Received: 5 September 2018 | Revised: 25 November 2018 | Accepted: 28 November 2018

DOI: 10.1002/env.2553

SPECIAL INVITED ARTICLE

WILEY

Deep echo state networks with uncertainty quantification for spatio-temporal forecasting

Patrick L. McDermott | Christopher K. Wikle

Department of Statistics, University of Missouri, Columbia, Missouri

Correspondence
Patrick L. McDermott, Jupiter Intelligence, Boulder, CO 80302.
Email: plmyt7@gmail.com

Funding information
U.S. National Science Foundation; U.S. Census Bureau, Grant/Award Number: SES-1132031; NSF-Census Research Network (NCRN) program, Grant/Award Number: DMS-1811745

Abstract

Long-lead forecasting for spatio-temporal systems can entail complex nonlinear dynamics that are difficult to specify a priori. Current statistical methodologies for modeling these processes are often highly parameterized and, thus, challenging to implement from a computational perspective. One potential parsimonious solution to this problem is a method from the dynamical systems and engineering literature referred to as an echo state network (ESN). ESN models use *reservoir computing* to efficiently compute recurrent neural network forecasts. Moreover, multilevel (deep) hierarchical models have recently been shown to be successful at predicting high-dimensional complex nonlinear processes, particularly those with multiple spatial and temporal scales of variability (such as those we often find in spatio-temporal environmental data). Here, we introduce a deep ensemble ESN (D-EESN) model. Despite the incorporation of a deep structure, the presented model is computationally efficient. We present two versions of this model for spatio-temporal processes that produce forecasts and associated measures of uncertainty. The first approach utilizes a bootstrap ensemble framework, and the second is developed within a hierarchical Bayesian framework (BD-EESN). This more general hierarchical Bayesian framework naturally accommodates non-Gaussian data types and multiple levels of uncertainties. The methodology is first applied to a data set simulated from a novel non-Gaussian multiscale Lorenz-96 dynamical system simulation model and, then, to a long-lead United States (U.S.) soil moisture forecasting application. Across both applications, the proposed methodology improves upon existing methods in terms of both forecast accuracy and quantifying uncertainty.

KEYWORDS

deep modeling, echo state networks, hierarchical Bayesian, long-lead forecasting, spatio-temporal

1 | INTRODUCTION

Spatio-temporal data are ubiquitous in engineering and the sciences, and their study is important for understanding and predicting a wide variety of processes. One of the chief difficulties in modeling spatial processes that change with time is the complexity of the dependence structures that must describe how such a process varies, and the presence of high-dimensional complex data sets and large prediction domains. It is particularly challenging to specify

This material encompasses the J. Stuart Hunter Lecture from the 28th Annual Conference of the International Environmetrics Society (TIES), presented 18 July 2018.

parameterizations for nonlinear dynamical spatio-temporal models (DSTMs) that are simultaneously useful scientifically (e.g., long-lead forecasting as discussed below) and efficient computationally. Statisticians have developed some multilevel (deep) mechanistically motivated models that can accommodate process complexity and the uncertainties in predictions and inference, typically within a hierarchical Bayesian paradigm (see the overview in Cressie & Wikle, 2011). However, these models can be computationally expensive, require prior information and/or a significant amount of data to fit, and are typically application specific. On the other hand, the science, engineering, and machine learning communities have developed alternative approaches for nonlinear spatio-temporal modeling, particularly in the neural network context (e.g., recurrent neural networks [RNNs]). These approaches can be very flexible but, again, are computationally expensive, require large amounts of training data, and/or require "pretraining." In addition, these approaches often do not provide formal measures of uncertainty quantification. There are, however, parsimonious approaches to RNNs in the engineering literature, such as the echo state network (ESN), although the standard implementation of this approach does not include spatio-temporal dependencies, deep learning, or formal uncertainty quantification. Here, we present a hierarchical deep statistical implementation of an ESN model for spatio-temporal processes with the goal of long-lead forecasting with uncertainty quantification.

The methodology presented here is motivated by the problem of long-lead forecasting of environmental processes. The atmosphere is a complex dynamical system and deterministic models of the system show extreme sensitivity to initial conditions. Because of that, skillful weather forecasts are only possible out to about 10–14 days (e.g., Stern & Davidson, 2015). However, dynamical processes in the ocean operate on much longer time scales, and many atmospheric processes depend crucially on the ocean as a forcing mechanism. This coupling between the slowly varying ocean and the faster varying atmosphere (and associated processes) allows for the skillful prediction of some general properties of the atmospheric state many months to over a year in advance (i.e., long-lead forecasting). Although statistical models are not as skillful as deterministic numerical weather prediction models for short-to-medium weather forecasting, they have consistently performed as well or better than deterministic models for long-lead forecasting (e.g., Barnston, Glantz, & He, 1999; Jan van Oldenborgh, Balmaseda, Ferranti, Stockdale, & Anderson, 2005). In some cases, fairly standard linear regression or multivariate canonical correlation analysis methods can be used to generate effective long-lead forecasts (e.g., Knaff & Landsea, 1997; Penland & Magorian, 1993). However, given the inherent nonlinearity of these systems, it has consistently been shown that well crafted nonlinear statistical methods often perform better than linear methods, at least for some spatial regions and time spans (e.g., Berliner, Wikle, & Cressie, 2000; Drosdowsky, 1994; Gladish & Wikle, 2014; Kondrashov, Kravtsov, Robertson, & Ghil, 2005; McDermott & Wikle, 2016; Tang, Hsieh, Monahan, & Tangang, 2000; Timmermann, Voss, & Pasmanter, 2001; Wikle & Hooten, 2010). It remains an active area of research to develop nonlinear statistical models for long-lead forecasting, and there is a need to develop methods that are computationally efficient, are skillful, and can provide realistic uncertainty quantification in the presence of multiple time and spatial scales.

Statistical approaches for nonlinear DSTMs have focused on accommodating the quadratic nonlinearity that is present in many mechanistic models of such systems (Kravtsov, Kondrashov, & Ghil, 2005; Richardson, 2017; Wikle & Hooten, 2010). These models, at least when implemented in a way that fully accounts for uncertainty in data, process, and parameters, can be quite computationally challenging, mainly due to the very large number of parameters that must be estimated. Solutions to this challenge require reducing the dimension of the state space, regularizing the parameter space, the incorporating of additional information (prior knowledge), and novel computational approaches (see the summary in Wikle, 2015). Parsimonious alternatives include analog methods (e.g., McDermott & Wikle, 2016; Zhao & Giannakis, 2016) and individual (agent)-based models (e.g., Hooten & Wikle, 2010).

RNNs provide an alternative approach to model multiple scale nonlinear spatio-temporal processes. In essence, RNNs are a type of artificial neural network, originally developed in the 1980s (Hopfield, 1982), that, unlike traditional feed-forward neural networks, includes memory and allows cycles that can process sequences in their hidden layers. That is, unlike feed-forward neural networks, RNNs explicitly account for the dynamical structure of the data. This has made them ideal for applications in natural language processing, speech recognition, and image captioning. These methods have not been used extensively for spatio-temporal prediction, although there are notable exceptions (Dixon, Polson, & Sokolov, 2017; McDermott & Wikle, 2017a). Like the quadratic nonlinear spatio-temporal DSTMs in statistics, these models have a very large number of parameters (weights), can be quite difficult to tune and train, and are computationally intensive. One way to get the advantages of a RNN within a more parsimonious parameter estimation context is through the use of reservoir computing methods—the most common of which is the ESN (Jaeger, 2001). In this case, the hidden states and inputs evolve in a dynamical reservoir in which the parameters (weights) that describe their evolution are drawn at random with most (e.g., 90% or upwards) assumed to be zero. Then, the only parameters that are estimated are

the output parameters (weights) that connect the hidden states to the output response. For example, in the context of continuous spatio-temporal responses, this is just a regression estimation problem (usually with a regularization penalty; e.g., ridge regression). These models will be developed in Section 3.1.

Historically, the reservoir parameters in the ESN are just chosen once, with fairly large hidden state dimensions. Although this often leads to good predictions, it provides no opportunity for uncertainty quantification. An alternative is to perform parametric bootstrap or ensemble estimation, in which multiple reservoir samples are drawn (e.g., McDermott & Wikle, 2017b; Sheng, Zhao, Wang, & Leung, 2013). This provides a measure of uncertainty quantification and allows one to choose smaller hidden state dimensions, essentially building an ensemble of weak learners analogous to many methods in machine learning (e.g., Friedman, Hastie, & Tibshirani, 2001). There have also been Bayesian implementations of the ESN model (e.g., Chatzis, 2015; Li, Han, & Wang, 2012), but none of these has been implemented within a Markov chain Monte Carlo (MCMC) framework, as is the case here, where multiple levels of uncertainties can be accounted for. In the context of spatio-temporal forecasting, McDermott and Wikle (2017a, 2017b) have shown that augmenting the traditional ESN with quadratic output terms (analogous to the quadratic nonlinear component in statistical DSTMs) and input embeddings (e.g., including lags of the input variables as motivated by Takens's (1981) representation in dynamical systems) can improve forecast accuracy compared to traditional ESN models.

Deep models have recently shown success in many neural network applications in machine learning (e.g., Krizhevsky, Sutskever, & Hinton, 2012). As mentioned above, deep (hierarchical) statistical models have also been shown to be effective in complex spatio-temporal models. There are challenges in training such models given the very large number of parameters (weights), so it can be advantageous to consider deep models that are also relatively parsimonious in their parameter space. It is then natural to explore the potential for deep or hierarchical ESN models. Thus, the proposed methodology provides a computationally efficient alternative to traditional deep models. The purpose of adding additional layers in the ESN framework is to model (learn) additional temporal scales (Jaeger, 2007). Deep ESN models provide a greater level of flexibility by allowing individual layers to potentially represent different time scales. The model presented here exploits the multiscale nature of deep ESN models for spatio-temporal forecasting.

Although deep ESN models have been considered in the engineering literature (Antonelo, Camponogara, & Foss, 2017; Jaeger, 2007; Ma, Shen, & Cottrell, 2017; Triefenbach, Jalalvand, Demuynck, & Martens, 2013), none of these approaches accommodates uncertainty quantification. Furthermore, these methods do not include a spatial component nor are they applied to spatio-temporal systems. Here, we develop a hierarchical ESN approach for spatio-temporal prediction that explicitly accounts for uncertainty quantification. Little, if any, of the research in the ESN literature has considered the ESN model within a hierarchical Bayesian framework, as is pursued here. We first present an ensemble approach, extending the work of McDermott and Wikle (2017b) to the deep setting and, then, consider a hierarchical Bayesian implementation of the deep ESN model that more rigorously accounts for observation model uncertainty.

Section 2 describes the motivating spatio-temporal long-lead forecasting problem, in this case, using Pacific sea surface temperature (SST) anomalies to forecast soil moisture anomalies over the midwest U.S. corn producing region (i.e., corn belt) six months in the future. Section 3 describes the hierarchical ESN methodology, first from the ensemble perspective and then the Bayesian perspective. Section 4 presents a simulation study using a non-Gaussian multiscale Lorenz-96 system and then presents the soil moisture forecasting example. We conclude with a brief discussion in Section 5.

2 | MOTIVATING PROBLEM: LONG-LEAD FORECASTING

In the context of atmospheric and oceanic processes, *long-lead forecasting* refers to forecasting atmospheric, oceanic, or related variables on monthly to yearly time scales. Although it is fundamentally impossible to generate skillful meteorological forecasts of atmospheric processes on time horizons of greater than about ten days to two weeks, the ocean operates on much longer time scales than the atmosphere and provides a significant amount of the forcing of atmospheric processes. Thus, this linkage between the ocean and the atmosphere can lead to skillful long-lead forecasts of the atmospheric state, or other processes linked to the atmospheric state (e.g., soil moisture) on time scales of months to a year (e.g., Philander, 1990). In this case, one cannot typically generate skillful meteorological point forecasts but can provide general distributional forecasts that are skillful, relative to naïve models such as climatology (long-term averages) or persistence (assuming current conditions persist into the future).

Historically, successful long-lead forecasting applications are typically tied to the El Niño/Southern Oscillation (ENSO) phenomenon, which shows quasiperiodic variability between warmer than normal ocean states in the central and eastern tropical Pacific ocean (El Niño) and colder than normal ocean states in the central tropical Pacific (La Niña). The ENSO

phenomenon accounts for the largest amount of variability in the tropical Pacific ocean and leads to world wide impacts due to atmospheric "teleconnections" (i.e., the shifting of the warm pools in the tropical Pacific also shifts the convective clusters of precipitation that drive upper atmospheric wave trains that in turn influence the atmospheric circulation; e.g., locations of jet streams). These atmospheric circulation changes then affect temperature, precipitation, and many responses to those variables such as habitat conditions for ecological processes, soil moisture, and severe weather, as described in Philander (1990). It has been demonstrated for over two decades that skillful long-lead forecasts of SST in the tropical Pacific (and, hence, ENSO) is possible with both deterministic and statistical models. In fact, this is one of the aforementioned situations in the atmospheric and ocean sciences where statistical forecasting is as good as, or better than, deterministic models (e.g., Barnston et al., 1999; Jan van Oldenborgh et al., 2005).

