Sub-Seasonal Climate Forecasting via Machine Learning: Challenges, Analysis, and Advances

Sijie He [*1], Xinyan Li [*1], Timothy DelSole[2], Pradeep Ravikumar[3], and Arindam Banerjee[1]

[1]Department of Computer Science & Engineering, University of Minnesota, Twin Cities
[2]Department of Atmospheric, Oceanic, and Earth Science, George Mason University
[3]Machine Learning Department, Carnegie Mellon University
Email: {hexxx893@umn.edu, lixx1166@umn.edu, tdelsole@gmu.edu, pradeepr@cs.cmu.edu, banerjee@cs.umn.edu}

Abstract

Sub-seasonal climate forecasting (SSF) focuses on predicting key climate variables such as temperature and precipitation in the 2-week to 2-month time scales. Skillful SSF would have immense societal value, in areas such as agricultural productivity, water resource management, transportation and aviation systems, and emergency planning for extreme weather events. However, SSF is considered more challenging than either weather prediction or even seasonal prediction. In this paper, we carefully study a variety of machine learning (ML) approaches for SSF over the US mainland. While atmosphere-land-ocean couplings and the limited amount of good quality data makes it hard to apply black-box ML naively, we show that with carefully constructed feature representations, even linear regression models, e.g., Lasso, can be made to perform well. Among a broad suite of 10 ML approaches considered, gradient boosting performs the best, and deep learning (DL) methods show some promise with careful architecture choices. Overall, suitable ML methods are able to outperform the climatological baseline, i.e., predictions based on the 30-year average at a given location and time. Further, based on studying feature importance, ocean (especially indices based on climatic oscillations such as El Niño) and land (soil moisture) covariates are found to be predictive, whereas atmospheric covariates are not considered helpful.

1 Introduction

Over the past few decades, major advances have been made in weather forecasts on time scales of days to about a week [33, 45, 36, 35]. Similarly, major advances have been made in seasonal forecasts on time scales of 2-9 months [4]. However, making high quality forecasts of key climate variables such as temperature and precipitation on sub-seasonal time scales, defined here as the time range between 2-8 weeks, has long been a gap in operational forecasting [35]. Skillful climate forecasts at sub-seasonal time scales would be of immense societal value, and would have an impact in a wide variety of domains including agricultural productivity, hydrology and water resource

[*]Equal Contribution

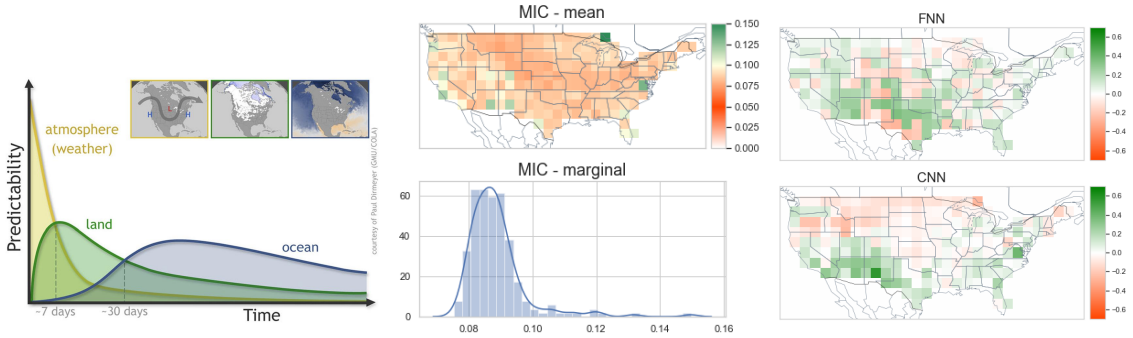
management, and emergency planning for extreme climate, etc. [37, 29]. The importance of sub-seasonal climate forecasting (SSF) has been discussed in great detail in two recent high profile reports from the National Academy of Sciences (NAS) [36, 35]. Despite the scientific, societal, and financial importance of SSF, the progress on the problem has been restricted [5, 9] partly because it has attracted less attention compared to weather and seasonal climate prediction. Also, SSF is arguably more difficult compared to weather or seasonal forecasting due to limited predictive information from land and ocean, and virtually no predictive information from the atmosphere, which forms the basis of numerical weather prediction (NWP) [49, 44] (Figure 1(a)).

There exists great potential to advance sub-seasonal prediction using machine learning techniques, which has revolutionized statistical prediction in many other fields. Due in large part to this potential promise, a recently concluded real-time forecasting competition called the Sub-Seasonal Climate Forecast Rodeo, was sponsored by the Bureau of Reclamation in partnership with NOAA, USGS, and the U.S. Army Corps of Engineers [26]. However, a direct application of standard black-box machine learning approaches to SSF can run into challenges due to the high-dimensionality and strong spatial correlation of the raw data from atmosphere, ocean, and land, e.g., Figure 1 shows that popular approaches such as Fully connected Neural Networks (FNN) and Convolutional Neural Networks (CNN) do not perform so well when directly applied to the raw data. Besides, sub-seasonal forecasting does not lie in the big data regime: about 40 years of reliable data exists for all climate variables, with each day corresponding to one data point, which totals less than 20,000 data points. Furthermore, different seasons have different predictive relations, and many climate variables have strong temporal correlations on daily time scales, further reducing the effective data size. Therefore, it is worth carefully and systematically investigating the capability of both classical and modern Machine Learning (ML) approaches including Deep Learning (DL) while keeping in mind the high-dimensionality, spatial-temporal correlations, and limited observational data available for SSF. While such a study can be extended to consider climate model output data at sub-seasonal time scales, we do not consider using model output data in the current study.

In this paper, we perform a comprehensive empirical study on ML approaches for SSF and discuss the challenges and advancements. Our main contributions are as follows:

- We illustrate the difficulty of SSF due to the complex physical couplings as well as the unique nature of climate data, i.e., strong spatial-temporal correlation and high-dimensionality.
- We show that suitable ML models, e.g., XGBoost, to some extent, capture predictability for sub-seasonal time scales from climate data, and persistently outperform existing approaches in climate science, such as climatology and the damped persistence model.
- We demonstrate that even though DL models are not the obvious winner, they still show promising results with demonstrated improvements from careful architectural choices. With further improvements, DL models present a great potential topic for future research.
- We find that ML models tend to select covariates from the land and ocean, such as soil moisture and El Niño indices, and rarely select atmospheric covariates, such as 500mb geopotential height or other indicators of the atmospheric general circulation.

Organization of the paper. We start with a review of related work in Section 2. Section 3 provides a formal description of the specific SSF problem targeted in this paper and demonstrates the difficulty of sub-seasonal climate forecasting for ML techniques. Next, we briefly discuss ML



(a) Sources of Predictability (b) MIC (c) Results of FNN and CNN

Figure 1: (a) Sources of predictability at different forecast time scales. Atmosphere is most predictive at weather time scales, whereas for SSF, land and ocean are considered important sources of predictability [49]. (b) Maximum information coefficient (MIC) [40] between temperature of week 3 & 4 and week -2 & -1. Small MICs (≤ 0.1) at a majority of locations indicate little information shared between the most recent date and the forecasting target. (c) Predictive skills of Fully connected Neural Networks (FNN) and Convolutional Neural Networks (CNN), in terms of temporal cosine similarity (see definition in Section 5), for temperature prediction over 2017-2018. Positive values closer to 1 (green) indicate better predictive skills.

approaches we plan to investigate (Section 4) followed by details on data and experimental setup (Section 5). Subsequently, Section 6.3 presents experimental results, comparing the predictive skills over 10 ML models, including several DL models. Finally, we conclude in Section 7.

2 Related Work

Although statistical models were used for weather prediction before the 1970s [14], since the 1980s weather forecasting has been carried out using mainly physics-based dynamic system models [4]. More recently, there is a surge of application for ML approaches to both short-term weather forecasting [7, 21, 38], and longer-term climate prediction [1, 8]. However, little attention has been paid on forecasting with sub-seasonal time scale which has been considered a "predictability desert" [56]. Recently, ML techniques have made great strides in statistical prediction in many fields, so it is natural to investigate whether it can advance sub-seasonal climate prediction. In particular, many advances have occurred in high-dimensional sparse models and their variants which could be suitable for spatial-temporal climate data [19, 20, 18, 10, 23]. Such models have been successfully applied to certain problems, e.g., predicting land temperature using oceanic climate data [10, 23]. Recently, promising progresses [26, 23] have been seen on applying ML algorithms to solve SSF.

Since SSF can be formulated as a sequential modeling problem [48, 55], bringing the core strength of DL-based sequential modeling, a thriving research area, has the great potential for a transformation in climate forecasting [22, 39, 42]. In the past decade, recurrent neural network (RNN) [16], and long short-term memory (LSTM) models [17], are two of the most popular sequential models and have been successfully applied in language modeling and other seq-to-seq tasks [47]. Starting from [48, 46], the encoder-decoder structure with RNN or LSTM has become one of the most competitive

Table 1: Description of climate variables and their data sources.

