Long-Range Temperature Forecasting Correction Techniques Using Machine Learning

By

Alexander G. Grossman

B.S. Electrical Engineering and Computer Science

Massachusetts Institute of Technology, 2020

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

AT THE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

MAY 2020

Signature of Author: _____

Department of Electrical Engineering and Computer Science
May 18, 2020

Certified by: _____

Srinivas Ravela
Principle Research Scientist, EAPS
Thesis Supervisor

Accepted by: _____

Katrina LaCurts
Chair
Master of Engineering Thesis Committee

Long-Range Temperature Forecasting Correction Techniques Using Machine Learning

By

Alexander G. Grossman

Submitted to the Department of Electrical Engineering and Computer Science on May 18, 2020 in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

We present solutions to four problems emerging in data-driven long-range weather prediction that were explored as part of an M.Eng Thesis. These problems are related to long-range prediction using a network of observing stations and climate indicators. The first problem relates to the correction of phase error in long-term temperature forecasts. The second problem involves the task of using correlated observed and proxy signals to update each other to improve forecasting accuracy. The third problem relates to the use of deep learning in the problem of predicting the future value of near oscillators. The fourth problem relates to the discovery of new, finer scale oscillation signals using Representation Learning based Dimensionality Reduction techniques. Together, our proposed solutions enable the use of inference and learning for data-driven long-range weather forecasting using context from the global climate system.

Thesis Supervisor: Sai Ravela
Title: Principal Research Scientist, EAPS

Contents

Chapter 1

Introduction

Long-range weather forecasts are applicable in many different areas of research and planning, ranging from estimating the long-term likelihoods of droughts and floods to designing the energy grid to estimating long-term oil and gas demand [30]. It is an important problem to be researched if we want to be able to make infrastructure changes to deal with the future effects of climate change.

The problem of long-range weather forecasting, however, is well-known to be difficult, as many meteorologists believe that weather can only be predicted at long-horizons with very low accuracy and that the predictions should be considered unreliable [31]. There a couple of reasons for this. Most weather forecasts are from numerical weather prediction models [31]. These aim to model dynamical equations and simulate them from a set of initial conditions [29]. Errors in initial conditions grow exponentially and contribute to the overall loss of predictability [31]. Further, numerical models must often approximate the governing equations so that key elements of the dynamics are either approximated or parameterized.

A consequence is that data-driven techniques remain important. Whether used for data assimilation, calibration, post-processing model outputs, e.g. model output statistics, or using data to derive data-driven models, e.g. in hurricane prediction, they are often required to "make models work."

Data-driven models are not known by themselves to have long-range predictability.

Indeed, nowcasting research indicates that even at intermediate horizons (e.g. a few hours for some problems) numerical prediction may be better. Yet, the costs of such numerical prediction are high, and there is a fundamental curiosity of how far a data-driven approach can be taken.

Work in the development of Graphical models to incorporate context from multiple sources, the emergence of deep learning algorithms to model nonlinearities, and concomitant approaches for dimensionality reduction give new impetus to a data-driven framework that is the subject of this thesis. Many successes have been presented that lead us back to an examination of just what data science can do. Our thesis addresses a few key problems to improve the performance of data-driven approaches.

A key difficulty in long-range prediction is evident when one observes that signals (e.g. temperature) have both oscillation and trend. As the forecast error grows and compounds, the errors no longer are restricted in amplitude but accumulate in timing (or "phase error"). This is a key reason for the long-term loss of predictability [34]. Our idea to resolve this issue performs phase correction using climate indicators, which we interpret as oscillators providing synchronization mechanisms. The phase that we are referring to in this case is referred to as the phase of the analytic signal, which is found using the Hilbert Transform, which is a signal that is made up of the original signal as its real part and its Hilbert transform as the imaginary part. When we discuss the phase of a signal going forward, we are specifically referring to the phase of the analytic signal.

Dimensionality emerges as a second key difficulty. High dimensionality is often an issue in numerical model fields, but low-dimensionality is typical when using individual sensor data alone. Inferences often demand additional constraints, which we posit observing system networks and global context can deliver, short of using numerical models themselves. Thus, the second idea this thesis explores is to explicitly capture global context with a spatial network model using a bayesian graphical model.

Important to our research direction is context obtained from climate indicators. Cli-

matologists have observed naturally occurring oscillations in the Earth's oceans and atmosphere over different chronological periods and geographical scales. The observed oscillations are then represented as one-dimensional signals by the National Oceanic and Atmospheric Administration, which are standardized values that represent the phase of each of these natural oscillators, or more clearly, where these oscillators are in their natural cycles at any moment. The data is calculated using various techniques, but most methods use some kind of proxy measurements for the phenomena such as averages of sea surface temperature or air pressure in different areas [25]. Some of these oscillation signals (indices) include the El-Niño Southern Oscillation (ENSO), Pacific Decadal Oscillation (PDO), North Atlantic Oscillation (NAO), and Arctic Oscillation (AO), as well as others to be mentioned later such as the East Atlantic Pattern.

A key hypothesis in this thesis is that climate indicators provide context for local prediction, the context, taking the form of a predictive model. The model changes from region to region. Nevertheless, together, indicators provide sufficient climate state context to constrain local predictions to an effective degree. The use of oscillations for long-term weather predictions are well-established in the relevant scientific literature [31]. Our results validate this hypothesis, but additionally, their role in phase correction, which has not hitherto been reported is shown.

Throughout this MEng Thesis, we will be presenting three separate items of analysis as three separate, self-contained chapters. We will discuss the synergies of the chapters at the end; as well as our ideas on how they can be best used together to improve long-term weather forecasts. We will show that long-term forecasts cannot be done using local data alone, and context is necessary to reduce the error due to a lack of information. The first part of this context is time, which global oscillators provide. The second part of this is space, which graphical models provide. Subsequently, we focus on deriving new oscillators as that will reduce the preceding two sources of error.

The three chapters focus on the following topics: (1) reducing the phase error and exploding error of long-term forecasts through phase correction; (2) using oscillators and site-models to improve predictions from context; and (3) using SST grids

to derive new explanatory "oscillators" for better context, as well as a method for deriving new oscillators borrowing from representation learning, a method that can be used to try to understand the underlying distributions of data using neural networks. Throughout this chapter, we will discuss the problem statements for each of these inquiries, and the data sources we use.

1.1 Problems Solved

1.1.1 Phase Correction

Why are we so interested in these oscillation indices that we just discussed? In other related work on long-range weather forecasting, we have found that the phases of climate oscillation signals can be nonlinearly composed, with almost perfect precision ($\approx 0.1^\circ$ phase error), into the phase of daily temperature signals in different locations in Bangladesh [23]. We have also observed similar results with New England temperature signals. We believe that this will be the case with worldwide temperature signals given that we have already observed this with two different locations across the world with datasets given to us by Lincoln Laboratory. To be sure, our initial effort only showed that if we knew the phases of proxies (climate index timeseries), we can predict the phase of a quantity of interest, nearly perfectly. This led us to believe that we could potentially predict future phases of the signal and reconstruct the timeseries that exhibit phase errors.

Thus, here, we will demonstrate the usefulness in using phase-correction techniques in the prediction of long-range weather. We will show that the phase of the analytic signals of global oscillators, found using the Hilbert Transform, can be used to correct for the phase, or more clearly, almost entirely reduce the "phase error," in long-term daily temperature prediction. We introduce a method for automatically correcting much of the phase error of a predicted temperature signal. This method is expanded on in Chapter 2. We also run a sample problem in other (extreme) statistics given to us by MIT Lincoln Laboratory, testing the likelihood that it will be too hot to work in Bangladesh.

We conclude there that much of long-range forecasting error is related to "phase error," and its correction provides for a much improved skill.

1.1.2 Global Prediction

It is not uncommon in local temperature forecasting to use nearby locations to help predict the temperature in an individual location; however, there is little to no research on developing explicit graphical inference models to exploit the spatiotemporal relationships between sites to aid in correction. We will be borrowing from graphical inference towards developing a model for spatial weather correction.

We will use the information present in the temperature and precipitation patterns of different temperature signals in different locations to provide information about the climatological relationships between different locations. We will then utilize information about these relationships to correct for the future predictions of temperature and prediction in some target location given future predictions of temperature and prediction in locations near that target location. Finally, we relate this to a Bayesian framework for accounting for the spatio-temporal relationships among sites. We conclude that the use of context from a spatial network improves predictive skill.

1.1.3 Oscillation Forecasting

Global social and environmental systems are dependent on the ability to forecast large-scale climate variability, which are related to how well we can predict climate indicators such as [4].

With respect to short-term forecasting of ENSO events and indices, dynamical (atmosphere–ocean coupled) models which use physical equations of the ocean and atmosphere are often used and it has seemed that over time, dynamical models have had better performance than statistically-based models. This is likely due to the spatial complexities and nonlinear characteristics in forecasting ENSO from the ocean and atmosphere [7]. However, issues lie with numerical models' availability due to computational cost and complexity [7].

Although ENSO forecasts from these atmosphere–ocean coupled models generally outperform current statistical models, state-of-the-art dynamical forecast systems do not provide a skilful prediction of ENSO and for other oscillations as well with lead times longer than one year [4, 3, 1]. ENSO has similar characteristics to other oscillations in terms of how it's measured [25], and in terms of its dynamics. We therefore evaluate oscillation forecasting literature as a whole as we have identified about ten oscillators that are useful in forecasting temperature [23].

This has led to the development of deep learning approaches towards resolving these issues with dynamical atmosphere–ocean coupled models. Some of these approaches are as follows: Convolutional Neural Nets have seen success in the fields of Tropical Cyclone Detection when taking in pressure and temperature gridded reanalysis data, which inspired our initial experiments on whether these grids, or the oscillation signals, can be used for accurate precipitation forecasting [2, 5]. Convolutional Neural Nets have also seen success in the problem of forecasting oscillations themselves using time-lagged Sea Surface Temperature (SST), Heat Content (HC) and/or Surface Pressure gridded data in the case of the North Atlantic Oscillation (NAO), ENSO and the Madden-Julian Oscillation [4, 3, 1, 6].

We also experimented with the Long-Short Term Memory (LSTM) model, which is a Recurrent Neural Network with gradient-squashing properties, that typically perform better on time-series data due to their recurrent properties (which take advantage of sequential properties) [16]. Long-Short Term Memory models have seen success in many other domains with time series forecasting [16]. The LSTM has been used most notably in [3], which uses a Convolutional LSTM [24] approach on the lagged NAO index signal to capture temporal dependencies in the dataset [3]. A Convolutional LSTM uses convolutional layers rather than fully connected layers in an LSTM to fully express the spatiotemporal relationships present in the data [24].

Simply, we wish to predict a vector of future values of some oscillation over time from grids of lagged Sea Surface Temperature, Heat Content and Pressure grid data. This has been done using deep learning approaches (in [4, 3, 1, 6] as just discussed), and we will discuss our proposed improvements to these approaches in Section (4.3.1).

This work won't have its own chapter but will be part of the discussion determining the usefulness of derived SST grid embeddings.

1.1.4 Oscillation Discovery

The question we then examine is whether ML/statistical techniques can discover the known oscillations that we have observed in climate research. The motivation of course is that if known oscillators can be found in a kind-of "hindcast" framework, then other oscillators that one might computationally find could be significant new discoveries. They are useful in their own right, but additionally, may serve either as a significant or useful part of an ensemble of clocks synchronizing long-range forecast phases.

