Prediction of ENSO Indices Using Deep Learning

A THESIS
SUBMITTED FOR THE DEGREE OF
Master of Technology

by

Naveen Reddi
Under the guidance of Prof. Ravi S Nanjundiah

Centre For Atmospheric and Oceanic Sciences
INDIAN INSTITUTE OF SCIENCE
Bangalore, India

2022 - 2024

Naveen Reddi
Under the guidance of Prof. Ravi S Nanjundiah
ORCID: xxxxx

CONTENTS

LIST OF FIGURES

Number *Page*

LIST OF TABLES

Number *Page*

Acknowledgements

I want to thank my supervisor, Prof. Ravi S. Nanjundiah for many insightful discussions during the development of the ideas in this report and the helpful comments on the text. His expertise in the area of study has been of immense help in understanding the concepts and in developing ideas.

I thank Dr Vybhav of ARTPARK and all my friends with whom I had invaluable discussions, which helped me in my research.

Finally, I would like to thank my parents for their constant support throughout my journey.

Abstract

This is a brief summary of my work on Deep learning for Enso Indices under Prof.Ravi S.Nanjundiah. Accurate prediction of the El Niño Southern Oscillation (ENSO) phenomenon is critical for understanding global climate patterns and mitigating associated risks. This study presents a novel approach to ENSO prediction utilizing a combination of autoencoders and ensemble learning techniques. Autoencoders are employed for feature learning, extracting relevant information from historical data, while ensemble models, including Bagging and Boosting, are utilized for prediction. The proposed methodology aims to address the challenges associated with ENSO prediction, such as data scarcity and model uncertainty. Evaluation results demonstrate the efficacy of the approach, showing significant improvements in prediction accuracy compared to traditional methods. The study contributes to the advancement of ENSO prediction capabilities, offering valuable insights for climate scientists and policymakers.

Chapter 1

INTRODUCTION

1.1 Overview of ENSO

The El Niño-Southern Oscillation (ENSO) stands as a prominent climate phenomenon, marked by cyclic variations in sea surface temperatures within the central and eastern Pacific Ocean. Its oscillations between warming phases, known as El Niño, and cooling phases, termed La Niña, exert a substantial influence on global weather patterns and climate variability. Notably, ENSO impacts various regions worldwide, including India, influencing precipitation, temperature, and extreme weather events.

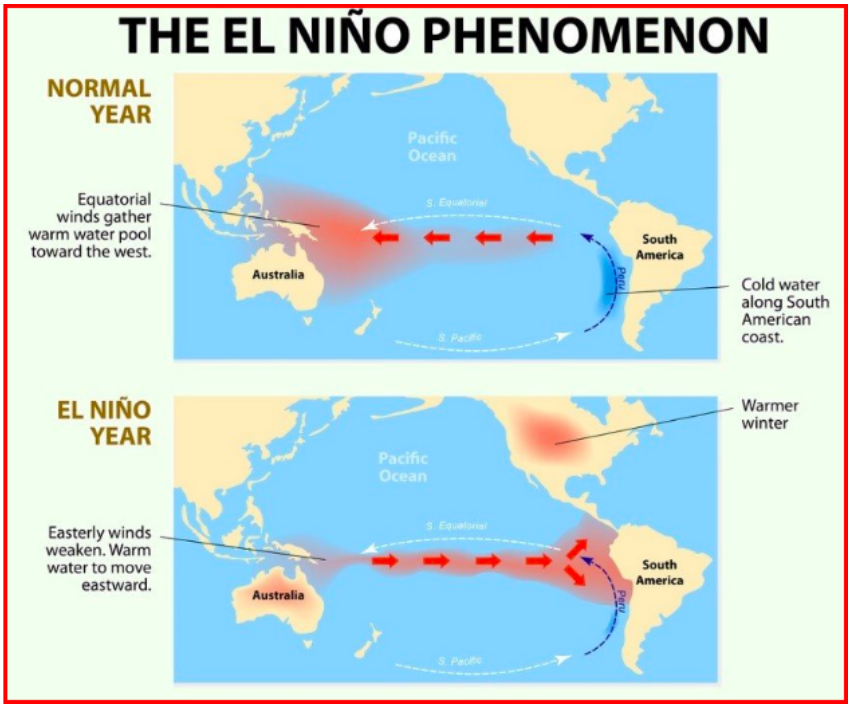

Figure 1.1: The El Niño Phenomenon

In India, the effects of El Niño are often associated with disruptions in the Indian summer monsoon, which is crucial for agriculture and water resources in the country. El Niño events typically lead to below-average monsoon rainfall in many parts of India, resulting in drought conditions, crop failures, and water shortages. Con-

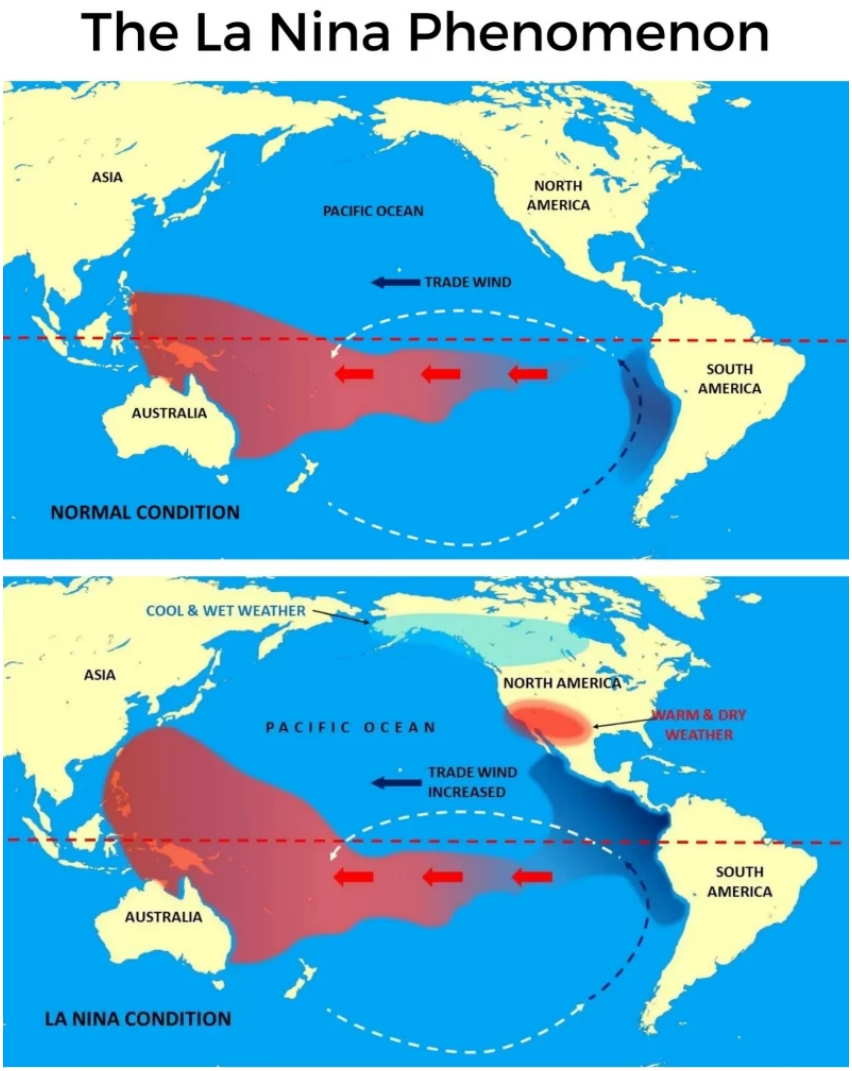

Figure 1.2: The La-Nina Phenomenon

versely, La Niña events can enhance monsoon rainfall, sometimes leading to excess precipitation and flooding in certain regions.