Type	Climate variable	Description	Unit	Spatial coverage	Data Source
Spatial-temporal	tmp2m	Daily average temperature at 2 meters	C°	US mainland	CPC Global Daily Temperature [13]
	sm	Monthly Soil moisture	mm		CPC Soil Moisture [25, 53, 12]
	sst	Daily sea surface temperature	C°	North Pacific & Atlantic Ocean	Optimum Interpolation SST (OISST) [41]
	rhum	Daily relative humidity near the surface (sigma level 0.995)	%	US mainland and North Pacific & Atlantic Ocean	Atmospheric Research Reanalysis Dataset [28]
	slp	Daily pressure at sea level	Pa		
	hgt10 & hgt500	Daily geopotential height at 10mb and 500mb	m		
Temporal	MEI	Bimonthly multivariate ENSO index	NA	NA	NOAA ESRL MEI.v2 [62]
	Niño 1+2, 3, 3.4, 4	Weekly Oceanic Niño Index (ONI)			NOAA National Weather Service, CPC [41]
	NAO	Daily North Atlantic Oscillation index			NOAA National Weather Service, CPC [3, 51]
	MJO phase & amplitude	Madden-Julian Oscillation index			Australian Government BoM [60]

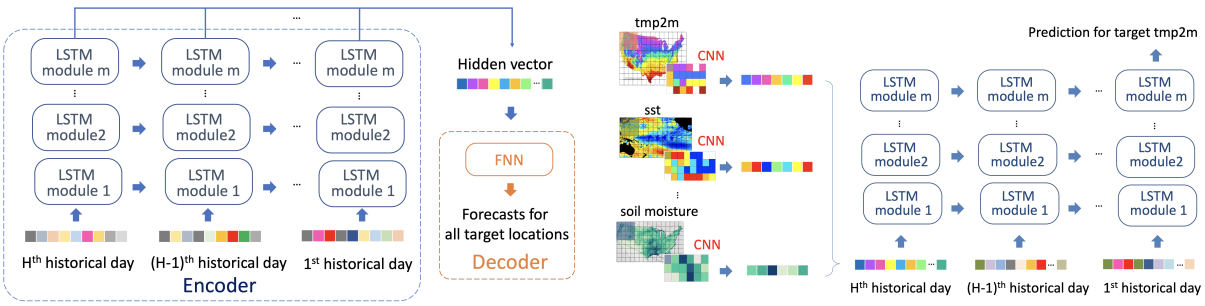
algorithms for sequence transduction. The variants of such model that incorporate mechanisms like convolution [61, 43] or attention mechanisms [2] have achieved remarkable breakthroughs for audio synthesis, word-level language modeling, and machine translation [54].

3 Sub-seasonal Climate Forecasting

Problem statement. In this paper, we focus on building temperature forecasting models at the forecast horizon of 15-28 days ahead, i.e., the average daily temperature of week 3 & 4. The geographic region of interest is the US mainland (latitudes 25N-49N and longitudes 76W-133W) at a 2° by 2° resolution (197 grid points). For covariates, we consider climate variables, such as sea surface temperature, soil moisture, and geopotential height, etc., that can indicate the status of the three main components, i.e., land, ocean, and atmosphere. Table 1 provides a detailed description.

Difficulty of the problem. To illustrate the challenge of SSF, we measure the dependence between the normalized average temperature of week -2 & -1 (1-14 days in the past) and week 3 & 4 (15-28 days in the "future") at each grid by maximum information coefficient (MIC) [40], an information theory-based measure of the linear or non-linear association between two variables. The values of MIC range between 0 and 1, and a small MIC value close to 0 indicates a weak dependence. To assess statistical significance, we apply moving block bootstrap [30] to time series of 2-week average temperature at each grid point from 1986 to 2018, with the block size of 365 days. The top panel in Figure 1(b) illustrates the average MIC from 100 bootstrap over the US mainland, and the marginal distribution of all locations is shown at bottom. Small MIC values (≤ 0.1), which indicates little predictive information shared between the most recent data and the forecasting target, to some extent, demonstrate how difficult SSF is.

From an ML perspective, applying black-box DL approaches naively to SSF is less likely to work due to the limited number of samples, and the high-dimensional and spatial-temporally correlated features. Figure 1(c) shows the performance of two vanilla DL models: Fully connected Neural



(a) Encoder (LSTM)-Decoder (FNN) (b) CNN-LSTM

Figure 2: Architectures of the designed DL models. (a) Encoder (LSTM)-Decoder (FNN) includes a few LSTM layers as the Encoder, and two fully connected layers as the Decoder. (b) CNN-LSTM consists of a few convolutional layers followed by an LSTM.

Networks (FNN) with ReLU activation and Convolutional Neural Networks (CNN), in terms of the (temporal) cosine similarity between the prediction and the ground truth at each location over 2017-2018. For most locations, their cosine similarities are either negative or close to zero. In addition, we evaluate 10 ML models with suitable hyper-parameter tuning using another metric called relative R^2 (see formal definition in Appendix A), which compares the predictive skill of a model to the best constant prediction based on climatology, the 30 year average from historical training data. Most of the models do not get even positive relative R^2 (details are presented in Appendix A), indicating that they perform no better than the long term average. Such results are another good indication that accurate SSF is hard to achieve.

4 Methods

Notation. Let t denote a date and g denote a location. The target variable at time t is denoted as $\mathbf{y}_t \in \mathbb{R}^G$, where G represents the number of target locations. More specifically, $y_{g,t}$ is the normalized average temperature over time $t+15$ to $t+28$, i.e., weeks 3 & 4 (details on normalization can be found in Section 5). $X_{g,t} \in \mathbb{R}^p$ denotes the p-dimensional covariates designed for time t and location g, which is also denoted as X_t if the covariates are shared by all locations $g \in G$.

ML (non-DL) models. We compare the following ML (non-DL) models with DL models.

- **MultiLLR [26].** MultiLLR introduces a multitask feature selection algorithm to remove the irrelevant predictors and integrates the remaining predictors linearly. For a location g and target date t^*, its coefficient β_g is estimated by $\hat{\beta}_g = \text{argmin}_\beta \sum_{t \in \mathcal{D}} w_{t,g}(y_{g,t} - \beta^T X_{g,t})^2$, where $\mathcal{D}$ is the temporal span around the target date's day of the year and $w_{t,g}$ is the corresponding weight. In [26], an equal data point weighting ($w_{t,g} = 1$) has been employed.
- **AutoKNN [26].** An auto-regression model with weighted temporally local samples where the auto-regression lags are selected via a multitask k-nearest neighbor criterion. The method only takes historical measurements of the target variables as input. The nearest neighbors of each target date are selected based on an average of spatial cosine similarity computed over a history of $M = 60$ days, starting one year prior to a target date t^* (lag $l = 365$). More precisely, the similarity between the target date t^* and a date t in the corresponding training set is formulated as

$\text{sim}_t = \frac{1}{M}\sum_{m=0}^{M-1}\cos(\mathbf{y}_{t-l-m}, \mathbf{y}_{t^*-l-m})$, where $\cos(\mathbf{y}_{t_1}, \mathbf{y}_{t_2})$ computes the (spatial) cosine similarity (see formal definition in Section 5), evaluated over G locations, between two given dates t_1 and t_2.

- **Multitask Lasso [50, 27].** It assumes $\mathbf{y}_t = X_t\Theta^* + \epsilon$, where $\epsilon \in \mathbb{R}^G$ is a Gaussian noise vector and $\Theta^* \in \mathbb{R}^{p\times G}$ is the coefficient matrix for all locations. With n samples, Θ^* is estimated by $\hat{\Theta}_n = \text{argmin}_{\Theta\in\mathbb{R}^{p\times G}} \frac{1}{2n}\|Y - X\Theta\|_2^2 + \lambda_n\|\Theta\|_{21}$ with $X \in \mathbb{R}^{n\times p}$ and $Y \in \mathbb{R}^{n\times G}$. λ_n is a penalty parameter and the corresponding penalty term is computed as $||\Theta||_{21} = \sum_i(\sum_j \Theta_{ij}^2)^{1/2}$.
- **Gradient Boosted Trees (XGBoost) [15, 6].** A functional gradient boosting algorithm using regression tree as its weak learner. The algorithm starts with one weak learner and iteratively adds new weak learners to approximate functional gradients. The final ensemble model is constructed by a weighted summation of all weak learners. It is implemented using the Python package XGBoost.
- **State-of-the-art Climate Baseline.** We consider two baselines from climate science perspective, both are Least Square (LS) linear regression models [59]. The first model has predictors as climate indices, such as NAO index and Niño indices, which are used to monitor ocean conditions. The predictor of the second model is the most recent anomaly of the target variable, i.e., anomaly temperature of week -2 & -1, with which the model, also known as *damped persistence* [52] in climate science, is essentially a first-order autoregressive model.

DL models. As shown in Figure 1(c), it is hard for vanilla deep learning models like FNN and CNN to achieve high prediction accuracy. Therefore, we design two variants of DL models to improve the performance, namely Encoder (LSTM)-Decoder (FNN) and CNN-LSTM.

- **Encoder (LSTM)-Decoder (FNN).** Inspired by Autoencoder widely used in sequential modeling [48], we design the Encoder (LSTM)-Decoder (FNN) model, of which the architecture is shown in Figure 2(a). Input of the model is features extracted spatially from covariates using unsupervised methods like Principal Component Analysis (PCA). The temporal components of covariates are handled by feeding features of each historical date into an LSTM Encoder recurrently. Then, the output of each date from LSTM is sent jointly to a two-layer FNN network using ReLU as an activation function. The output of the FNN Decoder is the predicted average temperature of week 3 & 4 over all target locations.
- **CNN-LSTM.** The proposed CNN-LSTM model directly learns the representations from the spatial-temporal data using CNN components [31]. Shown in Figure 2(b), CNN extracts features for each climate variable at all historical dates separately. Then, the extracted features from the same date are collected and fed into an LSTM model recurrently. The temperature prediction for all target locations is done by an FNN layer taking the output of the LSTM's last layer from the latest input.

