Scientific Research Publishing

Atmospheric and Climate Sciences, 2024, 14, 450-473
https://www.scirp.org/journal/acs
ISSN Online: 2160-0422
ISSN Print: 2160-0414

Wavelet Multiview-Based Hybrid Deep Learning Model for Forecasting El Niño-Southern Oscillation Cycles

Winston Zhou, Xiaodi Wang

The Mathematics Department, Western Connecticut State University, Danbury, USA
Email: winston666zhou@hotmail.com, wangx@wcsu.edu

How to cite this paper: Zhou, W. and Wang, X.D. (2024) Wavelet Multiview-Based Hybrid Deep Learning Model for Forecasting El Niño-Southern Oscillation Cycles. *Atmospheric and Climate Sciences*, **14**, 450-473.
https://doi.org/10.4236/acs.2024.144027

Received: September 9, 2024
Accepted: October 12, 2024
Published: October 15, 2024

Abstract

The El Niño-Southern Oscillation (ENSO) is a significant climate phenomenon with far-reaching impacts on global weather patterns, ecosystems, and economies. This study aims to enhance ENSO forecasting with the Extended Reconstruction Sea Surface Temperature v5 (ERSSTv5) climate model. The M-band discrete wavelet transforms (DWT) are utilized to capture multi-scale temporal and spatial features effectively. Long-short term memory (LSTM) autoencoders are also used to capture significant spatial and temporal patterns in sea surface temperature (SST) anomaly data. Deep learning techniques such as the convolutional neural networks (CNN) are used with non-image and image time series data. We also employ parallel computing in a various support vector regression (SVR) approximators to enhance accuracy. Preliminary results indicate that this hybrid model effectively identifies key precursors and patterns associated with El Niño events, surpassing traditional forecasting methods. Results of the hybrid model produce a correlation of 0.93 in 4-month lagged forecasting of the Oceanic Niño Index (ONI)—indicative of high success rate of the model. Future work will focus on evaluating the model's performance using additional reanalysis datasets and other methods of deep learning to further refine its robustness and applicability. We propose wavelet-based deep learning models which have potential to shine a light on achieving United Nations' 2030 Agenda for Sustainable Development's goal 13: "Climate Action", as an innovation with potential in improving time series image forecasting in all fields.

Keywords

El Niño-Southern Oscillation (ENSO), Autoencoders, Discrete Wavelet Transform (DWT), Convolutional Neural Network (CNN), Support Vector Regression (SVR)

1. Introduction

1.1. Background

The El Niño-Southern Oscillation (ENSO) is a critical climate phenomenon that drives significant variability in global weather patterns, primarily through its influence on sea surface temperatures (SST) and atmospheric pressure across the Tropical Pacific Ocean.

ENSO cycles consist of three different phases: El Niño, La Niña, and the neutral phase, each associated with distinct meteorological effects ([1]). El Niño events are mainly associated with a strong positive SST anomaly in the equatorial Pacific, being the reversal of Walker Circulation—the eastward movement pathways seen in normal conditions. As the Pacific Equatorial Counter Current strengthens, warmed water piles in the East Pacific. Rising parcels of warm, moist, less dense air are formed, resulting in a relative low-pressure system. Accumulation of moisture creates heavy cloud cover—leading to increased rainfall and flooding off the West coast of South America in regions such as Peru and Ecuador. Across the Pacific, severe droughts can occur in Australia, Southeast Asia, and Southern Africa, areas where there is less warm water to supplement rainfall ([2] [3]). Conversely, La Niña, with cooler SST anomalies, strengthens Walker circulation, with intensified circulation eastwards—producing drier conditions in the eastern Pacific and increased rainfall in the western Pacific, often resulting in significant floods in countries like Indonesia and the Philippines ([4]).

The United Nations 2023 Climate Change Conference (COP 28) includes a section for "Enhancing global efforts to strengthen resilience". The Global Goal on Adaption (GGA) notes that preparation for changes in areas including water, food, health, ecosystems, infrastructure, poverty eradication and cultural heritage ([5]) are to be improved for the future in all countries. We believe that the techniques developed in this study may advance efforts in the GGA through greater accuracy of short-term ENSO prediction.

The 1997-1998 El Niño event, one of the strongest recorded, is a particularly strong example of the widespread impacts of ENSO's meteorological and oceanographic effects. The intense positive SST anomaly resulted in unusually high rainfall in coastal regions of South America, particularly in Peru and Ecuador, leading to catastrophic flooding. Simultaneously, the near reversal of atmospheric circulation patterns suppressed upwards convection and thus rainfall in the Western Pacific and Southeast Asia, leading to severe droughts in countries such as Indonesia and Malaysia. These drought conditions caused various other detrimental effects, such as widespread forest fires, which created a severe episode of transboundary haze pollution that greatly impacted public health and caused significant economic damage overall ([6]). Australia experienced some of its most severe droughts on record, with rainfall deficits exceeding a massive 50% below normal in some areas, resulting in a significant drop in agricultural output as well as exacerbated water scarcity in said regions ([7]). This single event has led to over 2000 deaths and has affected approximately 400,000 people ([7]), numbers that could be greatly lessened

with proper forecasting and thus suitable amounts of time for preparation.

The socio-economic ramifications of El Niño are similarly significant, impacting various fields such as agriculture, health, and infrastructure. The 1997-1998 El Niño event led to an estimated $33 billion in economic damages globally, with losses in agriculture and infrastructure being particularly severe ([7]). The increased frequency of vector-borne diseases, such as malaria and dengue fever, during El Niño events further exacerbates public health challenges, particularly in tropical regions ([7]). Such impacts highlight the need for accurate predictive models to help mitigate the adverse effects of strong ENSO events and support effective response strategies.

During the 2010-2011 La Niña, Australia faced widespread flooding that led to $5.6 billion in damages and impacted over 200,000 people ([7]). La Niña can influence atmospheric chemistry by altering the distribution of trace gases, such as increased uptake of carbon dioxide by enhanced phytoplankton activity in the cooler waters ([8]). Enhanced phytoplankton growth can disrupt marine ecosystems by causing harmful algal blooms, which deplete oxygen and create dead zones detrimental to marine life, thus being a major threat to fisheries in zones of interest. Additionally, while lower atmospheric CO_2 may reduce short-term greenhouse gas effects, the disruption of oceanic carbon cycling can affect long-term climate stability and biodiversity, changes that require pre-emptive measures to account for in maintaining global economies and overall public health.

Understanding and predicting the meteorological impacts of ENSO events, such as those observed during the 1997-1998 El Niño, are essential for mitigating their adverse effects on vulnerable populations and ecosystems. However, current forecasting methods still face challenges in accurately predicting the onset, duration, and intensity of ENSO events. This research seeks to address these challenges by integrating advanced climate models with deep learning techniques, enhancing the precision and reliability of ENSO forecasts.

The objectives of this study were to enhance ENSO forecasting accuracy by integrating state of art mathematical tools and deep learning techniques to evaluate their effectiveness in processing climate data. Specifically, usage of 1) the M-band discrete wavelet transform (DWT) as a pre-processing step in facilitating multi-scale feature extraction is expected to significantly improve model accuracy. When combined with 2) parallel computing to incorporate all wavelet multi-view components in a regression, is likewise expected to improve quality of predictions. 3) Encoder-decoder autoencoders were applied to process and reconstruct image time series data, preserving spatial information, where differing performance in usage of multiple nodes and non-image components was to be found.

1.2. Related Works

In the context of ENSO prediction, two main types of models are commonly used: dynamic models and statistical models. Dynamic models are based on the physical

equations governing the atmosphere and oceans, integrating variables such as SST, wind, and pressure to simulate future climate conditions. These models are detailed and simulate complex interactions within the climate system but require significant computational resources. In contrast, statistical models rely on historical data to identify patterns and correlations that can be used to predict future ENSO events. These models, including machine learning and neural networks, are often less resource-intensive and can see patterns in data that may not be capable of being modeled in dynamic models, though they may lack the detailed physical realism of dynamic models ([9]).