There are essentially two general approaches to the long-lead forecasting of a response to SST forcing. One approach is to generate a long-lead forecast of SST (e.g., a six-month forecast) and then use contemporaneous relationships between the ocean state and the response of interest (e.g., midwest soil moisture) to generate a long-lead forecast of that response. This typically requires a dynamical forecast of SST and, then, some regression or classification model from SST to the outcome of interest. The alternative is to model the relationship between SST and the future response at a chosen lead time (e.g., forecasting midwest soil moisture in May given SST in November). In essence, this is a spatio-temporal regression, where one is predicting a spatial field in the future given a spatial field at the current time. In either case, linear models have been shown to perform reasonably well in these situations (e.g., Van den Dool, Huang, & Fan, 2003), but it is typically the case that models that include nonlinear interactions can perform more skillfully and can produce more realistic long-lead forecast distributions (e.g., Fischer, Seneviratne, Vidale, Lüthi, & Schär, 2007; McDermott & Wikle, 2016; Sheffield, Goteti, Wen, & Wood, 2004).

2.1 | Long-lead forecasting of soil moisture in the Midwest U.S. Corn Belt

Soil moisture is fundamentally important to many processes (e.g., agricultural production, hydrological runoff). In particular, the amount of soil moisture available to crops such as wheat and corn at certain critical phases of their growth cycle can make a significant impact on yield (Carleton, Arnold, Travis, Curran, & Adegoke, 2008). Thus, having a long-lead understanding of the plausible distribution of soil moisture over an expansive area of agricultural production can assist producers by suggesting optimal management approaches (e.g., timing of planting, nutrient supplementation, and irrigation). Given the aforementioned links between tropical Pacific SST and North American weather patterns, it is not surprising that skillful long-lead forecasts of soil moisture in major production areas of the U.S. are possible (e.g., Van den Dool et al., 2003). Indeed, the U.S. National Oceanic and Atmospheric Administration (NOAA) and the National Center for Environmental Prediction (NCEP) routinely provide soil moisture outlooks (forecasts) that are based on a combination of deterministic and statistical (constructed analog) models (e.g., see http://www.cpc.ncep.noaa.gov/soilmst/index_jh.html), although for shorter lead times than of interest here. Recently, McDermott and Wikle (2016) have shown that a Bayesian analog forecasting model could provide skillful high-spatial-resolution forecasts of soil moisture anomalies over the state of Iowa in the United States at lead times up to six months.

The application here will consider the problem of forecasting soil moisture over the midwest U.S. corn belt in May given data from the previous November (i.e., a six-month lead time). We consider May soil moisture because it is an important time period for planting corn in this region (e.g., Blackmer, Pottker, Cerrato, & Webb, 1989). This application corresponds to the long-lead spatio-temporal field regression approach to nowcasting described above, that is, we regress a tropical Pacific SST field in November onto the soil moisture field from the following May. The details associated with the data and model used for this example are given in Section 4.3.

3 | METHODOLOGY

Suppose we are interested in forecasting the spatio-temporal process $\mathbf{Z}_t \equiv (Z_t(\mathbf{s}_1), \ldots, Z_t(\mathbf{s}_{n_z}))'$ at a discrete set of spatial locations $\{\mathbf{s}_i \in D_z \subset \mathbb{R}^2 : i = 1, \ldots, n_z\}$ for time periods $\{t = 1, \ldots, T + \tau\}$, where T is the last time at which one has data and τ is an integer corresponding to the forecast lead time. Using a chosen linear dimension reduction method, $\mathbf{Z}_t$ can be decomposed such that $\mathbf{Z}_t \approx \mathbf{\Phi}\boldsymbol{\alpha}_t$, where $\mathbf{\Phi}$ is an $n_z \times n_b$ matrix of n_b spatial basis functions and $\boldsymbol{\alpha}_t$ is an n_b-dimensional vector of basis coefficients, indexed by $b = 1, \ldots, n_b$ (e.g., Cressie & Wikle, 2011, chapter 7). Next, assume we have inputs corresponding to a spatio-temporal data set. Specifically, let $\mathbf{x}_t \equiv (x_t(\mathbf{r}_1), \ldots, x_t(\mathbf{r}_{n_x}))'$ be a vector of n_x input variables that correspond to a discrete set of spatial locations $\{\mathbf{r}_d \in D_z \subset \mathbb{R}^2 : d = 1, \ldots, n_x\}$ at time t. (More generally, the input vector may consist of n_x time-varying input covariates that are not indexed by space; e.g., $\{\mathbf{r}_d \equiv r_d \subset R_d : d = 1, \ldots, n_x\}$).

Below, we provide a brief overview of the ensemble ESN model considered in McDermott and Wikle (2017b), followed by the multilevel (deep) extension that we call a deep ensemble ESN (D-EESN) model. We then describe a Bayesian version of the D-EESN model that we label (BD-EESN).

3.1 | Basic ESN background

The quadratic ESN model outlined in McDermott and Wikle (2017b) is given by the following:

$$\textbf{Data stage:} \quad \mathbf{Z}_t \approx \mathbf{\Phi}\boldsymbol{\alpha}_t \tag{1}$$

$$\textbf{Output stage:} \quad \boldsymbol{\alpha}_t = \mathbf{V}_1\mathbf{h}_t + \mathbf{V}_2\mathbf{h}_t^2 + \boldsymbol{\eta}_t, \;\; \boldsymbol{\eta}_t \sim \text{Gau}\left(\mathbf{0}, \sigma_\eta^2\mathbf{I}\right) \tag{2}$$

$$\textbf{Hidden stage:} \quad \mathbf{h}_t = g_h\left(\frac{\nu}{|\lambda_w|}\mathbf{W}\mathbf{h}_{t-1} + \mathbf{U}\tilde{\mathbf{x}}_t\right), \tag{3}$$

where $\mathbf{\Phi}$ is an $n_z \times n_b$ matrix of spatial basis functions, $\boldsymbol{\alpha}_t$ is an n_b-vector that contains the associated basis expansion coefficients (where $n_b << n_z$), $\mathbf{h}_t$ is an n_h-dimensional vector of hidden units, $g_h(\cdot)$ is a nonlinear activation function (e.g., a sigmoidal function such as a hyperbolic tangent function), λ_w is the spectral radius (largest eigenvalue) of $\mathbf{W}$, ν is a scaling parameter, and $\mathbf{W}$, $\mathbf{U}$, $\mathbf{V}_1$, and $\mathbf{V}_2$ are weight (parameter) matrices of dimension $n_h \times n_h$, $n_h \times n_{\tilde{x}}$, $n_b \times n_h$, and $n_b \times n_h$, respectively (defined below). The square parameter matrix $\mathbf{W}$ can be thought of analogously to a transition matrix in a vector autoregressive model in that it can capture transition dynamic interactions between various inputs. The scaling parameter ν helps control the amount of memory in the system and is restricted such that $0 \leq \nu \leq 1$ for stability purposes (Jaeger, 2007). We let $\tilde{\mathbf{x}}$ be an $n_{\tilde{x}}$-vector of m embeddings (lagged input values) given by

$$\tilde{\mathbf{x}}_t = \left[\mathbf{x}_t', \mathbf{x}_{t-\tau}', \mathbf{x}_{t-2\tau}', \ldots, \mathbf{x}_{t-m\tau}'\right]', \tag{4}$$

where each $\mathbf{x}_t'$ in $\tilde{\mathbf{x}}_t$ is a vector. Importantly, only σ_η^2 and the weight matrices $\mathbf{V}_1$ and $\mathbf{V}_2$ in the output stage (2) are estimated (usually with a ridge penalty). These weight matrices can be thought of similarly to regression parameters that weight the hidden units appropriately. In contrast, the elements of the "reservoir weight" matrices $\mathbf{W}$ and $\mathbf{U}$ in (3) are drawn from the following distributions:

$$\mathbf{W} = [w_{k,q}]_{k,q} \,:\, w_{k,q} = \gamma_{k,q}^w \,\text{Unif}(-a_w, a_w) + \left(1 - \gamma_{k,q}^w\right)\,\delta_0, \tag{5}$$

$$\mathbf{U} = [u_{k,r}]_{k,r} \,:\, u_{k,r} = \gamma_{k,r}^u \,\text{Unif}(-a_u, a_u) + \left(1 - \gamma_{k,r}^u\right)\,\delta_0, \tag{6}$$

$$\gamma_{k,q}^w \sim \text{Bern}(\pi_w), \;\; \gamma_{k,r}^u \sim \text{Bern}(\pi_u), \tag{7}$$

where $\gamma_{k,q}^w$ and $\gamma_{k,r}^u$ denote indicator variables, δ_0 denotes a Dirac function, and π_w (and π_u) can be thought of as the probability of including a particular weight (parameter) in the model. As is common in the machine learning literature, both a_w and a_u are also set to small values to help prevent overfitting. Similarly, the parameters π_w and π_u are set to small values to create a sparse network. Both the sparseness and randomness act as a regularization mechanism, which prevents the ESN model from overfitting to in-sample data. In summary, note that the hidden (reservoir) stage in (3) corresponds to a *nonlinear stochastic transformation of the input vectors* that are then regressed, with regularization, onto the output vectors in the output stage (i.e., (2) above), which are then transformed back to the data scale in (1).

Traditional ESN models (e.g., Jaeger, 2007; Lukoševičius & Jaeger, 2009) do not typically include the quadratic ($\mathbf{V}_2$) term in the output stage nor do they include the embeddings in the input vector, but McDermott and Wikle (2017b) found that those are often helpful when forecasting spatio-temporal processes. In addition, traditional ESN applications typically include a "leaking rate" parameter that corresponds to a convex combination of the previous hidden state $\mathbf{h}_{t-1}$ and the current reservoir value, $\mathbf{h}_t$ (e.g., Lukoševičius, 2012). We have not found this to be as useful for long-lead spatio-temporal forecasting as it has been in other applications and so omit it in our presentation for notational simplicity.

To facilitate uncertainty quantification, McDermott and Wikle (2017b) were motivated by parametric bootstrap prediction methods (e.g., Genest & Rémillard, 2008; Sheng et al., 2013) to consider an ensemble of predictions from the quadratic ESN model. In particular, algorithm 1 of McDermott and Wikle (2017b) provides a simple procedure that generates n_{res} forecast realizations $\{\widehat{\mathbf{Z}}_t^{(j)} = \mathbf{\Phi}\widehat{\boldsymbol{\alpha}}_t^{(j)} : j = 1, \ldots, n_{\text{res}}\}$ by implementing the reservoir model in (3) to obtain $\hat{\boldsymbol{\alpha}}_t^{(j)}$ in (2) through the use of n_{res} simulated (sample) weight matrices from (5) and (6). These samples can then be used in a Monte Carlo sense to obtain features of the predictive distribution (e.g., means and variances). As discussed in McDermott and Wikle (2017b), this bootstrap approach using fairly simple quadratic ensemble ESN (Q-EESN) models

acts as an ensemble of weaker learners and was shown to adequately capture the uncertainty associated with long-lead forecasting of tropical Pacific SST.

3.2 | Deep ensemble ESN (D-EESN)

Here, we develop a deep extension of the bootstrap-based Q-EESN model of McDermott and Wikle (2017b) described in Section 3.1. That model has multiple levels but is not a deep model in the sense that it has no mechanism to link multiple hidden layers, which might be important for processes that occur on multiple time scales. To accommodate such structure, we extend some of the deep ESN model components developed in Ma et al. (2017) and Antonelo et al. (2017) to a spatio-temporal ensemble framework in the following D-EESN model. In particular, the D-EESN model with $\ell = 1, \dots, L$ hidden layers is defined as follows for time period t and $L \geq 2$:

$$\textbf{Data Stage:} \quad \mathbf{Z}_t \approx \mathbf{\Phi}\boldsymbol{\alpha}_t \tag{8}$$

$$\textbf{Output Stage:} \quad \boldsymbol{\alpha}_t = \mathbf{V}_1\mathbf{h}_{t,1} + \sum_{\ell=2}^{L}\mathbf{V}_\ell g_h(\widetilde{\mathbf{h}}_{t,\ell}) + \boldsymbol{\eta}_t,\ \boldsymbol{\eta}_t \sim \text{Gau}\left(\mathbf{0}, \sigma_\eta^2\mathbf{I}\right), \quad \text{s.t. } \mathbf{v}'_{\ell,b}\mathbf{v}_{\ell,b} \leq c_v, \tag{9}$$

$$\textbf{Hidden Stage } \ell\textbf{:} \quad \mathbf{h}_{t,\ell} = f\left(\frac{\nu_\ell}{\lambda_{W_\ell}}\mathbf{W}_\ell\mathbf{h}_{t-1,\ell} + \mathbf{U}_\ell\widetilde{\mathbf{h}}_{t,\ell+1}\right), \ \text{ for } \ \ell < L, \tag{10}$$

$$\textbf{Reduction Stage } \ell+1\textbf{:} \quad \widetilde{\mathbf{h}}_{t,\ell+1} \equiv \mathcal{Q}(\mathbf{h}_{t,\ell+1}), \ \text{ for } \ell < L, \tag{11}$$

$$\textbf{Input Stage:} \quad \mathbf{h}_{t,L} = f\left(\frac{\nu_L}{\lambda_{W_L}}\mathbf{W}_L\mathbf{h}_{t-1,L} + \mathbf{U}_L\widetilde{\mathbf{x}}_t\right), \tag{12}$$

where $\mathbf{h}_{t,\ell}$ is an $n_{h,\ell}$-dimensional vector for the ℓth hidden layer and each parameter matrix $\mathbf{V}_\ell$ is defined as

$$\mathbf{V}_\ell \equiv \begin{bmatrix} \mathbf{v}'_{\ell,1} \\ \vdots \\ \mathbf{v}'_{\ell,n_b} \end{bmatrix}. \tag{13}$$

Let $\mathbf{h}_{1:T,\ell+1} \equiv [\mathbf{h}_{1,\ell+1}, \dots, \mathbf{h}_{T,\ell+1}]'$ be a $T \times n_{h,\ell+1}$ matrix. The function $\mathcal{Q}(\cdot)$ in (11) denotes a dimension reduction mapping function of this matrix (see below for specific examples) such that $\mathcal{Q} : \mathbf{h}_{1:T,\ell+1} \rightarrow \widetilde{\mathbf{h}}_{1:T,\ell+1}$ for $\widetilde{\mathbf{h}}_{1:T,\ell+1} \equiv \mathcal{Q}(\mathbf{h}_{1:T,\ell+1})$, where $\widetilde{\mathbf{h}}_{1:T,\ell+1}$ is a $T \times n_{\tilde{h},\ell+1}$ matrix such that $n_{h,\ell+1} \geq n_{\tilde{h},\ell+1}$. More concisely, this transformation is on the matrix $\mathbf{h}_{1:T,\ell+1}$, resulting in the transformed matrix $\widetilde{\mathbf{h}}_{1:T,\ell+1}$, and then, for each time period, the appropriate $\widetilde{\mathbf{h}}_{t,\ell+1}$ is extracted from the transformed matrix to construct (10). Through the inclusion of the $\widetilde{\mathbf{h}}_{t,\ell}$ terms in (9), the model can potentially weight layers lower down in the hierarchy that may represent different time scales or features (i.e., Ma et al., 2017). Note that the activation function $g_h(\cdot)$ is included in (9) to place the dimension reduction variables on a similar scale as the $\mathbf{h}_{t,1}$ variables (i.e., the hidden variables that have not gone through the dimension reduction transformation). See chapter 7 in Cressie and Wikle (2011) for other examples of deep hierarchical models and Ma et al. (2017) for a deep ESN example.