1.2 ENSO Indices

ENSO indices are quantitative measures used to monitor and predict climate variations associated with the El Niño-Southern Oscillation. One of the key indices is the Niño3.4 sea surface temperature anomaly. This index represents the departure from normal sea surface temperatures in the Niño3.4 region of the central equatorial Pacific Ocean, which is particularly sensitive to ENSO variability.

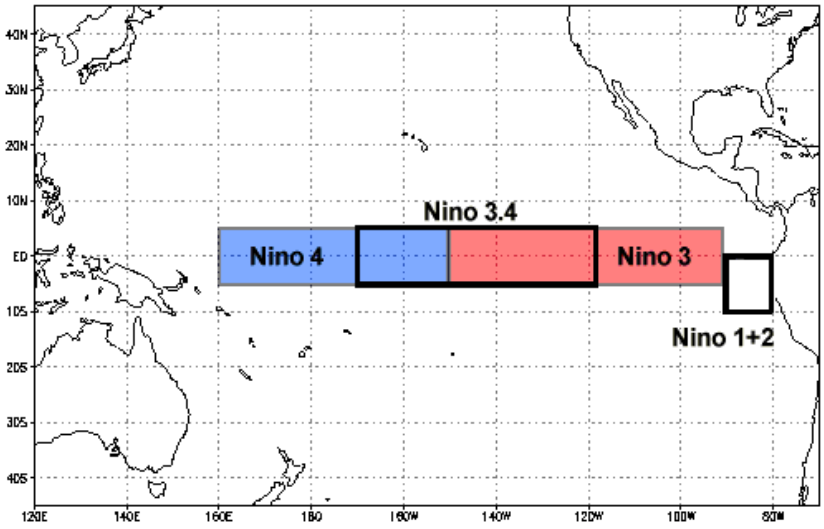

Figure 1.3: Map showing the Niño3.4 region along with other Niño regions used for monitoring ENSO

1.3 Objective

We propose a deep-learning method for predicting the El Niño-Southern Oscillation (ENSO). Given the vast amounts of climatic data available, data-driven machine learning and deep-learning techniques have emerged as promising tools for addressing climate science challenges.

Our approach utilizes an autoencoder model [1] for unsupervised feature learning from climatic variables, enabling the identification of potential predictors for ENSO indices. [2]. We employ ensemble prediction models to forecast these indices, specifically targeting the summer and monsoon period from June to September and March to May, as well as individual months within this timeframe. [3][4]. Accurate prediction of ENSO indices is crucial for gaining a comprehensive understanding of related climatic phenomena and improving oversight for effective climate management strategies.

By leveraging advanced deep-learning techniques [5] [6], our research aims to provide more reliable and timely predictions of ENSO events, thereby aiding in the development of better-informed climate adaptation and mitigation strategies.

Chapter 2

THE ENSO AND INPUT CLIMATIC VARIABLES

2.1 Climatic Variables

This chapter delves into the various climatic variables investigated to examine their relationship with the Niño (El Niño-Southern Oscillation) indices. Accurate prediction of ENSO events requires the analysis of multiple atmospheric and oceanic variables that significantly influence these indices. The primary variables considered in this study include:

Air Temperature (AT) Geopotential Height at 200 hPa (HGT) Sea Level Pressure (SLP) U-Wind at the Surface (UWND) V-Wind at the Surface (VWND) Sea Surface Temperature (SST).

2.2 Climatic Variables Combinations

The selection of these variables is based on their known significant influence on ENSO indices. To enhance the robustness of our predictive model, various combinations of these variables are also explored. These combinations include:

Air Temperature (AT) and Geopotential Height (HGT), Air Temperature (AT) and V-Wind (VWND), Geopotential Height (HGT) and V-Wind (VWND), U-Wind (UWND) and V-Wind (VWND), Sea Level Pressure (SLP) and Sea Surface Temperature (SST), Sea Level Pressure (SLP) and U-Wind (UWND), Sea Level Pressure (SLP) and V-Wind (VWND).

By examining these combinations, we aim to identify the most important predictors for forecasting the Niño indices.

2.3 Data Sources and Period

The climatic variables such as AT, HGT, UWND, SLP, and VWND are obtained from reanalysis-derived data provided by the National Centers for Environmental Prediction (NCEP). These data are available at a spatial resolution of 2.5° × 2.5°. The Sea Surface Temperature (SST) data are sourced from the National Oceanic and Atmospheric Administration (NOAA) Extended Reconstructed V3 dataset, available at a spatial resolution of 2.0° × 2.0°.

The input climatic variables are considered globally and at a monthly scale for the

period from 1958 to 2023 to ensure comprehensive analysis.

The Niño index, which serves as a key indicator for ENSO events, is obtained monthly from the NOAA Physical Sciences Laboratory (NOAA/PSD) for the period from 1958 to 2023.

Chapter 3

DATA PREPROCESSING

3.1 Introduction

Data preprocessing is a critical step in preparing climatic data for deep learning models. Proper preprocessing ensures that the data is clean, normalized, and structured optimally for model training and prediction. This chapter outlines the preprocessing steps, including spatial averaging, anomaly calculation, and normalization, undertaken to prepare the climatic variables for the study.

3.2 Spatial Averaging

To manage the vast amount of climatic data and to capture significant patterns, each variable is spatially averaged. Specifically, the input variables are averaged over a grid of 10° latitude × 20° longitude. This spatial resolution is chosen to balance the level of detail with computational efficiency, ensuring that the model can effectively learn from significant climatic patterns while reducing noise.

3.3 Anomaly Calculation

All climatic variables are considered at a monthly scale to capture seasonal variations and long-term trends. To focus on deviations from the norm, the data is converted to monthly anomaly data. Anomalies represent the difference between the observed value and the long-term average for that month. The anomaly for a variable can be calculated using the following equation:

$$AnamolyVariable_x^y = ClimaticVariable_x^y - mn(ClimaticVariable_m)$$

where $Climatic\ Variable_x^y$ is the variable in the m-th month of the y-th year; and $mn(Climatic\ Variable_m)$ is the average of the m-th month over the years.

3.4 Min-Max Normalization

Normalization of the input variables is performed to ensure that each variable contributes equally to the learning process. Min-Max normalization scales the data to a standard range, typically between 0 and 1, using the following formula:

$$X_{scaled} = \frac{(X - X_{min})}{(X_{max} - X_{min})}$$

Chapter 4

AUTOENCODERS

4.1 Introduction

In the realm of machine learning and deep learning, feature learning is a crucial step that allows models to automatically discover the representations needed for detection or classification tasks from raw data. One powerful tool for feature learning is the autoencoder. This chapter introduces autoencoders, explains their structure, and discusses how they are utilized for unsupervised feature learning in our study.

