8-2021

Learning-Based Modeling of Weather and Climate Events Related To El Niño Phenomenon via Differentiable Programming and Empirical Decompositions

Justin Le
Chapman University, jusle@chapman.edu

Learning-Based Modeling of Weather and Climate Events Related To El Niño Phenomenon via Differentiable Programming and Empirical Decompositions

A Dissertation by

Justin A. Le

Chapman University

Orange, CA

Schmid College of Science and Technology

Submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computational and Data Sciences

August 2021

Committee in charge:

Hesham M. El-Askary, Ph.D., Chair

Mohamed Allali, Ph.D.

Thomas C. Piechota, Ph.D.

Daniele C. Struppa, Ph.D.

CHAPMAN UNIVERSITY
SCHMID COLLEGE OF SCIENCE AND TECHNOLOGY
Computational and Data Sciences

The dissertation of Justin A. Le is approved.

Hesham El-Askary

Digitally signed by Hesham El-Askary
DN: cn=Hesham El-Askary, o=Chapman University, ou=Schmid College of Sciene and Technology, email=elaskary@chapman.edu, c=US
Date: 2021.06.11 21:34:17 -07'00'

Hesham M. El-Askary, Ph.D., Chair

Mohamed Allali

Digitally signed by Mohamed Allali
Date: 2021.06.14 16:58:53 -07'00'

Mohamed Allali, Ph.D.

Thomas Piechota

Digitally signed by Thomas Piechota
Date: 2021.06.15 07:54:46 -07'00'

Thomas C. Piechota, Ph.D.

Daniele Struppa

Digitally signed by Daniele Struppa
Date: 2021.06.16 09:15:44 -07'00'

Daniele C. Struppa, Ph.D.

May 2021

Learning-Based Modeling of Weather and Climate Events Related To El Niño Phenomenon via Differentiable Programming and Empirical Decompositions

by Justin A. Le

ACKNOWLEDGEMENTS

The path to the completion of the document you hold before you has been a winding one, full of much struggle and winding paths. One thing I know is that none of it would be possible without the help of the many supportive people in my life. I truly cannot claim I have done anything close in my life to deserving the support I have received.

Foremost, I would like to thank my advisor Dr. Hesham El-Askary, who believed in me from the beginning, the first day I stepped onto Chapman campus nearly 7 years ago. He provided the foundation upon which this entire work you have in your hands is built. Through all the ups and downs, he has never failed to offer insight and expertise, directing me in a way to help me realize my full potential. Throughout all the apparent dead ends, he was always happy to put in the time to help me research and find new avenues. I consider him not only an unrivaled expert and mentor, but also a role model in how I want to relate to the people I may have the opportunity to help in the future. In addition, each of the members of my thesis committee (Dr. Mohamed Allali, VP for Research Thomas Piechota, President Daniele Struppa) have left a very personal mark on my academic journey with their guidance, wisdom, and support. Professor Allali, as one of my first professors and advisors, set a high bar for how faculty can personally invested students. Dr. Piechota always supported my work and pushed my aspirations. President Struppa, my first contact at Chapman as an applicant, profoundly reshaped how I saw the relationship between mathematics, research, and making an impact in the world with a single conversation.

I also owe so much to my family, especially my parents, my role models who have always worked to instill in me a sense of curiosity, wonder, and fascination in the world around me. So

much of who I am today, and the skills that allow me to succeed, I can trace back to them. This work would not have been possible without them and their own tireless work, day by day, to build a life for me and my siblings even after all their struggle as young immigrants of the Vietnamese diaspora. I thank them for their patience and understanding through this long process, and for their loving support.

I would like to give very special thanks to Tiffany Nguyen – we entered this journey together, working through grad school side by side. Without her support and confidence in me this entire time, we would not have been able to see this day where we can celebrate accomplishing our dreams. Thank you for always inspiring me to be my best, through everything, and for always being there. I look forward to all our milestones we will accomplish together in the future!

And of course, I would like to thank my God, his faithfulness, and his irresistible, unrelenting love. If I ever boast, may it only be in his strength and sacrifice that has redeemed me, in the gifts he has given me so that I may do my part in carrying a light to the world.

VITA

Justin Le studied Physics as an undergraduate student at the University of California San Diego, with a specialization in Computational Physics. As an undergrad research assistant, he dived into the world of experimental and computational condensed matter and fell in love with the art of computational science and merging theoretical computer science, far-flung mathematical principles, and real-world problems. He began the doctoral program in Computational and Data Sciences in 2014, beginning research on applying mathematical principles to guide computational research in applications with earth systems sciences and climate science. Over the years, he has gained a reputation in the Haskell community, built around a programming language able to express the elegance of the mathematical computational ideals he was pursuing, by collaborating and releasing many open-source libraries pushing the ecosystem and publishing a large corpus of educational material. Through this, and his work in earth systems science, he has been invited to many talks, and continues to be driven by a pursuit of the application of new mathematical principles to computational data.

LIST OF PUBLICATIONS

Le, J. A., & El-Askary, H. M. (2016). Forecasting Interactions Between ENSO and Extreme Drought with Recurrent Neural Networks. *Asia Oceania Geosciences Society*.

Le, J. A. (2017). Introduction to Dependently Typed Programming. *Kiev FProg*.

Le, J. A., El-Askary, H. M., Allali, M., & Struppa, D. C. (2017). Application of recurrent neural networks for drought projections in California. *Atmospheric Research*, *188*, 100–106. https://doi.org/10.1016/j.atmosres.2017.01.002

Le, J. A. (2019). Recurrent Neural Networks and the study of Internal Node Activations for the understanding of Greater California Drought and El Niño Interactions. *American Geophysical Union*.

Le, J. A., El-Askary, H. M., Allali, M., Sayed, E., Sweliem, H., Piechota, T. C., & Struppa, D. C. (2020). Characterizing El Niño-Southern Oscillation Effects on the Blue Nile Yield and the Nile River Basin Precipitation using Empirical Mode Decomposition. *Earth Systems and Environment*, *4*(4), 699–711. https://doi.org/10.1007/s41748-020-00192-4

ABSTRACT

Learning-Based Modeling of Weather and Climate Events Related To El Niño Phenomenon via Differentiable Programming and Empirical Decompositions

by Justin A. Le

This dissertation is the accumulation of the application of adaptive, empirical learning-based methods in the study and characterization of the El Niño Southern Oscillation. In specific, it focuses on ENSO's effects on rainfall and drought conditions in two major regions shown to be linked through the strength of the dependence of their climate on ENSO: 1) the southern Pacific Coast of the United States and 2) the Nile River Basin. In these cases, drought and rainfall are tied to deep economic and social factors within the region. The principal aim of this dissertation is to establish, with scientific rigor, an epistemological and foundational justification of adaptive learning models and their utility in the both the modeling and understanding of a wide-reaching climate phenomenon such as ENSO. This dissertation explores a scientific justification for their proven accuracy in prediction and utility as an aide in deriving a deeper understanding of climate phenomenon. In the application of drought forecasting for Southern California, adaptive

learning methods were able to forecast the drought severity of the 2015-2016 winter with greater accuracy than established models. Expanding this analysis yields novel ways to analyze and understand the underlying processes driving California drought. The pursuit of adaptive learning as a guiding tool would also lead to the discovery of a significant extractable components of ENSO strength variation, which are used with in the analysis of Nile River Basin precipitation and flow of the Nile River, and in the prediction of Nile River yield to $p=0.038$. In this dissertation, the duality of modeling and understanding is explored, as well as a discussion on why adaptive learning methods are uniquely suited to the study of climate phenomenon like ENSO in the way that traditional methods lack. The main methods explored are 1) differentiable Programming, as a means of construction of novel self-learning models through which the meaningfulness of parameters arises from emergent phenomenon and 2) empirical decompositions, which are driven by an adaptive rather than rigid component extraction principle, are explored further as both a predictive tool and as a tool for gaining insight and the construction of models.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS

Abbreviation	Meaning
AMO	Atlantic Multidecadal Oscillation
ANN	Artificial Neural Network
ARMA	Auto Regressive Moving Average
EMD	Empirical Mode Decomposition
HHT	Hilbert-Huang Transform
IEVI	Inter-El Niño Variation Index
IMF	Intrinsic Mode Function
LSTM	Long-term Short-term Memory Network
PDI	Palmer Drought Index
PZI	Palmer Z-Index
RNN	Recurrent (Artificial) Neural Network
SOI	Southern Oscillation Index

1 Introduction

1.1 Goals of this Research

In the time of changing climatological processes, it is more important than ever to augment traditional scientific models with what assistance computer-aided modeling can bring. Sustainable development in the face of changing climate is critical on a humanitarian level, and even our limitations in our understanding of climatological processes like El Niño and our inability to properly model, study, and predict event outcomes poses critical problems to economic and agricultural agriculture.

Amid this change, other fields have seen a surge in the success of learning-based models such as Artificial Neural Networks and empirical decomposition. These adaptive methods are just now beginning to be used to great effect in climate modeling applications. Born out of research in the 1970's, these methods were, at the time, limited by the computational power available and accessible at that time. However, as accessible computational power began to grow to a level to support advanced empirical methods, their usage in the sciences has grown. Their success is not without controversy. In climate modeling, especially, the role that machine learning and related models play alongside traditional scientific models is still being debated, discovered, and refined. Several characteristic limitations of adaptive models – such as their "black-bock" nature, their apparent inability to explain their results, and their probabilistic nature – are often barriers in their acceptance as legitimate or useful tools. In this dissertation, we discuss what exactly the role of adaptive models – if any – must play in the landscape of climate science in the future, considering true cost in either case.

This dissertation first the application of such techniques to studying El Niño Southern Oscillation phenomenon as it relates to drivers of drought, runoff, and precipitation. These issues are of very critical social, environmental, and geopolitical importance, often costing millions in damage, displaying many populations, and driving famines and other humanitarian crises. Therefore, the application of learning-based models to not only modeling, but also helping to understand these phenomena are of high scientific and humanitarian priority.

2 Background

2.1 Drought

Beginning with late 2011, California has been facing its most intense and severe drought since historical recordings began in 1895 (Richman et al., 2015). This drought is often compared to other significant droughts in California history, including the particularly long-lasting Dust Bowl drought of the late 1920s to early 1930s and the droughts in 1976 - 1977 and the late 1980s to early 1990s (Robeson, 2015).

Short-term drought on its own is has pervasive negative impact on local ecosystems and economy, and also has impacts on issues of public health and recreation. However, long-term drought poses unique problems with regards to the loss of groundwater (which is effectively a non-renewable resource) and sinking of ground elevation due to depleted groundwater reservoirs.

Griffin et al. (2014) found that although the recent drought is not the longest drought in recorded history, it is the singular most extreme one when comparing rainfall deficits. Recently, over 70% of the state suffered extreme and exceptional drought where normally wet seasons of the yearly climate cycle have been underwhelming (Richman et al., 2015; Robeson, 2015). The uniqueness of this drought season was confirmed by analyzing other drought predictors such as abnormal temperatures (Jeong et al., 2014; Shukla et al., 2015). Howitt et al. (2015) estimated the economic damage to the 2015 agriculture caused by the drought to be $1.8 billion, with a total statewide economic cost for the same period of $2.7 billion. This is a 23% increase compared to the $2.2 billion in losses incurred in 2014 due to similar drought conditions (Howitt et al., 2014).

Furthermore, it is also estimated that as many as 21,000 agricultural and related jobs were lost in 2015, up 23% from 17,000 jobs lost in 2014. According to the Center for Watershed Sciences at Davis, an additional $2.8 billion and 21,400 jobs are projected to be lost due to drought in 2016 if conditions persist. Apart from socioeconomic losses, environmental losses are also anticipated. For example, as many as 58 million trees are in severe risk of dying in 2016 causing disastrous impacts on California ecology (Asner et al., 2015). Cook et al. (2015) suggests that the current drought trends could be the beginning of a larger drought taking place over the first half of the entire 21st century.

2.2 El Niño

On the other hand, the winter leading into 2016 was expected to bring heavy rain resulting from another powerful weather phenomenon, namely the El Niño Southern Oscillation (ENSO). ENSO affects tropical meteorological fields yet its influence is exerted by changing the largescale Walker circulation and associated convection and precipitation patterns (Slemr et al., 2016). ENSO is a periodic fluctuation in global climate with of a period between 2 to 7 years, and is strongest throughout the boreal winter season of peak years (Capotondi et al., 2014).

The ENSO effect on different regions of the globe is highly varied, but in California, strong El Niño seasons often manifest as periods of extreme and anomalous precipitation (El-Askary et al., 2004; El-Askary et al., 2013). A recent past El Niño event, during the winter of 1997 – 1998, led to extreme flooding in Los Angeles and caused multiple deaths and billions of dollars in flooding and related damages (Changnon, 1999). The (2015 – 2016) El Niño season has been projected by many to be abnormally strong (Hoell et al., 2016; Zhenya et al., 2015; Climate.gov, 2015).

However, the exact strength of that season is still investigated in the context of the previous strong (1997-1998) El Niño season. It is noteworthy that El Niño drives on average only about 6% of the precipitation variability in California (Savtchenko et al., 2015). It is not guaranteed that El Niño has the potential to resolve the accumulated deficit of precipitation, which is presently equivalent to an entire year of precipitation. However, chances may improve in a strong El Niño that also coincides with the peak of the wet season in California (December-February). Moreover, there is a large geographical difference in the anticipated impact of El Niño on California (Piechota and Dracup 1996; Piechota et al., 1997). Being able to accurately gauge the El Niño intensity is important for agricultural, development, and public safety planning applications. Despite several record examples of strong El Niño seasons, we believe that there are complex interactions of ENSO effects with a drought as intense as the current one affecting California.

It is known that California prolonged drought resulted from a multi-year precipitation decline and anomalous warm temperatures, that in turn resulted from anomalously persisting high pressure in the East Pacific, which significantly changed the normally observed atmospheric circulation patterns. In this light, this research attempts to forecast the affect that the upcoming El Niño season will have on the continuing drought.

3 Differentiable Programming

3.1 General Overview

Differentiable Programming is a powerful class of adaptive learning models based on the principle of training by gradient descent (LeCunn, 2018). It is the automation of model trainability, allowing for an algorithmic process (through automatic differentiation) for the training of any arbitrary model. It is analogous to the method of Maximum Likelihood Estimation for statistical modeling – the model is expressed as a function, and a mechanical and automatable process is derivable for transforming that function into a trainable model through gradient descent and sample-based optimization.

The Fundamental Model

At its core, a model can be described as a parameterized function from input to output.

$$f_p(x) = y$$

The important feature of the model is that, for every choice of parameterization p, a different function $f_p(x)$ is produced. In this light, the training (or estimation) of a model is the process of picking a p that provides the most suitable $f_p(x)$ function.

A classic example of this is linear regression: linear regression consists of the function $f(x) = \beta + \alpha x$, parameterized on α and β. The parameters are α and β, the input is x, and the output is y.

We can reframe $f_p(x)$ is as a partially applied $f(p, x)$. To show this, we start with the final type:

$$f: (P \times A) \to B$$

And curry the function, to recover the original formulation:

$$f: P \to (A \to B)$$

The fundamental goal of model training, therefore, is to identify the correct p to use, based on pairs of observables (x, y). It is the art of picking the best p to explain observation. This quantification of the "best" can be described using a loss function. One common example is the squared error:

$$(f_x(p) - y_x)^2$$

In general, picking the best parameter for the model involves picking the p that minimizes the relationship:

$$\text{loss}\big(y_x, f_x(p)\big)$$

Fundamental Advantage

The major revelation we gain from using this formulation is that f itself (and $loss$) may automatically differentiable (using automatic differentiation techniques available in many programming languages and frameworks). Furthermore, function composition (and higher-ordered curried function composition) is closed with respect to differentiability. And in the specification of f as a function within a computational context, we arrive at its gradient without

extra work – and with a computable gradient, one may apply a method such as Stochastic Gradient Descent.

If we get the *gradient* of the loss with respect to p ($\nabla_p \text{loss}(f_x(p), y_x)$), there is now a principled method of stochastic gradient descent:

1. Start with an initial guess at the parameter
2. Look at a random (x, y_x) observation pair.
3. Compute the gradient $\nabla_p \text{loss}(f_x(p), y_x)$ of our current p, which tells us a direction we can "nudge" p in to make the loss smaller.
4. Nudge p in that direction
5. Repeat from #2 until satisfied

As normal parameterized functions, such functions compose.

$$\begin{aligned} f_P &: B \to C \\ g_Q &: A \to B \\ (f \circ g)_{P \times Q} &: A \to C \end{aligned}$$

And this is closed with respect to differentiability.

A curious case is observable in multivariate linear regression:

$$f_{AB}(\mathbf{x}) = B\mathbf{x} + A$$

We can note that this is exactly the internal part of a fully-connected Feed Forward Artificial Neural Network layer, from literature. If this is post-composed with a non-linear (unparameterized) activation function, this is exactly an FF ANN Layer. When multiple such

functions are composed together, this is a feed-forward artificial neural network. And through principles of automatic differentiation, the gradient is derivable for free, and access to SGD is instantly available as well.

Time-Series Models

A logical step from here (that is relevant to the topic of this dissertation) would be the adaptation of this framework to time-series data. One straightforward attempt may be to directly parameterize on time:

$$f_p(x, t) = y$$

However, this data model is not quite desirable for expressing an empirical time series data set, because it does not encode the notation of causality. For a causal structure, $f(t_1)$ cannot explicitly depend on any $t_2 > t_1$. One method of embedding causality into the structure of a model is to instead thread explicit state:

$$f_p(x, s_{\text{old}}) = (y, s_{\text{new}})$$

$$f: (P \times A \times S) \rightarrow (B \times S)$$

By careful selection of state and input, we can construct any causality-preserving time series model explicitly. For example, an autoregressive model with degree 2 can be expressed as:

$$\begin{aligned} s_t &= x_t \\ y_t &= c + \phi_1 s_t + \phi_2 s_{t-1} \end{aligned}$$

with parameters ϕ_1, ϕ_2. In the explicit functional form, this is written as:

$$f_{c,\phi_1,\phi_2}(x,s) = (c + \phi_1 x + \phi_2 s, x)$$

A slightly more complex example is the fully-connected Recurrent Neural Network layer from literature, which has state dependence:

$$\begin{aligned} s_t &= \sigma(y_t) \\ y_t &= W_x \mathbf{x}_t + W_s \mathbf{s}_{t-1} + \mathbf{b} \end{aligned}$$

Parameterized on W_x and W_s matrices and vector $\mathbf{b}$. When expressed in explicit functional form, this is:

$$f_{W_x,W_s,\mathbf{b}}(\mathbf{x},\mathbf{s}) = \left(W_x\mathbf{x} + W_s\mathbf{s} + \mathbf{b}, \sigma(W_x\mathbf{x} + W_s\mathbf{s} + \mathbf{b})\right)$$

While this stateful encoding is useful on its own right, it is unsatisfactory in its unification with the previous formulation of stateless models. While automatic differentiation can be applied, the application of stochastic gradient descent or related training algorithms are not directly possible. We would like a way to unit both stateful and stateless models under the same framework.

A breakthrough can emerge when we begin treating these functions as simply differentiable functions, where we are able to perform manipulations on them in any way closed to differentiability.

For example, consider a method $R[\_]$ of promoting a stateless function into a stateful function by closing off a component of its input as state:

$$\begin{aligned} f_P &: (A \times B) \to B \\ R[f_P]_B &: A \to B \end{aligned}$$

$R[\_]$ here transforms a parameterized model on a tuple into a parameterized stateful model, with its state as the previous output. This previous output is fed to the original function, and its input becomes only the component of the tuple that is not pre-determined by previous state.

At the initial onset, this shows promise as a useful tool, because applying it to linear regression yields auto-regressive models directly. It can be therefore said that an auto-regressive model is simply the application of $R[\_]$ to linear regression.

Another method of promoting a stateless model to a stateful model is to feed back a history of its recent outputs:

$$\begin{aligned} f_P &: \mathbb{R}^{n+1} \to B \\ L[f_P]_{\mathbb{R}^n} &: \mathbb{R} \to B \end{aligned}$$

The input then becomes only a single component of the original vector; the other components are fixed to be the history of the function's previous input. A testament of this method's significance is the fact that applying $L[\_]$ to multivariate linear regression yields moving-average models. It can therefore be said that a moving average model is simply the application of L to a multivariate linear regression. Furthermore, the application of L and R together (sequentially) on a multivariate linear regression yields Autoregressive Moving-Average models.

Model Unification

At this point we have provided a method of conversion from stateless models to stateful models, in a way that we can construct a wide class of stateful models as simply mathematical transformation of stateless models. However, our original application (training via stochastic gradient descent) is not applicable to stateful models. We must therefore now present a method of conversion in the other direction, in order to fulfil the original promise of this formulation.

First, consider two ways of transforming a stateful model $f: A \to B$ over a sequence of A^n, presented as a time series. One would sequence the output of the state from the first A as the input state for the second, etc. One variation in choice would be to transform it into either $S[f]: A^n \to B^n$ and collect each resulting B, or as $S[f]: A^n \to B$ and only keep the final output. In this sense, we have converted a stateful function on a single input into a stateful function on a time-series slice of input.

The result of this is the explicit statefulness of the resulting function is now embedded directly into the function itself, and no longer in the explicit input and output state. At this point, we may convert our final stateful $(A^n \times S) \to (B^n \times S)$ into a stateless function by either:

- Simply dropping the state, yielding $A^n \to B^n$
- Treating the state as a trainable parameter, yielding $(A^n \times S) \to B$. This is possible because differentiability is preserved over tuples.

The successive application of extending the function over a time series and then removing the state through one of the two methods above effectively turns a stateful model $A \to B$ on single points of input into a stateless model $A^n \to B^n$ on a time series slice of inputs. This final form is completely differentiable and trainable using stochastic gradient descent. This transformation in literature was independently formulated as "backpropagation over time". In this formulation, it requires no explicit extra derivation, since it is an automatic result of simple manipulations of differentiable functions through operations closed with respect to differentiability.

Now, let us go over the class of models – artificial neural networks – that this method subsumes, as useful tools in their own rights.

3.2 Artificial Neural Networks

Artificial neural networks (ANNs) are a class of models roughly said to be inspired by "biological neural networks" – that is, the mechanisms and structure of neurons in an animal brain. In its modern interpretation, the definition of an artificial neural network has come to encompass a wide family of parameterized functions that are composed of several internal composed parameterized functions sequenced after each other (or in parallel with each other), where each layer of processing typically is interpreted as transformations of interconnected nodes, which output signals based on a weighted sum (or linear combination) of input signals from the previous input layer.

The process of training an ANN involves selecting the proper set of weights to parameterize each layer such that the inputs and outputs of the network properly model the function or phenomenon in question. A neural network designed for facial recognition would subject an input image to several linear and non-linear transformations parameterized in such a way that the output produces a vector identifying the face in the input image.

The strength of the ANN approach comes from the mathematical simplicity of the internal functions that compose a network; their analytic structure allows one to easily differentiate the error of a network's output with respect to each internal parameter, allowing for several forms of gradient descent and hill-climbing to be effective in optimizing a network for an ideal set of weights and parameters. Given a source of inputs and their expected outputs, one may compute the optimal parameterization for a network to model the relationship exactly by iterating a greedy hill-climbing strategy between different inputs and network errors.

The wide variety of different approaches to the implementation of artificial neural networks in practice comes from the wide array of possible configurations and properties of such internal layers and functions. By selecting different types of internal layers, one may adapt an ANN for a wide variety of problems.

Common applications of ANNs involve feed-forward networks trained with backpropagation (Cigizoglou et al., 2004). ANNs have been applied several times to rainfall and precipitation forecasting (Nastos et al., 2014; Bodri et al., 2000; Luck et al., 2000; Silverman et al., 2000; Sakellariou et al., 2004; Cigizoglou et al., 2004; Moustris et al., 2011). Badjate et al. (2009) explored RNNs for predicting precipitation and chaotic time series like sun spot occurrences. Different ANN topologies has been used to forecast extreme precipitation events and total precipitation levels, including recurrent Elman Networks (Maqsood et al., 2004) and Long-Term Short-Term Recurrent models (Nastos et al., 2014).

Artificial Neural Networks — both their feed-forward and recurrent varieties — have been used extensively in literature, to much success. This research attempts to show yet another important application for a pivotal winter season, to add to the body of research proving their success. In addition to this, however, this research also aims to show that the application of artificial neural networks (especially recurrent ones) can also have profound impacts on our understanding of these systems, and can be used as an aid to drive scientific development and research and help the discovery the of new indices and phenomenon.

3.3 Feed-Forward Neural Networks

The traditional feed forward neural network can be thought of as a universal function approximator, and is shown to be able to approximate any function $f: \mathbb{R}^n \to \mathbb{R}^m$ to arbitrary

precision with respect to $L^p(\mu)$ performance criteria (Hornik, 1990). Feed-forward ANNs consist of layers of neurons which receive weighted inputs from the outputs of preceding layers. Feed-forward ANNs have seen great success in fields like pattern recognition and classification.

The ANN approach to modeling functions involves successive processing steps (known as "layers") structured as parameterized linear and non-linear functions. Parameterized linear functions $f: \mathbb{R}^h \to \mathbb{R}^j$ and simple vectorized analytic functions are common choices for each layer. In the simplest case, with layer $h_i(w_i): \mathbb{R}^h \to \mathbb{R}^j$ (a layer parameterized by w_i), a feed-forward artificial neural network with n layers is the composition of functions $h(w_i)$:

$$N(w_1, w_2, \dots, w_n) = h_1(w_1) \circ h_2(w_2) \circ h_3(w_3) \circ \dots \circ h_n(w_n): \mathbb{R}^N \to \mathbb{R}^M$$

In the most common application, each internal layer can be chosen to be a *perceptron*: A parameterized affine transformation of its input, with outputs mapped by a (non-parameterized) differentiable function. In the perceptron model, a layer taking h inputs to j outputs would be parameterized by a $\mathbb{R}^{h\times(j+1)}$ affine transformation matrix and would compute its output as:

$$o_h = f(W_{h,0} + \sum_j W_{h,j}\, i_j)$$

With input vector i, parameterizing weight matrix W, and differentiable function $f: \mathbb{R} \to \mathbb{R}$.

It can be seen that the partial derivative of each matrix element in W with respect to some function on is easily computable in a straightforward way.

The activation function f is chosen to provide for a way for the model to express a wide range of non-linear functions. In practice, f is chosen to be a sigmoidal function in order to model the

biological inspiration of a neural network – under a given cut-off, the output is low, but after a cut-off the output is high. This is said to be modeled after the behavior of neural connections in biological neural networks, which only output after a given input threshold is received.

In practice, one should chose an f whose derivative has a non-zero magnitude in significant regions. This allows the partial derivative of each input weight with respect to the error of to be computable and non-zero, which provides more optimal behavior for the optimization methods commonly used. In the following subsections we will discuss the network optimization and other related issues.

Network Optimization

The advantage of structuring a computation in this manner is that such a chain of functions is easily differentiable using a simple multivariate generalization of the chain rule. Furthermore, this process can be done in a mechanical way through techniques collectively known as automatic differentiation.

Simple differentiation by hand (through propagation of a multivariate generalization of the chain rule) is possible in most situations for artificial neural networks, and in practice, most implementations of feed-forward neural networks compute gradients in this manner. However, every major software framework providing implementations of general artificial neural network provides mechanisms through automatic differentiation for computing general ANNs of general internal structure and parameterization.

Because of the layered structure of artificial neural networks, the Jacobian matrix of the results of the error of the neural network's output with respect to its parameters is computationally

simple, and, in most cases, extremely fast. Computing the Jacobian matrix with respect to an error function has the same time complexity as applying the function that the ANN encodes, itself. In fact, these two processes (computing the result of the encoded function, alongside its gradient) can be computed in tandem, at the same time, in order to reduce computation time.

The process of parameter selection then becomes a stochastic optimization problem on a differentiable function $\mathbb{R}^n \to \mathbb{R}^m$, which is a well-explored problem in the field of optimization. Somewhat surprisingly, general optimization algorithms have been shown to be unexpectedly effective. The efficacy of general optimization algorithms for the selection of optimal parameters for artificial neural networks plays a large role in the rising popularity of their application in various fields.

The canonical naïve approach involves a stochastic gradient descent, in which network parameters are shifted slightly in the direction that produces the lowest error over a given training set. The algorithm samples different known input-output pairs, and computes the error of each network output with respect to the known output.

This error is often computed as a sum-of-squared differences:

$$E(x_i, y_i) = (f(x_i) - y_i)^2$$

This is a common error function, because it has a relatively simple partial derivative:

$$\frac{\partial}{\partial w_j} E(x_i, y_i) = 2(f(x_i) - y_i) \frac{\partial}{\partial w_j} f(x_i)$$

The gradient of each parameter with respect to this error is computed efficiently, and each parameter is shifted slightly in the direction to minimize the error. This process is repeated for every known input-output pair (and this process itself repeated again multiple times), and the parameter space is traversed stochastically, with the long-term result of moving towards a local minimum in the parameter space that minimizes the error term for all known input-output pairs, collectively.

A large body of research has been conducted on effective stochastic optimization algorithms for finding optimal weights for a given Artificial Neural Network given a training set and layer structure.

Most variants involve a stateful gradient descent, with scheduled or tuned step sizes. Such variants include Momentum, Adagrad, Adam, and Nesterov accelerated gradient descent. Optimizers such as Adam vary their step sizes based on the current rate of convergence

Many approaches involve dynamic parameterizations that adjust convergence rates and account for previous motion and velocities, and also for the total time trained up to each step. However, in practice, the naïve approach is often sufficient in the case of feed-forward neural networks.

Computation of Gradients

Here we will illustrate, in detail, the case of computing parameter gradients in the simple case of linear feed-forward layer-based networks.

Finding the partial derivative using reverse-mode differentiation involves reasoning about the derivative of some function of the final result.

For a given layer l, we wish to step in the direction of $\frac{\partial}{\partial w} E(x_i, y_i)$. We can call the layer input a_i, and layer output $l_w(a_i)$. Our goal is to compute

$$\frac{\partial}{\partial w} E$$

Knowing that E is a function of l_w, we can reframe it as:

$$\frac{\partial}{\partial w} f(l_w(x_i))$$

Where f is "what happens after the layer", before we get to our result.

Using the chain rule, we see:

$$\frac{\partial f}{\partial w} = \frac{\partial l}{\partial w} \frac{\partial f}{\partial l}$$

We can compute $\frac{\partial l}{\partial w}$ (the partial derivative of the result of the layer with respect to a given weight) based on the structure of our layer.

We compute $\frac{\partial f}{\partial l}$ based in the derivatives of the *following* layers, *after* the current layer.

The algorithm to back-propagate in a layer-based method is:

If we are at the end of the chain, the derivative of the last layer with respect to the result is straightforward based on the structure of the final layer and the error function. This is the base case.

To compute the layer *before* the last year, find $\frac{\partial l}{\partial w}$ for that layer, and multiply it by the derivative of the *following layer*. This can be found from the derivative we compute for the final layer.

Repeat, each time taking the derivative of a given layer as the "rest of the network" partial for the layer before.

In this sense, we have an iterative process, and we can build the derivative for a given weight by building backwards from the final layer. This can be performed efficiently by using proper memoization and cacheing.

Limitations of Feed-Forward Networks

However, because feed-forward ANNs are inherently structured to approximate functions, they struggle in modeling dynamical systems and systems with an inherent temporal component; attempts to do so typically entail a large explosion of parameters. The straightforward approach is to concatenate time series terms as a long input vector; for example, if training a network on a time series of n-vectors that is h terms long, the one would utilize a feed-forward ANN representing a function $\mathbb{R}^{h\times n} \to \mathbb{R}^m$, which takes the full unstructured concatenation of the past history terms. This approach is highly unsatisfying for several reasons.

Firstly, in concatenating a time series into a single input vector, this vector loses all *temporal structure* that the network may wish to take advantage of. Time series data possesses temporal structure that encodes notions of recency, causality, and a partial ordering (or well-ordering, in some situations) of data points. One may potentially exploit this structure for several gains:

1. The ability to distinguish between short-term and long-term relationships, categorically: short-term and long-term links create unique relationships, and with temporal structure, one can distinguish between the two.
2. The ability to distinguish between forward-causality and backwards-causality relationships and correlations: whether two data points come before or after each other may influence the relationship those data points might have, with inherent asymmetry.
3. The ability to create universal generalizations over translation-symmetric intervals. Closely linked to the distinction between short-term and long-term relationships, if one is aware of temporal structure, one can exploit inherent time translation symmetry to a large extent. This is similar to the principle used by convolutional neural networks, which look for structures and features in a translation-symmetric manner.

In concatenating a time series into a single input vector, all of this structure is lost. Ideally, a neural network would be able to exploit features of this structure to make decisions and predictions and to aid in its interpretation and projections. However, with a concatenated time series, neural networks instead would be left alone to infer this structure by observing several data points.

This process introduces several independent parameters that are, by nature, interdependent. With such a high parameter space, overfitting becomes likely, and the value of the model decreases. The dimensionality of the true parameter space is most likely not full-rank, and models can easily overfit or become overdetermined. Overfitting is discussed in greater detail near the end of this chapter.

3.4 Recurrent Neural Networks

To obviate these issues, the notion of RNN families was introduced (Hopfield, 1982; Elman, 1993). While traditional feed-forward ANNs can be fully described as universal approximators of functions, RNN families can be described as universal approximators of dynamical processes, and it has been shown that RNN architectures can approximate arbitrary Turing machines in the same way feed-forward ANNs approximate functions (Hammer, 1998). RNNs add dynamics to traditional ANNs; specifically, they introduce an aspect of statefulness to a neural network: outputs from a given input can be influenced not only by the current input, but also by the residual state left behind from previous computations. This makes RNNs suitable for modeling time series and other such dynamical processes that explicitly depend on history.

In precise language, in contrast to traditional feed-forward neural networks that may represent functions $\mathbb{R}^n \to \mathbb{R}^m$, predicting outputs from inputs, recurrent neural networks internally encode:

$$\mathbb{R}^n \times \mathbb{R}^s \to \mathbb{R}^m \times \mathbb{R}^s$$

Given both a vector of n inputs and "state" of s components, the network produces a vector of m outputs alongside an updated vector of s state components. To apply such a network to a time series of n-vectors, one would provide the network with subsequent points in the series sequentially, updating the state parameter along the way. An initial state vector must be provided, and this initial state parameter is often treated as something that can be optimized in the same process as the weights are optimized.

This solves issues with the feed-forward approaches in several ways. Because each time series point itself is taken as an input to the neural network, this context and structure is directly

accessible to the neural network. Values from the same attribute simply are received in the same component at every step, instead of over several different set.

The idea of temporal relevancy is also preserved in this approach, because only the most recent state vector is accessible to the neural network during the process of its computation. This, any recently observed events will have a greater effect on the output of the network than past observed events, which corresponds to the expected structure of a time series. However, without proper care, temporal proximity can potentially have too much of an effect. This can become problematic for time series with substantial lag.

Finally, parameters remain minimal. There is no redundant duplication of parameters from the duplication of input vectors, and the parameter space of the neural network is kept small enough to even be analyzed by hand.

RNNs of different forms have been used in developing powerful language models (Sutskever et al., 2011) (Graves, 2013) (Mikolov, 2012), video classification (Donahue et al., 2014), image captioning (Vinyals et al., 2014), video captioning (Venugopalan et al., 2015), visual question answering (Ren et al., 2015), image generation (Gregor et al., 2015), and meteorological circulation modeling (Toggweilier et al., 2015).

Optimization of Recurrent Neural Networks

Recurrent neural networks are typically optimized in the same manner as feed-forward neural networks: through some variation of stochastic gradient descent. To do this, such networks are "unwound" (that is, the functions composed in the appropriate way) to simulate single functions taking many input vectors and returning many (or a single final) output vector. It is this final

function that is then differentiated using applications of a multivariate chain rule. The Jacobian matrix is again found for the parameter space, and each parameter is nudged towards the direction that produces the least error.

Back-Propagation Through Time

The process of "unwinding" a recurrent neural network is a mechanical one. All cyclical dependences are treated as dependencies on separate, previous inputs. Instead of treating multiple inputs over time as updating a mutable state, one treats them as the sequential update of a chain of immutable states.

That is, instead of an in-place modification of a state vector s over $t = 0,1,2$ ..., one would instead treat s as a series of distinct vectors, s_0, s_1, s_2 .... If the activation of internal node h depends on the state at a given time, one would now treat h as a series of distinct vectors, h_0, h_1, h_2 ..., each depending on both the input at time $t = 0,1,2$ ... and the input vector x at time t.

Explicitly, suppose a specific Recurrent Neural Network has the following update rule:

- $s \leftarrow$ initialize state
- For every time step t:
  - $x \leftarrow$ input at time t
  - $i \leftarrow f(x, s)$ — update internal node activations according to input and state
  - $s \leftarrow g(x)$ — update states according to input
  - $o \leftarrow h(i)$ — output computed according to internal ndoe activations
- Collect all intermediate o's for final result, or only take final o.

Here, s, x, i, and o are all treated as mutable variables that are updated for every input vector $x(t)$.

In order to unroll a network for the purpose of backpropagation through time, the above update rule would then be translated into an inductive definition on several different immutable vectors indexed by time:

- $x_t =$ input at time t
- $i_t = f(x_t, s_{t-1})$
- $s_t = g(i_t)$
- $o_t = h(i\_t)$
- $s_0 =$ initial state

For training purposes, one would then feed $\langle x_1, x_2, x_3, x_4 \dots x_n \rangle$ with $\langle y_1, y_2, y_3, y_4 \dots y_n \rangle$, , a finite list of n input and output pairs in the process being simulated. This would then generate the distinct vectors $\langle s_1, s_2, s_3, s_4 \dots s_n \rangle$, $\langle i_1, i_2, i_3, i_4 \dots i_n \rangle$, and $\langle o_1, o_2, o_3, o_4 \dots o_n \rangle$. Instead of x, s, and i being treated as mutable vectors, they are instead treated each as a series of vectors indexed by time.

While s_1, s_2 ... do not correspond to observable features of our system, o_1, o_2 ... are meant to predict y_1, y_2 ....

Therefore, for optimization, one would create an error function on $\langle o_1, o_2, o_3, o_4 \dots o_n \rangle$. Typical error functions are:

- The sum of squared errors of the final y_n and o_n:

$$E = \mathrm{SSE}(y_n, o_n)$$

This optimizes for the property that, after seeing n inputs, the vector is able to accurately predict y_n.

- The sum of the sum of squared errors between every y_t and o_t pair:

$$E = \sum_{t}^{n} \mathrm{SSE}\,(y_t, o_t)$$

This ensures that, in the process of seeing n inputs, the vector is accurately able to predict every y_t for the entire series of inputs, with equal emphasis on every y_t.

However, it is sometimes impractical to ask a network to predict y_1 which is molded as only a direct function on x_1 and s_0, without taking into account any previous inputs. Giving this item equal importance to the other items in the series may cause the network to emphasize short-term accuracy based on recent events at the expense of long-term accuracy based on events in the distant past.

- The sum of squared errors between every y_t and o_t pair, with more emphasis given on pairs near the end of the time series:

$$E = \sum_{t}^{n} \beta^{t-n}\,\mathrm{SSE}(y_t, o_t)$$

This is often taken as a halfway point between the first and second methods. The network must optimize for long-term predictions, but also be aware of short-term predictions as well.

Typically, in practice, the first method is taken. However, the first method optimizes for a network with no care for outputs before time n. Because of this, these networks must be "primed" and warmed up with $n - 1$ throwaway input vectors before being able to predict outputs at time n, $n + 1$, $n + 2$, in practice. The initial $n - 1$ outputs will be produced without any sort of optimization or control for accuracy, and the network will be able to only reasonably be expected to predict after a $n - 1$-length warm-up.

At this point we have a suitable single output to optimize on. This output is a function of all of the intermediate states, all of the intermediate internal activations, and all of the input vectors for all times t. Because of the construction of our network, we can apply a multivariate chain rule on this function. The Jacobian matrix is again found for the parameter space, and, at every observation, we may generate a gradient from every intermediate state and internal activation and weight. Each weight is then nudged towards the direction of steepest descent in error.

Note that this method produces a separate gradient for every single time t. Typically, the approach taken is to sum together these gradients and apply them all at once to the actual single weight parameterization of the neural network.

At this point, we have reduced the optimization of RNN's to the optimization of a larger Feed-Forward Neural Network, where multiple aspects of the unrolled weights are coupled.

Because of the mechanical nature and simplicity of this method, all research applied to the optimization of feed-forward neural networks can be directly applied to the optimization of recurrent neural networks.

The stochastic gradient descent methods, as well as the momentum-based gradient descent methods used for training feed-forward neural networks, work with similar efficacy, and, additionally, automatic differentiation methods make training neural networks implicitly a simple task. Many neural network framework support automatic differentiation and automatic optimization of both feed-forward and recurrent neural networks, making these tools extremely useful in practical application.

RNN and Data Shape

Fully Connected Data Structure

We chose, as model for this study to be a recurrent neural network comprised of two fully-connected recurrent layers with forty nodes per layer. Each internal layer is fully connected, with all outputs redirected to internal inputs. The output of each node is the result of the Rectified Linear Unit (ReLU) activation function applied to a weighted sum of both inputs from the previous layer and the previous outputs of the layer; these weights are the parameters to be trained for. Figure 1 shows only a 3 input node architecture as an example for illustration purposes. However, in this work we have used 14 input nodes, where all hidden layers receive input from not only their current input, but also the previous outputs of every other node in the input layer (including itself). Every node receives input from every node in the previous layer and the most recent outputs of the same layer. The network is trained by picking the relative strengths and contributions of each connection (arrow), and also the initial output of the hidden layers.

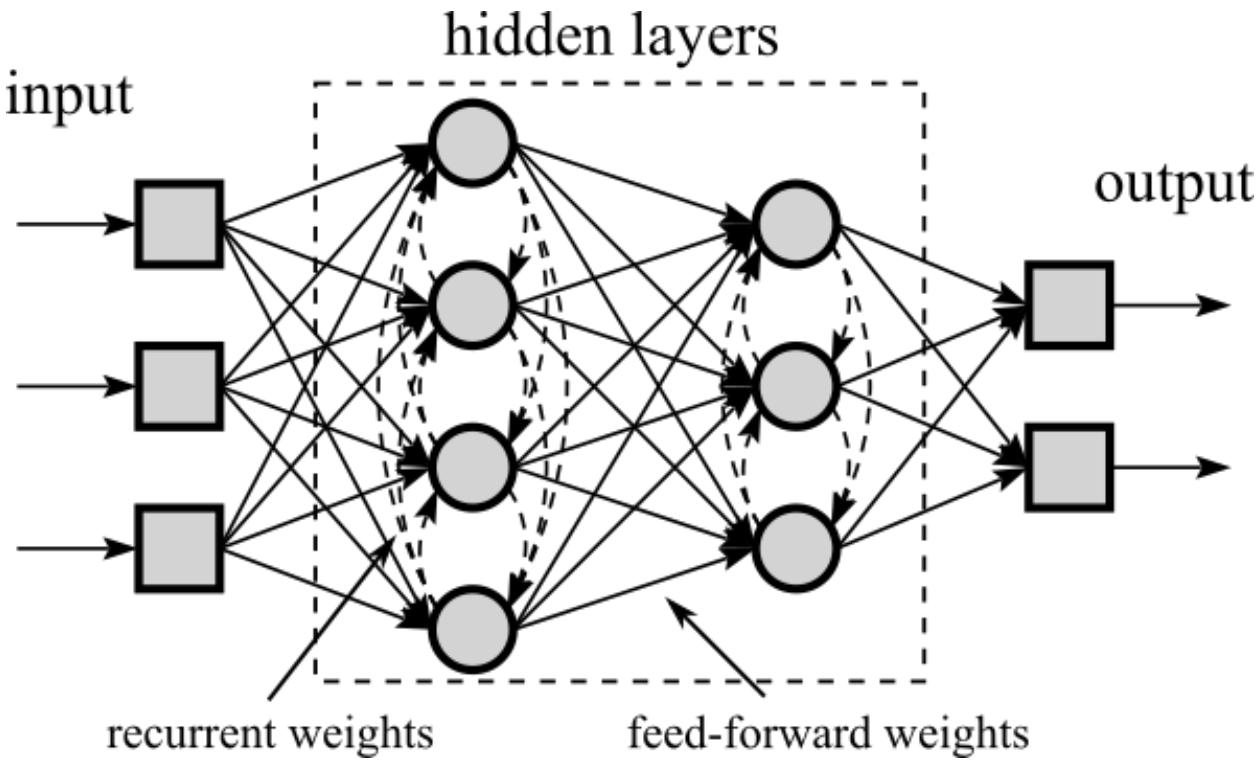

Figure 1: Internal structure of the recurrent neural network topology with multiple hidden layers

For a layer with n inputs and m outputs, the output of node $j \in 1 \ldots m$ at time $t+1$, from an input vector x, is:

$$y_j(t+1) = f\left(\sum_{i}^{n} w_{jk}\, x_j + \sum_{s}^{m} v_{sj}\, y_s(t)\right)$$

where $f(x)$, the Rectified Linear Unit activation function, is

$$f(x) = \begin{cases} 0 & x < 0 \\ x & x \geq 0 \end{cases}$$

w_{ij} is the matrix of weights of influences from the previous layer, and v_{sj} is the matrix of weights of influences from the previous activations of other nodes in that layer. By providing a non-linear activation function, we allow our network to exhibit non-linear behavior. Both weight matrices are trained parameters of the model, and $y(t=0)$, the initial state of the network, is

also a trained parameter; it is trained to create an adaptable initial condition from which prediction begins. For this network, the final output layer is a traditional feed-forward layer (that is, $\forall sj. v_{sj} = 0$) with the linear activation function $g(x) = x$.

