Data-driven deep learning emulators for geophysical forecasting*

Varuni Katti Sastry[1], Romit Maulik[2], Vishwas Rao[2], Bethany Lusch[2], S. Ashwin Renganathan[2], and Rao Kotamarthi[2,3]

[1] Freelance Researcher
[2] Argonne National Laboratory, Lemont, IL, USA
[3] University of Chicago, Chicago, IL, USA
{varunisastry@gmail.com},{rmaulik, vhebbur, blusch, srenganathan, vkotamarthi}@anl.gov

Abstract. We perform a comparative study of different supervised machine learning time-series methods for short-term and long-term temperature forecasts on a real world dataset for the daily maximum temperature over North America given by DayMET. DayMET showcases a stochastic and high-dimensional spatio-temporal structure and is available at exceptionally fine resolution (a 1 km grid). We apply projection-based reduced order modeling to compress this high dimensional data, while preserving its spatio-temporal structure. We use variants of time-series specific neural network models on this reduced representation to perform multi-step weather predictions. We also use a Gaussian-process based error correction model to improve the forecasts from the neural network models. From our study, we learn that the recurrent neural network based techniques can accurately perform both short-term as well as long-term forecasts, with minimal computational cost as compared to the convolution based techniques. We see that the simple kernel based Gaussian-processes can also predict the neural network model errors, which can then be used to improve the long term forecasts.

Keywords: Data-driven forecasting · Geophysical Systems · Deep Learning · LSTM · Gaussian Process Regression

1 Introduction

Forecasting the maximum temperature is a crucial capability for several applications relevant to agriculture, energy, industry, tourism and the environment. Improved accuracy in short and long-term forecasting of the air temperature has significant implications for cost-effective energy policy, infrastructure development, and downstream economic consequences [1,2]. Existing state of the art temperature forecasting methods rely on solving large scale partial differential equations (PDE), which generally requires the utilization of large computing resources and are therefore limited by access and considerations of energy-efficiency.

* This material was based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research (ASCR) under Contract DE-AC02-06CH11347.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_35

Machine learning methods promise to provide forecasts whose accuracy is comparable to the traditional methods at a much lower computational cost. This also allows for the possibility of using large ensemble-based forecasts that provide confidence intervals, which was hitherto considered expensive and impractical due to the large computational costs associated with the PDE-based methods. For these reasons, there has been a great degree of interest in building machine learning 'emulators' or 'surrogate models' for various geophysical data sets. There have been several studies on the use of machine learning for accelerating geophysical forecasts in recent times. Several rely on using machine learning methods to devise parameterizations for processes that contribute a significant cost to the numerical simulation of the weather and climate [3–7]. Other studies have looked at complete system emulators (i.e., forecasting from data alone) with a view to forecast without any use of and consequent limitations of equation based methods [8–13]. Other studies have looked at utilizing historical information for forecasting specific processes using data from the process alone [14–16]. Opportunities and perspectives for the use of data-driven methods for the geosciences may be found in [17, 18]. In this paper, we introduce a purely data-driven method for forecasting the maximum air temperature over the North American continent. In addition, we also provide an interpretable extension for improving the accuracy of these models using probabilistic machine learning. We achieve this by obtaining a low-dimensional affine subspace of the temperature on which a reduced system is evolved. Both dimensionality reduction and system evolution are performed using data-driven techniques alone with the former requiring a proper-orthogonal decomposition (POD) and the latter using a variety of time-series emulation methods. Among the methods investigated in this study, we include the long short-term memory (LSTM) neural network [19] which has previously been used for surrogate modeling of various geophysical and engineering applications [15, 20, 21]. In addition, we assess the viability of competing time-series forecast methods such as the bidirectional LSTM, previously deployed for a shallow-water equation surrogate [22]. In contrast to previous literature, we assess, for the first time, novel methods such as the sequence-to-sequence encoder-decoder model and the temporal convolutional network for geophysical emulation. These approaches forecast the evolution of POD-coefficients in the truncated POD space. Previously, extensive studies of this method with the LSTM have revealed issues related to stability when using observation data sets due to a relatively low signal to noise ratio. Therefore, we also introduce a error-correction module based on Gaussian process regression (GPR) that extends the viability of these compressed emulation methods and thereby obtain a more accurate forecast for a longer time into the future. Our experiments ascertain that the use of a probabilistic model to learn the bias in the neural network further enhances fidelity of forecasts in comparison to the standard, deterministic, neural network based approach.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_35