5 Data and Experimental Setup

Data description. Climate agencies across the world maintain multiple datasets with different formats and resolutions. Climate variables (Table 1) have been collected from diverse data sources and converted into a consistent format. Temporal variables, e.g., Niño indices, are interpolated to a daily resolution, and spatial-temporal variables are interpolated to a spatial resolution of 0.5° by 0.5°.

Preprocessing. For spatial-temporal variables, we first extract the top 10 principal components (PCs) as features based on PC loadings from 1986 to 2016 (for details, refer to Appendix B). Next, we normalize the data by z-scoring at each location and each date with the corresponding mean and standard deviation of the corresponding day of the year over 1986-2016 separately. Note that both training and test sets are z-scored using the mean and standard deviations of the same 30-year historical data. Temporal variables, e.g., Niño indices, are directly used without normalization.

Feature set construction. We combine the PCs of spatial-temporal covariates with temporal covariates into a sequential feature set, which consists not only covariates of the target date, but also covariates of the 7^{th}, 14^{th}, and 28^{th} day previous from the target date, as well as the day of the year of the target date in the past 2 years and both the historical past and future dates around the day of the year of the target date in the past 2 years (see Appendix B for a detailed example).

Evaluation pipeline. Predictive models are created independently for each month in 2017 and 2018. To mimic a live system, we generate 105 test dates during 2017-2018, one for each week, and group them into 24 test sets by their month of the year. Given a test set, our evaluation pipeline consists of two parts: (1) "5-fold" training-validation pairs for hyper-parameter tuning, based on a "sliding-window" strategy designed for time-series data. Each validation set uses the data from the same month of the year as the test set, and we create 5 such set from dates in the past 5 years. Their corresponding training sets contain 10 years of data before each validation set; (2) the training-test pair, where the training set, including 30-year data in the past, ends 28 days before the first date in the test set. We share more explanations, including a pictorial example, in Appendix B.

Evaluation metrics. Forecasts are evaluated by cosine similarity, a widely used metric in weather forecasting, between $\hat{\mathbf{y}}$, a vector of predicted values, and $\mathbf{y}^*$, the corresponding ground truth. It is computed as $\frac{\langle \hat{\mathbf{y}}, \mathbf{y}^* \rangle}{\|\hat{\mathbf{y}}\|_2 \|\mathbf{y}^*\|_2}$, where $\langle \hat{\mathbf{y}}, \mathbf{y}^* \rangle$ denotes the inner product between the two vectors. If $\hat{\mathbf{y}}$ represents the predicted values for a period of time at one location, it becomes *temporal cosine similarity* which assesses the prediction skill at a specific location. Whereas, if $\hat{\mathbf{y}}$ contains the predicted values for all target locations at one date, it becomes *spatial cosine similarity* measuring the prediction skill at that date. To get a better intuition, one can view spatial and temporal cosine similarity as spatial and temporal correlation respectively, measured between two centered vectors.

6 Experimental Results

We compare the predictive skills of 10 ML models on SSF over the US mainland. In addition, we discuss a few aspects that impact the ML models the most, and the evolution of our DL models.

6.1 Results of all methods

Temporal results. Table 2 lists the mean, the median, the 0.25 quantile, the 0.75 quantile, and their corresponding standard errors of spatial cosine similarity of all methods. Additional results based on relative R^2 can be found in Appendix C. XGBoost, Encoder (LSTM)+Decoder (FNN) and Lasso accomplish higher predictive skills than other presented methods, and can outperform climatology and two climate baseline models, i.e., LS with NAO & Niño, and damped persistence. Overall, XGBoost achieves the highest predictive skill in terms of both the mean and the median, demonstrating its predictive power. Surprisingly, linear regression with a proper feature set has good predictive performance. Even though DL models are not the obvious winner, with careful architectural selections, they still show promising results.

Table 2: Comparison of spatial cosine similarity of tmp2m forecasting for test sets over 2017-2018. XGBoost and Encoder (LSTM)-Decoder (FNN) have the best performance. Models achieve better performance using temporally global set compared to temporally local set.

Model	Mean(se)	Median (se)	0.25 quantile (se)	0.75 quantile (se)
Temporally Global Dataset				
XGBoost - one day	**0.3044(0.03)**	**0.3447(0.05)**	**0.0252(0.05)**	**0.5905(0.04**)
Lasso - one day	0.2499(0.04)	0.2554(0.06)	-0.0224(0.05)	0.5604(0.06)
Encoder (LSTM)-Decoder (FNN)	**0.2616 (0.04**)	**0.2995 (0.07)**	**-0.0719 (0.06)**	**0.6310 (0.05)**
FNN	0.0792(0.01)	0.0920(0.02)	0.0085(0.02)	0.1655(0.02)
CNN	0.1688(0.04)	0.2324(0.06)	-0.0662(0.06)	0.4768(0.04)
CNN-LSTM	0.1743(0.04)	0.2867(0.06)	-0.1225(0.07)	0.5148(0.04)
LS with NAO & Niño	**0.2415(0.03)**	**0.3169(0.04)**	**0.0454(0.05)**	**0.4624(0.03)**
Damped persistence	0.2009(0.04)	0.2310(0.06)	-0.0884(0.06)	0.5335(0.05))
MultiLLR	0.0684 (0.03)	0.1046 (0.05)	-0.1764 (0.06)	0.3156 (0.04)
AutoKNN	0.1457 (0.03)	0.1744 (0.05)	-0.1018 (0.06)	0.4000 (0.04)
Temporally Local Dataset				
XGBoost - one day	0.1965(0.04)	0.2345(0.05)	-0.0636(0.06)	0.5178(0.05)
Lasso - one day	0.1631(0.04)	0.2087(0.06)	-0.1178(0.05)	0.5059(0.05)
Encoder (LSTM)-Decoder (FNN)	0.1277 (0.04)	0.1272 (0.06)	-0.1558 (0.06)	0.4971 (0.06)

Spatial results. Figure 3 shows the temporal cosine similarity of all methods evaluated on test sets described in Section 5. Among all methods, XGBoost and the Encoder (LSTM)-Decoder (FNN) achieve the overall best performance, regarding the number of locations with positive temporal cosine similarity. Qualitatively, coastal and south regions, in general, are easier to predict compared to inland regions (e.g., Midwest). Such a phenomenon might be explained by the influence of the slow-moving component, i.e., Pacific and Atlantic Ocean. Such component exhibits inertia or memory, in which anomalous condition can take relatively long period of time to decay. However, each model has its own favorable and disadvantageous regions. For example, XGBoost and Lasso do poorly in Montana, Wyoming, and Idaho, while Encoder (LSTM)-Decoder (FNN) performs much better on those regions. The observations naturally imply that the ensemble of multiple models is a promising future direction.

Comparison with the state-of-the-art methods. MultiLLR and AutoKNN are two state-of-the-art methods designed for SSF on western US [26]. Both methods have shown good forecasting performance on the original target region. However, over the inland region (Midwest), Northeast, and South region, the methods do not perform so well (Figure 3). To be fair, even though a similar set of climate variables have been used in our work compared to the original paper [26], how we prepossess the data and construct the feature set are slightly different. Such differences may lead to relatively poor performance for these two methods, especially for MultiLLR.

6.2 Analysis and exploration

We analyze and explore several important aspects that could influence the performance of ML models.

Temporally "local" vs. "global" dataset. Our training set consists of all calendar months over the past 30 years, which we refer to as the temporally "global" dataset. Another way to construct the training set is to consider calendar months only within the temporal neighborhood of the test

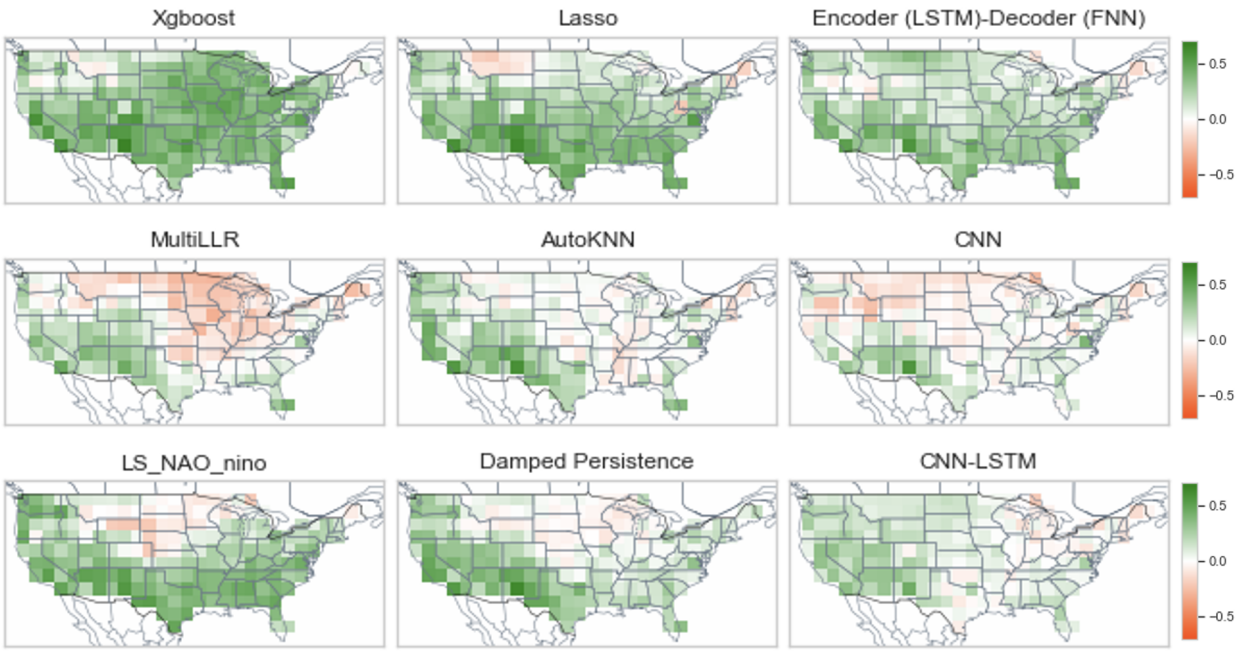
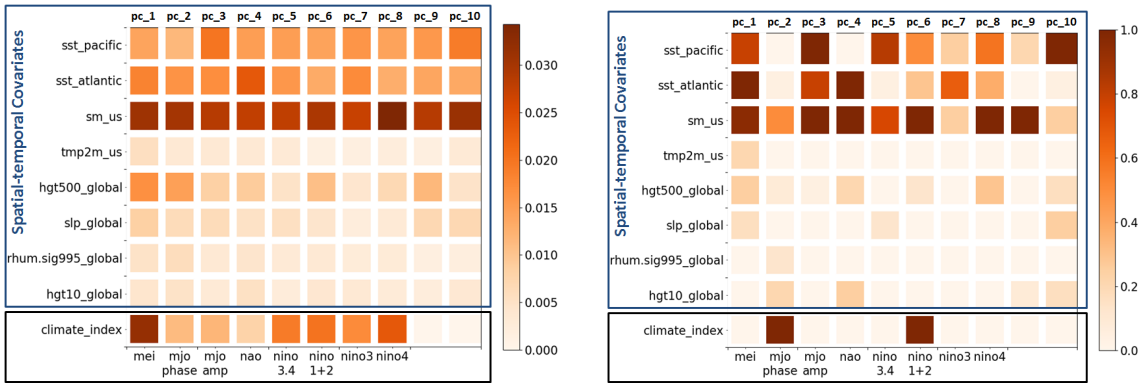