We hypothesize that there may be significantly more oscillation patterns that are predictive of climate state yet to be discovered and that state of the art machine learning tasks deployed correctly can help us identify these. This finding could potentially have significant consequences for our understanding of what drives long-term weather patterns from an atmospheric sciences perspective.

Most of the literature that we will refer to with respect to Oscillation Discovery are within the fields of Dimensionality Reduction and Representation Learning. This is the due to the lack of research surrounding the subject, but we will show the parallels between the oscillation discovery problem and the classic representation learning/dimensionality reduction problem in Chapter 4. We will not be going over fully unsupervised approaches as they do not factor into our eventual approach. We will be focusing on semi-supervised approaches as mentioned in [13, 14, 17]. We observe the following successes in using representation learning in a structured manner towards deriving the underlying distributions:

- Representation learning has seen success in building word embeddings to effectively represent words in lower dimensional spaces from the original bag-of-words representations using both traditional downstream tasks and transfer learning tasks [20, 21, 22].

- Representation learning has also seen success in many image recognition and segmentation tasks [18, 19].

These successes are due to the ability of representation learning, and more specifically deep autoencoders, to identify useful underlying data-generating distributions [17, 13].

Originally we formulated the problem as follows: Given a set of n uniformly spatially distributed historical weather (temperature/precipitation) signals with T time observations, we have a matrix of observation vectors, $Y \in \mathbb{R}^{T \times n}$, we want to identify a set of $K << n$ vectors, such that these signals are best representative of the phase variables of the oscillations in the data. We additionally have a set of historical climate model grids SST, HC and pressure grids with T observations as well, let's consider each grid to be $M \times N$, we can denote this dataset by $X \in \mathbb{R}^{T \times M \times N}$.

We will show that we can also formulate this problem as a Dimensionality Reduction/ Representation Learning problem of the form that we wish to learn some fixed vector of size K, $\hat{v}_t \in \mathbb{R}^K$ for each time step of the following form, with some loss function to be minimized, L, where each row in $\hat{v} \in \mathbb{R}^{T \times K}$ represents one of K signals over time steps 1 to T:

$$\hat{v} = [\hat{v}_t]_{t=1}^{T} = \hat{f}(X^{\text{lagged}}), \text{ where } \hat{f} = \arg\min_{f} L[g(f(X^{\text{lagged}})), Y]$$

Where X^{lagged} is a lagged form of the signal X with lag of l and memory of s i.e. where $X_t^{\text{lagged}} = [X_\tau]_{\tau=t-l-s}^{t-l}$. Therefore, this problem takes on the form of trying to learn some representation learnable by some function $f : \mathbb{R}^{T \times l \times M \times N} \mapsto \mathbb{R}^{T \times K}$ in a semi-supervised manner such that we can learn some other function $g : \mathbb{R}^{T \times K} \mapsto \mathbb{R}^{T \times n}$ that can best learn Y, our uniformly spatially distributed historical weather signals. This would be consider semi-supervised because we have a signal for which we want the output to best represent the distribution that's driving the fed-in input and output, without feeding in the exact distribution data that we want to learn, though.

Due to data constraints, we run experiments using the formulation that the signal

that we wish to predict, Y, is no longer a matrix of observation vectors, but a matrix of the input signal at time t, to be further elaborated on in Chapter 4, but where our function g takes on the form of $g : \mathbb{R}^{T\times K} \mapsto \mathbb{R}^{T\times L\times M\times N}$, where L takes on the value of 1 if just trying to predict a single time step and $L > 1$ if trying to predict more than one time step outwards. This approach provides us with a formulation towards using SST datasets to aid us in identifying important features in climate signals.

In summary, in chapter 4, we developed and applied techniques as we just discussed towards identifying these targeted underlying latent distributions that could help in identifying temperature and precipitation in certain regions as well as help in predicting oscillation values. We elaborate further in Chapter 4.

Chapter 2

Phase Correction

2.1 Introduction

When forecasting time-series, researchers often use data from the past to predict future values. This is done recursively such that, when extended well into the future, predictions are made with a memory comprising of previously predicted values, as those are, at the time, our history of values.

Errors compound in this process and for signals that are expected to have growth, decay or cycles, such as a temperature timeseries, phase errors form. By phase errors, we mean that features, such as the peak or point of high gradient, lags or leads the actual data. For example, in Figure 2.1, we show what happens when a recursively forecasted timeseries model is applied; and the phase error is evident. A large part of the forecast error is phase error, as one can see that, when there are sharp temperature increases or declines, even a small amount of phase error can cause a huge error between the predicted and actual values.

One could correct errors merely by viewing them as amplitude errors, however, this is a very nonlinear optimization problem. A small phase error produces a huge amplitude error. If one can directly address phase error, the overall improvements could be faster, better or easier. Simply "shifting" the signal may be much easier.

Irrespectively, correcting phase errors is easy to see as important, but known methods do not appear to do so effectively. Some work in this area can be found in [23][1], where the researchers demonstrate the importance of phase errors in weather forecasts and develop joint phase-amplitude correction models.

In spirit, this chapter pursues a similar approach. We will perform an initial time-series prediction for a long time-range, then extract a feature that is relatively insensitive to amplitude fluctuations. We perform this research by extracting the phase of the analytic signal associated with the predicted time series. We then find a way to correct the local phase feature, which is a tractable problem, in contrast to simultaneously correcting instantaneous amplitude and instantaneous phase errors. The corrected and initially predicted Hilbert phase features also yield a time-correction of the initial forecast that is easy to estimate using correspondence between features, which we do using Time-domain warping. The above sequence of steps are novel and lead to a new algorithm that factors the signal into amplitude and instantaneous phase.

The models to correct the instantaneous phase from which the retardation or advancement in time of the predicted signal can be estimated are based on using the "clocks" that large-scale climate indicators inherently contain. Using the instantaneous phases as an ensemble of climate indices together with the observed phases from the past of the signal under consideration, a machine can be programmed to estimate what the future Hilbert phases must be.

This breakthrough suggests that climate indicators can act as synchronizing clocks over long time ranges. The approach we present here naturally recovers seasonal cycles without any particular knowledge of them. Although doing so might seem moot, what is not is that the methodology is applicable to other cyclical variability where time does not provide a natural synchronizing mechanism.

In summary, we aim to use the phase of the analytic signals, found using the Hilbert Transform of the global oscillation indices described in Chapter 1 to phase correct our temperature forecasts 10 years out; see Chapter 1 for more detail on oscillation

[1] see http://stics.mit.edu

indices. The remainder of this chapter details our approach.

2.2 Related Work

For the most part, weather predictions today are formed using numerical weather prediction (NWP) methods. These approaches try to simulate the physical equations that define weather. The issues with these approaches are two-fold: the accuracies of these approaches are highly sensitive to initial conditions; and with demands for higher resolution forecasts to improve accuracy, there exists a tradeoff between computational demands for NWP methods and higher resolution forecasts, making many advances the function of computational advances [28, 29]. Following this, researchers have focused on the problem of weather prediction as a machine learning problem as well and not just a numerical simulation problem. In [29], convolutional approaches are used to improve the forecasting of precipitation up to 8 hours in advance. Other papers have used Convolutional Neural Network (CNN) and Long-Short Term Memory Network (LSTM) approaches for the long-term prediction of oscillation indices such as [3, 4, 1, 6]. Data-driven approaches are often sought because they capture elements of model error that are difficult to eliminate in numerical weather prediction models, due to their reliance on physics; however, phase correction can also be used as a feedback approach for phase error in those models. There is significantly less published work on the prediction of temperature out further than one year versus short and medium-term temperature prediction due to the previously discussed relative difficulties of the problem. There is also little research on the topic of the phase correction of signals.

2.3 The Approach

We will be adopting the following notation: Let $T \in \mathbb{Z}^+$ be the time steps and $y \in \mathbb{R}^T$ be the discrete-time temperature signal, $x_i \in \mathbb{R}^T$ be the oscillation signal for each oscillation index i. Let $\mathcal{H} : \mathbb{R}^T \mapsto \mathbb{R}^T$ be the Hilbert Transform. Let l and s be the associated prediction lead time and history used with a model.

2.3.1 Nominal Forecast Model (f_θ)

The nominal temperature prediction model follows prior work [23], and is based on an ensemble auto-regression approach. Adopting the notation currently used in this paper, we use a tree bagging regressor model [27] with 7 ensemble trees that in order to predict the temperature $\hat{y}[n+l]$ at time $n+l$ takes in $\vec{\Theta}_n$ as input, where

$$\vec{\Theta}_n = [\Theta[n-s] \dots \Theta[n]], \tag{2.1}$$

incoporates memory to the past s days and,

$$\Theta[n] = \begin{cases} y[n], & \text{if measured} \\ \hat{y}[n], & \text{otherwise} \end{cases} \tag{2.2}$$

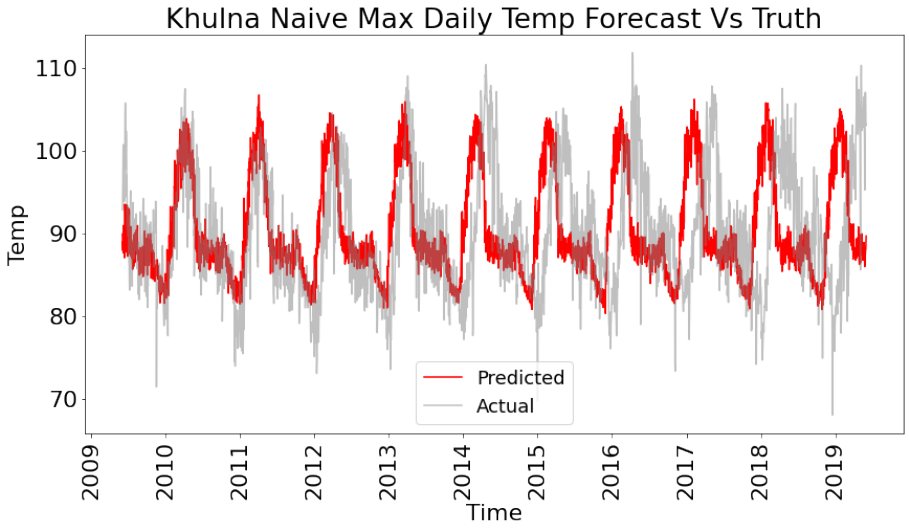
Or in words, the temperature signal from times $n-s$ up until n, consisting of initially measured then predicted values, once there are no more measured values. In summary, we have the following model:

$$\hat{y}[n+l] = f_\theta(\vec{\Theta}_n; \vec{\alpha}) \tag{2.3}$$

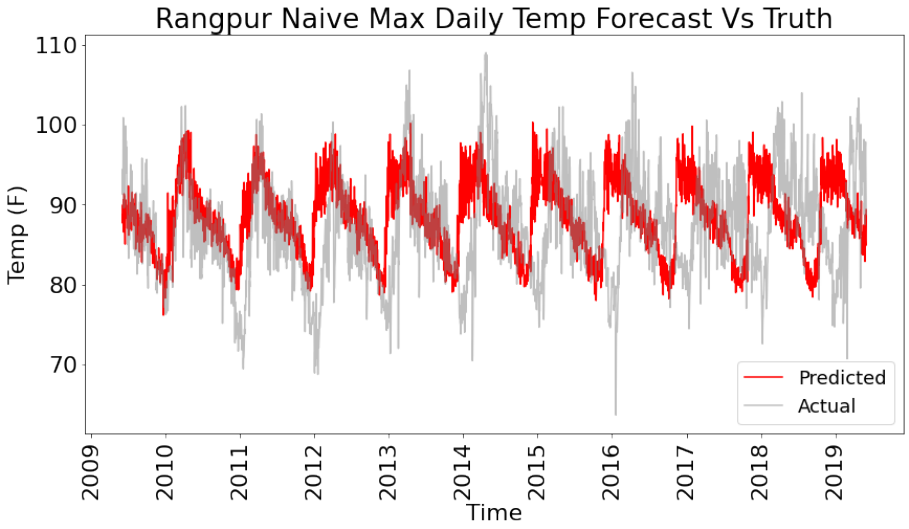
Therefore, when testing, the first l samples use only observed signals as input, but then $\hat{y}[\cdot]$ is used as a variable and so forth, making it a recurrent model. We notice in Figure 2.1 (and in [23]), that such models start to develop phase lags as we get further in time as the errors propagate, which of course is relevant when performing long-range weather forecasting. We can see this in Figure 2.1, as the actual values in blue start to lag the predicted values in green. This model can be shown as f_θ in Figure 2.2.

