the same set of tests listed in Table 2 as for the two-level system. Figure 9 shows that the time-averaged RMSD value is smallest in the test Numeric (Fig. 9a), while there are peaks of large errors outstanding between 5 and 8 MTUs in all the tests with NN models when EnKF is applied at every time step (Fig. 9b-e). During those time periods of large errors, sudden drops of predictive skill of NN models seem to occur and the uncontrolled errors accumulates. These errors might degenerate the overall performance by NN models and lead to a slightly larger time-averaged RMSD than in Numeric. However, errors do not keep on growing in time and disappear rather quickly so that models seem to find right tendencies again.

Now, we examine the performance of NN models when EnKF is applied at every 10th step, i.e. about 6 hour as done in the two-level tests. Figure 10a shows that the accuracy of analysis is marginally better than a direct insertion of observations since the time-averaged RMSD is only slightly smaller than observation error 1.0. In comparison with this, the growths of errors are better controlled in all the tests with NN models, but several enlarged peaks of large RMSD close to 3.0 are still outstanding (Fig. 10b-e). The accuracy of analysis is significantly better in the test Ens.Train, Multiple NN-I, and Multiple NN-II (Fig. 10c-e) than in the test Single (Fig. 10b).

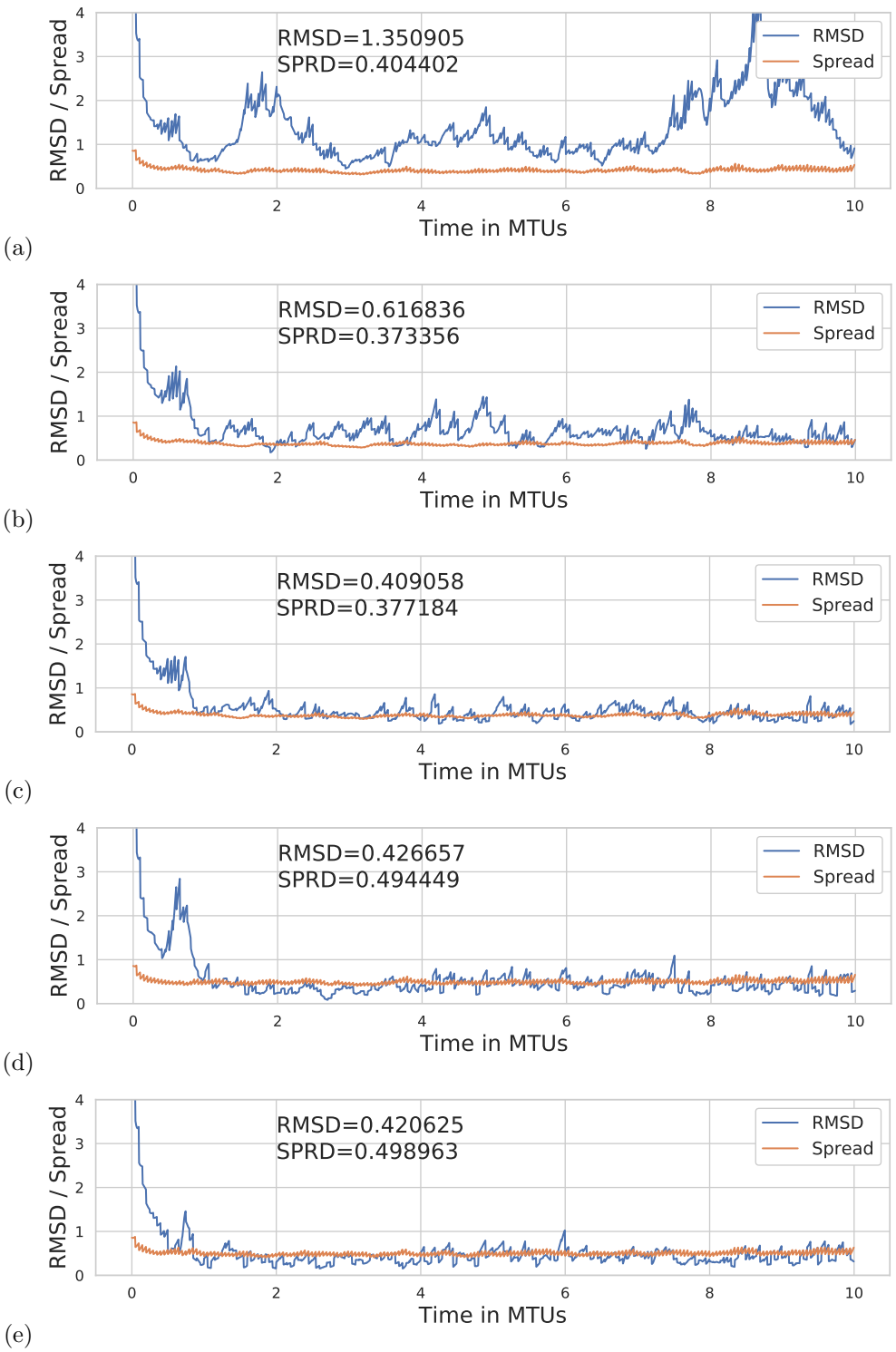


Figure 8: Time series of RMSD and SPRD in tests with (a) Numeric, (b) Single, (c) Ens.Train, (d) Multiple NN-I, and (e) Multiple NN-II when EnKF is applied every 10th time step for the two-level system with F=20.

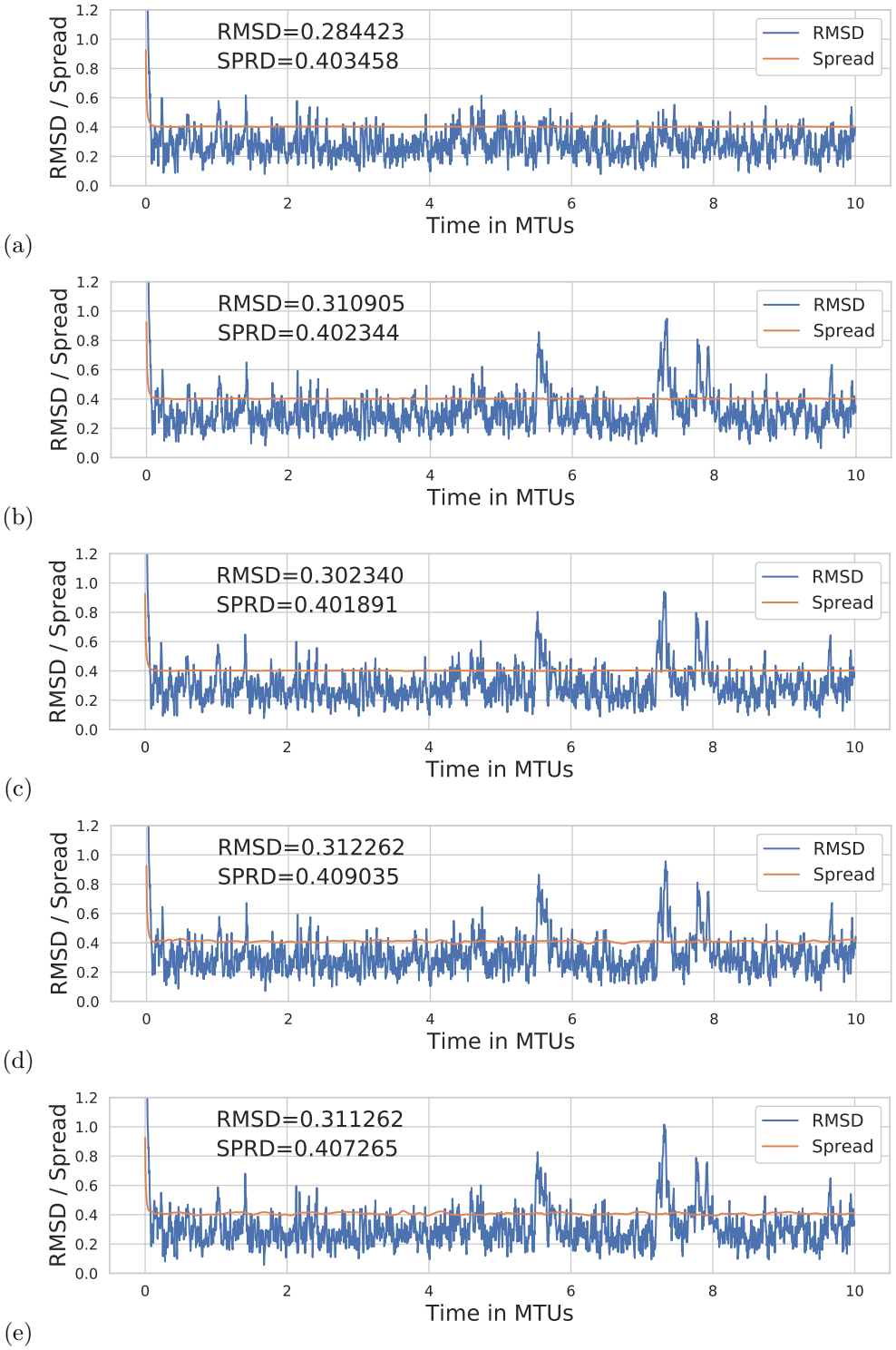
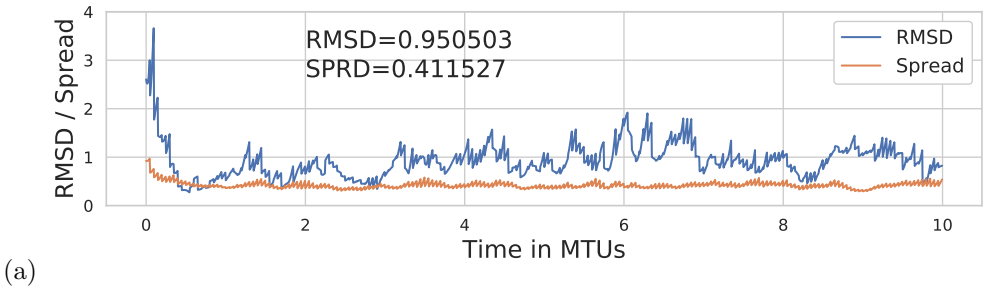