Figure 3: Temporal cosine similarity over the US mainland of ML models discussed in Section 4 for temperature prediction over 2017-2018. Large positive values (green) closer to 1 indicates better predictive skills. Overall, XGBoost and Encoder (LSTM)-Decoder (FNN) perform the best. Qualitatively, coastal and south regions are easier to predict than inland regions (e.g., Midwest).

date. For instance, to build a predictive model to forecast June in 2017, the training set can only contains dates in June (from earlier years), and months that are close to June, e.g., April, May, July, and August, over the past 30 years. Such a construction account for the seasonal dependence of predictive relations, for example summer predictions are not trained with winter data. We name such dataset as a temporally "local" dataset. A comparison between the "global" and "local" datasets has been listed in Table 2 where a significant drop in cosine similarity can be noticed when using "local" dataset for all of our best predictors, including XGBoost, Lasso, and Encoder (LSTM)-Decoder (FNN). We suspect such performance drop from "global" to "local" dataset may come from the reduction in the number of effective samples.

Feature importance. At sub-seasonal time scales, it is widely believed [49, 11] that land and ocean are the important sources of predictability, while the impact of atmosphere is limited (Figure 1 (a)). We study which covariate(s) are important, considered by ML models, based on the feature importance score. In particular, we compute the feature importance score from 2 ML models, XGBoost and Lasso (Figure 4). For XGBoost, the importance score is computed using the average information gain across all tree nodes a feature/covariate splits, while for Lasso, we simply count the non-zero coefficients of each model. The reported feature importance score is the average over 24 models (one per month in 2017-2018). In addition, we also provide measurement of feature importance based on Shapley value (Figure 5), a concept from game theory [32]. To determine the Shapley value of a given feature, we compute the prediction difference between a model trained with and without that feature. Since the effect of suppressing a feature also depends on other features, we have to consider all possible subsets of other features, and compute the Shapley values as a weighted



(a) XGBoost (b) Lasso

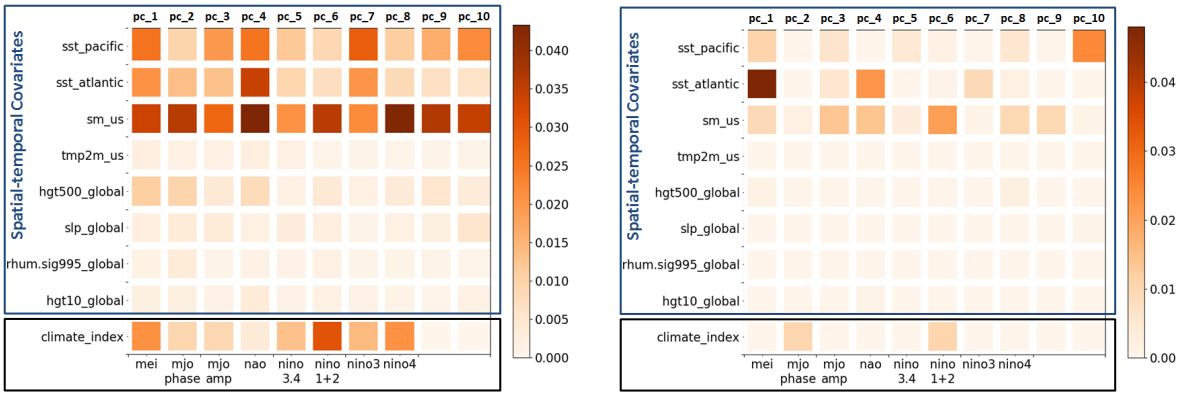
Figure 4: Feature importance scores computed from (a) XGBoost and (b) Lasso. Darker color means a covariate is of the higher importance. The first 8 rows contains the top 10 principal components (PCs) extracted from 8 spatial-temporal covariates respectively, and the last row includes all the temporal indices. Land component, e.g., soil moisture (3^{rd} row from the top) and ocean components, e.g., sst (Pacific and Atlantic) and some climate indices are the most commonly selected covariates.

average of all possible differences. As shown in Figure 4 and 5, among all covariates, soil moisture (3^{rd} row from the top) is the variable that has been constantly selected by both models. Another set of important covariates is the family of Niño indices. A LS model using those indices alone as predictors performs fairly well (Table 2). Sea surface temperatures (of Pacific and Atlantic) are also commonly selected. Such observations indicate that ML models pick up ocean-based covariates, some land-based covariates, and almost entirely ignore the atmosphere-related covariates, which are well aligned with domain knowledge [49, 11].

The influence of feature sequence length. To adapt the usage of LSTM, we construct a sequential feature set, which consists not only the target date, but also 17 other dates preceding the target date. However, other ML models, e.g., XGBoost and Lasso, which are not designed to handle sequential data, experience a drastic performance drop when we include more information from the past. More precisely, by including covariates from the full historical sequence, the performance of XGBoost drops approximately 50% compared to the XGBoost model using covariates from the most recent date only. A possible explanation for such performance degradation, as we increase the feature sequence length, is that both models weight covariates from different dates exactly the same without considering temporal information, thus some irrelevant historical information might mislead the model. In Appendix C, we provide a more detailed comparison among results obtained from various sequence lengths.

6.3 What happened with DL models?

As discussed in Section 3, applying black-box DL models naively does not work well for SSF. The improvement (Table 2), as we evolve from FNN to CNN-LSTM, and finally to Encoder (LSTM)-Decoder (FNN), demonstrates how the network architecture plays an important role, and leaves us plenty of space for further advancement. Below we focus on the discussion of feature representation, and the architecture design for sequence modeling. More discussion is included in Appendix C.



(a) XGBoost (b) Lasso

Figure 5: Shapley values computed from (a) XGBoost and (b) Lasso. Darker color means a covariate is of the higher importance. The observation that the most commonly selected covariates are from land and ocean are consistent with Figure 4.

Feature representation: CNN vs. PCA. Since SSF can be considered as a spatial-temporal prediction problem, CNN [31] is a natural choice to handle the spatial aspect of each climate covariate by viewing it as a map, and can be applied as a "supervised" way for learning feature representation. However, our results imply that models with CNN has limited predictive skills regarding both spatial and temporal cosine similarity. CNN, while doing convolution using a small kernel, mainly focus on spatially localized regions. However, the strong spatial correlation of climate variables restricts the effectiveness of CNN kernels on feature extraction. Meanwhile, PCA, termed Empirical Orthogonal Functions (EOF) [34, 57] in climate science, is a commonly used "unsupervised" feature representation method, which focuses on low-rank modeling of spatial covariance structure revealing spatial connection. Our results (Table 2) illustrate that PCA-based models have higher predictive skills than CNN-based models, verifying that PCA is a better technique for feature extraction in SSF.

Sequential modeling: Encoder-Decoder. With features extracted by PCA, we formulate SSF as a sequential modeling problem [48], where the input is the covariates sequence described in Section 5, and the output is the target variable. Due to its immense success in sequential modeling [46, 55], the standard Encoder-Decoder, where both Encoder and Decoder are LSTM [24], is the first architecture we investigate. Unfortunately, the model does not perform well and suffers from over-fitting, possibly caused by overly complex architecture. To reduce the model complexity, we replace the LSTM Decoder with an FNN Decoder which takes only the last step of the output sequence from the Encoder. Such change leads to an immediate boost of predictive performance. However, the input of the FNN Decoder mainly contains information encoded from the latest day in the input sequence and can only embed limited amount of historical information owing to the recurrent architecture of LSTM. To further improve the performance, we adjust the connection between Encoder and Decoder, such that FNN Decoder takes every step of the output sequence from LSTM Encoder, which makes a better use of historical information. Eventually, this architecture achieves the best performance among all investigated Encoder-Decoder variants (see a detailed comparisons in Appendix C).