4.2 Auto-encoder

A single-layer autoencoder is a type of artificial neural network that comprises an input layer, a single hidden (internal) layer, and an output layer. The encoder component of the network processes data from the input layer to the hidden layer, capturing the data's nonlinear characteristics. Conversely, the decoder component reconstructs the input data by mapping it from the hidden layer back to the output layer. Since the output layer mirrors the input layer, the hidden layer is essential for learning the complex and nonlinear patterns present in the data.

4.3 Structure of an Autoencoder

An autoencoder typically consists of two main components: the encoder and the decoder.

Encoder:

Function: The encoder compresses the input data into a latent-space representation.
Structure: It is a neural network that takes the input data and maps it to a lower-dimensional space (the bottleneck layer).The encoder consists of a series of layers, each one reducing the dimensionality of the data.

Latent Space (Bottleneck):

Function: The bottleneck layer is the compressed representation of the input data. It captures the most salient features needed to reconstruct the input.
Structure: It is the layer with the smallest number of neurons in the network, representing the reduced feature set.

Decoder:

Function: The decoder reconstructs the input data from the latent-space representation.

Structure: It is a neural network that takes the compressed data from the bottleneck layer and maps it back to the original input space. It generally mirrors the structure of the encoder.

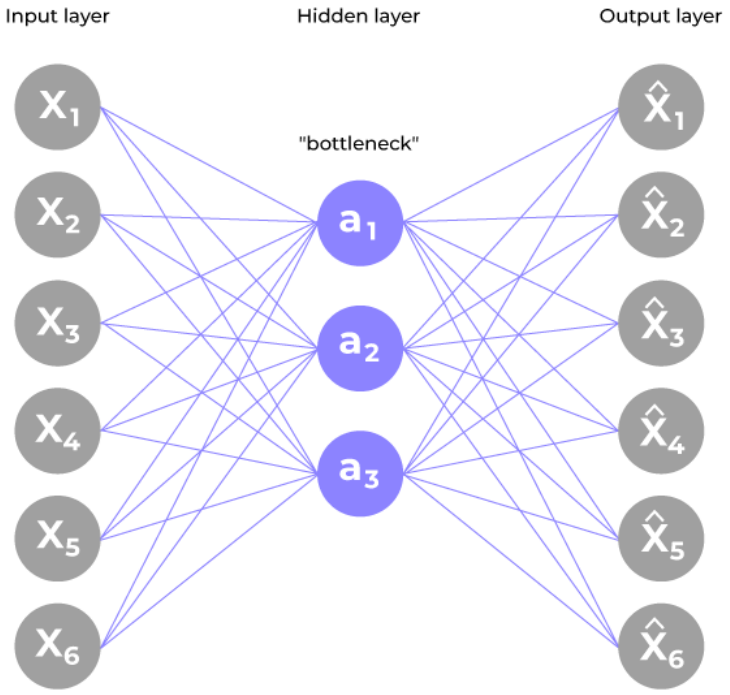

Figure 4.1: Simple Autoencoder

4.4 Working of Autoencoder

The autoencoder is trained to minimize the difference between the input and the reconstructed output. This is typically done using a loss function, such as mean squared error (MSE), which measures the reconstruction error. The training process involves backpropagation and optimization algorithms like gradient descent to adjust the weights of the network.

$$Loss = \frac{1}{n}\sum_{i=1}^{n} \|x_i - \hat{x}_i\|^2$$

where x_i is the input and $\hat{x}_i$ is the reconstructed output.

Chapter 5

FEATURE LEARNING BY AUTOENCODER

This chapter presents the design and implementation of autoencoder models used for analyzing individual climatic variables and their combinations.The key configurations, training procedures, and activation functions are detailed below.

5.1 Autoencoder Architecture

The autoencoder models are designed with specific ratios of input to hidden layer nodes, which differ based on whether the model processes individual variables or their combinations. For individual variables, the input to hidden layer node ratio is 100:20, whereas for combinations, it is 100:10. The inputs represent variables averaged over a 10° latitude by 20° longitude grid. Therefore, the number of input nodes for individual variables is 324 ((180/10) × (360/20)), and for combined variables, it is 648 (324 + 324), as each combination involves two variables.

Different autoencoders are trained separately for seven individual variables and seven combinations. The autoencoder designed for individual climatic variables, excluding Sea Surface Temperature (SST), has an architecture with 324 input nodes, 65 hidden nodes, and 324 output nodes. For SST, which is not present over land surfaces, the architecture is [132; 40; 132].

For combined variables, excluding the combination of Sea Level Pressure (SLP) and SST, the autoencoder structure is [648; 65; 648], corresponding to the 324 nodes for each variable. The autoencoder for the SLP + SST combination has a [456; 45; 456] architecture, with 324 nodes for SLP and 132 nodes for SST.

5.2 Training Objective

Feature learning in these autoencoders is performed by training the models with input variables, aiming to minimize reconstruction errors between the input and output layers. The models adjust their biases and weights during training to achieve this objective.

5.3 Activation Functions

The activation function used from the input layer to the hidden layer is a nonlinear hyperbolic tangent function. This nonlinear activation function is employed to

capture the inherent nonlinearity in the data.

Formally, let $node_i \ \mathbb{R}^n$ represent the input layer with n nodes. The activation of a neuron in the hidden layer $hidden_j$ is defined by following Equation.

$$\text{hidden}(\text{node}_j) = \text{func}\left(\sum_{i=1}^{n} \text{Weight}_i^{\text{hid}} \times \text{node}_i + \text{bias}_i^{\text{hid}}\right) \tag{5.1}$$

The activation of a neuron in the hidden layer ($\text{hidden}(\text{node}_j)$) is calculated using the hyperbolic tangent function $\text{func}(z)$, where z is the input to the neuron. The hyperbolic tangent function is defined as:

$$\text{func}(z) = \frac{e^{2z} - 1}{e^{2z} + 1}$$

$\text{hidden}(\text{node}_j)$ is the learned node of the hidden layer, $\text{Weight}^{\text{hid}}$ is the weight matrix from the input to the hidden layer, and Bias^{hid} is the bias of the hidden layer.

The activation function for the hidden to output layer transition is described by following Equation.

$$\text{output}(\text{node}_j) = \text{func}\left(\sum_{j=1}^{m} \text{Weight}_j^{\text{out}} \times \text{node}_j + \text{bias}_j^{\text{out}}\right) \tag{5.2}$$

The activation of a neuron in the output layer (node_d) is calculated using the linear activation function $\text{func}(z)$, where z is the input to the neuron. The linear activation function is defined as:

$$\text{func}(z) = (a \cdot z + b)$$

node_d belongs to $\mathbb{R}^m$, $\text{Weight}_{\text{out}}$ is the weight matrix, and bias_{out} belongs to $\mathbb{R}^m$ and is the bias of the output layer.

5.4 Identification of Potential Predictors

The features learned in the hidden layer are critical as they represent potential predictors of the indices. The final step in identifying these potential predictors involves applying a threshold to the learned weights. Specifically, a threshold is imposed on the weights of the connections between the input and hidden nodes. Only weights that are greater than some standard deviations above the mean of all weights are considered when evaluating the features of the hidden layer nodes. This threshold ensures that at least 10% of input nodes contribute to the calculation of

features at each hidden node. Consequently, all hidden nodes that meet this criterion are regarded as potential predictors.