2 Proper orthogonal decomposition

Projection-based reduced order models (ROMs) can effectively compress a high dimensional model while still preserving its spatio-temporal structure. The dimensionality reduction is performed through the projection step where the high dimensional model is projected onto a set of optimally chosen bases [23,24]. The mechanics of a POD-ROM (POD-based ROMs) can be illustrated for a state variable $\mathbf{t} \in \mathbb{R}^N$, where N represents the size of the computational grid. The POD-ROM then approximates $\mathbf{t}$ as the linear expansion on a finite set of k orthonormal basis vectors $\{\boldsymbol{\phi}_i \in \mathbb{R}^N,\ i = 1, \ldots, k\}$, alternatively called as POD bases. That is,

$$\mathbf{t} \approx \sum_{i=1}^{k} \tilde{t}_i \boldsymbol{\phi}_i, \tag{1}$$

where $\tilde{t}_i \in \mathbb{R}$ is the ith component of $\tilde{\mathbf{t}} \in \mathbb{R}^k$, which are the coefficients of the basis expansion. The $\{\boldsymbol{\phi}_i\}$ are the POD *modes*. POD modes in Equation 1 can be shown to be the left singular vectors of the snapshot matrix (obtained by stacking M snapshots of $\mathbf{t}$), $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_M]$, extracted by performing a singular value decomposition (SVD) on $\mathbf{T}$ [25,26]. That is,

$$\mathbf{T} \underset{\text{svd}}{=} \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top, \tag{2}$$

where $\mathbf{U} \in \mathbb{R}^{N \times M}$ and $\boldsymbol{\Phi}_k$ represent the first k columns of $\mathbf{U}$ after truncating the last $M - k$ columns based on the relative magnitudes of the cumulative sum of their singular values. The total L_2 error in approximating the snapshots via the truncated POD basis is then given as

$$\sum_{j=1}^{M} \left\| \mathbf{t}_j - (\boldsymbol{\Phi}_k \boldsymbol{\Phi}_k^\top)\mathbf{t}_j \right\|_2^2 = \sum_{i=k+1}^{M} \sigma_i^2, \tag{3}$$

where σ_i is the singular value corresponding to the ith column of $\mathbf{U}$ and is also the ith diagonal element of $\boldsymbol{\Sigma}$. It is well known that the POD bases are L_2-optimal and present a good choice for an efficient compression of high-dimensional data.

An important point to note is the effect of premature truncation when representing dynamics in the POD space. For real-world problems that are advection-dominated, an impractically high number of POD-bases need to be retained to faithfully represent the flow-field. However, this limits the gains of data-driven or equation-based surrogate modeling from the perspective of computational efficiency as well as accuracy. Therefore, it is necessary to devise evolution strategies that can preserve the effects of the truncated scales on the resolved ones. In reduced-order modeling parlance, this is often referred to as a model that is equipped with closure. In this study, we leverage ideas from the application of time-delay embedded machine learning techniques with analogs to the Mori-Zwanzig formalism to account for closure implicitly [27,28]. We remind the reader that the Mori-Zwanzig formalism proposes the use of memory to account for

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_35

errors due to coarse-graining of systems. In the following sections, we shall introduce time-delay embedded forecasting techniques to handle forecasting in the reduced-space spanned by truncated POD basis vectors.

3 Models

Figure 1 represents a block diagram of the two-stage model, with the first stage comprising of a neural network based forecasting model, followed by a second stage error correction model based on GPR. The neural network model is trained on the POD coefficients of the training set to obtain an $m-$step forecasting. The predicted data is compared against the true coefficients to get the error e. The true POD coefficients of the training set and the corresponding error e from the neural network model are used in the error correction model to develop a function mapping between the two entities. This function mapping is used to estimate the "predicted error" e' which is then added to the predicted coefficients from the neural network model during the deployment phase. This results in an improved m-step forecasting of the POD coefficients. First, we briefly describe different machine learning approaches that we use within our two-stage model.

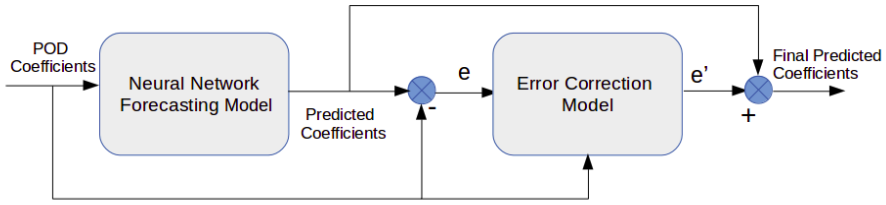


Fig. 1. Block diagram of the two-stage forecasting model.

3.1 Forecasting neural network models

In this paper, we consider a set of neural network models and compare the forecasting abilities of those models against each other. We consider two major families of neural network models. The first is Recurrent Neural Network (RNN) based algorithms such as vanilla LSTM, Bi-directional LSTM, and a sequence-to-sequence (Seq2Seq) Encoder-Decoder model. We also consider convolution neural network (CNN) based algorithms such as one-dimensional convolutional neural network (1D-CNN) and its variations such as vanilla-temporal convolutional neural network (TCN) and stacked-TCN. We briefly discuss these models below.