Understanding and predicting the El Niño-Southern Oscillation (ENSO) has been the focus of many studies due to its significant impact on global climate patterns. A foundational work by Ham *et al.* defined ENSO's characteristics and its variability, which provided a framework for subsequent research aimed at enhancing prediction accuracy ([10]). Historically, early studies predominantly relied on relatively simple statistical and less complex dynamical models, which, while useful, often struggled with capturing the complex, non-linear dynamics of ENSO ([2] [11]). These limitations have led to the integration of more sophisticated methodologies, such as machine learning and deep learning techniques.

Building on the advancements in deep learning, LSTM networks and CNNs have become essential tools in forecasting and time series analysis, particularly in the field of climate science. LSTMs are well-suited for capturing long-term dependencies and trends in sequential data, which is crucial for understanding trends over a period of time in a phenomenon like ENSO cycles ([10]). CNNs excel at extracting spatial features from image data, making them effective in identifying patterns in a time-series gridded data such as SST anomalies across the globe. When combined, these architectures have been shown to significantly improve the accuracy of forecasts by leveraging both temporal and spatial information ([11]).

Recent advances in deep learning have shown promise in improving short-term ENSO forecasting. Ham *et al.* (2019) introduced a deep learning framework that leveraged convolutional neural networks (CNNs) to capture spatial patterns and long short-term memory (LSTM) networks to model temporal dependencies in climate data [10]. This approach demonstrated enhanced predictive performance compared to traditional methods, highlighting the potential of deep learning in this field. Building on this, other researchers have explored hybrid models that combine multiple deep learning architectures to better capture the multiscale nature of climate data.

In addition to neural network model architecture innovations, the use of other techniques has been explored in order to increase robustness of predictions. Given ENSO's immensely chaotic nature, a meticulous pre-processing procedure is immensely beneficial to optimizing the model's ability to capture the underlying patterns causing the fluctuations in ONI values without accidentally picking up trends that were the results of external factors.

The discrete wavelet transform (DWT) serves as not only a key preprocessing step for de-noising but simultaneously offers the benefit of being multi-view,

presenting multiple different channels (components) holding potentially more information and trends that may increase accuracy of ENSO prediction ([12]).

Traditionally, the DWT was mainly used in denoising, where the approximation component—being the denoised version of the original image—would be used in on itself for machine learning; the detail components hold less relevant information and pick up on less trends compared to the approximation ([13]). In this study however, a parallel computing approach was used to integrate the main approximation channel with the information held in each of the detail components generated by the wavelet transform to further improve robustness of the final predictor model using forms of multiple regression analysis tools.

Autoencoders have been increasingly explored in the context of unsupervised learning for their ability to reproduce data while retaining and possibly amplifying essential features. Typically, autoencoders are employed in tasks where the goal is to compress data into a lower-dimensional space and then reconstruct it, which is particularly useful in handling large, complex datasets. However, prior studies have predominantly focused on dimensionality reduction of image datasets and where decoded back to original dimensions only—which were then used in the neural network ([14]). This research leverages encoder-decoder autoencoders to not only reduce dimensionality for image encoded data, but non-image data as well, which enhances the model's ability to pick up generalized key features to a higher extent. Such a reconstruction method allows for the conjunction with CNNs and LSTMs while simultaneously utilizing the denoising and anomaly-detection abilities of autoencoders to further enhance ENSO prediction accuracy.

2. Data and Methods

2.1. Data Selection and Preparation

Seen in the full project workflow, **Figure 1**, this study utilized the Extended Reconstructed Sea Surface Temperature version 5 (ERSSTv5) dataset, which provides global sea surface temperature (SST) anomalies. ERSSTv5 is a widely used dataset known for its accuracy in long-term climate studies, offering monthly SST data with a spatial resolution of $2^\circ \times 2^\circ$ from 1854 to present ([15]). This study used a time range of January 1949 to December 2014, of length 792 months. This dataset was chosen due to its comprehensive coverage and its relevance widespread in ENSO—related climate modeling studies and analyzing SST anomalies, which are critical in understanding and predicting ENSO events.

To ensure quality analysis, the image data was centered around the most relevant area for ENSO-related activity, the tropical Pacific region from the ERSSTv5 dataset. This region is crucial for detecting and understanding the sea surface temperature (SST) anomalies that are indicative of El Niño and La Niña events, as it encompasses the key zones where these phenomena typically manifest to the greatest extent ([15], **Figure 2**). This focus allowed for a more detailed and accurate analysis of the SST patterns associated with ENSO variability.

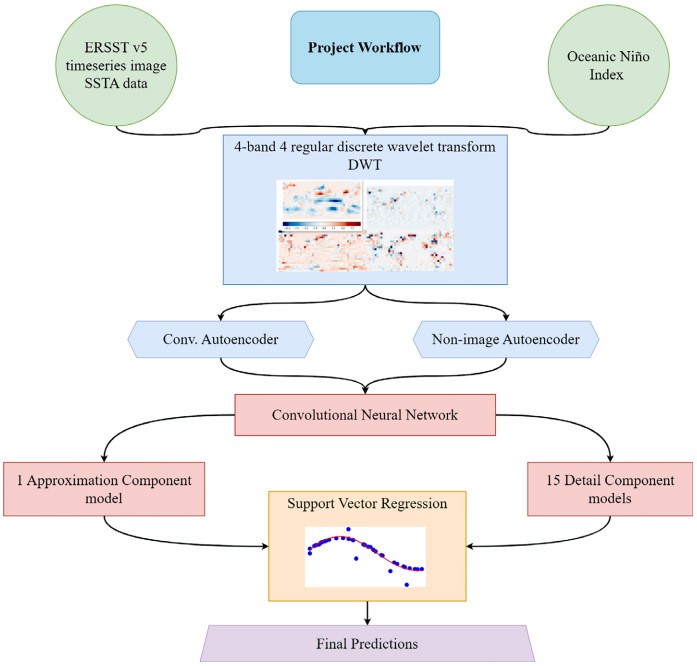


Figure 1. Project workflow.

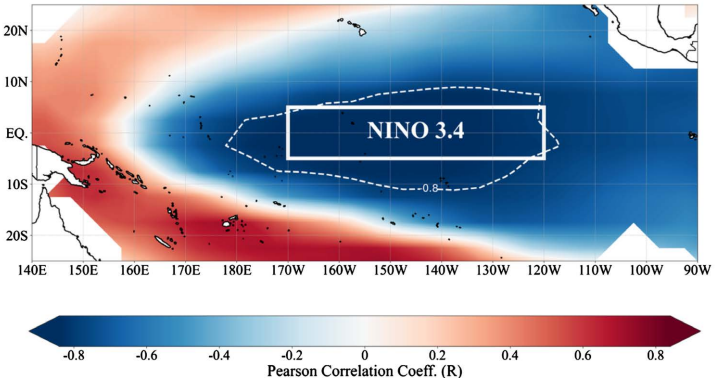


Figure 2. Pearson correlation coefficients between sea surface temperature (SST) anomalies and El Niño phases, highlighting significant correlations within the Niño 3.4 region (boxed area). The correlation map indicates strong correlations in the central equatorial Pacific, consistent with the expected warming or cooling patterns associated with ENSO events.

The SST anomaly was then standardized using scikit-learn's StandardScalar, which normalizes feature values by subtracting the sample mean and dividing by the sample standard deviation:

$$z_i = \frac{x_i - m}{s}. \tag{1}$$

In addition to SST data, the Oceanic Niño Index (ONI) was selected as a key indicator for identifying ENSO phases. The ONI is derived from SST anomalies in the Niño 3.4 region (**Figure 2**, [16]), covering the central equatorial Pacific [2]. It is the standard metric used to classify ENSO events, with thresholds set at ±0.5˚C deviations from the average. Given that the goal of the study was to accurately

forecast ENSO events, the target dataset was lagged 4 months ahead of time for the model to learn precursory patterns indicative of ensuing ENSO events.