The predictors are derived from the features learned at the hidden layer nodes, following the application of the weight threshold, as described in Equation 5.3. This equation illustrates the potential predictor calculated from the feature learned at the (i)-th hidden layer node. In this context, $Weight_{k:i}$ represents the weight of the edge between the (k)-th input node and the (i)-th hidden node. $Input_k$ corresponds to the (k)-th input node, while $threshold_i$ denotes the determined threshold for the (i)-th hidden node.

$$\{\text{Potential predictor}_{\text{Hidden node}}\}_i = \sum_{k=1}^{n} \left(\text{Weight}_{k:i} \cdot \text{Input}_k\right) \text{ if Weight}_{k:i} > \text{threshold}_i \tag{5.3}$$

Chapter 6

FEATURE RANKING AND CORRELATION STUDY

This chapter delves into the methodologies employed for feature ranking and the correlation analysis between the identified predictors and the indices. Given that the feature learning process accounts for the nonlinearity within the data, it is crucial to examine both linear and nonlinear relationships between the predictors and the indices.

6.1 Correlation Analysis

Feature ranking is conducted through both linear and nonlinear correlation studies between the predictors and the mean indices of studied months. Since the feature learning process incorporates the nonlinearity inherent in the data to identify potential predictors, it is essential to explore these nonlinear relationships in detail.

6.2 Feature Ranking

The correlation analysis is performed with a lead time ranging from 1 to 12 months to determine the month in which the predictor exhibits the highest correlation coefficient with the index. For instance, when considering the cumulative average index for the period June to September, a lead time of 1 month indicates the correlation between the predictor in May and the average index starting in June. when considering the cumulative average index for the period March to May, a lead time of 1 month indicates the correlation between the predictor in February and the average index starting in March.

6.3 Correlation Methods

One linear and three nonlinear correlations were considered.

Pearson correlation: To study the linear relationship between the predictors and indices.

$$\mu = \frac{\sum_{\text{year=1}}^{\text{num}} (\text{var1}_{\text{month}}^{year} - (\overline{\text{var1}}_{\text{month}}))(\text{var2}_{\text{month}}^{year} - (\overline{\text{var2}}_{\text{month}}))}{\sqrt{\sum_{\text{year=1}}^{\text{num}} (\text{var1}_{\text{month}}^{year} - (\overline{\text{var1}}_{\text{month}}))^2} \sqrt{\sum_{\text{year=1}}^{\text{num}} (\text{var2}_{\text{month}}^{\text{year}} - (\overline{\text{var2}}_{\text{month}}))^2}}$$

where $\text{var1}_{\text{month}}^{year}$ and $\text{var2}_{\text{month}}^{year}$ represent the indices (NINO) and potential predictor

for the $month$-th month in the $year$-th year, respectively.
$\overline{\text{var1}}_{\text{month}}$ and $\overline{\text{var2}}_{\text{month}}$ are the mean values for the $month$-th month, and num is the total number of years.

Kendall correlation: A nonlinear metric measuring the strength of the dependence between two variables based on the ranks of data.

$$\tau = \frac{\text{num}_{\text{concordant}} - \text{num}_{\text{discordant}}}{\frac{1}{2}n(n-1)}$$

where ($\text{num}_{\text{concordant}}$) and ($\text{num}_{\text{discordant}}$) denote the number of concordant and discordant sample pairs, respectively.
All values of v_i and w_i are unique, for (v_1, w_1), (v_2, w_2), ..., (v_n, w_n) belonging to the set of observations of var1 and var2.
A pair of observations (v_i, w_i) and (v_j, w_j) is concordant if the ranks for both elements agree, that is, if both $(v_i > v_j)$ and $(w_i > w_j)$, or both $(v_i < v_j)$ and $(w_i < w_j)$, where $(i \neq j)$. Otherwise, the pair is discordant.

Mutual information: Provides the information about one variable from the other.

$$\text{MI(var1, var2)} = \sum_{v1 \in \text{var1}} \sum_{v2 \in \text{var2}} p(v1, v2) \log \left(\frac{p(v1,v2)}{p(v1)p(v2)} \right)$$

Where p(v1,v2) denotes the joint probability function; and p(v1) and p(v2) represent the marginal probability distribution of var1 and var2.

Spearman's rank correlation: Quantifies the statistical dependence between the ranking of two variables.
Formally, for a sample of n size needs to be calculated as follows.
First, n raw scores for the variables var1 and var2 are converted to ranks, denoted by rankvar1 and rankvar2.
It changes all the $var1_i$ and $var2_i$ to $rankvar1_i$ and $rankvar2_i$, respectively, for i = 1, 2,.....n.

$$\rho = \frac{\text{cov(rankvar1,rankvar2)}}{\sigma_{\text{rankvar1}} \sigma_{\text{rankvar2}}}$$

where cov(rankvar1, rankvar2) denotes the covariance of the ranked variables var1 and var2; and $\sigma_{rankVar1}$ and $\sigma_{rankVar2}$ are standard deviations.

Chapter 7

CONSTRUCTION OF PREDICTOR SETS

To identify the most relevant predictors, a deep-learning method was employed to analyze individual variables and their combinations. Four predictor sets, namely predSet1, predSet2, predSet3, and predSet4, were constructed by selecting the top correlated predictors at their optimal lead month. Specifically, these sets consisted of the top 5, 10, 15, and 20 predictors, respectively, that exhibited the highest correlation with the indices at their respective lead months.

The prediction was made available at a month equal to the shortest lead time of any predictor in the set. For instance, if three predictors had lead months of January, March, and April, the prediction would be provided in April.

A total of 56 predictor sets were built for each index, corresponding to one type of correlation study. These sets were constructed by combining four different predictor sets with 14 different types of climatic variables, resulting in a total of $4 \times 14 = 56$ predictor sets. This process was repeated for all four correlation studies.

Chapter 8

MACHINE LEARNING TECHNIQUES

Two different machine-learning-based ensemble models, namely **Random Decision Forest(Bagging)** and **Gradient boosting(Boosting)** are used. The input to the models is the predictor sets designed and the output are the NINO indexes.

8.1 Bagging Technique

Random Forest is a popular ensemble learning method that combines multiple decision trees to improve the accuracy and robustness of predictions. It's a type of bagging (Bootstrap Aggregating) technique.

Working:

1. **Bootstrap Sampling**: Randomly select a subset of training data with replacement (bootstrap sample).
2. **Decision Tree**: Train a decision tree on the bootstrap sample.
3. **Feature Randomness**: Randomly select a subset of features to consider at each node of the decision tree.
4. **Tree Construction**: Construct the decision tree by recursively partitioning the data into subsets based on the selected features.
5. **Prediction**: Make predictions using the trained decision tree.
6. **Ensemble**: Combine the predictions from multiple decision trees (typically hundreds or thousands) to produce the final prediction.

Mathematical Formulation:

Let's denote the training data as $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i is the feature vector and y_i is the target variable.