2.3.2 Phase Prediction ($f_{\mathcal{H},l}$)

This task involves predicting the phase of the analytic signal found using the Hilbert Transform, which we will refer to as the "Hilbert phase," of a temperature signal l years into the future at some location using global oscillation signals as input.



(a) Khulna 10-year forecast of daily temperature



(b) Rangpur 10-year forecast of daily temperature maxima.

Figure 2.1: Data-Driven temperature forecasts (in red) will lag behind truth (in gray) at longer forecast lead times as a manifestation of nonlinear error growth.

Once again, adopting the notation as described above, we will let $y \in \mathbb{R}^T$ be the temperature signal, $x_i \in \mathbb{R}^T$ be the oscillation signal for each oscillation i. Let $\mathcal{H} : \mathbb{R}^T \mapsto \mathbb{R}^T$ be the Hilbert Transform. Let l and s be the associated lag and history of the model, where the history is how far back we retain information from the signal when training i.e. if we're trying to predict time $n + l$, we would use the times $n - s$ up until n. We will denote the Hilbert phase feature of the measured signal to be:

$$h[n] = \angle \mathcal{H}(y)[n] \tag{2.4}$$

To predict the phase of the analytic signal for a temperature signal at a time $n+l$, which we will denote as $\hat{h}[n+l]$, we use the following as features: the observed phases of the analytic signal found from the Hilbert transform for that temperature signal from times $n-s$ up until n, or $\vec{h}_n$, with

$$\vec{h}_n = [h[n-s] \dots h[n]] \tag{2.5}$$

as well as the phases of the analytic signal found by the Hilbert transform for the global oscillation signals at the same lagged times, which we will denote as $\vec{c}_n$ with

$$\vec{c}_n = [o_1[n-s] \dots o_O[n-s] \dots o_1[n] \dots o_O[n]]$$

where O is the number of oscillators and $o_j[n] = \angle\mathcal{H}(x_j)[n]$ is the value of the phase of the oscillator index j at index n. In summary, we have:

$$\hat{h}[n+l] = f_{\mathcal{H},l}(\vec{h}_n, \vec{c}_n; \vec{\beta}) \tag{2.6}$$

Then we run the model to predict the phase feature for $n \in [0, 364]$ and l from 0 to 3285 every 365 values, which gives a 10 year forecast of the phase with 10 different models. This, of course, can be easily adjusted to more or fewer years by changing the l values used. We then use a gradient boosting tree model to represent each $f_{\mathcal{H},l}$ for each l, which can capture the strong nonlinearities present in the data while also being regularized [27]. We train a separate model for each year of lead time (we use 10 years), we then use each model for each year of predictions. This model can be shown as $f_{\mathcal{H},l}$ in Figure 2.2.

We will also denote the phase feature (found using the Hilbert Transform) of the nominal forecast as:

$$h'[n] = \angle\mathcal{H}(\hat{y})[n] \tag{2.7}$$

The difference between $\hat{h}$ and h', for clarity, is as follows: $\hat{h}$ represents the predicted Hilbert phase signal of the temperature signal; while h' represents the Hilbert phase of the predicted signal. The latter is the signal containing the phase lag, while the former is the predicted signal meant to correct it.

2.3.3 Time-Lag Correction Model (f_3)

We can see in Figure 2.5 that the phase correction model in (2.1) can predict a phase from 10 years out using global oscillation data (the red line) that seems to not be phase shifted to the degree that the model in (2.2) seems to be (the blue line) relative to the actual values in green. We hope to be able to learn from this towards generating a phase-corrected temperature signal, $\hat{y}^{\mathbf{PC}}$.

The Approach: DTW

We use dynamic time warping to identify the time shift for our temperature signal and we find the matching π^* that minimizes the following equation:

$$\pi^* = \arg\min_{\pi} \sqrt{\sum_{(i,j)\in\pi} \left[\hat{h}[i] - h'[j]\right]^2 + \lambda \sum_{(i,j)\in\pi} [i-j]^2}$$
$$\text{s.t. } |i-j| < \gamma_1, |\hat{h}[i] - h'[j]| < \gamma_2, i \leq j \tag{2.8}$$

Therefore, this matching seeks to find the most similar pairing, while using λ to regularize for how far the indices are from one another and using γ_1 and γ_2 to restrict how far matched points can be from another and how far in value points can be from one another, respectively. We then use this matching to update our predicted temperature signal, $\hat{y}$. Therefore, we get:

$$\hat{y}^{\mathbf{PC}} = \hat{y}_{\pi^*}$$

And we only use predicted quantities to obtain the signal.

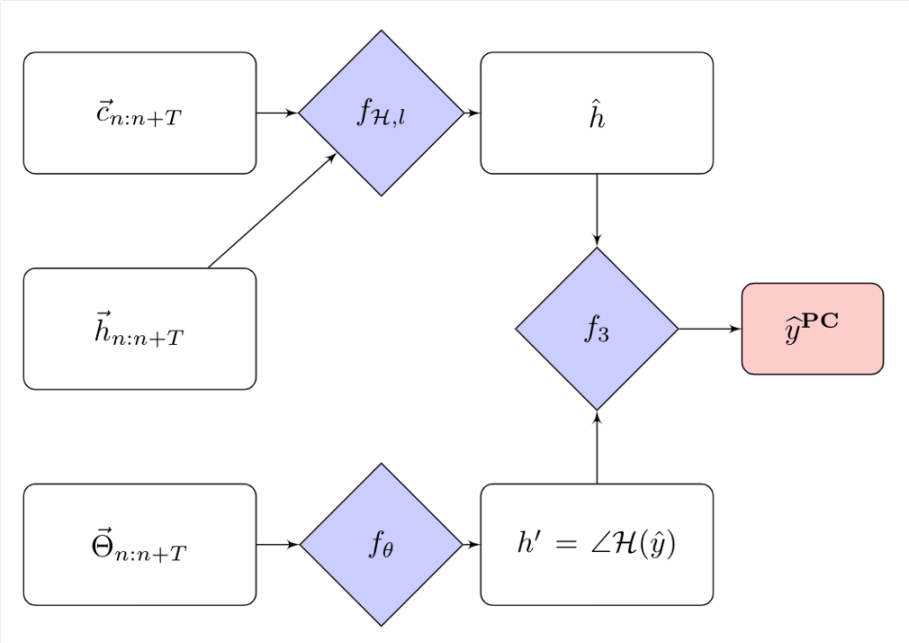


Figure 2.2: Provides a flow diagram of the model structure relating the preceding sections.

2.4 Examples

2.4.1 Nominal Forecasting Model (f_θ)

The bulk of the results for this section can be seen in confirming the phase lag in Figure 2.1, when applying the model to Rangpur and Khulna, Bangladesh, respectively.

2.4.2 Phase Prediction ($f_{\mathcal{H},l}$)

The results can be seen in Figure 2.4 for Khulna and Rangpur.

2.4.3 Time-Lag Correction Model (f_3)

We choose $\lambda = .07$, $\gamma_1 = 365$ and $\gamma_2 = .2$. We chose λ and γ_2 through experimentation and observation. We chose γ_1 as a value of 365 ensures obvious correct matching with a phase lag that doesn't exceed one year after 10 years. See Figure 2.5 to observe the effects of phase correction. When testing an approach where we

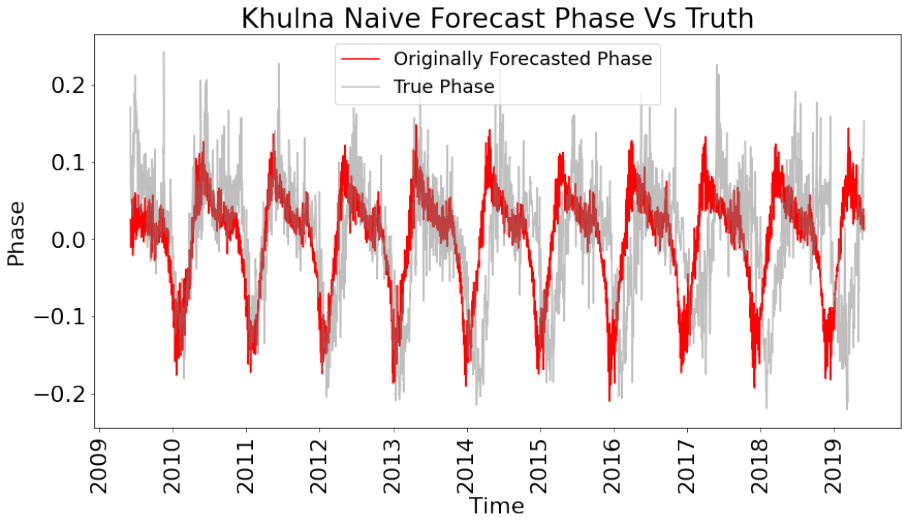


Figure 2.3: The instantaneous phase of the of the analytic temperature signals (forecast-red and true-gray) also contain relative time lag. We discover that simple regression machines can correct long range Hilbert phase errors, where we use the Hilbert phase to denote the phase of the analytic signal.