2.2. Data Pre-Processing

2.2.1. M-Band Discrete Wavelet Transform

M-band orthogonal discrete wavelet transforms (DWT) are utilized to decompose each image in the dataset into multiple frequency bands, enhancing the model's ability to analyze various levels of detail. An M-band DWT is determined by a filter bank consisting of M filters ($M \geq 2$), including a low-pass filter α and $M-1$ high-pass filters $\beta(j)$ for $j=1,\cdots,M-1$ ([5] [14]). This decomposition allows for the extraction of both fine and coarse features, capturing multi-scale spatial and temporal characteristics essential for ENSO prediction.

In this study, the M-band DWT used prior to deep learning models to capture patterns across different scales. The multi-scale approach facilitated by the M-band DWT will later allow the use of a combined detail model with parallel computing, to overall significantly boost the computational efficiency and accuracy of the predictive model ([17]).

An M-band orthogonal wavelet is considered N-regular if it has N vanishing moments; hence wavelets with more vanishing moments tend to be smoother with longer filters.

$$2\sum_{i=1}^{n}\alpha_i = \sqrt{M} \tag{2}$$

$$\sum_{i=1}^{n} i^k \beta_i^{(j)} = 0 \quad \text{for } k=0,\cdots,N-1;\ j=1,\cdots,M-1 \tag{3}$$

$$\alpha = \beta^{(j)} = 1 \quad \text{for } j=1,\cdots,M-1 \tag{4}$$

$$\alpha\beta^{(j)} = 0 \quad \text{for } j=1,\cdots,M-1 \tag{5}$$

$$\beta^{(i)}\beta^{(j)} = 0 \quad \text{for } i,j=1,\cdots,M-1 \text{ and } i \neq j \tag{6}$$

For adequate integer values of k, an M-band wavelet can be used to create an $Mk \times Mk$ wavelet transform matrix to decompose channels of a dimension.

An example 3-band DWT matrix is

$$W = \begin{bmatrix}
\alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & \alpha_6 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & \alpha_6 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & \alpha_6 \\
\alpha_4 & \alpha_5 & \alpha_6 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha_1 & \alpha_2 & \alpha_3 \\
\beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5 & \beta_6 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5 & \beta_6 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5 & \beta_6 \\
\beta_4 & \beta_5 & \beta_6 & 0 & 0 & 0 & 0 & 0 & 0 & \beta_1 & \beta_2 & \beta_3 \\
\gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 & \gamma_5 & \gamma_6 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 & \gamma_5 & \gamma_6 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 & \gamma_5 & \gamma_6 \\
\gamma_4 & \gamma_5 & \gamma_6 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma_1 & \gamma_2 & \gamma_3
\end{bmatrix}$$

The filters for said matrix being

$$\alpha = \begin{bmatrix} 0.3384 \\ 0.5308 \\ 0.7233 \\ 0.2390 \\ 0.0465 \\ -0.1459 \end{bmatrix}, \quad \beta = \begin{bmatrix} -0.1174 \\ 0.5443 \\ -0.0187 \\ -0.6991 \\ -0.1361 \\ 0.4270 \end{bmatrix}, \quad \gamma = \begin{bmatrix} 0.4036 \\ -0.6285 \\ 0.4606 \\ -0.4036 \\ -0.0786 \\ 0.2465 \end{bmatrix}.$$

It can be proven that all the DWT matrices we are using are orthogonal, *i.e.*, $W = W$ for all such DWT matrices W([18]).

The 2D DWT of an channel X is defined as the matrix WXW. This decomposition can be seen as an $M \times M$ block matrix where the top-left submatrix is the approximation (low frequency) component, and the other submatrices are detail (high frequency) components.

For example, a 2D 3-band DWT decomposes a channel X into a block matrix

$$WXW = \begin{bmatrix} A & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix}$$

where A is the approximation, or low frequency, component and D_{ij} are detail, or high frequency, components ([19]).

Figure 3 is a 88 × 180 (clipped for multiples of 4) pixel image of the mean SST anomalies of the world under Extended Reconstructed Sea Surface Temperatures Version 5 in October of 1949.

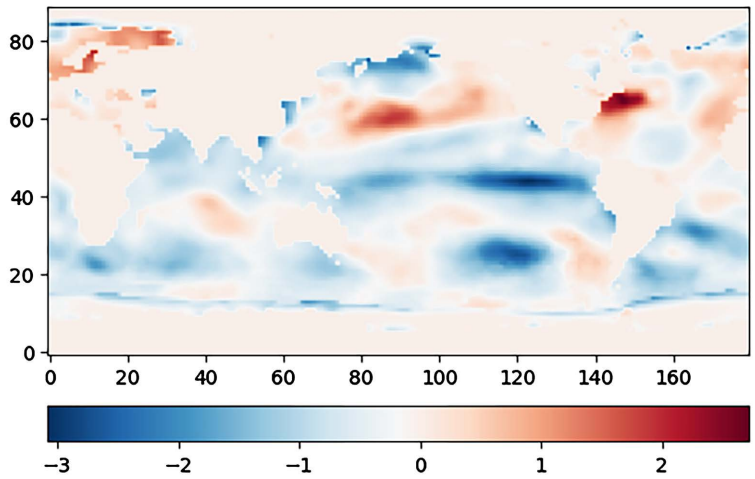
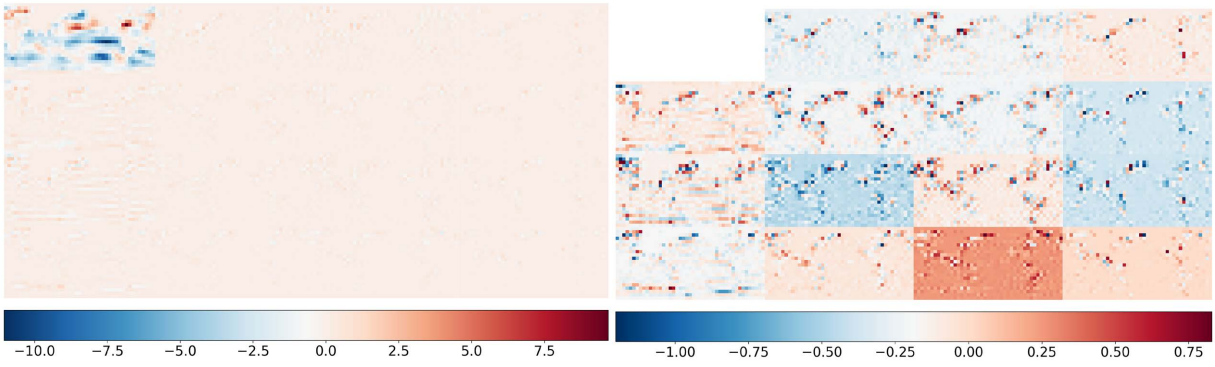


Figure 3. Original image of SST anomalies—observe the presence of a cold tongue in the Central Equatorial Pacific.

To pre-process the data, a 4-band 4-regular wavelet filter bank derived from [18] was used across each image of the 792 months in the SST anomaly time series dataset. As such, an approximation and 15 detail components were generated, each now of pixel size 22 × 45 (Figure 4).



(a) Approximation component (b) Detail components, scaled colors

Figure 4. Results of wavelet decomposition using the 4-band 4-regular wavelet. The cold tongue in the Tropical Pacific in the approximation (top left, subfigure 4(a)) remains noticeable as seen in the original, **Figure 3**. Because we applied a 4-band DWT to the image, the image is decomposed into 16 sub-images. The pixel values of detail components are very small compared to the approximation, so the subfigure 4(b) is the same as the subfigure 4(a) but with a smaller color gradient.

In addition to its usage in pre-processing the image data, the DWT was also applied to the target ONI index dataset for the purpose of denoising ([20]).