7 Conclusion

In this paper, we investigate the great potential to advance sub-seasonal climate forecasting using ML techniques. SSF, the skillful forecasts of temperature on the time range between 2-8 weeks, is a challenging task, since it is beyond the limit of atmospheric predictability commonly used in short-term weather forecasting. Besides, SSF is typically a high-dimensional problem with limited effective samples due to strong spatial-temporal correlation within climate data. We conduct a comprehensive analysis of 10 different ML models, including a few DL models. Empirical results show the gradient boosting model XGBoost, the DL model Encoder (LSTM)-Decoder (FNN), and linear models, such as Lasso, consistently outperform state-of-the-art forecasts. XGBoost has the highest skill over all models, and demonstrates its predictive power. ML models are capable of picking the climate variables from important sources of predictability in SSF, identified by climate scientists. In addition, DL models, with demonstrated improvements from careful architectural choices, are great potentials for future research.

8 Broader Impact

Skillful (i.e., accurate) climate forecasts on sub-seasonal time scales would have immense societal value. For instance, sub-seasonal forecasts of temperature and precipitation could be used to assist farmers in determining planting dates, irrigation needs, expected market conditions, anticipating pests and disease, and assessing the need for insurance. Emergency and disaster-relief supplies can take weeks or months to pre-stage, so skillful forecasts of areas that are likely to experience extreme weather a few weeks in advance could save lives. More generally, skillful sub-seasonal forecasts also would have beneficial impacts on agricultural productivity, hydrology and water resource management, transportation and aviation systems, emergency planning for extreme climate such as Atlantic hurricanes and midwestern tornadoes, among others [37, 29]. Inaccurate spatial-temporal forecasts associated with extreme weather events and associated disaster relief planning can be expensive both in terms of loss of human lives as well as financial impact. On a more steady state basis, water resource management and planning agricultural activities can be made considerably more precise and cost effective with skillful sub-seasonal climate forecasts.

Acknowledgement

The research was supported by NSF grants OAC-1934634, IIS-1908104, IIS-1563950, IIS-1447566, IIS-1447574, IIS-1422557, CCF-1451986. The authors would like to acknowledge the computing support from the Minnesota Supercomputing Institute (MSI) at the University of Minnesota.

References

[1] Hamada S Badr, Benjamin F Zaitchik, and Seth D Guikema. Application of statistical models to the prediction of seasonal rainfall anomalies over the sahel. *Journal of Applied meteorology and climatology*, 53(3):614–636, 2014.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Conference Track Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

[3] Anthony G Barnston and Robert E Livezey. Classification, seasonality and persistence of low-frequency atmospheric circulation patterns. *Monthly weather review*, 115(6):1083–1126, 1987.

[4] Anthony G Barnston, Michael K Tippett, Michelle L L'Heureux, Shuhua Li, and David G DeWitt. Skill of real-time seasonal enso model predictions during 2002–11: Is our capability increasing? *Bulletin of the American Meteorological Society*, 93(5):631–651, 2012.

[5] Lisette Martine Braman, Maarten Krispijn van Aalst, Simon J Mason, Pablo Suarez, Youcef Ait-Chellouche, and Arame Tall. Climate forecasts in disaster management: Red cross flood operations in west africa, 2008. *Disasters*, 37(1):144–164, 2013.

[6] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 785–794, 2016.

[7] Antonio S. Cofino, Rafael Cano, Carmen Sordo, and José M. Gutiérrez. Bayesian networks for probabilistic weather prediction. In *In Proceedings of the 15th Eureopean Conference on Artificial Intelligence (ECAI)*, pages 695–699, 2002.

[8] Judah Cohen, Dim Coumou, Jessica Hwang, Lester Mackey, Paulo Orenstein, Sonja Totz, and Eli Tziperman. S2s reboot: An argument for greater inclusion of machine learning in subseasonal to seasonal forecasts. *Wiley Interdisciplinary Reviews: Climate Change*, 10(2):e00567, 2019.

[9] Erin Coughlan de Perez and Simon J Mason. Climate information for humanitarian agencies: Some basic principles. *Earth Perspectives*, 1(1):11, 2014.

[10] Timothy DelSole and Arindam Banerjee. Statistical seasonal prediction based on regularized regression. *Journal of Climate*, 30(4):1345–1361, 2017.

[11] Timothy Delsole and Michael Tippett. Predictability in a changing climate. *Climate Dynamics*, 10 2017.

[12] Yun Fan and Huug van den Dool. Climate prediction center global monthly soil moisture data set at 0.5 resolution for 1948 to present. *Journal of Geophysical Research: Atmospheres*, 109(D10), 2004.

[13] Yun Fan and Huug Van den Dool. A global monthly land surface air temperature analysis for 1948–present. *Journal of Geophysical Research: Atmospheres*, 113(D1), 2008.

[14] Frederik Nebeker. *Calculating the weather: Meteorology in the 20th century.* Elsevier, 1995.

[15] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[16] Ken-ichi Funahashi and Yuichi Nakamura. Approximation of dynamical systems by continuous time recurrent neural networks. *Neural networks*, 6(6):801–806, 1993.

[17] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural Computation*, 12(10):2451–2471, 2000.

[18] Andre R Goncalves, Arindam Banerjee, and Fernando J Von Zuben. Spatial projection of multiple climate variables using hierarchical multitask learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[19] Andre R Goncalves, Puja Das, Soumyadeep Chatterjee, Vidyashankar Sivakumar, Fernando J Von Zuben, and Arindam Banerjee. Multi-task sparse structure learning. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 451–460, 2014.

[20] Andre R Goncalves, Fernando J Von Zuben, and Arindam Banerjee. Multi-task sparse structure learning with gaussian copula models. *The Journal of Machine Learning Research*, 17(1):1205–1234, 2016.

[21] Aditya Grover, Ashish Kapoor, and Eric Horvitz. A deep hybrid model for weather forecasting. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 379–386. ACM, 2015.

[22] Yoo-Geun Ham, Jeong-Hwan Kim, and Jing-Jia Luo. Deep learning for multi-year enso forecasts. *Nature*, 573(7775):568–572, 2019.

[23] Sijie He, Xinyan Li, Vidyashankar Sivakumar, and Arindam Banerjee. Interpretable predictive modeling for climate variables with weighted lasso. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1385–1392, 2019.

[24] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[25] Jin Huang, Huug M van den Dool, and Konstantine P Georgarakos. Analysis of model-calculated soil moisture over the united states (1931–1993) and applications to long-range temperature forecasts. *Journal of Climate*, 9(6):1350–1362, 1996.

[26] Jessica Hwang, Paulo Orenstein, Judah Cohen, Karl Pfeiffer, and Lester Mackey. Improving subseasonal forecasting in the western us with machine learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2325–2335. ACM, 2019.

[27] Ali Jalali, Pradeep Ravikumar, and Sujay Sanghavi. A dirty model for multiple sparse regression. *IEEE Transactions on Information Theory*, 59(12):7947–7968, 2013.

[28] E. Kalnay, M. Kanamitsu, R. Kistler, W. Collins, D. Deaven, L. Gandin, M. Iredell, S. Saha, G. White, J. Woollen, Y. Zhu, M. Chelliah, W. Ebisuzaki, W. Higgins, J. Janowiak, K. C. Mo, C. Ropelewski, J. Wang, A. Leetmaa, R. Reynolds, Roy Jenne, and Dennis Joseph. The ncep/ncar 40-year reanalysis project. *Bulletin of the American Meteorological Society*, 77(3):437–472, 1996.

[29] Toni Klemm and Renee A McPherson. The development of seasonal climate forecasting for agricultural producers. *Agricultural and forest meteorology*, 232:384–399, 2017.

[30] Hans R Kunsch. The jackknife and the bootstrap for general stationary observations. *The annals of Statistics*, pages 1217–1241, 1989.

[31] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[32] Stan Lipovetsky and Michael Conklin. Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, 17(4):319–330, 2001.

[33] Andrew C Lorenc. Analysis methods for numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 112(474):1177–1194, 1986.

[34] Edward N Lorenz. Empirical orthogonal functions and statistical weather prediction. 1956.

[35] National Academies of Sciences. *Next generation earth system prediction: strategies for subseasonal to seasonal forecasts*. National Academies Press, 2016.

[36] National Research Council. *Assessment of intraseasonal to interannual climate prediction and predictability*. National Academies Press, 2010.

[37] JW Pomeroy, DM Gray, NR Hedstrom, and JR Janowicz. Prediction of seasonal snow accumulation in cold climate forecasts. *Hydrological Processes*, 16(18):3543–3558, 2002.

[38] Y Radhika and M Shashi. Atmospheric temperature prediction using support vector machines. *International journal of computer theory and engineering*, 1(1):55, 2009.

[39] Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, et al. Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743):195–204, 2019.

[40] David N Reshef, Yakir A Reshef, Hilary K Finucane, Sharon R Grossman, Gilean McVean, Peter J Turnbaugh, Eric S Lander, Michael Mitzenmacher, and Pardis C Sabeti. Detecting novel associations in large data sets. *science*, 334(6062):1518–1524, 2011.

[41] Richard W Reynolds, Thomas M Smith, Chunying Liu, Dudley B Chelton, Kenneth S Casey, and Michael G Schlax. Daily high-resolution-blended analyses for sea surface temperature. *Journal of Climate*, 20(22):5473–5496, 2007.