Long-term Short-term Memory Structure

The network structure described above describes fully connected short-term memory networks. This structure is a direct generalization of the traditional linear-transformation feed-forward neural networks to a recurrent context. However, it has some significant issues. Because the entire state term is re-updated at every step, its memory is extremely short-term. Long-term effects naturally decay at an exponential rate. This can be ideal in many situations (such as for systems displaying low lag and high dependency on immediate past states); however, it is not ideal for systems high lag or non-trivial dependency on events in the distant past that aren't reflected in the immediate past.

In theory (and as shown by Hammer, 1998), a fully-connected short-term memory network can be constructed to model high-lag systems to arbitrary precision by varying the size of its internal state vectors. However, in practice, these configurations are very unstable to stochastic gradient descent, and exist in very small and isolated regions in the parameter space of a fully-connected short-term recurrent neural network. Such configurations exist in theory, yet are in practice unable to be found by stochastic gradient descent or other typical optimization techniques. At best, stochastic gradient descent will tend to converge on local minima that ignore long-term effects, and the chance configurations that consider long-term effects are often unstable in the face of typical optimization stochastic methods and are quickly lost. Essentially, because the

entire state is completely erased and re-computed at every step, any deviations from a previous state will amplified exponentially over the progression of the time series.

To overcome this, Long-Term Short-Term Memory Networks were independently developed by many sources. Instead of a complete re-write of the state vector at every step, the LSTM network introduces a second state vector that is typically passed virtually unchanged. The network, in sparse occasions, may make small and limited changes to given nodes of the long-term state vector. In this sense, the long-term state vector is long-term by *default*, and only changes under extreme circumstances which the network can be optimized to recognize. These networks couple a long-term state vector with the short-term state vector from the fully-connected recurrent neural networks described above. With the two working alongside each other, LSTM networks have proven to be more effective in problems with non-trivial long-term dependencies, overcoming the short-term issues of fully connected recurrent neural networks. LSTM networks are the chosen architecture of several of the applications cited above.

One example LSTM network separates out state for a layer into two component vectors: the layer's previous output, and a separately maintained memory vector. This is contrasted to the typical fully-connected network, which only has (transformed) previous layer output as its state.

If we define the "time-varying" vectors $x: \mathbb{R}^n$, $h: \mathbb{R}^m$, and $c: \mathbb{R}^s$, corresponding to layer input, layer output, and long-term memory vector, we can model an LSTM layer as encoding a function:

$$\mathbb{R}^n \times \mathbb{R}^s \to \mathbb{R}^m \times \mathbb{R}^s$$

The output h, again, is a simple matrix multiplication and activation function application from x and c. However, the *update* of c receives extra care. Previously, c was nothing more than a simple transformation of h:

$$c_t = f(h_t)$$

However, with LSTM networks, c_t is an altogether separate pool of numbers that is, for the most part, *preserved* over long periods of time. The network can then decide, selectively, to *forget* components given a certain input, and also to *remember* components of input and output vectors.

The key feature of LSTM layers is the *sparsity* of the forgetting and remembering actions. Instead of the state layer being completely reset on every iteration, it is preserved over time, and its components are forgotten and modified on a *sparse* basis.

Conceptually, the first modification to c_t is a "forget" action (made sparse by the sigmoidal activation function), and the second modification is a "remember" action (again made sparse by the sigmoidal activation function). Finally, c_t and x_t are combined together in the end to produce h_t.

3.5 Comparisons between Feed-Forward and Recurrent Networks

At an immediate level, deciding which kind of network is more effective and more efficient for a given domain is straightforward. As discussed, the appropriation of a feed-forward neural network to model time series data and dynamical systems is often contrived, creating ill-conditioned situations for training and yielding a large parameter-space with much redundancy in structure of the input vectors.

In the past, it has been shown that one can mitigate some of these drawbacks by attempting to reduce the redundancy in input vectors (using dimensionality-reducing techniques such as principal component analysis, or singular value decomposition) and to pre-train networks for given specific tasks. However, fundamentally, feed-forward neural networks are inherently an unnatural fit for time series data and dynamical systems.

RNNs, in contrast, are themselves an inherent model of a dynamical system, and the analysis of time series data is an immediate natural fit. While recurrent neural networks have been repurposed in the past as fixed-point machines and error-correcting models, their most natural mode of usage is in modeling dynamical systems. Their greatest success has been in these fields. However, it is interesting to note that, while feed-forward networks can be contrived to model dynamical systems, recurrent neural networks can also be contrived to model static decision problems, as well. The typical construction is to model the decision problem with an input of small sliding (or stochastically changing) window over neighborhoods in the static input vector. In a sense, this mimics the human behavior of interpreting visual data by rapidly inspecting narrow points of focus throughout an image. This method is surprisingly effective for data with large amounts of internal structure – for instance, images.

This alternative technique of repurposing RNNs to domains which Feed-forward neural networks are typically used has promise in the domain of weather and climate projections, as well, in the situation where one is interested in decisions or climate index projections based on snapshots of spatial geological or meteorological data.

Internal Node Activation Analysis with Recurrent Neural Networks

Trained RNNs provide an interesting insight to the time series that they attempt to model. Because RNNs typically contain fewer parameters than their feed-forward counterparts when applied to the same problem, it is possible to dynamically *track* and *analyze* the progress of the outputs of internal layers (known as "nodes") over the process of the modeling of the time series. The outputs of internal layers themselves create a time series that moves alongside the input time series. Through stochastic gradient descent, networks choose outputs and activations that represent important high-level characteristics of the problem domain. For example, in neural networks purposed for facial recognition, high-level features such as position and shape of facial body parts are reflected in internal layer activations. The optimization problem itself converges on important or significant high-level features the determination of its output decision.

During applications of recurrent neural networks (Karpathy, 2015), the analysis of internal activations has found interesting and significant high-level features expressed as time series of data. Radford et. al., 2017, used recurrent neural networks as a generative text model. Formally, the network was used to generate a new character after being given the previous characters seen in a text. The model treated its inputs and outputs as large sparse vectors which are zero everywhere, except for at a given index corresponding to the character the vector encodes (known in literature as a "one-hot" encoding). The Radford et. al. model, once trained on a large corpus of natural language training data, was then used to generate new strings of text by generating characters and using new characters to generate further new characters. When used as a generative text model, for instance, Radford et. al., 2017 optimized a model and discovered that a single activation node modeled sentiment, or positive and negative emotion, as a time series. By extracting the activation of the node in question, researchers can generate a time series of

positive and negative sentiment alongside analyzed text. Because these activation nodes are typically assigned high-level features without human interventions, network optimization can be used to find, isolate, and identify high-level features in time series data. While some of these features have immediate human interpretations (such as sentiment), inevitably, some of these features have no apparent interpretation.

There is promise in the prospect of using such processes to identify important high-level features in time series data that has gone undetected by scientists in this domain. It is a hope of this team to be able to analyze the trained network from this research to be able to identify new and interesting time-series climate indices that can be used to further scientific study.

Such a case-by-case analysis can be done by carefully inspecting nodes of a trained recurrent neural network. In itself, this is a testament to the power of the recurrent neural network — a similar analysis on a feed-forward neural network would be impossible, for three principle reasons:

1. Feed-forward neural networks analyzing time series would involve many times more input nodes, and therefore many times more internal nodes, making analyzing each one impractical.

   In addition, the number of input nodes and internal nodes scales with the size of the "window" used to predict on slices of the time series. If one wishes to give the network more or less of a window on which to make a prediction, one also scales quadratically (or exponentially, depending on choice of scaling method) the number of nodes and internal activations to inspect.

With recurrent neural networks, the number of internal nodes and therefore activations to inspect remains constant with the size of window used for predictions.

2. The structure of feed-forward neural networks cannot be *re-used* for different temporal windows. With a network trained for specific window sizes, it is not possible to directly apply the network to different window sizes without a major topological restructuring, which may completely change the activation structure of the network.

   For recurrent neural networks, one may use the same trained network for different window sizes. This is because activations already assume previous state, and state is processed the same way for every iteration of the network.

   Because of this, conclusions derived on the node activations for a network for one window size is immediately applicable to *all* window sizes. However, for feed-forward neural networks, conclusions derived on the node activations for a network for one window size is inapplicable to a different the activations of a network with a different window size.

   In essence, window-size is a parameter of the network structure itself for feed-forward networks, whereas it is an external parameter of the way one *runs* a network, for recurrent neural networks, and plays no role in its own internal structure.

3. Feed-forward neural networks analyzing time series lose all temporal structure; it is impossible to associate activations of specific nodes with specific points in time, or correlate activations with each other as a time series.

   With a recurrent neural network, the activation of each individual node, on its own, is itself a time series, and may be studied as such. There is no such time series activation analogy for feed-forward neural networks.

3.6 Network Training Methodology

To train the neural network, we use the well-known backpropagation through time (BPTT) algorithm, training over contiguous samples of history (Mozer, 1995; Werbos, 1999). This approach gives the network the ability to be trained to act on influences of up to 4 years into the past (though short-term influences will be much stronger).

BPTT is a gradient descent algorithm that optimizes the weight space by calculating the gradient of the error between model output and training data with respect to variations in weights and moving in the direction of negative gradient. By adjusting the step size, it is possible to tune our training process to be either quickly converging or potentially step past important local minima. For the model described in this research, we chose a step size of 2×10^{-3}.

To train on a contiguous sample of history, we ran the network over the entire length of the sample and we took the error value as the sum of the squared differences between outputs of the final network and known data for the next month into the future. From an initial network with randomly generated weights and , the network is trained over a shuffled collection of contiguous samples. Each full pass over the training set is known as an epoch. Accuracy of the model is run against the 150-point validation set of the most recent 150 months of weather data.

Overfitting

Upon training these models on data sets in this scale, one usually sees extremely quick convergence (within a matter of dozens of epochs, typically).

As the network is allowed to train further and gradients allowed to be descended further, convergence to a high correlation is typically fast, if it happens at all. Convergence speed

depends heavily on the number of parameters (hidden layers and their sizes) of the network. This signifies that the model is able to correctly identify features and signals inside the data set, and adjust the parameters in the network to account for each variation in the training set.

However, when tested with validation data (a section of our data set hidden during training), scores were less than satisfactory.

Indeed, there is an inherent issue in this analysis – overfitting. Overfitting occurs parameters over-constrain your model to the extent that, with an arbitrary choice of number of parameters, one can completely exactly model the input data.

However, exactly modeling the input data is nothing more than interpolation, since one also learns to exactly model the *noise* in the input. In the end, an over-fitted model can perfectly recreate the *noise* inherent in a training set, but potentially not the actual feature in question. It is often described as the phenomenon where models learn the noise, instead of the actual phenomenon.

It is not surprising, then, that with artificial neural networks having such a high number of parameters, overfitting is a constant problem that neural network users must actively combat.

Fundamental Problems

For mathematical models and learned models in general, an increase in model parameters produces greater risk of overfitting. In traditional modeling, an increases in parameter count are typically very visible, usually discouraged, and limits the generality and appeal of the model. In Neural Networks, this is something that is often overlooked, due to the very nature of neural

networks, and, and parameters can potentially explode exponentially without even realizing it. Even simple networks can involve tens of thousands of internal parameters, and deep learning models often have millions or more.

Intuitively, one can imagine modeling a simple one-dimensional function as a polynomial. Fitting 100 data points with a quadratic polynomial (which has three independent parameters) is only meaningful or possible (within an acceptable error) if the data has an underlying quadratic nature. The model polynomial can then be used to extrapolate or interpolate. However, fitting 100 data points with a 99-degree polynomial (exactly) is an extremely straightforward process, and the construction of an exactly fitting polynomial has been known since the time of Isaac Newton. Other techniques were pioneered by Edward Waring, Leonhard Euler, and Joseph Louis Lagrange. However, a 99-degree polynomial is unlikely to be useful for interpolation or extrapolation. Such techniques are susceptible to Runge's Phenomenon, which induces extremely high and unstable oscillations in between known data points. And, in fact, one typically does not want to match each point exactly – in doing so, one undoubtedly expends parameters matching noisy points in the data, which is almost certainly not desired. These issues are not unique to neural networks, but the large parameter count of most ANN networks makes this issue especially important.

Mitigating Overfitting

From this description, the "obvious" way to combat overfitting is increasing the size of your training set. The ratio between the number of parameters and number of training data points is a good measure of overfitting risk; this ratio is at the heart of model fitness indices like the Akaike

information criterion (Burnham, K. P.; Anderson, D. R., 2002). Therefore, adding more parameters should always be accompanied, if possible, by increasing your training set size.

Of course, this isn't always practical (or possible), especially for historical time-series data. In many practical situations, number of data points in the training set is limited by real-world constraints.

Because of this, many overfitting techniques in practice boil down to a couple of major categories:

- "Emulating" a larger training set, from a smaller training set, using clever resampling techniques.
- Preventing parameters from arriving at exact values during the training process.
- Constraining parameters in way that limits full variation.
- "Ensemble"-related techniques, which aggregate multiple potential fittings of parameters

<u>Emulation of larger training sets</u>

The fundamental cause of overfitting is the balance between the number of parameters of the model and the size of the training set. One approach to attack this fundamental problem is by increasing the size of the training set.

Of course, in practice, this isn't always an option. We are often limited by physical, monetary, political, or historical constraints. In many cases, we must settle for -mitigation techniques to simulate or emulate larger training sets. If done effectively, this tips the training set-parameter scale in the correct direction.

Done properly, this can force parameters out of overfitting due to the need to account for more data. Done *improperly*, however, this can inadvertently justify and solidify the model's biases and cause stronger overfitting as input and training data drift from real-world plausibilities to data conforming to the internal biases and structure of the model.

Prevention of parameters from exact values

Overfitting occurs when parameters reach a configuration that can exactly account for all variation within an input data set. This is possible in some situations where you have more free parameters than items in your training set. If possible,in order to improve fitness for our models will tend to exactly predict input training set, because that is what training systems are designed to do.

One way to prevent overfitting, then, is to prevent "fitting", itself. If the model is artificially restricted from fining its ideal minimum, it will need to found a way to account for this limitation. Often, this process moves the system out of its ideal maximum and overfitted state.

When done properly, this technique can push models out of a global overfitted minima and induce new local minima that mightfor data not in our training set as follows:

Constraining parameters to limit full variation

Because the fundamental cause of overfitting is the balance between the number of model parameters and the training size set, one other approach is to effectively reduce the number of model parameters by reducing their *dimensions of variation*. Certain restrictions on parameters (imposed as a regularization or loss function) can restrict the dimensions of freedom for the

model as a whole. There may be millions of parameters, but their variation is restricted to only be allowed under thousands of degrees of freedom.

In this sense, the "true" parameter space is *embedded* into a higher-dimensional space, and we only observe a higher-dimensional embedding of a lower-dimensional parameter space. This is one way to reduce the dimensionality of the parameter space (and the effective number of parameters) while keeping the dimensionality of the target space, which may be a fixed requirement of the model.

Ensemble Techniques

One of the root causes of overfitting is that any model can be *contrived* to fit the data in a given meaningless way if given enough parameters. However, it is much less likely for two separate models to fit the same data in the *same* contrived way. While two models together may individually overfit, the error in the predicted portion may average out to account for this.

In the best-case scenario, the two models overfit the data in a different way, and their non-sensical results can be averaged into useful information and predictive results.

In a bad-case scenario, two models may overfit the data in a different way, but their non-sensical results cannot be reconciled into useful information.

In the worst-case scenario, two models overfit the data in the same way, and produce the same nonsensical results. However, this is an unlikely situation for two unrelated models.

Noise Injection

Noise injection involves training while adding uniform or Gaussian noise (Zurr, et. al., 2009) to a training set that is different every epoch. In this way, one can make a 1000-point training set look like a 10000-point one – each time one samples any given point, one gets something slightly different. The theoretical justification is that most actual data points are most likely not exact or precise truths, and all are merely approximations of an underlying random variable/distribution. In adding noise to your data, one is really generating, through inference, equally valid samples of that underlying distribution. In the process, the model can witness the *distribution*, itself, and not any single point of that distribution. This prevents the network from overfitting on noise characteristics of the training set. It also practically prevents a neural network from ever exactly converging, because every epoch brings unique, never-before-seen input. In many cases, some applications may add noise **within** networks, between layers, as well. For example, instead of feeding the output of layer 3 directly into layer 4, some might inject Gaussian noise into the output of layer 3 before feeding it in, to prevent network parameters from converging too quickly and to force the network to train on a distribution instead of an exact point.

Randomization of Input Order

Typically, in one performs stochastic gradient descent by randomizing points from which a network is trained on, or randomizing the order of points. This may be performed in constant time (and constant space) using techniques such as reservoir sampling for large datasets that cannot fit in memory, or may also be done using in-memory shuffling and slicing of data sets.

Related to this technique is that of *mini-batching*, which involves computing model changes independently for several random training data points and then applying their changes *simultaneously* to update a neural network. This ensures that noisy points and outliers do not have any outsized effect on the training of the model.

Variations of batching, mini-batching, and reservoir sampling are common in literature and application, and the size of batches and mini-batches are often considered to be hyper-parameters of neural network training: parameters that defined the network structure, model and the logistics of the training process. This is in contrast to the model parameters that are meant to be train and inferred.

Choice of Activation Function

While not directly related to overfitting, the *vanishing gradient problem* has often been cited as a common cause of stagnation of neural networks on local minima. This problem comes from the fact that, at extreme regions, gradients in the error function on a per-node basis tend to shrink to zero.

However, by changing internal activation from sigmoidal (which has asymptotically zero gradients on both ends) to Rectified Linear Units (ReLu) (Glorot et al., 2011; LeCun et al., 2015), vanishing gradients become a far smaller issue, and the network has stronger impetus to move away from local minima.

This plays a role in reducing overfitting by providing stronger gradients and less stability for local minima, which forces networks to find more generalized responses, instead of extreme reactionary cases that adapt to noise more than signal.

Regularization

If models have the free choice to pick any parameters, it is possible that they may be driven to pick extreme, contrived parameters to match a given input set. This often happens when, for example, matching a high-degree polynomial to a small data set. The result often is a very exact fit, but with polynomial coefficients that are extremely large in magnitude that pay for exact fits with large swings in between points.

One common technique in model training in general that has proven to be very effective in ANNs and RNNs is *regularization*. It enforces an explicit *cost* to the magnitude of model parameters.

In loss-based training like stochastic gradient descent, networks are trained by minimizing a total loss or cost of a given configuration. Normally, this loss is defined by the difference between predicted and actual known values. However, we can impose an extra cost: magnitude of parameters. In a sense, this is imposing a cost in the *complexity* of a model.

Regularization in practice takes on many forms:

- L_2 **regularization** is imposing a cost based on the squared sum of all model parameters. Essentially, the cost is the L_2 norm of the components of the model parameters.

- L_1 regularization is imposing a cost based on the sum of the magnitude of all model parameters. It imposes a cost based on the L_1 norm of the components of the model parameters.
- L_∞ regularization imposes a cost based on the *maximum magnitude* of any model parameter.
- L_0 regularization imposes a cost based on the *sparsity* of the model: essentially, any non-zero parameter is given a fixed cost, independent of magnitude.

So, while an unregularized error function might look like:

$$E(x_i, y_i) = (f(x_i) - y_i)^2$$

A regularized one will look like:

$$E(x_i, y_i) = (f(x_i) - y_i)^2 - k\mathrm{Reg}(w)$$

The k parameter is adjusted based on how strong regularization is, or the effective *cost* of model complexity.

What this imposes is a cost for parameter magnitude: Each parameter feels a pressure to both minimize error *and* minimize its own magnitude. The final optimal result is the balance between these two pressures. Overfitted models are the result of striving for ultimate correctness. Regularized models balance correctness with norm: "too correct" is discouraged if it increases the norm. The model ultimately settles for an "adequately correct" model that minimizes the norm at the same time.

It remains to be discussed *why* a low-norm parameter is less likely to overfit than a high-norm parameter. The main motivation in the end is that parameter magnitude is used as a proxy for *model complexity*. It stands to reason, through principles like Occam's Razor, that given two models that can explain the same input data, the simpler model is preferred over the more complex one. While Occam's Razor is not an empirical basis for truth, it does philosophically motivate us to look for simpler models over more complex ones.

Regularization also serves to lower the effective degrees of freedom of parameter variation. It also serves as an extra cost that the model must overcome. If, in the end, if the model can still arrive at correct results in *the face of regularization*, it has proven itself more than a model that can arrive at correct results without any regularization pressure.

<u>Effects of Regularization</u>

To explain the effects of regularization, we can look at the effective gradient of each of the above common norms. In the case of L_2, we have:

$$E(x_i, y_i) = (f(x_i) - y_i)^2 - k \sum_j w_j^2$$

If we differentiate with respect to w_j, we get:

$$\frac{\partial}{\partial w_j} E(x_i, y_i) = 2(f(x_i) - y_i) \frac{\partial}{\partial w_i} f(x_i) - 2kw_j$$

This acts as a pressure along each training step that adjusts the parameter towards zero, with a force that acts in proportional to the current magnitude of the parameter. This force profile is found in nature as the profile of the spring force. It would not be too far from the truth to say that

the L_2 norm acts like a spring pulling the weight back to zero at each step, and that error magnitude is the pressure that keeps the weight from fully reaching zero. In the case of L_1, we have:

$$E(x_i, y_i) = (f(x_i) - y_i)^2 - k \sum_j |w_j|$$

If we differentiate this, we get:

$$\frac{\partial}{\partial w_j} E(x_i, y_i) = 2(f(x_i) - y_i) \frac{\partial}{\partial w_i} f(x_i) - k\mathrm{sgn}(w_j)$$

Effectively, this is a fixed pressure towards zero for a parameter that is independent of the current magnitude of the parameter. If L_2 acts like a spring, L_1 acts like a fixed force, such as gravity, that pulls with the same intensity no matter how far the parameter is from zero. This essentially acts to "cancel out" motion from error minimization by a fixed amount if it leads away from zero and increase motion from error minimization by a fixed amount if it leads towards zero.

The other two profiles are slightly more complex to analyze mathematically, but both essentially act as pressures to move towards zero that are selectively applied based on whether the weight is non-zero (in the case of L_0) or the current maximum (in the case of L_∞).

<u>Dropout</u>

Dropout is probably one of the more powerful modern techniques for preventing overfitting. Pioneered by Srivastava et al. (2004), dropout involves randomly ignoring nodes in a network during each training step.

For example, for a dropout rate of 50%, half of all internal neural network nodes are ignored (their outputs set to zero, and their gradients ignored) when training. Which nodes are turned off is determined randomly at every step.

Dropout is inspired by sexual selection in nature – it is based on the idea that by providing situations for self-diversification, an artificial neural network can speed up the rate of evolution by moving through different strategies and potential models faster by training different ones in parallel and re-combining the best of each attempt.

Dropout is significant because it emulates an ensemble "within" a single network: If a network has 10 hidden nodes, then, with a dropout rate of 50%, it has, essentially, $\binom{10}{5} = 252$ configurations of networks which are all almost separately training. A single network can encode 252 different networks, all within itself. A 100 node network with 50% dropout has 10^{29} configurations of on/off states of its nodes. Though each of these internal configurations are linked, the variety of configurations helps the network learn to adjust for extreme situations and ignore noise.

From a practical standpoint, it also teaches the network to build in *redundancy* to its model. It cannot rely on every node in every situation, so it must build in ways to account for randomly missing nodes in its own process. This forces the network to be more robust, and requires the network to dedicate systems of parameters to account for such uncertainties, which reduces the effective dimensionality of the parameter space.

It also, in effect, literally reduces parameters during each step, which reduces the likelihood of a single group of parameters overfitting to noise.

In the end, when making actual predictions, models are run with the full network enabled, taking advantage of all of the miniature ensembles developed during training.

One important aspect of dropout that gives it such utility is its versatility: one may now increase network size with a smaller fear of an overdetermined system. Traditionally, with more parameters, the probability of overfitting increases. However, with dropout, one may double network size while also doubling dropout rate. This can effectively negate a large part of the aspect of large parameter spaces contributing to overfitting.

4 Recurrent Neural Networks and California Drought Forecasting

4.1 Introduction

We will now discuss the application of artificial neural networks (in specific, fully-connected recurrent neural networks described previously) to study a particularly interesting case in contemporary climatology and meteorology: the 2015 – 2016 winter season in California, United States of America. In specific, we will examine the climate and weather of the season for the souther California region, climate divisions six (6) and seven (7).

The 2015 – 2016 winter season in question in California witnesses the intersection of two high-intensity regional and global phenomena — namely, the 2011 extreme drought in California and a historically powerful El Niño season.

Alone, both of these phenomenon rank among the highest in intensity in their class of manifestations. Alone, both of these phenomenon give strong predictive power on the climatology of the 2015 – 2016 winter season. However, the usual manifestation of these phenomenon make strong but contradictory predictions on the outcome of the 2015 – 2016 winter season.

Historically, strong droughts carry strong momentum in terms of precipitation and moisture that rarely break suddenly and in short amounts of time. And, historically, strong El Niño seasons bring large amounts of precipitation to the United States Southwest Coast.

At the time, many attempted to reconcile these two opposing pictures. Using artificial neural networks, we aimed to provide a supplementary voice, and perhaps shed some new insight.

Many models and forecasts at the time predicted that the El Niño trends would be able to create a meaningful impact on the ongoing drought and bring historically high levels of precipitation and moisture to California (specifically, Southern California). However, as will be discussed, our own models using recurrent neural networks show that such predictions over-estimated the strength of the El Niño season, and under-estimated the sustained momentum of the California drought and its influence in determining the overall climatology of the 2015 – 2016 winter season.

This chapter will discuss the application of Recurrent Neural Networks as a supplementary tool to current models, including questions on its accuracy and applicability to situations such as these. Furthermore, it will discuss potentials of such neural networks in driving scientific discovery and insight in this field as a way to augment current indices and understanding of Southern California climatology in a way that can be generalized to other regions, or even other climate indices.

4.2 Methods and Materials

Artificial Neural Network

We utilize a Fully-Connected Recurrent Neural Network as described in previously in chapter 7. Through trial, for our research, we found that a dropout rate of 10% produced results with the best validation scores.

Our internal node counts were 40 and 30, respectively, for two internal hidden layers, although we found similar results with other variations of internal layer sizes and counts.

Our usage of a Fully-Connected Recurrent Neural Network was arrived at after iterating through several different machine learning and artificial neural network structures – including fully-connected feed-forward networks, convolutional feed-forward networks, and long-term short-term memory recurrent neural networks. The fully-connected recurrent neural network was sufficient for prediction, without needing the extra power of long-term short-term recurrent networks.

A theoretical justification for the choice of a Fully-Connected Recurrent Neural Network is given in the previous chapter. For time series data, recurrent networks are the natural choice. It allows for temporal structure to be preserved in the input data, which may be exploited by the neural network. Because of the high sensitivity of our domain to temporal structure and causality, this this preservation of structure is critical for effective training. And, as discussed later, preserving temporal structure opens up a large world of possibilities for analysis of internal node activations. This in itself is one of the end-goals of our original research, and so the usage of a recurrent neural network over a feed-forward one was dictated both by our high-level goals and by the fundamental nature of our system.

Long-Term Projections

PZI data for each month is represented as a 14-element vector. The first twelve elements are indicator elements, representing the month of the data point where each component is either 0 or 1. The final two elements are the normalized, scaled PZIs from that month for both climate divisions. For training, data vectors are grouped together in contiguous samples of 48 months. Each 48-month group is paired with a single two element vector representing the two divisions' PZI data for the month right after the month of the final data vector in the group.

Thus, the network used has 14 input nodes and 2 output nodes. In order to project several months into the future, the projection for the next month is joined together with a 12-element prefix indicating the next month and used to predict the month after. This allows the network to step forward several months into the future, despite its ability in providing projections for only the immediate following month. This technique uses the RNN as a directed continuous state non-Markov feedback generator resembling a continuous space sequence memorizer (Wood et al. 2009), in a similar manner as strategy explored extensively by Graves, et al. (2014) to generate curves and paths from training data. Hopfield (1982) explored the convergence of steady-states of this process.

This type of structure is very pervasive in literature, both in academic and industrial usage of neural networks. It is commonly used in textual analysis and generation, in order to generate continuations in bodies of text or to imitate the style of a particular author. It is also used in physics simulations to model physics in specific situations where analytical analysis is impractical. It is very similar in nature to both hidden- and visible-variable Markov chain models, and can, itself, be a generalization of Markov models to infinite and continuous state spaces. Whereas, traditionally, the state of Markov models is constrained to be a finite set (like the state sets of finite automata such as Mealy and Moore machines), the state space of a recurrent neural network is described in the following equation. It is an uncountably infinite state that is the product of all of the state sizes of each hidden layer, for n hidden layers, where s_i represents the size of the state vector for layer i.

$$\mathcal{S} = \mathbb{R}^{\sum_i^n s_i} = \prod_i^n \mathbb{R}^{s_i}$$

In treating this as a generalization of hidden- and visible-variable Markov chain models, many of the stability and accuracy analysis conducted for such models is also applicable to the domain here. We explore this later in and take advantage of the link to make stronger conclusions about the strength and mathematical characteristics of our model.

Naturally, such a system has a strong tendency to evolve fixed-point steady-state behavior, when given no driving input and instead asked to predict only on its previous outputs. This susceptibility is especially strong for fully-connected recurrent neural networks (and is somewhat less of an issue for alternative topologies such as Long-Term Short-Term Memory Networks).

The twelve input nodes dedicated to month number here somewhat mitigate this, by providing a constant driving input. This serves to extend the period of recurrence for steady-state behavior to a minimum of twelve months.

Due to the nature of our situation, because we only wish to predict up to three or four months in the future, steady-state and fixed-point behavior do not impact the predictive utility of our model in practice, and are only notable as theoretical concerns for long-term predictions and modeling.

Note that this issue is no different than fixed-point and steady-state issues for hidden- and visible-variable Markov models (and other related finite automata analogies); however, being a continuous-space generalization, the problem is significantly less pronounced.

However, in the future, analysis of this fixed-point steady-state oscillation may prove to yield some insight into the fundamental nature of the annual oscillations of the PZI anomaly and the domain in question. There is some promise shown in hidden markov model analysis and Hopfield network analysis to show that fixed-points of such models contain potential insight, and

can be studied for a better understanding not only of the model, but of the physical situation being examined.

4.3 Results

Training

We train using the well-established backpropagation-through-time algorithm, discussed in the previous chapter, using straightforward batching and stochastic gradient descent. We use the final-result error function, out of the three options discussed in the previous chapter.

As expected, convergence with these models happen very quickly. Due to the relatively high number of parameters of the network in comparison with the amount of data types available, local minima are found within a matter of dozens of epochs, typically, depending on the size of internal layers. This convergence rate aligns with what is expected in similar literature.

After several dozen iterations of stochastic gradient descent, one sees correlation between observed data and network outputs ranging between 0.7 and 0.8. These high correlations are expected for training sets; it signifies that the model is able to correctly identify the important features in our PZI training set, and the network parameters are properly adjusted to account for all or most of the features in the training data set in order to provide such high correlations against the expected results.

Validation

As can be expected, without action taken to prevent overfitting, these models tend to validate poorly. Initial validations were unsatisfactory, and trained networks fared little better than random guessing in predicting on PZI, despite their fast convergence.

However, with the overfitting reduction techniques described in the previous chapter, the networks converge slightly slower, but consistently to similar trained states. We also reduced the difference between the training and validation data correlations and achieved a consistent results regardless of initial starting state.

Despite underestimation in rare extreme cases, the networks output agrees strongly with observed data. Longer historical datasets are necessary for a better training, as the 1895 – 2005 historical record includes only a few El Niño events to analyze. Although we acknowledge the limitations of our effort, we consider the extracted results of the performed methodology quite satisfactory in forecasting such extreme values for the next year, based only on historical PZI time series.

Figure 1 shows the one, two, and three-month forward projections for the 2006 – 2015 validation set alongside observed values and the corresponding correlation plots. The black dashed lines correspond to perfect fit ($y = x$), while the red solid lines to the least-square fit. The total variation in y during the examined period for the 1 month ahead step is explained by the linear relationship between x and y represented by $y = 0.697x - 0.510$, x represents the observed PZI values and y represents the forecasted values.

Examining the decline in predictive power as the network projects further into the future, we find that the model has statistically significant predictive power with correlation coefficients ranging from 0.610 to 0.434 for between one to three months ahead, validating it for confident predictions. It is noteworthy that for longer-term predictions, the developed RNN tends to overestimate, though in the correct direction of variance. Moreover, it is notable that the network

validates better as it progresses down the timeline, using the first yew years to develop its internal state to gauge the current context.

For our trained networks we found that, after a period of time on the order of one year, the state from previous months are forgotten and the network settles into a periodic steady-state feedback cycle. However, the projected data can still be analyzed for results until the time when the residual influences of the past fade away. Furthermore, analysis of activation profiles of internal nodes for RNNs can be shown to yield physically insightful results which will be discussed in a future work.

Medium-Term Projections

Running the forward prediction method from the PZI records leading to January 2016, we found that PZI of the following month represents a drier than average record, despite the anticipated the wet late fall and early winter season (Hoell et al., 2016; Zhenya et al., 2015; Climate.gov, 2015), hence we expect a return to drought conditions. Figure 2a shows a solid comparison between the powerful 1997–1998 El Niño season used here as a baseline when comparing with the 2015-2016 El Niño season. Figure 2b shows the actual observed precipitation levels for the current season compared to precipitation for the 1997-1998 El Niño season which confirms the validity of our model's projections of a drier season associated with the 2015-2016 El Niño as compared to the 1997-1998 one.

The thick lines represent the model's direct output while the light lines show a measure of the model's uncertainty, calculated via a Monte Carlo process simulating stochastic noise in subsequent prediction steps to account for potential errors in the model (Figure 2c). The grey dashed line is the baseline 1997-1998 El Niño season, superimposed over their respective months

in the 2015-2016 season presented by the dark dashed lines. The PZI anomaly peaks at 0.242 standard deviations below the monthly average in May of 2016, and quickly sinks back to 0.924 standard deviations below monthly average by August of the same year. In total, 2016 will be a drier year than average with a -0.715 PZI anomaly for the California South Coast Drainage climate division.

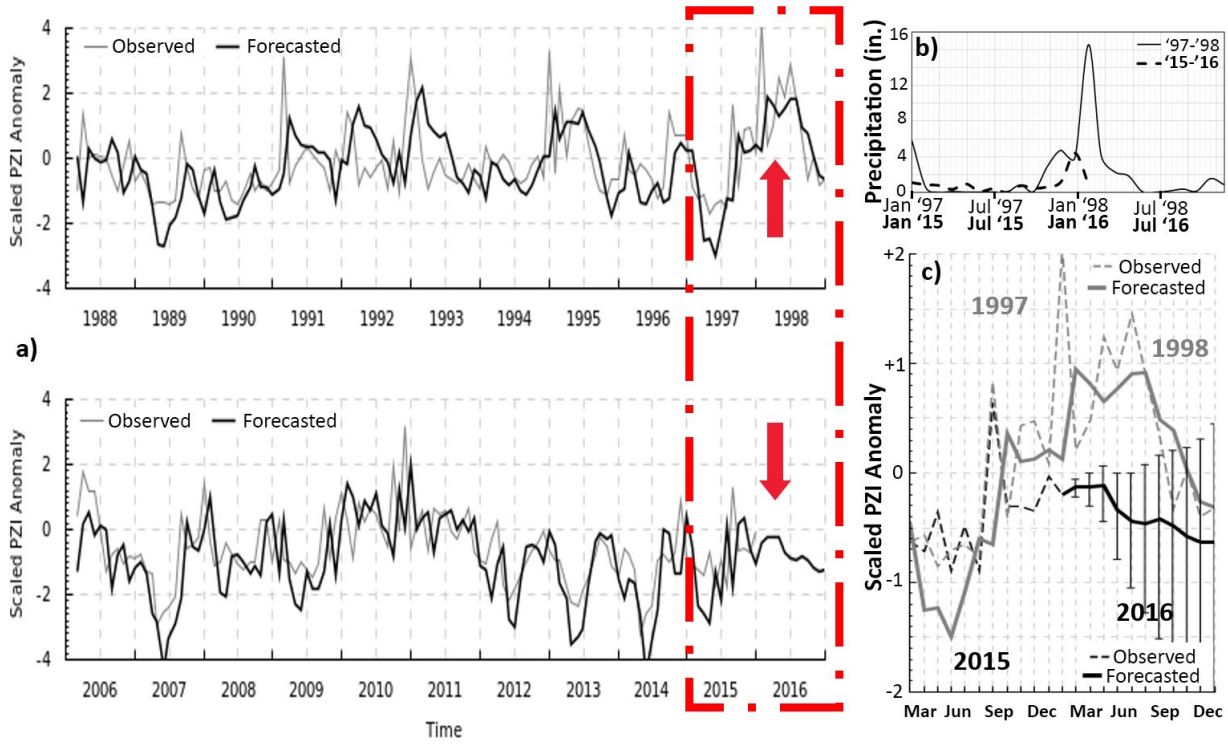

Figure 2: (a) Comparing observed and forecasted PZI data for the two El Niño season in question (b) Observed PZI anomalies for California Climate Division 6 for the two El Niño seasons in question, confirming the low-precipitation season that the model predicts (c) Detailed look at model output projections for PZI for the year 2016 compared to observed values and predictions for 1998, with model uncertainties

The projected PZI for 2015-2016, indicates a much weaker season as compared to the 1997-1998. The data points themselves are contrasted with that several baseline El Niño seasons in

Table 1. This season can be contrasted with the baseline 1997 – 1998 El Niño season, which saw a February 1998 that was 4.13 standard deviations above the average PZI for the month, and a 1998 that was 1.1 standard deviations higher than that of the average year. It can also be contrasted with the 1982 – 1983 El Niño season, which saw a peak anomaly of 2.22 standard deviations above the monthly average in April of 1983, and saw a 1983 that was 1.15 standard deviations above the annual average.

With confidence, it can be concluded that the 2015 – 2016 season proves to be underwhelming in precipitation, and that drought conditions will persist past this winter season. The worst of the drought has apparently passed since 2013 with an annual anomaly of -1.30 and 2014 with -1.17 as compared to 2015 with -0.85, with a projected -0.715 annual anomaly for 2016, continuing a general trend of slow but steady emergence from the current drought season. While immediate predictions towards values one month into the future have strong predictive power, it cannot be assumed that longer term forward projections maintain the same predictive power.

Table 1. Historical El Niño Palmer Z Index Levels (Anomalies in Standard Deviations)

Season	Peak anomaly	Peak anomaly month	Annual anomaly
1957 – 1958	3.03	April	0.5
1982 – 1983	2.22	April	1.15
1997 – 1998	4.13	**February**	1.1
2009 – 2010	1.1	January	0.45
2015 – 2016	−0.242	**May**	−0.715

To clearly show the skill of our proposed method, the correlation between the observed and forecasted PZI anomaly, for past known weak to severe El Niño responses according to NOAA, is computed and presented in Table 2. P-values are also provided corresponding to the likelihood

of the null hypothesis that the observed and the forecasted PZI are uncorrelated for the entire period under investigation. In other words very low P-values show that the correlation between forecasted and observed PZI is statistically significant. The model was intentionally trained on the entire history (all years), rather than, on known El Niño years for the purposes of drought projection.

The main motivation of doing so is to be able to provide the model with more information on the behavior of the system in all situations. Having the model trained only on El Niño years, it would not be able to observe and learn from recurring phenomenon that do not normally occur on El Niño years. Training only on El Niño years would arbitrarily deny the model the chance to learn from these phenomenon, which might become significant in the specific season we are attempting to study. Moreover, by training on non- El Niño seasons, the model has the opportunity to distinguish between El Niño and Non-El Niño seasons and learn the degree of adjustment required in the context of the years leading into each season.

Table 2: Observed (obs.) versus projected (proj.) PZI correlation coefficients with corresponding P-value for past El Niño events categorized as weak, moderate, strong and very strong showing the RNN model skill

Weak	CC (obs. vs. proj.)/Pvalue	Moderate	CC (obs. vs. proj.)/Pvalue	Strong	CC (obs. vs. proj.)/Pvalue	Very Strong	CC (obs. vs. proj.)/Pvalue
1953-54	0.608/0.0179	1951-52	0.822/0.0005	1957-58	0.837/0.0003	1982-83	0.623/0.0156
1958-59	0.837/0.0003	1963-64	0.892/0.0000	1965-66	0.869/0.0001	1997-98	0.870/0.0001
1968-69	0.556/0.0302	1986-87	0.873/0.0001	1972-73	0.694/0.0061		
1969-70	0.575/0.0253	1991-92	0.332/0.1459				
1976-77	0.483/0.0559	2002-03	0.892/0.0000				

1979-80	0.688/0.0067
1994-95	0.584/0.0232
2004-05	0.937/0.0000
2006-07	0.880/0.0000

From the above table the model showed some skill towards the majority of El Niño years regardless of their strength. The correlation coefficients varied around 0.7 which is quite similar and slightly higher than our validation data that was presented in Figure 2. We did not include 2009-2010, 2015-2016 in this analysis as they are part of the validation dataset used in our forecast model.

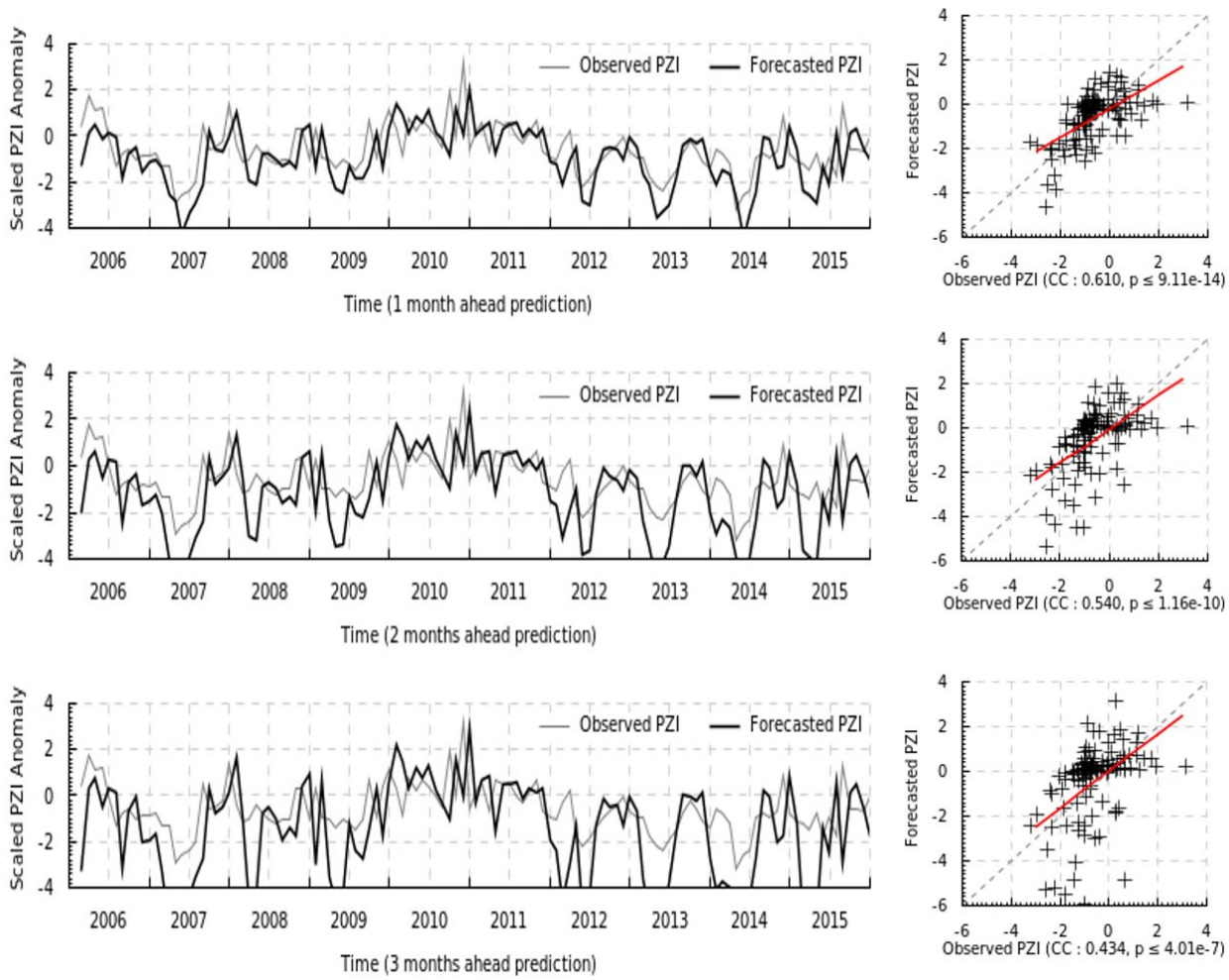

Figure 3: Time series correlation plot between observed and forecasted PZI using RNN for climate divisions 6 & 7 wit lead times of 1 month (top), 2 months (middle), 3 months (bottom), starting January 2006

4.4 Internal Activation Analysis

Internal node activation analysis for our trained network will be an important topic of future study. Shown Figure 4 is a sample of select internal activation time series of our model, shown alongside PZI.