Given a signal $S \in \mathbb{R}^n$,

W = 4-band 4-regular wavelet filter bank:

For each detail coefficient $d_i\,(i=1,2,3)$:

$$\tilde{S} = W \times S = \begin{pmatrix} a \\ d_1 \\ d_2 \\ d_3 \end{pmatrix} \tag{7}$$

$$a, d_1, d_2, d_3 \in \mathbb{R}^{\frac{n}{4}}$$

$$\text{Threshold } \delta_i = \sigma_i \sqrt{2\log\left(\frac{n}{4}\right)} \quad \text{for } i = 1,2,3 \tag{8}$$

where σ_i is the standard deviation of the elements of d_i.

$$d_i^* = \begin{pmatrix} d_{11}^* \\ \vdots \\ d_{1\frac{n}{4}}^* \end{pmatrix} \text{ where } d_i^* = \begin{cases} d_{ij}, & \text{if } |d_{ij}| \geq \delta_i, \\ 0, & \text{if } |d_{ij}| < \delta_i \end{cases} \text{ for } i = 1,2,3 \text{ and } j = 1,\cdots,\frac{n}{4}. \tag{9}$$

$$\tilde{S}^* = \begin{pmatrix} a \\ d_1^* \\ d_2^* \\ d_3^* \end{pmatrix} \tag{10}$$

$$S^* = W^t \times \tilde{S}^* \quad (\text{where } S^* \text{ is the denoised signal}) \tag{11}$$

This approach was beneficial for removing any unexpected anomalies in the data that may have been caused by fluctuations in various other variables in the global climate—seen in **Figure 5**.

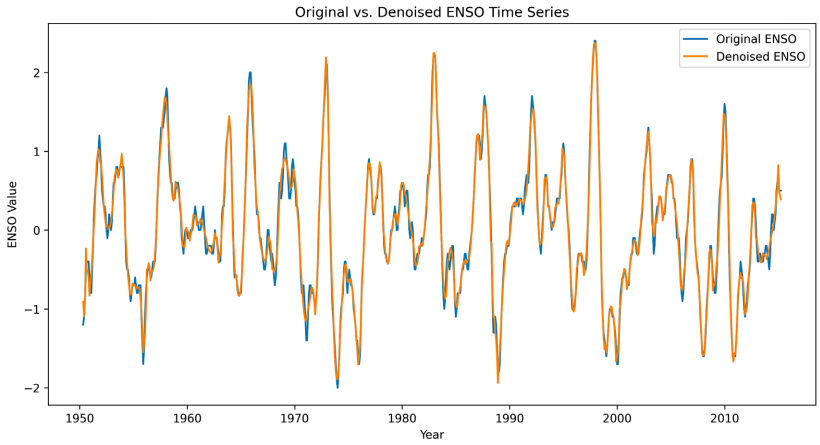


Figure 5. Original ONI index vs. DWT—denoised ONI index from April 1950 to April 2015.

2.2.2. Non-Image Autoencoders

Autoencoders, particularly those paired with LSTM networks, are effective for dimensionality reduction in time-series data to capture critical patterns stored in the SST anomaly data. Autoencoders will output multiple nodes or channels based on trends observed in the original dataset, each with different configurations that the autoencoder decides will best match information patterns observed in the original dataset ([21]). By selecting the nodes with highest Pearson correlation to the target dataset, training data can be further denoised and optimized for retaining the most ENSO-correlated patterns in the SST anomaly dataset.

This study utilized a hybrid model, with both original and flattened versions of the time-series image data to capture both spatial and generalized trends, respectively—which were then used through a concatenate layer in the CNN.

In traditional autoencoder architecture, the encoder compresses the input data into a lower-dimensional representation (latent space), capturing essential features, while the LSTM network learns temporal patterns. This compressed representation is then passed through a decoder, which reconstructs the data back to its original form, preserving the temporal sequences and features.

This method is particularly valuable in applications where feature extraction from high-dimensional time-series data is very useful, as the non-image data will capture key generalized patterns associated with ENSO trends, hence flattened input data was used to create multiple nodes of encoded data in the form of vectors. Reconstruction from the decoder thus only maintains the temporal dimensions as the final output nodes are all in vector form.

2.2.3. Convolutional Autoencoders

Convolutional autoencoders, designed for image data, combine convolutional layers with autoencoder architecture to perform both dimensionality reduction and data reconstruction while preserving spatial information.

In this setup, instead of flattening the input image data, the encoder directly applies convolutional filters to the input image, extracting features and reducing

the spatial dimensions (Figure 6). The decoder then reconstructs the image from the compressed representation, maintaining the same spatial dimensions as the original time series image dataset. This technique is particularly effective for processing and denoising image sequences, allowing the model to learn more highly ENSO-correlated particular spatial features for time-series forecasting.

In our case, 128 channels were reconstructed from the non-image autoencoder, and 4 channels from the convolutional autoencoder. Each was trained with 200 epochs and earlystopping to prevent overfitting.

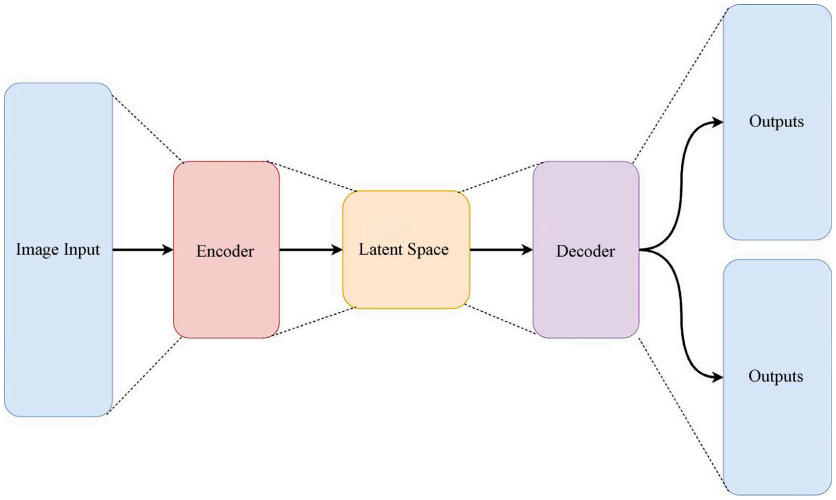


Figure 6. Autoencoder schematic for both non-image and convolutional image autoencoder architecture. The input layer represents the original time series data, where each time step corresponds to either a feature vector (flattened) or an image. The decoder reconstructs the input back to its original dimensions N in multiple different channels/nodes in both the non-image autoencoder and in the image autoencoder.

2.3. Model Framework

Convolutional Neural Networks

Neural Networks are composed of three layers—the Input layer, hidden layers, and output layers. Each are made of neurons, which receive an input from the previous layer and perform a weighted sum with trainable weights w and adds a trainable bias value b_i. Finally, the neuron applies a specified activation function f to produce an output value z_n, which serves as the input for the neurons in the following layer. Such layers that receive and produce an input are called Dense layers. All outputs can be defined by these two equations:

$$y_i = b_i + \sum_{j=1}^{n} w_j x_j, \tag{12}$$

$$z_i = f(y_i). \tag{13}$$

A Convolutional Neural Network (CNN) is a type of neural network that is often used in the case of learning features from image input data ([22]). For an input variable x consisting of c channels, each $P \times P$ pixels, the CNN applies N convolutional filters, each of size $Q \times Q$. The output $y_{i,j,n}$ at each layer is

calculated as:

$$y_{i,j,n} = \sum_{s=0}^{Q-1}\sum_{t=0}^{Q-1} w_{s,t,n} \cdot x_{i+s,j+t} + b_n. \tag{14}$$

It can be seen that the initial input layer has dimensions of (None, 12, 8, 20, 4) in the model framework, **Figure 7**. The dimension 12 arises from the usage of sequences. Prior to model training, the dataset was reformatted into a dataframe-like sequence, such that there are moving windows of 12 months at a time (Jan 1949 to Dec 1949, Feb 1949 to Jan 1951, etc.), each correlating to a predicted ONI value 4 months ahead of time. This procedure was carried out for both the image and non-image data.

The model framework begins with processing image data through time-distributed convolutional layers. These layers allow the model to capture spatial features independently for each time step, an important method when handling sequential data as it preserves temporal information necessary for making predictions based on a series of inputs. The Rectified Linear Unit (ReLU) activation function is employed in these convolutional layers to introduce non-linearity, allowing the model to handle complex patterns in the data. Specifically, the ReLU function filters out negative values using the equation:

$$f(x) = \max(x, 0) \tag{15}$$

This function outputs the maximum between the input value and zero, effectively retaining positive values and discarding negative ones, which helps the model learn any relevant features.