Figure 9: Time series of RMSD and SPRD in tests (a) Numeric, (b) Single, (c) Ens.Train, (d) Multiple NN-I, and (e) Multiple NN-II when EnKF is applied at every time step for the three-level system with F=20.

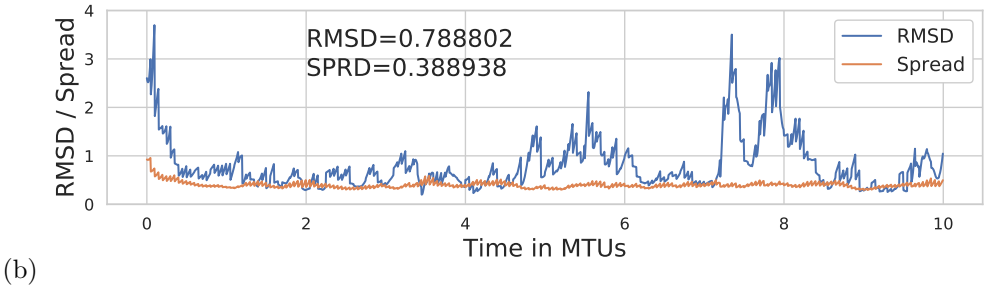
The RMSD values are controlled around the value 0.5 mostly except the period between 5 and 6 MTUs, 7 and 8 MTUs, and close to 10 MTUs in the test Multiple NN-I, Multiple NN-II and Ens.Train. These results may suggest that NN models have the potential in reproducing large-scale flows without resolving smaller-scale processes as well as interactions with them. However, a notable point is that NN models deviate significantly when the multi-level dynamic system possibly undergoes highly-nonlinear transitions that they are probably rarely trained for. To investigate this point, we examine background error covariances and the evolution of the large-scale variables at specific positions for those time periods when NN models suffer from finding right tendencies. All of the following investigations are about the results from the test when EnKF is applied once in 10 time steps.