Recurrent Neural Network based models: RNNs have been the most obvious choice in recent times to model sequential data given their ability to characterise the temporal dependencies [29,30]. LSTMs are by far the most well known variant of RNNs [31], where the gated structure helps in learning long-term dependencies, by allowing the flow of information across several time steps. For this reason, they have been extensively used in time series predictions especially applications such as weather forecasting [32–34]. Bi-directional LSTM is a variant of LSTM, where one set of LSTMs is used across the forward direction of the sequential data, while another set is applied to the same sequence in the

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_35

reverse direction. Bi-LSTM is known to outperform vanilla LSTM [32, 35, 36], due to the fact that they tend to capture the temporal structure better than the latter by considering correlations in both forward and backward directions. For a prediction task, which is framed as generating a window of outputs in the future based on a window of inputs from the past, the bi-directional element of the modeling strategy ensures all components of the input contribute to the forecast of all components of the output. Though Seq2Seq encoder-decoder models traditionally have been associated with natural language processing (NLP) applications [37], they have been shown to be well suited for time series based forecasting problems too [38].

Convolutional network based models : Though traditionally convolutional neural networks have been applied to image data [39, 40], there are several variants of CNNs adapted to learn the temporal information from time series data [41]. Among them, the 1d-CNN is the simplest of the CNN models where the output is the convolution of input and the convolutional kernel over a single temporal dimension. We also use the more advanced TCNs that employ dialated causal convolution [42,43]. Such a network is seen to have information propogated over a larger number of time steps as compared to the recurrent models [44]. We further use a stacked-TCN architecture by stacking multiple TCN residual blocks.

3.2 Error Correction model

The predictions from the neural network model can be improved upon with a error correction model. This is espcially helpful in long term forecasting using neural network models with feedback, as discussed in detail in the following sections [45]. We use GPR as the primary error correction tool to improve the forecasts from the neural network models. The Gaussian process (GP) model is a well known non-parametric supervised learning method especially for regression problems [46]. Instead of point estimates, GP takes a Bayesian approach to providing a probability distribution for a set of possible functions that fit the training data. It specifies a prior distribution over the functions that characterizes the assumptions we make about the underlying functionality of the data yet to be seen. Upon observing the training data, the prior distribution is updated through their likelihood function. This resultant distribution, called the posterior distribution can be used to compute the predictive posterior distribution on the new unobserved (test) data. The prior distribution that we assume is completely defined by the choice of the kernel. We use a simple squared exponential kernel which can effectively forecast the errors, while being computationally efficient. We use the `sklearn`, a machine learning toolbox in Python, to implement GPR for our purposes.

4 Dataset and Numerical Results

We use the DayMET Version 3 model data which provides estimates of daily weather parameters for the North America region [47]. The data has 1x1-km spatial resolution and a temporal resolution of 1 calendar day. For most methods, we use the 2014-2015 data for training and 2016 data for validation purposes. The test forecasts are performed for the year 2017. For experimental study, we choose the maximum air temperature abbreviated as `tmax` to be the quantity of interest. We note that this procedure can be seamlessly extended to other

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_35

phyiscal quantities too. The raw data is preprocessed to remove all the oceanic points, leaving us with a snapshot of 201928 grid points mapping to the North American continental land mass. The POD basis set is computed on the snapshot data through the truncated SVD procedure with length k. The goal of POD is to contruct the orthogonal basis such that the resulting linear subspace can effectively represent the spatial and the temporal dynamics of `tmax`. A truncation of length k=5 results in 92% of the energy being conserved, implying the need for some time-delay based prediction model to account for unresolved scales. The coefficient matrix generated from the POD procedure is of dimension Txk, where T represents the number of temporal observations in the time series and k is the number of modes. Since we consider approximately two years of training data, $T_{train} = 730$. $T_{validation} = T_{test} = 365$, mapping to one year of temporal data for validation and test procedure respectively. The data is preprocessed using a traditional minmax scaler to scale each of the modes in the training data individually. Then the same scaling parameters are used to project the validation and test data. Figure 3 presents the histogram of the POD coefficients across the primary mode (mode 0) and the successive secondary modes (modes 1-3) for the DayMet data. As seen from the figure, the primary mode has a multimodal character, while all the secondary modes appear to be Gaussian. Figure 4 shows the autocorrelation plot for each of the 0th-3rd modes, which suggests that the primary mode observes a higher degree of autocorrelation compared to the secondary modes. This coefficient matrix, which holds the temporal information across different modes, is the input to the neural network models. We transform this time series dataset into a supervised learning problem by using the standard sliding window walk-forward method and perform a multivariate multi-step forecasting. We structure each sample of the training and validation data to have n time steps of input data (known as the window length) and to have the next m timesteps ($n + 1$ to $n + m$) be the labelled output data. We consider two scenarios for building the training data required for the supervised learning models.