Method	MAE (μ, σ)
Phase Correction (PC)	$(2.207, 1.421)$
No PC	$(3.626, 1.919)$

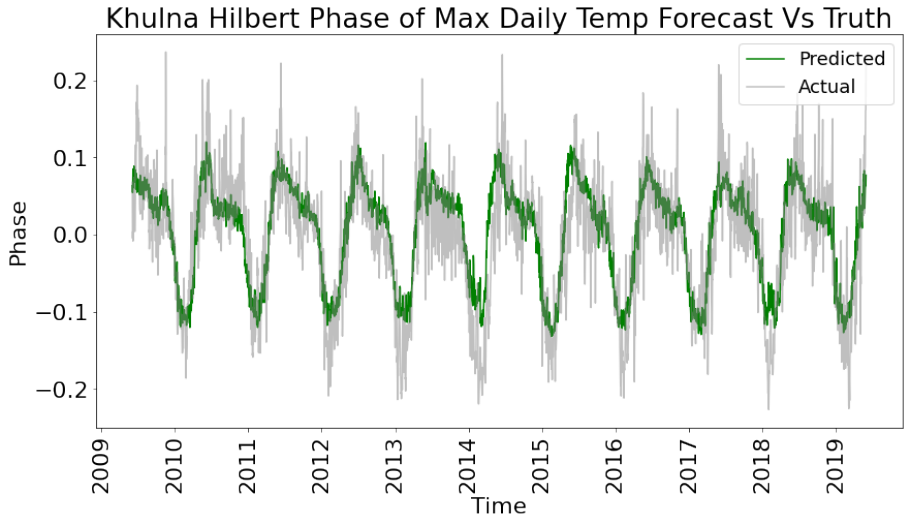
Table 2.1: **Khulna** Errors (in Kelvin) when trained through 2005 over 500 Splits of 10 years starting from from random dates between 2006 and 2009.

train the model through 2006, then randomly sample 250 starting points for the model within 2006-2009 and start our forecast from there, we yield Figure 2.6 where we see that the relative error is reduced as our forecast moves outward. This method also yields the results in Khulna in Table 2.4.3.

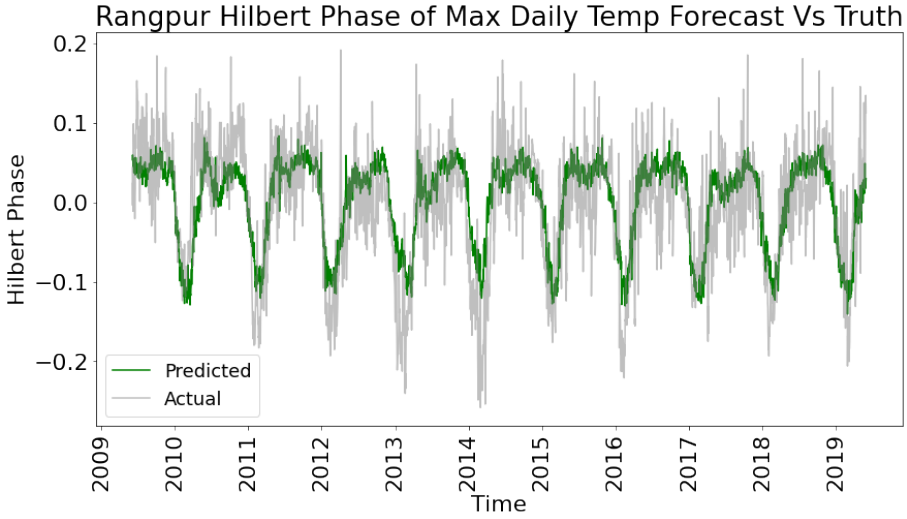
In cross-validation over a larger time frame, which should be considered more valid, we still see a reduction of approximately 20-25% of the error. As can be seen in Table 2.2.

Method	MAE (μ, σ)
Phase Correction (PC)	$(2.097, 1.839)$
No PC	$(2.706, 2.413)$

Table 2.2: **Khulna** Errors (in Kelvin) Cross-Validated over 5 Splits from 1996-Present



(a) Khulna 10-year forecast of daily hilbert phase of temperature maxima



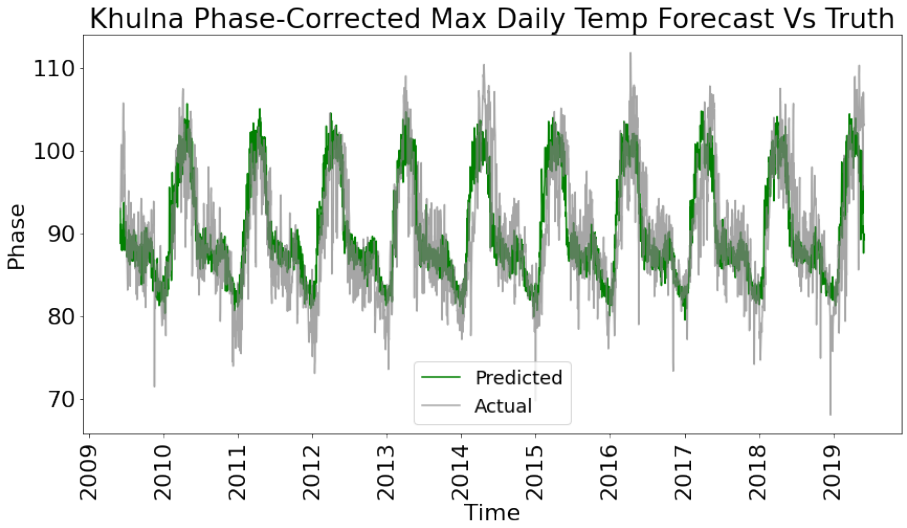
(b) Rangpur 10-year forecast of daily hilbert phase of temperature maxima.

Figure 2.4: Predicted ten-year corrections for the Hilbert-phase feature (green). The predicted values match the true phase (gray) with no time-lag. The corrected Hilbert-phase features and nominal temperature forecast. Hilbert-phase feature analysis corrects temperature forecast phase errors.

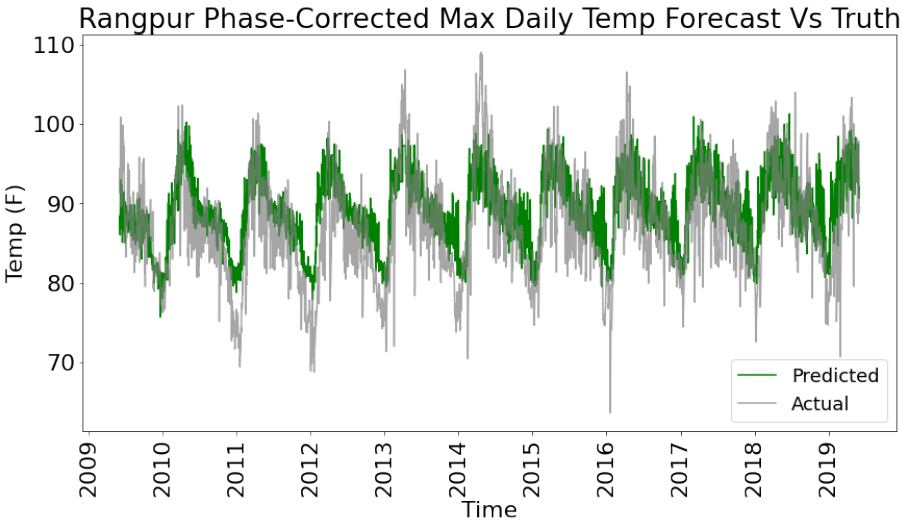
2.5 Discussion

2.5.1 Extremes and bounds:

We also worked on the problem of testing "hot days," or days that have a max temperature beyond 95% of days. As visible in Figure 2.7, and from the large reduction in mean square error (almost 60%!) we are able to improve significantly the skillfulness of predicting extreme periods of heat.



(a) Khulna phase corrected 10-year forecast of daily hilbert phase of temperature maxima



(b) Rangpur phase corrected 10-year forecast of daily temperature maxima.

Figure 2.5: To obtain this, we find the ideal "path" from dynamically time warping the Hilbert phase of the predicted temperature signal (Hilbert phase of output of f_2) onto the predicted Hilbert phase of the signal (output of f_1). We then use this path to transform the predicted temperature signal (output of f_2) to get the output of f_3 above.

2.5.2 Site Network work

Now that we have forecasts for an individual location, we can once again use a site network model previously developed to correct for these temperatures to an even greater degree by using the predictions of surrounding locations. This site network will be fully elaborated on in Chapter 3.

Method	MAE (μ, σ)
Phase Correction (PC)	$(2.133, 1.778)$
No PC	$(2.774, 2.469)$

Table 2.3: **Rangpur** Errors (in Kelvin) Cross-Validated over 5 Splits from 1996-Present

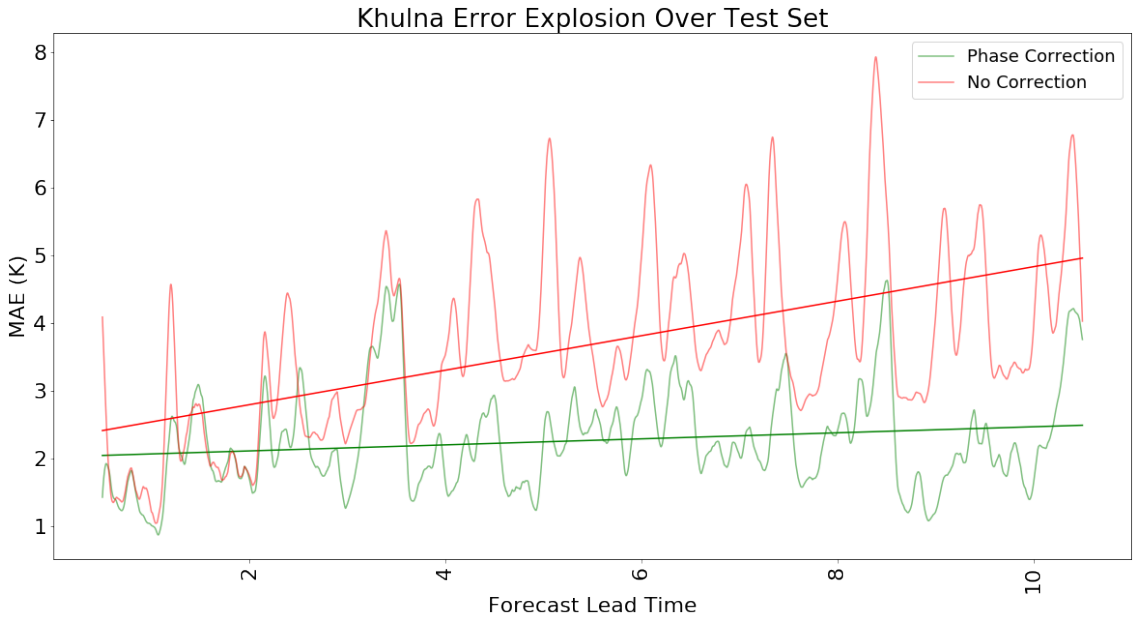


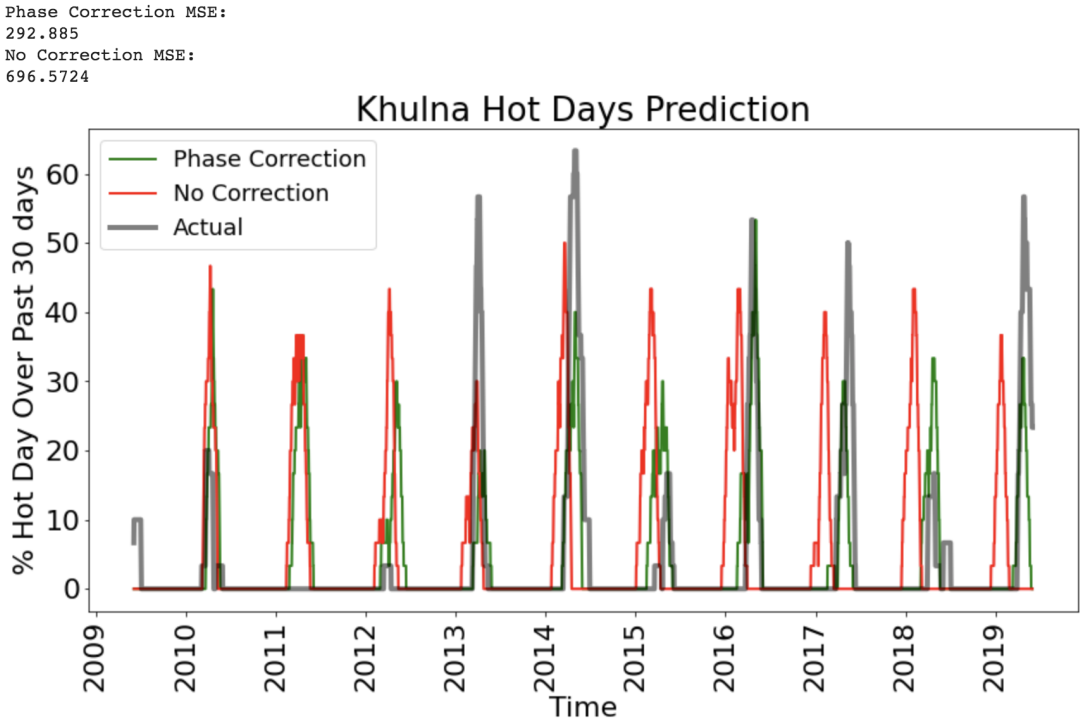
Figure 2.6: Khulna Error Over Time, Forecast Lead Time is in Months

2.5.3 ENSO prediction:

We tried to apply this to the actual signals of oscillators (such as ENSO) as well and of course, excluding itself from the training set of oscillators as a result. However, the model didn't seem to perform as well in the setting of monthly granularity, as oscillation values are typically recorded on a monthly scale, rather than the daily scale of temperature.