[42] Tapio Schneider, Shiwei Lan, Andrew Stuart, and Joao Teixeira. Earth system modeling 2.0: A blueprint for models that learn from observations and targeted high-resolution simulations. *Geophysical Research Letters*, 44(24):12–396, 2017.

[43] Xingjian Shi, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Deep learning for precipitation nowcasting: A benchmark and a new model. In *Advances in neural information processing systems*, pages 5617–5627, 2017.

[44] A. J. Simmons and A. Hollingsworth. Some aspects of the improvement in skill of numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 128(580):647–677, 2002.

[45] Adrian J Simmons and Anthony Hollingsworth. Some aspects of the improvement in skill of numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 128(580):647–677, 2002.

[46] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852, 2015.

[47] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*, 2012.

[48] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[49] The National Oceanic and Atmospheric Administration. *Subseasonal and Seasonal Forecasting Innovation: Plans for the Twenty-First Century.* Annotated outline of NOAA's draft subseasonal and seasonal (S2S) forecasting report, 2018.

[50] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society,*, pages 267–288, 1996.

[51] HM Van den Dool, S Saha, and AAke Johansson. Empirical orthogonal teleconnections. *Journal of Climate*, 13(8):1421–1435, 2000.

[52] Huug Van den Dool, Principal Scientist Cpc, and Huug Van Den Dool. *Empirical methods in short-term climate prediction.* Oxford University Press, 2007.

[53] Huug Van den Dool, Jin Huang, and Yun Fan. Performance and analysis of the constructed analogue method applied to us soil moisture over 1981–2001. *Journal of Geophysical Research: Atmospheres*, 108(D16), 2003.

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[55] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542, 2015.

[56] Frédéric Vitart, Andrew W Robertson, and David LT Anderson. Subseasonal to seasonal prediction project: Bridging the gap between weather and climate. *Bulletin of the World Meteorological Organization*, 61(23), 2012.

[57] Hans Von Storch and Francis W Zwiers. *Statistical analysis in climate research.* Cambridge university press, 2001.

[58] Larry Wasserman. *All of statistics: a concise course in statistical inference.* Springer Science & Business Media, 2013.

[59] Sanford Weisberg. *Applied linear regression*, volume 528. John Wiley & Sons, 2005.

[60] Matthew C. Wheeler and Harry H. Hendon. An all-season real-time multivariate mjo index: Development of an index for monitoring and prediction. *Monthly Weather Review*, 132(8):1917–1932, 2004.

[61] Shi Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.

[62] Brian G Zimmerman, Daniel J Vimont, and Paul J Block. Utilizing the state of enso as a means for season-ahead predictor selection. *Water resources research*, 52(5):3761–3774, 2016.

A Difficulty of the Problem

A.1 Dependence between historical data and forecasting target

In section 3, the dependence between the most recent historical data (the normalized average temperature of week -2 & -1) and the forecasting target (the normalized average temperature of week 3 & 4) is measured by maximum information coefficient (MIC). Here we show the results measured by Pearson correlation coefficient [58], and Spearman's rank correlation coefficient [58] (Figure 6). Small values ($\leq$0.2) of Pearson correlation and Spearman's rank correlation at a majority of locations, which verify that there is little information shared between the most recent date and the forecasting target, once again, demonstrate how difficult SSF is.

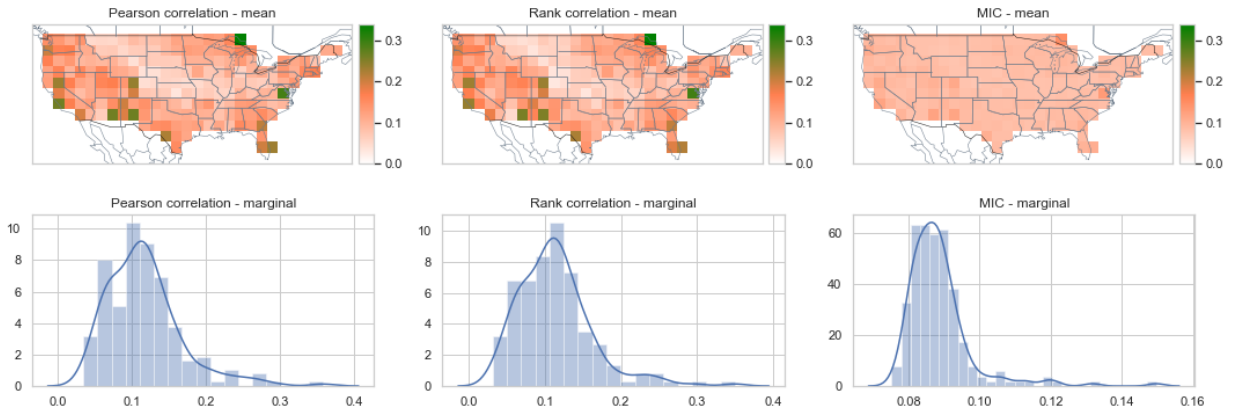


Figure 6: Pearson correlation, Spearman's rank correlation and MIC between 2m temperature of week -2 & -1 and week 3 & 4. Small values ($\leq$0.2) of Pearson correlation and Spearman's rank correlation at a majority of locations verify the fact, as we illustrate in the main paper using MIC, that there is little information shared between the most recent date and the forecasting target.

A.2 Relative R^2

In the main paper, we introduce cosine similarity, which is widely used in weather prediction evaluation, as an evaluation metric. Here we formally define the other evaluation metric, namely relative R^2 as

$$\text{Relative } R^2 = 1 - \text{Relative MSE} = 1 - \frac{\sum_{i=1}^{n}(\mathbf{y}_i^* - \hat{\mathbf{y}}_i)^2}{\sum_{i=1}^{n}(\mathbf{y}_i^* - \bar{\mathbf{y}}_{\text{train}})^2} \ , \tag{1}$$

where $\hat{\mathbf{y}}$ denotes a vector of predicted values, and $\mathbf{y}^*$ be the corresponding ground truth. We use relative R^2 to evaluate the relative predictive skill of a given prediction $\hat{y}$ compared to the best constant predictor $\bar{y}_{\text{train}}$, the long-term average of target variable at each date and each target location computed from training set. A model which achieves a positive relative R^2 is, at least, able to predict the sign of y^* accurately. The results of temporal and spatial relative R^2 over the US mainland of ML models discussed in section 4 are shown in Table 3 and Figure 8 respectively.

B Data and experimental setup

B.1 Data sources

The data described in Table 1 were downloaded from the following sources:

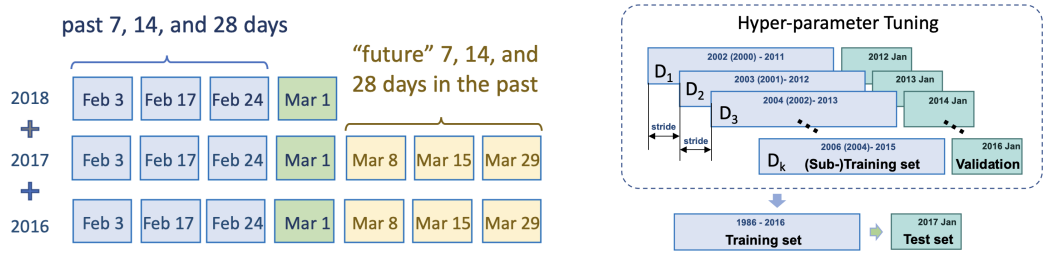
- Temperature (tmp2m): `https://www.esrl.noaa.gov/psd/data/gridded/data.cpc.globaltemp.html`
- Soil moisture (sm): `https://www.esrl.noaa.gov/psd/data/gridded/data.cpcsoil.html`
- Sea surface temperature (sst): `https://www.ncdc.noaa.gov/oisst`
- Relative humidity (rhum), sea level pressure (slp), and geopotential height (hgt): `ftp://ftp.cdc.noaa.gov/Datasets/ncep.reanalysis/surface/`
- Multivariate ENSO index (MEI): `https://psl.noaa.gov/enso/mei/`
- Niño indices: `https://www.cpc.ncep.noaa.gov/data/indices/wksst8110.for`
- North Atlantic Oscillation (NAO) index: `ftp://ftp.cpc.ncep.noaa.gov/cwlinks/norm.daily.nao.index.b500101.current.ascii`
- Madden Julian Oscillation (MJO) phase & amplitude: `http://www.bom.gov.au/climate/mjo/graphics/rmm.74toRealtime.txt`

B.2 PCA prepossessing

As mentioned in section 5 of the main paper, one way for feature extraction is to apply PCA to spatial-temporal variables. To do so, let's consider sst of Pacific ocean as an example. Daily sst of Pacific ocean is originally stored in a matrix, of which each element represents the sea surface temperature at each grid point of Pacific ocean. The covariance matrix can be computed by flattening each matrix into a 1-D vector, viewing each element in the matrix as a feature and each date as one observation. Such covariance matrix captures spatial connection among grid points of Pacific ocean. By considering all dates from 1986 to 2016, we can extract the top 10 principal components (PCs) as features based on PC loadings computed from the corresponding covariance.

B.3 Feature set construction

To better utilize historical information, we construct a sequential feature set by including not only covariates of the target date, but also covariates of the 7^{th}, 14^{th}, and 28^{th} day previous from the target date, as well as the day of the year of the target date in the past 2 years and both the historical past and future dates around the day of the year of the target date in the past 2 years. Such selection of historical dates mainly bases on the temporal correlation. Figure 7(a) provides a detailed example on how to construct feature set for Mar 1, 2018: we concatenate covariates from Mar. 1 in 2018, 2017, and 2016, their corresponding 7^{th}, 14^{th}, and 28^{th} days in the past, and 7^{th}, 14^{th}, and 28^{th} "future" days in 2017 and 2016. In total, we include $H = 18$ historical days in our feature set for each date.