For a given hidden unit ℓ, λ_{w_ℓ} denotes the largest eigenvalue of the square matrix $\mathbf{W}_\ell$, where as in the basic ESN model above. $\mathbf{W}_\ell = [w^{(\ell)}_{k_\ell,q_\ell}]_{k_\ell,q_\ell}$ is drawn from the following distribution:

$$w^{(\ell)}_{k_\ell,q_\ell} = \gamma^{w_\ell}_{k_\ell,q_\ell}\ \text{Unif}\left(-a_{w_\ell}, a_{w_\ell}\right) + \left(1 - \gamma^{w_\ell}_{k_\ell,q_\ell}\right)\ \delta_0, \quad \gamma^{w_\ell}_{k_\ell,q_\ell} \sim \text{Bern}\left(\pi_{w_\ell}\right), \tag{14}$$

and each element of the parameter matrix $\mathbf{U}_\ell = [u^{(\ell)}_{k_\ell,r_\ell}]_{k_\ell,r_\ell}$ is drawn from

$$u^{(\ell)}_{k_\ell,r_\ell} = \gamma^{u_\ell}_{k_\ell,r_\ell}\ \text{Unif}\left(-a_{u_\ell}, a_{u_\ell}\right) + \left(1 - \gamma^{u_\ell}_{k_\ell,r_\ell}\right)\ \delta_0, \quad \gamma^{u_\ell}_{k_\ell,r_\ell} \sim \text{Bern}\left(\pi_{u_\ell}\right). \tag{15}$$

The embedded input vector $\widetilde{\mathbf{x}}_t$ is defined as in (4) above. Estimation of $\mathbf{V}_1, \dots, \mathbf{V}_L$ is carried out through the use of ridge regression using the ridge hyperparameter r_v, where there is a one-to-one relationship between r_v and the constant c_v in (9). Note that we do not include a quadratic output term in this model as we did in (2) for simplicity, but this could easily be added if the application warrants (note that both applications presented here did not benefit from the addition of quadratic output terms in the deep model setting).

The bootstrap ensemble prediction for the D-EESN model is presented in Algorithm 1 below. In particular, Algorithm 1 starts by drawing $\mathbf{W}_\ell$ and $\mathbf{U}_\ell$ for every layer in the D-EESN model. Next, these parameters are evaluated sequentially, starting with the input layer defined in (12) and ending with the hidden layer that comes directly before the output stage (i.e., (10) for $\ell = 1$). Finally, with all of the hidden states calculated, ridge regression is used to estimate the regression parameters in (9).

Algorithm 1 D-EESN algorithm

Data: $\{\boldsymbol{\alpha}_t, \tilde{\mathbf{x}}_t : t = 1, \ldots, T\}$

Input: hyper-parameters: $\{\nu_1, \ldots, \nu_L, n_{\tilde{h},2}, \ldots, n_{\tilde{h},L}, n_{h,1}, r_v\}$, chosen by cross-validation.

for $j = 1, \ldots, n_{res}$ **do**

Simulate $\mathbf{W}_\ell$ and $\mathbf{U}_\ell$ from (14) and (15) for $\ell = 1, \ldots, L$

Calculate $\{\mathbf{h}_{t,L} : t = 1, \ldots, T\}$ with (12) using $\mathbf{W}_L$ and $\mathbf{U}_L$

for $\ell = L - 1$ *to* 1 **do**

Calculate $\{\widetilde{\mathbf{h}}_{t,\ell+1} : t = 1, \ldots, T\}$ using (11)

Calculate $\{\mathbf{h}_{t,\ell} : t = 1, \ldots, T\}$ with (10) using $\mathbf{W}_\ell$ and $\mathbf{U}_\ell$

end

Use ridge regression to calculate $\mathbf{V}_1^{(j)}, \ldots, \mathbf{V}_L^{(j)}$

Calculate $\widehat{\boldsymbol{\alpha}}_t^{(j)}$ from (9)

Calculate out-of-sample forecasts $\{\widehat{\mathbf{Z}}_t^{(j)} = \boldsymbol{\Phi}\widehat{\boldsymbol{\alpha}}_t^{(j)} : t = T + 1, \ldots, T + n_f\}$

end

Output: Ensemble of forecasts $\{\widehat{\mathbf{Z}}_t^{(j)} : t = T + 1, \ldots, T + n_f ; j = 1, \ldots, n_{res}\}$

The reservoir hyperparameters $\{\nu_1, \ldots, \nu_L, n_{\tilde{h},2}, \ldots, n_{\tilde{h},L}, n_{h,1}, r_v\}$ are chosen by cross-validation driven by a genetic algorithm (GA; see Section 4.1 for further details). As in McDermott and Wikle (2017b), we found that fixing the parameters $\{\pi_{w_1}, \ldots, \pi_{w_L}, \pi_{u_1} \ldots, \pi_{u_L}, a_{w_1}, \ldots, a_{w_L}, a_{u_1}, \ldots, a_{u_L}\}$ and the number of hidden units for all of the layers except the first (i.e., $\{n_{h,2}, \ldots, n_{h,L}\}$) was a reasonable assumption here because the dimension of all the hidden units besides the top layer (i.e., $n_{h,1}$) are eventually reduced using the dimension reduction transformation. See Appendix A for details on specific choices for these parameters.

Note that this model consists of a series of *linked nonlinear stochastic transformations of the input vector* that are available for prediction. In addition, the data reduction steps act similarly to the pooling step in a convolutional neural network (Krizhevsky et al., 2012). That is, it simultaneously reduces the dimension of the hidden layers and provides a summary of the important features. Similar to how ridge regression acts to reduce redundancy in the classic ESN model, dimension reduction serves this purpose in the deep framework. Due to the limited amount of research on deep ESNs, the question of which dimension reduction method to use for $Q(\cdot)$ in (11) is still an open research question. Here, principal component analysis (PCA) and Laplacian eigenmaps (Belkin & Niyogi, 2001) were explored as possible choices for $Q(\cdot)$. Although these methods were selected to represent both linear and nonlinear dimension reduction techniques, there are certainly other choices that could be explored in future applications.

3.3 | Bayesian deep ensemble ESN (BD-EESN)

The D-EESN model given in Section 3.2 is very efficient to implement, but at a potential cost of not fully accounting for all sources of uncertainty in estimation and prediction. In particular, the data stage given in (8) does not account for truncation error in the basis expansion nor the error associated with the estimates of the regression or residual variance parameters in (9). This can easily be remedied via Bayesian estimation at these stages. To our knowledge, this is the first ESN model for spatio-temporal data to be implemented within a traditional hierarchical Bayesian framework.

We develop the Bayesian deep EESN (BD-EESN) model in a general form here so that it can be applied to both the traditional EESN and the D-EESN described above. In particular, let

$$\textbf{Data stage:} \quad \mathbf{Z}_t|\boldsymbol{\alpha}_t \sim \mathrm{D}(\boldsymbol{\mu}(\boldsymbol{\alpha}_t), \Theta), \tag{16}$$

$$\textbf{Output stage:} \quad \boldsymbol{\alpha}_t = \frac{1}{n_{\text{res}}} \sum_{j=1}^{n_{\text{res}}} \left[\boldsymbol{\beta}_1^{(j)} \mathbf{h}_{t,1}^{(j)} + \sum_{\ell=2}^{L} \boldsymbol{\beta}_\ell^{(j)} g_h\left(\tilde{\mathbf{h}}_{t,\ell}^{(j)}\right) \right] + \boldsymbol{\eta}_t, \tag{17}$$

where $\boldsymbol{\eta}_t \sim \text{Gau}(\mathbf{0}, \sigma_\eta^2 \mathbf{I})$ and D denotes an unspecified distribution; for both of the applications discussed here, $\boldsymbol{\mu}(\boldsymbol{\alpha}_t) \equiv \boldsymbol{\Phi}\boldsymbol{\alpha}_t$ and $\boldsymbol{\Theta} \equiv \boldsymbol{\Sigma}_z$. Here, $\boldsymbol{\Sigma}_z$ is defined as a (known) $n_z \times n_z$ spatial covariance matrix (e.g., accommodating the basis truncation and/or measurement error); see Section 4 for application-specific details. The regression matrices in (17) are defined as follows for $\ell = 1, \ldots, L$ and $j = 1, \ldots, n_{\text{res}}$:

$$\boldsymbol{\beta}_\ell^{(j)} \equiv \begin{bmatrix} \boldsymbol{\beta}_{\ell,1}^{(j)\prime} \\ \vdots \\ \boldsymbol{\beta}_{\ell,n_b}^{(j)\prime} \end{bmatrix}. \tag{18}$$

In this notation, $\mathbf{h}_{t,1}^{(j)}$ is the jth sampled $n_{h,1}$-dimensional vector from an $n_{h,1} \times T$ dynamical reservoir generated as in the D-EESN model in Section 3.2; that is, we generate $j = 1, \ldots, n_{\text{res}}$ reservoir samples offline using (10) from the D-EESN model (for $\ell = 1$). Similarly, $\widetilde{\mathbf{h}}_{t,2}^{(j)}, \ldots, \widetilde{\mathbf{h}}_{t,L}^{(j)}$ are also sampled a priori and come from the dimension reduction stage(s) of the D-EESN model (i.e., (12) above). That is, both $\mathbf{h}_{t,1}^{(j)}$ and $\{\widetilde{\mathbf{h}}_{t,\ell}^{(j)} : \ell = 2, \ldots, L\}$ are treated as fixed covariates for the BD-EESN model. Therefore, the output model takes an ensemble approach in terms of generating a suite of nonlinear stochastic transformed input variables for a Bayesian regression, where each $\{\boldsymbol{\beta}_\ell^{(j)} : \ell = 1, \ldots, L\}$ are matrices of regression parameters as defined in (18). Note the obvious similarity between the process model in (9) and the one for the D-EESN model in (17). If all of the $\boldsymbol{\beta}_\ell^{(j)}$ terms for $\ell \geq 2$ are set to zero, the process model in (17) can be used with the traditional EESN model (or the Q-EESN model by simply adding quadratic terms). Although the BD-EESN can still be implemented as a Bayesian model without using a distribution in (16), including a distribution at this stage allows the model to more rigorously account for uncertainty associated with the data and the rank-reduced dimension reduction.

This model is clearly overparameterized, so the regression parameters in the BD-EESN model are given stochastic search variable selection (SSVS) priors (George & McCulloch, 1997). While one of the many variable selection priors from the Bayesian variable selection literature can be used here, we choose to use SSVS priors for their ability to produce efficient shrinkage. In particular, SSVS priors can shrink a (large) percentage of parameters to zero (in a spike-and-slab implementation) or close to zero, while leaving the remaining variables unconstrained. Thus, the regression parameters in the BD-EESN model are given the following hierarchical prior distribution for $\ell = 1, \ldots, L$ and $j = 1, \ldots, n_{\text{res}}$:

$$\begin{aligned} \beta_{\ell,\text{b},k_\ell}^{(j)} \;\Big|\; \gamma_\ell^{\beta_\ell} &\sim \gamma_\ell^{\beta_\ell}\, \text{Gau}\left(0, \sigma_{\beta_\ell,0}^2\right) + \left(1 - \gamma_\ell^{\beta_\ell}\right)\, \text{Gau}\left(0, \sigma_{\beta_\ell,1}^2\right), \\ \gamma_\ell^{\beta_\ell} &\sim \text{Bernoulli}(\pi_{\beta_\ell}), \end{aligned} \tag{19}$$

where k_ℓ indexes the hidden units for a particular layer, $\sigma_{\beta_\ell,0}^2 \gg \sigma_{\beta_\ell,1}^2$, and π_{β_ℓ} is the prior probability of including a particular variable in the model. Finishing the prior specifications for the model, the variance parameter σ_η^2 is given an inverse-gamma prior such that $\sigma_\eta^2 \sim \text{IG}(\alpha_\eta, \beta_\eta)$. The hyperparameters $\sigma_{\beta_\ell,0}^2, \sigma_{\beta_\ell,1}^2, \pi_{\beta_\ell}, \alpha_\eta$, and β_η are problem specific (see the examples in Section 4.1 below).