Without feedback: Here we scan through the entire training and validation datasets and build the samples by the sliding window method. For example, the input of the first sample consists of data from $1 : n$ timesteps, and the corresponding output is the data from $n + 1 : n + m$ timesteps. The second sample's input data is the data from $2 : n + 1$ timesteps, and its correspoding output is from $n+2 : n+m+1$, and so on. This input-output structure is used to train different neural network models. For the prediction phase, the input is structured similarly to the one in the training phase. The model forecasts are evaluated and compared against the true values. This method can be effectively used to perform short-term forecasting, as the model deployment is made completely on the historical true values.

With feedback: This method has a training phase similar to the one above. It differs from the above method during the deployment phase, where the model forecast of the previous window is fed as the input to the next window. For example, to begin with the first n timesteps are input to the model, to forecast the next m timesteps. These m timesteps of predicted data are evaluated against

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_35

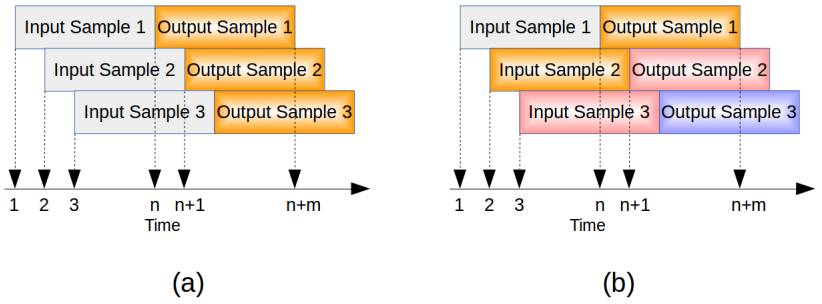


Fig. 2. Input-Output samples for a) no feedback b) with feedback

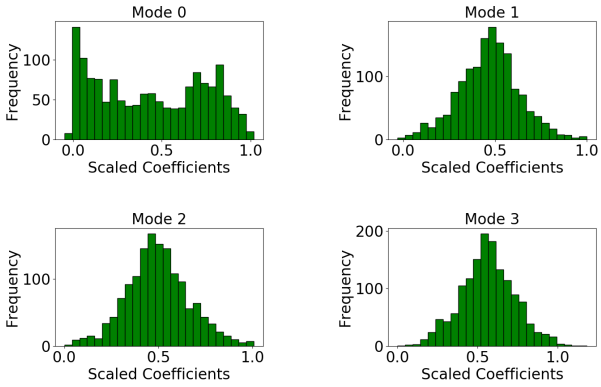


Fig. 3. Histogram of POD coefficients for primary mode (mode 0) and three secondary modes(mode 1-3).

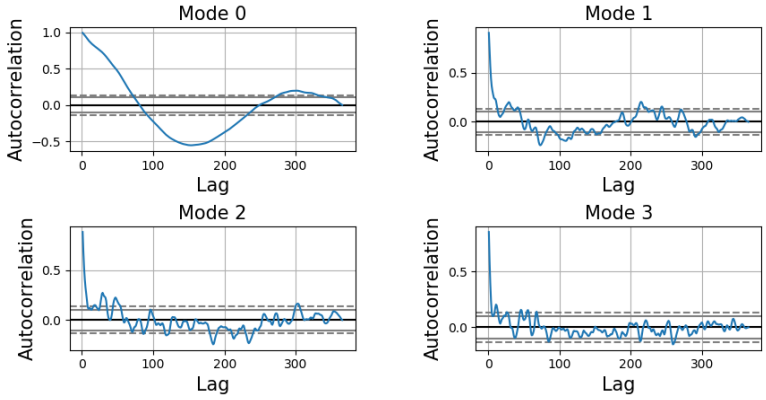


Fig. 4. Autocorrelation plots of POD coefficients for primary mode and secondary modes for test data.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_35

the true values and input to the model to produce the next set of predictions and so on. Such a method helps in long-term forecasting where the entire forecasting window is broken down into smaller forecasting periods. For the sake of simplicity, we consider a sliding interval of 1 while building the input samples for the model, and the number of input and output timesteps, that is, the window length, is $n = m = 7$ so that we are conducting weekly forecasts. We use a batch size of 32 and a learning rate of 0.0001 with the Adam optimizer, and these parameters are kept constant across the different neural network models. At first, we compare the forecast errors across the different models, without feedback. Figure 5 shows the error distributions for the test data for the different models, namely - LSTM, Bi-LSTM, Encoder-Decoder LSTM, 1D-Convolution, TCN, and STCN. The error distribution of each mode is shown seperately. We see that, while a simple Encoder-Decoder LSTM architecture is able to forecast quite accurately over the short term, the difference in the relative error for each of the modes is marginal over the different models. Therefore, we take a look at the training and the validation loss to evaluate the performance of the different models. Figure 6 and Figure 7 present the training and validation loss (we use mean squared error as the loss function) respectively during the training phase. It can be seen from the figures and table 1 that the Bi-LSTM, Encoder-Decoder, LSTM and 1D-Convolution are faster and more efficient to train than the other temporal convolution models that take longer to train with a very noisy training loss.