2.6 Conclusion

We have introduced a method for resolving phase error that can be applied to any signal that is a function of time, including any temperature of interest at any location worldwide. The results in Figure 2.6 seem to show that the phase error is a large part of what makes long-range forecasting difficult, as the slope of the error (as



(a) Khulna daily temperature maxima "hot days" forecast .

Figure 2.7: Hot days prediction over the past 10 years on Khulna. We can see that the hot days forecast of the phase corrected model has much more usefulness 5-10 years down the line. Relative Mean Squared Errors above.

a function of lead time) noticeably decreases when applying our phase correction techniques.

We believe that larger improvements are to come, as not all of the phase error has been resolved, as well as the corrected signal having parts where it is excessively "blocky," as a result of the errors in dynamically time warping the signal. We do believe that this is an informative study for the reduction of phase error in long-term weather forecasting. We believe that an informative future project would be the application of phase correction to NWP ensembles on medium and long range forecasting problems, as well as exploring what portion of the deviations in those ensembles lies with the "phase error."

Chapter 3

Global Correction

3.1 Introduction

This chapter develops a method using proximal context from neighborhoods of a network of observing stations to improve temperature prediction at an individual site. Specifically, we introduce a Gaussian Graphical Model [37] that is identified or trained using spatial correlations of proxy data between stations to infer the marginal posterior distributions (end expectations) of quantities of interest at a given site.

The correlations are not causal, but provide additional constraints. For example, if this model was to be used over the whole of the United States and we had a prediction such that some locations in interior states will be significantly colder than locations in northeast states, then this model would look to either correct the temperatures of the northeast states upwards or the interior states' predictions downward, depending on what all of the other locations around it point to as the likely relationships and predictions in this context.

Referred to as a Graphical Model [37], the approach implements inference by message passing between nodes. The messages are changes or updates that one node must experience as other nodes are being constrained. Together, the messages between all nodes equilibrates their posterior distributions. Graphical Models are well developed and here we use Gaussian Graphical Models in particular [37]. We trained a Gaussian

Graphical Model model that automatically weighs contextual corrections versus the local predictions at a given site.

Throughout the rest of this chapter, we will further elaborate on how this method works and show an example of it being applied on a dataset given to us by Lincoln Laboratory regarding monthly temperatures in the New England region.

3.2 Data Processing

The data that we received from Lincoln Laboratory contained 10 different locations and the monthly (daily) averages of the temperature and the monthly (daily) averages of rainfall and snowfall for each location from 1950-2018; where monthly (daily) averages refers to the daily average value of the signal for each month. The data was not completely "clean" so we first needed to clean it up. We used the Expectation Maximization (EM) algorithm [32] to fill in the missing precipitation values for each individual county separately, where precipitation refers to the sum of rain and snow, where each is denoted by monthly average total inches. We found the best results when rain and snow were combined for a single precipitation field. We also found best results when the EM algorithm was used separately for each month as there exist clear differences in the distribution of weather across different times of the year, i.e. the natural distribution is multimodal.

3.2.1 Metric

Since the data is multivariate, we define a norm to compare two records (or vectors) of the data. The distance between the individual locations is based on a Mahalanobis distance, where, in Q, x represents the location and t the time:

$$Q_{xt} = \begin{bmatrix} \text{tempdiff}_{xt} \\ \text{precip}_{xt} \end{bmatrix}$$

$$d(i,j) = \frac{1}{T}\sum_{t=1}^{T}(Q_{it} - Q_{jt})^T COV^{-1}(Q_{it} - Q_{jt}) \tag{3.1}$$

COV is a $2x2$ matrix that is a covariance matrix of $Q_{it} \in \mathbb{R}^2, 1 \leq t \leq T, 1 \leq i \leq L$ with respect to itself. This is supposed to represent the covariances of the predictive features with themselves across the datasets. We are using L to indicate the number of locations in our system. We use T to represent the number of timesteps in our training set.

We then use pairwise distances between sites to define a Kernel that establishes the graph where whose edges are the distances the counties as nodes. We can then perform message passing (Belief Propagation) to improve the accuracy of predictions at the L sites. Note that, implicit in this approach is the interpretation of the symmetric positive definite matrix, a kernel, emerging from pairwise distances, as a high-dimensional Gaussian distribution, which is approximated by a sparse Graphical model.

3.3 The Global Algorithm

We work on each indicator (Temp Diff and Precipitation) individually. This model is to be for a certain time t and we will take our predictions for such a time t and normalize them over the expected affinities between locations.

We have our input $x \in \mathbb{R}^{L \times 1}$ (e.g. temperature), which is our local prediction for the time N in which we want to evaluate. We have our true values, $f_{xt} \in \mathbb{R}^{1 \times 1}$ for each time t and location x. Our local predictions $\hat{f}_{xt} \in \mathbb{R}^{1 \times 1}$ for each time and location, our locations, L, our kernel, $\Sigma = J^{-1} \in \mathbb{R}^{L \times L}$ where J is defined by

$$J = [d(i,j)]_{\forall (i,j) \in L \times L} \in \mathbb{R}^{L \times L} \tag{3.2}$$

This gives us a kernel, Σ, such that locations with smaller distances from each other, where distance is referred to as $d(\cdot, \cdot)$, will be highly correlated and vice versa.

$$\underline{\mu} = [\mu_x]_{\forall x \in L} = \frac{1}{T} \sum_{t \in T} f_{xt} \tag{3.3}$$

$$h^- = \Sigma^{-1}\underline{\mu} \in \mathbb{R}^{L\times 1} \tag{3.4}$$

We can then define our graphical model as

$$G = (J, \delta h^-) \tag{3.5}$$

Letting

$$\delta h^- = h - h^- \tag{3.6}$$

Where

$$h = \Sigma^{-1}x \in \mathbb{R}^{L\times 1} \tag{3.7}$$

and

$$x = [\hat{f}_{xt} \forall x \in L] \in \mathbb{R}^{L\times 1} \tag{3.8}$$

δh^- represents the "potential differences". We can then run belief propagation on the gram matrix J and the potential vector δh^- to obtain δh^+, which are the normalized "potential differences" with respect to the global predictions. We can see in Figure 3.1 what that looks like.

$$\text{BP}(J, \delta h^-) \Rightarrow \delta h^+ \in \mathbb{R}^{L\times 1} \tag{3.9}$$

We then perform Belief Propagation [37], which is a method for computing the marginal distribution of unobserved variable, conditional on observed variables, in a graphical model through message passing. in this case we do to calculate an update vector on our forecasts, through estimation of the marginal distributions on this graphical model. We have to perform Loopy BP [37], to local convergence, which doesn't have an exact global solution, due to the cyclic nature of the graph.

$$\hat{h} = \delta h^+ + h^- \tag{3.10}$$

$$\hat{x} = \Sigma\hat{h} \in \mathbb{R}^{L\times 1} \tag{3.11}$$

We now have obtained globally predicted values for each location $\hat{x}$ that we adjusted by our model from the locally predicted values x. We do this by learning an overar-

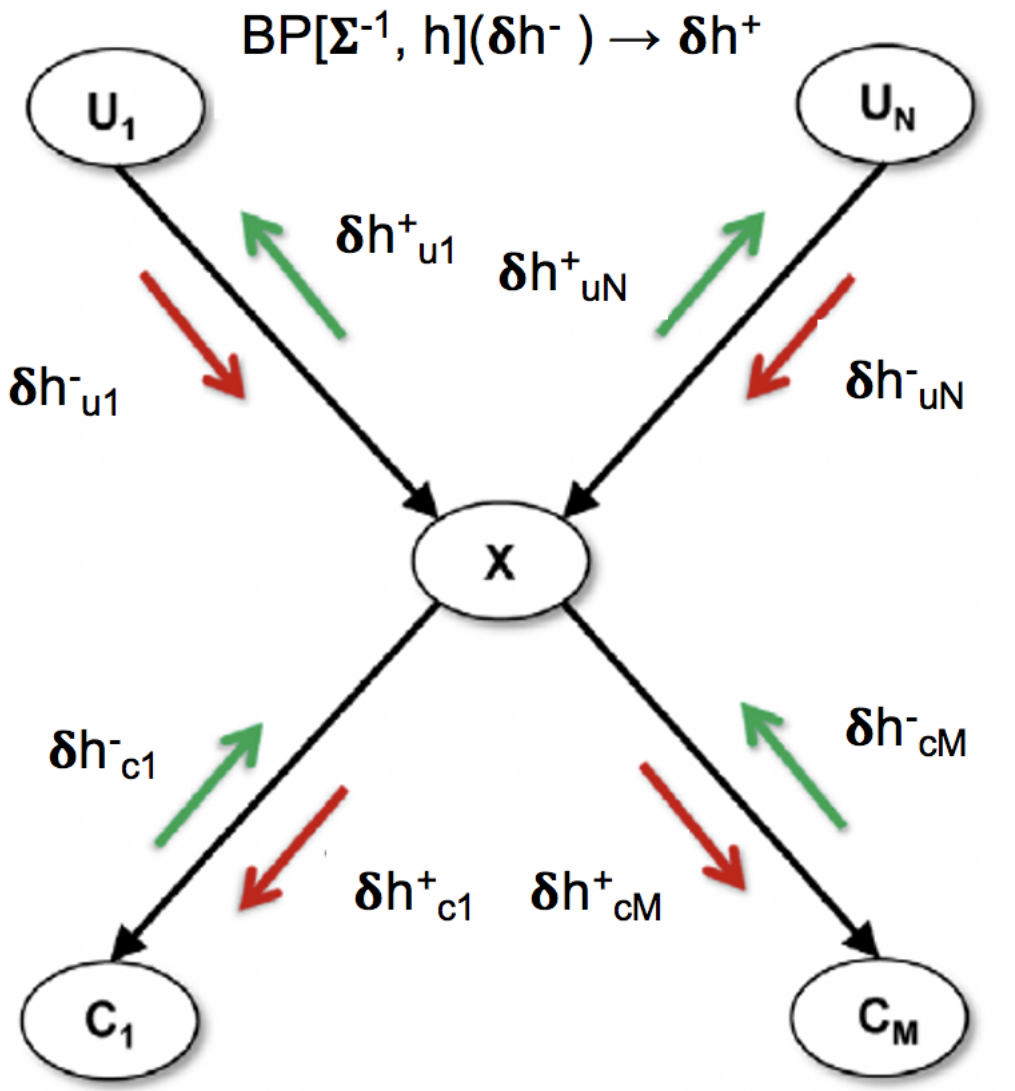


Figure 3.1: A basic representation of what a model like when it goes through belief propagation on a graphical level. This isn't an exact representation of our model from a nodal perspective but the edges represent the message passing structure that our model takes on. Credits: Edited image from [36].

ching distribution using a "sum-product"-like algorithm from our observations and our observed covariances which we essentially assume to be true.