Because the parameters here are given conjugate priors, the full-conditional distributions that make up the Gibbs sampler MCMC algorithm needed to implement this model are straightforward to sample from. A summary of the entire estimation procedure for the BD-EESN can be found in Algorithm 2. Forecasts for the BD-EESN are made in a similar manner as the D-EESN (as shown by the output step of Algorithm 1). That is, training is carried out using data up to time period T, whereas out-of-sample forecasts are continually made τ periods into the future using the corresponding lagged input variable to generate the appropriate hidden state variables. Although other Bayesian computation strategies could be pursued here, we implement the proposed methodology using MCMC estimation techniques for their natural convenience with hierarchical models defined in a conditional manner. Finally, at each iteration of the MCMC algorithm, the sampled regression parameters are used with these hidden states to produce out-of-sample forecasts.

Algorithm 2 Outline of estimation procedure for the BD-EESN.

1. Use Algorithm 1 from the D-EESN model with a genetic algorithm-based cross-validation to pick the reservoir hyper-parameters that make up the D-EESN model (i.e., the input for Algorithm 1).
2. Using the hyper-parameters selected in Step 1, generate n_{res} reservoirs from the hidden layers of the D-EESN model to be used in the Bayesian model as covariates.
3. Use Gibbs sampling to estimate $\boldsymbol{\alpha}_{1:T}, \boldsymbol{\beta}_1^{(1:n_{res})}, \ldots, \boldsymbol{\beta}_L^{(1:n_{res})}$, and σ_η^2, while treating the n_{res} reservoirs generated in Step 2 as (fixed) covariates.

4 | SIMULATION AND SOIL MOISTURE EXAMPLES

Here, we describe the model setup used to implement the D-EESN and BD-EESN models on a complex simulated process and for the soil moisture long-lead forecasting problem that motivated these models.

4.1 | Model setup

The previously mentioned cross-validation for the D-EESN is carried out using a GA (e.g., Sivanandam & Deepa, 2007) contained in the `GA` package (`https://cran.r-project.org/web/packages/GA`) from the R statistical computing program (`http://cran.r-project.us.org`). Unlike the basic ESN model, the number of hyperparameters for the deep EESN model can increase quickly as the number of layers increases (e.g., with five layers and a relatively coarse search grid, the number of total parameters in the search space can easily approach 10^6), thus rendering grid search approaches computationally burdensome at best (i.e., Ma et al., 2017). Through the use of a GA, the hyperparameters that make up a deep ESN can be selected at a fraction of the computational cost. The GA was implemented with 40 generations and a population size of 20 for all of the applications presented below. This is the first implementation, to our knowledge, of either a traditional or deep ESN within an ensemble framework using a GA. The bounds of the parameter search space for each of the hyperparameters in the D-EESN model can be found in Appendix A on Table A1.

All ensemble models are comprised of 100 ensembles; we did not find any of the applications to be overly sensitive to this choice. Using this number of ensembles represents a compromise between computational efficiency and achieving consistent (reproducible) results (e.g., we found that using less than approximately 30 ensembles produced unstable results, whereas values greater than 100 did not change the results significantly); see Appendix A for more details on the specification of model parameters. Note that previous bootstrap ensemble ESN papers have generally used far fewer ensembles (e.g., Sheng et al., 2013). For context, on a 2.3-GHz laptop, the D-EESN algorithm defined in Algorithm 1 takes 4.3 and 17.5 seconds for a two-layer and seven-layer model, respectively, using 100 ensembles with the Lorenz-96 application described in Section 4.2 below. Note that R code for both applications can be found at the following open source repository: `https://github.com/PatrickMcDermottResearch/BD-EESN`.

Regarding the previous discussed dimension reduction function for the reduction stage of the D-EESN model (i.e., (11) above), PCA basis functions were selected for both applications and models using cross-validation. Although, we should note that the model was not very sensitive to this choice among the basis functions considered. Next, for simplicity, we used the same dimension (i.e., $n_{\tilde{h},\ell}$) for each dimension reduction layer in the D-EESN model; a similar assumption was made in Ma et al. (2017). Similar to McDermott and Wikle (2017b), all of the hyperparameters in the set $\{\pi_{w_1}, \ldots, \pi_{w_L}, \pi_{u_1}, \ldots, \pi_{u_L}, a_{w_1}, \ldots, a_{w_L}, a_{u_1}, \ldots, a_{u_L}\}$ are fixed at 0.10. Finally, $n_{h,\ell}$ for $\ell \geq 2$ was set to 84 for all of the deep models. The model was not sensitive to this value with values ranging between 60 and 100 producing nearly identical results for the metrics evaluated below. As previously discussed, the bootstrap framework presented here aims to create an ensemble of weak learners, thus the selection of a moderate value for $n_{h,\ell}$.

Estimation of all the Bayesian models considered here is carried out using a Gibbs sampler MCMC algorithm with 5,000 iterations where the first 1,000 iterations are treated as burn-in for both applications. The trace plots showed no evidence of nonconvergence (more details regarding convergence are given below). The specific hyperparameters used for the Bayesian implementation can be found in Table 1. Note that more restrictive priors (in terms of regularization) are employed for the models with more regression parameters (i.e., models with more hidden layers). This improves the mixing of the MCMC algorithm along with preventing the model from overfitting.

TABLE 1 Specific hyperparameters used for the various BD-EESN implementations

Application	Priors
BD-EESN models with $L = 2$	$\pi_{\beta_\ell} = .25, \sigma^2_{\beta_\ell,0} = 5, \sigma^2_{\beta_\ell,1} = .001, \alpha_\eta = 1$, and $\beta_\eta = 1$ for $\ell = 1, 2$
BD-EESN models with $L > 2$	$\pi_{\beta_\ell} = .10, \sigma^2_{\beta_\ell,0} = 4, \sigma^2_{\beta_\ell,1} = .001, \alpha_\eta = 1$, and $\beta_\eta = 1$ for $\ell = 1, \ldots, L$

Note. The hyperparameters were selected so the in-sample MSE of the BD-EESN roughly matched the in-sample MSE for the corresponding D-EESN model. None of the applications was overly sensitive to the specified priors, with moderate variations from the values given below producing similar results. BD-EESN = Bayesian deep ensemble echo state network; D-EESN = deep ensemble echo state network.

Both the D-EESN and BD-EESN are compared against the previously described single-layered Q-EESN model in order to investigate the added utility of using a deep framework. In addition, naïve or simple forecasting methods are often employed for difficult long-lead forecasting problems such as the soil moisture application, to act as a baseline for comparison. We consider both a climatological and linear DSTM here as baseline models for the soil moisture application (for consistency, we also compare to the linear model in the deep Lorenz-96 simulation study described below). The linear DSTM model is defined as follows:

$$\mathbf{Z}_t|\boldsymbol{\alpha}_t, \Sigma_z \sim \text{Gau}(\Phi\boldsymbol{\alpha}_t, \Sigma_z), \tag{20}$$

$$\boldsymbol{\alpha}_t = \mathbf{M}\boldsymbol{\alpha}_{t-1} + \boldsymbol{\eta}_t^{(L)}, \tag{21}$$

where $\mathbf{M}$ is an $n_b \times n_b$ transition matrix, the innovation distribution is $\boldsymbol{\eta}_t^{(L)} \sim \text{Gau}(\mathbf{0}, \Sigma_\eta^{(L)})$, and Σ_z is an $n_z \times n_z$ residual covariance matrix. The model parameters are estimated via a two-stage least squares procedure. Specifically, the residual covariance matrix is estimated using the in-sample residuals from the projection of the data onto the empirical orthogonal function (EOF) basis functions $\boldsymbol{\Phi}$ and the innovation covariance matrix, $\Sigma_\eta^{(L)}$, and the transition matrix, $\mathbf{M}$, are then estimated by least squares on these projections, $\boldsymbol{\alpha}_t$ (see chapter 7 of Cressie & Wikle, 2011, for a comprehensive overview of linear DSTMs and their estimation.)

The models considered here are evaluated in terms of both out-of-sample prediction accuracy and the quality of their respective forecast distributions (i.e., uncertainty quantification). In particular, mean squared prediction error (MSPE), defined here as the average squared difference between out-of-sample realizations and forecasts averaged over space and time, is calculated for every model. The forecast distribution coverages are evaluated using the continuous ranked probability score (CRPS). The CRPS considers the quality of a given model's uncertainty quantification by comparing the forecast distribution with the observed values (see, e.g., Gneiting & Katzfuss, 2014). The classic CRPS formulation for Gaussian data (i.e., Gneiting, Raftery, Westveld, & Goldman, 2005) is used for the soil moisture application, whereas the deep Lorenz-96 simulation uses the closed form expression for CRPS with log-Gaussian data from Baran and Lerch (2015). Regarding the soil moisture application, as previously noted, standard forecasting methodologies for difficult long-lead problems are often compared to simpler forecast methods such as climatological or linear forecasts. The *skill score* (SS) is an evaluation metric that compares a forecasting method to some reference forecast (e.g., Wilks, 2001). Assuming one wants to compare a *reference* forecast to some *model* forecast in terms of MSPE, SS is defined as follows:

$$\text{SS} = 1 - \frac{\text{MSPE(Model)}}{\text{MSPE(Reference)}}. \tag{22}$$

In our application, SS is calculated for each location in the soil moisture application by calculating MSPE across time. Thus, by computing an SS for each spatial location, we can obtain a spatial field that shows where the new model improves upon a particular reference model.

4.2 | Simulation study: Deep Lorenz-96 model

The deterministic model of Lorenz (1996), often referred to as the Lorenz-96 model, is frequently used as a simulation model in the atmospheric science and dynamical systems literature because it incorporates the quadratic nonlinear interactions of the famous Lorenz (1963; i.e., butterfly) model in a one-dimensional spatial setting. A multiscale extension of this model (see Wilks, 2005) has gained popularity in the atmospheric science and applied mathematics literature for its ability to represent realistic nonlinear interactions between small and large scale variables (Grooms & Lee, 2015). That is, the multiscale Lorenz-96 model operates on multiple scales in both space and time, consisting of locations with slowly varying large-scale behavior and locations with fast varying small-scale behavior. Thus, the multiscale Lorenz-96 model represents a very relevant example for the multiscale deep EESN methodology developed here. To make this simulation model even more realistic, we extend it by adding an additional data stage to allow for non-Gaussian data types. We will refer to this simulation model as the deep Lorenz-96 model. To our knowledge, this deep Lorenz-96 model is novel.

The spatial structure of the Lorenz-96 system (as shown in Figure 1) is a one-dimensional circular structure (i.e., periodic boundary conditions) where each large-scale location is evenly spaced. Furthermore, each large-scale location is associated with J small-scale locations (see Figure 1). Using the process model parameterization from

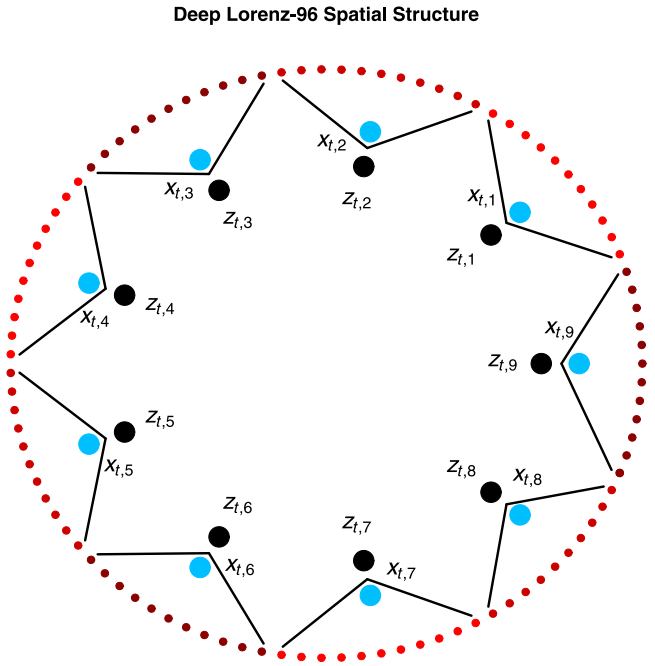

FIGURE 1 Description of the spatial structure for the deep Lorenz-96 model with nine large-scale locations that each have 11 associated small-scale locations, which gives 99 small-scale locations. For the sake of visualization, we have used a smaller number of large- and small-scale locations here than in the analyzed deep Lorenz-96 simulated data example. The various red-shaded small circles represent small-scale locations (i.e., $y_{j,k}$ in (23)), whereas the large blue and black circles denote the large-scale locations x_k and z_k, respectively, from the process model defined in (23)

Chorin and Lu (2015) and the inclusion of a data model, our deep Lorenz-96 model is defined as follows for time period t:

$$\begin{aligned} \text{Data Model:} \quad & z_{t,k} \sim \text{LogGaussian}\left(\frac{|x_{t,k}|}{c}, \sigma_\eta^2\right), \\ \text{Process Model:} \quad & \frac{dx_k}{dt} = x_{k-1}(x_{k+1} - x_{k-2}) - x_k + F + \frac{h_x}{J}\sum_j y_{j,k}, \\ & \frac{dy_{j,k}}{dt} = \frac{1}{\epsilon_L}[y_{j+1,k}(y_{j-1,k} - y_{j+2,k}) - y_{j,k} + h_y x_k], \end{aligned} \tag{23}$$

where c, F, ϵ_L, h_x, and h_y are user-defined parameters, $z_{t,k}$ and $x_{t,k}$ correspond to large-scale locations, $y_{j,k}$ corresponds to a small-scale location for $j = 1, \ldots, J$ and $k = 1, \ldots, K$, and $|\cdot|$ denotes the absolute value. The parameter F corresponds to a forcing term and helps determine the amount of nonlinearity in the model. In this setting, ϵ_L is a "time separation parameter" such that smaller values lead to a faster varying temporal scale for the small-scale locations. The contribution of the small-scale locations to the large-scale locations (and vice versa) is determine by h_x and h_y, respectively. Finally, the scaling parameter c helps ensure that the log-Gaussian data model does not produce unrealistically large realizations (we let $c = 2$ here).