Such a manner of processing is necessary as the model will be able to learn much more information for a prediction if inputted 12 months of data at a once compared to just one. Following the convolutional operations, max-pooling layers reduce the dimensionality while retaining key features for following layers.

After the convolution and pooling steps, the downsampled image data is passed through a Convolutional Long Short-Term Memory (ConvLSTM2D) layer ([23]). This layer is critical for capturing both spatial and temporal dependencies in the data, which is especially important for time-series prediction tasks. The ConvLSTM2D layer uses trainable weights within a 3 × 3 convolutional window and applies the hyperbolic tangent (tanh) activation function, which scales input values to a range between −1 and 1. The tanh activation function is particularly suited for the ConvLSTM layer as it helps keep the temporal dependencies needed across sequences while also controlling the signal amplitude.

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{16}$$

Concurrently, the non-image data, representing the sequence of features correlated with the target dataset, is processed through a dense layer. This dense output is reshaped and repeated to align with the dimensions of the image features before being concatenated with the pooled image features, combining the processed image and nonimage output before the final dense and dropout layers.

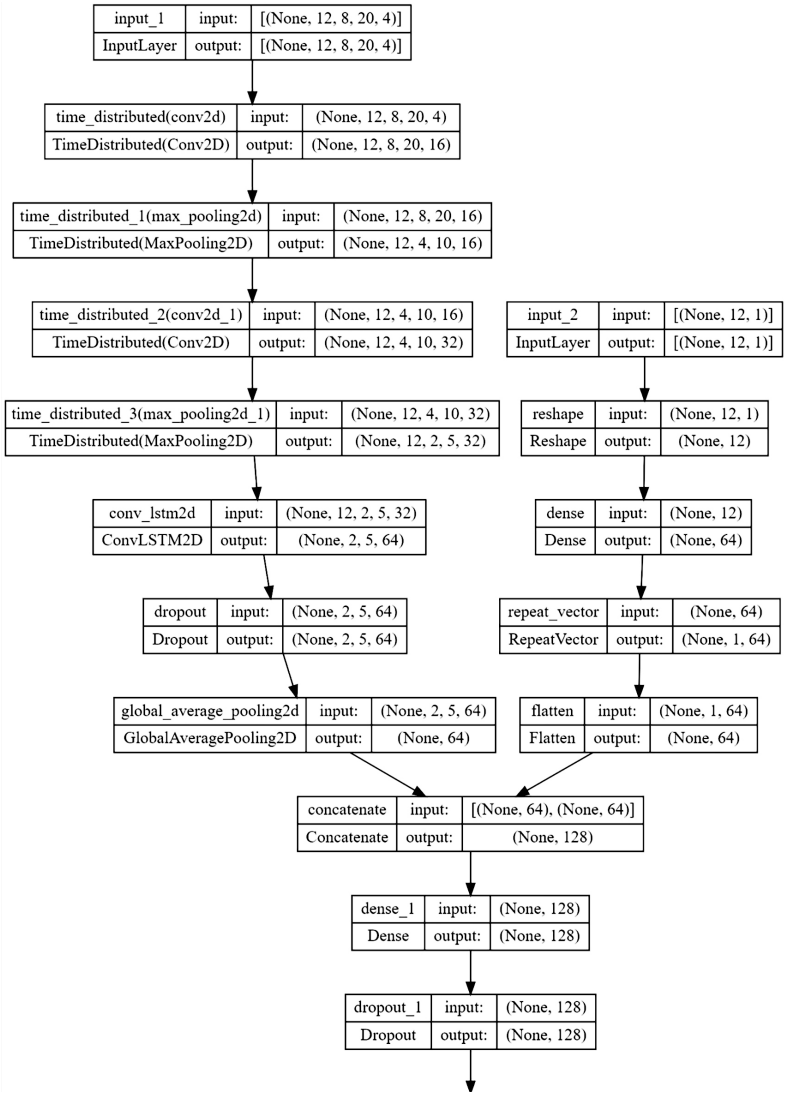


Figure 7. Flow chart of layers included in the full Convolutional Neural Network. The left column of layers is for the image data, the right column is for processing the nonimage data. They are concatenated before final dense layers to output a single forecasted value. Input dimensions of (None, 12, 8, 20, 4) are for time steps, sequence windows, height, width, and channels, respectively.

Dropout regularization is applied to these dense layers to prevent overfitting. By randomly dropping a fraction of the neurons during training, the dropout regularization helps ensure that the model does not become too dependent on any single feature, improving model precision. The dropout rate is set to 0.2 in order to maintain a balance between reducing overfitting and retaining model complexity ([24]). The final output is then obtained through a softmax activation function in the Dense layer, which transforms the combined features into a single predicted value.

2.4. Support Vector Regression

Recall that while the detail components of the discrete wavelet transform are generally not used, they are still components of the original dataset, and hence hold

potentially useful information for further optimizing the model. In this study, the focus shifted to refining the final predictive model using a parallel computing approach with multiple regression analysis tools, specifically the Support Vector Regression (SVR) ([25]). After the wavelet transform decomposed the signal into approximation and detail components, the regression step was crucial for combining these components into a unified predictive model.

Support Vector Regression (SVR) was chosen for its high capacity in managing high-dimensional and complex data. SVR aims to approximate the underlying patterns in the data while minimizing prediction errors. A distinguishing feature of SVR is its margin of tolerance, defined by the ϵ -insensitive loss function. This margin allows the model to disregard small deviations within a specified range (ϵ), focusing on more significant errors that exceed this threshold. This approach enhances the final model's ability to handle noisy data and outliers effectively ([26]).

Given the already relatively well-performing model from just the wavelet approximation component of the dataset, we decided to optimize the regression to fit to the residuals between the true and approximation-predicted values:

$$r(t) = y_{\text{true}}(t) - y_0(t) \tag{17}$$

To optimize these residuals, SVR was applied to the detail components $D_j(t)$ for $j = 1, 2, \cdots, 15$. The SVR model was used to predict the residual $r(t)$ based on a linear combination of the detail components:

$$\hat{r}(t) = \sum_{j=1}^{15} w_j D_j(t)$$

Here, w_j are the weights learned by the SVR model for each detail component $D_j(t)$. The final predictor $\hat{y}(t)$ was obtained by adding the optimized residuals back to the initial approximation:

$$\hat{y}(t) = y_0(t) + \hat{r}(t)$$

Support Vector Regression Kernels

To test out reliability of different SVR models, two different kernels were tested: the Radial Basis Function (RBF) kernel and the linear kernel.

The RBF kernel specializes in its ability to map data into a higher-dimensional space where linear relationships can be more easily identified. Such a property enhances its ability particularly in capturing complex, non-linear relationships between the residuals and the detail components, notably in cases where the data is not linearly separable ([27]). The RBF kernel is defined as:

$$K(x_i, x_j) = \exp\left(-\gamma \sum_{k=1}^{n} \left\| x_{i,k} - x_{j,k} \right\|^2\right) \tag{18}$$

where $x_{i,k}$ and $x_{j,k}$ are the *k*-th feature values of the *i*-th and *j*-th data points, respectively, and γ controls the kernel's spread. The optimization process in SVR with an RBF kernel involves adjusting these kernel parameters to fit the data. The predicted residuals $\hat{r}(t)$ in this case are indirectly influenced by the kernel parameters rather than directly learned weights.

A SVR linear kernel was also utilized, which indeed behaves similarly to

multivariate linear regression in that both methods attempt to fit a linear relationship between the input features (in this case, the detail components $D_j(t)$) and the output (the residuals $r(t)$). In both methods, the goal is to find a set of weights w_j that best fit the data.

$$K(x_i, x_j) = \sum_{k=1}^{n} x_{i,k} \cdot x_{j,k} \tag{19}$$

where $x_{i,k}$ and $x_{j,k}$ are the *k*-th feature values of the *i*-th and *j*-th data points, respectively.