The Random Forest algorithm can be formulated as:

For each bootstrap sample $\mathcal{D}_b$ of size N from $\mathcal{D}$:
Train a decision tree T_b on $\mathcal{D}_b$.

Predict the target variable using T_b for each instance in $\mathcal{D}$.
Combine the predictions from all decision trees using voting or averaging. The final prediction for a new instance x can be calculated as:

$\hat{y} = \frac{\sum_{b=1}^{B} T_b(x)}{B}$

where B is the number of decision trees, and $T_b(x)$ is the prediction from the b^{th} decision tree.

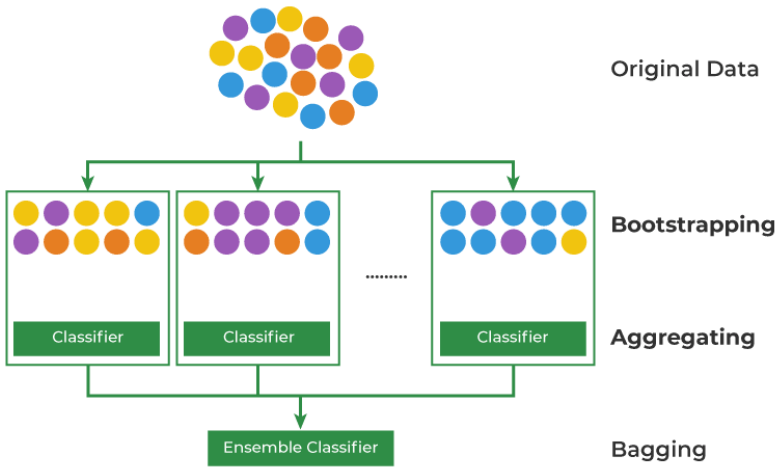

Figure 8.1: Bagging Process

8.2 Boosting Technique

Gradient Boosting is a popular ensemble learning method that combines multiple weak models to create a strong predictor. It's a type of boosting technique.

Working:

1. **Initialize**: Initialize a weak model (typically a decision tree) with a constant value.
2. **Residuals**: Calculate the residuals between the target variable and the current prediction.
3. **Gradient**: Compute the gradient of the loss function with respect to the residuals.
4. **Update**: Update the model parameters by moving in the direction of the negative gradient.

5. **Additive Model**: Add the updated model to the ensemble.

6. **Repeat**: Repeat steps 2-5 until convergence or a stopping criterion is reached.

Mathematical Formulation:

Let's denote the training data as $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i is the feature vector and y_i is the target variable.

The Gradient Boosting algorithm can be formulated as:

Initialize the model parameters θ_0 and the prediction $\hat{y}_0$.
For each iteration t = 1, 2, ..., T:
Compute the residuals: $r_i = y_i - \hat{y}_{t-1}(x_i)$.
Compute the gradient: $g_i = -\frac{\partial L(y_i, \hat{y}_{t-1}(x_i))}{\partial \hat{y}_{t-1}(x_i)}$.
Update the model parameters: $\theta_t = \theta_{t-1} - \alpha \sum_{i=1}^{N} g_i$.
Update the prediction: $\hat{y}_t(x) = \hat{y}_{t-1}(x) + \alpha \sum_{i=1}^{N} g_i$.
The final prediction is: $\hat{y}(x) = \hat{y}_T(x)$.

where $L(y, \hat{y})$ is the loss function, α is the learning rate, and T is the number of iterations.

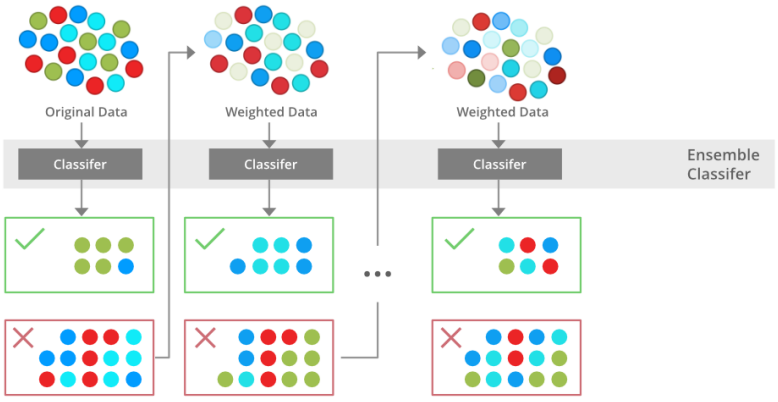

Figure 8.2: Boosting Process

Chapter 9

PERFORMANCE METRICS

In this chapter, we outline the performance metrics used to evaluate the predictive accuracy and robustness of our models.By employing these performance metrics, we gain a comprehensive understanding of our model's predictive capabilities. Pearson Correlation assesses the linear relationship between predicted and actual values, RMSE provides an aggregate measure of prediction error magnitude, and the F1 Score balances precision and recall in classification tasks. Together, these metrics offer a robust framework for evaluating the effectiveness of our predictive models.

9.1 Pearson Correlation

Pearson Correlation Coefficient is a measure of the linear correlation between two variables, denoted as (X) and (Y). It ranges from -1 to 1, where 1 indicates a perfect positive linear relationship, -1 indicates a perfect negative linear relationship, and 0 indicates no linear relationship.

$$\mu = \frac{\sum_{i=1}^{n}(y_{pred}-(\overline{y}_{pred}))(y_{test}-(\overline{y}_{test}))}{\sqrt{\sum_{i=1}^{n}(y_{pred}-(\overline{y}_{pred}))^2}\sqrt{\sum_{i=1}^{n}(y_{test}-(\overline{y}_{test}))^2}}$$

y_{pred} and y_{test} are the predicted and actual values.
$\overline{y}_{pred}$ and $\overline{y}_{test}$ are mean of predicted and actual values.
n is the number of data points.

9.2 Root Mean Square Error (RMSE)

Root Mean Square Error (RMSE) is a widely used measure of the differences between values predicted by a model and the values actually observed. It provides an aggregate measure of the model's predictive accuracy and is sensitive to large errors.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

y_i is the actual observed value.
$\hat{y}_i$ is the predicted value.
n is the number of observations.

9.3 F1 Score

The F1 Score is a measure of a model's accuracy in binary classification tasks. It is the harmonic mean of precision and recall, providing a balance between these two metrics. The F1 Score ranges from 0 to 1, with 1 being the best possible score.

$$\textbf{F1 Score} = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Precision is the ratio of correctly predicted positive observations to the total predicted positives:

$$\textbf{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

Recall is the ratio of correctly predicted positive observations to all actual positives:

$$\textbf{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

Chapter 10

TRAINING AND TESTING

In this chapter, we outline the training and testing procedures employed in our study. We utilized Bagging (Random Forest) and Boosting (Gradient Boosting) techniques, evaluated using Pearson Correlation and Root Mean Square Error (RMSE) as performance metrics.

The training data spans from 1958 to 2007, while the testing data covers the period from 2008 to 2023. The models were trained for individual months from March to September, as well as for the mean indices of the March-April-May (MAM) and June-July-August-September (JJAS) periods.

10.1 Evaluation of the Proposed Approach

A total of 12 different climatic variables (both individual and combined) are considered, and the study is based on four types of correlation: Pearson, Kendall, Mutual Information, and Spearman.