Future analysis can be applied by comparing PZI with precipitation data and other climate indices. We may find that some internal node activations may closely align with other known climate indices, such as regional NDVI, SOI and MEI El Niño indices, surface temperature,

precipitation, cloud coverage, soil moisture, and other climate time series. A high and significant of correlation of an internal node activation with any of these indices would be a significant find, and would show that the network arrives at important high-level features used by climatologists to perform its calculations.

As a second stage of analysis, one may identify node activations which have no apparent physical interpretation, and attempt to apply these nodes to other climate situations to gauge if they have any predictive power or useful physical insight.

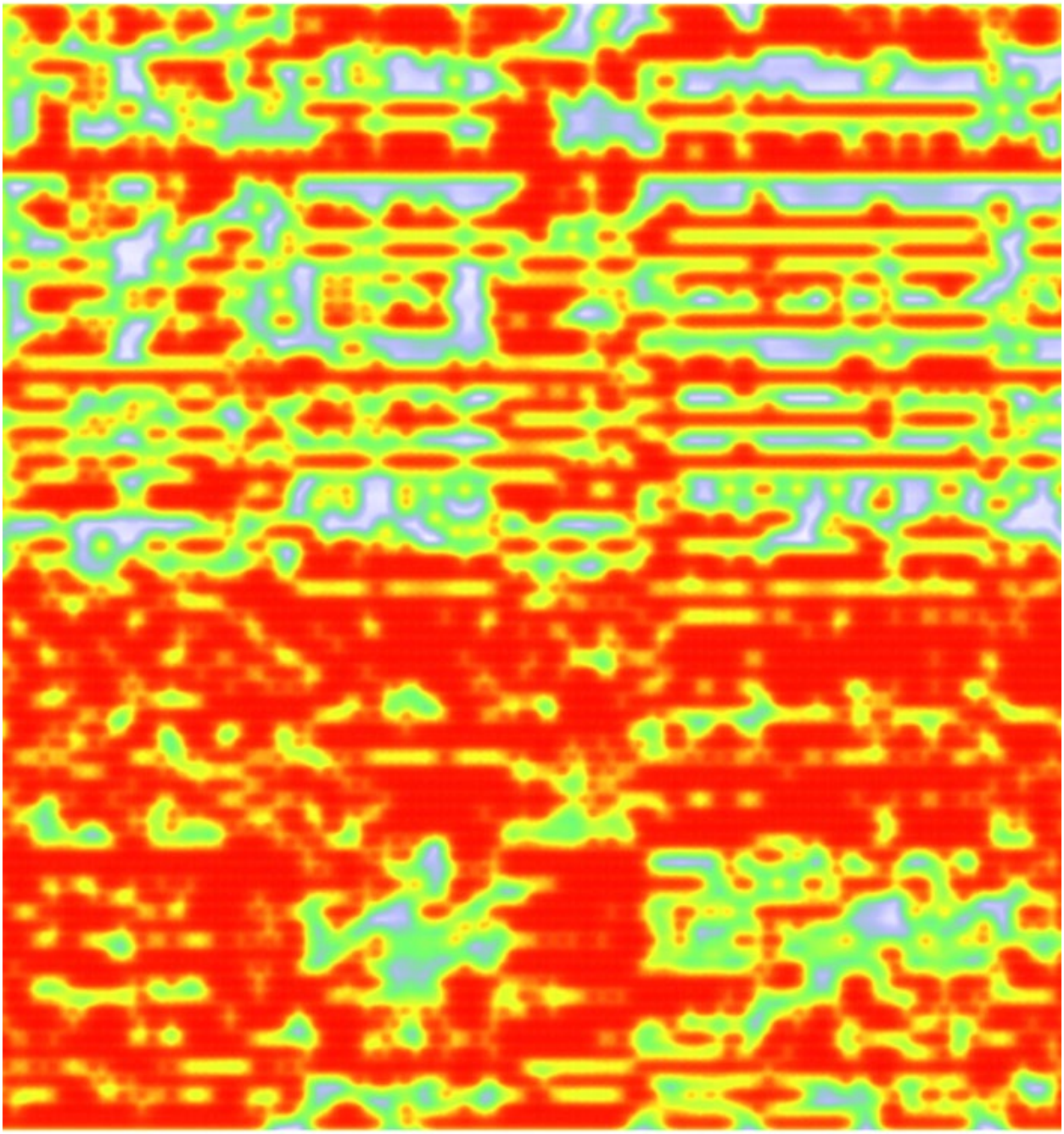

Figure 4: Activation analysis of a trained neural network; horizontal axis represents time, and vertical axis represents each discrete node's activation history

As discussed in the previous chapter, this line of research has shown much promise in other fields, leading to discoveries and innovations in sentiment analysis, textual, and audio generation. The transfer of these innovations to domains such as weather and climatology can

drive the discovery of new important weather indices, validate current ones, and give a greater understanding to how different weather indices and environmental variables interact and are related to one another.

4.5 Contrasts to Feed-Forward Results

Such a case-by-case analysis can be done by carefully inspecting nodes of a trained recurrent neural network. In itself, this is a testament to the power of the recurrent neural network — a similar analysis on a feed-forward neural network would be impossible, for three principle reasons:

1. Feed-forward neural networks analyzing time series would involve many times more input nodes, and therefore many times more internal nodes, making analyzing each one impractical.

   In addition, the number of input nodes and internal nodes scales with the size of the "window" used to predict on slices of the time series. If one wishes to give the network more or less of a window on which to make a prediction, one also scales quadratically (or exponentially, depending on choice of scaling method) the number of nodes and internal activations to inspect.

   With recurrent neural networks, the number of internal nodes and therefore activations to inspect remains constant with the size of window used for predictions.
2. The structure of feed-forward neural networks cannot be *re-used* for different temporal windows. With a network trained for specific window sizes, it is not possible to directly apply the network to different window sizes without a major topological restructuring, which may completely change the activation structure of the network.

For recurrent neural networks, one may use the same trained network for different window sizes. This is because activations already assume previous state, and state is processed the same way for every iteration of the network.

Because of this, conclusions derived on the node activations for a network for one window size is immediately applicable to *all* window sizes. However, for feed-forward neural networks, conclusions derived on the node activations for a network for one window size is inapplicable to a different the activations of a network with a different window size.

In essence, window-size is a parameter of the network structure itself for feed-forward networks, whereas it is an external parameter of the way one *runs* a network, for recurrent neural networks, and plays no role in its own internal structure.

3. Feed-forward neural networks analyzing time series lose all temporal structure; it is impossible to associate activations of specific nodes with specific points in time, or correlate activations with each other as a time series.

   With a recurrent neural network, the activation of each individual node, on its own, is itself a time series, and may be studied as such. There is no such time series activation analogy for feed-forward neural networks.

4.6 Conclusions

This research addressed the rationale of using PZI as a significant precipitation indicator to address the anticipated heavy rain over Southern California driven by the strong 2015-2016 El Niño season.

By investigating many ANN models, utilizing proven effective RNN configurations and applying them to analyze over a century of monthly PZI data, it is shown with strong confidence

that precipitation associated with the 2015-2016 El Niño season is currently and will continue to be weaker than that of the historic 1997-1998 El Niño season.

From this, we anticipated that drought conditions will continue to persist (albeit at an alleviated level) beyond this winter. These forecasts are made with a model that is well tested with significant high correlations on a ten-year validation set, with p-values $p<10^{(-6)}$ for predictions up to three months into the future. Such projections are confirmed through current observed precipitation levels and PZI values for 2016 as compared to those of the 1997-1998 season.

These results are consistent with the observed data that we now possess, and as our models continually are run on this data set, we observe now the unfolding of our model's predictions, which were more accurate than other prevailing models and published predictions at the time.

Our team is currently in the process of applying these models to new climate divisions in hopes of achieving similar validation scores and demonstrating the general applicability of our model. Already, promising results are shown when applying these models to other climate regions in California and the United States, and also to different climate indices. The general applicability of the model consistently verified when attempting to apply it to different climate situations and data sets.

In addition, a visual look at internal node activation shows promise in the usage of these techniques for confirming and identifying climate indices and high-level features that can find applications in situations outside of drought projections, and in the greater field of climatology in general. Simple correlation analyses and manual sifting through nodes (which is made possible

and easier using recurrent neural networks) is a road that shows promise in yielding valuable insight, as shown by successes in using the same process in other fields and domains.

5 Recurrent Neural Networks for Understanding Climate

5.1 Introduction

We will now expand on the promise in the past chapter – of using internal node activations to help us discover insight about the actual physical system being modeled. We again project on regional PZI, which tracks moisture conditions and changes in moisture conditions for a given month. PZI was chosen over direct precipitation measurements due to its nature as a more meaningful proxy of drought conditions, which can be distinctly useful in economic planning decisions over raw precipitation measurements and projections. The utility of PZI as a model predictand was demonstrated and expanded on in the previous work. PZI is used here as a broader hydrologic measure that integrates soil moisture, run-off, and other ecologically significant factors to create a physically meaningful metric for the prediction of future drought conditions. PZI itself has a high correlation with precipitation, which shows its usefulness as a proxy for precipitation predictions.

For the purpose of this study, the Palmer Z-Index can be considered a memoryless index indicating current drought level, based on empirically measured and calculated numbers. It is computed based on factors such as soil moisture capacity, total precipitation and potential evapotranspiration, moisture recharge, run-off, and moisture loss. Precipitation and PZI are correlated with a coefficient of about 0.77, making PZI a useful proxy for precipitation, while also including more meaningful measurements of drought intensity. An important property for its usage as a predictand is also its lack of explicit reference to any previous values; unlike

Palmer Z-Index, PZI is re-computed from direct measurements every month, and so does not have any explicit autoregressive factors.

The historical Z-indices for all California Climate Divisions were gathered from the Global Historical Climatology Network (GHCN) *nClimDiv* data set (ftp://ftp.ncdc.noaa.gov/pub/data/cirs/climdiv/), which gives per-climate division aggregates for raw and processed surface data measurements. Values in the *nClimDiv* data set are calculated using area-weighted averages of points on a 5 km-resolution grid overlaid across each division. This resolution is high enough to ensure sufficient spatial sampling, especially for the climate divisions being analyzed in this paper (*Vose et al.* 2014). Points are assigned climate data based on spatial interpolation of nearby stations, with topographic and network variability taken into account in the interpolation process. GHCN subjects the data to regular quality assurance reviews to ensure correctness. For California South Coast Drainage and South East Desert Basin climate divisions, data from a total of 526 and 184 stations respectively, are taken into account and aggregated. Aggregated station data are available from January 1895 to January 2016, on a month to month basis, giving 1452 total data points. The most recent 120 months of data are set aside for validation, and the remaining 1332 points are used for training purposes.

From the *nClimDiv* data set, the gathered historical Z-indices were shifted to historical monthly averages and re-scaled to have standard deviation equal to unity. This is done to remove yearly periodic components in mean and variance that might arise in the process of the Z-indices computation.

That is, an aggregate μ_M average or each month and σ_M standard deviation for each month was computed, and the normalized PZI for year Y and month M is computed as:

$$\hat{Z}_{Y,M} = \frac{Z_{Y,M} - \mu_M}{\sigma_M}$$

This process also removes a large deal of autocorrelation from the time series, making the prediction much more meaningful.

5.2 Network Structure

For this work, we elaborate on our previous paper by moving from fully-connected recurrent layers to Long-Term Short-Term Memory (LSTM) recurrent layers. LSTM layers maintain an isolated and curated "memory" as its state, which is highly persistent and left mostly unchanged from time step to time step. The network is made to treat changes to memory with strong discretion. This allows networks to maintain long-term memory in ways that fully-connected recurrent layers could not.

The final structure of our network is two LSTM layers with 16 and 12 nodes each, followed by a fully-connected feed-forward output layer with no activation function. We apply 50% dropout after each LSTM layer and apply L^2 regularization with regularization coefficient 0.1.

The network is trained ("unrolled") with a lookback of 24 months. That is, its mode of operation is, once provided a 24-month history, to output the PZI of the next month. The 24-month history is used to prime the state in order to make the final prediction. The initial state is used as a trained parameter, as well.

Training involves picking the parametrization of each LSTM layer, the parametrization of the final output layer, and the values in the initial state of the network, that would, if shown 24 months, output properly the next month's PZI.

The LSTM layer with an input $\mathbb{R}^n$ and output $\mathbb{R}^m$ consists of two state components: its memory cells $\mathbb{R}^m$ and the previous output, $\mathbb{R}^m$. It is parametrized by four $(n+m) \times m$ real-valued matrices (and four $\mathbb{R}^m$ bias vectors), which control the behavior of four "gates". Roughly speaking, each gate controls how input and previous output affect the components of the cell state.

1. Forget Gate: Provides the logic of the network to the ability erase components in the cell state, depending on the previous output and the current input. Its output is essentially binary.
2. Input Gate: Provides the logic of the network the ability to write components in the cell state, depending on the previous output and the current input.
3. Update Gate: Provides the new values to write into the components of each cell, based on the previous output and the current input. While the input gate determines whether to write, the update gate computes *what* is written.
4. Output Gate: Provides the output of the RNN layer, which is provided as the input to the next layer.

Assuming previous cell state C_{t-1}, previous output h_{t-1}, and input x_t, we can compute the output h_t and next cell state C_t via the following equations:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

Where:

- W_f and b_f are the weights and biases parametrizing the forget gate.
- W_i and b_i are the weights and biases parametrizing the input gate.
- W_C and b_C are the weights and biases parametrizing the update gate.
- W_o and b_o are the weights and biases parametrizing the output gate.
- $[x, y] : \mathbb{R}^{n+m}$ represents an element in the direct product of two vectors $x : \mathbb{R}^n$ and $y : \mathbb{R}^m$
- $\cdot$ represents matrix-vector multiplication.
- $*$ represents the component-wise product of two vectors
- $\sigma(x)$ represents the sigmoid function, $\sigma(x) = \frac{1}{1+e^{-x}}$
- $\tanh(x)$ represents the hyperbolic tangent, $\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$

The four weight matrices and four bias vectors represent the parameters that must be trained for the LSTM network to make accurate predictions. The initial cell state C_0 and initial previous output h_0 are also treated as trained parameters.

The final layer is a fully-connected feed-forward layer, which is essentially a linear combination (plus a bias) of the input. It is represented by:

$$o_t = W \cdot h_t + b$$

Where W and b are a weight matrix and bias vector that are also to be trained for.

5.3 Methodology

PZI data for each month are represented as 13-element vectors. The first twelve elements are indicator elements, encoding in a "one-hot" form the month of the data point. If the index of the vector matches the current month of the data point, the component is 1; otherwise, 0. The final element is the normalized and scaled PZI for that month. PZI is normalized to have mean 0 and standard deviation 1 when considering historical PZI for that specific month.

For training, data vectors are grouped together in contiguous samples of 24 months. Each 24-month group is paired with a single one-element vector representing the PZI data for the month right after the month of the final data vector in the group. Thus, the network used has 13 input nodes and 1 output node.

For predicting many months into the future, the projection for each next month is joined together with the appropriate 12-element one-hot vector. Then, the network is run again, treating the output as a part of the known input. This allows the network to step forward several months in the future. This technique utilizes the RNN as a directed continuous-space sequence memorizer (Wood et al. 2009), a technique explored extensively by Graves, et al. (2014) for the generation of curves and paths from training data. The steady-state convergence of this method was explored in-depth by Hopfield (1982).

To train the neural network to find the correct parameters on a per-division basis, we use the backpropagation through time (BPPT) algorithm, taken over contiguous samples of history (*Mozer*, 1995; *Werbos*, 1999). The process involves turning a $f_p: S \times A \rightarrow S \times B$ model into a $g_p: S \times A^N \rightarrow S \times B$ model by repeatedly composing the function with itself, threading in updating states and successive input values.

The final unrolled result $S \times B$ is defined as, given initial s_0 and a_t inputs:

$$(s_t, b_t) = f_p(s_{t-1}, a_t)$$

taking (s_N, b_N) as the final unrolled result.

Furthermore, we train on the input state by turning s_0 into a trained parameter and dropping the final state, giving us a final unrolled function $h_{(p,s_0)}: A^N \to B$. For our purposes, we set $N = 12$, $A \sim \mathbb{R}^{13}$, and $B \sim \mathbb{R}^1$.

This final $A^N \to B$ function is fully differentiable, so we can use it via normal gradient descent through backpropagation. For our purposes, we use an Adam optimizer for more efficient convergence, which dynamically adjusts step size based on momentum and train time.

The network is initialized with randomly generated weights and initial state, based on a gaussian distribution centered around zero with variance 0.2. The network is trained against a 90/10 split of the previous 1492 months of weather data, where inputs are 24-month vectors, and outputs are 1-month predictand.

In the end, we train seven distinct models – one model for each climate division. However, each model has identical structure, as the goal is to find a universally adaptable structure.

It is possible also to initially train a model on a single division, and then proceed to train on a different division. This pre-training method is used in many mainstream ANN applications.

The LSTM network converges extremely quickly, within the span of several passes over the training data (commonly known as "epochs"), to extremely low RMSE values and high correlation (between 0.9 and 1.0). However, this model is highly overfitted, and validates poorly, providing RMSE as high as 1.5 and correlations as low as 0.1.

In over-fitting the data, the model fits its parameters to specific noise of the training data, essentially acting as a k-nearest-neighbors approximator. In order to mitigate overfitting, we apply three separate techniques:

First, we inject gaussian noise into the training data (Zur et al., 2009), which serves to expand the effective size of the training data set.

Second, we apply the dropout technique (Srivastava et al., 2014). Dropout is a regularization tool designed to mitigate overfitting; the network is essentially taken as the average of several semi-dependent networks. The model is forced to only be able to operate with a small subset of its nodes at any given iteration, forcing itself to build in redundancies and multiple ways to compute an answer.

Thirdly, we apply L^2 regularization, constraining the model parameters to be as "simple" as possible by encouraging smaller, more compact parameter choices. This prevents the model from overfitting by picking extreme parameters that match more on noise than on actual signal and is a common component of neural network training.

All these methods sacrifice convergence on the training data for validation on the test data, increasing test data correlation.

5.4 Results

We see that these networks converge consistently within around two hundred epochs to consistently similar trained states and validation results. Network output agrees strongly with observed data, based on a baseline linear autoregressive model. Also included is the baseline autocorrelation between successive items in the list.

Table 3: Statistics for trained models per division on RNN

Division	Val. RMSE	Val. Corr.	Corr. p-val	Baseline RMSE	Auto Corr.	Accuracy Gain	Corr. Gain
Cal-1	0.9719	0.3070	$< 10^{-3}$	1.274	0.1875	31.1%	63.7%
Cal-2	0.9555	0.3190	$< 10^{-4}$	1.244	0.2258	30.2%	41.3%
Cal-3	0.9088	0.3002	$< 10^{-3}$	1.177	0.3066	29.5%	-2.09%
Cal-4	0.9690	0.4434	$< 10^{-7}$	1.133	0.3583	16.9%	23.8%
Cal-5	0.8989	0.5415	$< 10^{-11}$	1.142	0.3479	27.0%	55.6%
Cal-6	0.8062	0.6003	$< 10^{-14}$	1.120	0.3722	38.9%	61.2%
Cal-7	0.8772	0.3320	$< 10^{-4}$	1.151	0.3368	31.2%	-1.43%
Ave	**0.9125**	**0.4060**	**-**	**1.177**	**0.3050**	**29.3%**	**34.6%**

The alternative method – pre-training first on one division, and then training the resulting model on different divisions – provides comparable results, but with faster training times after the initial pre-training period is finished.

In the following plots, we include the observed and forecasted PZI for all regions. For each region, we include a sample of the network running on training input to show convergence. The same time period is chosen for all regions.

Also included are sparkline representation of internal network activations for the training data set, which is explained in the upcoming section.

Result Convergence and Predictions

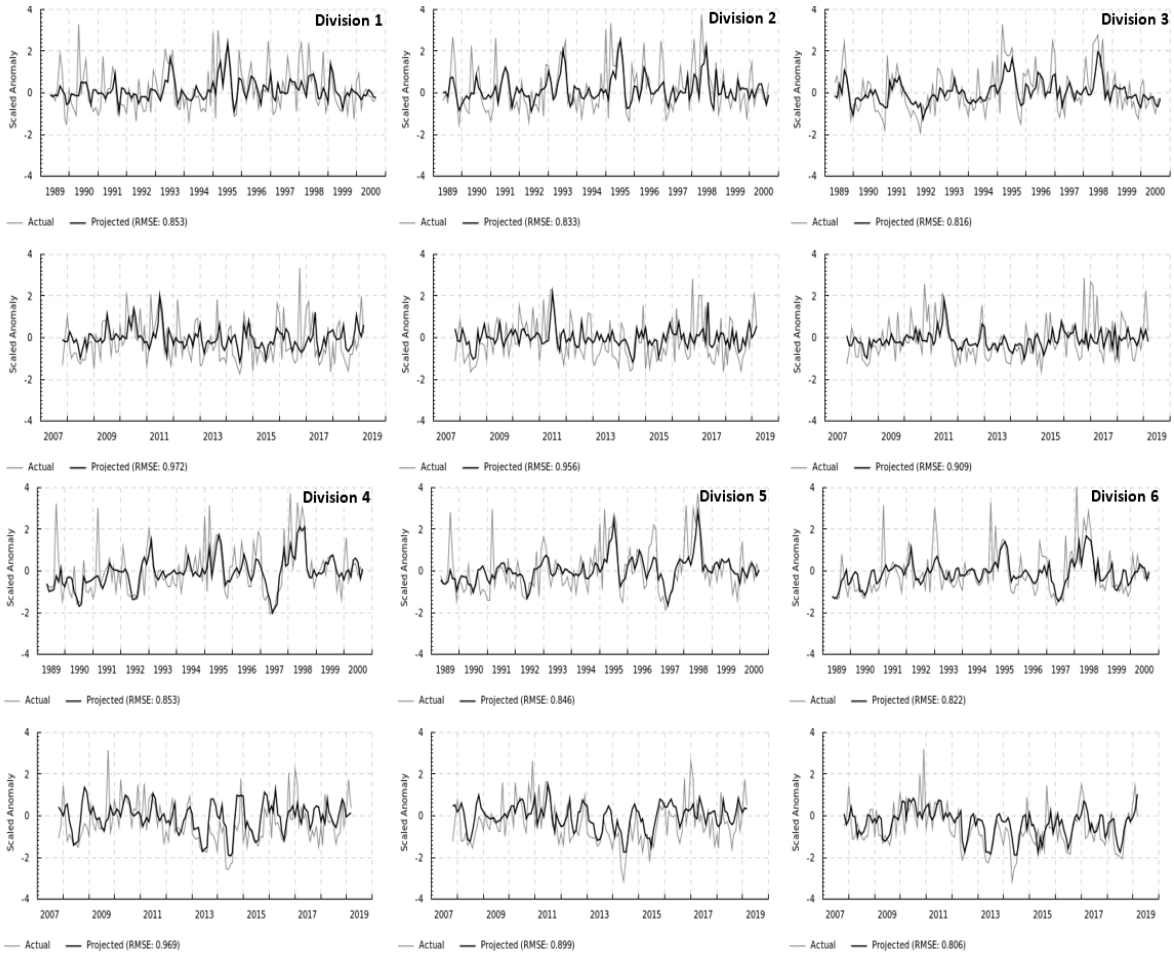

Figure 5: Convergence and validation results per-division on RNN

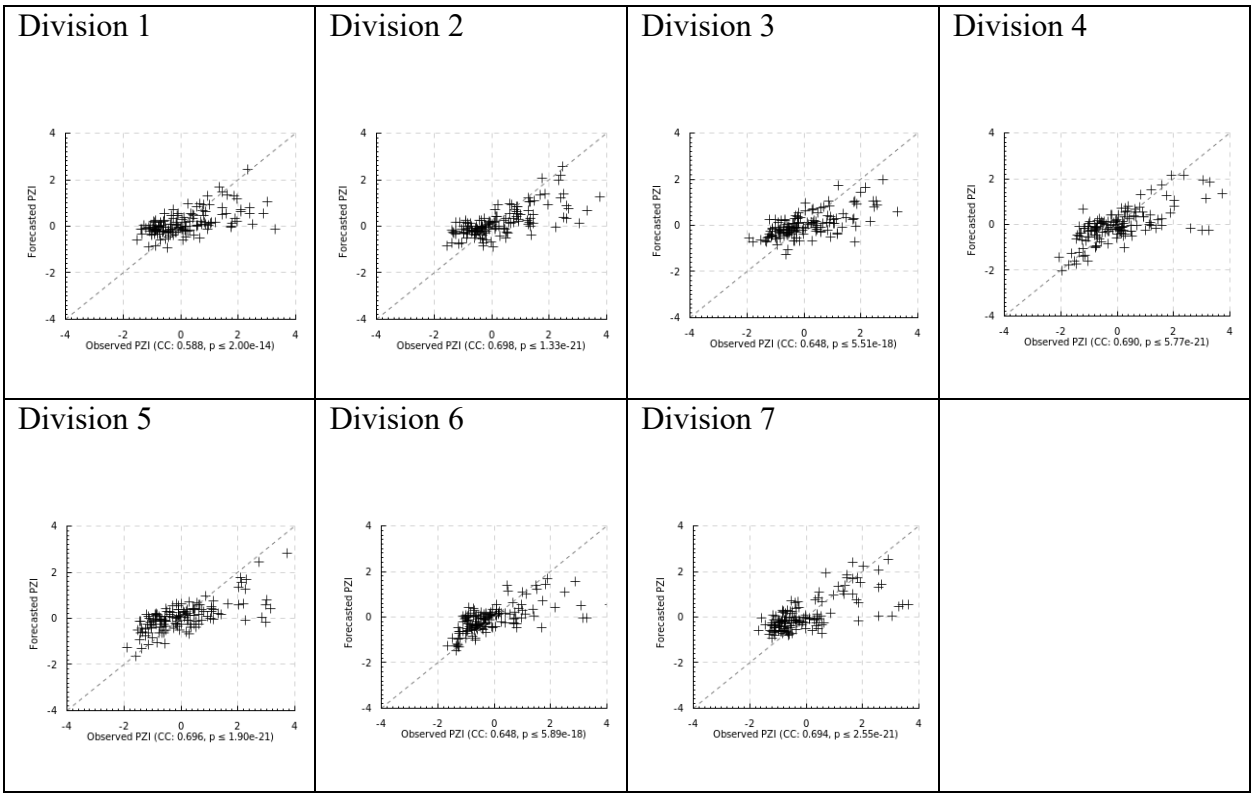

Figure 6: Convergence correlations for trained RNN networks per division

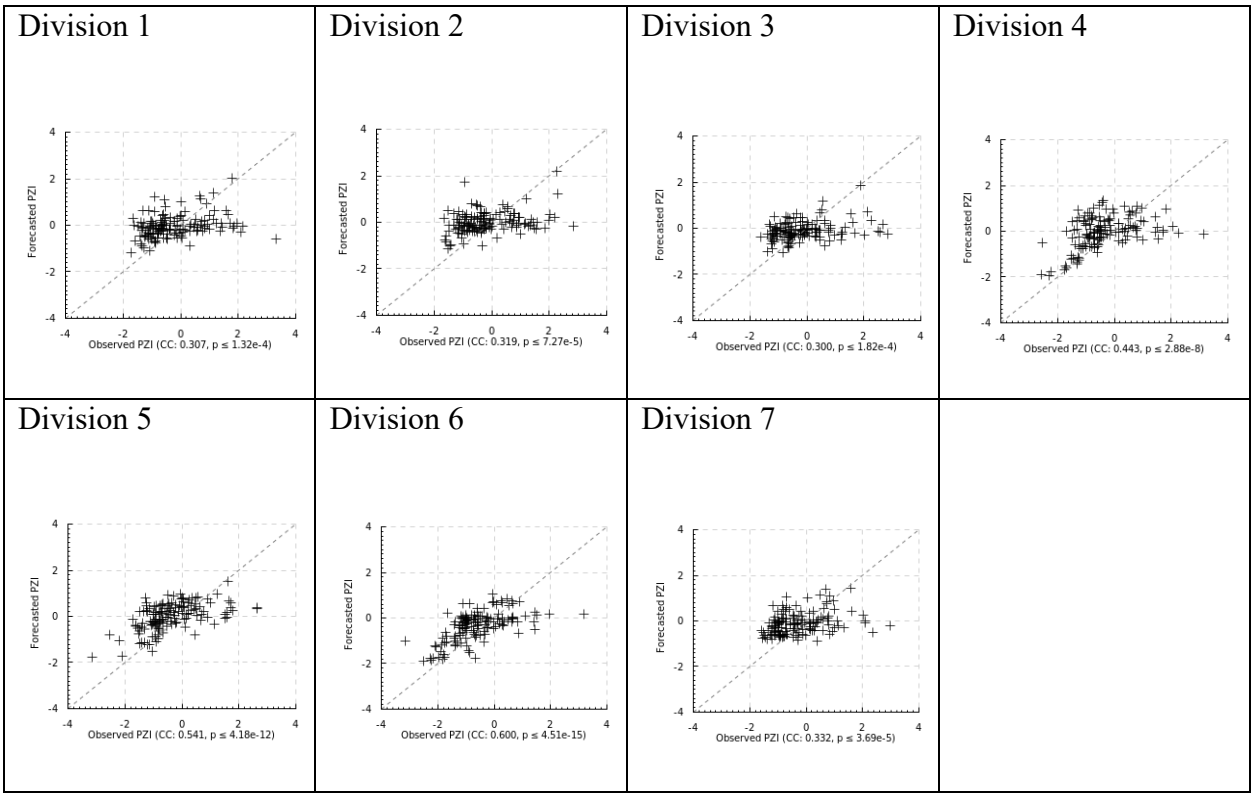

Figure 7: Validation correlations for trained RNN networks per division

Geographical Variation

From the table of model accuracies, we can observe that there exists some variation in the learnability for different regions. In particular, the model has the capability to learn California Climate Divisions 6 and 7 to a higher degree of accuracy than it can learn the other five divisions.

This variation in learnability is also apparent in the baseline autoregressive models, however; it performs stronger than statistical methods with the same input data set.

It is not clear why both RNN models and statistical models underperform in divisions 1-5, as opposed to 6 and 7. Initially, one would presume that coastal regions have smaller month-to-

month variability, and so have a higher in-system autoregressive memory. However, Division 1, the worst performing model of all seven, is a coastal region.

One similarity that Divisions 6 and 7 have that are not shared with the other divisions is the fact that they both completely cover the southern regions of California, located nearby the Gulf of California and equatorial Atlantic currents. This regulation might create more predictability within these regions, imposing limitations on month-to-month variations.

Sparkline Monitoring

By analyzing the activations of internal neurons outputted by each LSTM layer, we generate "sparkline" plots of each activation as a time series. Here are examples of sparkline outputs from the network trained on Division 6. First, its activation as it trains, looking at the window around the significant El Niño event in 1997:

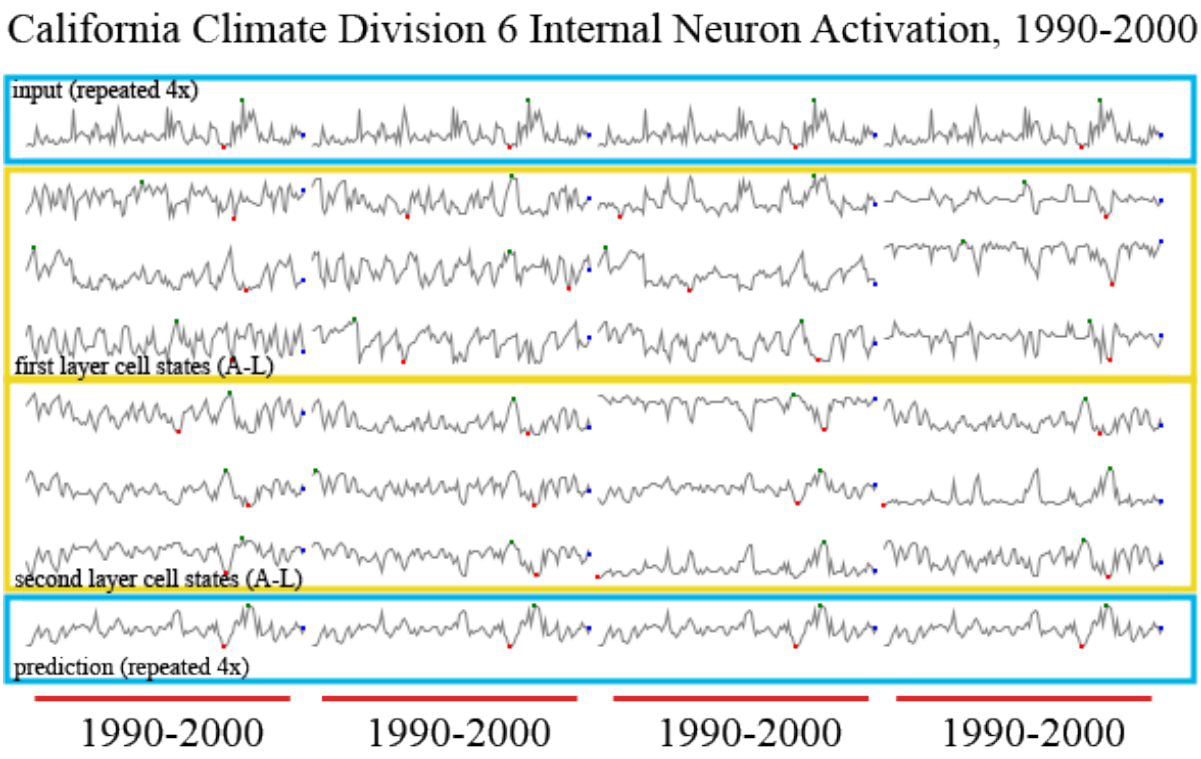

Figure 8: Labeled cell state time series sparkline of network being run over training data, Climate Division 6, 1990-2000.

The first row represents the input time series (PZI, where high is wet and low is dry), repeated four times for ease of vertical comparison. The final row represents the output time series predicted by the model, again repeated four times for ease of vertical comparison

Rows two through four represent the time-varying value of each of the 12 state cells of the first LSTM layer as a time series. Identify these as 1A, 1B, through 1L. Rows five through seven represent the cell states of the second LSTM layer. Id entify these as 2A, 2B, through 2L.

In this sense, cell states act as "random-access memory" of a computer, or as current chemical balances in the brain according to the biological analogy.

From this, we may identify high-level features that the network detects as useful from the input data. We can point out specific characteristics from cells 1G, 1J, 1H, 2C, and 2H:

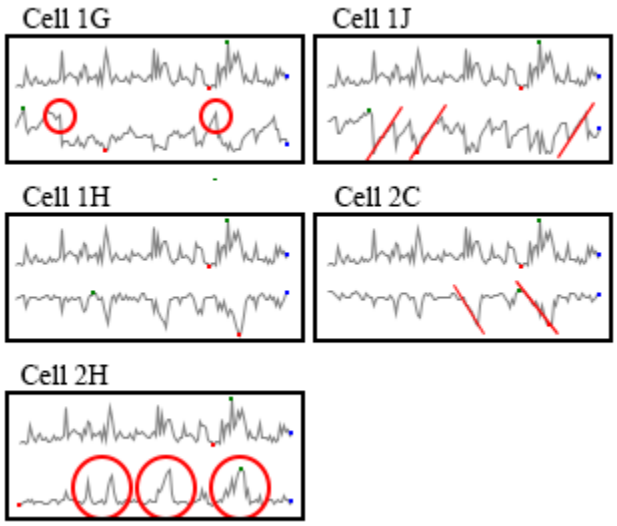

Figure 9: Highlighted features in the time series for specific cell activations for California Climate Division 6, 1990-2000. In each box, the top row is the input PZI, and the bottom row is the activation of a given cell state as a time series in response to the input data.

- Cell 1G characteristically spikes on the leading edges of large rainy seasons but drops out immediately during the body of the rainy season. This suggests that this cell has been selected to forecast the beginning of significant rainy seasons, but not to remain activated during the season itself
- Cell 1J appears to spike (negatively) according to sudden storm surges, and then decay in linear time. This suggests that Cell 1J keeps track of the "time since previous storm" for the network, providing the network with much needed temporal context.

- Cell 1H appears to simply keep track of the previous seen PZI. This suggests that the neural network incorporates an autoregressive model within its larger model and takes the previous seen PZI as an important part of predicting the next PZI.
- Cell 2C appears to grow linearly throughout the course of a large storm season. This suggests that 2C represents some sort of internal timer for the duration of a storm season, providing the network with context on how long a current storm season is lasting.
- Cell 2H is nearly always zero, except for during large storm seasons. This suggests that 2H is keeping track of whether we are currently within a large storm season, acting like a binary gating mechanism.

5.5 Conclusions

This paper has shown that the prediction of PZI using autoregressive methods based on LSTM recurrent neural networks has promise. Though the model's accuracy varies between divisions, it is consistently higher than baseline statistical methods by an average of 30%, and can prove to be an important tool that can be generalized over different climate divisions. The parallel training of different networks over different divisions is proven effective, and rapid training via pre-training also shows promise for generalization to many different division types. The role of machine learning and data science within climate and weather projection is still under discussion. However, we believe that, due to the strong predictive power shown by this method, it can be a useful tool guiding the process of climate science-based projections, as well as a tool in guiding types of indices and significant features within weather time series data. In addition, we have shown that interesting features may be easily found using the sparkline-based visualization method of internal activations. Future research will be dedicated to the evaluation of these sparkline activations methods and their comparison with known physical indices.

6 Empirical Mode Decomposition

6.1 Introduction

Empirical Mode Decomposition and the Hilbert-Huang Transform, developed at NASA in the 1990's by Norden E. Huang et al., are methods of time series decomposition and domain transformation, respectively. Their development was made to address shortcomings of competing and contrasting methods such as the Fourier and Wavelet decomposition and transform when applied to data from non-stationary and non-linear systems

The analysis of "empirical" time-series data (data corresponding to observations of physical systems) is a topic of great importance in scientific analysis within the domain of the physical system. For example, analyzing time series data of climate indices by identifying important features and characterizing and summarizing different aspects of the time series are all important intermediate steps for deciding a course of action, and to gain a deeper understanding of the system in question.

From a high-level perspective, Empirical Mode Decomposition aims to decompose a time series into non-stationary components that may be understood and analyzed within their own terms. The Hilbert-Huang transform reframes the series into a frequency-over-time domain (much like a sparse wavelet decomposition), by following the relative movement of each of these non-stationary components. Empirical Mode Decomposition aims to provide an "empirical" (numerical and physically rooted, in contrast to mathematically formal) method of decomposition of data into a series of modes in order of higher frequency variances, where each

mode is meant to represent physically meaningful aspects of the process at different time scales within the length of the time period. The Hilbert-Huang transform builds on this to recombine the relative contributions of each mode into frequency-time domain representation that tracks how each mode contributes to the total variation of the series.

Contrasting Comparable Models

Common comparable methods, such as Fourier and Wavelet transforms, operate under critical assumptions on the underlying data set. Namely, they assume that the data comes from a physical system that is exhibits some combination of linearity and stationarity. For example, the core assumption of both the Wavelet and Fourier Transform is that observations of the system in question can be understood as the sum of contributions from many independent systems, and that each independent system contributes to the final phenomenon in a linear way. The Fourier Transform also a further assumption of stationarity, and that the behavior of each of these subsystems is consistent and stable over the time period being analyzed.

The popularity of these transforms derives from the fact that the mathematical principles that underly the Fourier and Wavelet transform are strong mathematical ideals that allow for powerful tools for reasoning, drawing from the vast fields of linear algebra and real analysis and the conclusions they allow one to draw. In addition, many idealized physical systems are linear (due the linearity of many fundamental physical laws), and many empirically studied physical systems are close enough to linear to ignore effects of nonlinearity. Much time series data studied for physical systems is also taken at a scale where effects of stationarity is assumed or is at least achievable through formal normalization techniques.

However, the apparent ubiquitous status of these transforms gives much undeserved comfort to those seeking to apply them to many systems where they are not suitable. In much current literature and instructive material, Fourier and Wavelet transforms are recommended as first tools without first assessing their validity within the physical system being analyzed, whether informal or formal.

Because these tools are purely numerical processes detached from their mathematical premises, it is possible to use them in a variety of situations without giving thought to the physical sources of the time series processes in question. These methods give numerical answers that appear to be interpretable, without needing to give thought to the underlying mathematical principles underneath the analysis.

However, the physical reality of the time series being studied and analyzed is often not as mathematically clear. Many physical systems of note are not linear and do not obey general principles of linear superposition; many physical systems also do not exhibit stationarity in any meaningful extent, nor can be renormalized or reinterpreted to be stationary. When time series data from observations corresponding to these systems are subjected to Fourier and Wavelet decomposition, it is possible that the resulting components of decomposition are no longer physically meaningful if the effects of nonlinearity or non-stationarity are large enough. This leads to the inability to make conclusions from analysis, or potentially worse, errors in analysis or potential over-confidence in the results of analysis.

Effects in Nonlinearity

Linearity in a physical system can be described as the validity of the superposition principle: the principle that observed quantities can be understood in terms of independent systems whose

contributions to the observable quantity add together in an independent way. For example, if we consider the state of the system in phase space $p_t \in P$ and the progression of p_t to be governed by a time-progression operator $\mathcal{H}: P \to P$ representing the laws of physics of the system, we notate the time-progression of p_t as $\mathcal{H}p_t$. If $\mathcal{H}$ is a linear operator, we may instead divide our system as the linear combination of two arbitrary separate components

$$p_t = c_v u_t + c_v v_t$$

$$p_t, u_t, v_t \in P$$

in which u_t and v_t are themselves both valid positions in phase space P.

If $\mathcal{H}$, the operator representing the laws of progression of our physical system, is linear, this would mean that:

$$\mathcal{H}p_t = \mathcal{H}(c_u u_t + c_v v_t)$$

$$\mathcal{H}p_t = \mathcal{H}(c_u u_t) + \mathcal{H}(c_v v_t)$$

$$\mathcal{H}p_t = c_u(\mathcal{H}u_t) + c_v(\mathcal{H}v_t)$$

In other words, we can fully understand the progression of p_t through P without considering how $\mathcal{H}$ acts on p_t as a unit, and instead understand the progression of p_t simply in terms of how $\mathcal{H}$ acts independently on u_t and v_t, which may be much simpler for analysis.

If we know that our system is linear, we may take advantage of linear decompositions like the Fourier transform. We can decompose our point in phase space p_t into a linear combination of its Fourier components:

$$p_t = \int_{\omega} c_\omega u_t^\omega$$

(where $u_t^\omega \in P$ represents a single basis of the Fourier decomposition corresponding to frequency ω and c^ω corresponds to the Fourier component at that frequency ω)

Then, we can understand the progression of system p_t through time $\mathcal{H}p_t$ as:

$$\mathcal{H}p_t = \mathcal{H}\left(\int_{\omega} c_\omega u_t^\omega\right)$$

$$\mathcal{H}p_t = \int_{\omega} \mathcal{H}(c_\omega u_t^\omega)$$

$$\mathcal{H}p_t = \int_{\omega} c_\omega(\mathcal{H}u_t^\omega)$$

In other words, it is valid to understand the progression of system p_t through time $\mathcal{H}p_t$ in terms of how $\mathcal{H}$ acts *individually* and *independently* on each component, u_t^ω. And, ideally, the analysis of of $\mathcal{H}u_t^\omega$ is much simpler than the analysis of the whole part.

However, an attempt at similar analysis of $\mathcal{H}$ is non-linear will either provide meaningless results or forget all results. For example, it may be the case that

$$\mathcal{H}p_t = \mathcal{H}(u_t + v_t)$$

$$\mathcal{H}p_t = \mathcal{H}u_t + \mathcal{H}v_t + \mathcal{H}u_t\mathcal{H}v_t$$

In this case, any sort of decomposition analysis must take so-called "cross terms" into consideration, and we cannot meaningfully analyze $\mathcal{H}p_t$ in terms of only $\mathcal{H}u_t$ and $\mathcal{H}v_t$ unless nonlinear terms cross-terms can be shown to be insignificant.

While it may be possible to perform the numerical transformations corresponding to a Fourier transform on a time series, the results are not meaningful, and do not represent any actual real physical aspect of the system at hand if the system is not linear. Either we end up ignoring important extra terms like cross-terms or look at an aspect that is altogether unrelated to the underlying physical processes.

Non-linear physical systems and time-progression operators are commonplace in the study of physical systems, including wave dynamics.

Fourier and wavelet analysis both require a linear system in order to produce physically meaningful results from their numerical decompositions. Clearly, there is a need for a method of decomposition that does not depend on mathematical linearity.