(a) Sequential feature set for Mar 1, 2018

(b) Evaluation pipeline

Figure 7: (a) Sequential feature set: to construct feature set at Mar. 1, 2018, we concatenate covariates from Mar. 1 in 2018, 2017, and 2016, their corresponding 7^{th}, 14^{th}, and 28^{th} days in the past, and 7^{th}, 14^{th}, and 28^{th} "future" days in 2017 and 2016. (b) Evaluation pipeline: to test SSF in Jan 2017, the training set covers historical 30 year ends at Dec 4, 2016 (the last available date). 5 validation sets include dates from each Jan between 2012 to 2016, with the corresponding training sets generated by applying a moving window of 10 years and a stride of 365 days on data start from 2000.

B.4 Evaluation pipeline

Predictive models are created independently for each month in 2017 and 2018. To mimic a live forecasting system, we generate 105 test dates during 2017-2018, one for each week, and group them into 24 test sets by their month of the year. Given a test set, our evaluation pipeline consists of two parts (Figure 7(b)):

- "5-fold" training-validation pairs for hyper-parameter tuning, based on a "sliding-window" strategy designed for time-series data. Each validation set uses the data from the same month of the year as the test set. For instance, if the test set is Jan 2017, the corresponding 5 validation sets are Jan 2012, Jan 2013, Jan 2014, Jan 2015, and Jan 2016 respectively. Each validation set corresponds to a training set containing 10 years of data and ending 28 days before the first date in the validation set. Specifically, if the validation set starting from Jan 1, 2016, the training set is from Dec 4, 2005 to Dec 4, 2015. Such construction is equivalent to apply a sliding-window of 10-year with a stride of 365 days on data from 2002.
- The training-test pair, where the training set, including 30-year data in the past, ends 28 days before the first date in the test set. For example, to test SSF in Jan 2017, i.e., Jan 1, Jan 8, Jan 15, Jan 22, and Jan 29, the training set starts from Dec 4, 1986 and ends at Dec 4, 2016, which is the 28^{th} day before Jan 1, and the last date we have the ground truth for the target variable.

C Additional Results

C.1 Temporal and spatial results of relative R^2

Table 3 lists the mean, the median, the 0.25 quantile, the 0.75 quantile, and their corresponding standard errors of relative R^2 for all models. A positive relative R^2 indicates a model can at least predict the sign of the target variable correctly. Again, XGBoost achieves the highest predictive skill

Table 3: Comparison of relative R^2 of tmp2m forecasting for test sets over 2017-2018. A positive relative R^2 indicates a model predicting the sign of the target variable correctly. XGBoost achieves the highest relative R^2.

Model	Mean(se)	Median (se)	0.25 quantile (se)	0.75 quantile (se)
Temporally Global Set				
XGBoost - one day	**0.0760(0.03)**	**0.0974(0.03)**	**-0.0449(0.03)**	**0.2434(0.03)**
Lasso - one day	0.0552(0.02)	0.0321(0.02)	-0.0309(0.01)	0.1295(0.02)
Encoder (LSTM)-Decoder (FNN)	**-0.0353 (0.05)**	**0.0596(0.05)**	**-0.2409 (0.06)**	**0.2426 (0.05)**
FNN	-0.5777(0.29)	-0.0183(0.15)	-0.0794(0.13)	0.0213(0.13)
CNN	-0.0564(0.03)	0.0284(0.02)	-0.0266(0.02)	0.0570(0.02)
CNN-LSTM	-0.1164(0.05)	0.0263(0.03)	-0.0862(0.03)	0.0698(0.03)
LS with NAO & all nino - daily	**0.0418(0.01)**	**0.0535(0.01)**	**-0.0078(0.01)**	**0.0949(0.01)**
Damped persistence	0.0266(0.01)	0.0414(0.02)	-0.0542(0.02)	0.1354(0.02)
MultiLLR	-0.0571 (0.02)	0.0034 (0.02)	-0.1156 (0.03)	0.0797 (0.02)
AutoKNN	0.0181 (0.01)	0.0260 (0.02)	-0.0531 (0.02)	0.1041 (0.01)
Temporally Local Set				
XGBoost - one day	-0.0337(0.03)	0.0396(0.03)	-0.1310(0.04)	0.1873(0.03)
Lasso - one day	-0.0028(0.02)	0.0327(0.02)	-0.0613(0.02)	0.0996(0.02)
Encoder (LSTM)-Decoder (FNN)	-0.2333 (0.06)	-0.1116 (0.06)	-0.4694 (0.09)	0.1808 (0.06)

in terms of both the mean and the median, demonstrating its predictive power. Linear regression, like Lasso, with a proper feature set has good predictive performance. Both XGBoost and Lasso have larger positive relative R^2 in terms of the mean, and can still outperform climatology and two climate baseline models, i.e., LS with NAO & Niño, and damped persistence. Even though Encoder (LSTM)-Decoder (FNN) has a slightly negative mean relative R^2, it has the second largest median and 0.75 quantile among all models, showing its potential for further improvement.

Figure 8 shows the spatial relative R^2 of all methods. XGBoost and Lasso are able to achieve positive relative R^2 for most of the target locations. Encoder (LSTM)-Decoder (FNN) shows better predictive skill over the southern US compared to other regions. MultiLLR and AutoKNN manages to obtain non-negative relative R^2 for the coastal area in the western US but their predictive performance drops in the rest of locations. All other baseline methods struggle to reach positive relative R^2 for most of the target locations.

C.2 Analysis on feature importance

To emphasis the importance of the land-based covariates, e.g., soil moisture and the ocean-based covariates, e.g., NAO and Niño indices, we compare the prediction performance among (1) the model trained with all covariates, (2) the model trained without soil moisture, and (3) the model trained without NAO and Niño indices (Table 4 and Table 5). Most models experience a performance deterioration when we exclude certain "important" covariates.

C.3 The influence of feature sequence length

We compare the prediction performance under 3 different settings, referred to as "one day", "four days", and "all days" respectively. For feature set construction, "one day" includes covariates at the target date only, "four days" also covers the 7^{th}, 14^{th}, and 28^{th} days previous to the target date, and

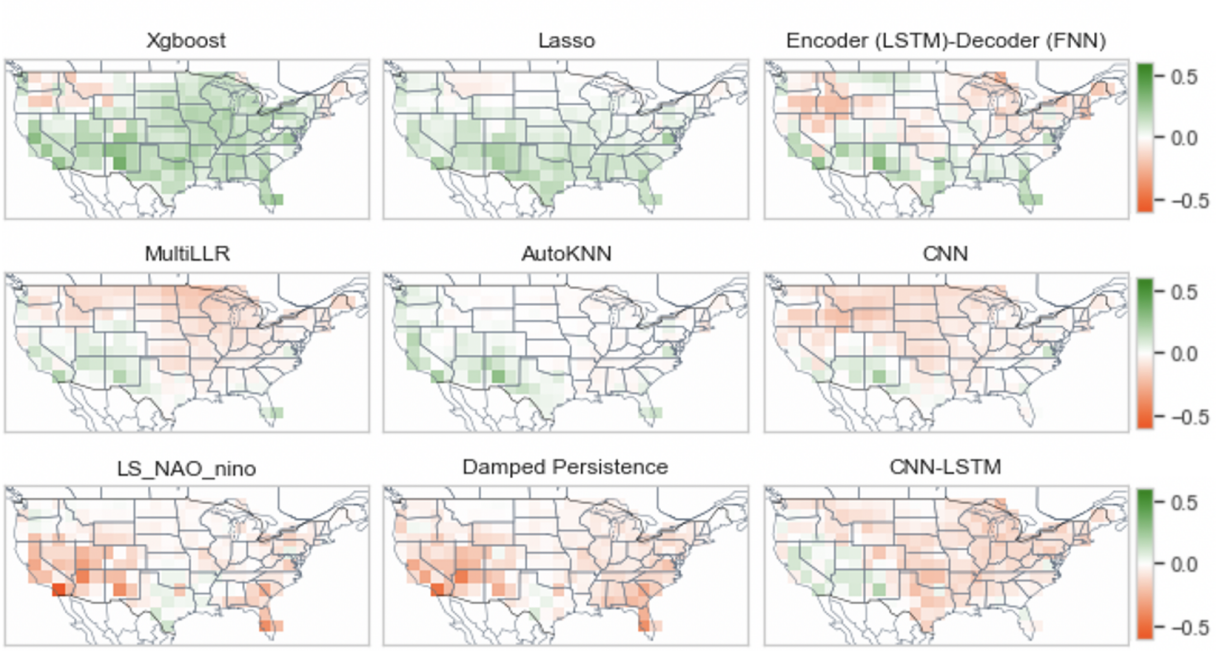


Figure 8: Temporal relative R^2 over the US mainland of ML models discussed in section 4 for temperature prediction over 2017-2018. Large positive values (green) closer to 1 indicates better predictive skills.

"all days" uses the exact feature sequence we use for LSTM-based models. Comparison of predictive skills under each setting, measured by both cosine similarity and relative R^2, can be found in Table 6 and Table 7. Both XGBoost and Lasso enjoy a performance boost using "one day" values. Especially for XGBoost, the performance of "one day" is approximately 50% better than using "all days". A possible explanation for such performance degradation as we increase the feature sequence length is that both models weight covariates from different dates exactly the same without considering temporal information, thus more noise has been introduced.