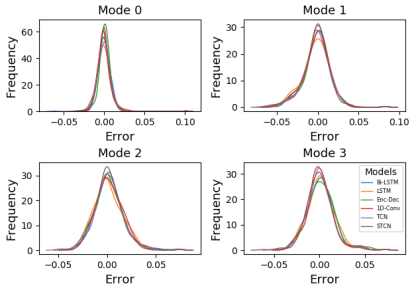


Fig. 5. Comparision of error density of test data without feedback over different models across the primary and secondary modes.

Model	Mean epoch time(s)	Total epochs	Total time to train(s)
LSTM	0.60	923	553.8
Bi-LSTM	1.146	655	750.6
Enc-Dec	0.81	624	505.4
1D-Conv	0.21	547	114.8
TCN	1.70	1033	1756
STCN	3.28	1797	5894

Table 1. Training time statistics for different models with early stopping enabled

Next, we consider the prediction with feedback models that provide long term forecasts. Figure 8 provides a comparision of the distributions of the relative errors across the different neural network models. It is natural to see an increase in the relative errors compared to the models without feedback (an increase of almost an order of magnitude). Across the different models, notice that the TCN

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_35

	w/o feedback			w/ feedback, before EC			w/ feedback, before EC		
	Train.	**Val.**	**Test**	**Train.**	**Val.**	**Test**	**Train.**	**Val.**	**Test**
Mode 0	0.05	0.04	0.05	0.15	0.15	0.16	0.10	0.10	0.09
Mode 1	0.11	0.10	0.10	0.13	0.10	0.12	0.03	0.02	0.03
Mode 2	0.11	0.09	0.10	0.11	0.12	0.12	0.03	0.03	0.03
Mode 3	0.12	0.11	0.12	0.13	0.10	0.14	0.04	0.04	0.04

Table 2. Mean absolute error statistics for training, validation and test data over different modes for i) without feedback, ii) with feedback and before error correction, iii) with feedback and error correction

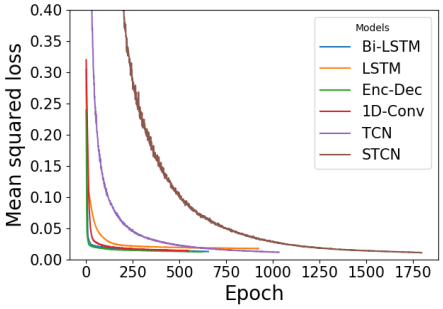


Fig. 6. Mean squared error training loss for the various models

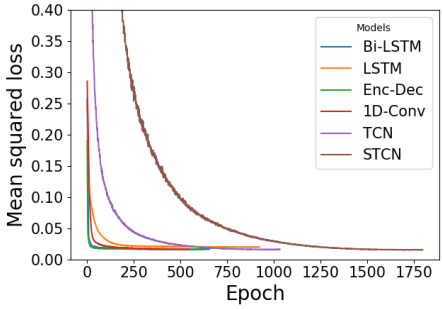


Fig. 7. Mean squared error validation loss for the various models

based models exibit higher standard deviations in their error distributions. The relative errors for each of the secondary modes exhibit a normal distribution, and we observe a bi-modal distribution for the primary mode, which encourages us to use an error correction model over the errors generated from the neural network model. Figures 9 and 10 compare the error distribution and mean absolute error for the Bi-LSTM model with feedback, before and after the error correction is applied. As seen in figures 3 and 9, for mode 0, both the coefficients and the errors from the neural network forecasts exhibit a bi-modal distribution, due to which the GPR does not fit the error data well for the first few days. But for long-term forecasting, we see that the GPR fits the error data well and the overall error distribution for each of the modes is more concentrated around the mean (0) after the error correction model is deployed, as compared to the error distribution from the neural network model forecasts that have a higher variance. Notice from table 2, that the mean absolute error for the no-feedback case is relatively small compared to the with-feedback scenario, which can be reduced by using an error correction model.

Figure 11 shows the viablity of the forecast frameworks in physical space where the GPR-based error correction is seen to enhance accuracy particularly when the emulator is applied recursively with feedback.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_35

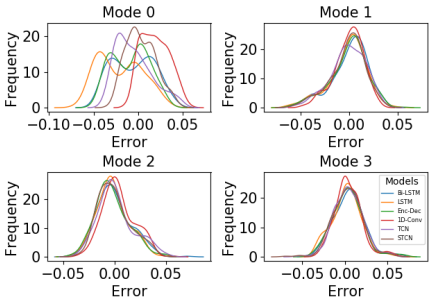


Fig. 8. Comparision of error density of test data with feedback over different models across the primary and secondary modes.