Effects in Non-stationarity

Stationarity in a physical system describes, informally, a general property of time-symmetry in observed results. If observable X is considered as a stochastic variable under an unconditional joint probability distribution, stationarity is formalized as the claim that that probability distribution is time-symmetric, with no explicit dependency in time. The properties of this probability distribution – including mean and variance – are not dependent on time.

In practice, it is the constraint that certain statistical properties of this observable are unchanged under time-shift; such statistical properties must be identical at any point over the course of the time series. Usually, this is formalized by specifying exactly which statistical properties are taken to hold constant over time.

The Fourier Transform is fundamentally a method of the decomposition of stationary processes. The fundamental assumption of the Fourier Transform is that, given a time series, it can be characterized as the sum of pure sinusoidal terms, combined as a superposition based on magnitude and phase, together known as Fourier coefficients. The entirety of the time series is given a global Fourier coefficient for each frequency term. Locality and variations in the time domain are represented as global fixed quantities in the frequency domain – the magnitude and phase of each sinusoidal contribution is stated as fixed for the entire duration of the time series.

This, the Fourier Transform is ill-suited for situations where the statistical characteristics of the time series data change in a non-periodic way over the duration of the time series. A numerical fourier decomposition can be performed which assigns a complex number c_ω for each frequency, but this number is detached from any physical meaning or interpretation: there may be no such constant frequency component operating at that associated magnitude or phase throughout the duration of the time series. A number may be computed, but it is without any physical meaning, and is instead an artifact of the numerical quantification of the observable. Any interpretation based on it is unfounded within the actual mathematics of the Fourier transform or the physics of the system. The coefficient has no relation to any aspect of the physical system in situation with nonstationarity.

Wavelet transforms, in contrast, are designed to allow for the analysis of non-stationary data. Instead of yielding a fixed, unchanging coefficient, it describes each coefficient as itself a time series, and so the effects of each decomposed component's contribution preserve their time-locality. But, at least for Fourier transforms, non-stationarity renders any potential interpretation physically meaningless.

6.2 Methods

We will show that Empirical Mode Decomposition and the Hilbert-Huang Transform are suitable methods for the decomposition and analysis of a time series that is does not require linearity nor stationarity. First, we outline the process of Empirical Mode Decomposition and the Hilbert-Huang transform, in enough detail for implementation. In contrast to mathematical a-priori models such as the Fourier and Wavelet Transforms which are based on projections to a pre-selected choice of basis components, EMD and HHT derive the meaningful decomposed components based on the shape of the data directly.

6.3 Empirical Mode Decomposition

Empirical Mode Decomposition is an iterative process for extracting Intrinsic Mode Functions from a time series, all of which are mutually orthogonal contributions to the original time series of different scale frequency variations. The Hilbert-Huang transform then operates on these Intrinsic Mode Functions. However, IMFs on their own are a useful decomposition of a time series into constituent components.

Denote the original time series as $X(t)$, and each IMF $c_j(t)$ as a time series on the same domain. Empirical Mode Decomposition decomposes a time series into:

$$X(t) = r_n(t) + \sum_{j}^{n} c_j(t)$$

That is, a point-wise sum of each of the n mutually orthogonal IMFs, plus some residual term $r_n(t)$.

The iterative process of selecting each IMF is known as *sifting*, and it selects out a high-frequency component of the time series at each stage.

IMFs and Orthogonality

The utility of intrinsic mode functions arises out of their mutual orthogonality. They are a way to view individual aspects of the system as independent contributions to the full system, without bringing any a priori expectations of fixed frequency to play.

If we define the inner product between two time series as

$$\langle a, b \rangle = \sum_{t} a(t)b(t)$$

Then this orthogonality condition can be stated as

$$\forall j. \forall k \neq j. \langle c_j, c_k \rangle = 0$$

In practice, with these numerical and empirical processes, the a weaker, more acceptable orthogonality condition is satisfied:

$$\forall j. \forall k \neq j. \left| \frac{\langle c_j, c_k \rangle}{\sqrt{\langle c_j, c_j \rangle \cdot \langle c_k, c_k \rangle}} \right| < \epsilon$$

ϵ is the acceptable threshold of non-orthogonality.

The process of *sifting* is the process of separating out successive IMFs, under the condition of their mutual orthogonality, in order to understand the time series as a whole in terms of these empirically extracted components.

Sifting

The process proceeds in the following way:

1. Begin with r_0 representing the time series to be decomposed X. r_j will represent the residual term at step $j = 0$. As the algorithm proceeds and j increases, IMF terms of increasing frequency variation will be subtracted from the residual term at each step until no more IMF terms may be extracted.
2. At each step $j \in 1..n$, perform the process of extracting an IMF term. This process is known as the *sifting process* and will proceed until no IMF term can be extracted.
    1. Begin with $h_{j,0} = r_{j-1}$. $h_{j,i}$ will represent empirical refinements of the jth IMF term to be extracted. As the algorithm proceeds and i increases, $h_{j,i}$ will represent further refinements of the IMF term.
    2. At each step $i \in 1..k$:
        1. Create a cubic spline $e_{\max}$ connecting all local maxima on $h_{j,i-1}$, denoted as the maximal envelope.
        2. Create a cubic spline $e_{\min}$ connecting all local minima on $h_{j,i-1}$, denoted as the minimal envelope.
        3. Define $m_{j,i} = (e_{\max} + e_{\min})/2$ as the point-wise average of the envelopes from (i) and (ii): $m_{j,i}$ denotes a slow-moving "center of mass" for $h_{j,i-1}$
        4. Compute $h_{j,i} = h_{j,i-1} - m_{j,i}$. Because $m_{j,i}$ corresponds to a center of mass on $h_{j,i-1}$, $h_{j,i}$ is essentially $h_{j,i-1}$ re-centered about zero.
        5. Repeat (b) until some stoppage criteria has been reached. This will be discussed later. Denote the final i as k.
    3. Define jth IMF term $c_j = h_{j,k}$.
    4. Compute $r_j = r_{j-1} - c_j$, subtracting out the new IMF c_j.
3. Repeat (2) until residual term r_j is a monotonic function, and so (2.b.i) and (2.b.ii) cannot be computed. Denote the final j as n.

After this process, we arrive at our set of IMFs, c_j , $j \in 1..n$, and a final residual term r_n, in a fashion such that $X = r_n + \sum_j^n c_j$, as previously claimed.

This process should also produce each c_j in a way such that they are mutually orthogonal, or nearly orthogonal. If perfect orthogonality is desired, promise has been shown by methods refining IMFs via Gram-Schmidt orthonormalization.

Stopping Condition

The process of extracting IMFs has a natural stopping point, since steps (2.b.i) and (2.b.ii) both will not be computable if the residual r_{j-1} (and thus $h_{j,0}$) is monotonic. For a monotonic function, no local minima or local maxima will exist, making both the maximal and minimal envelopes undefinable. This gives us a natural stopping point for the extraction of IMFs, and thus for n, the index of the final IMF term and the final residual.

However, the sifting process, used for finding each IMF empirically, does not have a well-defined stopping process. Conceptually, each IMF seeks to remove low-frequency shifts in the center of mass of the current residual series, re-centering the series to have a center of mass about 0. This quality is enough to sustain the weaker orthogonality condition mentioned above, between each successive IMF. However, because this notion of "center of mass" m_i is not so well-defined, this process involves repeatedly subtracting out further refinements of m until a suitable one is found.

The stopping condition for this refinement (sifting) is left up the choice of the implementation or those running the decomposition. Four commonly used ones include Standard-Deviation Based Method, S Number Criterion, Threshold Method, and Energy Different Tracking. All of these are premised on the assumption that $h_{j,k}$ converges on k, and we quit with a satisfactory IMF once we achieve an arbitrarily close approximation to this point of convergence.

Standard Deviation

The original proposed stopping condition, based on standard deviation, is inspired by the Cauchy Convergence Test. In it, we compute a measure of standard deviations of differences between each improvement of $h_{j.k}$:

$$SD_{j,k} = \sum_{t} \frac{\left|h_{j,k-1}(t) - h_{j,k}(t)\right|^2}{h_{j,k-1}^2(t)}$$

This indicates, approximately, a measure of global convergence for $h_{j,k}$ by computing how much each successive k changes h. We then stop with a final IMF candidate when this measure decreases past a pre-determined lower threshold.

The intent behind this method is to claim that $h_{j,k}$ converges on k, and we define a numerical measurement of deviation from this point of convergence. We then quit once this numerical measurement is arbitrarily small enough.

S-Number Criterion

The S-Number criterion provides another way of defining convergence. Define $s_{j,i}$ as the number of extrema and zero-crossings of IMF candidate $h_{j,i}$. The S-Number $S_{j,i}$ of $h_{j,i}$ is then defined as

$$S_{j,i} = \max_{n} \forall w \in \{1..n\}. \left| s_{j,i-w} - s_{j,i} \right| \leq D$$

Where D is a pre-chosen integer threshold, typically 0 or 1. That is, it is the length n of the longest possible run backwards from $h_{j,i}$ where $s_{j,i-w}$ for all $w \leq n$ is within a range of D from $s_{j,i}$. If

$D = 0$, it is the current streak of IMF candidates where $s_{j,i}$ does not change. If $D = 1$, it is the current stream of IMF candidates where $s_{j,i}$ values differ at most by one.

In this criterion, convergence is defined as a stability of $s_{j,i}$, or a stability of the number of extrema and zero-crossings of successive IMF candidates.

A pre-chosen limit for $S_{j,i}$ is chosen, and an IMF candidate $h_{j,i}$ is deemed as suitable if $S_{j,i}$ exceeds this pre-chosen limit. In other words, it is deemed suitable as soon as a consecutive run of suitable length is discovered.

<u>Rilling-Flandrin-Goncalves Threshold Method</u>

One method, proposed by Rilling, Flandrin and Gonçalvés, involves inspecting only the current $h_{j,i-1}$ and $m_{j,i}$. This method defines convergence as the point where $m_{j,i}$ is small enough that the IMF candidate is sufficiently zero-centered. Alongside defining $m_{j,i} = (e_{\max} + e_{\min})/2$ as the point-wise average of the maximal and minimal envelops, also define *amplitude* term $a_{j,i} = (e_{\max} - e_{\min})/2$, corresponding half the range of between the envelopes. From these, we can define an *evaluation function* $\sigma_{j,i} = |m_{j,i}/a_{j,i}|$, the result of point-wise division between the absolute values of terms in $m_{j,i}$ and $a_{j,i}$.

This method is parameterized by three chosen terms: α, θ_1, and θ_2. Use α to partition the total time series of length T into two intervals, the first running from $t = 0$ to $t = t_\alpha = (1 - \alpha)T$, and the second running from $t = t_a$ to $t = T$. The stopping criteria is for $\sigma_{j,i}$ is met when both:

$$\forall\, 0 < t < t_\alpha.\ \sigma_{j,i}(t) < \theta_1$$

$$\forall\, t_\alpha < t < T.\ \sigma_{j,i}(t) < \ \theta_2$$

Rilling et. Al. propose sensible values for $\alpha, \theta_1, \theta_2$ as 0.05, 0.05, and 0.5.

The intent is to specify a global condition of smallness for $m_{j,i}$ in comparison to vertical range of the time series itself.

Interpreting IMFs

The fundamental properties used when interpreting both a single IMF and a collection of IMFs are:

- Mutual orthogonality
- Summation to the original time series X, i.e. $X = r_n + \sum_j^n c_j$
- Decreasing frequency of oscillations

Each IMF can, to an extent, be interpreted as a self-contained and independently progressing system, and the total observed behavior of the system can be taken to be the sum behavior of each IMF.

Due to the orthogonality of each IMF, it can be said that IMF terms each describe a *different* aspect of the system at play. The internal aspects of the system manifesting in the first IMF, for instance, represent *different* internal aspects of the system manifesting in the second IMF. Orthogonality expresses the intent that each IMF models a different, independent internal process.

The property of summation to the original time series ensures that looking at each IMF gets close to the totality of all internal aspects of the system. By analyzing each of the finite number of IMFs, one can be assured that each potential internal aspect is covered in typically at most one IMF.

The property of decreasing frequency of oscillations from one IMF to the next orders the IMFs under some arbitrary structure. While the collection of IMFs is essentially unordered, the natural structuring from high-frequency oscillations describing high-frequency processes to low-frequency oscillations describing low-frequency processes is useful for the organization of conclusions and thoughts, and helps place specific types of processes somewhere along an ordered structure in the space of IMFs.

6.4 Hilbert-Huang Transform

One mutually orthogonal IMFs have been extracted, it is possible to perform a Hilbert Spectral analysis on the results, by examining the instantaneous frequency for each IMF. The result is what is commonly referred to as "skeleton lines", illustrating the motion of each IMF over the instantaneous frequency space as a function of time.

This can be contrasted to the wavelent transform, which provides a dense function on the space of frequency against time. Instead of a fully defined and dense mapping, instead EMD and HHT produce a sparse picture of k lines moving through frequency space, over time – it is sparse in frequency, but not in time. For a wavelet transform, information can be gleamed from the relative magnitudes of different locations in frequency-time space. For the HHT, information can be gleamed more naturally as a progression of relative frequency densities as a function of time. The story naturally highlights the most significant frequency contributions at each point in time, preserving the flow in time domain.

The eventual picture painted by a Hilbert-Huang transform is that of each IMF tracing out a continuous path along frequency space over time. By analyzing the motion of the collective cluster

of IMFs over time, a picture of an evolving system moving through different frequency characteristics over time emerges.

Instantaneous Frequency

The concept of instantaneous frequency of an IMF looks at local varions in the magnitude of time series IMF and interprets it as oscillatory motion with some instantaneous frequency. It is different from the concept of a global frequency of an IMF – rather, it is a property about any local continuous neighborhood in an IMF.

Instantaneous frequency can be computed as a time series, with a frequency value corresponding to the local neighborhood around each point in time in the original time series.

For time series h, we can denote its Discrete Hilbert Transform $\tilde{h}\ as$:

$$\tilde{h}(k) = \begin{cases} \frac{2}{\pi} \sum_{t \text{ odd}} \frac{h(t)}{k-t};\ k \text{ even} \\ \frac{2}{\pi} \sum_{t \text{ even}} \frac{h(t)}{k-t}; k \text{ odd} \end{cases}$$

We can then construct a complex series d_j from IMF c_j, defined as:

$$d_j = c_j + \tilde{c_j}$$

d_j(t), as a time series, is known as the *analytic signal*, and corresponds to a helix moving along time through the complex plane. If we re-write d_j in its polar form, in terms of instantaneous magnitude a_j and phase ϕ_j:

$$a_j^2 = c_j^2 + \tilde{c_j}^2$$

$$\tan \phi_j = \frac{\widetilde{c_j}}{c_j}$$

$$d_j = a_j e^{i\,\phi_j}$$

There is more than one potential solution for ϕ_j. Under the physical interpretation, we choose a version of ϕ_j such that $\phi_j(t)$ is *monotonically increasing*, interpreting the progression of the phase of the signal as marching forward through time. Furthermore, $\phi_j(t+1)$ must be the unique *minimal* solution to $\tan\big(\phi_j(t+1)\big) = \frac{\widetilde{c_j}(t)}{c_j(t)}$ such that $\phi_j(t+1) \geq \phi_j(t)$. This ensures that

$$0 \leq \big[\phi_j(t+1) - \phi_j(t)\big] < 2\pi$$

We may then define *instantaneous frequency* ω_j of IMF c_j as the rate of change in ϕ_j:

$$\omega_j(t) = \frac{1}{2\pi}\frac{\delta\phi_j(t)}{\delta t}$$

This can be estimated using finite difference methods; the simplest method would be a single backwards difference method:

$$\omega_j(t) = \frac{1}{2\pi}\frac{\phi_j(t) - \phi_j(t-1)}{\Delta t}$$

Alternatively, a centered difference method may be used, at the cost of losing one potential known value in ω_j.

The above constraints on the progression of ϕ_j ensure that ω_j is always within the range:

$$0 \leq \omega_j < \frac{1}{\Delta t}$$

This effectively restricts the instantaneous frequency to be non-negative (always moving forwards) and within a given measurable range. The minimal resolution of our discrete time series is not able to meaningfully detect a frequency greater than $\frac{1}{\Delta t}$. A smaller timestep of discretization would allow for the resolution of a greater instantaneous frequency. Under this discretization, all frequencies $f + \frac{n}{\Delta t}$, for positive integer n, are fundamentally indistinguishable.

Limitations of Resolution of Discretization

In theory, the restriction on $\phi_j(t+1)$ being the minimal solution of $\tan\big(\phi_j(t+1)\big) = \frac{\tilde{c}_j(t)}{c_j(t)}$ is not a priori justified. However, this is chosen because the minimal resolution of the time series cannot resolve or distinguish any frequency *faster* than $f_{\text{max}} = f_{\text{sample}} = \frac{1}{\Delta t}$, and so frequencies $f + \frac{n}{\Delta t}$, for positive integer n, are indistinguishable. The restriction on the progression of $\phi_j(t+1)$ ensures that of this family of indistinguishable frequencies, the smallest is chosen (arbitrarily).

Ontologically, this means for a time series with $\Delta t = 0.1s$, the highest possible distinguishable instantaneous frequency is $f_{\text{max}} = \frac{1}{\Delta \text{t}} = 10$ Hz. This means that a "true" physical instantaneous frequency of 15 Hz will be interpreted as an instantaneous frequency of 5 Hz. This is an unavoidable epistemological issue arising from the limitations of discretization, as the two numerical values of instantaneous frequency are indistinguishable in the discretized time series. This is an artifact of the discretization.

However, if this same time series is resampled at $\Delta t = 0.05s$, the highest possible distinguishable instantaneous frequency becomes $f_{\text{max}} = \frac{1}{\Delta \text{t}} = 20$ Hz. At this range, a "true" physical instantaneous frequency of 15 Hz can be distinguished from a 5 Hz frequency.

Conceptually, this limitation implies that we cannot observe frequencies higher than the frequency of sampling itself. For example, for a time series sampling one observation per day, any frequencies higher than day^{-1} cannot be resolved. An instantaneous frequency of $5\ \text{day}^{-1}$ (five times per day) cannot be meaningfully distinguished from $4\ \text{day}^{-1}$ (four times per day).

Skeleton Lines

From an original time series X, we identify k mutually orthogonal IMFs $c_j, j \in \{1..k\}$ such that $X - \sum_j^k c_j$ is monotonic and negligible.

From each IMF c_j, we perform a discrete Hilbert transform and interpret a_j and ω_j as its instantaneous magnitude and frequency, respectively. $\omega_j(t)$ represents the instantaneous frequency of c_j at time t, and $a_j(t)$ represents its magnitude as a contribution to the overall power of the series at the associated $\omega_j(t)$.

ω_j and a_j together specify what is called a *Skeleton Line* that traces a path along frequency-time space, each point on that path having a location along the frequency aspect associated with ω_j and a magnitude and intensity associated with a_j.

The full collection of k skeleton lines resulting from the IMFs yielded by Empirical Mode Decomposition form the *Hilbert-Huang Transform* of the time series.

Interpretation of the Hilbert Huang Transform

The Hilbert-Huang transform takes a series from a single-valued scalar-codomain function dense on the time domain into, essentially, a multi-valued pair-codomain function dense on the time domain, where each 2-vector represents frequency and magnitude

$$X\text{:time} \rightarrow \text{value}$$

$$\mathrm{HHT}(X)\text{:time} \rightarrow (\text{freq} \times \text{value})^k$$

Because of this, the most immediately natural way to represent a plot of the result of HHT according to this structure is as a plot in frequency against time, with multiple lines stacked upon each other. Each line represents a different IMF, and each one traces a path according to $\omega_j(t)$. Each point in the line is then associated an amplitude, intensity, or energy term according to with $a_j(t)$.

In the structure of this transformation, it is tempting to view frequency and amplitude as interchangeable. Under this viewpoint, one may decide to plot amplitude over time, and associate each point in each line with the frequency at that point.

However, due to the construction of IMFs, looking at frequency against time provides one interesting benefit: because IMFs tend to decrease in instantaneous frequencies as more are constructed, this provides a natural structure to the Hilbert-Huang Transform. In practice, each skeleton line carves out its own "band" in frequency, and lines are stacked upon each other in a way where they rarely intersect. Viewing a Hilbert-Huang Transform structured in this way provides a natural stratification to the structure of the transformation, in a way that plotting amplitude against time cannot.

In addition, while amplitudes may vary strongly between IMFs, with no natural limit or bound, frequency is well-behaved: always, $0 \leq \omega_j < \frac{1}{\Delta \mathrm{t}}$. Even plotting period ($T = \frac{1}{\omega}$), which ranges from $\Delta t < T < \infty$, having the notion of banded frequencies provided by the increasing-frequency (or decreasing-period) structure of the IMFs keeps the skeleton lines organized in a clean way when presented this way.

Physically, skeleton lines can be interpreted as the progression of different aspects of the original time series through frequency space; the relative intensity at each point indicates which lines were dominant at which periods in time.

Different Projections of HHT

This skeleton line representation is unusual with respect to other common decompositions and transforms, and at times it may be difficult to extract useful summary terms based on this transformation.

Because of this, there are numerous ways to project this fundamental structure into alternative structures and presentations that may be more illuminating and present more natural ways to extract summary terms.

<u>HHT Sparse Spectrum</u>

The *Hilbert-Huang Transform Spectrum* takes each time step as a separate slice and provides a sparse spectrum at each point in time. The frequency spectrum at time t consists of k mass points: the point at each ω_j, with intensity a_j. Taken in this way, each time t can be visualized in a way like that of a Fourier Transform, with different intensities corresponding to different frequencies. However, the main qualitative difference is that, while a Fourier Transform is *dense* on frequency space, this spectrum is *sparse* on frequency space, with only k mass points.

<u>HHT Dense Spectrum</u>

However, it can be useful to densify this frequency space by discretizing it into a fixed number of divisions between 0 and $\frac{1}{\Delta t}$. It can also be natural to discretize the period space (or its logarithm),

between Δt and ∞. We may denote this discretization schema with $[-]$, where $[f]$ represents the equivalence class that frequency f falls under.

Under such a discretization, we may define the HHT Dense Spectrum by associating each frequency discretization band with the sum of all IMF amplitudes at the point where they enter the band of discretization.

$$\mathrm{HHT}(X)(t,[f]) = \sum_{j,[\omega_j(t)]=[f]}^{k} a_j(t)$$

Under this formation he spectrum at each point begins to look qualitatively like that of a discrete Fourier Transform, and what is produced is essentially a *separate frequency spectrum* for each point in time t.

In the end, this behaves like a windowed Fourier Transform, where each time slice provides a separate frequency spectrum breakdown.

This dense spectrum representation also is very similar to the result of a Discrete Wavelet Transform, as well: it associates each point in frequency and time space with an intensity, in the same way that a Discrete Wavelet Transform behaves.

<u>Marginal Spectrum</u>

If frequencies are discretized according to some scheme $[-]$, a *marginal spectrum* may be generated:

$$h_X([f]) = \sum_t \mathrm{HHT}(X)(t,[f])$$

$[f]$ represents the *equivalence class* of some frequency f according to the chosen discretization scheme. To compute $h([f])$, sum over the *magnitudes* $a_j(t)$ of each IMF component over time where the frequency discretization class $[\omega_j(t)]$ matches the marginal frequency you are computing. We may also define the *mean marginal spectrum*, normalized over time period length:

$$n_X([f]) = \frac{h_X([f])}{T}$$

The Marginal Spectrum associates each frequency discretization class to a total power. This assigns a total power for any given frequency over the entire course of the time series, and qualitatively behaves like a Discrete Fourier Transform, which also associates frequencies to total power over the course of a time series.

Unlike a Fourier Transform, which relies on principles of stationarity and linearity, the marginal spectrum via a Hilbert-Huang transform is empirically derived and therefore has a more physically meaningful interpretation.

This marginal spectrum sums over time, and so loses any sort of temporal resolution. It also sums over each IMF, so loses the distinction of separation of processes that IMFs confer. In return, a total summary over the course of the time series can be analyzed.

<u>Instantaneous Energy</u>

The instantaneous energy of the time series at any given point can be computed as

$$E[X](t) = \sum_{j}^{k} a_j^2(t)$$

This measure sums the total squared amplitude contributions of each IMF, providing a measure of the instantaneous energy of the system at any given point in time.

This can be thought of as a marginalization along the time axis. It sums over all frequency contributions, so loses the notion of travel through frequency space, and sums over each IMF, which loses the separation of processes that IMFs confer. In return, a single quantity as a function of time can be analyzed.

Degree of Stationarity

One may also compute the Degree of Stationarity at each (discretized) frequency, defined by Huang et al., 1998

$$\mathrm{DS}_X([f]) = \frac{1}{T}\sum_{t}^{T}\left(1 - \frac{\mathrm{HHT}(X)(t,[f])}{n_X([f])}\right)^2$$

This computes a measure of stationarity along each frequency band indicated by the discretization $[-]$, and ranges between 0 and 1, where 0 indicates perfect stationarity and 1 indicates perfect non-stationarity. For a purely stational frequency band with $\mathrm{DS}_X([f]) = 0$, original signal X may be treated as stationary along frequency band $[f]$. Under this circumstance, a Fourier Transform of X would yield physically meaningful results along that any frequency band $\mathrm{DS}_X([f]) = 0$, because that frequency band satisfies the stationarity constraint for Fourier Analysis.

We may also look at a modified version of the Degree of Stationarity, the Degree of Statistical Stationarity, which averages the Hilbert-Huang Transform over a time window

$$\mathrm{DSS}_X([f],\Delta T) = \frac{1}{T}\sum_{t}^{T}\left(1 - \frac{\overline{HHT(X)(\Delta T \text{ about } t,[f])}}{n([f])}\right)^2$$

The overbar indicates an average of HHT results with a window of ΔT about t. The Degree of Statistical Stationarity may be useful when characterizing the time series as random value.

Typically, $\mathrm{DS}_X([f])$ has a higher value than $\mathrm{DSS}_X([f],\Delta T)$. In the limiting case, DSS approaches DS as ΔT approaches zero, or becomes small compared to period $\frac{1}{[f]}$. This shows that even though a signal may be overall non-stationarity at a given frequency band, there can often be windows bands ΔT wide that exhibit local stationarity, especially at high frequency, or low $\frac{1}{[f]}$.

<u>Dominant and Expected Frequencies</u>

Other summary statistics may be computed for each time step based on ω_j and a_j. For instance, one may compute the *dominant* frequency at each time point

$$f_{\mathrm{dom}}(t) = \omega_{\arg\max_j a_j(t)}$$

Or, alternatively, a dominant frequency according to some discretization schema $[-]$:

$$\hat{f}_{\mathrm{dom}}(t) = \arg\max_{[f]} \mathrm{HHT}_X(t,[f])$$

And the center of mass of frequency for each point:

$$f_{\mathrm{com}}(t) = \frac{\sum_j^k a_j(t)\,\omega_j(t)}{\sum_j^k a_j(t)}$$

These give a measurement of the most significant frequency contribution at each point in time and may indicate the dominance of certain frequency bands within the signal as a function of time.

6.5 Analysis of Simulated Data

Here we apply Empirical Mode Decomposition and Hilbert-Huang Transform to simulated data. We construct data in a way to show non-stationarity and present it as a non-linear combination of internal systems.

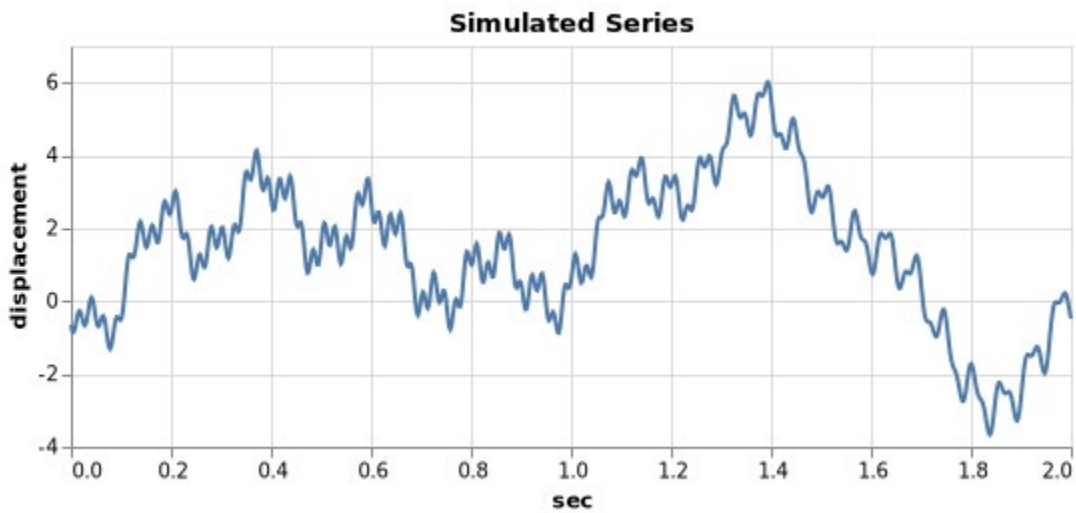

Figure 10: Simulated time series for EMD/HTT demonstration

This series is constructed as a sum of five components, each with varying frequency and amplitude in contribution:

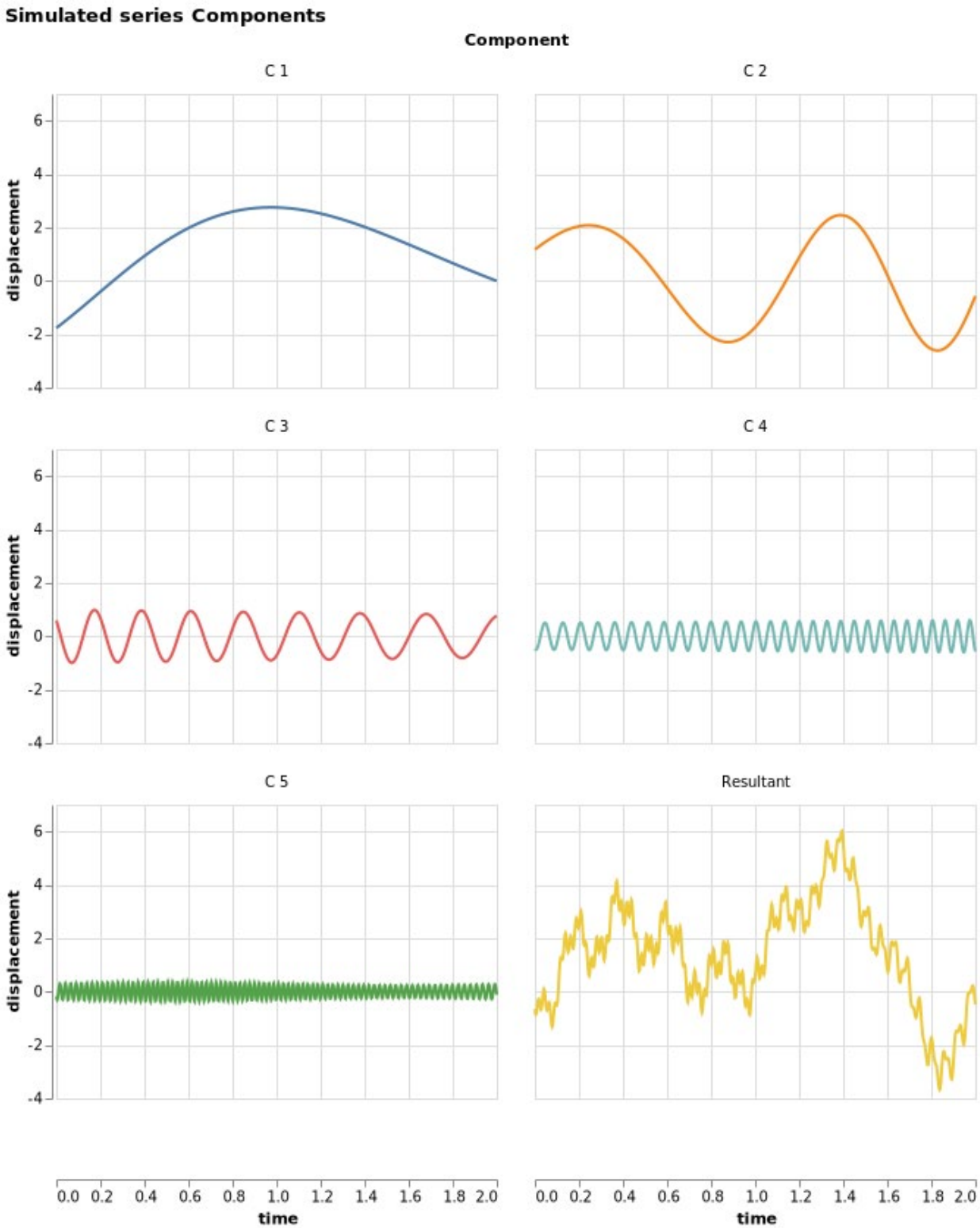

Figure 11: Underlying components for the simulated series

If we plot out the amplitudes of each component as a function of time, we get a skeleton plot:

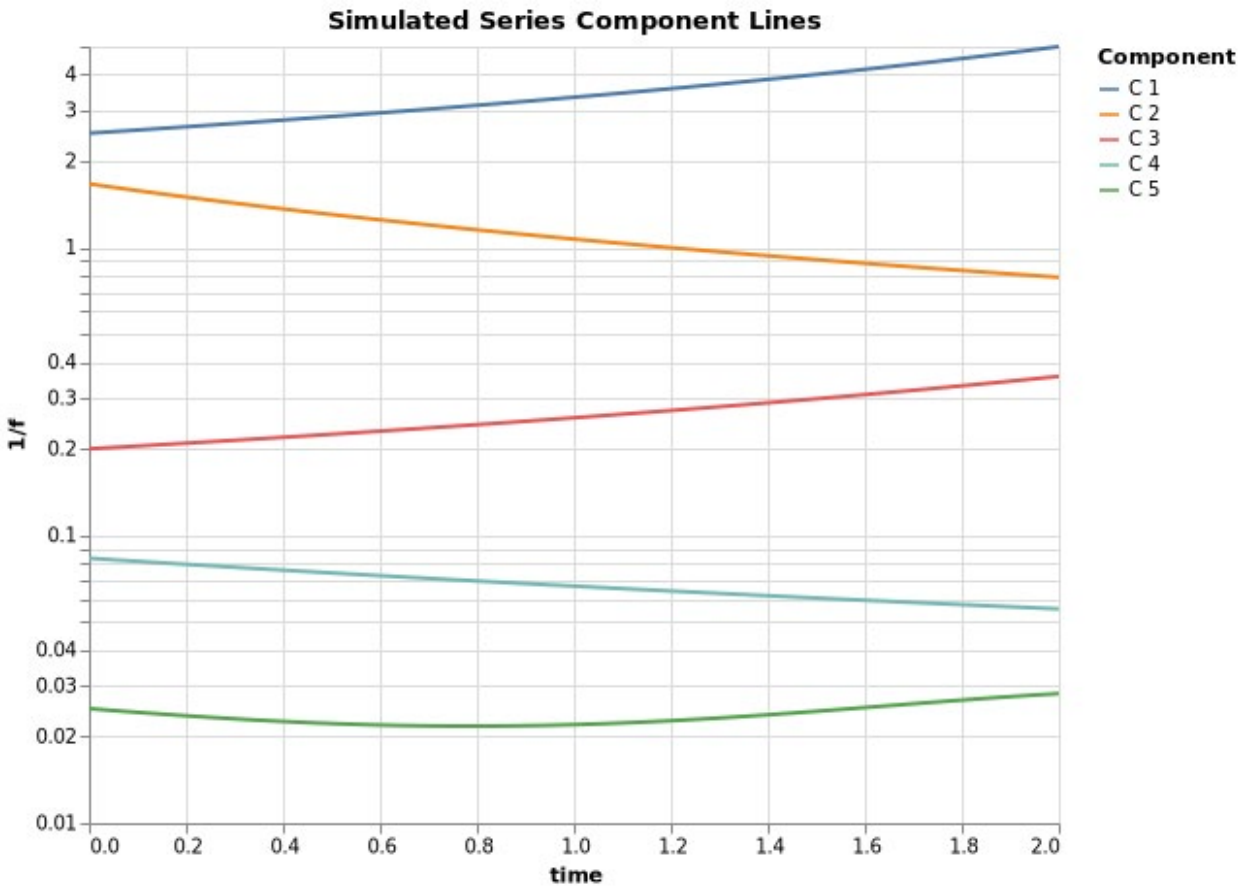

Figure 12: Components of simulated series in frequency-time space

For a more accurate illustration, we may also plot the variation in amplitude over time by varying the relative thickness of each line as a function of time:

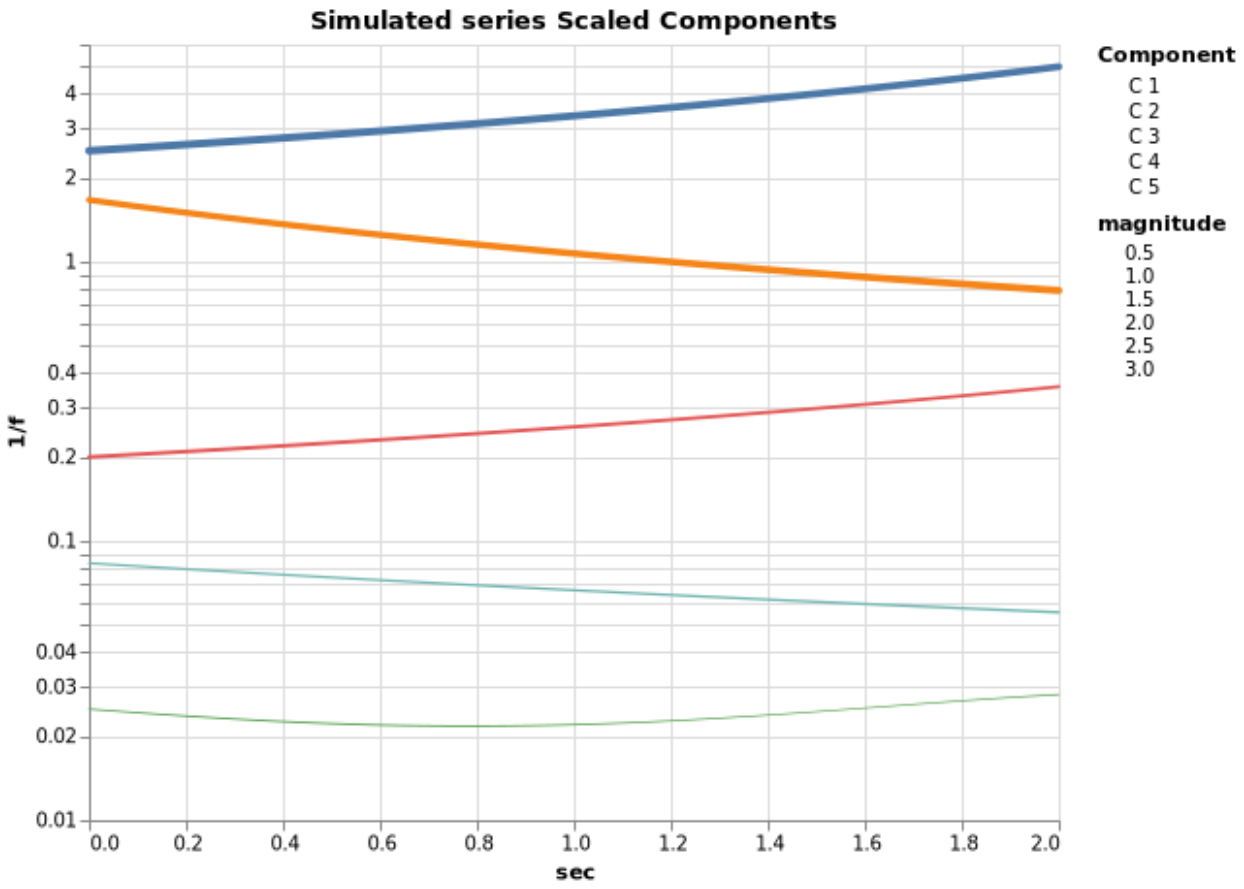

Figure 13: Components of simulated series in frequency-time space, scaled by magnitude

On these, we may perform Empirical Mode Decomposition, which aims to separate out each original component contribution.

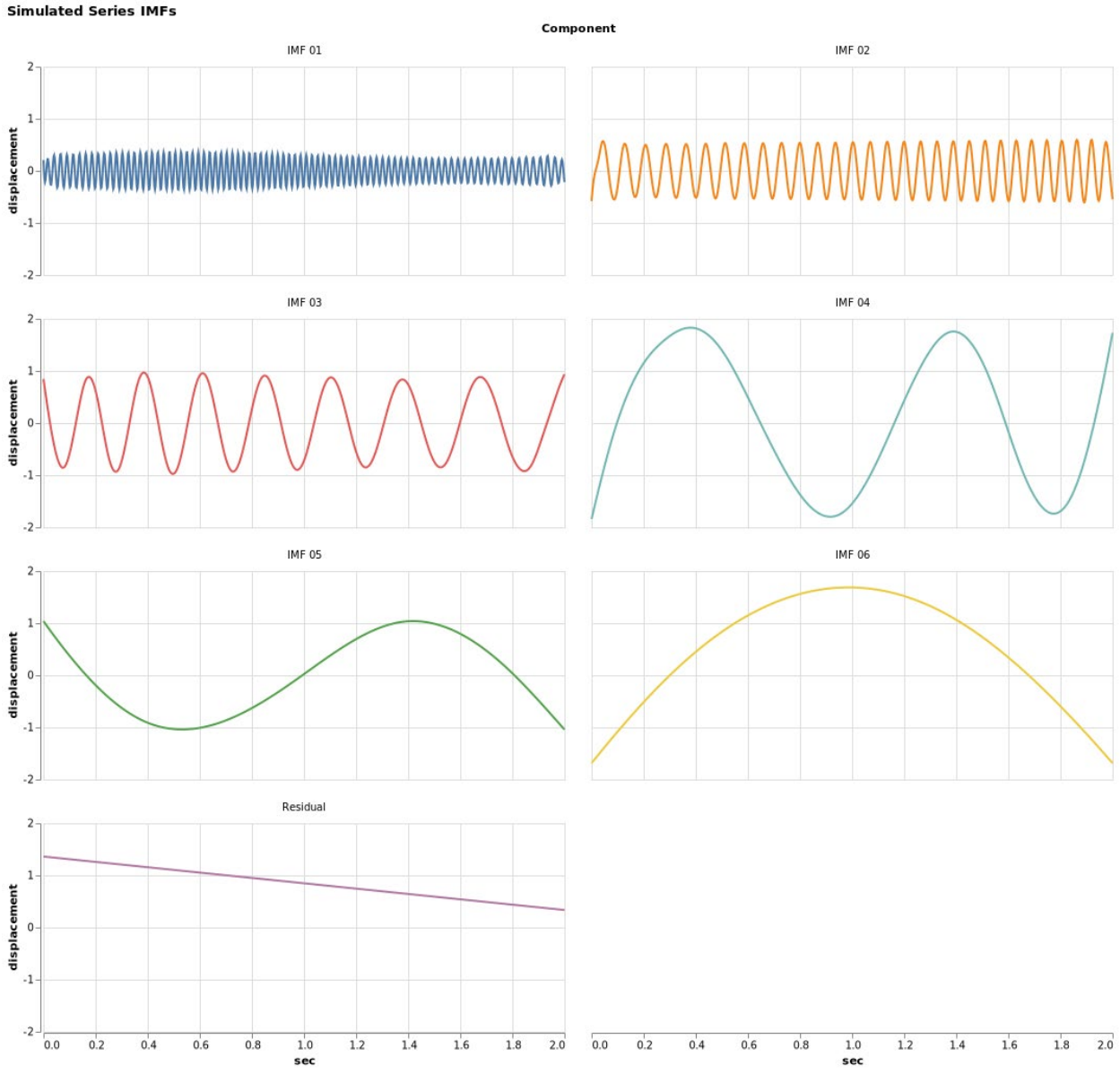

Figure 14: Result of EMD decomposition on simulated series

We see that the EMD does appear to recover many of the original components. IMF 1 corresponds to original component 5, IMF 2 corresponds to component 4, IMF 3 responds to component 3, IMF 4 responds to component 2, and IMF 6 corresponds to component 1. The only caveat appears to be that IMF 5 does not seem to correspond to any original component and is instead an artifact of the decomposition.

To illustrate this, we may compute a grid of mutual alignments $\frac{<u,v>}{\sqrt{<u,u><v,v>}}$ based on each pair of inner products, scaled:

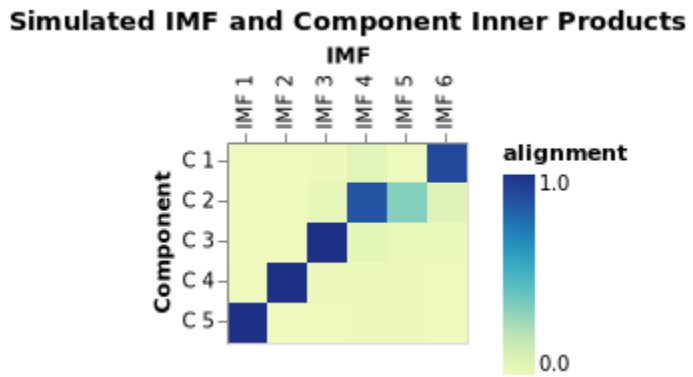

Figure 15: Inner products between IMFs and expected components for simulated series.