Because the methodology presented here concerns multiscale spatio-temporal modeling, the parameters for the deep Lorenz-96 model are selected to emphasize the multiscale behavior in the process (as can be seen from the simulated deep Lorenz-96 data shown in Figure 2). As outlined in Appendix B, we use the following settings: $h_x = -1.90, h_y = 1$, and $\eta_L = .045$, while we follow (Chorin & Lu, 2015) in setting $F = 10, K = 18$, and $J = 20$. The variance term in (23) is set such that $\sigma_\eta^2 = .25$. An Euler solver is used to numerically solve the Lorenz-96 equations in (23) using a time step of $\delta = .10$. After a burn-in period, 510 periods of simulated data from the deep Lorenz-96 model are retained, with the final 75 periods held out and treated as out-of-sample realizations.

Only the large-scale process $z_{t,k}$ is treated as observed here; thus, both the large- and small-scale Lorenz-96 variables (i.e., x_k and $y_{j,k}$) are considered unobserved. Therefore, unlike the soil moisture application described below, both the input and output of the model are the same process (i.e., $z_{t,k}$). The input and output are separated by three periods here (i.e., the lead time is set to three periods), in order to make the problem slightly more difficult. The previously discussed embedding lag (i.e., τ in (4)) is set to the lead time (three periods) for the deep Lorenz-96 D-EESN implementations. The number of

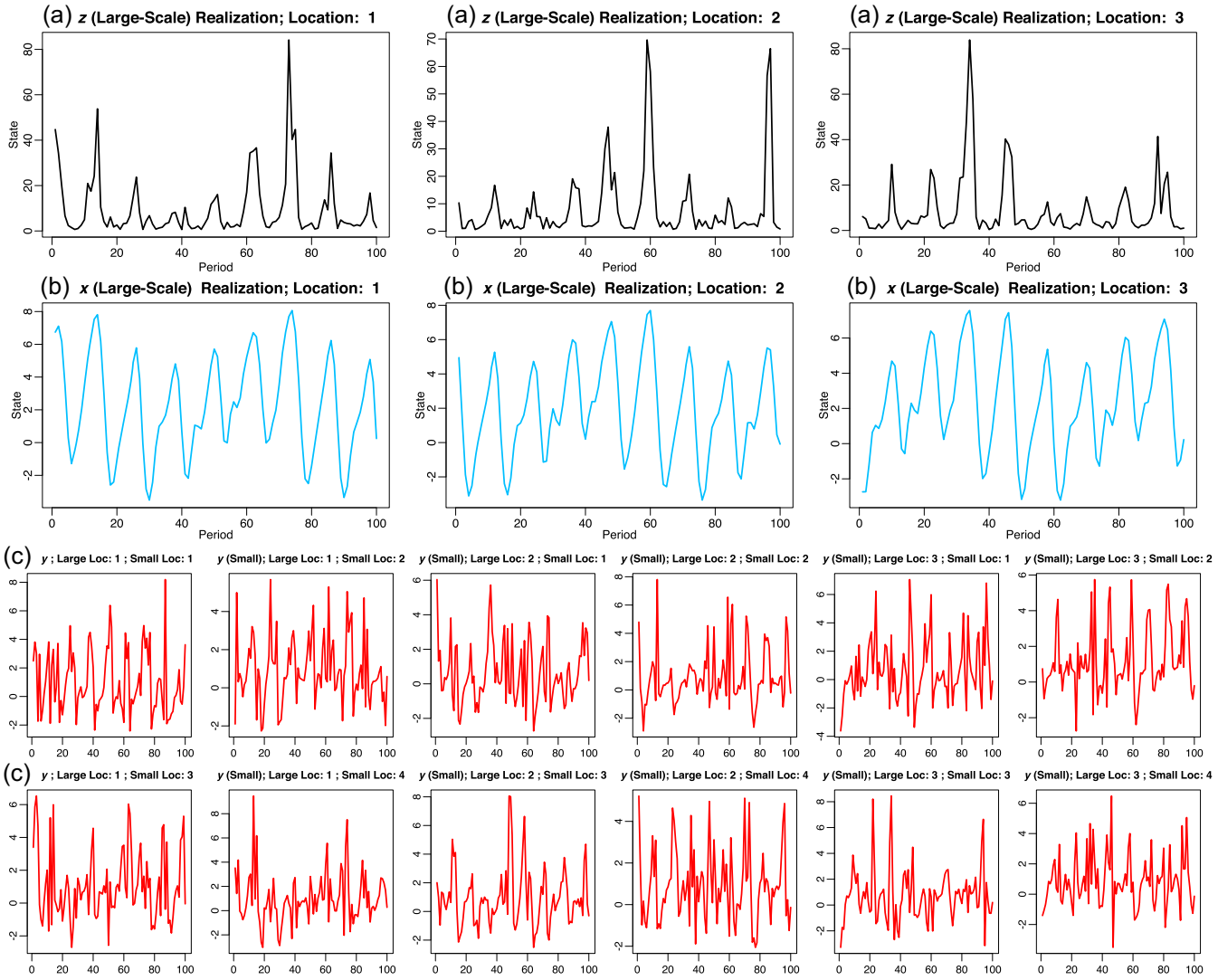

FIGURE 2 A simulation from the deep Lorenz-96 model defined in (23) for 100 time periods. The black, blue, and red plots follow in order the three equations defined in (23). (a) Realizations from the data model in (23) for three locations, where these locations are transformations of the slowly varying large-scale locations shown in the second row (i.e., (b)). (b) Realizations for three large-scale locations from the process model defined in (23). (c) Realizations corresponding to small-scale locations for each of the three large-scale locations displayed in the second row. Note that each column displays four (of the twenty) slowly varying small-scale locations associated with a particular large-scale location

embedding lags (i.e., m in (4)) is selected with the GA (using cross-validation) to be three. Finishing the specification of the data models in (8) and (16), $\mathbf{\Phi}$ is defined as $\mathbf{\Phi} \equiv \mathbf{I}$, where $\mathbf{I}$ is an 18×18 identity matrix, a log-Gaussian distribution is used for the unspecified distribution in (16), and the covariance matrix is set such that $\Sigma_z = \sigma_z^2 \mathbf{I}$, where $\sigma_z^2 = \sigma_\eta^2$.

The out-of-sample validation metrics in Table 2 show the seven-layer D-EESN models to be the best forecast model for the deep Lorenz-96 data. In particular, both of the seven-layer D-EESN models perform better in terms of MSPE and CRPS than the Q-EESN or linear model. Although not shown here, D-EESN models with 2–6 layers all performed better than the Q-EESN model and worse than the seven-layer D-EESN model in terms of MSPE, with the MSPE monotonically decreasing as layers were added. We found that, after seven-layers, any gain in forecast accuracy was very minimal in comparison to the extra computation required to keep adding layers.

This D-EESN model's ability to predict and quantify uncertainty is illustrated in Figure 3, which shows forecast summaries for the first six locations with the seven-layer D-EESN model. Despite the clear nonlinearity in the process, the model correctly forecasts much of the overall quasicyclic nature and intensity of the process, while producing uncertainty metrics that cover many of the true values. The Bayesian version of the seven-layer D-EESN model produces similar but worse MSPE and CRPS values compared to the non-Bayesian version. It is not surprising that the non-Bayesian version performs better in terms of MSPE as the hyperparameters for both models are tuned using the non-Bayesian version of

TABLE 2 Results for the deep Lorenz-96 simulation study in terms of mean squared prediction error (MSPE) and continuous ranked probability score (CRPS)

Model	MSPE	CRPS
Q-EESN	82.64	3,876.45
BQ-EESN	81.51	3,875.61
D-EESN (7 L)	**76.00**	**3,872.72**
BD-EESN (7 L)	79.08	3,873.87
Lin. DSTM	100.12	4,557.78

Note. *Q-EESN* refers to the quadratic EESN, *BQ-EESN* denotes the Bayesian version of the Q-EESN model, *D-EESN (7 L)* is the D-EESN model with seven layers, and *BD-EESN (7 L)* denotes the BD-EESN model with seven layers. Note that smaller is better for both MSPE and CRPS. EESN = ensemble echo state network; DSTM = dynamical spatio-temporal model.

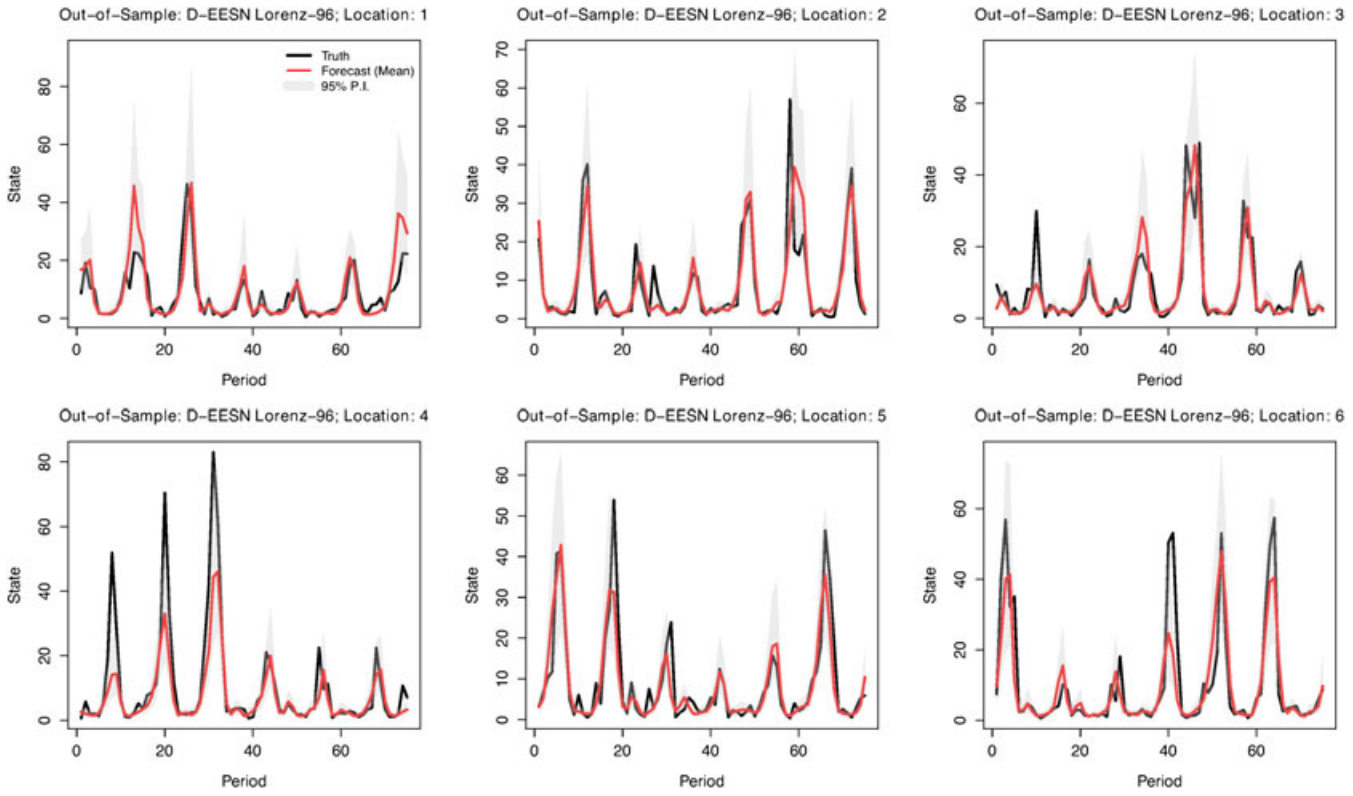

FIGURE 3 Out-of-sample forecast summary plots for six of the 18 large-scale locations from the deep Lorenz-96 simulation study using the seven-layer deep ensemble echo state network (D-EESN) model. The black line denotes the true simulated value in each plot (i.e., $z_{t,k}$ in (23)), whereas the red line denotes the forecasted mean. In a given plot, the shaded gray area represents the 95% pointwise prediction intervals (P.I.s) over all ensembles

the model. However, it is somewhat surprising that the CRPS is slightly worse for the Bayesian version (note, however, that this is not the case with the soil moisture example shown below).

4.3 | Midwest soil moisture long-lead forecasting application

The previously mentioned soil moisture data comes from the Climate Prediction Center's (CPC) high resolution monthly global soil moisture data set (Fan & Van den Dool, 2004; `https://iridl.ldeo.columbia.edu/SOURCES/.NOAA/.NCEP/.CPC/.GMSM/.w/`). As described in Smith, Reynolds, Peterson, and Lawrimore (2008), the driving inputs used to create this derived data consist of global

precipitation and temperature data along with an accompanying land model. Spatially, the data domain covers 35.75°N–48.75°N latitude and 101.75°W–80.25°W longitude at a resolution of 0.5° × 0.5°. The monthly data set begins in January 1948 and goes through December 2017. While the model is trained on monthly data from January 1948 through November 2011, only the May forecasted values from the out-of-sample period covering 2012–2017 are used to evaluate the model, given the importance of May soil moisture for planting corn in the U.S. corn belt. We follow the common practice in the atmospheric science literature of converting data into anomalies by subtracting the respective in-sample monthly means from the data.