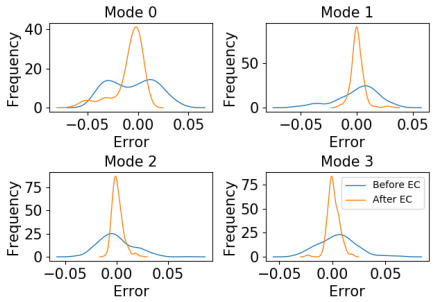


Fig. 9. Comparision of error density of test data with feedback for Bi-LSTM model across the primary and secondary modes, before and after error correction model is applied.

5 Conclusion

In this work, we present multiple data-driven approaches for forecasting maximum air temperature in the North-American region. We combine dimensionality reduction and system evolution using different flavors of deep learning models and deploy these techniques on real data. We mainly consider two scenarios in training and validating the supervised learning models. First, we consider a scenario where the training data is continuously enriched with new measurements *(without feedback)*, which is useful for producing short-term forecasts. The second scenario is developed for a longer forecast horizon *(with feedback)*.

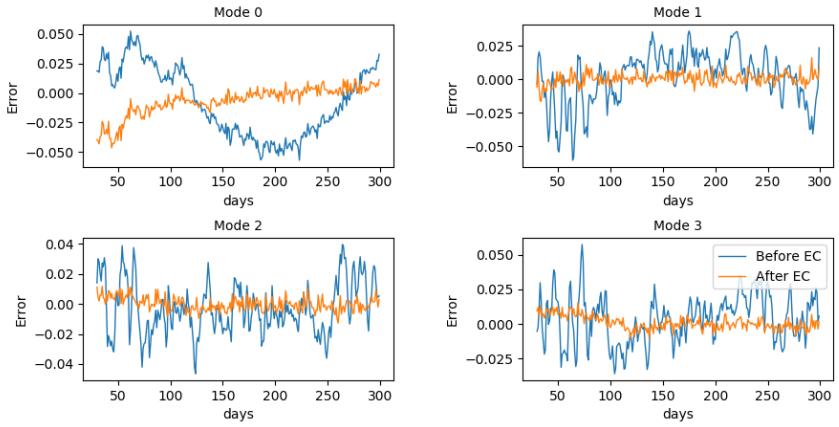


Fig. 10. Error of test data for the first 300 days of Bi-LSTM model with feedback, across the primary and secondary modes before and after the error correction is applied.

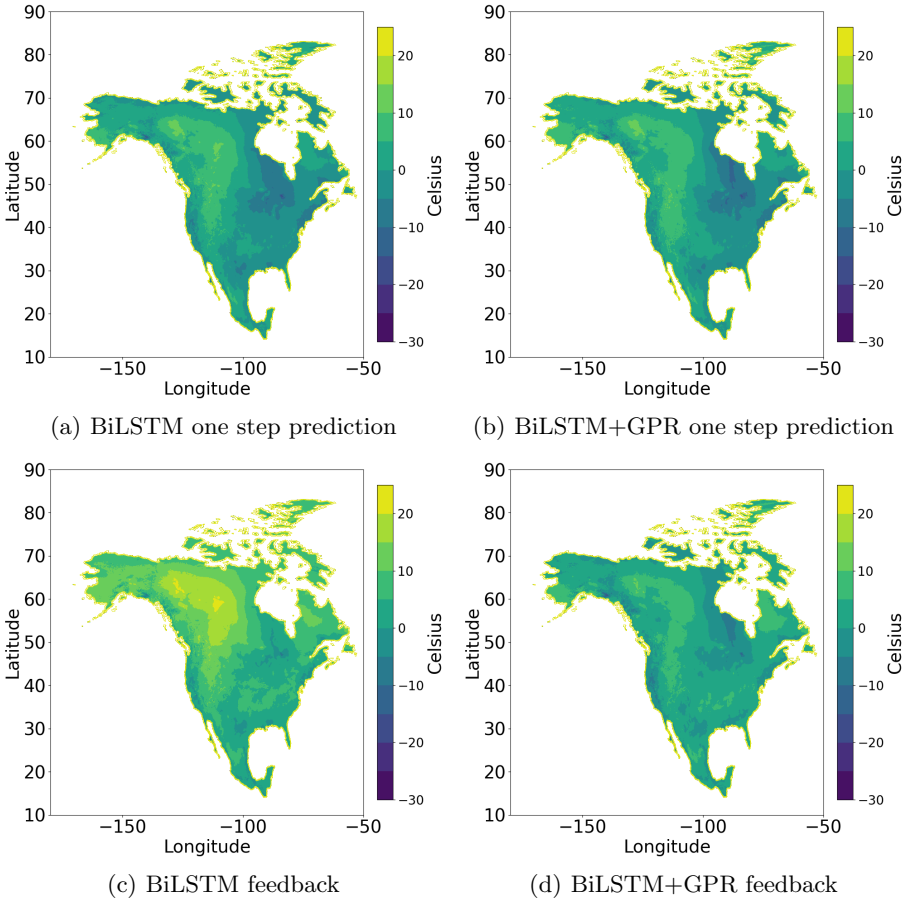
ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_35

In this scenario, the output of the supervised learning model from one cycle is used as the input in the next cycle and, as expected in such cases, the errors tend to accumulate and the method suffers from higher inaccuracies and instabilities than in the 'without feedback' case. We improve the 'with feedback' case by deploying a error correction term using GPR, and this vastly improves the forecasts. The forecast error in both primary and secondary modes is much smaller and hence allows for accurate forecasts for a much longer horizon. We have demonstrated these methods by carrying out extensive tests on DayMet data. In the future we will study the evolution of the discrete Lyapunov exponents to understand the stability of these methods and improve them. Additionally, we will also deploy state-of-the-art machine learning models to study multiple correlated physical quantities. Finally, we will also explore the use of Gaussin-mixtures as an error correction model to improve the error forecasts.