Figure 11 shows background error covariances computed over time and space in the tests, Ens.Train, Multiple NN-I, and Multiple NN-II. The error covariances distinguish the test Multiple NN-II (Fig. 11c) from the other two tests Ens.Train and Multiple NN-I. Ensemble members show stronger tendency to evolve in the same direction as each other in the test Multiple NN-II than Ens.Train and Multiple NN-I. This coincides with that analysis in the test Multiple NN-II are least accurate among the three tests (see Fig. 10e). Accordingly, we presume that the background error covariances indicate to which degree errors of ensemble members covariate and this can be related with the accuracy of analysis (Fig. 10). The collective behavior of ensemble members shows the lack of diversity so that it increases chance for the ensemble mean to drift away from the true state. To validate this reasoning, we have examined individual variables and present here the evolution of x_8 for which NN models produce significantly inaccurate predictions for the time period 7 and 8 MTUs. Also, we show the temporal changes of another variable x_5 for a comparison as NN models suffer less severely from incorrect estimation of tendencies than x_8.

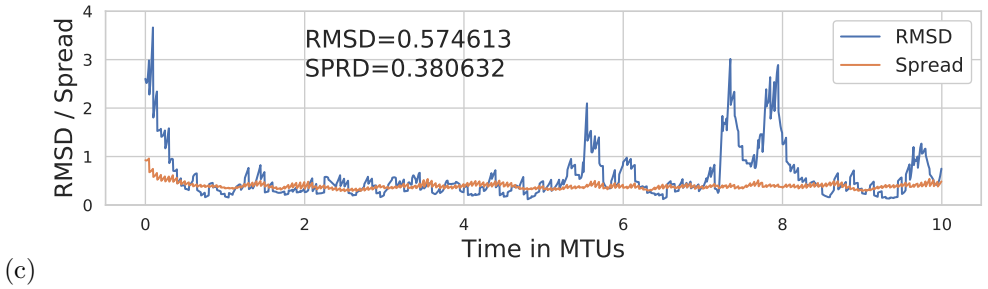
Figure 12 shows the time series of background states of those variables in Numeric and Ens.Train, and they are compared to their corresponding true states between 7 and 8 MTUs. The background states of x_5 in the two tests approximates the true states during the most of the time period. Only after 7.8 MTU, they deviate notably from the true state. Meanwhile, the temporal variations of the true x_8 are much more rapid and larger in magnitude than those of x_5. Prior to 7.2 MTUs the background states in Numeric deviate from the true state more than those in Ens.Train. However, between 7.2 and 7.5 MTUs the background states of x_8 in Ens.Train change in considerably retard of the true states so that errors become larger than in Numeric. Thereafter, there is a short period of sharp increase in the magnitude of the true x_8 after 7.7 MTU and then the true state drops rapidly again, while the changes of background states delay again. In a summary, the NN model in Ens.Train has potentials for identifying right direction of state changes mostly but lack skills in producing tendencies strong enough to represent rapid transitions of nonlinear dynamics closely. These two points lead us to examine if the NN models tend to estimate tendencies depending on the probability of the occurrence of states that they have learned about. This may



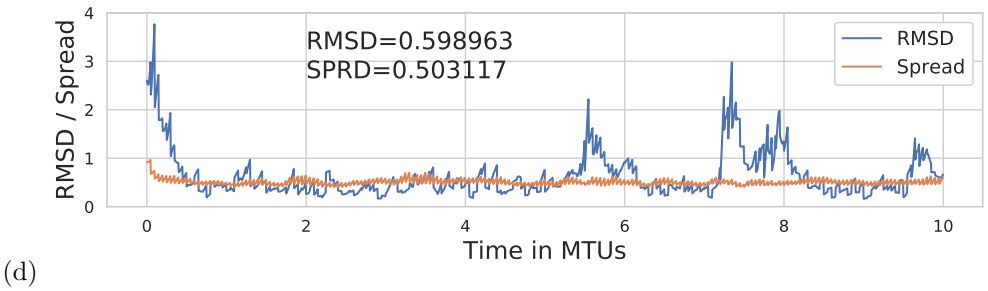
(a)