The graphical model provides us with the insights on inference and is particularly useful for high-dimensional problems. Under certain assumptions, the BP on a Gaussian Graphical Model is equivalent to standard Linear-Gaussian Bayesian estimation. The latter can be computationally simpler when the burdens of estimating the information matrices are low. Given the cyclic nature of the grahical model, we

would also have to use Loopy BP, which is an inexact approximation for BP.

Thus, for small networks, direct bayesian estimation can be performed for an exact solution as a function of x. Both the solution that was just presented along with the solution that we will present next are useful in different settings.

We simplified this mechanism to be able to optimize an output vector given our set of predicted temperatures, x, as follows. We are interested in finding

$$\hat{x} = \arg\max_{Y} P[Y|x] = \arg\max_{Y} P[x|Y]P[Y] \in \mathbb{R}^{L\times 1} \tag{3.12}$$

Such that $\hat{x}$ is the corrected temperatures of $x \in \mathbb{R}^{L\times 1}$, which are the inputs to the correction, which is the predicted values of temperature for each location at some timestep. We will use X and x interchangeably going forward in this chapter; using X when doing algebra. We will also be using $\mu \in \mathbb{R}^{L\times 1}$ as defined above to represent the means of each locations' values

$$P(Y|x) \propto e^{(X-Y)^T\Sigma_{XX}^{-1}(X-Y)}e^{(Y-\mu)^T\Sigma_{YY}^{-1}(Y-\mu)} \tag{3.13}$$

Which is saying that the likelihood function is equivalent to the likelihood of the output Y given the mean values we have seen previously times the likelihood of the output given the input values and their observed distributions.

Of course, when the derivative is zero is when the likelihood will have a stationary point. We also know that since $f(\cdot)$ is a multivariate gaussian that we can take the natural logarithm first and we will have an easier and equivalent function to set to zero, giving us:

$$f(Y) = \frac{1}{2}(X-Y)^T\Sigma_{XX}^{-1}(X-Y) + \frac{1}{2}(Y-\mu)^T\Sigma_{YY}^{-1}(Y-\mu) \tag{3.14}$$

$$\frac{df}{dY} = 0 = -\Sigma_{XX}^{-1}(X-Y) + \Sigma_{YY}^{-1}(Y-\mu) \tag{3.15}$$

Therefore,

$$Y = (\Sigma_{XX}^{-1} + \Sigma_{YY}^{-1})^{-1}[\Sigma_{XX}^{-1}x + \Sigma_{YY}^{-1}\mu] \tag{3.16}$$

We can now substitute our values that we defined before back in to give us a solution for Y, plugging in $\Sigma_{XX} = J^{-1}$, as we approximated J above in the GGM. Please note that Σ_{XX} is not the covariance of X, but an alternate metric, based on the kernel, has been chosen. We then define the (paramaterized) inverse covariance matrix of the output to be $\Sigma_{YY}^{-1} = \lambda I$, which gives us a single parameter to optimize over, λ, that represents the inverse of the variance for the predictions.

$$\hat{x} = Y = (J + \lambda I)^{-1}[Jx + \lambda I\mu] \tag{3.17}$$

Observing, our solution for $\hat{x}$, we can see that when $\lambda \to 0$, we get a solution of $\hat{x}$ that would be just predicting our input x. While if $\lambda \to \infty$, then $\hat{x}$ approaches the mean values, since the inverse of the variance would now approach zero; this provides us with a value of λ to optimize over, where are a larger value of λ shifts the values towards the mean, μ. We can now easily see how shifting the value of λ will achieve the effects of the graphical model by providing a normalization constant for controlling how our forecasts will be normalized to what we've seen in the past, while allowing us to optimize over the marginal distributions, exactly like we wanted to do in the graphical model. Now, we can find the value of λ such that we maximize the accuracy of our results on our set of training data. Given more data, we would want to estimate λ and/or J as a function of time in order to account for temporal differences across the spatial relationships.

In summary, the Gaussian Graphical Model can be constructed over a network-wide Kernel defined from a metric on multivariate quantities. A simpler model is to perform explicit Bayesian estimation, which we have done for each field separately.

3.4 Coupling with Local Prediction

To demonstrate usefulness, we use a simple xgboost [27] model (gradient boosting tree) to predict a value (temp difference from the monthly mean, precipitation difference from the monthly mean), at time N, b time in the future at an individual location. This model takes in $|T|*(k+1)$ values as input where $T \in [N-b-T, N-b]$

as the timestep we wish to predict, N, is b months in the future and k is the number of climate oscillators of interest, the additional value used for each time frame (since we use $k+1$ values) is the value of the signal itself that we wish to predict at each time within T. We use mean squared error as our loss function. There are likely better ways to preprocess the data but the purpose is to demonstrate the usefulness of the global context provided by the oscillators and then the additional updates from the site (global) model. Gradient boosting trees are a natural choice for problems using oscillators as inputs given their usefulnesses for different periods of the year, as some oscillators have more importance in different areas in different seasons than others, naturally, given that they are climate oscillators.

3.5 Results

First, we broke up the data into 7 year splits between 1948-2018. We tested on the last 5 folds of the 10-fold cross-validation, training the local model and the global parameters only up until the previous year. We then used the results over the splits to generate Figure 3.2 below that best demonstrates the difference in errors when using the global model.

3.6 Conclusion

This model has the goal of using the forecasts from the locations around a location, along with the observed correlations and observations from these locations, to make future updates towards "normalizing" these forecasts. We can see in our results, as plotted in the Figure 3.2, that the model tends to reduce the error in our forecasts. This approach would likely be even more effective over a larger site model with more locations. We believe this would be more effective with more granular data as well, which would also clarify the relationships among the sites.

We also believe the model could be much improved by adding a time component to our Bayesian model. An Ensemble Kalman filter or Particle filter [33] are natural

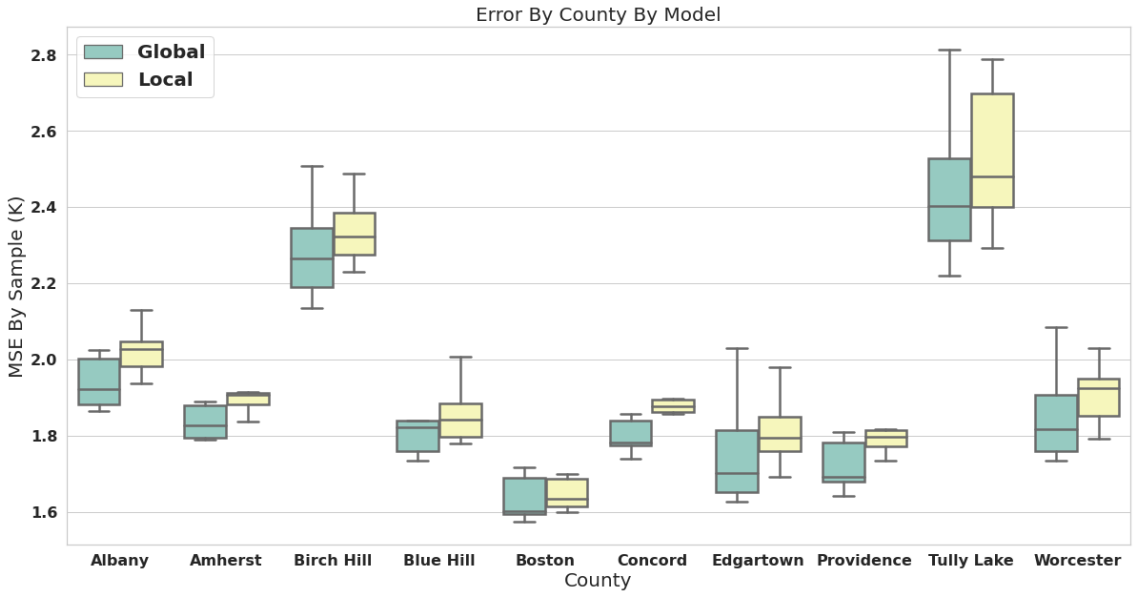


Figure 3.2: The "globally corrected" forecasts are teal and the local forecasts are tan. This figure demonstrates the usefulness in the correction of the locations from each other. This is trained and tested using a 12 month lag ($b = 12$) for temperature prediction.

choices, however, our biggest limitation in training is the lack of data when using a monthly time scale as we do in this problem.

Another approach could be to have the normalization constant and covariance matrix that is seasonal or monthly. A linear approach could work in a limited data setting with a one-hot vector under the form that would take time in as a 12-dimensional one-hot; however, a system would need $12L^2$ weights to train if the covariance matrix as a function of t, $\Sigma_{XX}^{(t)} = \Sigma_{XX} + \beta \mathbb{1}^{(t)}$, where β is the weights matrix and $\mathbb{1}^{(t)}$ represents the one-hot matrix of time. The issue thus still remains; in our example of 10 locations and a few hundred samples then we are subject to overfitting due to there being more weights than datapoints to represent this simple problem; this makes it possible for the weights to memorize the data but not capture the underlying relationships, implying that we need daily data and/or a more deparameterized problem formulation in order for this approach to be more reliable.

Chapter 4

Oscillation Discovery and Prediction

4.1 Introduction

In this chapter, we explore ways of improving the state-of-the-art convolutional models [8] used in oscillation forecasting through the lens of data representation. Furthermore, we aim to use insights from oscillation forecasting to help develop models towards discovering new oscillations and vice versa. We hypothesize that deep learning models can provide insights on large sets of simulation data for new oscillator discoveries. We will further elaborate on the proposed models of choice and why research has shown that these are appropriate in the following subsections.

We argue that if oscillations are the drivers behind observed or simulated temperature signals, then we can formulate a Representation Learning problem to identify the latent subspaces driving long-term weather in individual regions. Representation learning has been used for time-lagged modeling [13], and offers substantial advantages over Principal Component Analysis (PCA), and Kernel-PCA techniques. Convolutional forms of representational learning appear to be particularly suited to the gridded (here 2D) fields under study.

State of the art long-term oscillation forecasting models typically use SST, HC and

surface pressure grid data [4, 3, 6]. Through the brief experiments in this chapter, we aim to explore how SST grid data can be used for the understanding of oscillators.