This shows us that IMFs 1, 2, and 3 almost perfectly recover original components 5, 4, and 3, respectively. IMFs 4 and 6 appear to recover components 2 and 1, respectively, to a large degree, but not to as perfect an extent as 1, 2, and 3. IMF 5 here seems to share some part with Component 2, but otherwise seems to be purely an artifact of the decomposition. Numerically, the alignment appears to trend slightly weaker as frequency decreases.

Each IMF is mutually orthogonal:

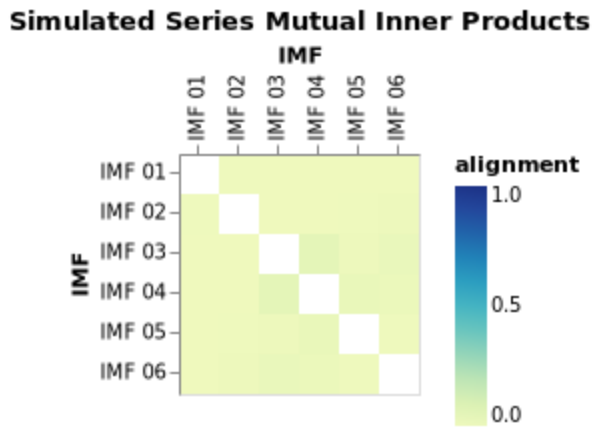

Figure 16: Inner products between different IMFs for simulated series.

Of these, IMF 3 and 4 are the most non-orthogonal, but at an alignment factor of 0.05 (within a range from 0 to 1), these are effectively orthogonal.

From inspection there appears to be no way to a priori exclude IMF 5 from our analysis, despite it being a superfluous frequency.

When we apply the full Hilbert-Huang transform, we start to be able to see how these IMFs interact together in more detail. Looking at the skeleton lines plot:

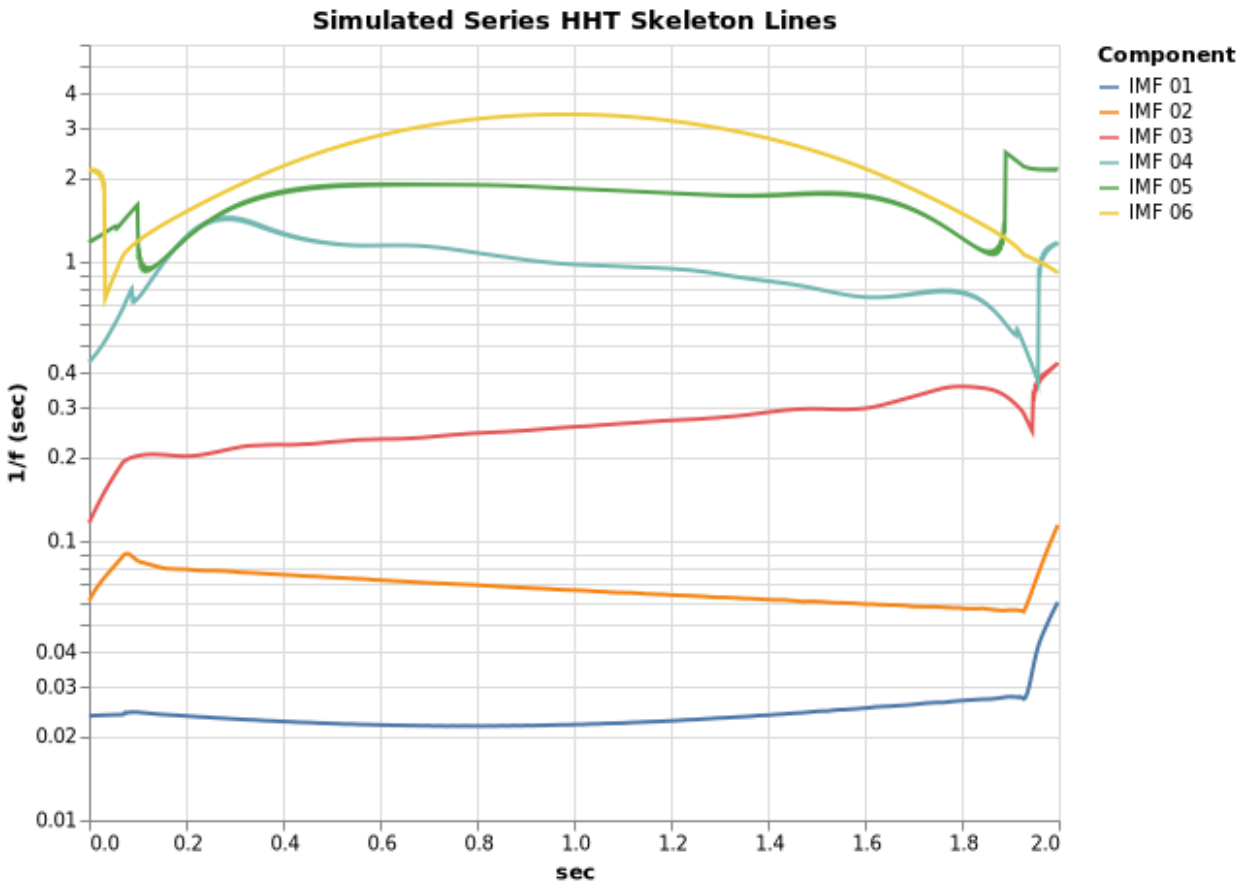

Figure 17: Hilbert-Huang transform applied to simulated series, displayed in skeleton line form

Aside from apparent boundary artifacts, this corresponds well to the skeleton line plot of the original components. If we overlay the two together, we see the similarities:

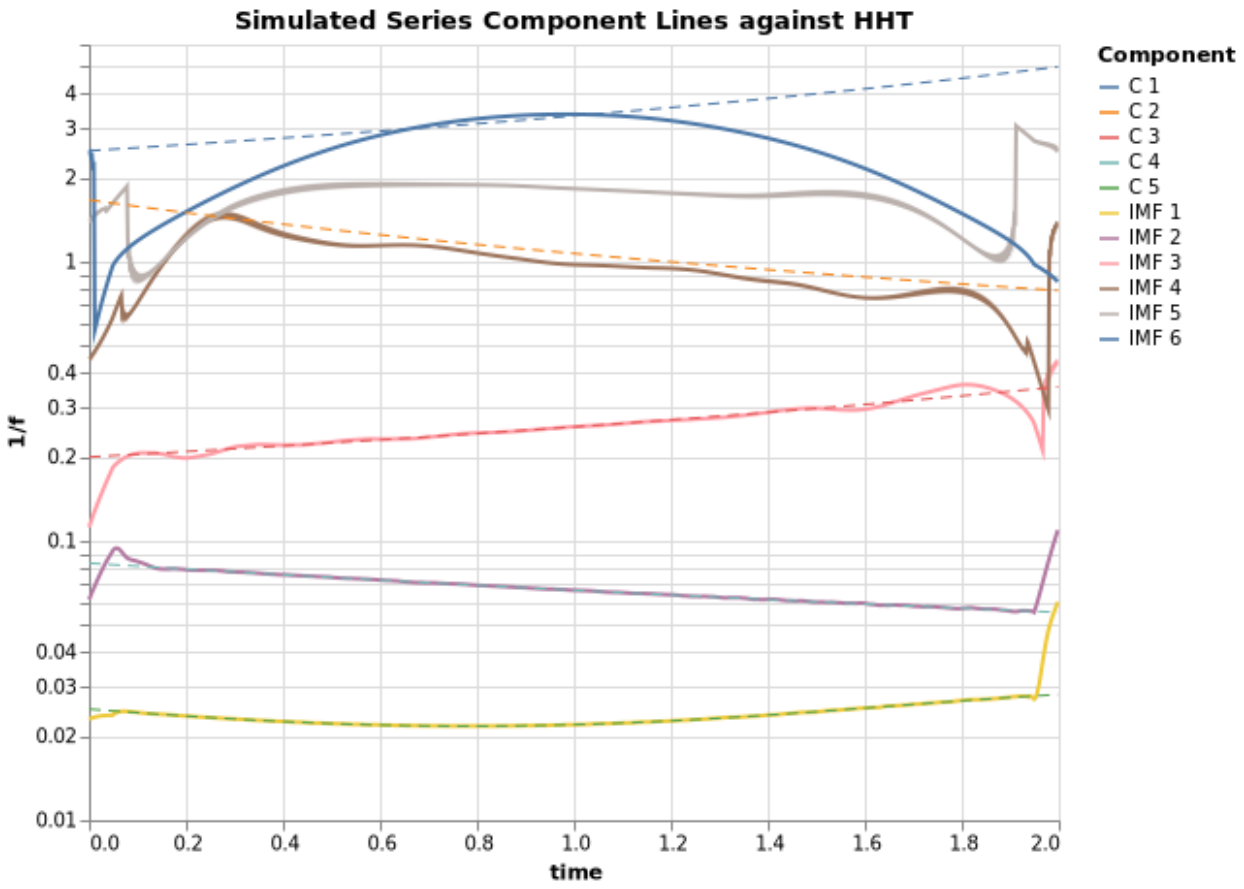

Figure 18: Comparing expected simulated components with those derived from EMD/HHT.

Here we observe directly the alignments of IMFs 1, 2, and 3 with components 4 and 5. We also observe the slightly weaker alignment of IMF 4 and Component 2, and the noticeably weaker alignment of IMF 6 and Component 1. We also see that IMF 5, although having a clearly defined skeleton line, does not correspond to any original component.

To shed light on how these IMFs contribute to the overall system, we may look at the scaled skeleton plot, where each line is scaled according to its magnitude in the HHT:

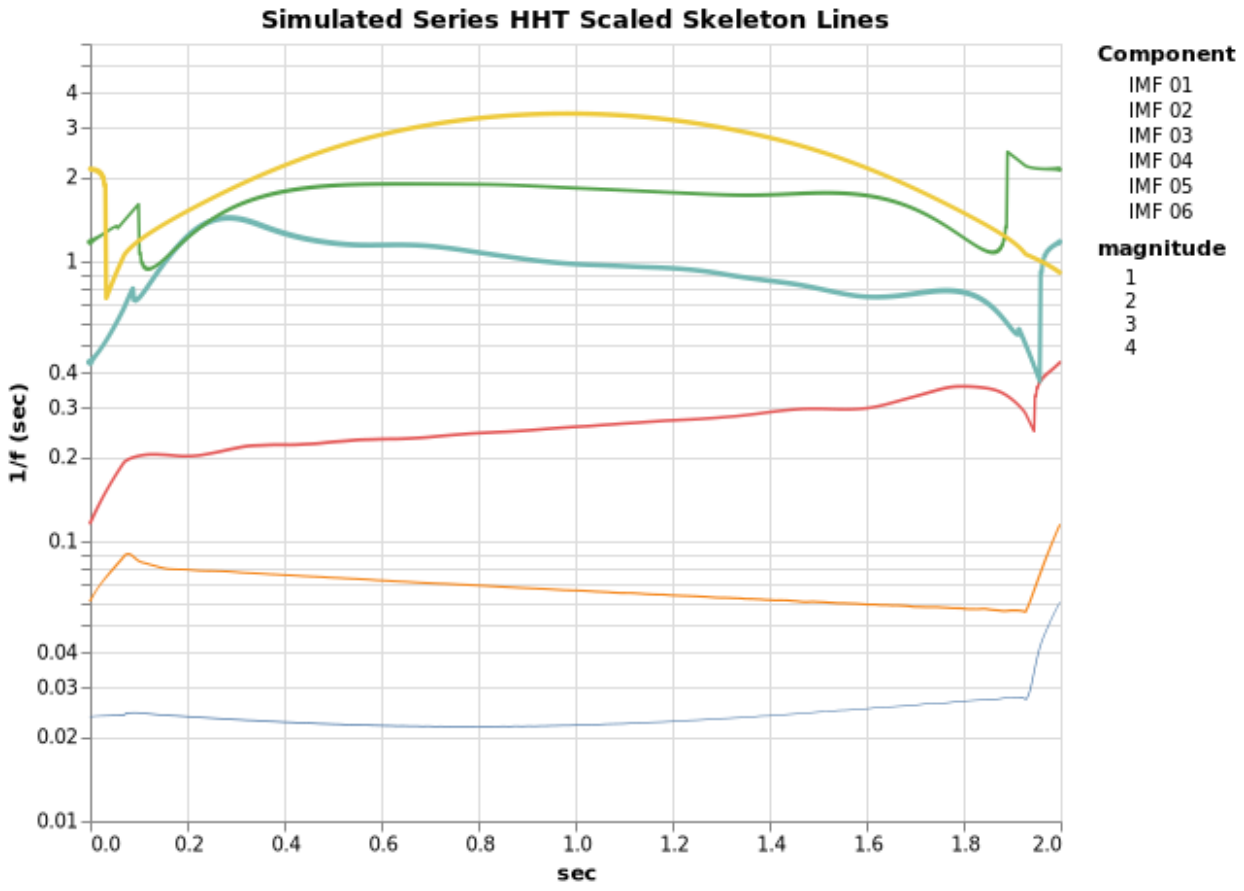

Figure 19: Skeleton lines for the HHT of the simulated series, scaled by magnitude.

The thickness of each line corresponds to their associated component's magnitudes.

IMFs 1, 2, and 3 appear to carve out their own frequency band. However, in this plot, IMFs 4, 5, and 6 give the appearance of fighting over the 1Hz-3Hz frequency band. From this plot, we can directly see that IMF 5 is quantitatively smaller than its surrounding members.

Quantitatively, we can find the average energy (magnitude squared) of each IMF

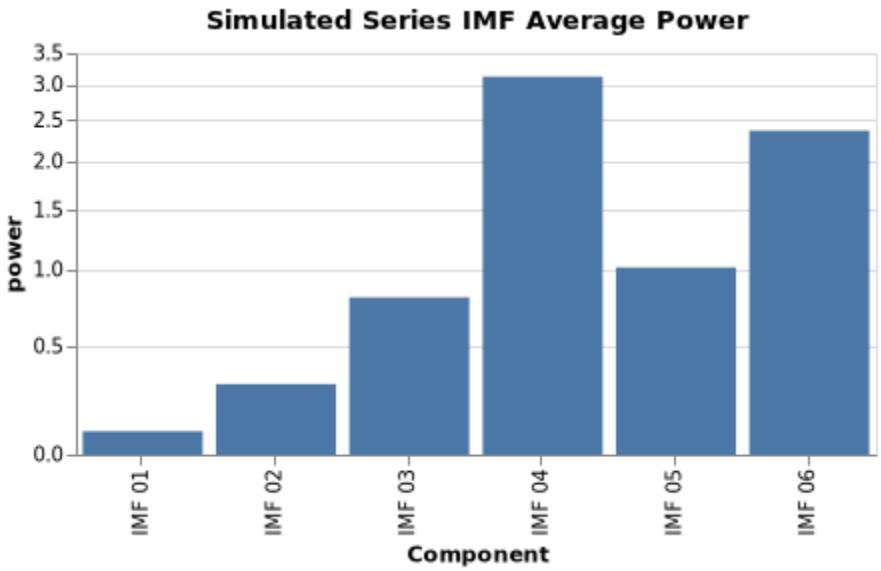

Figure 20: Average power for each IMF of the simulated series.

And we note that IMF 5 is lesser in contribution than its neighbors in its competing frequency bands, IMFs 4 and 6.

We can also examine the relative contribution of each IMF to the total power of the system as a function of time using a stream plot:

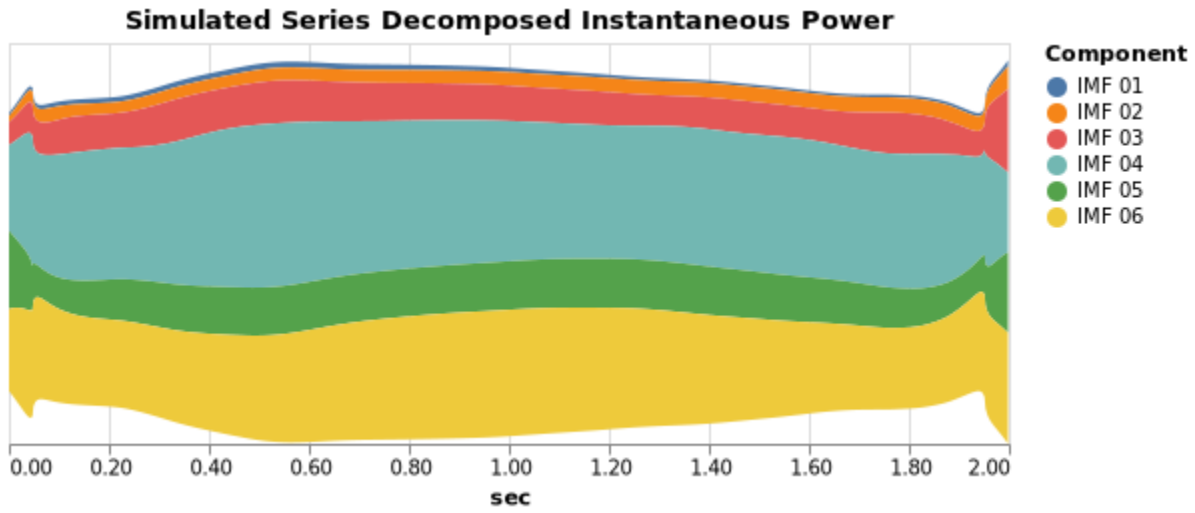

Figure 21: Instantaneous power for simulated series IMFs.

We see here that IMF 5 contributes a relatively small amount of power over the course of the time series, compared to its immediate neighbors 4 and 6. From this we can conclude a priori that it is of not much significance within its own frequency band.

The first few IMFs, IMFs 1, 2, and 3, all contribute equal or less power than IMF 5. However, each dominate in their respective frequency bands. Within this crowded frequency band of 1Hz-4Hz, where IMFs 4, 5, and 6 all vie for attention, IMF 5 is clearly less important. This gives us justification to a priori dismiss IMF 5 as having significance in the underlying system, even without knowing the original components.

We can compute the Dense Hilbert-Huang Transform under a discretization dividing the log of period space into thirty even spaces:

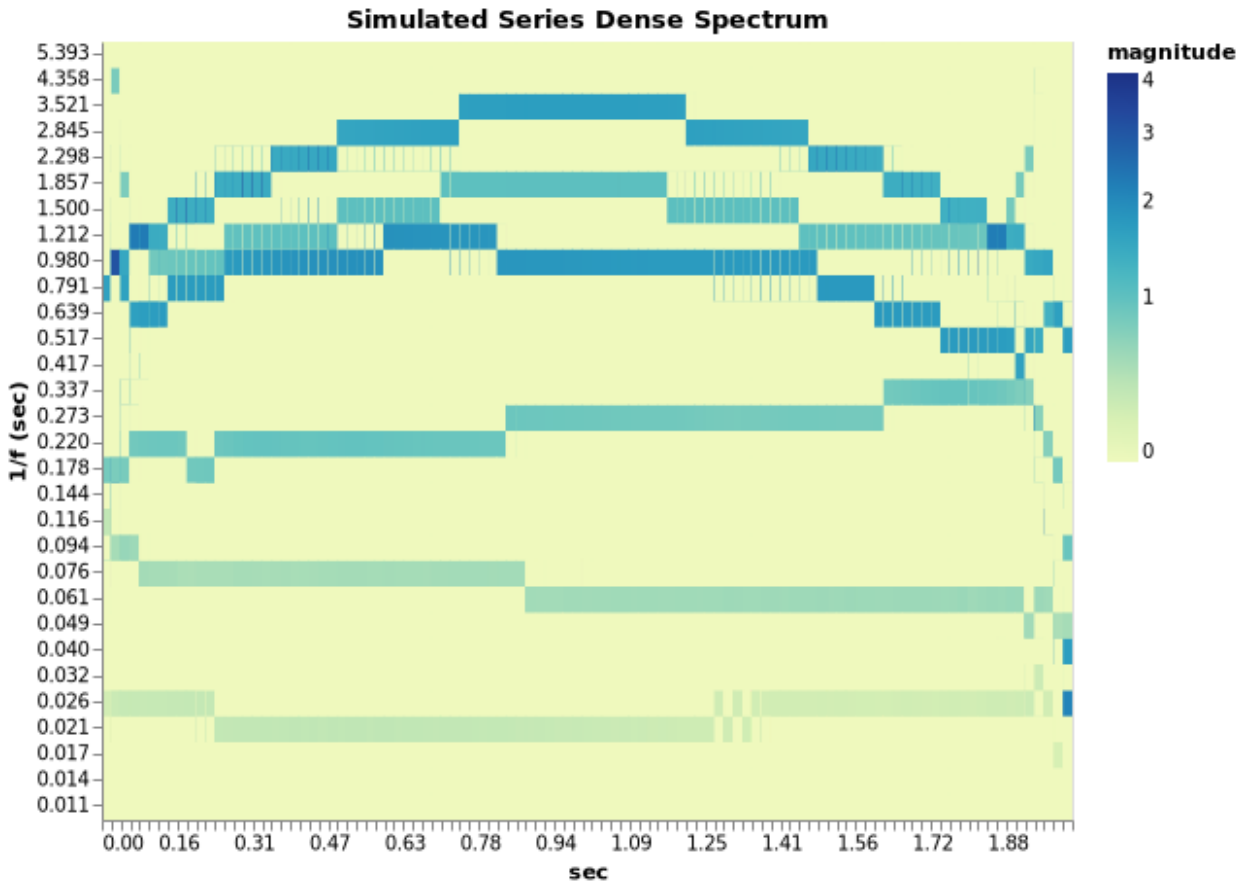

Figure 22: Dense spectrum derived from HHT of simulated series.

This gives a clear picture of a time-localized frequency plot, where we can track a given frequency's contribution to the signal at a given time. In addition, we can trace clearly the contributions of components 2, 3, 4, and 5.

This can be compared, in format, to the Wavelet Decomposition. Compare the above plot to the wavelet decomposition against the Morlet Wavelet with $\sigma = 10$:

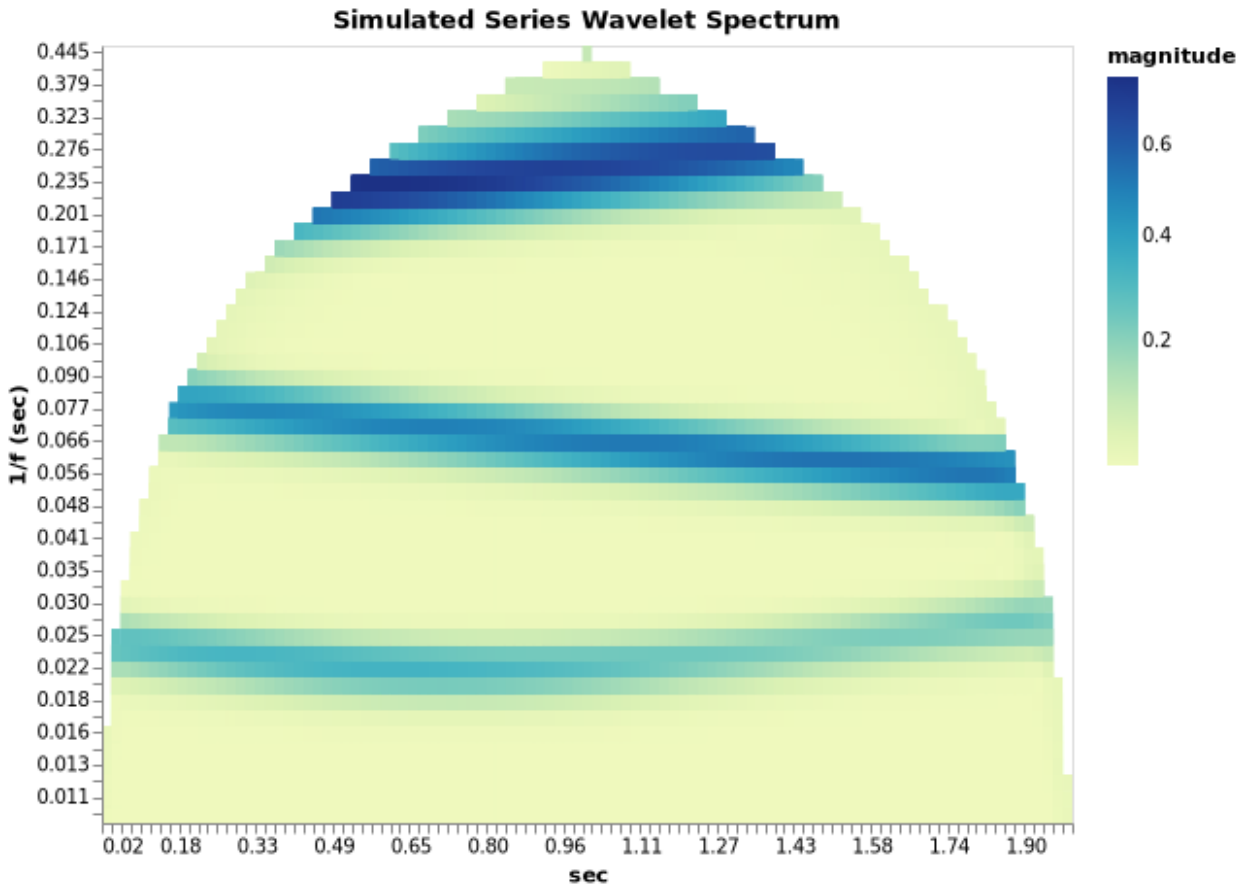

Figure 23: Wavelet decomposition of simulated series, contrasted with the dense spectrum derived from HHT.

Within its cone of influence, the wavelet does identify components 3, 4, and 5. Qualitatively, its identification of these components is less precise: it they are spread out over a wider range of frequencies than their true underlying components are. In this respect, the HHT provides a sharper resolution for frequency-space localization.

Note also that due to the effects of the Cone of Influence, the wavelet decomposition cannot identify any components acting above a period of 0.44s, or a frequency of about 2.3Hz. This means that Components 1 and 2, acting at periods 0.8s and higher (or at frequencies 1.25Hz or lower) are completely out of the reach for a wavelet transform within its Cone of Influence.

We can compute the Instantaneous Power:

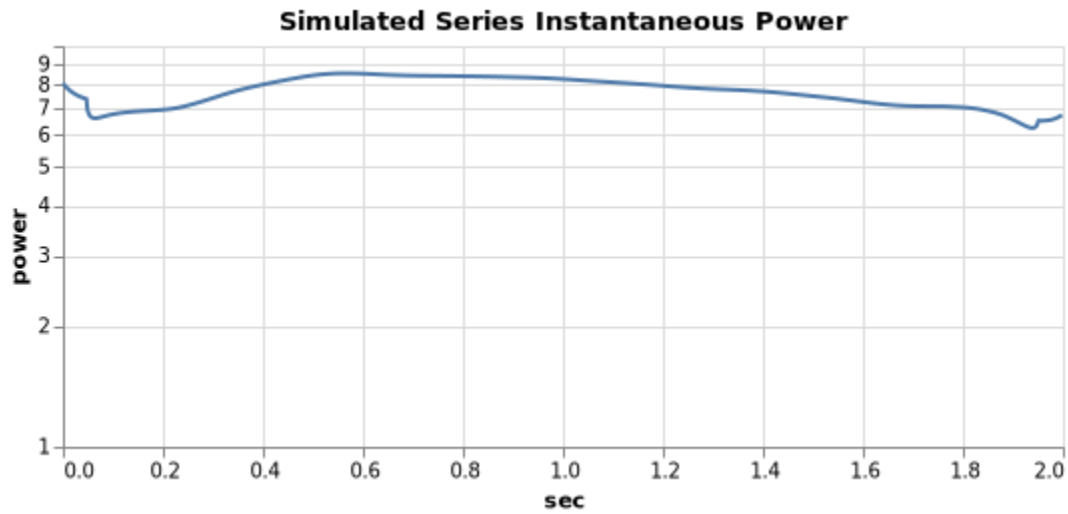

Figure 24: Total instantaneous power for infinite series HHT as a function of time.

Which demonstrates that our power was constant throughout the course of the series, with some gradual fluctuations that correspond to the power fluctuations of the underlying components due to changes in amplitude. We can observe that the power spectrum matches the outline of the stream plot above of relative IMF power over time.

The marginal frequencies provide us a time-averaged power spectrum on frequency

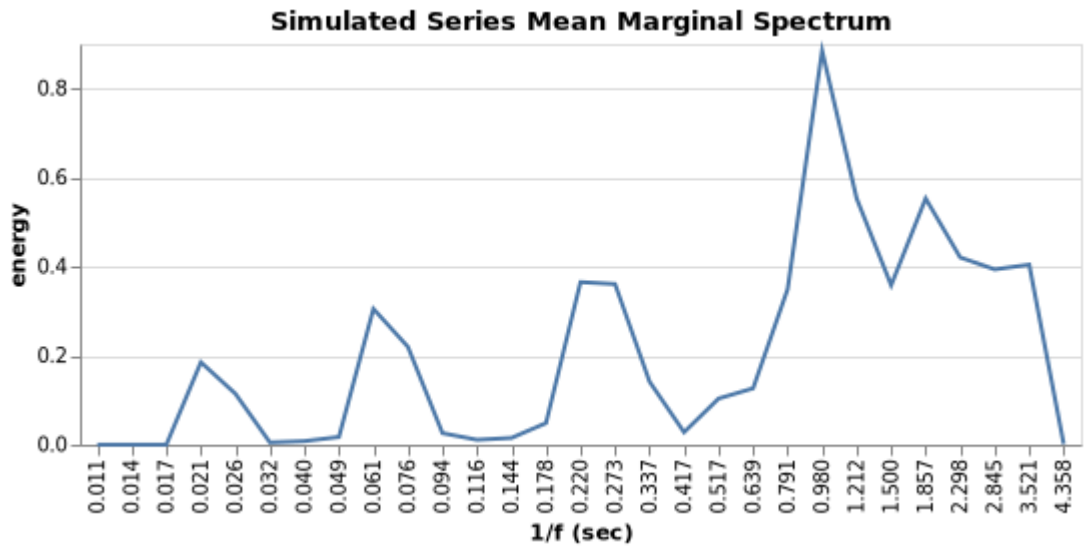

Figure 25: Mean marginal spectrum of simulated series, derived from HHT

Under the marginal spectrum we can clearly identify the frequency bands of Components 3, 4, and 5, as well as a bimodal frequency band associated with Components 1 and 2.

This is directly comparable to the Fourier Decomposition:

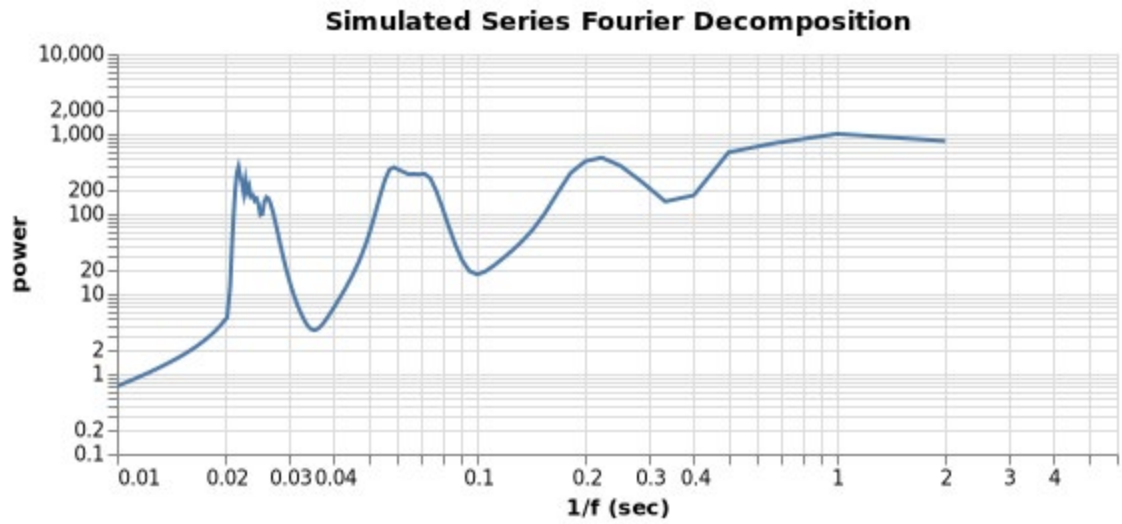

Figure 26: Fourier decomposition of simulated series, in contrast with mean marginal spectrum.

We get a similar display of information with the Fourier Decomposition at high frequencies. However, at low frequencies (and high periods), the resolution of the Fourier Decomposition becomes extremely poor. It has a hard time defining the shape of the frequency bands at frequencies smaller than 1Hz. And, while our Marginal Spectrum from the HHT shows the full shape of the frequency band up to 0.23Hz (period of 4.4s), the Fourier Decomposition cannot resolve any information at all below 0.5Hz (period of 2s). We reach the minimum frequency boundary of the Fourier Decomposition (or maximum period), but the minimum boundary of the HHT extends much lower. Its maximum period extends much higher. Component 1 is more or less completely out of the reach of the Fourier Decomposition, just like how Components 1 and 2 are out of reach for the Wavelet Decomposition.

This comes with a trade-off in accuracy – our resolutions of Components 1 and 2 are significantly less accurate than our resolutions of Components 3, 4, and 5. However, even being able to see

some clear picture of Components 1 and 2 with the HHT is already something that the Wavelet Decomposition and the Fourier Decomposition cannot match.

Computing the Degree of Stationarity:

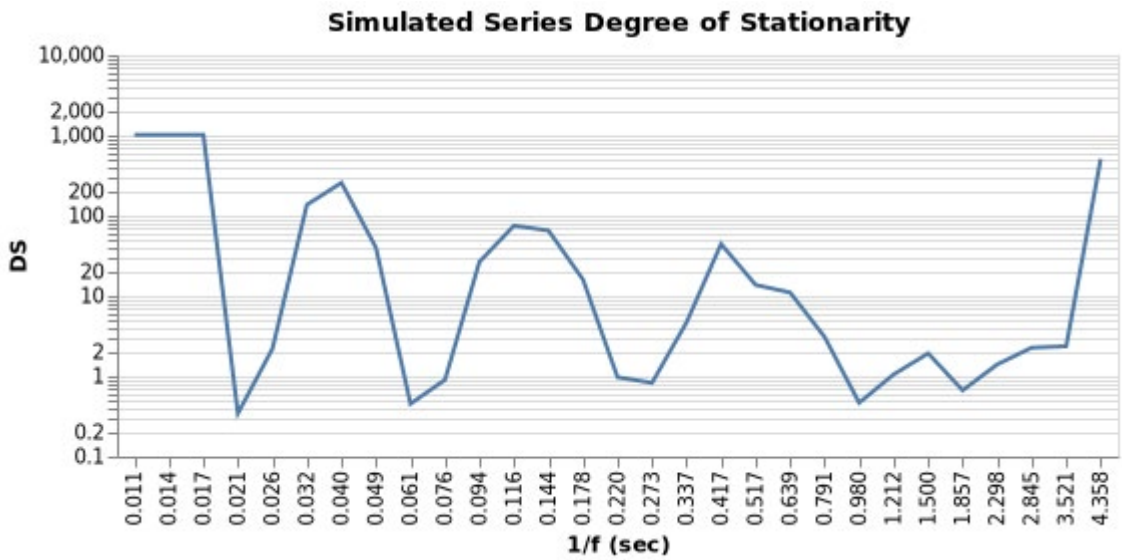

Figure 27: Degree of stationarity for simulated series, derived from HHT.

We can see that the degree of stationarity varies largely, with some edge effects. The low points in DS correspond to highly stationary frequency bands, where a DS of 0 corresponds to a perfectly stationary band. We observe that no band is perfectly stationary, although each of the underlying components carve out small valleys of stationarity around their frequency bands. This plot allows us to analyze the validify of each portion of our Fourier Decomposition: around each component's band, the Fourier Decomposition becomes more valid, albeit never perfectly valid.

6.6 Analysis of Climate Data

Here we apply the same analysis to climate data. First, we will look at a time series with a clear dominating frequency contribution, where the results should match what we expect. Then, we will

look at one where the underlying frequency components are less clear and more ambiguous and compare the conclusions of the HHT against those of the Wavelet and Fourier Transforms.

Sunspot Record

Let us look at monthly sunspot data, which counts the number of sunspots observed on the sun for each month. This time series is widely accepted to be dominated by the Solar Cycle, a periodic behavior of the sun in which it switches polarity of its magnetic field every eleven years.

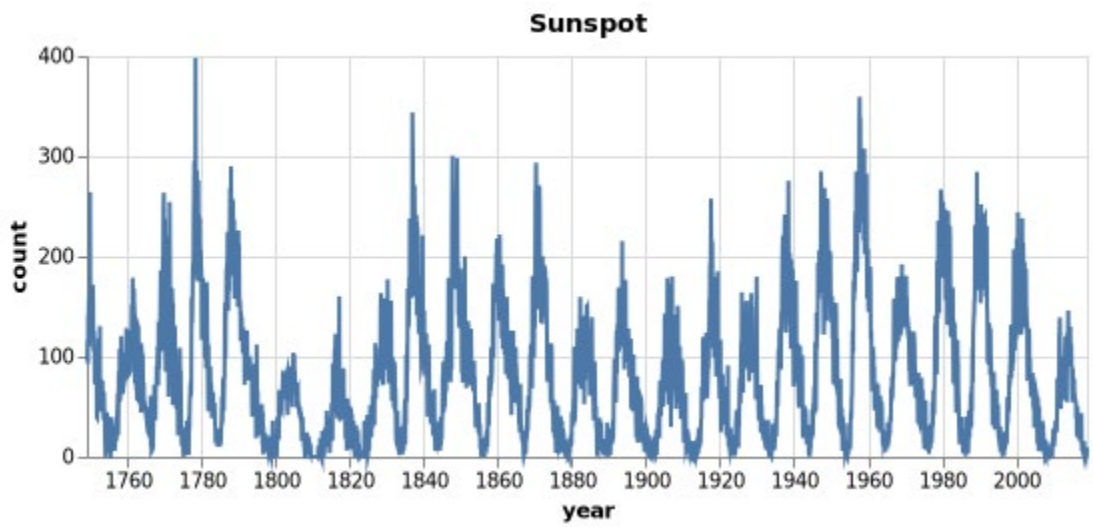

Figure 28: Sunspot time series

This periodicity is clearly visible in the series. Under Empirical Mode decomposition, we see the isolated IMFs:

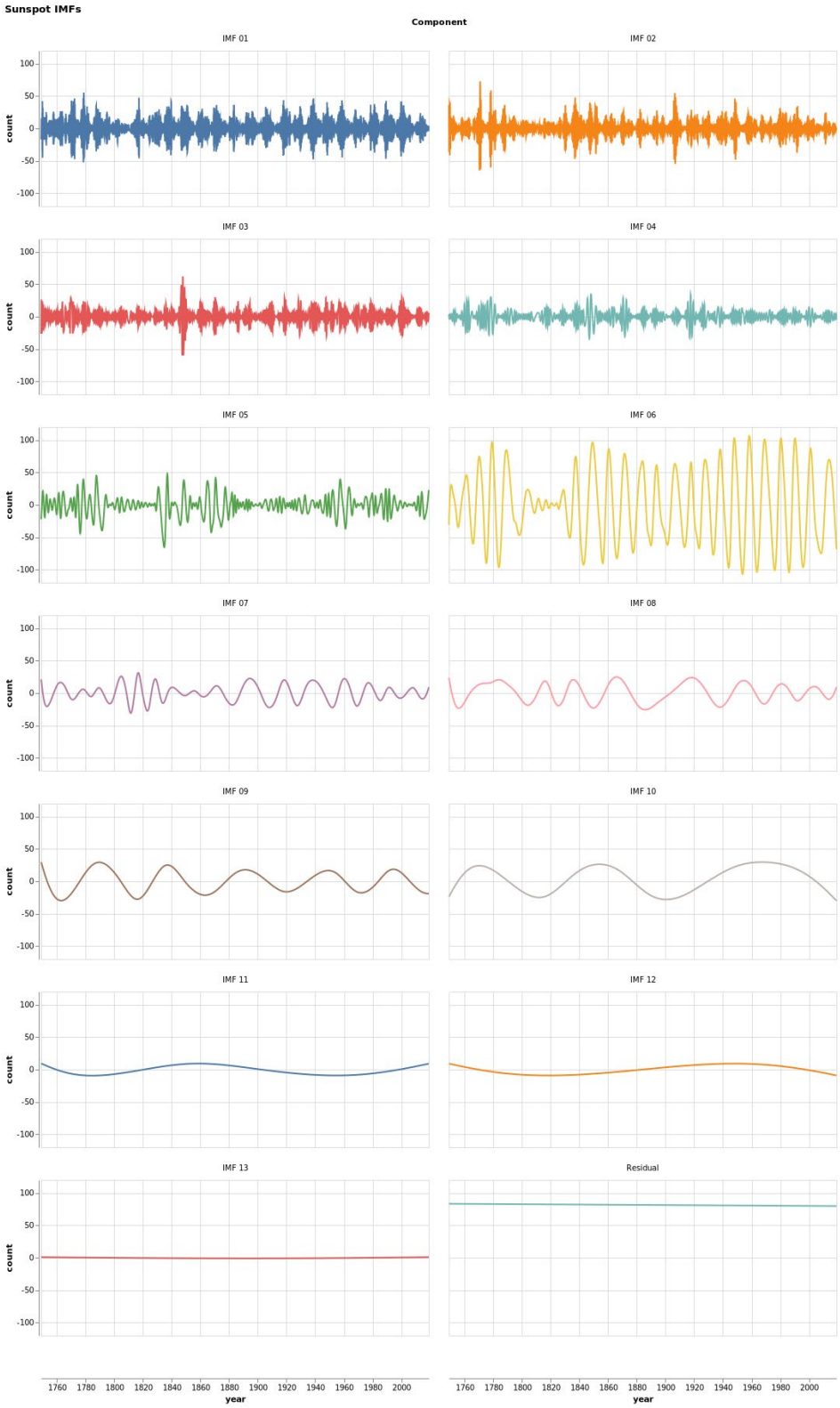

Figure 29: Results of EMD applied to sunspot time series.

The clearly dominant IMF is IMF 6, which has an apparent periodicity of about eleven years. This IMF clearly gives us the trend in Solar Cycle. Even more interestingly, it appears to have a "dip" in intensity around the 1820's, which corresponds to a period known as the Dalton Minimum. The overall effect is that we isolate the Solar Cycle trend, and its intensity as a function of time.

We can compute each IMF's mutual alignments:

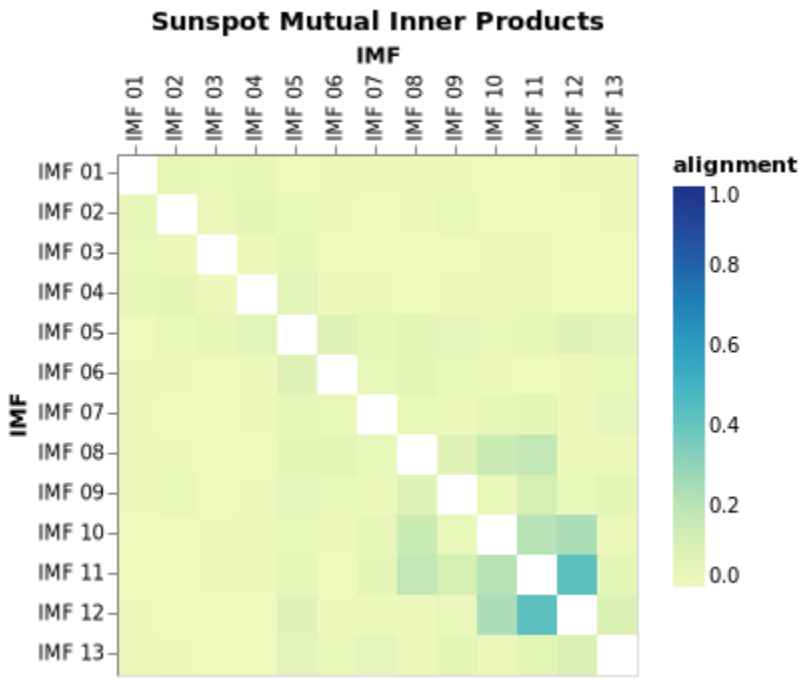

Figure 30: Mutual inner products for sunspot series, showing mutual orthogonality.

We see that all IMFs are orthogonal, except for IMFs 10, 11, and 12, which have some noticeable degree (> 0.3) of alignment. However, most pairings are completely orthogonal with respect to the other.