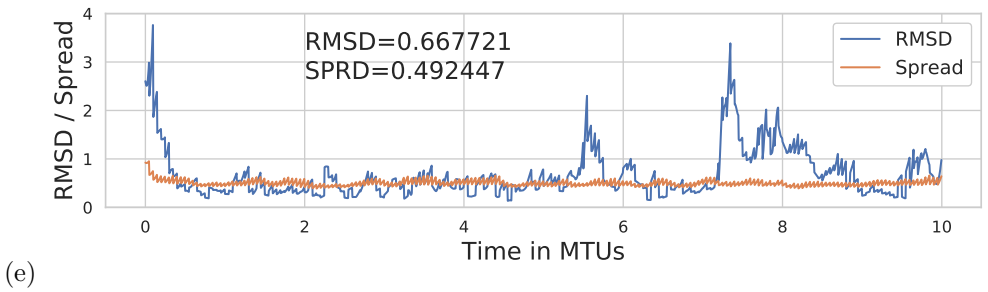
(b)



(c)

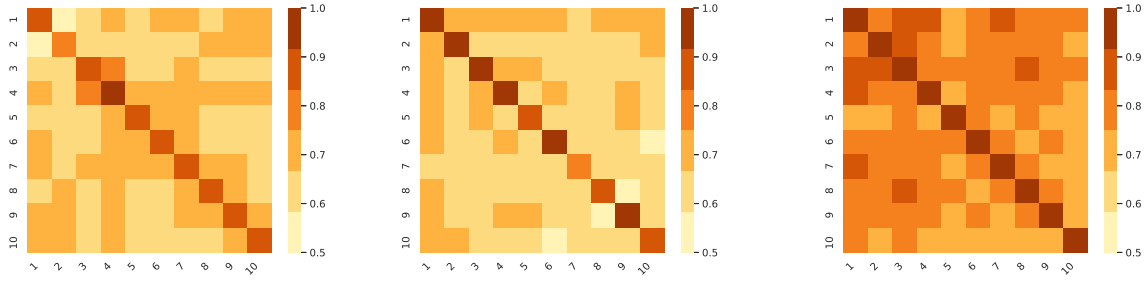


(d)



(e)

Figure 10: The same is as in fig. 9, except that EnKF is applied at every 10th time step.



(a) (b) (c)

Figure 11: Background error covariances in tests (a) Ens.Train, (b) Multiple NN-I, and (c) Multiple NN-II for the three-level system with F=20.

not be surprising but the clarifications may provide some insights for the design of a more skillful NN model.