References

1. Juan-Carlos Ciscar, Ana Iglesias, Luc Feyen, László Szabó, Denise Van Regemorter, Bas Amelung, Robert Nicholls, Paul Watkiss, Ole B Christensen, Rutger Dankers, et al. Physical and economic consequences of climate change in europe. *Proceedings of the National Academy of Sciences*, 108(7):2678–2683, 2011.
2. Simon N Gosling, Jason A Lowe, Glenn R McGregor, Mark Pelling, and Bruce D Malamud. Associations between elevated atmospheric temperature and human mortality: a critical review of the literature. *Climatic change*, 92(3):299–341, 2009.
3. Pierre Gentine, Mike Pritchard, Stephan Rasp, Gael Reinaudi, and Galen Yacalis. Could machine learning break the convection parameterization deadlock? *Geophysical Research Letters*, 45(11):5742–5751, 2018.
4. Noah D Brenowitz and Christopher S Bretherton. Prognostic validation of a neural network unified physics parameterization. *Geophysical Research Letters*, 45(12):6289–6298, 2018.
5. Stephan Rasp, Michael S Pritchard, and Pierre Gentine. Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, 115(39):9684–9689, 2018.
6. Peter D Nooteboom, Qing Yi Feng, Cristóbal López, Emilio Hernández-García, and Henk A Dijkstra. Using network theory and machine learning to predict el niño. *Earth System Dynamics*, 9(3):969–983, 2018.
7. Azam Moosavi, Vishwas Rao, and Adrian Sandu. Machine learning based algorithms for uncertainty quantification in numerical weather prediction models. *Journal of Computational Science*, page 101295, 2021.
8. Yunjie Liu, Evan Racah, Joaquin Correa, Amir Khosrowshahi, David Lavers, Kenneth Kunkel, Michael Wehner, William Collins, et al. Application of deep convolutional neural networks for detecting extreme weather in climate datasets. *arXiv preprint arXiv:1605.01156*, 2016.
9. Sebastian Scher. Toward data-driven weather and climate forecasting: Approximating a simple general circulation model with deep learning. *Geophysical Research Letters*, 45(22):12–616, 2018.
10. Jonathan A Weyn, Dale R Durran, and Rich Caruana. Improving data-driven global weather prediction using deep convolutional neural networks on a cubed sphere. *Journal of Advances in Modeling Earth Systems*, 12(9):e2020MS002109, 2020.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_35



(a) BiLSTM one step prediction

(b) BiLSTM+GPR one step prediction

(c) BiLSTM feedback

(d) BiLSTM+GPR feedback

Fig. 11. Error contours displaying representative emulation performance on testing data set on the 30th of May, 2017. The first row contours indicate results from one step predictions using a BiLSTM without (a) and with GPR corrections (b). The second row displays error contours from a feedback application of the machine learning frameworks where there is no correction (c) and where a GPR is used to adjust the outputs of the BiLSTM. The use of the GP model is seen to improve predictions significantly for this case.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_35