To investigate what is happening during the Dalton Minimum, we can perform a Hilbert-Huang transform and plot the skeleton lines:

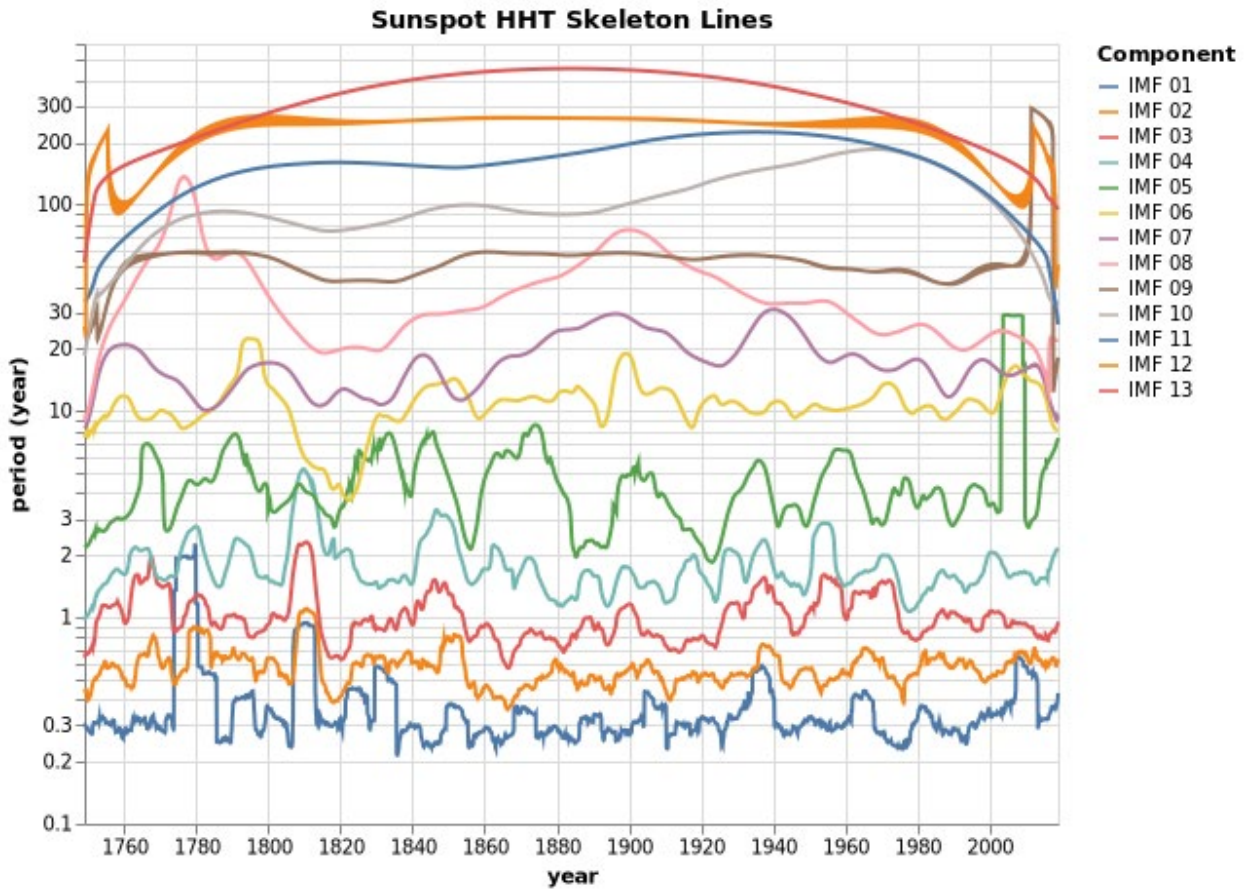

Figure 31: Skeleton lines for sunspot series, derived from HHT

This skeleton line plot is a bit less clean than our simulated time series, showing multiple underlying components each vying for similar frequency bands. We may also observe some interesting artifacts of the sifting process: in some situations, IMFs might swap aspects of the underlying components they are attempting to track. Note the large discontinuity for IMF 1 during the 1770's – it jumps suddenly to period 2yr before resuming its normal behavior at period 4mo. IMF 1 appears to track the progression of a 4mo periodicity within the variation of sunspot count; however, IMF 1 mis-tracks to instead track IMF 4's 2yr periodicity temporarily.

Such jumps appear to be artifacts of tracking. Essentially, if one sees IMF isolation as the attempt to track a single underlying component of shifting frequency over time, some issues may arise as components intersect in frequency.

This skeleton plot is somewhat visually cluttered. To clear out the noise, we may look instead at the scaled HHT skeleton plot, where line thickness corresponds to magnitude:

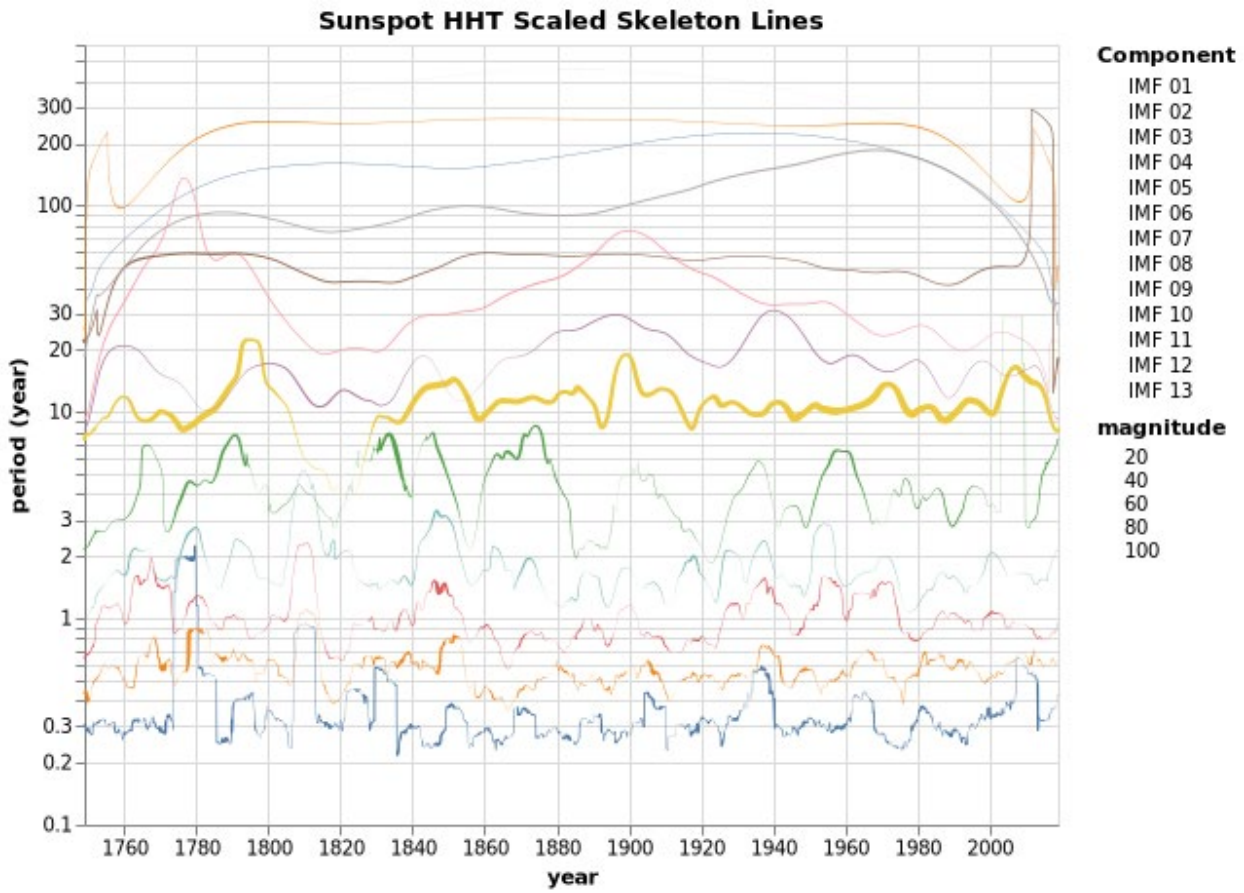

Figure 32: Scaled skeleton lines plot for sunspot time series, derived from HHT.

In this figure, we see the clearly dominant component is IMF 6, which we had previously identified as tracking the 11yr periodicity of the solar cycle.

Looking directly at the evolution of IMF 6 through frequency space, we can identify some interesting variability. Most notably is the fact that it almost disappears in magnitude during the Dalton Minimum. Secondly, we can see that the periodicity is not always exactly at 11yr. At times, it moves up and down, taking on periods generally between 8yr and 15yr. This reflects the fact that solar cycles do not always last exactly 11 years: their duration varies within a range around this central point. Using the HHT, we can track the evolution of this variation between solar cycles as a function of time.

Note that, during the Dalton Minimum, the tracking process has problems with exactly identifying the Solar Cycle: at times, it appears that IMF 7 temporarily "takes over" the tracking of the solar cycle during the Dalton Minimum, until the solar cycle trend becomes strong enough for IMF 6 to resume tracking.

All other components pale in comparison to IMF 6; however, we can notice some components that appear to stand out. For example, IMF 9 appears to carry a strong signal within its frequency band at 50-60yr periodicity. This predicts some meaningful periodicity occurring every 5 solar cycles. This periodicity can be visually observed in the plot of the original time series. IMF 5 also appears to have non-trivial contribution at the 4yr-7yr period range. The HHT also gives us a hint at finding the elusive predicted ultra-low-frequency trends and long-term variability, identifying one at around 250yr in IMF 12.

We can collect the total observed energy for each IMF:

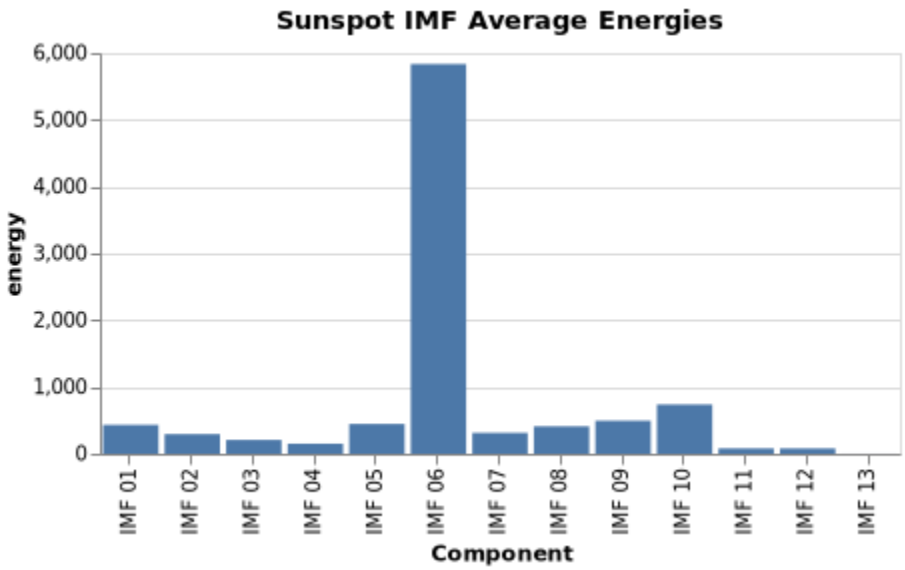

Figure 33: Average energies for each component of the sunspot time series.

The stream plot of power over time gives us a picture of the relative contributions of each IMF over time:

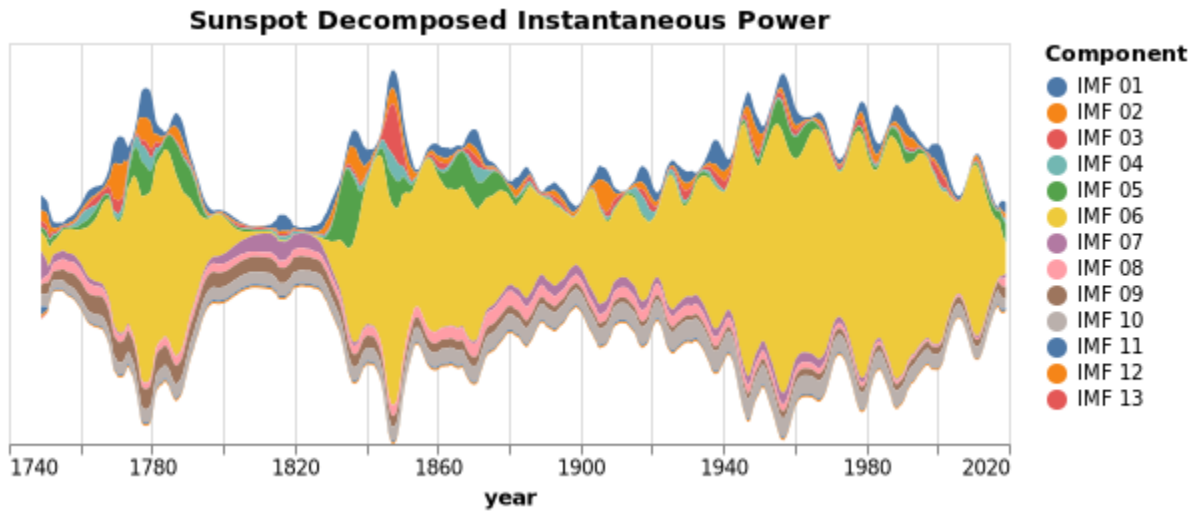

Figure 34: Instantaneous power plot of sunspot series frequencies.

It is clear, again that IMF 6 is the dominating force throughout the entire time series: however, during the Dalton Minimum, the Solar Cycle periodicity loses tower, and the overall spectrum becomes dominated by IMFs 1, 7, 8, 9, and 10.

Outside of the Dalton Minimum, we see that IMFs 8 and 10 remain persistent strong contributions, which we know happen around the 50yr periodicity band.

This decomposition sheds a lot of insight into the underlying dynamics of the system.

Looking at the dense spectrum, we merge the contributions of each IMF together into a map of frequency and time:

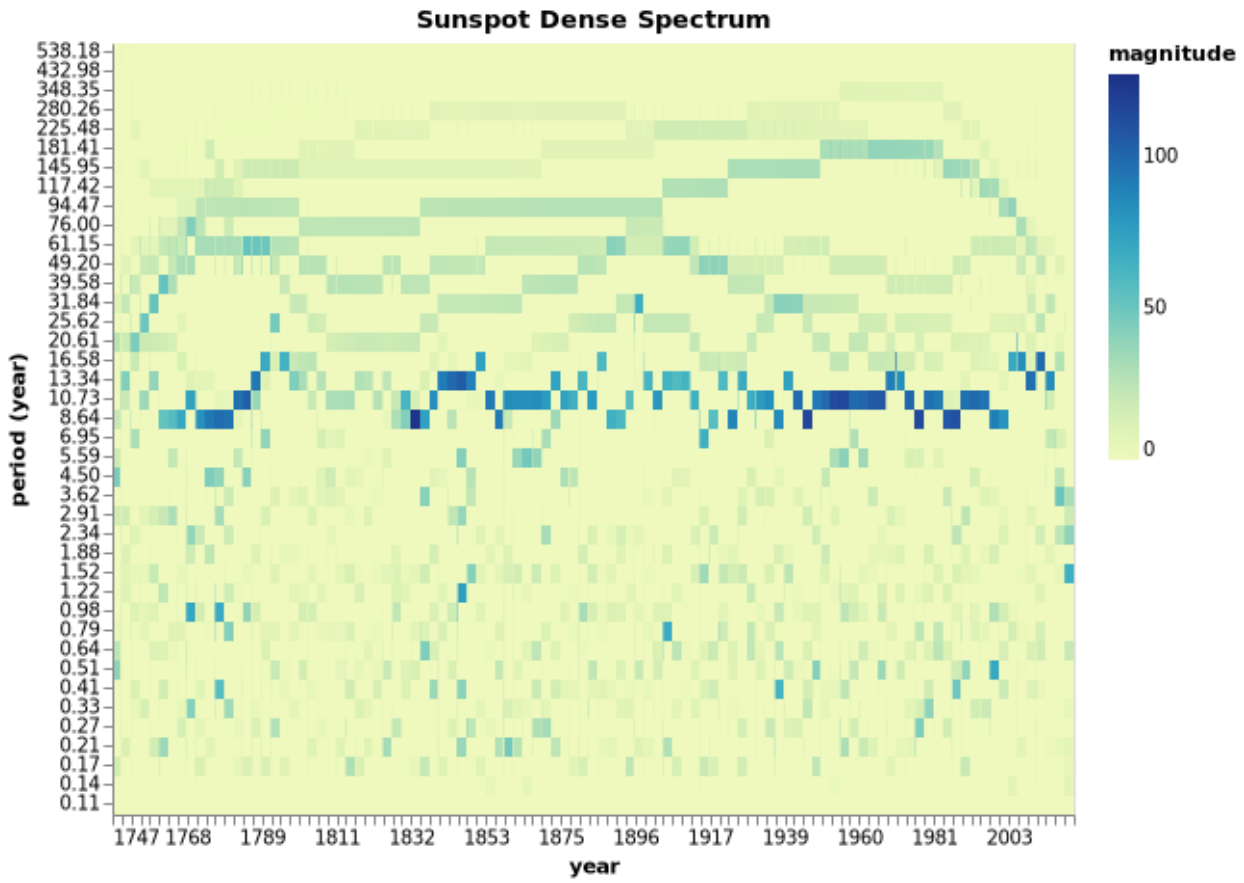

Figure 35: Dense spectrum plot for sunspot series, derived from HHT.

In this dense HHT spectrum plot, we clearly identify the 11yr Solar Cycle trend. This dense spectrum plot also shows us clearly the continuity of the Solar Cycle Trend during the Dalton Minimum: it appears continuous (but growing smaller in magnitude) throughout the period, even though we have seen that this continuity is the result of IMF 7 taking over the job temporarily. This paints a unified picture of how the two work together.

In addition, we also can notice more clearly IMFs 10 and 11 coming together to form a strong ~150yr-periodic cycle throughout the 1900's. This indicates a strong signal of 150yr periodicity throughout the 1900's that is visible from the alignment IMFs 10 and 11. This signal is often identified as the Gleissberg cycle.

Compare the dense plot to the Wavelet Decomposition of the same time series:

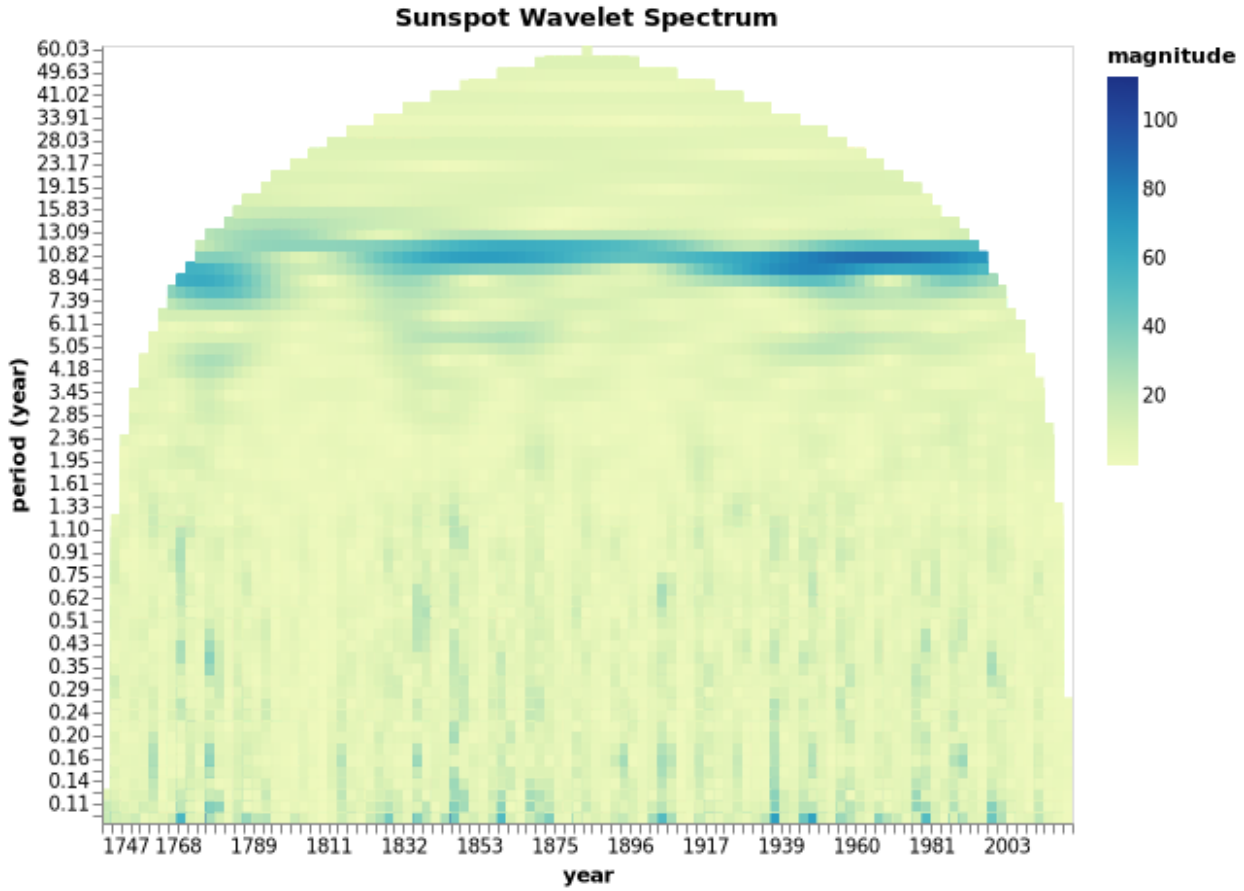

Figure 36: Wavelet decomposition for sunspot time series, to contrast with dense spectrum plot.

As expected, the wavelet decomposition identifies the same 11yr dominant trend, as well as apparent localized peaks in the sub-year periods. However, the Cone of Influence of the Wavelet Decomposition does not allow for any analysis at periods higher than 60 years, rendering the longer-term 100yr and 200yr trends out of reach. The Gleissberg cycle is then not visible in the wavelet decomposition. In addition, the 11yr Solar Cycle is not as clearly defined in frequency space, rather being spread out over a large bandwidth. Because of this, the temporal variability of periodicity is not as clear.

Flattening down time-locality, we can construct the marginal spectrum:

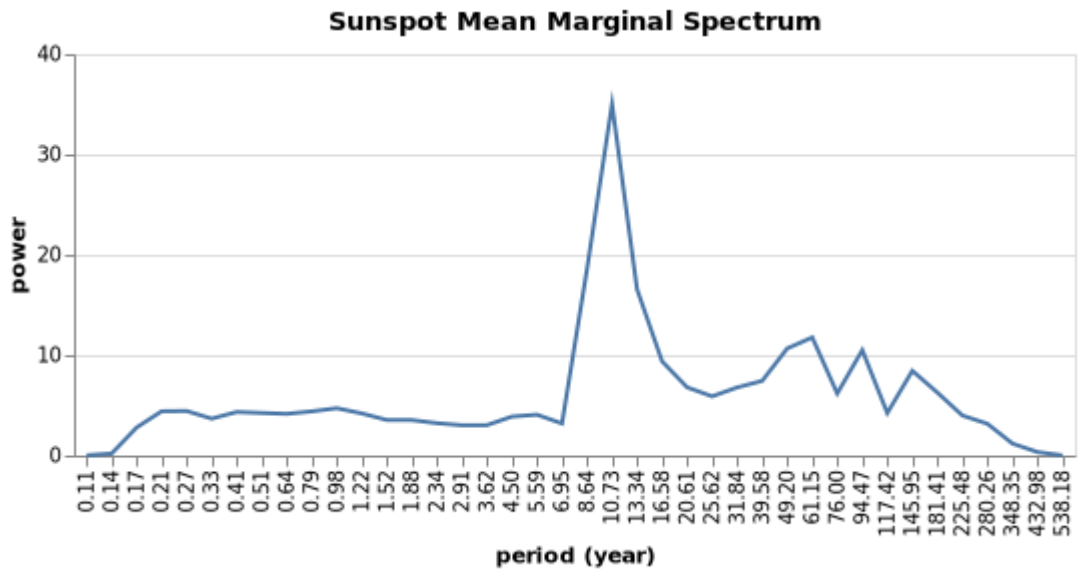

Figure 37: Mean marginal spectrum for sunspot series

In the marginal spectrum, we see clearly again the peak in periodicity at 11yr, but also the trends noticed earlier at 50yr, 100yr, and 150yr (including the Gleissberg Cycle). This indicates a strong 11yr periodicity, with some higher-period effects that can be observed.

This is a clearer picture than the associated Fourier Decomposition:

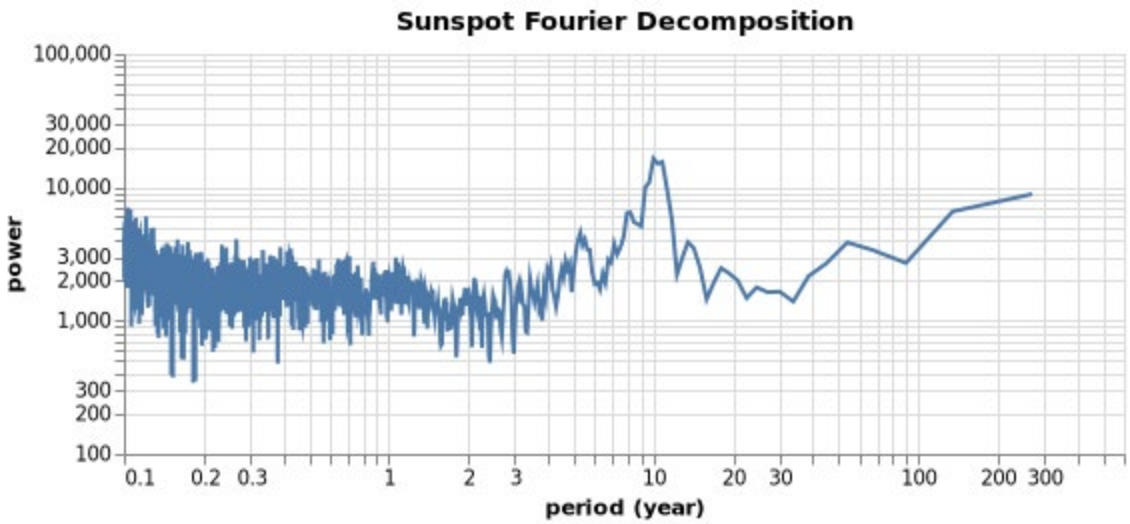

Figure 38: Fourier decomposition for sunspot series, to contrast with mean marginal spectrum

Again, we see many of the same peaks: One at 11yr, and one at 50yr. However, higher-frequency variations are not as clearly resolved, and low-frequency variations occur at such low resolution that it is impossible to meaningfully determine the location of high-period frequency peaks; the Gleissberg cycle is not visible.

Finally, we may look at the Degree of Stationarity:

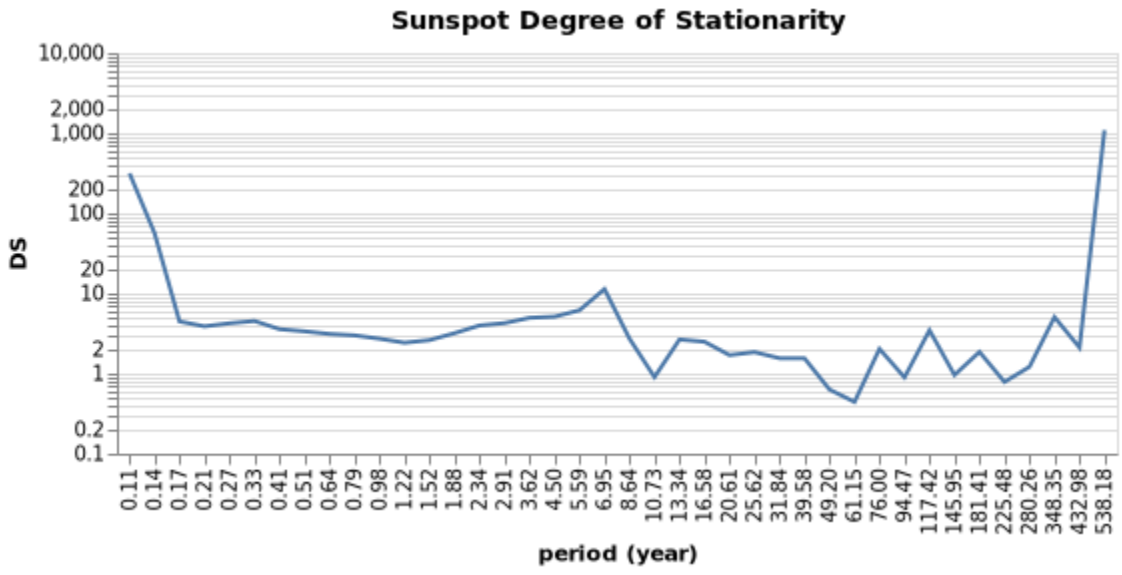

Figure 39: Degree of stationarity for sunspot series.

We see that our series is in general very non-stationarity. However, we do have valleys of stationarity at the 11yr and 60yr periodicities. This can be taken to indicate that the 11yr Solar Cycle is stationary throughout the duration of our time series and is a factor of high stability – something we can visually observe and confirm.

Atlantic Multidecadal Oscillation

Now, let us look at a data set that has less clearly established periodic trends, where energy is shared evenly between many competing periodic cycles. The Atlantic Multidecadal Oscillation (AMO) is derived from sea surface temperature variations in the North Atlantic Ocean.

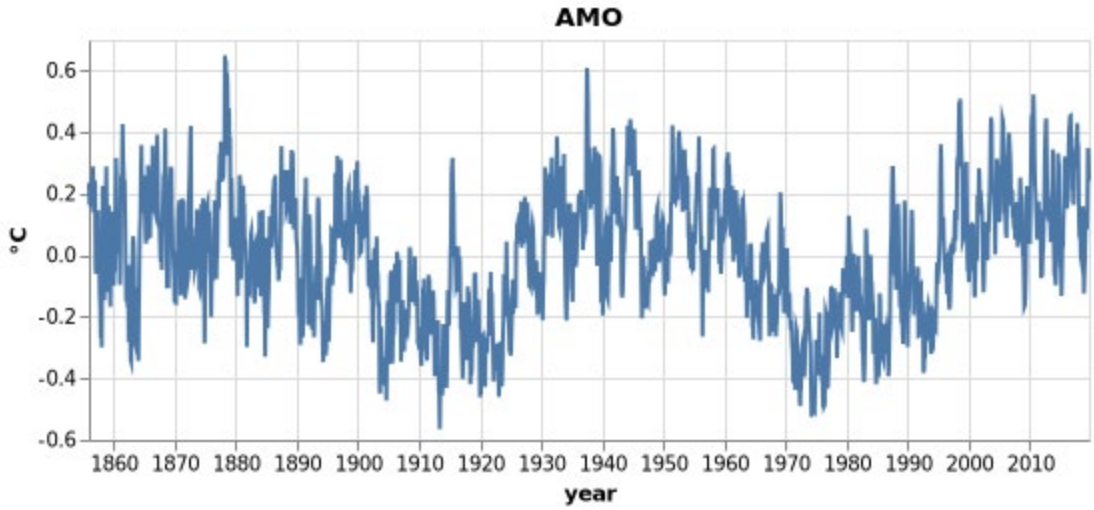

Figure 40: Atlantic multidecadal oscillation (AMO) time series

From visual inspection, we see that there is some recent dominant periodicity with a cycle length of about 70 years ("Multidecadal" oscillation), but it is clearly not as dominant as what was observed in the Sunspot record. In addition, this periodicity seems to not be as apparent in the 1800's. We can visually observe other elements of periodicity: during the mid-1900's, a 5-year periodicity is clearly visible. In the 1800's, a 10-year periodicity seems to dominate

This non-stationarity is a clear sign of the utility of EMD and HHT. For a Fourier Transform, we have no indication of the time dependency of the strength of these signals. While the Wavelet transform may plausibly give us the shifts between 5-year and 10-year periodic trends over the time series, the dominant 70-year cycle is out of the cone of influence.

Let us begin with an EMD to identify the IMFs:

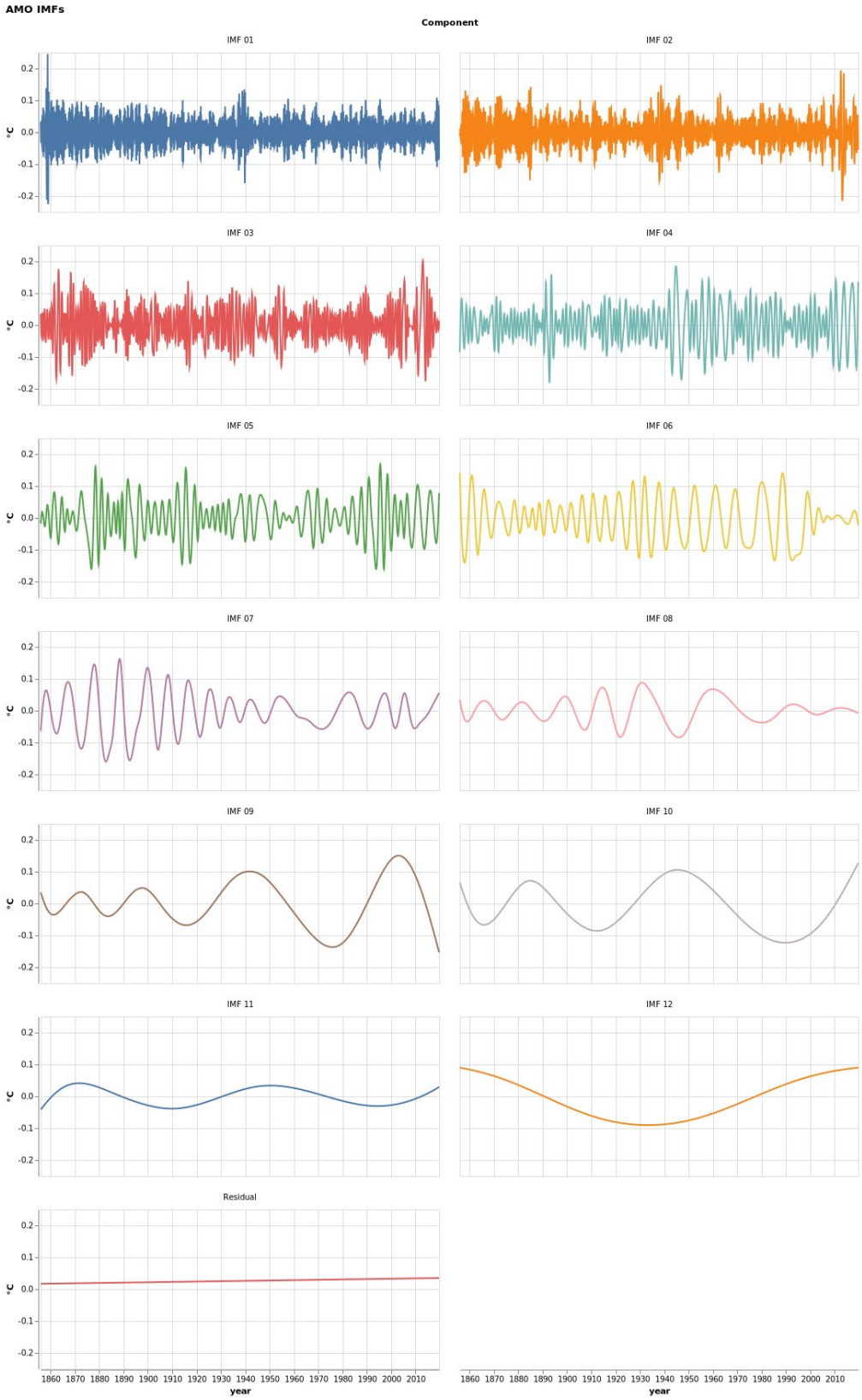

Figure 41: Empirical Mode Decomposition of AMO time series

Immediately, we can identify which IMFs correspond to our visual analysis earlier:

- IMF 6 represents the 5yr cycles observed to be dominant in the mid-1900's: its strength surges during this time period, and later loses clarity outside of it.
- IMF 7 represents the 10yr cycles observed to dominate the trend of the 1800's. It peaks in strength at the turn of the century and dies out by the 1930's
- IMFs 9, 10 and 11, together, appear to represent the 70yr multidecadal oscillation visible in the latter half of the time series.

In addition, we see the powerful IMF 12, which suggests a periodicity over which the entire time series completes one cycle. IMF 12 was not immediately visible, but its isolation through EMD makes it clear. With the large IMF 12 cycle, we may see the whole time series as the completion of one period of the cycle, where the 1920's-1970's is an anomalous deviation.

The EMD is a powerful tool in its ability to immediately isolate each of the visual cycles we observed from the time series.

In terms of mutual orthogonality, the IMFs appear to be satisfactory.

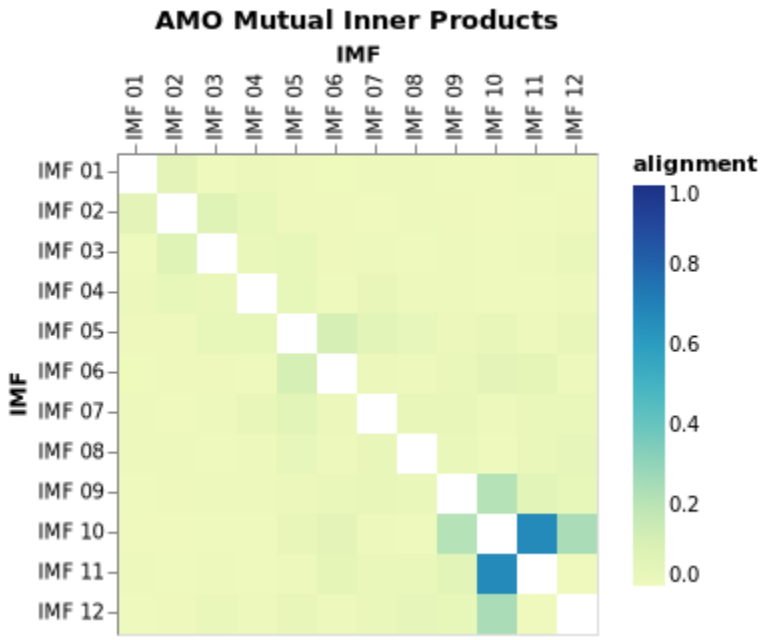

Figure 42: Mutual inner products of each IMF of AMO time series, to show mutual orthogonality.

Only IMFs 10 and 11 together appear to have some alignment (factor 0.5). Investigating further, we can see the relative energy contributions of each IMF:

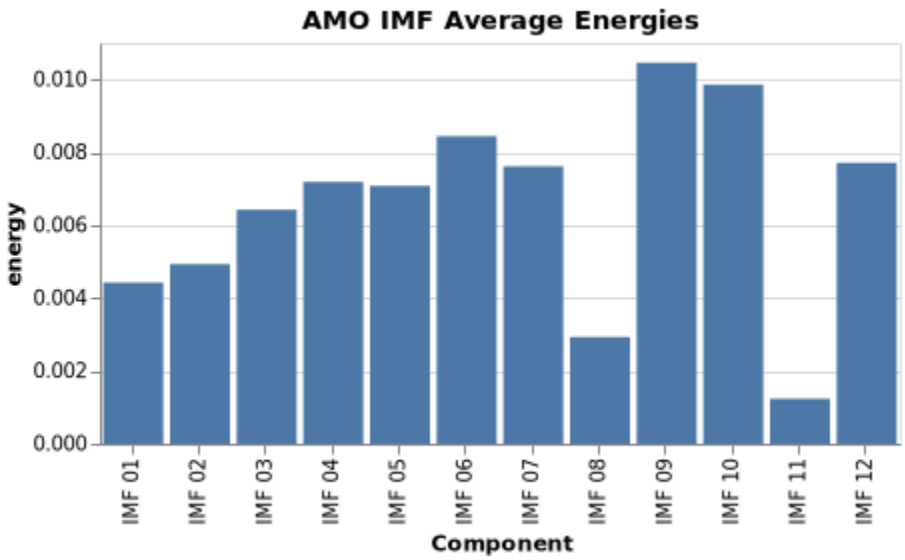

Figure 43: Average energies for each IMF of the AMO time series

From this, we can conclude that IMF 11, being very highly aligned with IMF 10 and having very little total energy, is most likely an artifact of the decomposition process, and has little physical meaning. Its effects are duplicated in the effects of IMF 10. This parallels our situation with IMF 5 with our simulated series.

A look at the skeleton plot confirms our association of IMFs with periodicities:

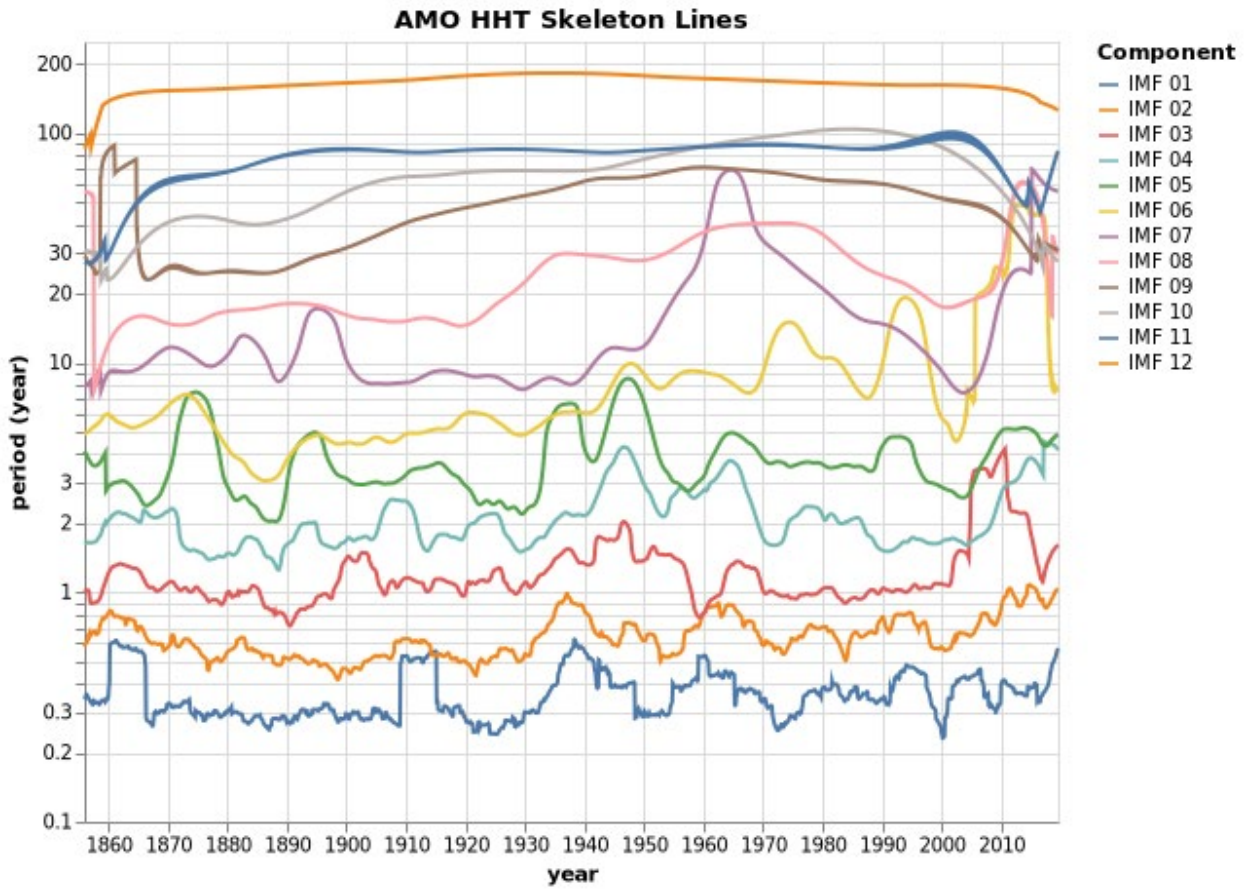

Figure 44: Skeleton lines plot for AMO time series, derived from HHT.

Again, we observe some similar phenomenon of "tracking error", this time mostly with IMF 1 sometimes taking on the behavior tracked by IMF 2. In addition, we see IMF 3 sometimes apparently tracking the behavior of IMF 5. Other than that, there appears to be no other sharp discontinuities arising from tracking errors.

Our previous analysis is confirmed: IMF 7 hovers around the 10yr periodicity until the 1940's, then becomes unstable. IMF 6 clearly tracks a strong five-year cycle up until the 1970's, after which it appears to take over the role of IMF 7. IMFs 9, 10, and 11 show an interesting journey: they begin apart, but slowly converge in 1960 to together all represent the Multidecadal Oscillation

in the latter half of the time series. IMF 12 also clearly corresponds to the 150-year periodicity we identified.

Of note also is IMF 3, which tracks a yearly oscillation. IMF 3 tracks the seasonal 1-year cycle for most of its course.