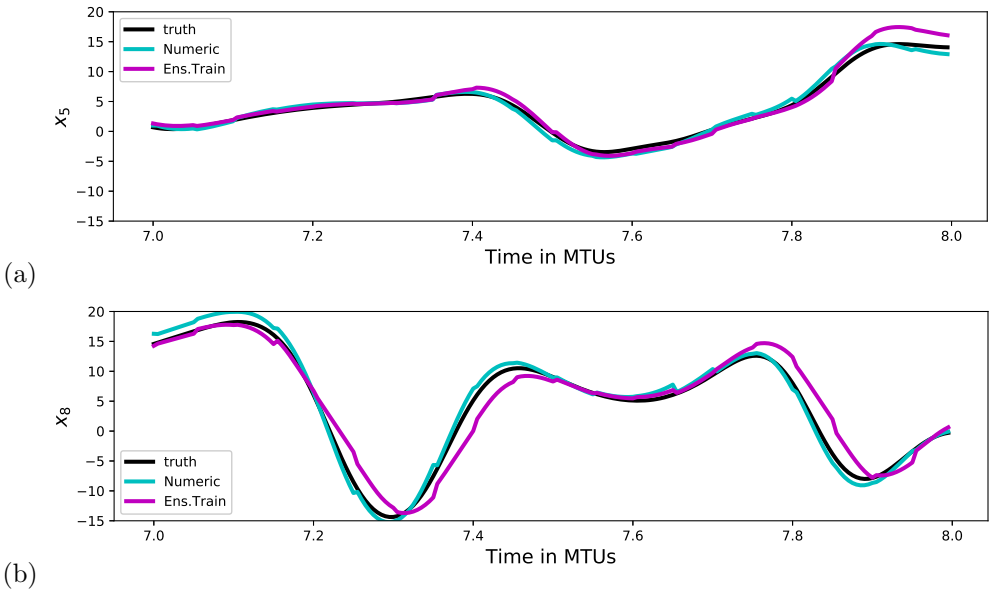


Figure 12: Time series of the the true state, background states in Numeric and Ens.Train for (a) x_5 and (b) x_8.

We have examined the Probability Density Function (PDF) of the normalized background states of x_k in the test Ens.Train in comparison with the PDFs of the true states during the test period between 2 and 10 MTUs. The number of samples for x_k each from the true and predicted states in the test Ens.Train is 12,800 and they are distributed approximately to a Gaussian function (Fig. 13a). Also, the PDF of all x_k states in the training data is shown together, which is close to a Gaussian distribution curve (Fig. 13a). The PDF of the background states in Ens. Train resembles that of true states, which may indicate that NN models are skillful overall in representing

the true dynamics. Now, we have examined the PDF of x_8 specifically as done for the time series of the predictions (Fig. 12b). Figure 13b shows the PDF of the normalized x_8 from the training data set, true trajectories, and background states in Ens.Train for the time period between 2 and 10 MTUs. The PDF of the true states is distributed well within the cap of the PDF of the training data mostly, but peaks sharply in the values between 0.2 and 0.3 of the normalized x_8 (Fig. 13b). Although the PDF of the background states approximate that of the true state in general, some discrepancy can be unavoidable in that range of the x_8 value as the peak extrudes far beyond the value of the Gaussian PDF of the training data. This may imply that the NN models presumably have more difficulties in identifying the right tendencies for such state changes than those in the other ranges of the value.

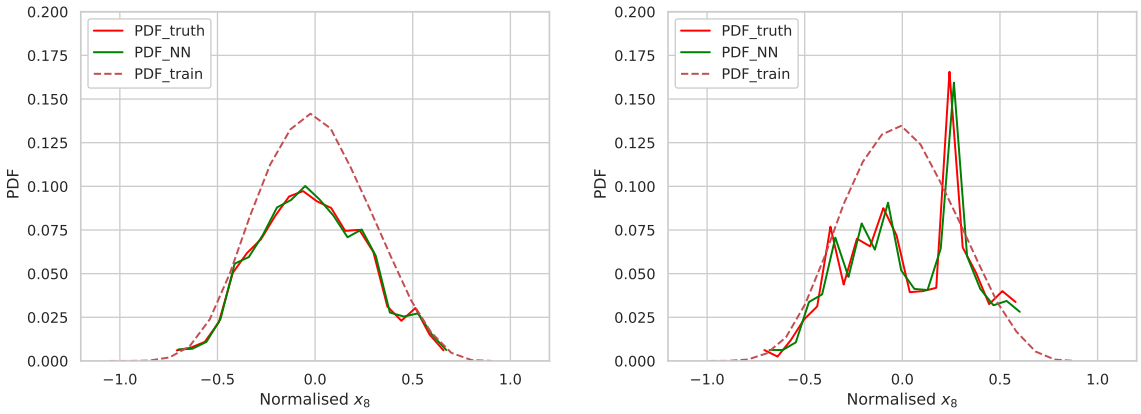


Figure 13: Probability Density Function (PDF) for the normalized values of (a) the all x_k and (b) x_8 between 2 and 10 MTUs in the test Ens.Train ($\mathrm{PDF_{NN}}$) in comparison with those in the training data set ($\mathrm{PDF_{train}}$) and the true states ($\mathrm{PDF_{truth}}$).