11. Stephan Rasp and Nils Thuerey. Purely data-driven medium-range weather forecasting achieves comparable skill to physical models at similar resolution. *arXiv preprint arXiv:2008.08626*, 2020.
12. Ashesh Chattopadhyay, Ebrahim Nabizadeh, and Pedram Hassanzadeh. Analog forecasting of extreme-causing weather patterns using deep learning. *Journal of Advances in Modeling Earth Systems*, 12(2):e2019MS001958, 2020.
13. Eduardo Rocha Rodrigues, Igor Oliveira, Renato Cunha, and Marco Netto. Deepdownscale: a deep learning strategy for high-resolution weather forecast. In *2018 IEEE 14th International Conference on e-Science (e-Science)*, pages 415–422. IEEE, 2018.
14. Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *arXiv preprint arXiv:1506.04214*, 2015.
15. Romit Maulik, Romain Egele, Bethany Lusch, and Prasanna Balaprakash. Recurrent neural network architecture search for geophysical emulation. IEEE Press, 2020.
16. Dominic J Skinner and Romit Maulik. Meta-modeling strategy for data-driven forecasting. *arXiv preprint arXiv:2012.00678*, 2020.
17. Anuj Karpatne, Imme Ebert-Uphoff, Sai Ravela, Hassan Ali Babaie, and Vipin Kumar. Machine learning for the geosciences: Challenges and opportunities. *IEEE Transactions on Knowledge and Data Engineering*, 31(8):1544–1554, 2018.
18. Peter D Dueben and Peter Bauer. Challenges and design choices for global weather and climate models based on machine learning. *Geoscientific Model Development*, 11(10):3999–4009, 2018.
19. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
20. Sk M Rahman, Suraj Pawar, Omer San, Adil Rasheed, and Traian Iliescu. Nonintrusive reduced order modeling framework for quasigeostrophic turbulence. *Physical Review E*, 100(5):053306, 2019.
21. Arvind T Mohan and Datta V Gaitonde. A deep learning based approach to reduced order modeling for turbulent flow control using lstm neural networks. *arXiv preprint arXiv:1804.09269*, 2018.
22. Romit Maulik, Bethany Lusch, and Prasanna Balaprakash. Non-autoregressive time-series methods for stable parametric reduced-order models. *Physics of Fluids*, 32(8):087115, 2020.
23. Kunihiko Taira, Maziar S Hemati, Steven L Brunton, Yiyang Sun, Karthik Duraisamy, Shervin Bagheri, Scott TM Dawson, and Chi-An Yeh. Modal analysis of fluid flows: Applications and outlook. *AIAA J.*, pages 1–25, 2019.
24. Peter Benner, Serkan Gugercin, and Karen Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM review*, 57(4):483–531, 2015.
25. Clarence W. Holmes, Philip., Lumley, John L., Berkooz, Gahl and Rowley. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, volume 36. 1998.
26. Anindya Chatterjee. An introduction to the proper orthogonal decomposition. *Current science*, pages 808–817, 2000.
27. Romit Maulik, Arvind Mohan, Bethany Lusch, Sandeep Madireddy, Prasanna Balaprakash, and Daniel Livescu. Time-series learning of latent-space dynamics for reduced-order model closure. *Physica D: Nonlinear Phenomena*, 405:132368, 2020.
28. Chao Ma, Jianchun Wang, et al. Model reduction with memory and the machine learning of dynamical systems. *arXiv preprint arXiv:1808.04258*, 2018.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_35

29. Y. Yu, X. Si, C. Hu, and J. Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural Computation*, 31(7):1235–1270, 2019.
30. Gábor Petneházi. Recurrent Neural Networks for Time Series Forecasting. *arXiv e-prints*, page arXiv:1901.00069, December 2018.
31. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.
32. Zahra Karevan and Johan A.K. Suykens. Transductive lstm for time-series prediction: An application to weather forecasting. *Neural Networks*, 125:1–9, 2020.
33. Qin Zhang, Hui Wang, Junyu Dong, Guoqiang Zhong, and Xin Sun. Prediction of sea surface temperature using long short-term memory. *IEEE Geoscience and Remote Sensing Letters*, PP, 05 2017.
34. Clifford Broni-Bediako, Ferdinand Katsriku, Tatsuo Unemi, Norihiko Shinomiya, Jamal-Deen Abdulai, and Masayasu Atsumi. El niño-southern oscillation forecasting using complex networks analysis of lstm neural networks. 01 2018.
35. M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.
36. Sima Siami Namini, Neda Tavakoli, and Akbar Siami Namin. The performance of lstm and bilstm in forecasting time series. pages 3285–3292, 12 2019.
37. Richard Socher, Cliff Lin, Andrew Ng, and Christopher Manning. Parsing natural scenes and natural language with recursive neural networks. pages 129–136, 01 2011.
38. Guorui Shen, Jürgen Kurths, and Ye Yuan. Sequence-to-sequence prediction of spatiotemporal systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(2):023102, 2020.
39. Dan Ciresan, Ueli Meier, Jonathan Masci, Luca Maria Gambardella, and Jürgen Schmidhuber. Flexible, high performance convolutional neural networks for image classification. pages 1237–1242, 07 2011.
40. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, May 2017.
41. Shaojie Bai, J. Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. 03 2018.
42. Jining Yan, Lin Mu, Lizhe Wang, Rajiv Ranjan, and Albert Y Zomaya. Temporal convolutional networks for the advance prediction of enso. *Scientific reports*, 10(1):1–15, 2020.
43. Philippe Remy. Temporal convolutional networks for keras. https://github.com/philipperemy/keras-tcn, 2020.
44. Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.
45. Themistoklis P Sapsis and Andrew J Majda. Blending modified gaussian closure and non-gaussian reduced subspace methods for turbulent dynamical systems. *Journal of Nonlinear Science*, 23(6):1039–1071, 2013.
46. Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.
47. MM Thornton, PE Thornton, Y Wei, BW Mayer, RB Cook, and RS Vose. Daymet: Annual climate summaries on a 1-km grid for north america, version 3. ornl daac, oak ridge, tennessee, usa, 2016.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_35