This analysis is further strengthened by the scaled skeleton lines plot:

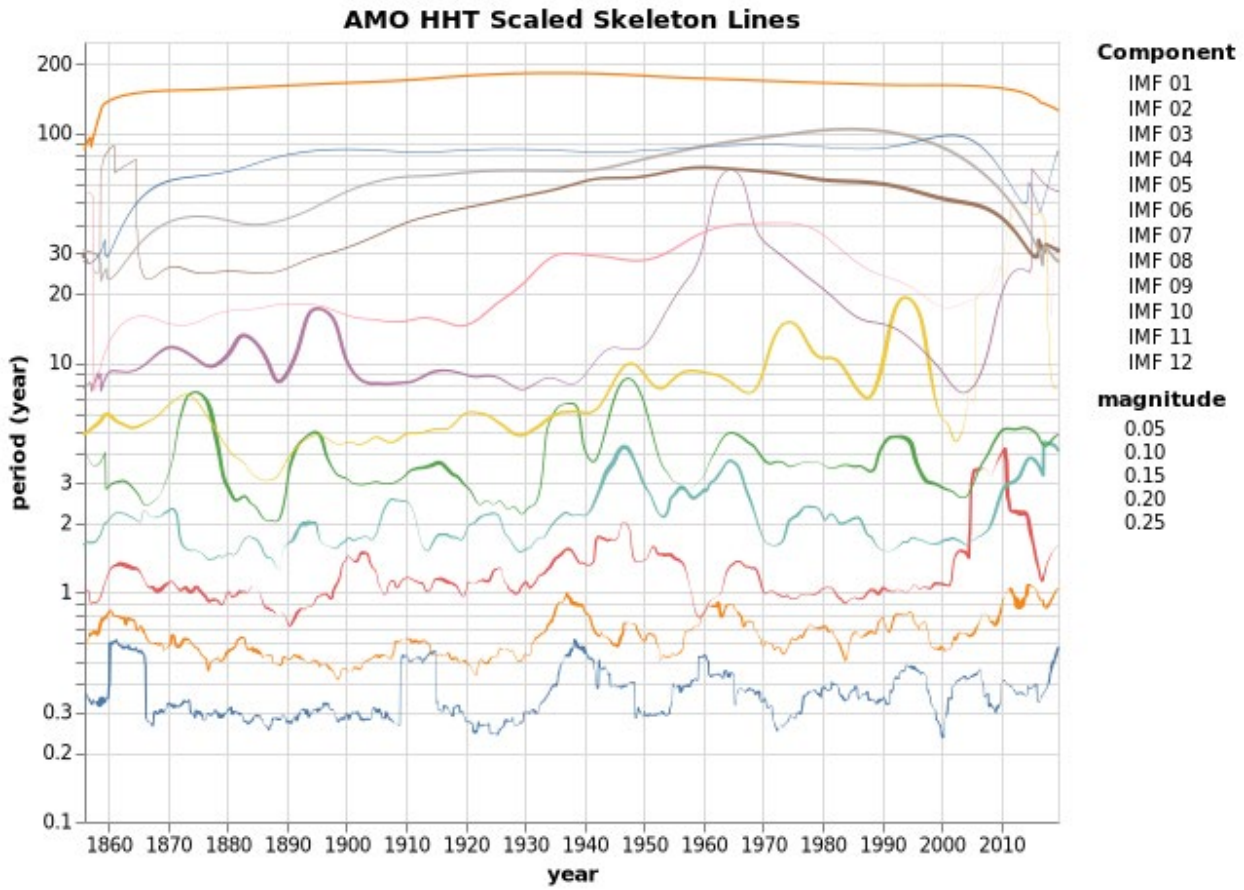

Figure 45: Scaled skeleton line plot for AMO time series, derived from HHT

Here, it is apparent that no single IMF ever dominates the entire time series. Instead, IMFs trade dominance. In the early part of the time series, IMF 7's 10yr cycle (and to an extent, IMF 5) dominates. In the middle part of the time series, IMF 6's 5yr cycle dominates. Then towards the

end of the time series, IMF 9 and 10 together dominate with their Multidecadal Oscillation. And, under the whole time series, IMF 12 remains strong and stable. In this view, the role of IMF 6 "displacing" IMF 7 in the 1930's is clear.

We can also see that while the 1yr seasonal cycle exists, it is never strongly dominating.

If we collapse the frequency story, we can see the stream plot:

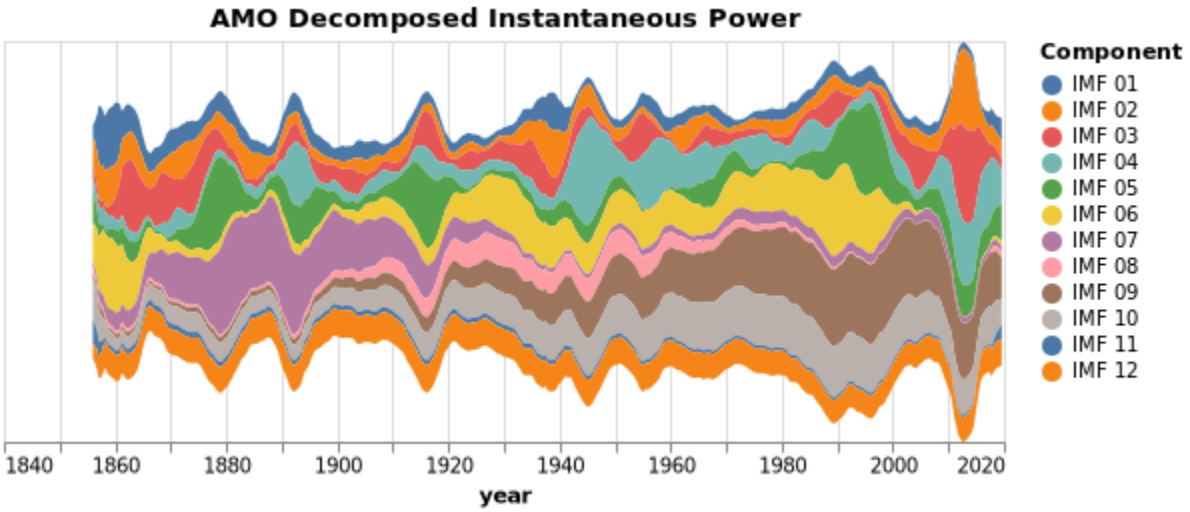

Figure 46: Instantaneous power for each IMF of AMO time series

This clearly represents the trade-off in power between each IMF. As anticipated, IMF 7 dominates in the first half of the time series, before trading off its share to IMF 6. IMF 6 is briefly the dominating factor around the 1940's and 1950's, before IMFs 9 and 10 together become the largely dominating contribution.

This plot also shows the stability of IMF 12 as an underlying force driving the larger-scale and longer-period oscillation. IMF 3, the seasonal cycle, also is constant, but much smaller.

In order to properly assess the dominance of IMF 9 and 10 together, we may look at the dense spectrum:

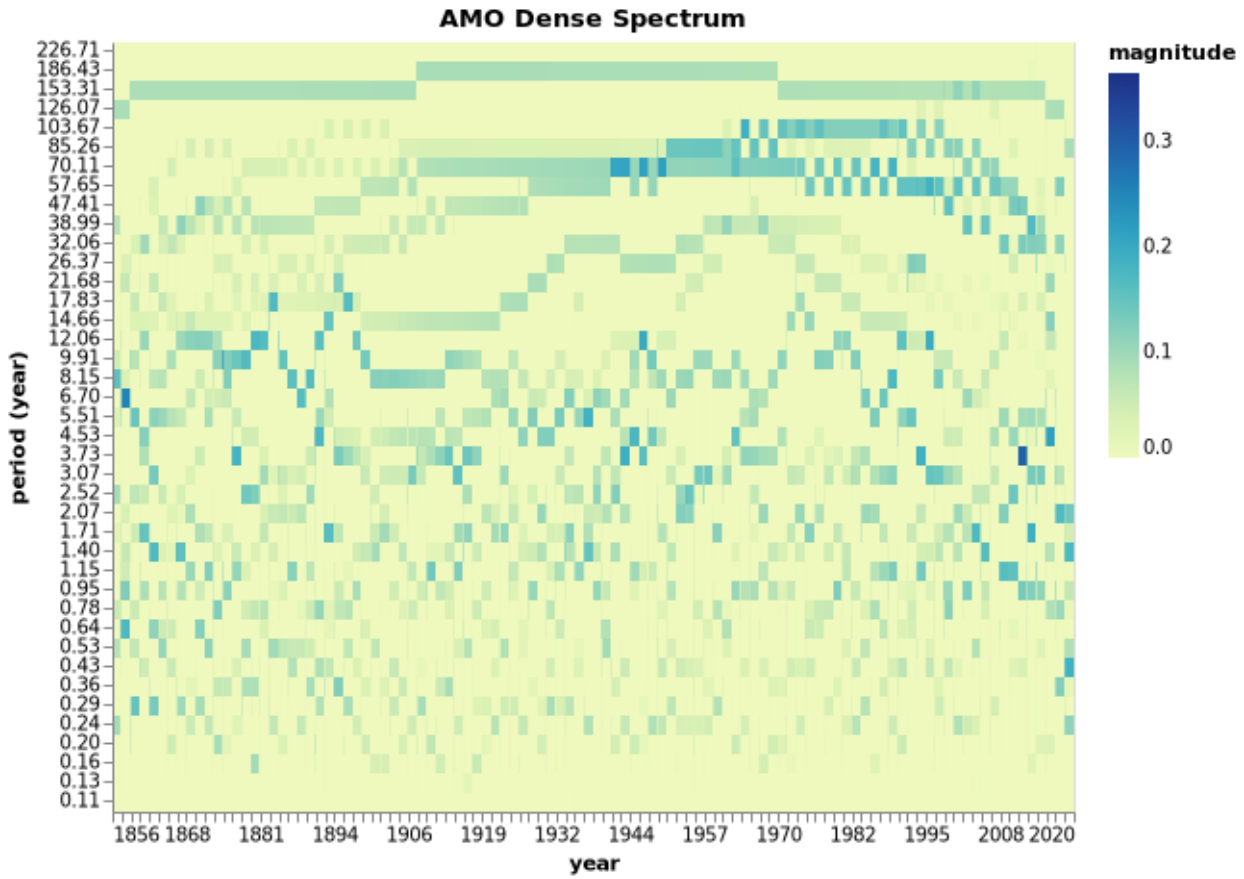

Figure 47: Dense spectrum plot for AMO time series, derived from HHT

Again, the trends noticed before can be seen in the patches around 10yr in the 1800's and 5yr in the mid-1900's. However, in this visualization, the convergence of IMFs 9 and 10 forms a very intense patch in the 70yr-80yr periodicity in the latter third of the time series. Together, they create an island of intense periodicity during their alignment.

The wavelet transform shows some similar conclusions, in a limited way:

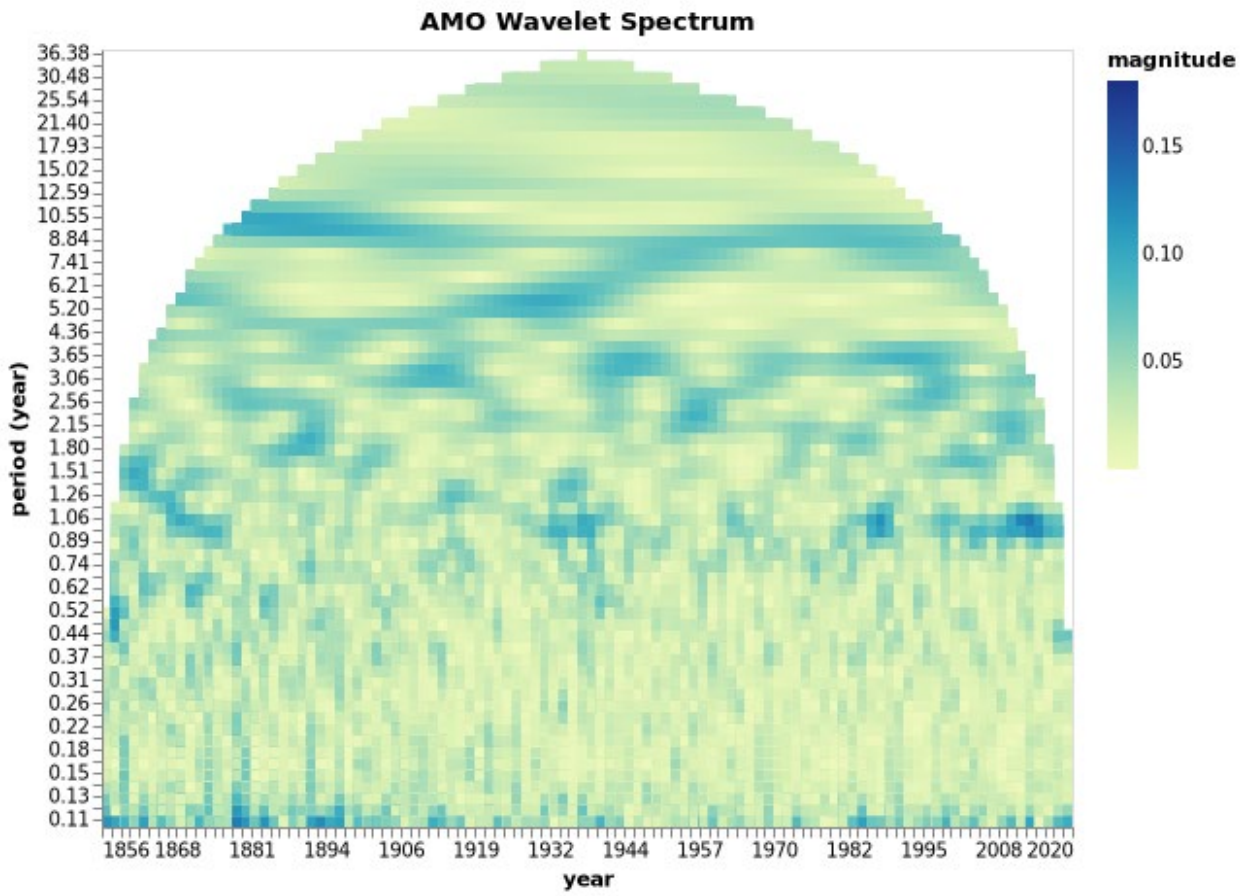

Figure 48: Wavelet decomposition for AMO time series, in contrast with its dense spectrum plot

The wavelet transform properly identifies the 10yr periodicity in the early 1800's, as well as a shift to the 5-year periodicity in the 1920's. A similar picture of the 5-year periodicity "displacing" the 10-year periodicity is also apparent. However, the wavelet transform is unable to resolve the multidecadal component in the latter third of the time series, and the 150yr cycle associated with IMF 12.

It can be noted that peaks in the Dense HHT spectrum appear as fuzzy islands in the wavelet transform, suggesting that they both are identifying the same (temporally) hyper-local phenomenon. One difference, qualitatively, is that with the HHT, we have a story stringing these

local phenomena together: we identify when one evolves into the next. For the wavelet transform, they can be nothing more than isolated islands.

Of course, computing the marginal spectrum destroys the nuance in the time dependence:

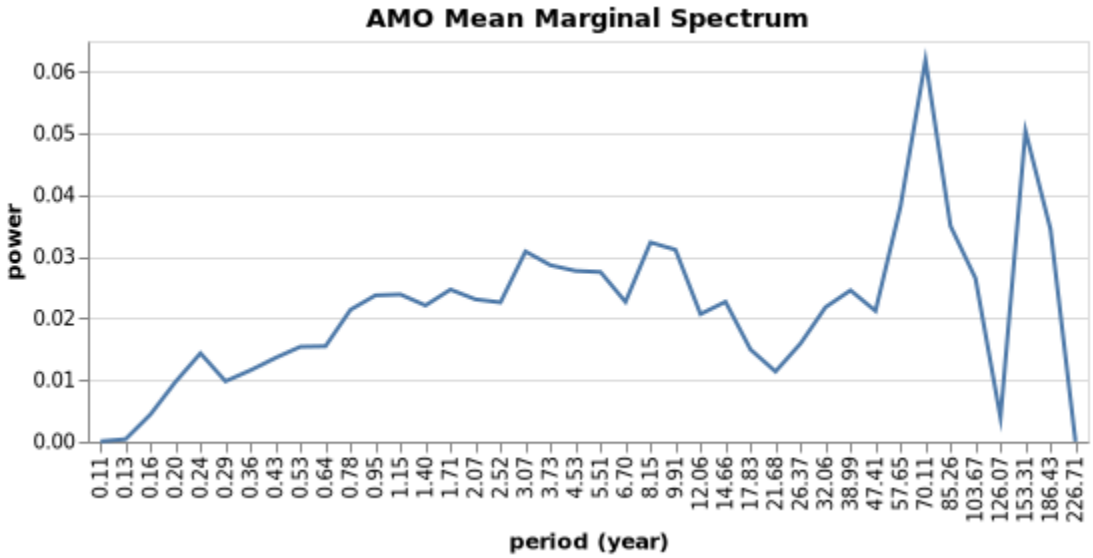

Figure 49: Mean marginal spectrum for AMO plot, derived from HHT.

But is still useful, nevertheless. The Multidecadal component is clearly visible, as well as the stable 150yr cycle corresponding to IMF 12. However, the 5yr and 10yr cycles appear lost under the general power sub-12yr cycles. This lends itself to the fact that the interesting aspects of the 5yr and 10yr cycle derive not from their existence in the data set, but rather from how they evolve over the duration of the time series. This is lost in marginal views like the MMS and the Fourier Decomposition:

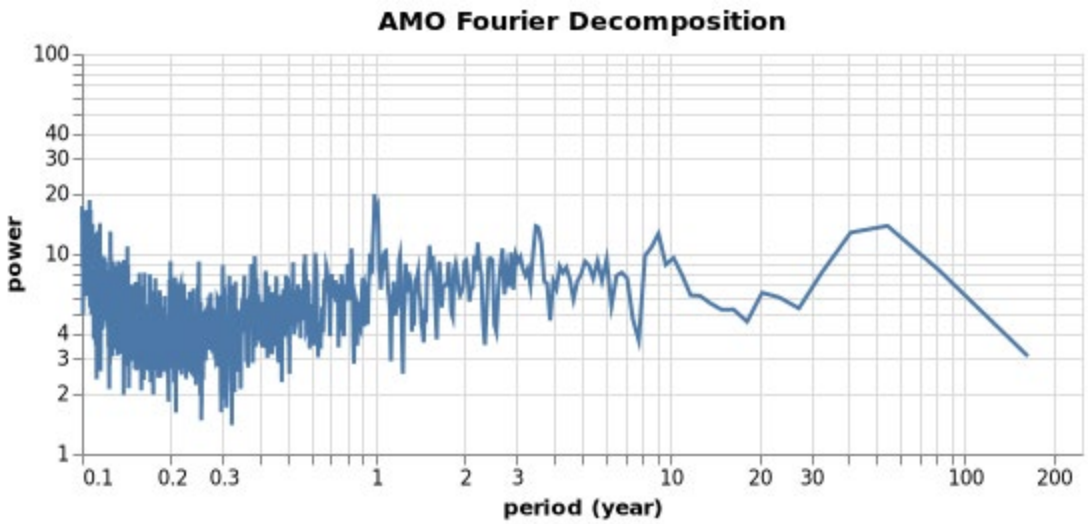

Figure 50: Fourier decomposition for AMO time series, in contrast with its mean marginal spectrum.

The Fourier Decomposition, at least, can resolve the 10yr and a 3.5yr cycle, due to its greater resolution at higher frequencies. It is even able to resolve a low bandwidth 1yr seasonal periodicity, which the mean marginal spectrum could not. However, at higher frequencies, its low resolution makes it barely able to resolve the multidecadal component, and completely unable to resolve the long-term 150yr cycle.

The plot of Degree of Stationarity aligns with our expectations:

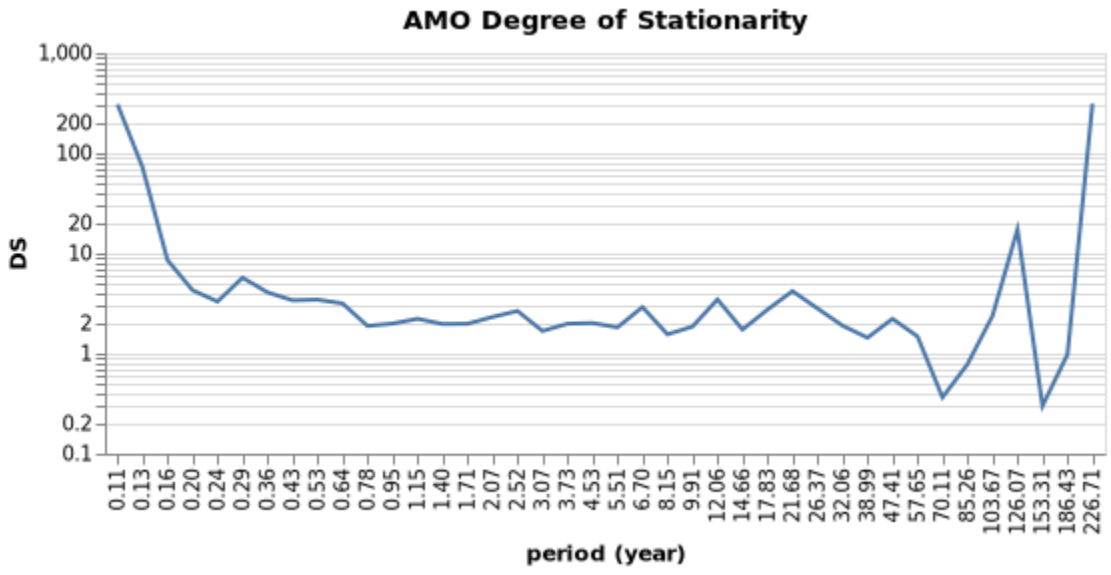

Figure 51: Degree of stationarity plots for AMO time series

This time series is extremely non-stationary at low periods and high frequencies, which we see from the dense spectrum and wavelet transform. However, there is a strong degree of stationarity at the Multidecadal aspect, which dominates for a large portion of the time series. Additionally, the most stationary aspect of the time series is its long-term 150yr variation, associated with IMF 12. From our analysis, we know it is constant-power and constant-frequency, and this stationarity is reflected in our Degree of Stationarity plot.

6.7 Conclusion

In conclusion, the Hilbert-Huang Transform is a powerful tool when used in conjunction with tools like Fourier Analysis and Wavelet Analysis. Not only is it more suited for analyzing non-linear and non-stationary data, it may also serve as a measure of discussing the validity of specific stationary analysis.

For the simulated data, it was able to more meaningfully pick out the components that we were interested in, and the components corresponding to the underlying system we were constructing. For the actual climate data, we observe that effects in non-stationarity and non-linearity were properly observed in the result of the transformation. It provided us the unique ability to "track" developments of contributions to the total time signal as entities that varied over time.

7 ENSO Effects on NRB Rainfall analyzed via EMD

7.1 Introduction

The El Niño Southern Oscillation (ENSO) is a major global contributor to interannual climate variability due to its influence towards disrupting normal large-scale Walker circulation in the South Pacific (Rasmusson and Arkin 1985; Slemr et al. 2016). The source of its disruption stems from the variability in the strength of the easterly trade winds (L'Heureux 2014). ENSO's impact on a global scale is often reflected in local events such as extreme flooding to extreme drought conditions (Fer et al. 2017). Le et al. (2017) modeled the interaction between large ENSO seasons and drought in North America, confirming the relationships established by Ropelewski et al. (1986), Dai and Wigley (2000), and Holmgren et al. (2006).

In East Africa, the Empirical Orthogonal Teleconnection (EOT) technique is utilized to isolate specific ENSO-driven patterns showing the direct connection with vegetation and agriculture yields (Van Den Dool et al. 2000; Fer et al. 2017). As a driver of interannual climate variability, with great impact on food security, ENSO is known to strongly contribute to Nile River Basin (NRB) precipitation patterns (Abtew et al. 2009). The Nile River in East and Northeast Africa drains an area of $2.9 \times 10^6 km^2$, and has strongly shaped the economic development of all countries in its drainage basin. Nile river flow plays an important role in all countries within its basin, and is mostly influenced by East and Northeast Africa's climate. Precipitation in these regions contributes to the overall flow of the River Nile from Tanzania, in the south, to Egypt, in the north. Therefore, an understanding of the driving forces affecting its flow is crucial for the purpose of

characterizing its impact on local economies, which in turns, requires the investigation of geographical variability of precipitation within the NRB. The river has two main tributaries, namely, the White Nile, which flows from Lake Victoria along the Kenya/Tanzania Border, and the Blue Nile that flows from Lake Tana in Ethiopia. We will focus our attention on the Blue Nile, since it is known that its flow is affected by the strength in the ENSO cycle (Amarasekera et al. 1997) and it contributes up to 60% of the River Nile yield. Significant negative and positive correlations between Pacific Ocean Sea Surface Temperature (SST) and the Nile River discharge were found (Amarasekera et al. 1997) Specific portions of discharge and precipitation, influenced by Pacific SST variations, were characterized to describe the regional variability for these contributing proportions. Millions of people in the semi-arid to arid regions of Kenya, Ethiopia and other NRB countries are facing water scarcity and frequent drought issues that might be linked to ENSO (Zaroug et al. 2014; Thomas et al. 2019). They found, using a discrete event-based approach that ENSO affects the region around the Blue Nile source in Lake Tana, which contributes around 60-69% of the main Nile discharge. The Pacific Ocean Sea Surface Temperature in the Niño 3.4 region and the meteorological and hydrological drought measurements in the upper catchment of the Blue Nile were used in that analysis. The El Niño and La Niña occurrences and associated intervals matched significantly with the patterns of flooding and drought events. The principal component of precipitation variance is an annual cycle (seasonal variation) – characterized by rainy seasons typically between July to September (Salih et. al., 2018). However, for the purposes of long-term planning, effort has been dedicated into identifying longer-scale variations and the signal driving variation between different years and decades.

In an effort to develop flooding and drought models, Siam and Eltahir (2015) analyzed historical data sets and defined four distinct modes of natural variability in NRB flow. They identified a

region in the Southern Indian Ocean that characterizes up to 28% of the interannual variability in Nile River flow. Together with historical ENSO readings, Pacific Ocean SST and SOI variation explain 44% of Nile River flow variability. In addition, they link anomalous events and show that global models incorporating ENSO can be used to characterize the NRB hydrology. To predict river flow at specific locations, Wang and Eltahir (1999) aggregated several historical data sets and other sources of historical information regarding ENSO indices, Ethiopia precipitation, and Nile flow readings, to establish predictive indices for Aswan flow. They applied Bayesian analysis using conditional categorical probabilities to create a discriminant forecasting algorithm. A synoptic index is constructed to characterize the forecast skill. They conclude that ENSO readings are by far the most valuable predictor on large (2-3 months) time scales, but that precipitation and river flow information can be useful in predicting on medium-range (monthly) time scales.

In this study, we further quantify the link between El Niño and decadal-scale variation in NRB precipitation by applying Empirical Mode Decomposition (EMD) and the Hilbert-Huang Transform (HHT) with the purpose of decomposing the signal in terms of a small number of Intrinsic Mode Functions (IMFs) characterizing their non-stationary oscillatory variations. In the process, we find that a specific NRB IMF and a specific Southern Oscillation Index IMF (namely, the IMF characterizing approximately duodecadal variations) are strongly correlated. This seems to imply the existence of a single global physical process driving both NRB precipitation and the interannual variation present in the SOI. In this paper we quantify the nature of this shared causality, and we show that such causality exists at different lagged responses. This link provides us a powerful statistical insight on NRB precipitation on a per-region basis, an important tool for characterizing its long-term variability, and also a viable predictive index for important hydrological measurements such as the Blue Nile yield. It is important to note that the goal is not

to establish a causal link between SOI and NRB and Blue Nile Yield, but rather that all three share mutual driving process that influences them, more clearly identifiable using Empirical Mode Decomposition than simple correlation or direct analysis.

The high social importance of these indices in the face of global climate change is a global crisis, as argued at the Davos world economic forum. Transnational water management is a critical issue in the upcoming years as the world moves to address the 2030 Sustainable Development Goals (SDGs), as both a goal contributing to water security in the face of climate change and also as an important indicator (Indicator 6.5.2) of the progress of the global motion towards a sustainable world.

7.2 Study Area

The primary area of study for this paper involves the atmospheric study of the primarily East-Africa Nile River Basin region (depicted in Figure 52) as it relates to global ENSO effects, and impacts the fields of atmospheric science as well as the mathematical and signal-processing fields of empirical signal decomposition.

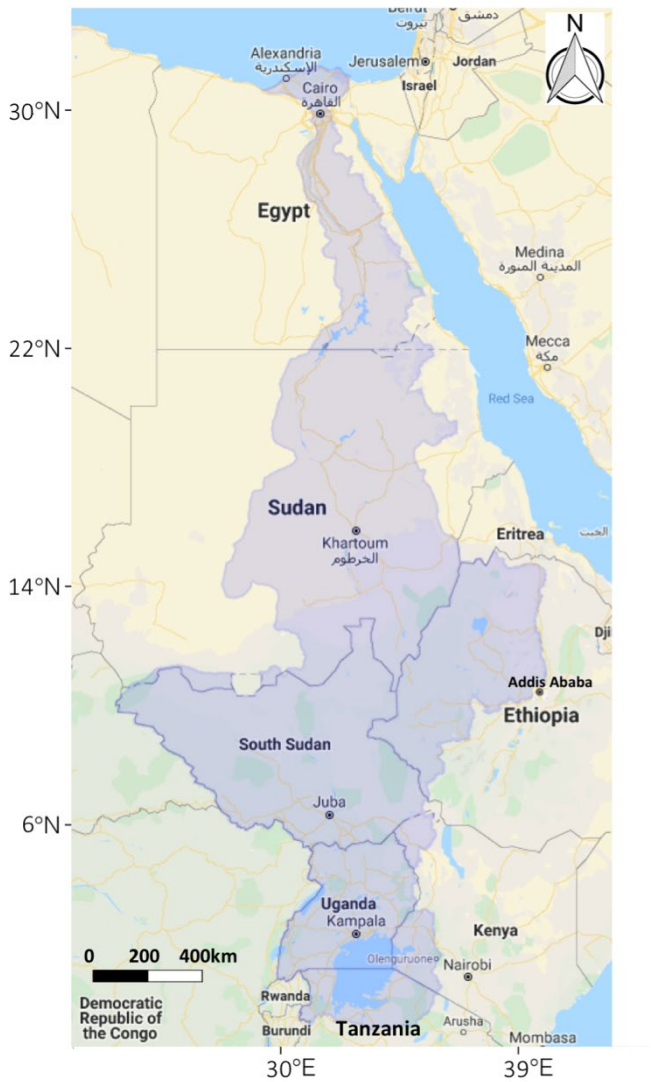

Figure 52: Nile River Drainage basin, the area of study. (This figure is taken from Li et. al., 2020, and used with permission)

7.3 Material and Methods

Materials

EMD and HHT analysis is performed on precipitation records from the CHIRPS Pentad Dataset (Funk et al. 2014), and also the Southern Oscillation Index, a climatology index associated with ENSO (Chen 1982). SOI data is gathered from the work in Ropelewski and Jones (1987), which contains records starting from January 1866. Historical precipitation (measured as monthly

anomalies of the Pentad temporal scale) for the NRB countries of Egypt, Ethiopia, Kenya, South Sudan, Sudan, Tanzania, and Uganda, starting from January 1981 is analyzed. Blue Nile flow data at Grand Ethiopian Renaissance Dam (GERD) measurement station from 1990 to 2014 was available from official GERD communications and resources. Software implementation of EMD is written in Haskell (Le 2018).

Methods

Empirical Mode Decomposition

EMD has successfully been used to study nonstationary physical systems, in fields ranging from neuroscience (Pachori 2008; Pigorini et al. 2011) to solar surface dynamics (Nakariakov et al. 2010; Bellini et al. 2014). IMFs proved to be powerful tool for predictive analysis, where using statistical models to predict IMFs, one can predict the progression of the time series as a whole (Abadan and Shabri 2014). In this research we applied the EMD and HHT, as a powerful predictive tool, on the NRB precipitation by isolating different physical processes amongst which is the one driven by ENSO as represented by SOI data (Huang et al. 1996, 1998). The power of EMD in isolating physically meaningful signals, driven by El Niño and quasi biennial oscillation, from precipitation and temperature data was proven over South Africa (Zvarevashe et al. 2019) and central and eastern pacific (Kidwell et al. 2014). On the eastern and western US, namely Virginia and California, EMD along with other tools showed the El Niño impact on precipitation variability using rain gauge and climate division data (El-Askary et al. 2004, 2012).

EMD aims to decompose a time series, precipitation in our case, as a sum of a small number of non-stationary components, IMFs, which may be understood and analyzed in isolation. Each IMF traces an independent non-stationary physically meaningful process that contributes to the full

series, for example seasonality, annual variability, El Niño cycle, decadal oscillation, etc. The HHT re-frames each series as instantaneous-frequency-over-time (much like sparse wavelet decomposition), tracing the progression of each IMF over instantaneous frequency space as a function of time. Thus, we can trace the process of one single physical process as it moves through instantaneous frequency space over time. For a general real-valued time series $x(t)$ of length T, the series is decomposed in terms of a sum of N (typically small) mutually orthogonal IMFs $c_i(t)$ and a residual series $r(t)$.

$$x(t) = \left[\sum_{i}^{N} c_i(t)\right] + r(t) \tag{1}$$

The HHT then allows the visualization of each IMF c_i as a curve in frequency-time space $\omega_i(t)$ with a magnitude $A_i(t)$ associated at each point.

$$c_i(t) \Rightarrow \langle \omega_i(t), A_i(t) \rangle \tag{2}$$

<u>Physical Interpretation of IMFs for Precipitation and SOI Data</u>

EMD produces IMFs which are mutually orthogonal for practical purposes, and each correspond to the contribution of an independent non-stationary physical process (or the sum of independent physical processes with similar time scales of variability). Junsheng et al. (2006) has shown that when one has a time series when the underlying physical processes are known, EMD yields IMFs that matches on each underlying series. Figure 53 shows EMD applied to the monthly Ethiopia precipitation records (recorded as monthly anomalies in the Pentad temporal scale) since 1980, yielding IMFs with different time scales of variability. The collection of nine IMFs is mutually

orthogonal in L^2 (by their construction), and, according to the theory of EMD, each IMF most likely tracks the progression of a separate physical process driving Ethiopia precipitation.

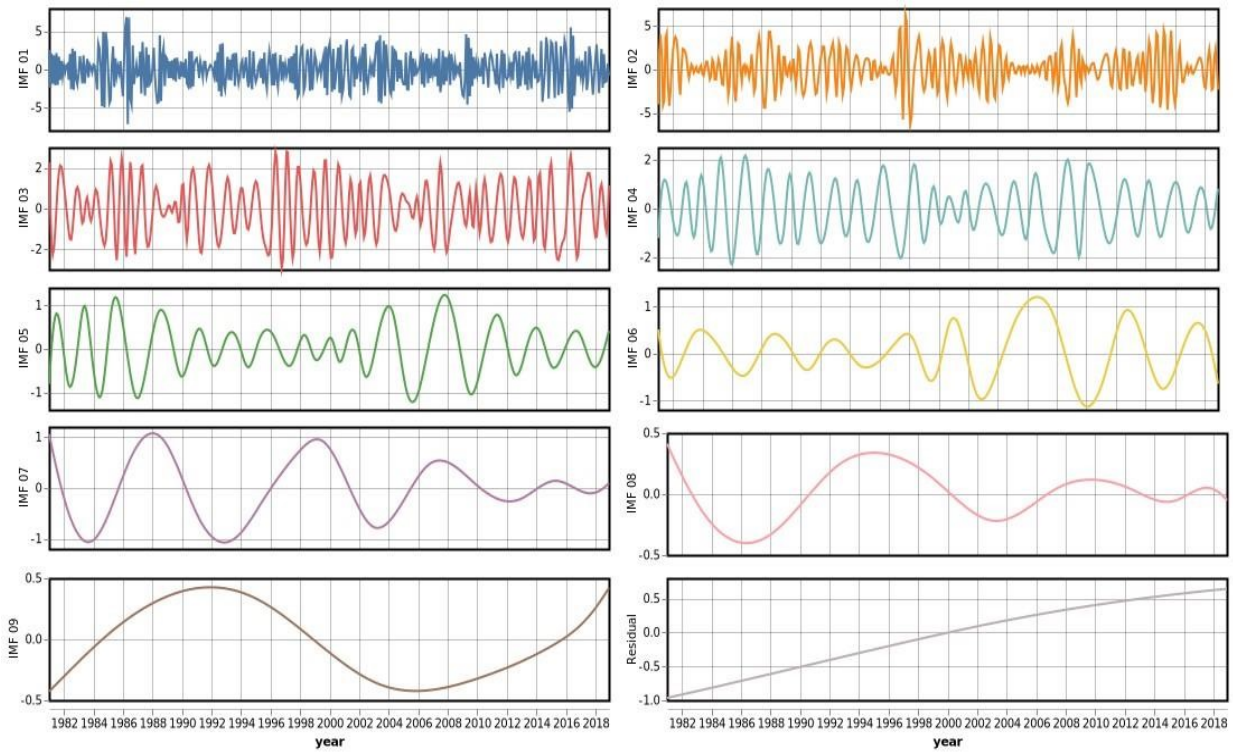

Figure 53: IMFs from EMD applied to historical Ethiopia Precipitation.

The full decomposition of SOI monthly recordings from January 1866 to January 2019 (Figure 54) isolates 14 IMFs at varying time scales. Of these, it can be proposed that IMF 6 corresponds to El Niño and La Niña occurrences: its non-stationary periodicity matches the historical record of large El Niño and La Niña events. In particular, the three largest El Niño events in recorded history are observed in 1982 – 83, 1997 – 98, and 2014 – 2016 as negative swings in IMF 6. El Niño events with their varying strength and impact on wetness and dryness were presented using recurrent neural networks by Le et al. (2017). Performing a correlation analysis between annual totals of IMF 6 and other ENSO SST time series (such as NIÑO 3.4, NIÑO 1, NIÑO 4) yields correlation

coefficient around 0.5, similar to the results of directly performing correlation analysis between SOI and such SST indices.

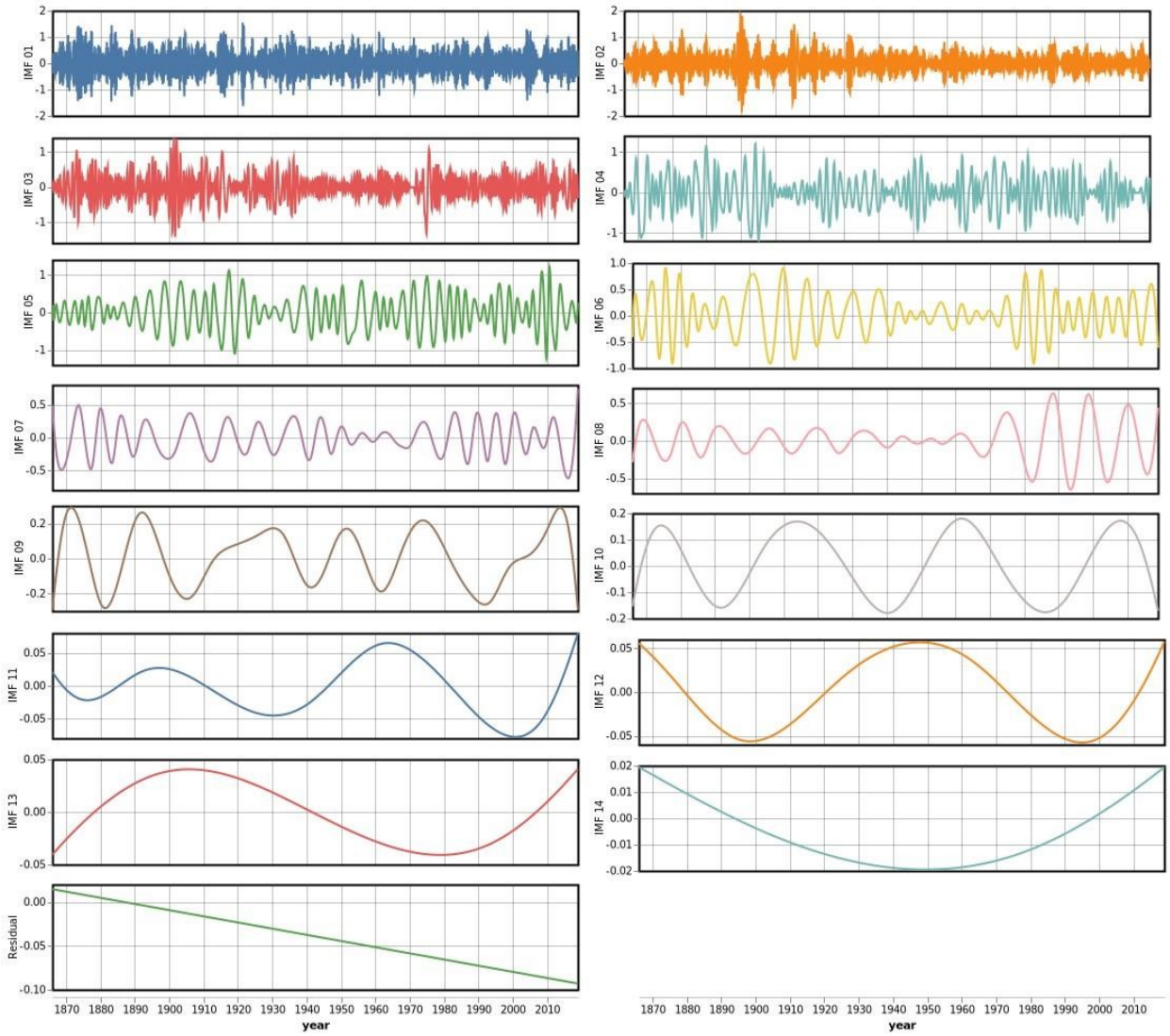

Figure 54: IMFs from EMD applied to historical SOI records

7.4 Results and Discussion
Hilbert-Huang Transform

Application of the HHT to the SOI shows the progression of each of these IMFs through frequency as a function of time, as depicted in Figure 55. The transform shows the range of variability in

which each IMF dominates. IMFs have been shown to correspond to meaningful physical processes when applied to a wide variety of physical systems. By studying a single IMF, it is possible to analyze a single physical process contributing to the variation of the system at that time scale. It is also possible to match this observed physical subprocess with other known physical processes. For example, IMF 1 accounts for quarterly variations, IMF 3 accounts for annual variations, and IMF 7 accounts for variations on the order of six to twelve years. Longterm 30-year variations are accounted for in IMF 9. By this association, IMF 6 corresponds to variations in the strength of the Easterly Trade Winds and size of Walker Cell disruptions, factoring the influence of climatology as previously discussed (Le et al. 2017). The (nonstationary) oscillation of IMF 6 represents the fundamental periodicity of this cycle, while isolating out variations in El Niño/La Niña intensity as reflected in SOI. It can be used as a binary indicator, if symmetric thresholds are used, to determine if such an event occurs, while not being influenced by the relative intensity of each event. However, on top of the dominant periodicity, there are extra factors that drive the relative strength between El Niño events as reflected by SOI. These factors, by the orthogonality conditions of EMD, are seen to be captured largely in IMFs 7 and 8.

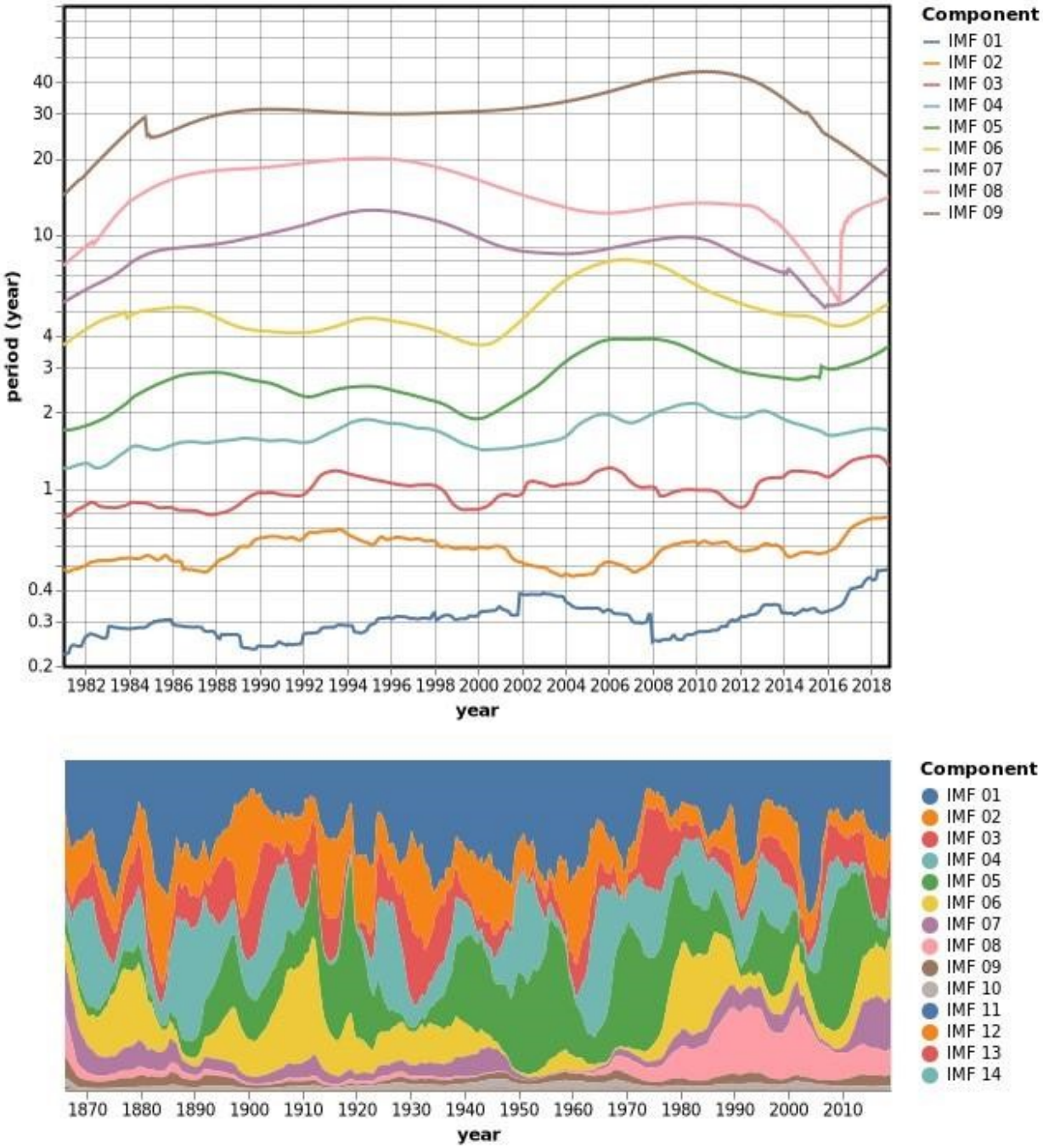

Figure 55: Two displays of data resulting from HHT transformations. (a) Skeleton lines arising from HHT from historical Ethiopia Precipitation IMFs. (b) Stacked area plot of SOI IMF relative instantaneous power.