4 Conclusions

In this study, we examine the predictive skill of NN models for the multi-scale Lorenz '96 systems and suggest their potentials in capturing large-scale atmospheric dynamics without resolving smaller-scale processes explicitly. The predictions by NN models are more accurate than those by a numerical solver for the single-level Lorenz '96 equation until the saturation of prediction errors approaches. As long as the data assimilation is applied at every time step, the predicted large-scale states approximate true states well for a long time period of prediction-analysis cycle. Even when ensemble data assimilation is applied only once in 10 time steps ($\sim$ 6 hour) NN models are skillful in representing the evolution of the dynamics, while the accuracy of analysis is nearly no better than a direct insertion of observations when the numerical solver is used. Especially, the NN model trained by a data set containing dozens of long-term ensemble simulations of the Lorenz '96 system outperforms other NN model ensemble approaches. However, it has been also shown that

the performance of NN models can drop suddenly in an extreme situation when a fully turbulent system evolves rapidly. The system show rapid transitions of states from a maximum and to a minimum value of the large-scale variable at a certain spatial positions within a time period of about 1 MTUs ($\sim$ 5 days). The cyclic application of data assimilation makes effective adjustment to the predictions by NN models, but the correction may not be enough to prevent the model from estimating inaccurate tendencies in such circumstances. It is probably due to that NN models have less chances to learn the extreme events and may underestimate their occurrences. Besides, we have shown that the predictability of the true system can drop sharply when the dynamical system undergo such transitions, and this coincides with the outstanding peak of large errors in the NN model tests. The NN models seem to response sensitively to an event of low predictability and predictions drift away from the true states until the model resume estimating a relevant tendency again. Consequently, our interests have been drawn to questions how we can deal with the limitations of NN models in representing highly unpredictable evolution of an atmospheric dynamical system. In this study, we have focused on figuring out where the weakness of the NN model originates and proposing what we can do to alleviate the difficulties. One of feasible approach can be the construction of training datasets describing features that a dynamical system can have as fully as possible to provide NN models with comprehensive information about the system. Simultaneously, we also acknowledge that it is a challenging task to sample a number of datasets which can satisfy physical constraints and at the same time represent diverse states observed in nature (Paul & Aires, 2015). Also, the computational costs and requiring storage would increase rapidly with the increase of degrees of freedom of targeting problems. Nevertheless, it would be critical to carefully produce datasets such as "reanalysis ensemble (Carton et al., 2019)" that can be used for diverse purposes, probably also for the training of NN models. Another notable point is that individual networks are trained independent of each other and this may result in ensemble members eventually similar to each other so that they don't contribute much for the diversity of a whole ensemble. Loyola R. (2006) have suggested training the ensemble members by using such as different initial conditions, training patterns, topology of the neural network, and training algorithms to ensure that all the members not make the same errors. Among those suggestions, we plan to examine training algorithms to improve the diversity of a whole ensemble such as a negative correlation learning (Liu & Yao, 1999) and search for ways to minimize the collective behaviors of NN model ensembles.

Acknowledgments

The author thanks Samuel Hatfield for the source codes that are used for the experiments discussed in this study. This work was funded by the Korea Meteorological Administration Research and Development Program under Grant KMI2018-07010 and KMI2020-01115 . A software package for this research is available in this in text data citation reference: `10.5281/zenodo.5528329`.

References

Borovkova, S., & Tsiamas, I. (2019). An ensemble of lstm neural networks for high-frequency stock market classification. *Journal of Forecasting*, *38*, 600-619.

Brajard, J., Carrassi, A., Bocquet, M., & Bertino, L. (2020). Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: a case study with the lorenz 96 model. *Geoscientific Model Development Discussions*, *44*, 1–21.