In particular, we use the SST/pressure grids over time as input to an autoencoder targeting a certain output. An advantage of this is to find latent oscillations that are not immediately apparent in the gridded data itself as a function of the learned latent autoencoder representation. This allows us to identify the "underlying distributions" of long term temperature signal, providing a basis for an "oscillator index" similar to indices with respect to specific global climate patterns [25].

The rest of this chapter discusses why a representation learning approach is a good way to attempt to generate these oscillation indices, and the results that our approaches yield when attempting this problem.

4.2 Methods

The computational discovery of oscillators can lead to stable predictors for synchronizing the observed signals across the earth. The discovery can be primed using models or data itself, though the former lends itself much more easily to systematic learning.

For this chapter, we retain our formulation framework from Chapter 1 such that $g(f(X^{\text{Lagged}}))$ takes on the form of an Autoencoder. However, we have a large difference in how we use it here as we ran into cost constraints with Lincoln Laboratory with respect to the spatially distributed temperature data. Therefore, in order to run similar tests for identifying, or approximating, oscillators, we trained a model to learn an underlying representation of the data-generating distribution of predicting SST data from a lagged (or unlagged) version of itself. The function f in our work will thus take on the form of a Convolutional encoder, while g will take on an inverse convolutional decoder such that the internal layer will best represent the oscillators of the data. The fundamental idea, to be sure, is to use representational learning to learn the features that captures essential information to synchronize other observed data. To that extent, the internal or latent features will themeselves function as

oscillators.

In this chapter, we use three classes of auto- and lagged-encoders on Sea Surface Temperature grids to develop insights into the prediction of oscillations. We will demonstrate each of the experiments below along with the motivation for each. At the end, we will show some surprising results. We use reanalysis grids of the surface temperature from NOAA [25] that cover a 73×144 patch; we condense this to a 72×144 patch for numerical ease. We will consider a grid at time t to be X_t going forward.

4.2.1 Direct Autoencoder

The Direct Autoencoder [13] model that we use utilizes a similar but slightly more complicated structure than what is shown in Figure 4.1. Similarly, this model takes a 2d matrix as input and a 2d matrix as output, in this case, taking in the current time SST grid, X_t, as both input and output, i.e. training f and g to optimize $||X_t - g(f(X_t))||_2^2$ with the goal of compressing it down to a $k = 40$-dimensional representation, the output of f, for each time t that best represents the grid. This is being explicitly learned via the downstream task of building this vector representation as an embedding layer for a convolutionally downstream task to learn this vector that is being trained by the propagation of the losses from the upstream task of re-learning the original SST grid from the information using an in identical, but inverse, network. The network that we use for the downstream task consists of 4 convolutional layers, each of 5 filters and of filter sizes of 3×3, 5×5, 7×7, 5×5 in order. Each layer has a relu activation function along with max pooling, batch normalization, gaussian noise and dropout layers. The upstream task has the same layers in opposite orders, and the max pooling is inverted such that the same hypothesis space used to encode the data is available to decode it; the only difference with the upstreamed inverse is that the inverse has a final layer with 1 filter in order to have a correctly formatted output, as we only have a single output channel.

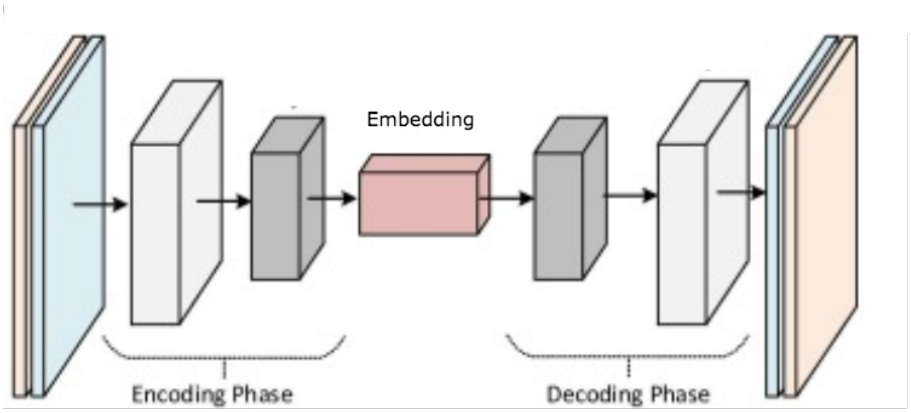


Figure 4.1: A generalization of the Direct Autoencoder model used.

4.2.2 Lagged Autoencoder

The Lagged Autoencoder takes on a similar structure to what we saw in the Direct Autoencoder. The main difference is on the first input convolutional layer (also consisting of the max pooling, batch normalization, gaussian noise and dropout layers). Rather than having one of these layers, we have L of them. Then after that layer (with a max pooling operation as well), the outputs of the layers are added together and then fed into the model that we have in the direct autoencoder. The difference here is that we have L 2D matrix inputs now. During training, when trying to predict X_t as an output, we use $X^{\text{lagged}} = \{X_{t-L}, ..., X_{t-1}\}$ as inputs; or more simply, it takes in the previous L SST grids ending with time step $t-1$ in order to predict the SST grids at time step t. We optimize over $||X_t - g(f(X^{\text{lagged}}))||_2^2$. This optimization allows us to use previous information that accounts for the movements of sea surface temperature to help differentiate these embeddings from the previous embeddings, which don't explicitly account for SST movement.

4.2.3 Skipgram Autoencoder

The Skipgram Autoencoder takes on a similar structure to what we saw in the Lagged Autoencoder. The main difference is on the last output inverse convolutional layer (also consisting of the max pooling, batch normalization, gaussian noise and dropout layers). Rather than having one of these layers, we have M of them.

Therefore, right before the last layer of the convolutional model, we have M different layers, each with its own weights and each for a different 2D output. The difference here is that we have M 2D matrix outputs now. During training, we now are trying to predict $X_{t+1}, ..., X_{t+M}$ as an output, we use $X^{\text{lagged}} = \{X_{t-L}, ..., X_{t-1}\}$ as inputs; or more simply, it takes in the previous L SST grids ending with time step $t-1$ in order to predict the following M SST grids starting with time step $t+1$. We optimize over $\sum_{\tau=1}^{L} ||X_{t+\tau} - g(f(X^{\text{lagged}}))_{t+\tau}||_2^2$, or the sum of squares of the output vector of size M.

This formulation allows us to similarly use previous information that accounts for the movements of sea surface temperature to help differentiate our embeddings. However; this type of model training is called skipgram training as we use surrounding values to build a representation for the current value and has become popular when training word embeddings as it requires the knowledge of the intermediate word in order to guess the next words [21, 20]. We think of this process similarly but with respect to sea surface temperatures, as it must truly understand the grid at time step t from the previous grids in order to be able to effectively predict the future time steps after it, since we skip time step t in our model.

4.2.4 Results

We test how each of these encodings can do when trying to predict the ENSO index. Since the ENSO is a monthly scale, we average our encodings for each month so that we have embeddings on a monthly scale as well.

We needed some metric to determine which of these embeddings worked best. We decided to use the explained variance when feeding only our embeddings into a Gaussian Process (GP) Regressor to predict the ENSO values, to determine, on a test set, how much of the variance of ENSO forecasts are explained when only having these encodings as input. This gives us a measure of the embeddings' usefulness in an ENSO prediction setting. We use a GP model given that both our inputs and outputs takes on the form of a gaussian, which makes it a natural approach for a problem with limited data and non-negligible like this one.

A GP Regressor is a nonparametric, bayesian approach to regression that tends to work well on smaller datasets (which ours is since we converted it to monthly values). Given a Gaussian Process Regression models ability to infer a similar distributional structure (RBF kernel) over the outputs from the inputs, we chose to evaluate these embeddings using the out-of-domain explained variance of each of these models; when the smoothing factor of the gaussian kernel is optimized over the training set. The training set is of course the same set in which the encoders that encode the embeddings are trained. See [35] for more information on Gaussian Process Regressors and why they're useful in this context.

We then ran trained each of the above autoencoder formats on 100 epochs of training with Adam on a training set consisting on the first 90% of samples. We then train a Gaussian Process model on those encoding outputs on the same first 90% of samples.

We used a lag of 1 month and where α, which is the coefficient that determines the smoothness of the Gaussian kernel that's used for noise, was chosen through cross-validation on the first 90%. We set the skipgram forward time in months to be $M = 6$, and the lag time for the skipgram and lagged models to be $L = 24$; we defined these parameters above. We use only the embedding generated at time t as the input into the GP to predict the ENSO value at time $t + 1$. We see in Figure 4.2 the comparative results between the architectures.

4.2.5 Conclusions

From the figure, we can observe that that the Direct Autoencoder model can explain 60% of the variance one-month out and can still explain 30% of the variance a year and a half out. This model is only trained on the SST grid itself, indicating the impressive ability of an embedding to be able to capture so much information with respect to the ENSO index, without any information to the ENSO index itself during training, as the embedding is the only input into this model. It is surprising that the Direct Autoencoder model appears to be the best given its lack of ability to account for SST movement; however, it is also the only model to take in the grid as input directly, so that may give future insights for further development.

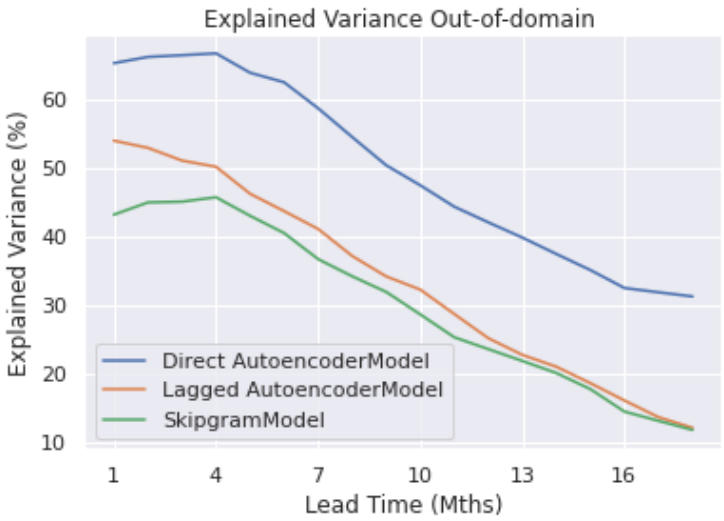


Figure 4.2: Explained variance by architecture. "Mths" refers to the number of lagged months. Explained Variance refers to the percentage of the variance of the output explained by only the input embedding (percentage reduction of MMSE from variance of Y).

In conclusion, we showed three different ways to predict the underlying embeddings of Sea Surface Temperature grids that definitely contain information necessary to predict oscillations and may contain unknown oscillators themselves.

4.3 Future Work

4.3.1 Oscillation Forecasting

We argue that the use of a convolutional learner for oscillation forecasting could be improved through the use of residual blocks rather than the traditional convolutional blocks used in [4]. A residual block is similar to a convolutional block with relu activation function, but rather contains a skip connection as well as demonstrated in Figure 4.3 [8]. We have seen in many research findings that residual blocks provide CNNs with additional information as the addition of residual blocks retains the fundamental network structure of our original model while also providing skip connections to skip certain convolutional layers that may have filter sizes that are not necessary towards learning. This makes it easy to see its potential use in deep regresssion problems involving CNNs considering its' flexible structure. We

see in [9] that Residual connections tend to stabilize the results of deep regression problems that naturally tend to be less stable due to the nature of "memorization" involved because of the relative ratio of weights to data in these types of problems. We propose that using residual networks could also lead to improvements to the oscillation forecasting approaches seen in *Nature* and *atmosphere* [4, 3].