In both SVR with a linear kernel and multivariate linear regression, the goal is to find a set of weights w_j that best fit the data. However, recall that SVR differs in its approach towards error minimization through a different kind of loss function that includes a margin of tolerance, allowing it to be more suited to noisy data such as SST anomalies which experience great variation from factors other than just ENSO events.

In experiments for both kernels, fitting the regressions was done with predicted values from the detail component models on the validation set, while forecast accuracy was measured using the test dataset.

2.5. Training and Testing

Given the added dimension due to sequences, a DataGenerator class was used to input the image, non-image and target datasets into the model for both training and testing, along with a "on epoch end" shuffler to reduce overfitting. Data was split into subsets of proportions 0.6, 0.2, and 0.2 for the training, validation and test sets, respectively.

It is to be noted that the image autoencoder outputs four nodes, or four different reconstructed datasets each emphasizing different information from the original SST anomaly dataset. As such, tests were run with both each of the four nodes individually (where the channel dimension in the CNN would be 1), compared to the performance seen in the CNN with a channel dimension of 4, with all nodes integrated into the datasets.

In addition, control experiments were run without the wavelet transform and the non-image autoencoded data (non-image data was set to a array of just ones) to see the magnitude of improvement gained with the non-image data, wavelet transform and SVR (seen in Table 1).

Table 1. Summary of experimental setups. Autoencoders were used for all experiments. DWT was used prior to both non-image and image autoencoders. Wavelet approximaion used in experiments 1 - 3. Models trained with 24 epochs, Earlystopping on validation loss to prevent overfitting.

Experiment	Wavelets	Non-Image Data	Channel Dimension	Regression	Description
1	No	No	1	N/A	No DWT, no non-image data

Continued

2	No	Yes	1	N/A	No DWT, with non-image data
3	Yes	Yes	1	N/A	DWT, each node separately
4	Yes	Yes	4	N/A	DWT, all nodes together
5	Yes	Yes	4	Linear	DWT, all nodes, linear regression
6	Yes	Yes	4	RBF	DWT, all nodes, RBF regression

Metrics

Metrics calculated to measure accuracy of models include: 1) Pearson correlation for evaluating how well trends are followed, 2) Root Mean Square Error (RMSE) and 3) Mean Absolute Error for error from true values, and 4) coefficient of determination (R^2) for overall performance. In addition, three other metrics were used to determine capacity of event forecasting in general ([28]):

5) Critical Success Index (CSI):

$$\mathrm{CSI} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP}}$$

CSI ranges from 0 to 1, with values closer to 1 indicating better performance. A higher CSI means that the model correctly identifies more positive instances while lesser false positives and false negatives ([29]).

6) Hit Rate (HR):

$$\mathrm{HR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

HR ranges from 0 to 1, with values closer to 1 indicating better performance. A higher HR means the model correctly identifies a greater proportion of actual positive instances.

7) False Alarm Ratio (FAR):

$$\mathrm{FAR} = \frac{\mathrm{FP}}{\mathrm{TP} + \mathrm{FP}}$$

FAR ranges from 0 to 1, with values closer to 0 indicating better performance. A lower FAR means fewer false alarms, *i.e.*, the model makes fewer incorrect positive predictions.

- True Positives (TP): Instances correctly predicted as positive.
- False Negatives (FN): Instances that are positive but incorrectly predicted as negative.
- False Positives (FP): Instances that are negative but incorrectly predicted as positive.

A ONI threshold of 0.5 was used, as such is what ([30]) classifies as what is considered a El Niño or La Niña event.

3. Results

In this section, we will review the overall performance of each of model's experiments through evaluation of metrics and predicted monthly time series (2001-12-01 to 2014-12-01).

Time Series Prediction Graphs

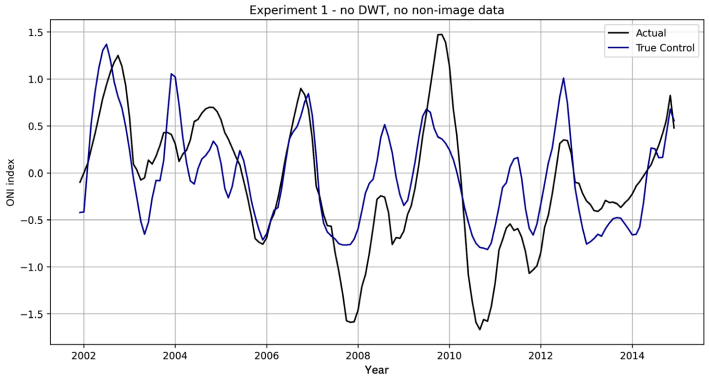


Figure 8. Experiment 1—no DWT, no non-image data, with one node. Trend is followed somewhat, however most predictions are very far off in value.

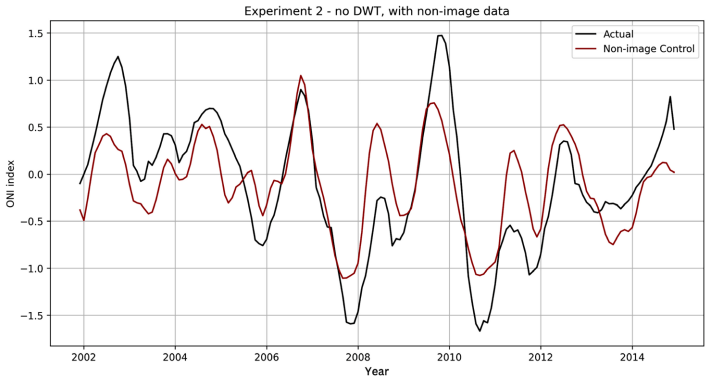


Figure 9. Experiment 2—no DWT, with non-image data, with one autoencoder node. Notable improvements from Experiment 1.

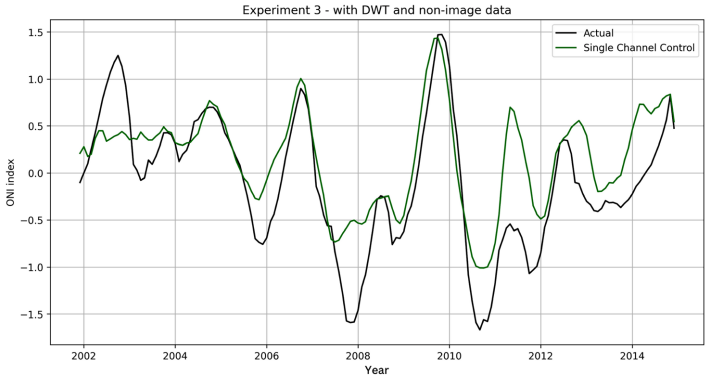


Figure 10. Experiment 3—with DWT (approximation component used) and non-image data, with a singular autoencoder node. Improvements from Experiments 1 and 2 (**Figure 8**, **Figure 9**) can be seen visually.

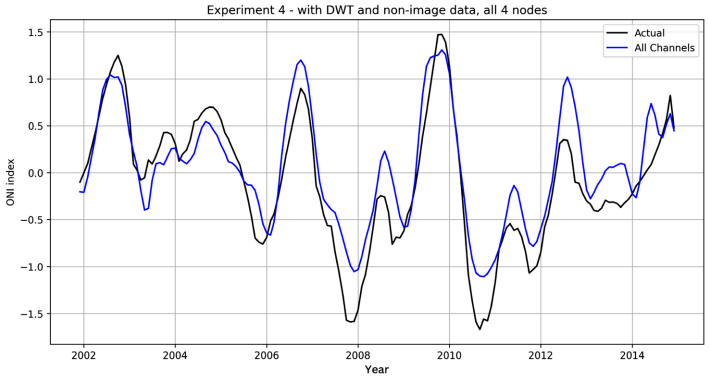


Figure 11. Experiment 4—with DWT (approximation component used) and non-image data, with all four nodes used together. Noticeable improvements from Experiments 3 (**Figure 10**), likely due usage of all nodes providing more possibly ENSO-correlated information.