Dimension reduction for the soil moisture data is carried out using EOFs (i.e., spatial-temporal principal component analysis; see, e.g., Cressie & Wikle, 2011, chapter 5). Therefore, Φ is obtained using the first 15 EOFs, which account for 80% of the variation in the soil moisture data (note that the model was not sensitive to small variations in this choice). The data model spatial covariance matrix Σ_z is calculated using the following formulation from equation 7.6 in Cressie and Wikle (2011): $\Sigma_z = \sum_{\ell_z=n_b+1}^{n_z} \tilde{\lambda}_{\ell_z} \Phi_{\ell_z} \Phi'_{\ell_z} + \tilde{c} \quad \mathbf{I}$, where $\tilde{\lambda}_{\ell_z}$ and Φ_{ℓ_z} are the remaining eigenvalues and eigenvectors, respectively, that are not used in the decomposition of $\mathbf{Z}_t$, and $\tilde{c}$ is a constant (set to 0.01). Because the data are converted into anomalies, a Gaussian distribution is used for the distribution in (16).

The monthly SST data comes from the extended reconstruction SST (ERSST) data set (Smith et al., 2008), `http://iridl.ldeo.columbia.edu/SOURCES/.NOAA/.NCDC/.ERSST`) and cover the same temporal period as the soil moisture data. The spatial domain for the SST data is given by 29°S–29°N latitude and 124°E–70°W longitude with a resolution of 2° × 2°, and covers much of the mid-Pacific ocean. Dimension reduction is once again carried out using EOFs by retaining the first five EOFs of the SST data set, which account for almost 72% of the variation in the data (the model was also not overly sensitive to this choice). The embedding lag for the input is set to the lead time of six periods, and the number of embedding lags is selected to be three using cross-validation with the GA. Convergence for the MCMC Gibbs algorithm sampler was further assessed using the Gelman–Rubin diagnostic (Gelman & Rubin, 1992) with four chains, which did not suggest any lack of convergence.

Table 3 shows the performance of various models for out-of-sample long-lead forecasting of May soil moisture. Both the Bayesian and non-Bayesian D-EESN models perform the best in terms of MSPE. These deep models also represent the largest improvement over the climatological forecast as illustrated by the SS percentages. Further, both D-EESN models also outperform the single-layered Q-EESN model, suggesting the soil moisture application benefits from a deep framework. Notably, the two and three-layer models appear to have similar MSPE values, indicating that two layers is likely sufficient, although the CRPS is slightly better for the three-layer BD-EESN model. While the Bayesian and non-Bayesian models perform similarly in terms of forecast accuracy, the Bayesian D-EESN models perform much better in terms of CRPS than the non-Bayesian versions. In particular, the Bayesian version produces CRPS values that are almost 16% lower than the non-Bayesian version.

Figure 4 shows the posterior predictive means and standard deviations for 2014 and 2015 over the prediction domain as given by the two-layer BD-EESN model. The model appears to mostly pick up the correct pattern of the soil moisture

TABLE 3 Validation results for the long-lead soil moisture forecasting application using mean squared prediction error (MSPE), continuous ranked probability score (CRPS), and percentage of skill score values greater than zero (i.e., % SS>0)

Model	MSPE	CRPS	% SS>0
Q-EESN	3,507.84	239.16	50.60%
BQ-EESN	3,463.28	196.59	55.20%
D-EESN (2 L)	3,303.64	229.96	60.00%
BD-EESN (2 L)	**3,296.39**	190.45	58.24%
D-EESN (3 L)	3,307.51	224.68	60.32
BD-EESN (3 L)	3,299.04	**189.80**	**63.54%**
Lin. DSTM	3,509.63	198.09	50.00%
Climatological	3,642.54	-	-

Note. The prefix "B" denotes a Bayesian version of the model, and "L" denotes the number of layers in a given model. Q-EESN = quadratic ensemble echo state network; BQ-EESN = Bayesian quadratic ensemble echo state network; D-EESN = deep ensemble echo state network; BD-EESN = Bayesian deep ensemble echo state network; DSTM = dynamical spatio-temporal model.

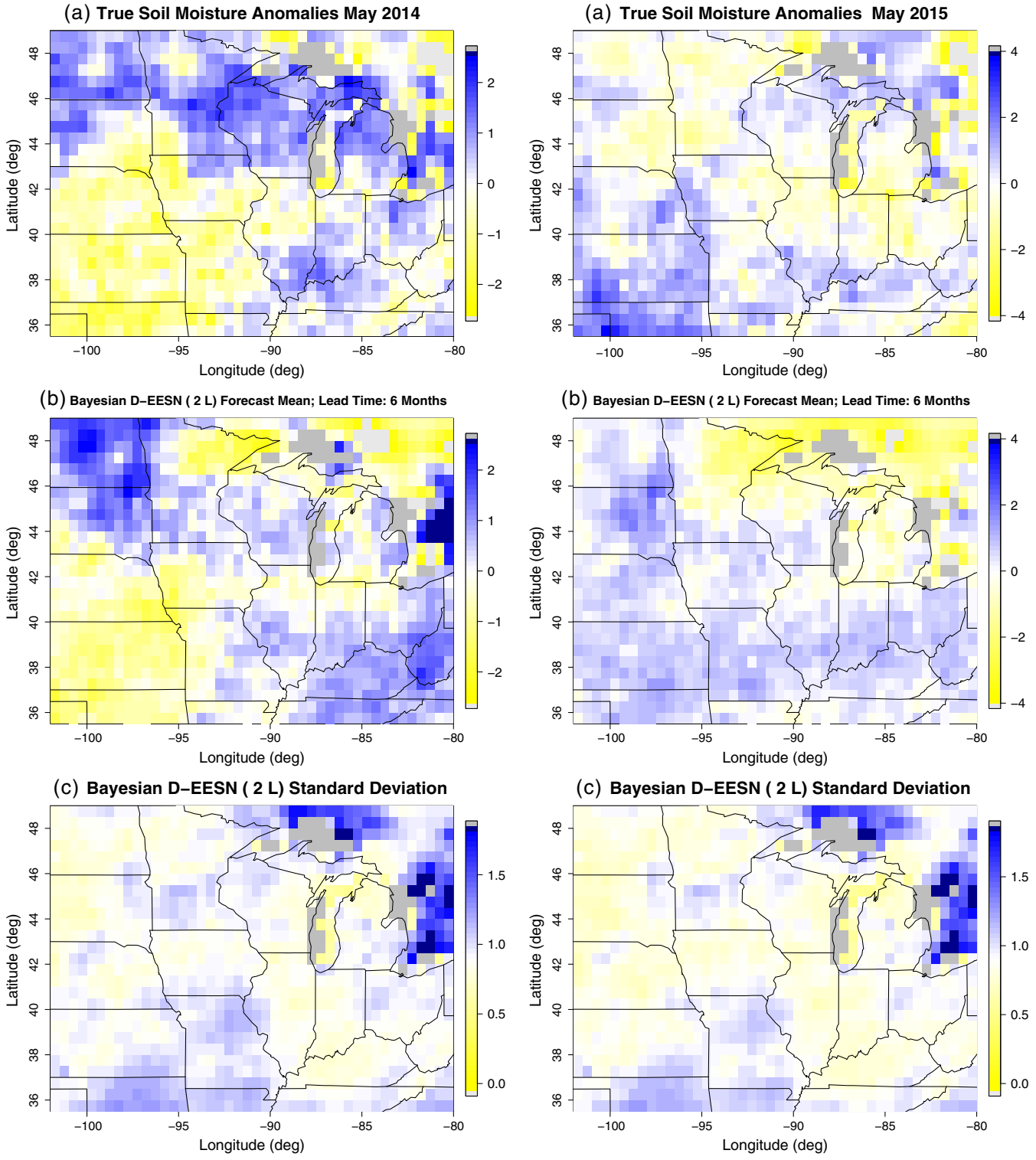

FIGURE 4 Posterior summaries for the soil moisture application in May 2014 and 2015 using the two-layer Bayesian deep ensemble echo state network model. (a) True soil moisture values for each spatial location. (b) Posterior predictive mean values for each spatial location. (c) Posterior predictive standard deviations for each spatial location. Note that, for the sake of visualization, each plot has been standardized by their respective means and standard deviations. In addition, note that the scale bars are different for the 2014 and 2015 anomalies to improve forecast contrast. We have removed extreme outliers for the sake of visualization (indicated by gray and black grid squares). D-EESN = deep ensemble echo state network

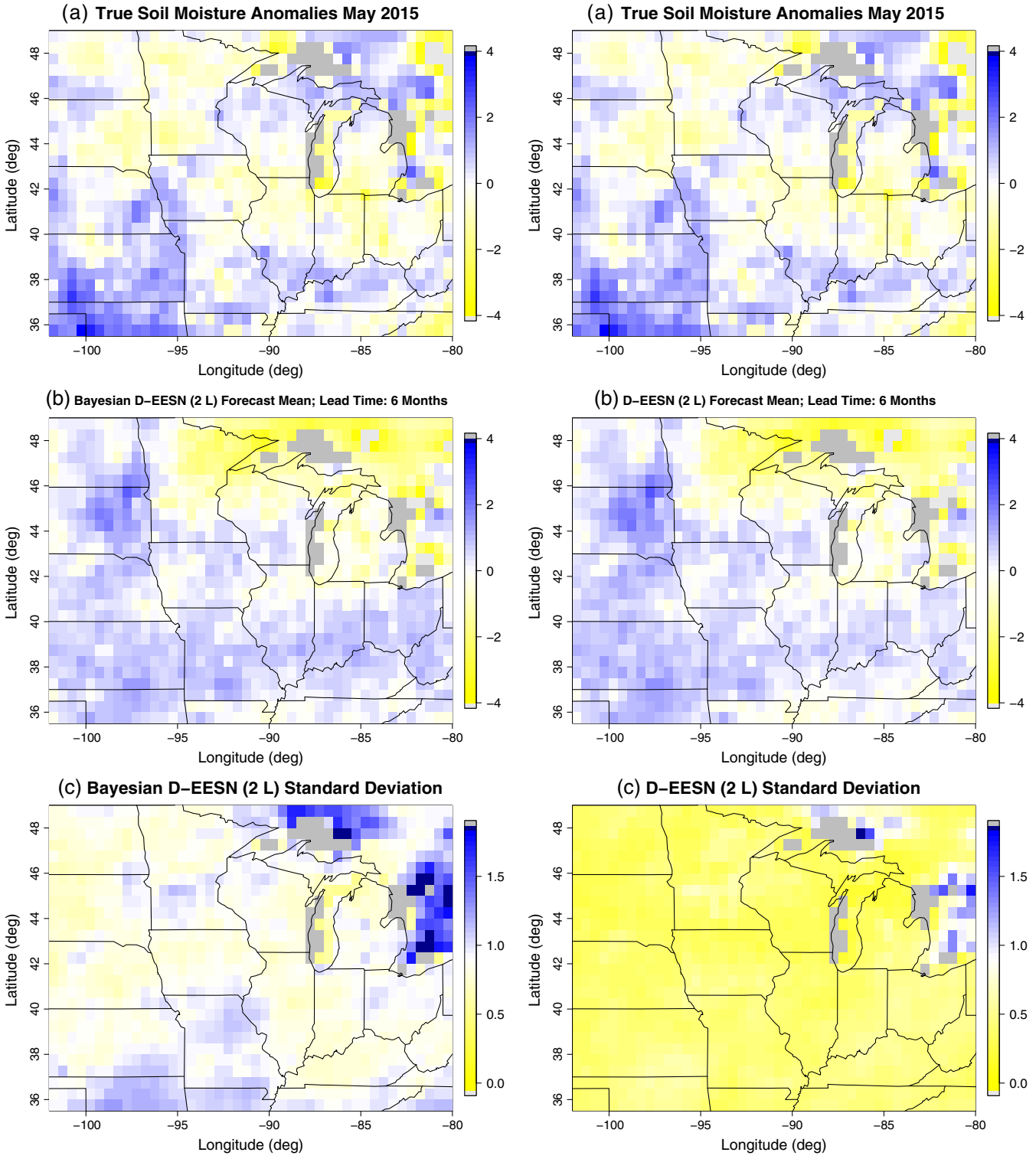

FIGURE 5 May 2015 forecast summaries for both the Bayesian and non-Bayesian two-layer deep ensemble echo state network (D-EESN) model. (a) True soil moisture values for each spatial location. (b) Forecasted mean values with a given forecasting method for each spatial location. (c) Forecast standard deviations with a given forecasting method for each spatial location. Note that, for the sake of visualization, each plot has been standardized by their respective means and standard deviations. We have removed extreme outliers for the sake of visualization (indicated by gray and black grid squares)

anomaly signal in the western and mid-central part of the spatial domain for both years, while struggling more with the upper Midwest states in 2014. Regarding the critical corn producing Midwest U.S., the model suggests an overall large amount of uncertainty in Missouri, especially in 2014, whereas Iowa appears to have a moderate to low amount of uncertainty for both years. As previously noted, the BD-EESN produces considerably better uncertainty metrics for the soil moisture data than the non-Bayesian D-EESN. The relative difference between these two predictive uncertainties can be seen in Figure 5, where both versions of the two-layer D-EESN model are plotted for 2015. Although the forecast means are very similar in Figure 5, the non-Bayesian D-EESN produces much smaller standard deviations across the spatial domain compared with the Bayesian version. Considering the inherent difficulty in predicting soil moisture six months into the future, the standard deviations for the non-Bayesian model appear unrealistically low. This point is confirmed by the Bayesian model producing a considerably lower CRPS value than the non-Bayesian version.