Four predictor sets are constructed for each case. Therefore, for both indices over a particular time span (whether aggregate or individual months), there are 192 sets of prediction results (12 climatic variables × 4 correlation types × 4 predictor sets). Exhaustive predictions are performed using all these sets; however, only the predictions with the best accuracy for the indices will be presented.

Chapter 11

TEST RESULTS

In this chapter, we present the results of the predictive models trained using Bagging and Boosting techniques. The performance of the models is evaluated using Correlation and RMSE metrics. We also provide visual comparisons of the observed and predicted values for each month from March to September, as well as the MAM (March, April, May) average and JJAS (June, July, August, September) average indices.

11.1 MAM avg Bagging

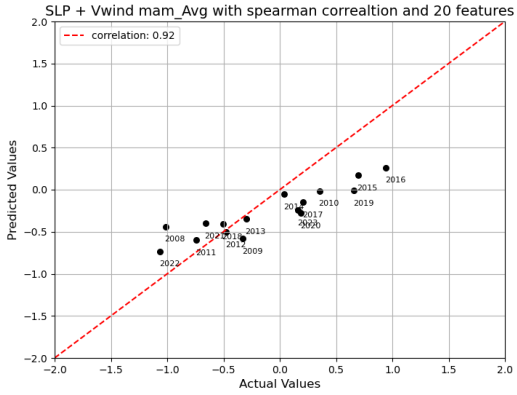

Figure 11.1: Variations in the obs and pred

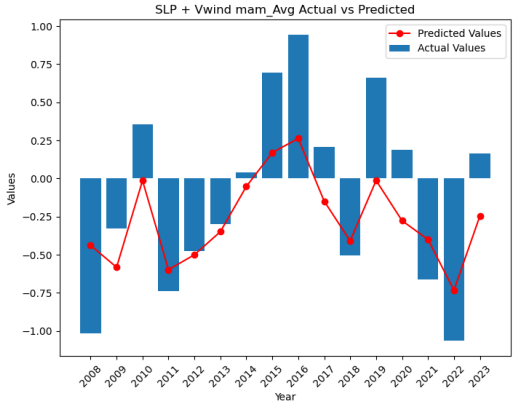

Figure 11.2: Predicted vs Actual

11.2 MAM avg Boosting

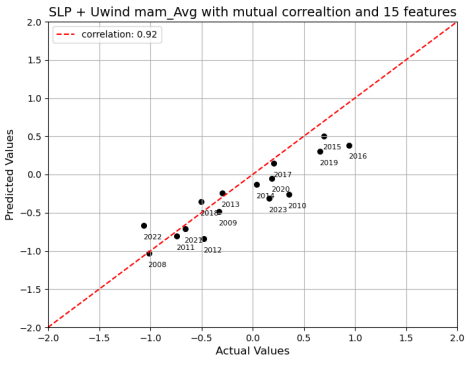

Figure 11.3: Variations in the obs and pred

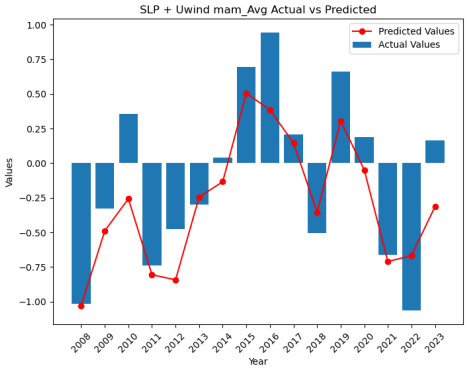

Figure 11.4: Predicted vs Actual

11.3 MARCH Bagging

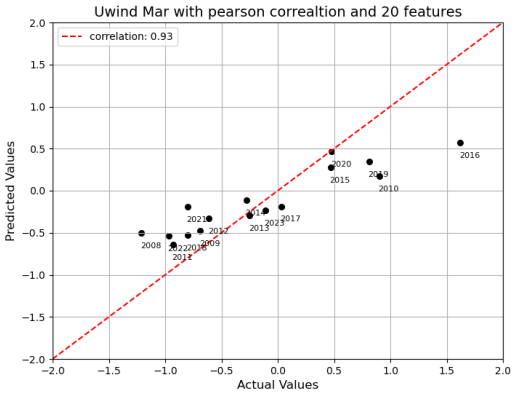