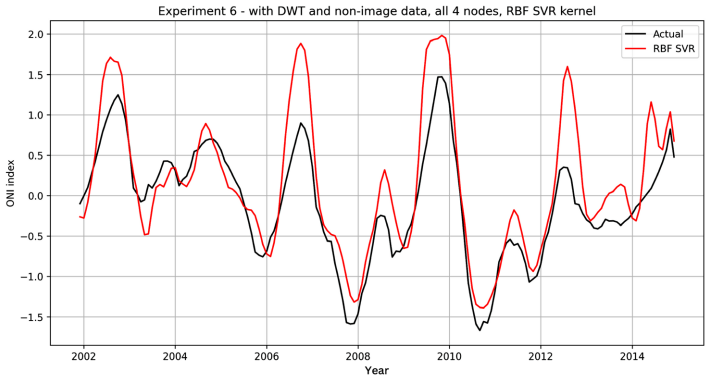


Figure 12. Experiment 6—with DWT, non-image data, with all four nodes used together, and RBF SVR applied for detail components. Not much noticeable visual improvement from (**Figure 11**).

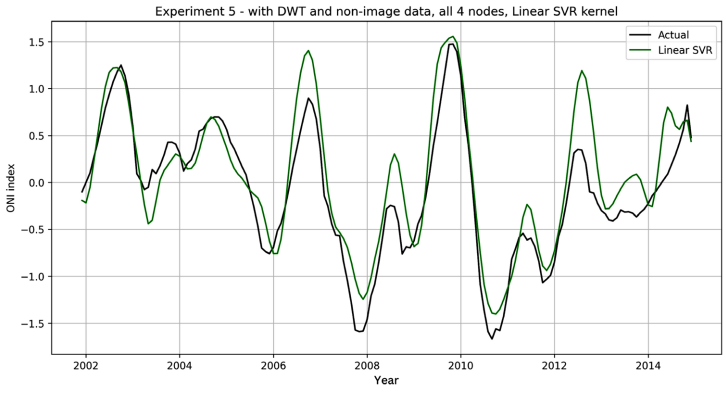


Figure 13. Experiment 5—with DWT, non-image data, with all four nodes used together, and linear SVR applied for detail components. Small improvement from Experiment 4 (**Figure 11**) can be seen, as it fits better to the actual curve.

4. Discussion

As expected, the true control group without wavelets, non-image data, or multiple

nodes performed the worst in all metrics (Figure 8). The hybrid model with non-image data greatly improved all metrics, indicating that the non-image data likely aided the model's overall perception of both trends of events and the magnitudes of ENSO events (Figure 9).

Table 2. Performance of all 6 experiments in 1) Pearson Correlation, 2) Root Mean Squared Error (RMSE), 3) Mean Absolute Error (MAE), 4) Coefficient of Determination (R^2), 5) Hit Rate (HR), 6) False Alarm Rate (FAR), 7) Critical Success Index (CSI). Values for best performance within each metric are bolded.

Exp	Description	Correlation	RMSE	MAE	R^2	Hit Rate	FAR	CSI
Exp 1	No DWT, no non-image data	0.747	0.473	0.398	0.549	0.364	0.556	0.250
Exp 2	No DWT, with non-image data	0.777	0.449	0.372	0.593	0.375	**0.200**	0.343
Exp 3	DWT, each node separately	0.850	0.461	0.356	0.571	0.625	0.444	0.417
Exp 4	DWT, all nodes together	0.923	0.315	0.261	0.800	0.781	0.324	0.568
Exp 5	DWT, all nodes, linear SVR	**0.930**	**0.312**	**0.245**	**0.803**	0.875	0.349	0.596
Exp 6	DWT, all nodes, RBF SVR	0.910	0.435	0.326	0.618	**0.938**	0.362	**0.612**

The addition of the wavelet transform significantly improved almost all metrics, namely in the massive 67% increase in hit rate. This is indicative that the denoising quality of the wavelet transform was very effective in aiding the model's ability to pick up on precursors, increasing accuracy for predicting changes and trends in ENSO events (Figure 10). Before the wavelet transform, the model likely picked up external noise in the dataset and viewed it as indicative of variations in ENSO, thus scoring far worse in hit rate, CSI, FAR, and correlation.

Using all four autoencoder nodes as channels likewise can be seen to have improved all metrics remarkably. This improvement is likely due to the greater amounts of relevant information from the four different reconstructions storing more patterns in the SST anomaly data, indicating that the decoder's varied alterations from the original data did in fact correctly replicate ENSO-correlated trends in the data (Figure 11).

Usage of the linear Support Vector Regression appeared to have also improved all metrics, scoring the highest in correlation, RMSE, MAE, and R^2. This is to be expected, as the extra information (albeit noisy) made up for any inaccurate biases created from using just the approximation of the wavelet transform. However, in the general forecasting metrics other than FAR, the RBF Regression scored far higher than in both the control and the linear SVR.

It is seen that while the addition of the RBF SVR improved the model's ability

to forecast the cycle trends themselves (Figure 12), the magnitudes of ENSO strength were more accurate than those modeled in the linear SVR and the control (Figure 13). Though the RBF likely picked up on more non-linear ENSO precursors in the detail components for higher quality in the predictions of starts and ends of ENSO events, the comparatively poor R^2, RMSE and MAE were credible to the noisy components generating bias in the model's interpretation of the strength in SST anomaly variations.

5. Conclusions

We developed a hybrid deep learning model for El Niño event prediction using monthly SST anomaly data and additional climate features. M-band discrete wavelet transforms (DWTs) were applied to pre-process the data, generating multi-scale features. Usage of autoencoders to reconstruct original data into non-image and image data with high ENSO-indicative patterns were used. Support vector regression (SVR) approximators were then employed for integrating wavelet detail components into the model, further improving final predictions. Multiple control experiments were run to evaluate the extent of improvements from each technique used.

In addition to successfully producing a robust deep learning model for ENSO forecasting, the hypotheses that a) incorporating wavelet transforms (DWT) for multi-scale feature extraction and b) combining all components with the SVR, and the c) usage of both autoencoded image and non-image data in deep learning models, would increase accuracy were supported, though to varying degrees within each metrics.

Experiment results indicate that the autoencoder non-image data produced notable, improving CSI by 37% from the true control experiment. Usage of the DWT on one node also greatly improved all metrics, with a 67% increase in hit rate. The usage of all four autoencoder channels improved the model the most, significantly improving almost all metrics by a large margin. The final hybrid model, integrating all these enhancements, achieved the highest overall performance in almost all metrics, with a correlation of 0.930, a CSI of 0.612, and advances in overall predictive accuracy, validating the effectiveness of this combined approach (Table 2).

6. Future Research

Some drawbacks of this model are the low resolution of input data. As the dimensions were cut into a quarter of the original, input data to the CNN was relatively low, potentially worsening the quality of the predictions.

Computational efficiency was also a drawback in terms of size of data that could be processed, trained upon, or utilized in any way. The entirety of this research was done with a CPU, hence in various cases such as the convolutional autoencoder, only 4 reconstructions were made to reduce computational times. An increase in channels may allow for greater ENSO correlated information learned from the model.

Such a principle also applies in the architectures of the deep learning techniques. In both the LSTM autoencoders and the CNN, a balance had to be struck between architecture complexity and computational efficiency. Greater complexity of the model may allow for superior feature selection.

New forms of neural networks may also improve predictions. Usage of the recently developed Vision Transformer Model may perform better over a regular convolutional neural network [31]. Quantum convolutional neural networks, liquid neural networks, and continuous time recurrent quantum neural networks likewise may further improve robustness of the model.

Acknowledgements

The authors would like to first thank Western Connecticut State University for providing the facilities where research was conducted.

The authors are grateful for Andrew Li's guidance and mentorship. He has aided considerably in the development of the Discrete Wavelet Transform and the creation of the convolutional neural network architecture.