Next, the previously defined SS in (22) is shown in Figure 6 for the two-layer BD-EESN model, where a climatological forecast is used as the reference model. Values of SS greater than zero represent locations where the BD-EESN model improved upon the climatological forecast, whereas values less than zero indicate locations where the model did worse than the climatological forecast. Overall, the BD-EESN model outperforms the climatological forecast in the central western part of the domain and performs worse in the east central and northern part of the domain. In particular, the model does worse in these regions by predicting too little soil moisture in the north central part of the domain, relative to the truth, and overpredicts the amount of soil moisture in the east central part of the domain. Critically, across much of the agriculturally important corn belt, including much of Iowa, the BD-EESN model improves upon the climatological forecast.

Finally, it is important to note that the multiple levels in the deep models allow for different time scales in the predictors—this is the advantage of the deep architecture. How do we know that deep structure is giving multitime-scale predictors? Figure 7 shows the most frequently chosen predictors for the first EOF coefficient in the soil moisture example ($\alpha_t(1)$). These predictors exhibit different time scales. Although this shows that different time scales are important for the prediction, it is not so clear how to interpret these predictors given the complex nonlinear transformations that generated them in the deep ESN structure. This lack of interpretability is a fundamental problem with many deep models in the machine learning context as well, and is an active area of investigation in statistics and machine learning.

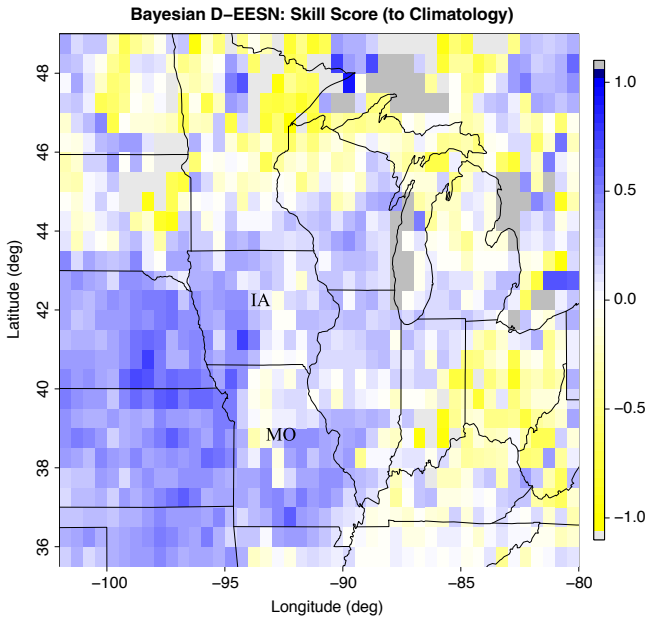

FIGURE 6 Skill score (SS) plot for the soil moisture application with the Bayesian two-layer deep ensemble echo state network (D-EESN) model. SS is calculated using (22) from above, where the *reference model* used here is a climatological forecast. For each forecasting method, the MSPE is calculated by averaging over the out-of-sample periods. Values of SS greater than zero indicate an improvement of the two-layer BD-EESN model over the climatological forecast, whereas values less than zero indicate locations where the two-layer BD-EESN model performed worse than climatology. We have removed extreme outliers for the sake of visualization (indicated by gray grid squares). Note that the label IA denotes the state of Iowa and the label MO denotes the state of Missouri

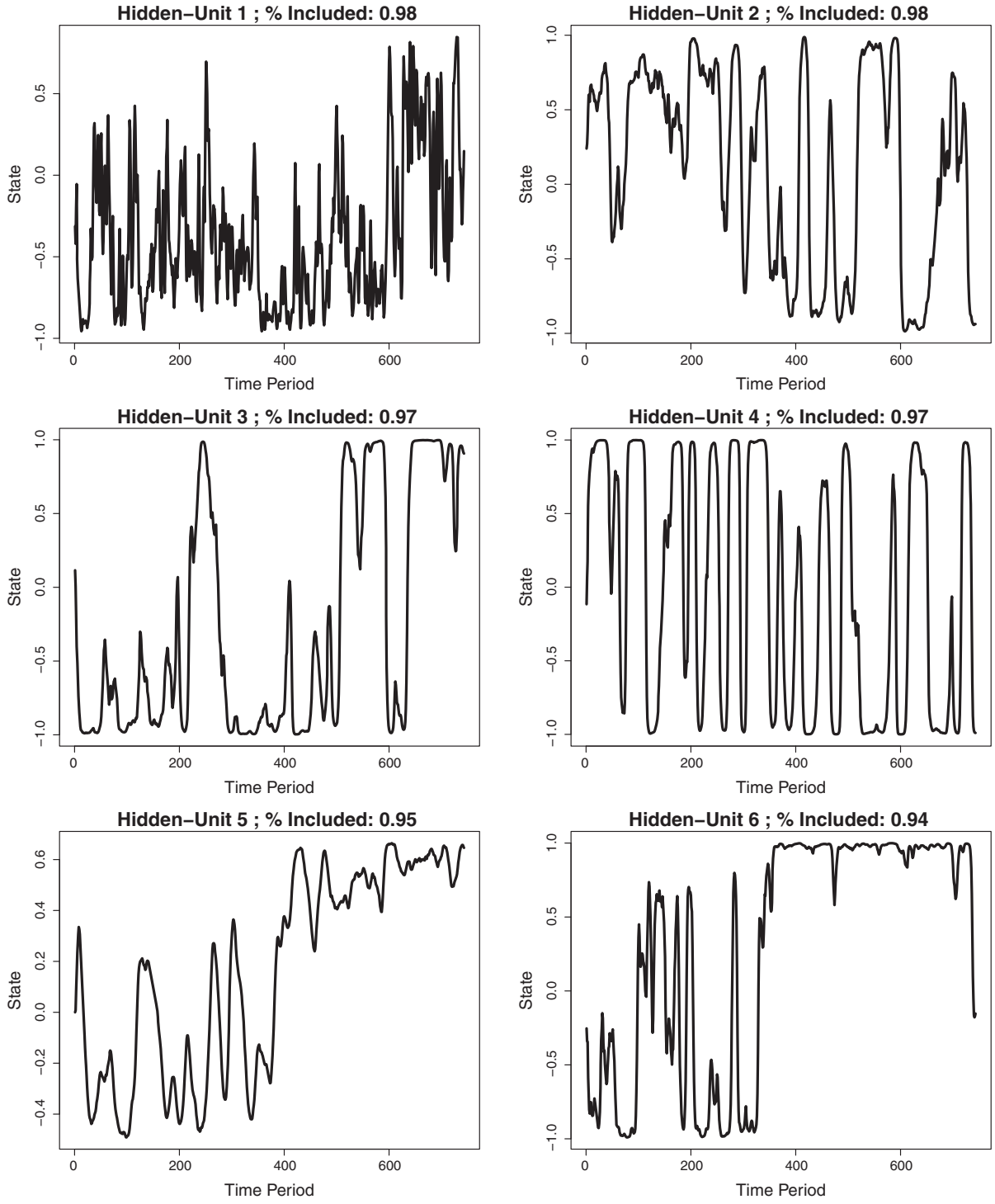

FIGURE 7 Time series of the six most chosen hidden unit predictors used to predict the first soil moisture empirical orthogonal function coefficient ($\alpha_t(1)$), as chosen via stochastic search variable selection in the Bayesian deep ensemble echo state network model. The percentage of Markov chain Monte Carlo iterations that include each predictor is given shown above each series

5 | DISCUSSION

Many spatio-temporal processes are best considered to be nonlinear with multiple interacting space and time scales. Parametric statistical models can account for realistic (science-based) nonlinear interactions through nonlinear latent processes and dependence structure on parameters, but these models are very expensive to implement in a statistically rigorous parametric framework. Similarly, deep machine learning models are powerful but require a large amount of

training data, are computationally expensive, and typically do not provide formal uncertainty quantification. ESNs provide a viable parsimonious alternative with simple modifications but do not naturally accommodate multiple time scales or uncertainty quantification. When extended in a multilevel deep representation, the ESN model can accommodate multiple time scales. However, despite the large amount of research into deep models, uncertainty quantification has rarely been considered in their application. Given the demonstrated predictive ability of these methods, having the ability to quantify uncertainty with deep models is very powerful and widely applicable. Through the use of an ensemble framework and a fully Bayesian implementation, the deep ESN models for spatio-temporal processes presented here, D-EESN and BD-EESN, respectively, allow for uncertainty quantification. The non-Gaussian multiscale Lorenz-96 simulation example showed that these deep ESN models improved upon the traditional ESN model in terms of MSPE, while also producing very robust estimates of the forecast uncertainty. Furthermore, the Bayesian version of the D-EESN model provides a formal framework in which many data types can be considered and multiple levels of uncertainties can be accounted for. The results for the BD-EESN with the soil moisture data illustrated this point by considerably improving upon the uncertainty metrics produced by the D-EESN model. We were also able to identify the spatial locations where the deep models improved upon simpler forecast methods, thus giving model developers and resource managers potentially useful information.

Regarding the long-lead soil moisture forecasting application, we note that it is more common to treat such difficult forecasting problems as categorial instead of continuous. That is, treating the response as categorical by relabeling continuous values with qualitative values (e.g., below average, average, above average). Unlike the RNN literature, most of the ESN literature has focused on continuous responses. Given the flexibility of the BD-EESN, a categorical model formulation can be developed and is the subject of future research. In addition, we note that the soil moisture forecasts may also benefit from using other climate indexes or more local variables (such as precipitation) as inputs into the model.

In conclusion, the deep ESN methodology presented here can be thought of as a regularized spatio-temporal regression presented similar to a generalized additive model. However, the inputs (predictors) are stochastically and dynamically transformed here (many times). Although the spatio-temporal regression model is not dynamic, the transformations are dynamic through the ESN structure. Further, multiple levels of transformation allow for different time and spatial scales in the predictor variables to affect the response, and the multiple copies (replications) of the transformation allow for reproducibility (stability) and provide a dimension expansion. Note that these replications of the deep ESN are given equal weight in (17), but this need not be the case in general. It is important to re-emphasize that this model is very easy to implement and relatively efficient (compared with deep parametric statistical models and deep machine learning models) due to the reservoir approach in the ESN and simple regularization—it can be computed fairly quickly on a laptop.

As discussed above, there are many more possible choices for the dimension reduction function used with the hidden units from the D-EESN model. One potential choice that has been unexplored in the literature is a *convolution operator*. Deep image classification methods have shown convolution operators to be extremely powerful for slowly learning general (spatial) features (Krizhevsky et al., 2012). It is possible that the deep RNN framework could also benefit from such tools, especially in situations where the input has explicit spatial structure. More general challenges concern interpretation and inference of model components, the incorporation of known scientific knowledge, and the consideration of multivariate/multitype outputs. These are topics of research for deep models in machine learning as well.

ACKNOWLEDGEMENTS

This work was partially supported by the U.S. National Science Foundation (NSF) and the U.S. Census Bureau under NSF Grant SES-1132031, funded through the NSF-Census Research Network (NCRN) program, and NSF Award DMS-1811745.

ORCID

Patrick L. McDermott https://orcid.org/0000-0002-0734-2770

REFERENCES

Antonelo, E. A., Camponogara, E., & Foss, B. (2017). Echo State Networks for data-driven downhole pressure estimation in gas-lift oil wells. *Neural Networks*, *85*, 106–117.

Baran, S., & Lerch, S. (2015). Log-normal distribution based Ensemble Model Output Statistics models for probabilistic wind-speed forecasting. *Quarterly Journal of the Royal Meteorological Society*, *141*(691), 2289–2299.

Barnston, A. G., Glantz, M. H., & He, Y. (1999). Predictive skill of statistical and dynamical climate models in SST forecasts during the 1997–98 El Niño episode and the 1998 La Niña onset. *Bulletin of the American Meteorological Society*, *80*(2), 217–243.

Belkin, M., & Niyogi, P. (2001). Laplacian Eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing Systems 14 (NIPS 2001)*.

Berliner, L. M., Wikle, C. K., & Cressie, N. (2000). Long-lead prediction of pacific SSTs via Bayesian dynamic modeling. *Journal of Climate*, *13*(22), 3953–3968.

Blackmer, A. M., Pottker, D., Cerrato, M. E., & Webb, J. (1989). Correlations between soil nitrate concentrations in late spring and corn yields in Iowa. *Journal of Production Agriculture*, *2*(2), 103–109.

Carleton, A. M., Arnold, D. L., Travis, D. J., Curran, S., & Adegoke, J. O. (2008). Synoptic circulation and land surface influences on convection in the Midwest U.S. "Corn Belt" during the summers of 1999 and 2000. Part I: Composite synoptic environments. *Journal of Climate*, *21*(14), 3389–3415.

Chatzis, S. P. (2015). Sparse Bayesian recurrent neural networks. In *Lecture Notes in Computer Science. Vol. 9285. Machine Learning and Knowledge Discovery in Databases European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part II* (pp. 359–372). Cham, Switzerland: Springer International Publishing Switzerland.

Chorin, A. J., & Lu, F. (2015). Discrete approach to stochastic parametrization and dimension reduction in nonlinear dynamics. *Proceedings of the National Academy of Sciences of the United States of America*, *112*(32), 9804–9809.