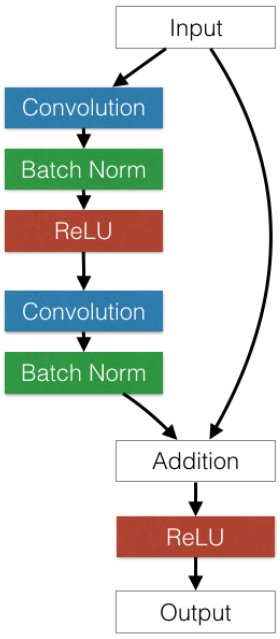


Figure 4.3: Resnet Block

Given that many of these CNN oscillation forecasting models use lagged gridded data [4, 1, 6], or lagged versions of the signal itself [3], it seems natural that we propose using a Convolutional LSTM as introduced by [24] as it takes advantage of the spatial components of the grid take while also taking advantage of the time components as seen in an LSTM [24]. We weren't able to have the time or resources to train one effectively in this work. We also propose additional transfer learning [26], potentially on other oscillations, but the approach that we believe has the most potential is as follows: We would replace the final output of each time step of the LSTM with a new output set of layers, borrowing from the upsampling technique of the CNN Regression Model introduced in [10]. We would give the model the task of predicting SST from lagged SST, Heat Content and Pressure grids. This information would provide context that would likely be useful when predicting oscillations. Since transfer learning tasks have seen success with similar model structures and problems in other fields as well as success with this same problem, we argue that additional

transfer learning would help if focused on the correct task [26].

4.3.2 Oscillation Discovery

The Future Work is quite clear in this section given the data constraints that we had. Given spatially distributed temperature signals, one would be able to perform the complete experiment proposed in Chapter 1, while our arguments in Section (4.1) would still hold with respect to its ability to bottleneck the desired information. That would ultimately be an important source of transfer learning and additional understanding in future work of this problem.

Also, continuing on our hypothesis in the preceding subsection, we also argue that a model choice for the representation learning providing residual connections would potentially enhance the features by essentially providing an adjustable hypothesis space that exceeds the current hypothesis space of the model architectures. As briefly mentioned in (4.1), our results that show that the direct autoencoder yields better results may be due to a lack of LSTM structure when training the lagged autoencoder and skipgram models. When using skipgram in [21], it wasn't necessary to use an LSTM to yield good results; however, give the size of the grids used, it may be necessary in this problem to yield better results than the direct autoencoder model.

Chapter 5

Conclusion

The first two problems solved in this work come together naturally. The first problem focuses on time-synchronization using phase variables derived from analytic signals. A key contribution we've made is to use the phase features of global indices, interpreted as oscillators, to provide a synchronization mechanism for correcting the phase of the temperature forecasts. This was successful, leading to improvements even when we considered "extreme" statistics of the temperature signals. The second problem focuses on spatial-equilibration between sites that looks at the relative correlations of primary and proxy variables; this is thus an amplitude correction model. Both contributions adapt and recompose the forecast signal. Time and space corrections can be applied as a post-processing step to any nominal forecasting model that is localized in one or both of these dimensions. Thus, the proposed approaches are general and not limited to long-range weather forecasting per se.

The follow on question for detecting oscillators from data or model simulations also naturally emerges. Effectively synchronizing clocks of quantities of interest to latent features are not just useful in their own right, but might lead to additional indicators of climate dynamics. This was the subject of Chapter 4.

Looking forward to the future, we believe that the following areas can be expanded on within this research with potential success. Adding a time-component to λ and J within the site-correction model could have great benefits within site-prediction networks on a daily or hourly scale. Adding more locations also should increase the

site-correction models' effectiveness given the models reliance on cross-correlations.

The application of phase correction to numerical weather simulation models [34] is something that would be interesting to look at, as phase error is considered to be a potentially large component of numerical weather prediction model errors. The expansion of the work in Chapter 4 as it could be more complete given more time and computational resources – as our resource precluded us from training the lagged models for ideal periods of time – and there are endless tasks to experiment with and evaluate given enough resources.

Applying these correction-techniques to precipitation forecasting would be interesting as well; we were unable to develop a model accurate enough at long range precipitation to be able to test our correction methods.

5.1 Acknowledgements

I would like to thank Dr. Sai Ravela of the MIT Earth Signals and Systems Group for his guidance. I would also like to thank Fulvio Fabrizi of MIT Lincoln Laboratory for his suggestion in using oscillations for long-term weather forecasting, as well as Tom Reynolds, also of Group 43 in MIT Lincoln Laboratory, for providing data. This work was performed as part of subaward PO 7000452592 under Air Force contract FA8702-15-D-0001 and ONR grant N00014-19-1-2273. The contents are those of the author(s) and do not necessarily represent the official views of, nor an endorsement from the Navy or Air Force or the Government.

Bibliography

[1] Benjamin A. Toms, Karthik Kashinath, Prabhat and Da Yang. Deep Learning for Scientific Inference from Geophysical Data: The Madden-Julian Oscillation as a Test Case, 2019; arXiv:1902.04621.

[2] Yunjie Liu, Evan Racah, Prabhat, Joaquin Correa, Amir Khosrowshahi, David Lavers, Kenneth Kunkel, Michael Wehner and William Collins. Application of Deep Convolutional Neural Networks for Detecting Extreme Weather in Climate Datasets, 2016; arXiv:1605.01156.

[3] Yuan, Shijin & Luo, Xiaodan & Mu, Bin & Li, Jing & Dai, Guokun. Prediction of North Atlantic Oscillation Index with Convolutional LSTM Based on Ensemble Empirical Mode Decomposition, 2019, Atmosphere. 10. 252. 10.3390/atmos10050252.

[4] Ham, Y., Kim, J. & Luo, J. Deep learning for multi-year ENSO forecasts. Nature 573, 568–572 (2019) doi:10.1038/s41586-019-1559-7

[5] Matsuoka, Daisuke & Nakano, Masuo & Sugiyama, Daisuke & Uchida, Seiichi. (2018). Deep learning approach for detecting tropical cyclones and their precursors in the simulation by a cloud-resolving global nonhydrostatic atmospheric model. 5. 10.1186/s40645-018-0245-y.

[6] Broni-Bediako, Clifford & Katsriku, Ferdinand & Unemi, Tatsuo & Shinomiya, Norihiko & Abdulai, Jamal-Deen & Atsumi, Masayasu. (2018). El niño-southern oscillation forecasting using complex networks analysis of LSTM neural networks.

[7] Barnston AG, Tippett MK, L'Heureux DeWitt DG (2012), Skill of Real-Time Seasonal ENSO Model Predictions during 2002–11: Is Our Capability Increasing?. Bull. Amer. Meteor. Soc.

[8] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke and Alex Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, 2016; arXiv:1602.07261.

[9] Stéphane Lathuilière, Pablo Mesejo, Xavier Alameda-Pineda and Radu Horaud. A Comprehensive Analysis of Deep Regression, 2018; arXiv:1803.08450.

[10] Jun Yuan, Bingbing Ni and Ashraf A. Kassim. Half-CNN: A General Framework for Whole-Image Regression, 2014; arXiv:1412.6885.

[11] Aapo Hyvarinen, Hiroaki Sasaki and Richard E. Turner. Nonlinear ICA Using Auxiliary Variables and Generalized Contrastive Learning, 2018; arXiv:1805.08651.

[12] Eftekhari, Armin & Forouzanfar, Mohamad & Moghaddam, Hamid & Alirezaie, Javad. (2010). Block-wise 2D kernel PCA/LDA for face recognition. Information Processing Letters. 110. 761-766. 10.1016/j.ipl.2010.06.006.

[13] Yoshua Bengio, Aaron Courville and Pascal Vincent. Representation Learning: A Review and New Perspectives, 2012; arXiv:1206.5538.

[14] Michael Tschannen, Olivier Bachem and Mario Lucic. Recent Advances in Autoencoder-Based Representation Learning, 2018; arXiv:1812.05069.

[15] Deep shared representation learning for weather elements forecasting]

[16] Yuxiu Hua, Zhifeng Zhao, Rongpeng Li, Xianfu Chen, Zhiming Liu and Honggang Zhang. Deep Learning with Long Short-Term Memory for Time Series Prediction, 2018; arXiv:1810.10161.

[17] What Regularized Auto-Encoders Learn from the Data-Generating Distribution

[18] Daniel C. Castro, Jeremy Tan, Bernhard Kainz, Ender Konukoglu and Ben Glocker. Morpho-MNIST: Quantitative Assessment and Diagnostics for Rep-

resentation Learning, 2018, Journal of Machine Learning Research 20 (2019); arXiv:1809.10780.

[19] Jun Li, Daoyu Lin, Yang Wang, Guangluan Xu and Chibiao Ding. Deep Discriminative Representation Learning with Attention Map for Scene Classification, 2019; arXiv:1902.07967.

[20] Yu-An Chung and James Glass. Speech2Vec: A Sequence-to-Sequence Framework for Learning Word Embeddings from Speech, 2018; arXiv:1803.08976.

[21] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in NIPS, 2013.

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018; arXiv:1810.04805.

[23] http://essg.mit.edu/blog/longrange-weather-forecasting-machine-learning-khulna-example (Confidential)

[24] 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, 2015; arXiv:1506.04214.

[25] noaa.gov

[26] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang and Chunfang Liu. A Survey on Deep Transfer Learning, 2018; arXiv:1808.01974.

[27] Chen T *et al.*, XGBoost: eXtreme Gradient Boosting `github.com/dmlc/xgboost`

[28] Peter Bauer, Alan Thorpe, and Gilbert Brunet. The quiet revolution of numerical weather prediction. Nature, 525(7567):47–55, 2015

[29] Sønderby et al. MetNet: A Neural Weather Model for Precipitation Forecasting. 2020.

[30] Hribar, Rok and Potočnik, Primož and Šilc, Jurij and Papa, Gregor. A comparison of models for forecasting the residential natural gas demand of an urban area, in Energy, 2018.

[31] Das et al. Chapter 4. Weather and Climate Forecasts for Agriculture.

[32] Dempster, A., Laird, N., Rubin, D. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society. Series B (Methodological), 39(1), 1-38. Retrieved May 17, 2020, from www.jstor.org/stable/2984875

[33] Kalman, Rudolph Emil. A New Approach to Linear Filtering and Prediction Problems, 1960.

[34] Ravela, S., Emanuel, K., and McLaughlin, D.: Data assimilation by field alignment, Physica D, 230, 127–145, 2007

[35] Matthias Bauer, Mark van der Wilk, and Carl Edward Rasmussen. Understanding probabilistic sparse gaussian process approximations. In Advances in neural information processing systems, pages 1533–1541, 2016.

[36] Dura-Bernal S, Wennekers T, Denham SL (2012) Top-Down Feedback in an HMAX-Like Cortical Model of Object Perception Based on HierarchicalBayesian Networks and Belief Propagation. PLoS ONE 7(11): e48216. doi:10.1371/journal.pone.0048216

[37] Bishop, Christopher M. Pattern Recognition and Machine Learning. New York: Springer, 2006.