Therefore, it is clear that the decomposition of the Ethiopia precipitation time series isolates nine independent, mutually orthogonal signals that correspond to non-stationary physical processes at different time scales. Those independent signals form the overall variation of the precipitation record over Ethiopia which can be robustly scaled to the whole NRB region.

In those IMFs, we observe the relative strength of the physical process driving the variation in effects of El Niño events as reflected through negative anomalies in SOI over time, with large swells in times of larger and more intense events. This fact can be seen in the stacked area plot (Figure 55b) of instantaneous power of each IMF, derived from the HHT. Each layered color represents the relative power of the influence of each IMF at each point in time. While IMF 6 (the primary event signal) has a relatively steady contribution (except during the mid-century lull), IMFs 7 and 8 appear to surge in power during known spikes in event strength.

Therefore, although we found that IMF 6 is tracking the periodic events in SOI itself, there is a separate, orthogonal physical process tracked by IMFs 7 and 8 that drives internnual Niño variability as reflected in SOI. There is therefore an underlying process, which has not yet been identified so far and is currently not yet studied, that contributes to interannual variability. It is an orthogonal physical circulation that strongly determines the relative strength of subsequent events. Hence, the SOI IMF 6 is now considered to be an El Niño indicator index, due to its ability to identify El Niño and La Niña events, isolating out variations in intensity. As a result, the IMF 6 is now useful as a "binary" indicator in establishing whether or not an El Niño or La Niña event is happening (by setting symmetric thresholds about 0). The SOI IMFs 7 and 8 are named the Interannual ENSO Variability Indices (IEVIs), as they are interannual variability indices that are derived from the study of ENSO. We differentiate between them as IEVI α and IEVI β, respectively, as a pair of indices for predicting the intensity of a given El Niño or La Niña event,

should one occur in that year. Hence, these IEVIs would shed the light on NRB precipitation linkage as discussed later.

NRB Precipitation and SOI Data Comparison

Applying the EMD and HHT to NRB precipitation records (as Pentad anomalies) from January 1981 to December 2018, shows that many NRB precipitation IMFs, especially decadal IMF, correlate strongly with the IEVIs, particularly with IEVI β, yet with a varying ranges of lag (Figure 5). NRB precipitation IMFs correlations with SOI IMFs are represented at different time scales. Each IMF is noted with the timescales it varies in, derived from the HHT, and each correlation coefficient is noted with the NRB IMF delay, computed using direct descent, for that correlation. In this method the lag is increased, up to four years, as long as the correlation also increases, until the point where increasing lag will decrease the correlation coefficient. This provides an effective measure to obtain a meaningful lag without the risk of overfitting which would occur if lags are permitted to slide past a local maximum. It is clear that precipitation for every NRB country yields an IMF that highly correlates with IEVI β, SOI IMF 8, predominantly at zero lag. Yet it is noteworthy that precipitation for the majority of NRB countries still yield an IMF that correlates, in some weaker manner, with IEVI α, SOI IMF 7. Each Precipitation IMF corresponds to an underlying physical process that drives the variation in precipitation for that country. The fact that Ethiopia IMF 7 correlates at $\rho = 0.719$ with IEVI β at 0 month lag means that the underlying physical process driving the interannual variability (IEVI β) is the same as the one driving variability Ethiopia precipitation between different decades. Therefore, it can confidently be concluded that interannual variability is strongly associated with decadal variability in this case owed to the observed high correlation at 0 lag.

Table 4: Correlations between NRB nations precipitation IMFs and SOI IMFs. Each IMF is noted with the approximate range of periodic variability the IMF accounts for, and each correlation is noted with the lag of correlation in months.

	SOI 5 (~3.47 yr)	SOI 6 (~6.04 yr)	SOI 7 (~8.43 yr)	SOI 8 (~12.09 yr)	SOI 9 (~30.11 yr)	SOI 10 (~44.56 yr)
Egypt 5 (~2.91 yr)	-0.25 (3)	-0.20 (4)	0.02 (0)	0.05 (4)	-0.04 (5)	-0.00 (0)
Egypt 6 (~5.65 yr)	-0.13 (6)	0.21 (2)	-0.09 (0)	-0.13 (35)	-0.08 (37)	-0.07 (11)
Egypt 7 (~8.79 yr)	-0.04 (0)	0.04 (0)	-0.01 (3)	-0.66 (0)	-0.07 (0)	-0.06 (5)
Egypt 8 (~12.41 yr)	0.01 (0)	0.06 (0)	-0.35 (0)	-0.49 (0)	-0.09 (0)	-0.05 (0)
Egypt 9 (~32.14 yr)	0.03 (14)	0.05 (9)	0.01 (0)	0.09 (0)	-0.90 (0)	-0.95 (33)
Ethiopia 5 (~3.01 yr)	0.22 (0)	0.34 (1)	0.09 (18)	0.00 (0)	0.01 (1)	0.02 (0)
Ethiopia 6 (~5.34 yr)	-0.22 (9)	0.24 (6)	0.19 (4)	-0.23 (22)	-0.01 (0)	0.05 (3)
Ethiopia 7 (~9.64 yr)	0.04 (0)	0.08 (25)	-0.34 (18)	0.72 (0)	0.09 (0)	0.12 (10)
Ethiopia 8 (~15.83 yr)	0.04 (0)	0.05 (14)	0.08 (0)	-0.16 (0)	-0.33 (47)	0.16 (11)
Ethiopia 9 (~28.95 yr)	-0.00 (1)	0.07 (10)	-0.15 (19)	0.12 (0)	-0.73 (0)	-0.84 (25)
Kenya 5 (~2.72 yr)	-0.67 (2)	-0.18 (6)	-0.02 (0)	0.04 (3)	-0.02 (4)	0.02 (4)
Kenya 6 (~5.47 yr)	-0.18 (8)	0.44 (4)	0.13 (2)	0.16 (0)	0.09 (46)	0.07 (20)
Kenya 7 (~10.05 yr)	-0.08 (0)	-0.15 (0)	0.48 (1)	0.62 (34)	0.10 (5)	0.07 (0)
Kenya 8 (~16.38 yr)	0.03 (0)	-0.01 (0)	0.17 (0)	0.19 (19)	-0.42 (44)	-0.03 (0)
Kenya 9 (~35.49 yr)	-0.04 (15)	-0.04 (14)	-0.09 (0)	-0.13 (0)	0.87 (10)	0.85 (47)
South Sudan 5 (~3.07 yr)	0.11 (0)	0.28 (1)	-0.05 (0)	0.01 (0)	0.05 (11)	-0.03 (3)
South Sudan 6 (~5.07 yr)	-0.17 (17)	0.23 (2)	0.41 (5)	0.08 (4)	0.05 (0)	0.03 (7)
South Sudan 7 (~11.32 yr)	-0.03 (6)	-0.07 (0)	0.15 (0)	0.68 (15)	0.10 (0)	0.14 (5)
South Sudan 8 (~17.91 yr)	0.01 (0)	-0.05 (12)	-0.12 (0)	-0.19 (0)	0.11 (0)	-0.07 (0)
South Sudan 9 (~29.39 yr)	0.00 (0)	0.07 (9)	-0.15 (18)	0.09 (0)	-0.79 (0)	-0.88 (29)
Sudan 5 (~3.46 yr)	0.01 (0)	0.20 (0)	0.12 (16)	0.11 (24)	-0.01 (0)	-0.06 (0)
Sudan 6 (~6.02 yr)	-0.13 (14)	-0.20 (3)	0.12 (0)	-0.16 (45)	-0.07 (31)	-0.01 (6)
Sudan 7 (~6.90 yr)	0.04 (0)	0.06 (0)	-0.65 (18)	0.38 (3)	0.02 (0)	0.15 (10)
Sudan 8 (~10.14 yr)	0.03 (0)	0.13 (21)	-0.27 (28)	0.86 (6)	-0.10 (38)	0.07 (8)
Sudan 9 (~22.17 yr)	0.05 (0)	0.06 (13)	0.04 (0)	-0.22 (1)	-0.26 (0)	-0.24 (47)
Tanzania 5 (~3.01 yr)	-0.51 (0)	-0.12 (0)	-0.16 (2)	0.06 (3)	0.06 (47)	0.05 (17)
Tanzania 6 (~4.96 yr)	-0.15 (0)	0.03 (0)	0.30 (15)	0.02 (0)	-0.10 (47)	-0.08 (47)
Tanzania 7 (~7.78 yr)	-0.03 (0)	-0.00 (0)	0.23 (0)	0.43 (0)	0.05 (0)	0.01 (0)
Tanzania 8 (~11.24 yr)	0.02 (0)	0.15 (16)	0.32 (0)	0.63 (0)	-0.24 (29)	0.07 (0)
Tanzania 9 (~16.13 yr)	-0.03 (1)	0.11 (16)	0.07 (0)	0.17 (0)	-0.35 (8)	-0.19 (47)
Uganda 5 (~2.58 yr)	-0.41 (3)	0.18 (6)	-0.06 (9)	-0.05 (3)	-0.02 (0)	-0.05 (5)
Uganda 6 (~4.69 yr)	-0.20 (8)	0.23 (0)	0.38 (6)	0.00 (2)	0.03 (0)	0.04 (5)
Uganda 7 (~7.73 yr)	0.03 (7)	0.19 (13)	0.18 (0)	0.51 (15)	-0.09 (31)	-0.01 (0)
Uganda 8 (~11.99 yr)	-0.06 (14)	0.01 (0)	-0.15 (0)	0.84 (29)	0.28 (6)	0.07 (24)
Uganda 9 (~17.28 yr)	0.02 (0)	-0.04 (9)	-0.07 (0)	-0.15 (0)	0.21 (0)	0.09 (25)

The strong correlations between NRB precipitation IMFs and IEVI suggests that the Nile River yield and total accumulation is somehow dependent on ENSO strength and variability.

ENSO strength accounts for ~ 22% of the annual variance in the Blue Nile and Atbara rivers' flow, which primarily drain Ethiopia, Eritrea, Sudan, and South Sudan (Amarasekera et al. 1997). In agreement with these findings, we suggest that ENSO strength as reflected in the isolated IMF of SOI is strongly linked to precipitation in Ethiopia, the primary drainage basin of these two rivers. Therefore, we can deduce that the Ethiopia, Sudan, and South Sudan precipitation will be strongly linked with the IEVIs and ENSO. Speaking of the NRB countries, the link is established with varying correlative power geographically with IEVI α (SOI 7) and IEVI β (SOI 8) (Figure 56).

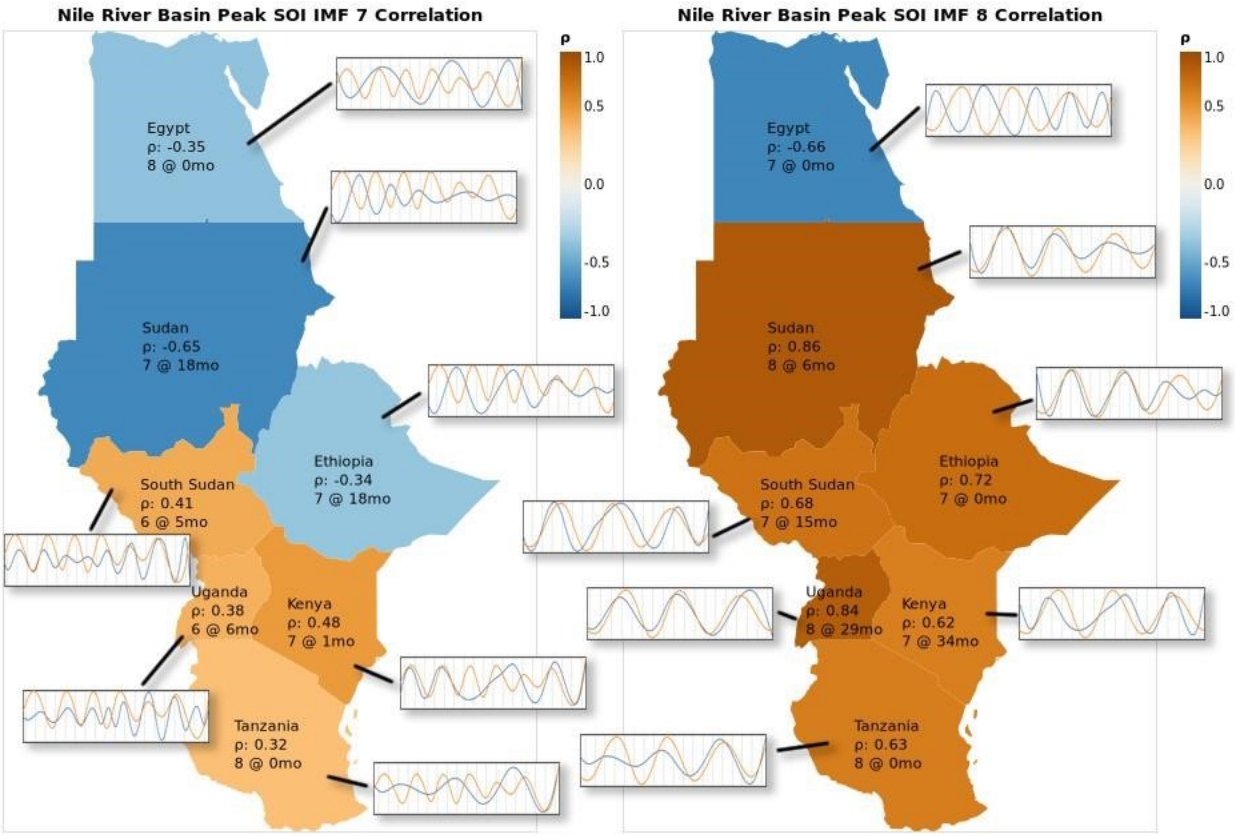

Figure 56: Map of NRB nations colored and overlaid with correlations between national precipitation IMFs and IEVI α (SOI 7) and IEVI β (SOI 8). Insets depict

the actual IMF of national precipitation against lagged IEVI component, where are highlighted and discussed in the text.

For instance, Ethiopia and Sudan show the strongest correlation with IEVI β, at 0.72 and 0.86, respectively, while South Sudan is not much lower, with 0.68. ENSO variation is known to have a much weaker influence on the White Nile flow (Amarasekera et al. 1997). This adds to our confirmed observations that Kenya and Tanzania, out of all NRB countries, have the two lowest correlations with IEVIs. Precipitation IMFs for Ethiopia and northward, downstream of Lake Tana, negatively correlates with IEVI α, opposite to countries extending from Lake Victoria to Sudan. However, precipitation for all NRB countries, except for the Mediterranean bordered Egypt, positively correlates with IEVI β. Because of our usage of IEVI β instead of a direct El Niño index, we can be certain that our claims of dependency and correlation, specifically, result from the interannual variability derived from ENSO, and not simply SST or other environmental internnual factors. In other words, we track ENSO cycle, specifically, and not any other potential overlapping periodicity.

Blue Nile Yield Prediction

IMFs of physical systems can be predicted using traditional statistical models, such as ARIMA models (Abadan and Shabri 2014). A hybrid discrete Bayesian model proves to be effective in linking ENSO-based factors and NRB precipitation activity (Wang and Eltahir 1999; Siam and Eltahir 2015). If a traditional model can project IEVI α and β, then it is possible to predict on the precipitation levels decadal variability for NRB countries. This is important for NRB nations with lagged correlations between precipitation IMFs and IEVIs. The importance stems from the fact that predictions in IEVIs will manifest as correlations, at a known lag time, in precipitation of NRB

countries. Therefore, we can for instance predict any hydrological variable that might be driven by precipitation. To demonstrate this ability the Blue Nile yield will be predicted annually based on IEVIs and autocorrelative terms. The Blue Nile yield data from the GERD measurement site from 1900 to 2014 was made available from official sources through GERD communications. The location of the measurement station with respect to the watershed of the Blue Nile river is depicted in Figure 7c. Since the IEVIs have a monthly sampling frequency, as opposed to the yearly frequency of our yield data set, next year prediction will be based on two groups of predictors. These are namely, the twelve IEVI α measurements from the year before the observed measurement and the autocorrelative terms represented by the six previous measurements of the Blue Nile yield. For simplicity, this is a simple ARMA model represented as a multivariate linear regression on annual total Blue Nile Yield y_i:

$$y_i = \boldsymbol{b}^\top \boldsymbol{\alpha}_{i-1} + d_1 y_{i-1} + d_2 y_{i-2} + \cdots + d_6 y_{i-6} \tag{3}$$

$\boldsymbol{\alpha}_i$ is the 6-vector of IEVI $\boldsymbol{\alpha}$ readings for year i for the months of June to November, and y_i is the Blue Nile Yield for year i for months June to November. The model is parameterized by a 6-vector $\boldsymbol{b}$ $(b_1, \ldots, b_6)$ and the six coefficients $d_1, \ldots, d_7$. These parameters are fitted according to ordinary linear least squares estimation (Hayashi 2000). The actual estimation involves a series of matrix multiplications and inversions, involving the Moore-Penrose Pseudoinverse methodology (BenIsrael and Greville 2003).

Our justification for the ARMA model arrives from the fact that it is the simplest possible model requiring the least a priori assumptions: it posits that the Blue Nile Yield has both linear auto-

regressive contributions (that is, that it is resistant to sudden year-by-year changes) and a linear moving average contribution from an external contributing factor (IEVI α, in our situation).

Figures 6a and b show the estimated model when fitted to the full time series, compared to actual historical Blue Nile yield. The error in the fully fitted model is RMSE $8.1 \times 10^9 m^3$, with a Pearson correlation coefficient of $\rho = 0.52$. The fitted model against IEVI α explains 30% of the variability of the Blue Nile Yield.

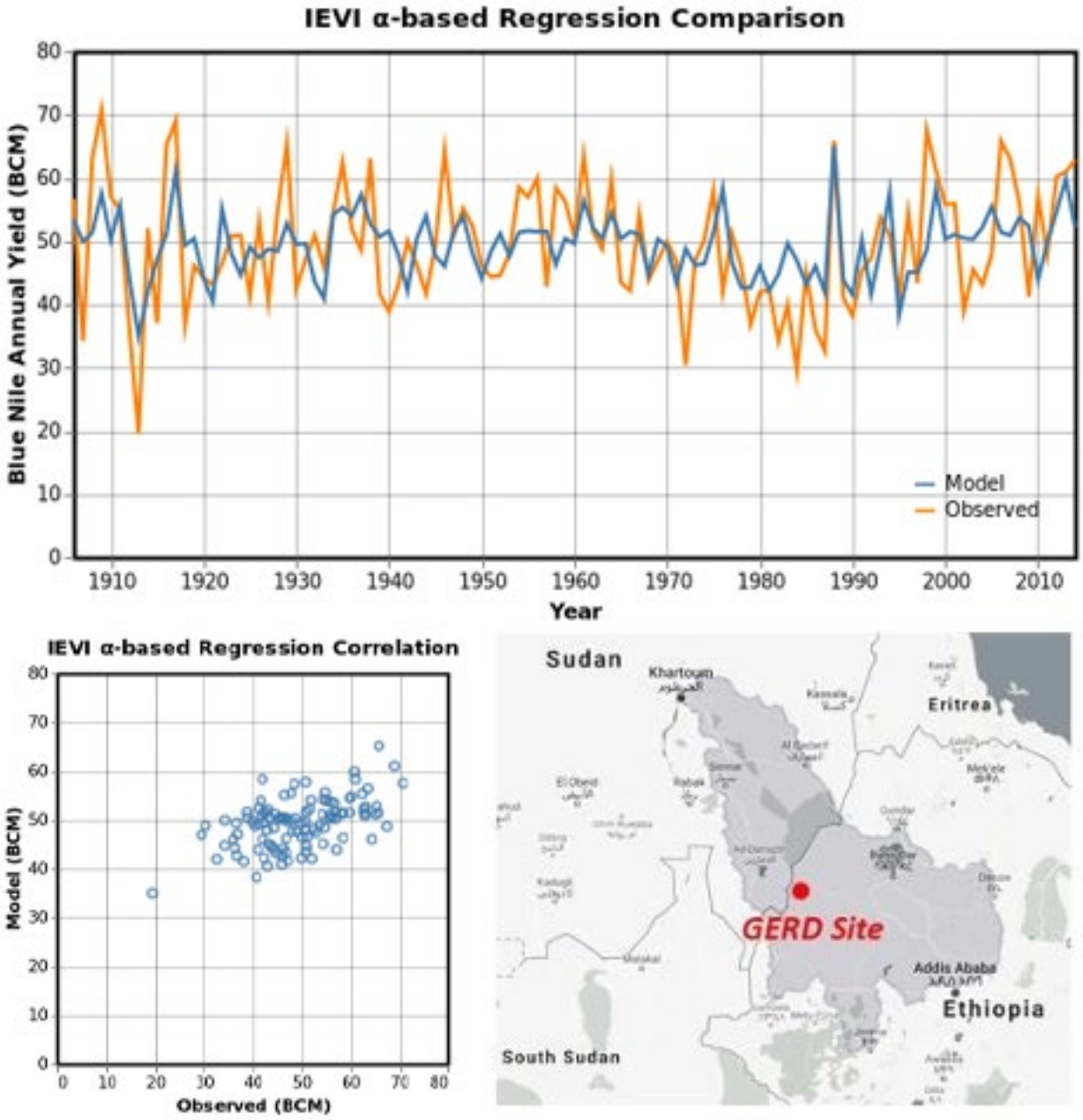

Figure 57: A multivariate linear regression based on IEVI to predict Blue Nile Yield. (a) Model output against measured values. (b) Correlation plot between model and measured values. (c) Location of measurement station with respect to the Blue Nile watershed (highlighted) and the surrounding regional borders.

Other model inputs were considered, such as precipitation, the actual SOI time series, and other IMFs — however, predicting only on IEVI α gives the most significant results against the null hypothesis. Predicting on other parameters using this method tends to overfit. This means that IEVI α is a stronger unbiased predictor than directly using SOI, or even NRB precipitation.

These methods also suggest that, while each SOI IMF is physically meaningful, IEVI α is closely tied with precipitation and related long-term phenomenon, such as drought and periods of heavy rain. This initial model strongly suggests that IEVI α's inherent physical properties lend itself to be able to predict on significant geophysical and hydrological processes. In the future, more advanced statistical or data-driven models may prove to be even more effective. This is a very important finding where it addresses the Blue Nile Yield as a very a significant measure in addressing a serious societal issue with transboundary implications on Ethiopia, Sudan and Egypt.

7.5 Conclusions

The 2020 Global Risk Report lists Climate Action failure and Extreme weather as top global risks in terms of both likelihood and impact. The SDGs, established in 2015, set a course of action for addressing upcoming potential global crises; SDG 6 acknowledges the role of Clean Water and Sanitation in sustainable development. The SDG establishes transboundary cooperation as an important indicator in the progress for this goal. Therefore, accurate modeling of transboundary hydrological resources like precipitation runoff and Nile River flow are integral in addressing the future of sustainability and climate action.

We have shown how the introduction of Climate Indices IEVI α and β account for the inter-annual variability of El Niño as reflected by SOI and also drives many physical processes. We have shown that our inter-annual El Niño variability index, expressed by IEVI β has extremely strong

correlations (up to $\rho = 0.864$) for NRB precipitation Decadal variation, as isolated by EMD. We have also shown that a statistically significant association of NRB Precipitation decadal variation is our inter El Niño variability index, expressed by IEVI β, and that these correlations should allow the IEVIs predictive models to characterize decade-to-decade precipitation levels on NRB nations. The geographic distribution of correlation with IEVI β also matches that predicted by the conclusions of the cited works. All countries but Egypt vary in precipitation in the same direction as IEVI β, whereas Egypt varies in the opposite direction. We attribute this change in direction due to Egypt's influence by the Mediterranean Sea's dependence on El Niño. A weaker effect (ρ $=$

0.44) is found in that all southern Nile River Basin countries vary in the same direction as IEVI α, whereas all northern NRB countries downstream of Lake Tana vary in the opposite direction.

Physically, our conclusions match known properties about the NRB. Comparing the relative dependence of Blue Nile and White Nile dependence on ENSO based on established literature, we observe the correct geographical distribution of correlation. We expand on previous results by uncovering a more physically meaningful index on which to build models and make predictions, instead of simply raw SOI and precipitation. By applying EMD, we aimed to isolate a signal corresponding only to inter-event variability. Because of this filtering, our correlation factors are known to correspond only to inter-ENSO variability, and not only significant events themselves. This gives a strong footing on which to base claims about inter-ENSO variability (and not only interannual variability) as it affects Nile River flow. To solidify this claim, we predict on Blue Nile River yield based only on IEVI, with successful results. These results show that EMD has uncovered an underlying process mutually driving both ENSO (as reflected in SOI and

precipitation measurements) and also physical processes in the Nile River Basin. It should be noted that this does not attempt to justify a causal link between SOI and NRB processes, but rather a mutual causality. Further study should involve the usage of modeling IEVI β and IEVI α to show the exact accuracy of such models on decadal variability of NRB country precipitation, as well as further studies of the IEVIs against the precipitation and climate of other regions. In addition, further study could link the IEVIs to the variations within swells of the thermocline, Walker Circulation deviations, and Southern Easterly Trade Wind deviations.

8 Summary & Conclusion

Learning-based adaptive models show promise as tools within their proper context in climate science and other empirically based scientific methods, with their predictive power. One major factor limiting the usage of learning-based adaptive models is their inability to properly explain the predictions they make. While their predictive power has been established and is gradually gaining acceptance, science is fundamentally a tool for building understanding of the world. We believe that progress in the understandability of recurrent neural networks and in the analysis of EMD IMFs can help build progress along this front. This dissertation presents a road forward in this endeavor, in the hopes of inspiring further research in the future. Indeed, the assistance of empirical and adaptive models in guiding scientific progress (and the movement of focus away from simply their predictive power) may become an indispensable tool in the future of computer-aided scientific modeling. The role of adaptive models must not be as the replacement of traditional models, but rather as a tool to help test models and as a guiding tool to help guide scientific innovation. Adaptive models and machine learning must not supplant traditional scientific models. Instead, they must fill these new roles: roles that were much needed but missing or lacking in the past. Only working together and in complementary manner can traditional scientific models and machine learning/adaptive models create a future of more sustainable promise.

References

"Quantifying increased groundwater demand from prolonged drought in the East African Rift Valley." Science of the Total Environment 666 (May): 1265–72. https://doi.org/10.1016/j.scitotenv.2019.02.206.

Abadan, Sarah, and Ani Shabri. 2014. "Hybrid empirical mode decomposition-ARIMA for forecasting price of rice." Applied Mathematical Sciences 8: 3133–43. https://doi.org/10.12988/ams.2014.43189.

Abtew, Wossenu, Assefa M. Melesse, and Tibebe Dessalegne. 2009. "El Niño Southern Oscillation link to the Blue Nile River Basin hydrology." Hydrological Processes, n/a– n/a. https://doi.org/10.1002/hyp.7367.

Amarasekera, Kishan N., Robert F. Lee, Earle R. Williams, and Elfatih A. B. Eltahir. 1997. "ENSO natural variability flow tropical rivers." Journal of Hydrology 200: 24–39.

Asner, G., Brodrick, P.G., Anderson, C.B., Vaughn, N., Knapp, D.E., Martin, R.E, 2015. Progressive forest canopy water loss during the 2012-2015 California drought, Proceedings of the National Academy of Sciences, 2015, 249 – 255.

Badjate, S.L., Dudul, S.V., 2009. Multi step ahead prediction of north and south hemisphere sun spots chaotic time series using focused time lagged recurrent neural network model, WSEAS Trans. Inf. Sci. Appl., 6 (4) , 684 – 693.

Bellini, G., J. Benziger, D. Bick, G. Bonfini, D. Bravo, M. Buizza Avanzini, B. Caccianiga, et al. 2014. “Final results of Borexino Phase-I on low-energy solar neutrino spectroscopy.” Physical Review D 89 (11): 112007. https://doi.org/10.1103/PhysRevD.89.112007.

Ben-Israel, Adi, and Thomas N. E. Greville. 2003. Generalized Inverses. CMS Books in Mathematics. New York: Springer-Verlag. https://doi.org/10.1007/b97366.

Bodri, L., Cermak, V., 2000. Prediction of extreme precipitation using a neural network; application to summer flood occurrence in Moravia, Adv. Eng. Softw., 31, 211 – 221.

Capotondi, A., Wittenberg, A., Newman, M., 2014. Understanding ENSO Diversity. Bull. Amer. Meteor. Soc., 96, 921–938.

Changnon, S.A., 1999. Impacts of 1997-98 El Niño Generated Weather in the United States. Bull. Amer. Meteor. Soc., 80 (9), 1819–1827, doi: 10.1175/1520-0477(1999)080<1819:IOENOG>2.0.CO;2.

Chen, W. Y. 1982. “Assessment of Southern Oscillation Sea-Level Pressure Indices.” Monthly Weather Review 110 (7): 800–807. https://doi.org/10.1175/15200493(1982)110%3C0800:AOSOSL%3E2.0.CO;2.

Cigizoglou, H. K., Alp, M., 2004. Rainfall-runoff modelling using three neural network methods, Artificial Intelligence and Soft Computing - ICAISC 2004: 7th International Conference, 166 – 171.

Climate.gov (October 2015), what to expect this winter: NOAA’s outlook reveals what conditions are favored across the US. (Available at: https://www.climate.gov/news-

features/blogs/enso/what-expect-winter-noaa%E2%80%99s-outlook-reveals-what-conditions-are-favored)

Cook, B.I., Ault, T.R., Smerdon, J.E., 2015. Unprecedented 21st century drought risk in the American southwest and central plains, Science Advances, 1 (1), 10.1126/sciadv.1400082

Dai, A., and T. M. L. Wigley. 2000. "Global patterns of ENSO-induced precipitation." Geophysical Research Letters 27 (9): 1283–6. https://doi.org/10.1029/1999GL011140.

Daniele Struppa. 2012. "Computational methods for climate data." Wiley Interdisciplinary Reviews: Computational Statistics 4 (4): 359–74. https://doi.org/10.1002/wics.1213.

El-Askary, H, S Sarkar, L Chiu, M Kafatos, and T El-Ghazawi. 2004. "Rain gauge derived precipitation variability over Virginia and its relation with the El Niño southern oscillation." Advances in Space Research 33 (3): 338–42. https://doi.org/10.1016/S02731177(03)00478-2.

El-Askary, H., Allali. M., Rakovski, C., Prasad, A., Kafatos, M., Struppa, D., 2012. Computational methods for climate data, WIREs Comp. Stat., 4, 359–374. doi: 10.1002/wics.1213.

El-Askary, H., Sarkar, S., Chiu, L., Kafatos, M., El-Gahzawi, T., 2004. Rain Gauge Derived Precipitation Variability over Virginia and its Relation with the EL NIÑO Southern Oscillation (ENSO), Advances in Space Research, 33, 338-342, 2004, doi: 10.1016/S0273-1177(03)00478-2

Elman, J. L., 1993. Learning and development in neural networks: the importance of starting small, Cognition 48 (1), 71-99.

Fer, Istem, Britta Tietjen, Florian Jeltsch, and Christian Wolff. 2017. "The influence of El Niño-Southern Oscillation regimes on eastern African vegetation and its future implications under the RCP8.5 warming scenario." Biogeosciences 14 (18): 4355–74. https://doi.org/10.5194/bg-14-4355-2017.

Funk, C. C., Peterson, P. J., Landsfeld, M. F., Pedreros, D. H., Verdin, J. P., Rowland, J. D., … Verdin, A. P. (2014). A quasi-global precipitation time series for drought monitoring: U.S. Geological Survey Data Series 832. Usgs. https://doi.org/10.3133/ds832

Glorot, X., Bordes, A., Bengio, Y., 2011. Deep Sparse Rectifier Neural Networks, Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS-11), 15, 315-323.

Griffin, D., Anchukaitis, K.J., 2014. How unusual is the 2012-2014 California drought?, Geophys. Res. Lett., 41, 9017-9023.

Hayashi, Fumio. 2000. "Econometrics. 2000." Princeton University Press. Section.

Hoell, A., Hoerling, M., Eischeid, J., Wolter, K., Dole, R., Perlwitz, J., Xu, T., Cheng, L., 2016. Does El Niño intensity matter for California precipitation, Geophys. Res. Lett., 43, 819–825, doi:10.1002/2015GL067102.

Holmgren, M, P Stapp, C R Dickman, C Gracia, S Graham, J R Gutiérrez, C Hice, et al. 2006. "A synthesis of ENSO effects on drylands in Australia, North America and South America." Vol. 6. www.biouls.cl/enso/.

Hopfield, J.J., 1982. Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences, 79 (8), 2554-2558.

Howitt, R., Macewan, D., Medellin-azuara, J., 2014. Economic Analysis of the 2014 Drought for California Agriculture, Center for Watershed Sciences, University of California, Davis, California.

Huang, Norden E., Steven R. Long, and Zheng Shen. 1996. "The Mechanism for Frequency Downshift in Nonlinear Wave Evolution." Advances in Applied Mechanics 32 (C). https://doi.org/10.1016/S0065-2156(08)70076-0.

Jeong, D.I., Sushami, L., Khaliq, M.N., 2014. The role of temperature in drought projections over North America, Climatic Change, 127.

Junsheng, Cheng, Yu Dejie, and Yang Yu. 2006. "Research on the intrinsic mode function (IMF) criterion in EMD method." Mechanical Systems and Signal Processing 20 (4): 817–24. https://doi.org/10.1016/j.ymssp.2005.09.011.

Kahya, E., Dracup, J.A., 1993. U.S. Streamflow patterns in relation to the El Niño Southern Oscillation, Whater Resour. Res., 29, 2491 – 2503.

Kidwell, Autumn, Young-Heon Jo, and Xiao-Hai Yan. 2014. "A closer look at the central Pacific El Niño and warm pool migration events from 1982 to 2011." Journal of Geophysical Research: Oceans 119 (1): 165–72. https://doi.org/10.1002/2013JC009083.

L'Heureux, Michelle. 2014. "What is the El Niño–Southern Oscillation (ENSO) in a nutshell? NOAA Climate.gov." https://www.climate.gov/news-features/blogs/enso/whatel-ni%7B/~%7Bn%7D%7Do–southern-oscillation-enso-nutshell.

Le, J. A. (2019). emd: Empirical Mode Decomposition and Hilbert-Huang Transform in Haskell. Retrieved from https://hackage.haskell.org/package/emd

Le, J. A., H. M. El-Askary, M. Allali, and D. C. Struppa. 2017. "Application of recurrent neural networks for drought projections in California." Atmospheric Research 188: 100– 106. https://doi.org/10.1016/j.atmosres.2017.01.002.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning, Nature, 521, 436–444.

Li, W.; El-Askary, H.; Lakshmi, V.; Piechota, T.; Struppa, D. Earth Observation and Cloud Computing in Support of Two Sustainable Development Goals for the River Nile Watershed Countries. Remote Sens. 2020, 12, 1391. https://www.mdpi.com/2072-4292/12/9/1391

Luck, K.C., Ball, J.E., Sharma, A., 2000. A study of optimal model lag and spatial inputs to artificial neural networks for rainfal forecasting, J. Hydrol. , 227 (1 – 4), 141 – 150.

Maqsood, I., Khan, M.R., Abraham, A., 2004. An ensemble of neural networks for weather forecasting, Neural Comput & Applic., 13: 112-122, doi:10.1007/s00521-004-0413-4.

Moustris, K.P., Larissi, I.K., Nastos, P.T., Paliatsos, A.G., 2011. Precipitation forecast using artificial neural networks in specific regions of Greece, Water Resour. Manag., 25, 1979 – 1993.

Mozer, M. C., 1995. A Focused Backpropagation Algorithm for Temporal Pattern Recognition, Backpropagation: Theory, Architectures, and Applications, 137–169.

Nakariakov, V M, A R Inglis, I V Zimovets, C Foullon, E Verwichte, R Sych, and I N Myagkova. 2010. "Oscillatory processes in solar flares." Plasma Physics and Controlled Fusion 52 (12): 124009. https://doi.org/10.1088/0741-3335/52/12/124009.

Nastos, P., Pallastos, A., Koukouletsos, K., Larissi, I.K., Moustris, K.P., 2014. Artificial neural networks modeling for forecasting and the maximum daily total precipitation at Athens, Greece, Atmospheric Research, 144, 141 – 150.

Pachori, Ram Bilas. 2008. "Discrimination between Ictal and Seizure-Free EEG Signals Using Empirical Mode Decomposition." Research Letters in Signal Processing 2008: 1–6. https://doi.org/10.1155/2008/293056.

Palmer, W.C. in U.S. Weather Bureau, Res. Pap., 45, 1965.

Piechota, T.C, Dracup, J.A., Fovell, R.G., 1997. Western US streamflow and atmospheric circulation patterns during El Niño-Southern Oscillation, Journal of Hydrology, 201(1), 249-271, ISSN 0022-1694, http://dx.doi.org/10.1016/S0022-1694(97)00043-7. (http://www.sciencedirect.com/science/article/pii/S0022169497000437) Keywords: Western US streamflow; Principal component analysis (PCA); Cluster analysis; Jackknife analysis; Atmospheric circulation patterns

Piechota, T.C., Dracup, J.A., 1996. Drought and Regional Hydrologic Variation in the United States: Associations with the El Niño-Southern Oscillation, Water Resour. Res., 32(5), 1359–1373.

Pigorini, Andrea, Adenauer G. Casali, Silvia Casarotto, Fabio Ferrarelli, Giuseppe Baselli, Maurizio Mariotti, Marcello Massimini, and Mario Rosanova. 2011. "Time– frequency spectral analysis of TMS-evoked EEG oscillations by means of Hilbert–Huang transform." Journal of Neuroscience Methods 198 (2): 236–45. https://doi.org/10.1016/j.jneumeth.2011.04.013.

Rasmusson, Eugene M., and Phillip A. Arkin. 1985. "Chapter 40 Interannual climate variability associated with the El Niño/ Southern Oscillation." In, 697–725. https://doi.org/10.1016/S0422-9894(08)70736-0.

Richman, M., Leslie, L., 2015. Uniqueness and Causes of the California Drought, Procedia Computer Science, 6, 428-435.

Robeson, S.M., 2015. Revisiting the recent California drought as an extreme value, Geophys. Res. Lett., 42 (16), 6771 – 6779.

Ropelewski, C. F., and P. D. Jones. 1987. "An Extension of the Tahiti–Darwin Southern Oscillation Index." Monthly Weather Review 115 (9): 2161–5. https://doi.org/10.1175/1520-0493(1987)115%3C2161:AEOTTS%3E2.0.CO;2.

Sakellariou, N. K., Kambezidis, H.D., 2004. Prediction of the total rainfall amount during August and November in the Athens Area, Greece, Fresenius Environ. Bull, 13 (3), 289 – 292.

Salih, A. A. M., Elagib, N. A., Tjernstrom, M., Zhang, Q. 2018. “Characterization of the Sahelian-Sudan rainfail based on observations and regional climate models.” Atmospheric Research, 202 (4). https://doi.org/10.1016/j.atmosres.2017.12.001

Savtchenko, A.K., Huffman, G., Vollmer, B., 2015. Assessment of precipitation anomalies in California using TRMM and MERRA data, J. of Geophys. Res.: Atmospheres, 120 (16), 8206 – 8215, doi:10.1002/2015JD023573.

Shukla S., Safeeq, M., AghaKouchak, A., Guan, K., Funk, C., 2015. Temperature impacts on the water year 2014 drought in California, Geophys. Res. Let., 42 (11).

Siam, M. S., and E. A. B. Eltahir. 2015. “Explaining and forecasting interannual variability in the flow of the Nile River.” Hydrology and Earth System Sciences 19 (3): 1181–92. https://doi.org/10.5194/hess-19-1181-2015.

Silverman, D., Dracup, J.A., 2000. Artificial neural networks and long-range precipitation predictions in California, J. Appl. Meterol, 39 (1), 57 – 66.

Slemr, F., Brenninkmeijer, C.A., Rauthe-Schöch, A., Weigelt, A., Ebinghaus, R., Brunke, E.-G., Martin, L., Spain, T.G., O’Doherty, S., 2016. El Niño–Southern Oscillation influence on tropospheric mercury concentrations, Geophys. Res. Lett., 43, doi:10.1002/2016GL067949.

Slemr, Franz, Carl A. Brenninkmeijer, Armin Rauthe-Schöch, Andreas Weigelt, Ralf Ebinghaus, Ernst-Günther Brunke, Lynwill Martin, T. Gerard Spain, and Simon O‘Doherty. 2016. “El-Niño Southern Oscillation (ENSO) influence on tropospheric mercury concentrations.” Geophysical Research Letters, n/a–n/a. https://doi.org/10.1002/2016GL067949.

Srivastava N., Hinton, G., Krizhevsky, A., 2014. Dropout – A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research, 15, 1929 – 1958.

Szép I.J., Mika, J., Dunkel, Z., 2005. Palmer drought severity index as soil moisture indicator: physical interpretation, statistical behaviour and relation to global climate, Physics and Chemistry of the Earth, Parts A/B/C, 30, 1-3, 231-243, ISSN 1474-7065.

Thomas, Evan A., Joseph Needoba, Doris Kaberia, John Butterworth, Emily C. Adams, Phoebe Oduor, Denis Macharia, Faith Mitheu, Robinson Mugo, and Corey Nagel. 2019.

Van Den Dool, H. M., S. Saha, and Å Johansson. 2000. "Empirical orthogonal teleconnections." Journal of Climate 13 (8): 1421–35. https://doi.org/10.1175/15200442(2000)013%3C1421:EOT%3E2.0.CO;2.

Vose, R.S., Applequist, S., Squires, M., Durre, I., Menne, M.J., Williams Jr., C.N., Fenimore C., Gleason, K., Arndt, D., 2014. Improved historical temperature and precipitation time series for U.S. climate divisions. J. Appl. Meteor. Climatol., 53, 1232–1251

Wang, Guiling, and Elfatih A. B. Eltahir. 1999. "Use of ENSO information in medium- and long-range forecasting of the Nile floods." Journal of Climate 12 (6): 1726–37. https://doi.org/10.1175/1520-0442(1999)012%3C1726:UOEIIM%3E2.0.CO;2.

Werbos, P. J., 1990. Backpropagation through time: what it does and how to do it, Proceedings of the IEEE, 78 (10), 1550-1560, doi: 10.1109/5.58337

Wood, F., Archambeau, C., Gasthaus, J., James, L., Teh, Y.W., 2009. A stochastic memoizer for sequence data, Proceedings of the 26th Annual International Conference on Machine Learning, 1129–1136.

Zaroug, M. A. H., E. A. B. Eltahir, and F. Giorgi. 2014. "Droughts and floods over the upper catchment of the Blue Nile and their connections to the timing of El Niño and la Niña events." Hydrology and Earth System Sciences 18 (3): 1239–49. https://doi.org/10.5194/hess-18-1239-2014.

Zhenya, S., Qi, S., Ying, B., Xunqiang, Y., Fangli, Q., 2015. The prediction on the 2015/16 El Niño event from the perspective of FIO-ESM, Acta Oceanologica Sinica, 34(12), 67–71, doi: 10.1007/s13131-015-0787-4.

Zur, R.M., Jiang, Y., Pesce, L.L., Drukker, K., 2009. Noise injection for training artificial neural networks: A comparison with weight decay and early stopping. Medical Physics, 36(10), 4810–4818. http://doi.org/10.1118/1.3213517.

Zvarevashe, Willard, Symala Krishnannair, and Venkataraman Sivakumar. 2019. "Analysis of Rainfall and Temperature Data Using Ensemble Empirical Mode Decomposition." Data Science Journal 18 (1): 46. https://doi.org/10.5334/dsj-2019-046.