Cressie, N., & Wikle, C. K. (2011). *Statistics for spatio-temporal data*. Hoboken, NJ: John Wiley & Sons.

Dixon, M. F., Polson, N. G., & Sokolov, V. O. (2017). Deep learning for spatiotemporal modeling: Dynamic traffic flows and high frequency trading. arXiv preprint arXiv:1705.09851.

Drosdowsky, W. (1994). Analog (nonlinear) forecasts of the Southern Oscillation index time series. *Weather and Forecasting*, *9*(1), 78–84.

Fan, Y., & Van den Dool, H. (2004). Climate Prediction Center global monthly soil moisture data set at 0.5° resolution for 1948 to present. *Journal of Geophysical Research: Atmospheres*, *109*(D10), 109.

Fischer, E. M., Seneviratne, S. I., Vidale, P. L., Lüthi, D., & Schär, C. (2007). Soil moisture–atmosphere interactions during the 2003 European summer heat wave. *Journal of Climate*, *20*(20), 5081–5099.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *Springer Series in Statistics. The elements of statistical learning: Data mining, inference, and prediction*. New York, NY: Springer Science & Business Media.

Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, *7*(4), 457–472.

Genest, C., & Rémillard, B. (2008). Validity of the parametric bootstrap for goodness-of-fit testing in semiparametric models. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, *44*(6), 1096–1127.

George, E. I., & McCulloch, R. E. (1997). Approaches for Bayesian variable selection. *Statistica Sinica*, *7*(2), 339–373.

Gladish, D. W., & Wikle, C. K. (2014). Physically motivated scale interaction parameterization in reduced rank quadratic nonlinear dynamic spatio-temporal models. *Environmetrics*, *25*(4), 230–244.

Gneiting, T., & Katzfuss, M. (2014). Probabilistic forecasting. *Annual Review of Statistics and Its Application*, *1*, 125–151.

Gneiting, T., Raftery, A. E., Westveld, A. H., III, & Goldman, T. (2005). Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. *Monthly Weather Review*, *133*(5), 1098–1118.

Grooms, I., & Lee, Y. (2015). A framework for variational data assimilation with superparameterization. *Nonlinear Processes in Geophysics*, *22*(5).

Hooten, M. B., & Wikle, C. K. (2010). Statistical agent-based models for discrete spatio-temporal systems. *Journal of the American Statistical Association*, *105*(489), 236–248.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, *79*, 2554–2558.

Jaeger, H. (2001). *The "echo" state approach to analysing and training recurrent neural networks—With an erratum note* (GMD Report 148). Bonn, Germany: German National Research Center for Information Technology.

Jaeger, H. (2007). *Discovering multiscale dynamical features with hierarchical echo state networks* (Report No. 10). Bremen, Germany: Jacobs University Bremen.

Jan van Oldenborgh, G., Balmaseda, M. A., Ferranti, L., Stockdale, T. N., & Anderson, D. L. (2005). Did the ECMWF seasonal forecast model outperform statistical ENSO forecast models over the last 15 years? *Journal of Climate*, *18*(16), 3240–3249.

Knaff, J. A., & Landsea, C. W. (1997). An El Niño–Southern Oscillation climatology and persistence (CLIPER) forecasting scheme. *Weather and Forecasting*, *12*(3), 633–652.

Kondrashov, D., Kravtsov, S., Robertson, A. W., & Ghil, M. (2005). A hierarchy of data-based ENSO models. *Journal of Climate*, *18*(21), 4425–4444.

Kravtsov, S., Kondrashov, D., & Ghil, M. (2005). Multilevel regression modeling of nonlinear processes: Derivation and applications to climatic variability. *Journal of Climate*, *18*(21), 4404–4424.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, *25*, 1097–1105.

Li, D., Han, M., & Wang, J. (2012). Chaotic time series prediction based on a novel robust echo state network. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(5), 787–799.

Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, *20*(2), 130–141.

Lorenz, E. N. (1996). *Predictability: A problem partly solved*. Paper presented at the Seminar on Predictability, Reading, UK.

Lukoševičius, M. (2012). A practical guide to applying echo state networks. In *Neural networks: Tricks of the trade* (pp. 659–686). Berlin, Germany: Springer.

Lukoševičius, M., & Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, *3*(3), 127–149.

Ma, Q., Shen, L., & Cottrell, G. W. (2017). Deep-ESN: A multiple projection-encoding hierarchical reservoir computing framework. arXiv preprint arXiv:1711.05255.

McDermott, P. L., & Wikle, C. K. (2016). A model-based approach for analog spatio-temporal dynamic forecasting. *Environmetrics*, *27*(2), 70–82.

McDermott, P. L., & Wikle, C. K. (2017a). Bayesian recurrent neural network models for forecasting and quantifying uncertainty in spatial-temporal data. arXiv preprint arXiv:1711.00636.

McDermott, P. L., & Wikle, C. K. (2017b). An ensemble quadratic echo state network for non-linear spatio-temporal forecasting. *STAT*, *6*, 315–330.

Penland, C., & Magorian, T. (1993). Prediction of Niño 3 sea surface temperatures using linear inverse modeling. *Journal of Climate*, *6*(6), 1067–1076.

Philander, S. (1990). *El Niño, La Niña, and the Southern oscillation*. San Diego, CA: Academic Press.

Richardson, R. A. (2017). Sparsity in nonlinear dynamic spatiotemporal models using implied advection. *Environmentrics*, *28*(6), e2456.

Sheffield, J., Goteti, G., Wen, F., & Wood, E. F. (2004). A simulated soil moisture based drought analysis for the United States. *Journal of Geophysical Research: Atmospheres*, *109*(D24).

Sheng, C., Zhao, J., Wang, W., & Leung, H. (2013). Prediction intervals for a noisy nonlinear time series based on a bootstrapping reservoir computing network ensemble. *IEEE Transactions on Neural Networks and Learning Systems*, *24*(7), 1036–1048.

Sivanandam, S. N., & Deepa, S. N. (2007). *Introduction to genetic algorithms*. Berlin, Germany: Springer Science & Business Media.

Smith, T. M., Reynolds, R. W., Peterson, T. C., & Lawrimore, J. (2008). Improvements to NOAA's historical merged land–ocean surface temperature analysis (1880–2006). *Journal of Climate*, *21*(10), 2283–2296.

Stern, H., & Davidson, N. E. (2015). Trends in the skill of weather prediction at lead times of 1–14 days. *Quarterly Journal of the Royal Meteorological Society*, *141*(692), 2726–2736.

Takens, F. (1981). Detecting strange attractors in turbulence. In *Lecture Notes in Mathematics. Vol. 898. Dynamical systems and turbulence: Proceedings of a Symposium Held at the University of Warwick 1979/80* (pp. 366–381). Berlin, Germany: Springer-Verlag Berlin Heidelberg. Retrieved from https://doi.org/10.1007/BFb0091924

Tang, B., Hsieh, W. W., Monahan, A. H., & Tangang, F. T. (2000). Skill comparisons between neural networks and canonical correlation analysis in predicting the equatorial Pacific sea surface temperatures. *Journal of Climate*, *13*(1), 287–293.

Timmermann, A., Voss, H. U., & Pasmanter, R. (2001). Empirical dynamical system modeling of ENSO using nonlinear inverse techniques. *Journal of Physical Oceanography*, *31*(6), 1579–1598.

Triefenbach, F., Jalalvand, A., Demuynck, K., & Martens, J. P. (2013). Acoustic modeling with hierarchical reservoirs. *IEEE Transactions on Audio, Speech, and Language Processing*, *21*(11), 2439–2450.

Van den Dool, H., Huang, J., & Fan, Y. (2003). Performance and analysis of the constructed analogue method applied to U.S. soil moisture over 1981–2001. *Journal of Geophysical Research: Atmospheres*, *108*(D16).

Wikle, C. K. (2015). Modern perspectives on statistics for spatio-temporal data. *Wiley Interdisciplinary Reviews: Computational Statistics*, *7*(1), 86–98. https://doi.org/10.1002/wics.1341

Wikle, C. K., & Hooten, M. B. (2010). A general science-based framework for dynamical spatio-temporal models. *Test*, *19*(3), 417–451.

Wilks, D. S. (2001). A skill score based on economic value for probability forecasts. *Meteorological Applications*, *8*(2), 209–219.

Wilks, D. S. (2005). Effects of stochastic parametrizations in the Lorenz '96 system. *Quarterly Journal of the Royal Meteorological Society*, *131*(606), 389–407.

Zhao, Z., & Giannakis, D. (2016). Analog forecasting with dynamics-adapted kernels. *Nonlinearity*, *29*(9).

How to cite this article: McDermott PL, Wikle CK. Deep echo state networks with uncertainty quantification for spatio-temporal forecasting. *Environmetrics*. 2019;30:e2553. https://doi.org/10.1002/env.2553

APPENDIX A

D-EESN HYPERPARAMETERS

All of the hyperparameters for the D-EESN, along with their respective state (i.e., fixed or optimized), a description, and potential values are listed in Table A1. The fixed hyperparameters did not seem to be sensitive to their particular value. Similar results were found with regards to this sensitivity in McDermott and Wikle (2017b). More concretely, all of the sparseness parameters for the D-EESN model (i.e., $\pi_{w_1}, \ldots, \pi_{w_L}$ and $\pi_{u_1}, \ldots, \pi_{u_L}$) were set to small values (i.e., 0.10). Due

TABLE A1 Detailed descriptions of all of the hyperparameters in the deep ensemble echo state network (D-EESN) model

Hyperparameter	State	Description	Search space/fixed value
m	Optimized	# of embeddings	$\{0, 1, \ldots, 5\}$
ν_ℓ	Optimized	Spectral radius	$[0, 1]$
$n_{\tilde{h},\ell}$	Optimized	# of hidden units for reduced layer ℓ	$\{6, 7, \ldots, 20\}$
$n_{h,1}$	Optimized	# of hidden units for layer 1	$\{25, 26, \ldots, 75\}$
$n_{h,\ell}$ for $\ell \geq 2$	Fixed	# of hidden units	84
r_v	Optimized	Ridge parameter	$[.0001, .01]$
$\pi_{w_1}, \ldots, \pi_{w_L}$	Fixed	Sparseness for $\mathbf{W}_\ell$	0.10
$\pi_{u_1}, \ldots, \pi_{u_L}$	Fixed	Sparseness for $\mathbf{U}_\ell$	0.10
$a_{w_1}, \ldots, a_{w_L}$	Fixed	Uniform bounds for $\mathbf{W}_\ell$	0.10
$a_{u_1}, \ldots, a_{u_L}$	Fixed	Uniform bounds for $\mathbf{U}_\ell$	0.10

Note. In particular, for each hyperparameter (or set of hyperparameter), the column "State" denotes if the hyperparameter is optimized or assumed fixed. The column "Description" contains a description of the respective hyperparameter. The last column labeled "Search space/fixed value" contains the value at which fixed hyperparameters are fixed at and the search space for hyperparameters that are optimized.

to the number of hidden units, it is common in the ESN literature to set these values to small values to prevent overfitting and, thus, encourage spareness (Lukoševičius, 2012). Moreover, a similar assumption of fixing sparseness parameters is also commonly made in the Bayesian variable selection literature when using SSVS priors, which share a similar form as the mixture distributions used in the D-EESN model.

Further, the bounds for the uniform distributions in the D-EESN model (i.e., $a_{w_1}, \ldots, a_{w_L}$ and $a_{u_1}, \ldots, a_{u_L}$) are also set at small values. Because the matrix $\mathbf{W}_\ell$ is always rescaled by the spectral radius ν_ℓ, the bounds of the uniform distribution for $\mathbf{W}_\ell$ have very little impact (see Lukoševičius, 2012). Similarly, we found that changing the bounds for the parameter matrix $\mathbf{U}_\ell$ had little, if any, impact on the underlying forecasts. Finally, the number of hidden units for $n_{h,\ell}$ when $\ell \geq 2$ is also fixed because the dimension for these hidden units is always reduced through the dimension reduction transformation. A similar assumption is made in Ma et al. (2017).

APPENDIX B

DEEP LORENZ-96 MODEL HYPERPARAMETERS

Descriptions of the hyperparameters in the deep Lorenz-96 model, along with their specific fixed values, can be found in Table B1. Here, we follow the common practice in the literature of fixing many of the hyperparameters in the Lorenz-96, when using this model in a simulation setting. The hyperparameters F, K, h_y, and J are all set at the same values used in

TABLE B1 Detailed descriptions of all of the hyperparameters in the deep Lorenz-96 model

Hyperparameter	State	Description	Fixed value
c	Fixed	Scaling for the log-Gaussian distribution	2
F	Fixed	Forcing parameter	10
ϵ_L	Fixed	Time-scale separation parameter	0.025
h_x	Fixed	Interaction parameter	−1.90
h_y	Fixed	Interaction parameter	1
K	Fixed	# of y variables per x variable	18
J	Fixed	# of x variables	20
σ_η^2	Fixed	Log-Gaussian variance	.25

Note. In particular, for each hyperparameter (or set of hyperparameters), the column "State" denotes if the hyperparameter is optimized or assumed fixed. The column "Description" contains a description of the respective hyperparameter. The last column labeled "fixed value" contains the value at which fixed hyperparameters are set.

Chorin and Lu (2015). Additionally, ϵ_L and h_x are set so as to facilitate temporal multiscale behavior. Because the stated purpose of the developed methodology was to forecast multiscale systems, these parameter settings allow us to investigate the effectiveness of the proposed model. The log-Gaussian scaling parameter (i.e., c) and variance (i.e., σ_η^2) were selected to regulate the amount of nonlinearity in the model, along with ensuring that the simulations produced realistic data.