Carton, J. A., Penny, S. G., & Kalnay, E. (2019). Temperature and salinity variability in the soda3, ecco4r3, and oras5 ocean reanalyses, 1993–2015. *J. Clim.*, *32*, 2277-2293.

Chollet, F. e. a. (2015). *Keras.* Retrieved from `https://github.com/fchollet/keras`

Dueben, P. D., & Bauer, P. (2018). Challenges and design choices for global weather and climate models based on machine learning. *Geoscientific Model Development Discussions*, *11*, 3999-4009.

Eslami E., Y. C. A. S., A. K. Slman, & Lops, Y. (2019). A data ensemble approach for real-time air quality forecasting using extremely randomized trees and deep neural networks. *Neural Computing and Applications*, *32*, 7563–7579.

Hatfield, S. (2019). *Evaluating neural network-based surrogates for the forecast model in an ensemble kalman filter.* Retrieved from `https://github.com/samhatfield/ml_da`

Hsieh, W. W., & B., T. (1998). Applying neural network models to prediction and data analysis in meteorology and oceanography. *Bull. Amer. Metor. Soc*, *79(9)*, 1855-1870.

Liu, Y., & Yao, X. (1999). Ensemble learning via negative correlation. *Neural Networks*, *12*, 1399–1404.

Lorenz, E. N. (1996). Predictability: A problem partly solved. In *Proc. ecmwf seminar on predictability* (Vol. I, p. 1-18).

Lorenz, E. N., & Emanuel, K. A. (1998). Optimal sites for supplementary weather observations: Simulation with a small model. *J. Atmos. Sci.*, *55*, 399-414.

Loyola R., D. G. (2006). Applications of neural network methods to the processing of earth observation satellite data. *Neural Networks*, *19*, 168-177.

Ott, E., Hunt, B. R., Szunyogh, I., Zimin, A. V., Kostelich, E. J., Corazza, M., ... A., Y. J. (2004). A local ensemble kalman filter for atmospheric data. *A local ensemble Kalman filter for atmospheric data assimilation*, *56A*, 415–428.

Paul, M., & Aires, P. (2015). Using shannon's entropy to sample heterogeneous and high-dimensional atmospheric datasets. *Quart. Journal of Roy. Metor. Soc.*, *141*, 469-476.

Rasp, S., & Lerch, S. (2018). Neural networks for postprocessing ensemble weather forecasts. *Mon. Wea. Rev.*, *146*, 3885–3900.

Scher, S. (2018). Toward data-driven weather and climate forecasting: Approximating a simple general circulation model with deep learning. *Geophysical Research Letters*, *45*, 12616–12622.

Scher, S., & Messori, G. (2019). Generalization properties of feed-forward neural networks trained on lorenz systems. *Nonlin. Processes Geophys.*, *26*, 381-399.

Scher, S., & Messori, G. (2021). Ensemble methods for neural network-based weather forecasts. *Journal of Advances in Modeling Earth Systems*, *13*, e2020MS002331.

Schmidhuber, J. (2014). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85-117.

Sharkey, A. J. C. (1996). On combining artificial neural nets. *Connection Science*, *8:3-4*, 299-314.

Sterk, A. E., & van Kekem, D. L. (2017). Predictability of extreme waves in the lorenz-96 model near intermittency and quasi-periodicity. *Complexity*, *2017*(Article ID 9419024), 14.

Thornes, J. A., Düben, P. D., & Palmer, T. (2017). On the use of scale-dependent precision in earth system modelling. *Quart. Journal of Roy. Metor. Soc.*, *143*, 897-908.

Weyn, J. A., Durran, D. R., & Caruana, R. (2019). Can machines learn to predict weather? using deep learning to predict gridded 500-hpa geopotential height from historical weather data. *Journal of Advances in modeling earth systems*, *11*, https://doi.org/10.1029/2019MS001705.

White, C., Ushizima, D., & Farhat, C. (2019). eural networks predict fluid dynamicssolutions from tiny datasets. *arXiv preprint*, arXiv:1902.00091.