Figure 11.5: Variations in the obs and pred

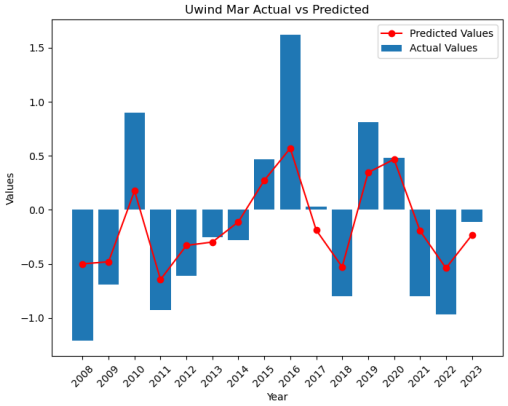

Figure 11.6: Predicted vs Actual

11.4 MARCH Boosting

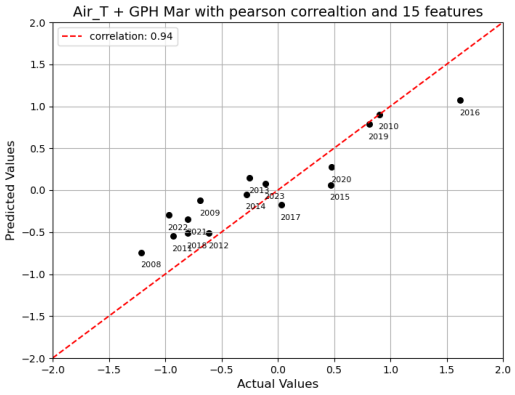

Figure 11.7: Variations in the obs and pred

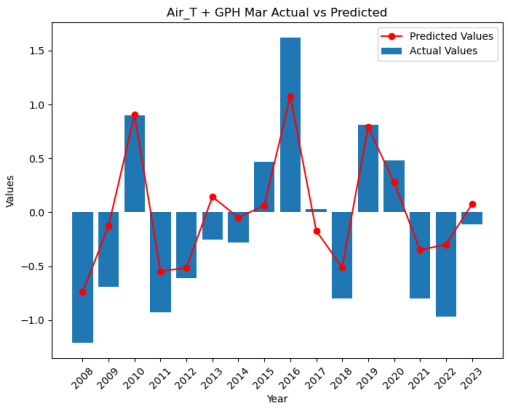

Figure 11.8: Predicted vs Actual

11.5 April Bagging

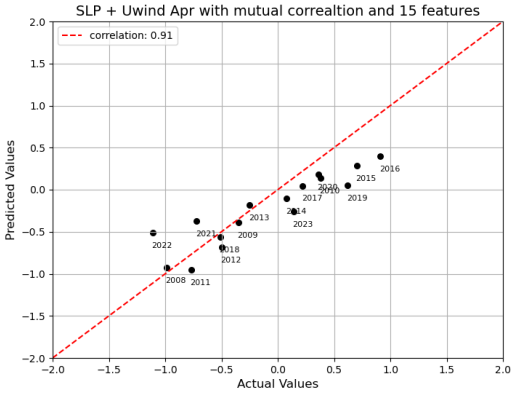

Figure 11.9: Variations in the obs and pred

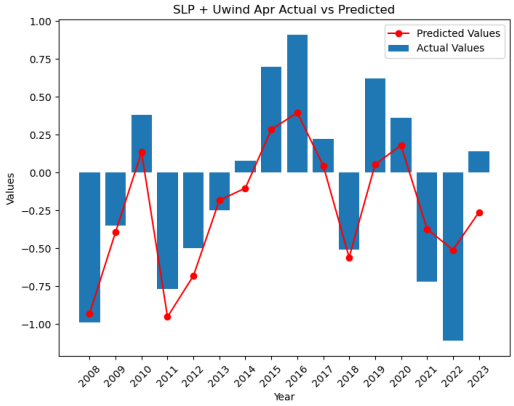

Figure 11.10: Predicted vs Actual

11.6 April Boosting

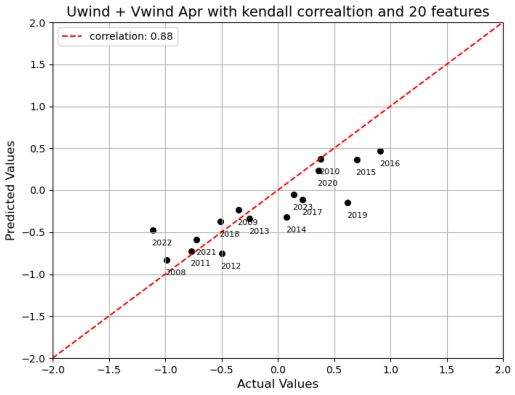

Figure 11.11: Variations in the obs and pred

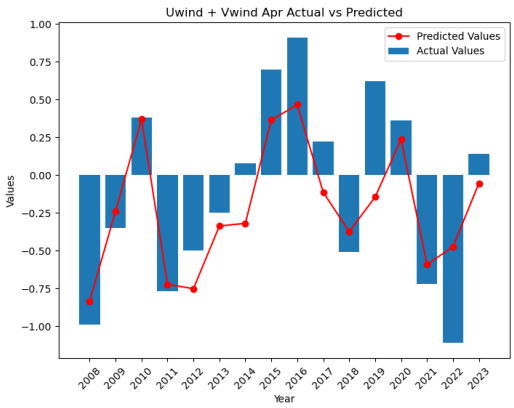

Figure 11.12: Predicted vs Actual

11.7 May Bagging

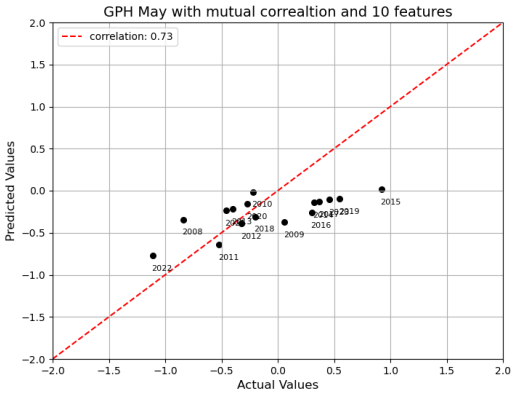

Figure 11.13: Variations in the obs and pred

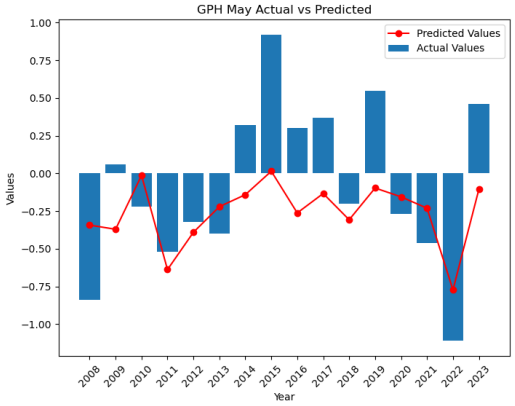

Figure 11.14: Predicted vs Actual

11.8 May Boosting

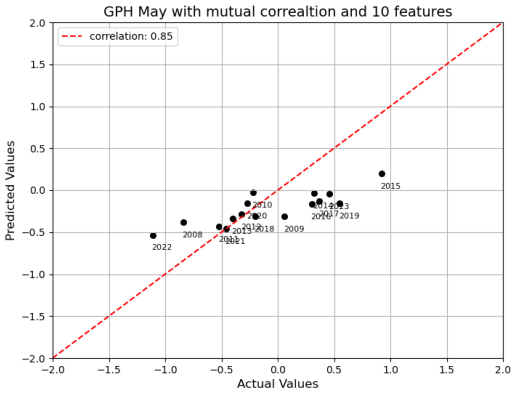

Figure 11.15: Variations in the obs and pred

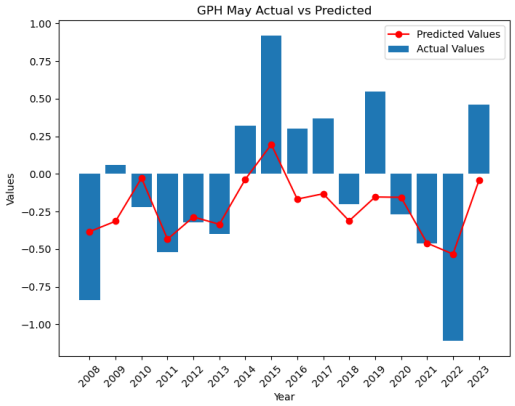

Figure 11.16: Predicted vs Actual

In the figures presented above, a comparative analysis is provided between the Bagging and Boosting models with performance metric as Correlation for the months of March, April, May, and the MAM (March-April-May) average.