C.4 Discussion on deep learning models

Results of DL models. Table 8 and Table 9 compare the predictive skills of 5 DL models discussed in section 6.3, measured by both cosine similarity and relative R^2. Significant improvements can been observed as we evolve from the standard Encoder (LSTM)-Decoder (LSTM), to Encoder (LSTM)-Decoder (FNN)-last step, where "last step" indicates that FNN Decoder only uses the last step of the output sequence from LSTM Encoder, and finally to Encoder (LSTM)-Decoder (FNN) with FNN Decoder uses every step of the output sequence from LSTM Encoder.

One issue with Encoder (LSTM)-Decoder (FNN) is that the input features are shared by all target locations, which requires the model to identify the useful information for each locations without any help from the input.

Autoregressive (AR) component. Currently, the Encoder(LSTM)-Decoder(FNN) clearly considers climate covariates on a global scale, which are shared by all target locations. Nevertheless,

Table 4: Comparison of cosine similarity of tmp2m forecasting for test sets over 2017-2018 using different feature set. Excluding soil moisture or climate indices (NAO & Niño) leads to a deterioration in the predictive performance.

Model	Mean(se)	Median (se)	0.25 quantile (se)	0.75 quantile (se)
XGBoost - one day	**0.3044(0.03)**	**0.3447(0.05)**	**0.0252(0.05)**	**0.5905(0.04)**
XGBoost - one day (w/o soil moisture)	0.2685(0.03)	0.2797(0.05)	0.0703(0.04)	0.5492(0.05)
XGBoost - one day (w/o nao & all nino)	0.2081(0.03)	0.1640(0.05)	-0.0588(0.04)	0.5246(0.05)
Lasso - one day	0.2499(0.04)	0.2554(0.06)	-0.0224(0.05)	0.5604(0.06)
Lasso - one day (w/o soil moisture)	0.2638(0.04)	0.2912(0.05)	0.0032(0.06)	0.5655(0.05)
Lasso - one day (w/o nao & all nino)	0.1956(0.04)	0.2573(0.07)	-0.1657(0.06)	0.5533(0.05)
Encoder (LSTM)-Decoder (FNN)	0.2616 (0.04)	0.2995 (0.07)	-0.0719 (0.06)	0.6310 (0.05)
Encoder (LSTM)-Decoder (FNN)(w/o soil moisture)	0.2157 (0.04)	0.2909 (0.07)	-0.1106 (0.07)	0.5443 (0.07)
Encoder (LSTM)-Decoder (FNN)(w/o nao & all nino)	0.2236 (0.04)	0.2395 (0.06)	-0.1527 (0.07)	0.5989 (0.06)

Table 5: Comparison of relative R^2 of tmp2m forecasting for test sets over 2017-2018. Excluding soil moisture or climate indices (NAO & Niño) leads to a smaller or even negative relative R^2, showing that it becomes harder for the model to predict the sign of the target variable correctly.

Model	Mean(se)	Median (se)	0.25 quantile (se)	0.75 quantile (se)
XGBoost - one day	**0.0760(0.03)**	**0.0974(0.03)**	**-0.0449(0.03)**	**0.2434(0.03)**
XGBoost - one day (w/o soil moisture)	0.0370(0.03)	0.0322(0.03)	-0.0564(0.03)	0.2225(0.03)
XGBoost - one day (w/o nao & all nino)	-0.0161(0.03)	-0.0079(0.04)	-0.1618(0.03)	0.2426(0.04)
Lasso - one day	0.0552(0.02)	0.0321(0.02)	-0.0309(0.01)	0.1295(0.02)
Lasso - one day (w/o soil moisture)	-0.0161(0.03)	-0.0079(0.04)	-0.1618(0.03)	0.2426(0.04)
Lasso - one day (w/o nao & all nino)	0.0003(0.02)	0.0457(0.02)	-0.1113(0.03)	0.1641(0.02)
Encoder (LSTM)-Decoder (FNN)	-0.0353 (0.05)	0.0596(0.05)	-0.2409 (0.06)	0.2426 (0.05)
Encoder (LSTM)-Decoder (FNN)(w/o soil moisture)	-0.1083 (0.05)	0.0314 (0.05)	-0.3022 (0.08)	0.2252 (0.05)
Encoder (LSTM)-Decoder (FNN)(w/o nao & all nino)	-0.0802 (0.04)	0.0124 (0.05)	-0.3032 (0.06)	0.2446 (0.05)

SSF depends on not only global climate condition but also local weather change. Therefore, we seek a way to improve the model by adding an autoregressive (AR) component to capture the "local" information from historical data. We consider two variants of Encoder (LSTM)-Decoder (FNN). The first variant contains an AR component with the input as historical temperature at each target location, denoted as Encoder (LSTM)-Decoder (FNN)+AR. The second one includes both historical temperature and historical temporal climate variables, i.e., climate indices, as input features, denoted as Encoder (LSTM)-Decoder (FNN)+AR (CI). For both models, the final forecast is computed as a linear combination of the prediction from Encoder (LSTM)-Decoder (FNN) and the prediction from AR component for each location. Unexpectedly, as shown in Table 8 and Table 9, simply adding the AR component to our Encoder(LSTM)-Decoder(FNN) does not help the model to perform better. However, we believe there is a better way to involve local information, and such modification is a promising direction that worth investigation in the future.

Table 6: Comparison of spatial cosine similarity for tmp2m forecasting over 2017-2018 using various length of feature sequence. Including longer historical sequence leads to a deterioration in the predictive performance of XGBoost and Lasso.

Model	Mean(se)	Median (se)	0.25 quantile (se)	0.75 quantile (se)
XGBoost - all days	0.2080(0.03)	0.1582(0.05)	-0.0466(0.05)	0.5383(0.05)
XGBoost - four days	0.2433(0.03)	0.2203(0.05)	0.0561(0.04)	0.5168(0.06)
XGBoost - one day	**0.3044(0.03)**	**0.3447(0.05)**	**0.0252(0.05)**	**0.5905(0.04**)
Lasso - all days	0.2160(0.04)	0.2258(0.07)	-0.1381(0.06)	0.5384(0.06)
Lasso - four days	0.2247(0.04)	0.1952(0.07)	0.0572(0.06)	-0.5700(0.06)
Lasso - one day	**0.2499(0.04)**	**0.2554(0.06)**	**-0.0224(0.05**)	**0.5604(0.06)**

Table 7: Comparison of relative R^2 (with training set mean) for tmp2m prediction for test set over 2017-2019 using different length of feature sequence. Including longer historical sequence leads to a smaller or even negative relative R^2 for both XGBoost and Lasso.

Model	Mean(se)	Median (se)	0.25 quantile (se)	0.75 quantile (se)
XGBoost - all days	-0.0200(0.03)	-0.0010(0.04)	-0.1499(0.04)	0.2304(0.04)
XGBoost - four days	0.0242(0.03)	0.0193(0.03)	-0.0786(0.03)	0.1882(0.04)
XGBoost - one day	**0.0760(0.03)**	**0.0974(0.03)**	**-0.0449(0.03)**	**0.2434(0.03)**
Lasso - all days	-0.0167(0.03)	0.0367(0.03)	-0.0639(0.02)	0.1588(0.03)
Lasso - four days	0.0518(0.02)	0.0266(0.02)	-0.0542(0.02)	0.1653(0.03)
Lasso - one day	**0.0552(0.02**)	**0.0321(0.02)**	**-0.0309(0.01)**	**0.1295(0.02)**

Table 8: Comparison of cosine similarity of tmp2m forecasting for test sets over 2017-2018 using different deep learning architectures.

Model	Mean(se)	Median (se)	0.25 quantile (se)	0.75 quantile (se)
Encoder (LSTM)-Decoder (LSTM)	0.0740(0.03)	0.0358(0.04)	-0.1569(0.03)	0.2584(0.04)
Encoder (LSTM)-Decoder (FNN)-last step	0.1614 (0.05)	0.2061 (0.08)	-0.2590 (0.08)	0.5720 (0.08)
Encoder (LSTM)-Decoder (FNN)	**0.2616 (0.04**)	**0.2995 (0.07)**	**-0.0719 (0.06)**	**0.6310 (0.05)**
Encoder (LSTM)-Decoder (FNN)+AR	0.1733 (0.04)	0.1922 (0.06)	-0.0863 (0.07)	0.5225 (0.06)
Encoder (LSTM)-Decoder (FNN)+AR (CI)	0.1852 (0.04)	0.1986 (0.05)	-0.0838 (0.06)	0.5164 (0.05)

Table 9: Comparison of relative R^2 of tmp2m forecasting for test sets over 2017-2018. A positive relative R^2 indicates a model predicting the sign of the target variable correctly.

Model	Mean(se)	Median (se)	0.25 quantile (se)	0.75 quantile (se)
Encoder (LSTM)-Decoder (LSTM)	-0.3947(0.05)	-0.2999(0.05)	-0.6606(0.08)	-0.0537(0.05)
Encoder (LSTM)-Decoder (FNN)-last step	-0.1709 (0.06)	0.0217 (0.06)	-0.4569 (0.11)	0.2278 (0.06)
Encoder (LSTM)-Decoder (FNN)	**-0.0353 (0.05)**	**0.0596(0.05)**	**-0.2409 (0.06)**	**0.2426 (0.05)**
Encoder (LSTM)-Decoder (FNN)+AR	-0.0414 (0.04)	-0.0041 (0.05)	-0.3027 (0.07)	0.2309 (0.05))
Encoder (LSTM)-Decoder (FNN)+AR (CI)	-0.0563 (0.03)	-0.0380 (0.05)	-0.2365 (0.05)	0.1951 (0.04)