The authors are likewise grateful for Adeethyia Shankar's mentorship. He has provided notable assistance in the development of the wavelet denoise function and aided in calculation of statistics.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Philander, S.G.H. (1985) El Niño and La Niña. *Journal of the Atmospheric Sciences*, **42**, 2652-2662. https://doi.org/10.1175/1520-0469(1985)042<2652:enaln>2.0.co;2

[2] Trenberth, K.E. (1997) The Definition of El Niño. *Bulletin of the American Meteorological Society*, **78**, 2771-2777. https://doi.org/10.1175/1520-0477(1997)078<2771:tdoeno>2.0.co;2

[3] Donald Ahrens, C. (2016) Meteorology Today: An Introduction to Weather, Climate, and the Environment. 11th Edition, Cengage Learning.

[4] Collins, M., An, S., Cai, W., Ganachaud, A., Guilyardi, E., Jin, F., *et al.* (2010) The Impact of Global Warming on the Tropical Pacific Ocean and El Niño. *Nature Geoscience*, **3**, 391-397. https://doi.org/10.1038/ngeo868

[5] UNFCCC (CMA) (2023) Draft Decisions: Adaptation, Global Goal on Adaptation (GGA). UN Climate Change Conference—United Arab Emirates, Nov./Dec. 2023, Session CMA 5.

[6] Glantz, M.H. (1999) Currents of Change: El Niño's Impact on Climate and Society. Cambridge University Press.

[7] IARAN (2023) El Niño o Report.

[8] Smith, J. and Johnson, L. (2022) Impact of La Niña on Atmospheric Chemistry and Carbon Cycling. *Journal of Climate*, **35**, 1001-1015.

[9] Liu, J. and Wang, X. (2023) A Wavelet-Based Deep Learning Framework for Predicting Peak Intensity of Hurricanes in the Atlantic Ocean. *Atmospheric and Climate Sciences*, **13**, 587-606. https://doi.org/10.4236/acs.2023.134033

[10] Ham, Y., Kim, J. and Luo, J. (2019) Deep Learning for Multi-Year ENSO Forecasts. *Nature*, **573**, 568-572. https://doi.org/10.1038/s41586-019-1559-7

[11] Latif, M. and Keenlyside, N.S. (2009) El Niño/Southern Oscillation Response to Global Warming. *Proceedings of the National Academy of Sciences*, **106**, 20578-20583. https://doi.org/10.1073/pnas.0710860105

[12] Hou, Z.-Z., Yang, W.-C. and Shi, Z.-Q. (2001) Discrete Wavelet Transform for Multiple Decomposition of Gravity Anomalies. *Chinese Journal of Geophysics*, **44**, 529-537.

[13] Pan, Z.H. and Wang, X.D. (1996) Wavelets and Dynamic Pattern Recognition. *Proceedings of* 3*rd International Conference on Signal Processing* (*ICSP*'96), Vol. 2, 1304-1307. https://doi.org/10.1109/icsigp.1996.566536

[14] Ibebuchi, C.C. and Richman, M.B. (2024) Deep Learning with Autoencoders and LSTM for Enso Forecasting. Cambridge University Press.

[15] Huang, B., Thorne, P.W., Banzon, V.F., Boyer, T., Chepurin, G., Lawrimore, J.H., *et al*. (2017) Extended Reconstructed Sea Surface Temperature, Version 5 (ersstv5): Upgrades, Validations, and Intercomparisons. *Journal of Climate*, **30**, 8179-8205. https://doi.org/10.1175/jcli-d-16-0836.1

[16] Webb, E. (2020) Reanalysis of the Extended Multivariate Enso Index. Master's Thesis, The University of North Carolina.

[17] Daubechies, I. (1992) Ten Lectures on Wavelets. Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9781611970104

[18] Lin, T., Xu, S., Shi, Q. and Hao, P. (2006) An Algebraic Construction of Orthonormal M-Band Wavelets with Perfect Reconstruction. *Applied Mathematics and Computation*, **172**, 717-730. https://doi.org/10.1016/j.amc.2004.11.025

[19] Li, A., Liu, J., Liu, O. and Wang, X. (2023) Identifying Thermokarst Lakes Using Discrete Wavelet Transform-Based Deep Learning Framework. In: Pertusa, A., *et al*., Eds., *Pattern Recognition and Image Analysis*, Springer Nature, 479-489. https://doi.org/10.1007/978-3-031-36616-1_38

[20] Shankar, A., Chang, S., Wang, X. and Zhao, Y. (2022) Wavelet Based Machine Learning Approaches towards Precision Medicine in Diabetes Mellitus. *The FASEB Journal*, **36**, 290-297. https://doi.org/10.1096/fasebj.2022.36.s1.r6003

[21] Hinton, G.E. and Salakhutdinov, R.R. (2006) Reducing the Dimensionality of Data with Neural Networks. *Science*, **313**, 504-507. https://doi.org/10.1126/science.1127647

[22] Martinez, J.C. (2020) Introduction to Convolutional Neural Networks CNNs.

[23] Shi, X.J., Chen, Z.R., Wang, H., Yeung, D.-Y., Wong, W.-K. and Woo, W.-C. (2015) Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. *Advances in Neural Information Processing Systems*, Volume 28, 802-810.

[24] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014) Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, **15**, 1929-1958.

[25] Pedregosa, F., Varoquaux, G., Gramfort, A., *et al*. (2011) Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research*, **12**, 2825-2830.

[26] Smola, A.J. and Schölkopf, B. (2004) A Tutorial on Support Vector Regression. *Statistics and Computing*, **14**, 199-222. https://doi.org/10.1023/b:stco.0000035301.49549.88

[27] Bishop, C.M. (2006) Pattern Recognition and Machine Learning. Springer.

[28] EUMeTrain (n.d.) Verification of Categorical Forecasts: A Conceptual Introduction.

[29] Schaefer, J.T. (1990) The Critical Success Index as an Indicator of Warning Skill. *Weather and Forecasting*, **5**, 570-575. https://doi.org/10.1175/1520-0434(1990)005<0570:tcsiaa>2.0.co;2

[30] NOAA (2024) El Niño and La Niña (Oceanic Niño Index-Oni).

[31] Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lucic, M. and Schmid, C. (2021) Vivit: A Video Vision Transformer. 2021 *IEEE/CVF International Conference on Computer Vision* (*ICCV*), Montreal, 10-17 October 2021, 6836-6846. https://doi.org/10.1109/iccv48922.2021.00676

Appendix

1) Data Availability

The monthly averaged Sea Surface Temperature Anomaly data from the Extended Reconstructed Sea Surface Temperature Data can be found from this OpenDAP link: http://iridl.ldeo.columbia.edu/SOURCES/.NOAA/.NCDC/.ERSST/.version5/.anom/T/. The 3-month running mean standardized Ocean Nino Index Data can be found from NOAA's website: https://origin.cpc.ncep.noaa.gov/products/analysis_monitoring/ensostuff/ONI_v5.php. Data is available from 1950 to present; this study used April 1950 to April 2015.

2) Wavelet Filter Bank

Here is the 4-band 4-regular wavelet filter bank used—derived from [14].

α	$\beta^{(1)}$	$\beta^{(2)}$	$\beta^{(3)}$
0.08571302	−0.1045086525	0.2560950163	0.1839986022
0.1931394393	0.1183282069	−0.2048089157	−0.662289313
0.3491805097	−0.1011065044	−0.250343323	0.6880085746
0.5616494215	−0.0115563891	−0.2484277272	−0.1379502447
0.4955029828	0.6005913823	0.4477496752	0.0446493766
0.4145647737	−0.2550401616	0.0010274	−0.0823301969
0.2190308939	−0.4264277361	−0.0621881917	−0.0923899104
−0.1145361261	−0.082739818	0.5562313118	−0.0233349758
−0.0952930728	0.0722022649	−0.2245618041	0.0290655661
−0.1306948909	0.2684936992	−0.3300536827	0.0702950474
−0.0827496793	0.1691549718	−0.2088643503	0.0443561794
0.0719795354	−0.443703932	0.220295183	−0.0918374833
0.0140770701	0.0849964877	0.0207171125	10.0128845052
0.0229906779	0.1388163056	0.0338351983	0.0210429802
0.0145382757	0.0877812188	0.0213958651	0.0133066389
−0.0190928308	−0.1152813433	−0.0280987676	−0.0174753464