11.9 JJAS avg Bagging

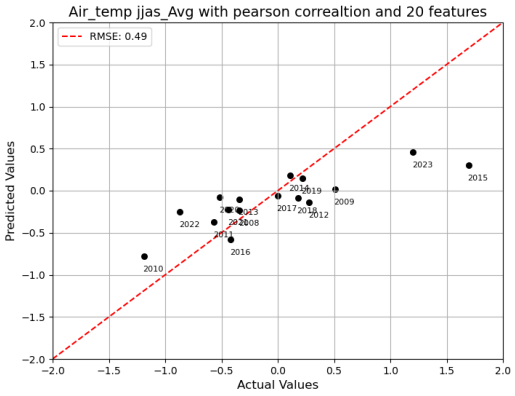

Figure 11.17: Variations in the obs and pred

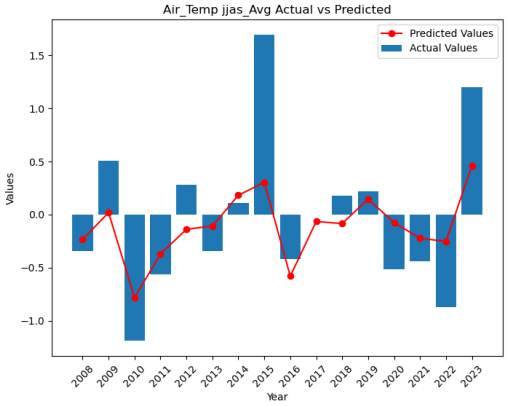

Figure 11.18: Predicted vs Actual

11.10 JJAS avg Boosting

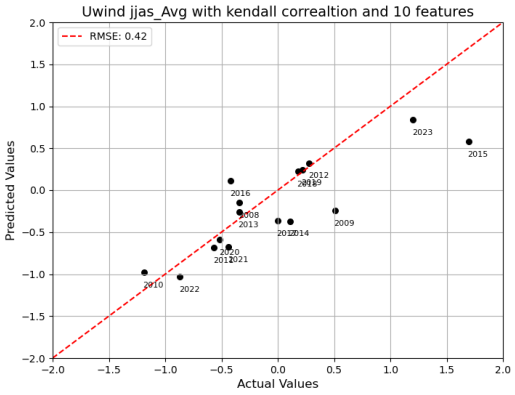

Figure 11.19: Variations in the obs and pred

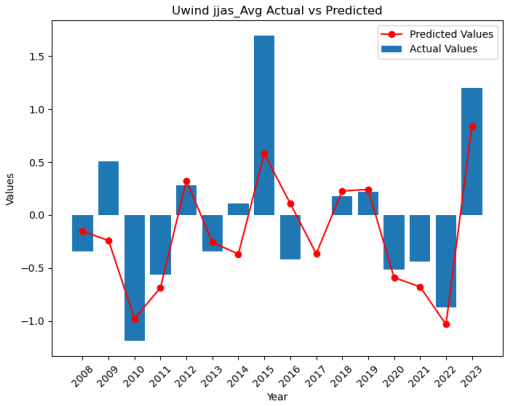

Figure 11.20: Predicted vs Actual

11.11 June Bagging

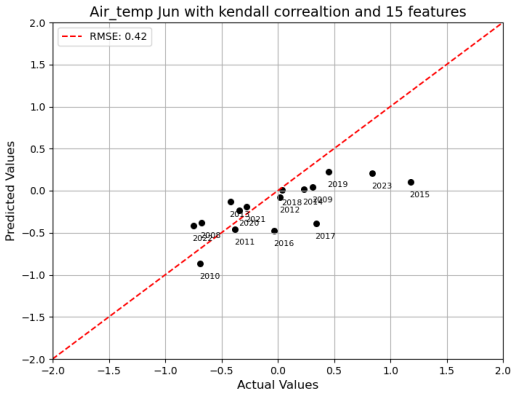

Figure 11.21: Variations in the obs and pred

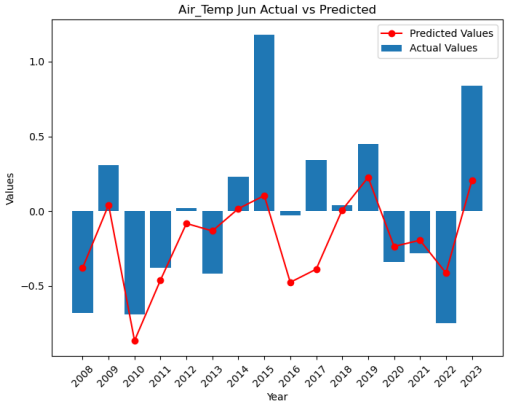

Figure 11.22: Predicted vs Actual

11.12 June Boosting

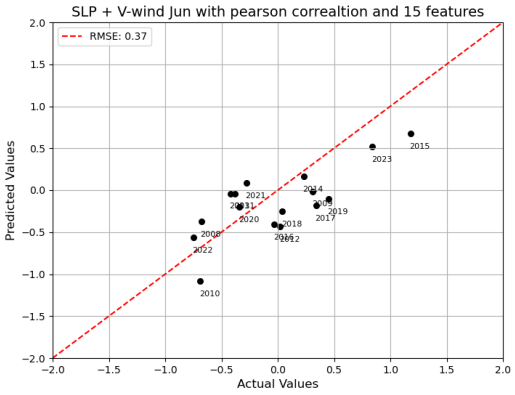

Figure 11.23: Variations in the obs and pred

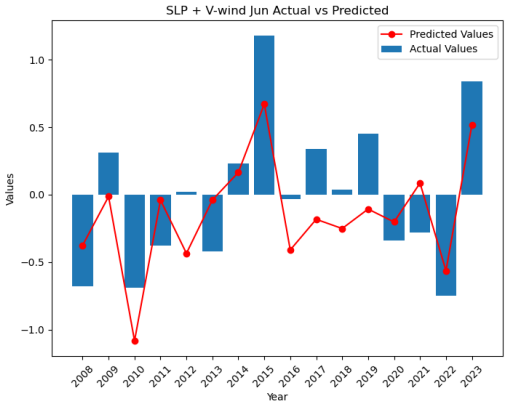

Figure 11.24: Predicted vs Actual

11.13 July Bagging

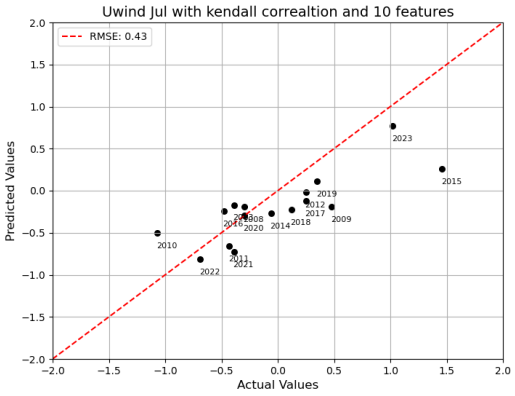

Figure 11.25: Variations in the obs and pred

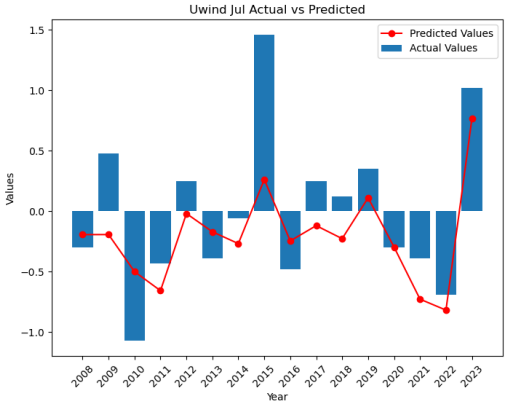

Figure 11.26: Predicted vs Actual

11.14 July Boosting

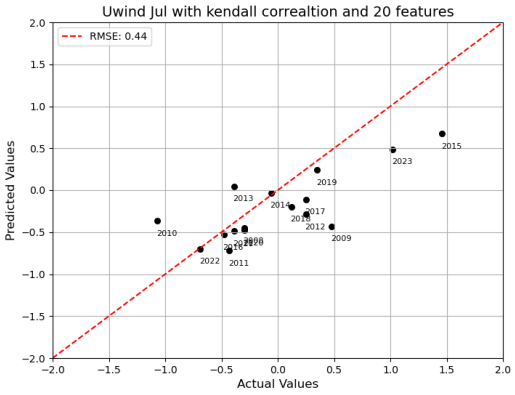

Figure 11.27: Variations in the obs and pred

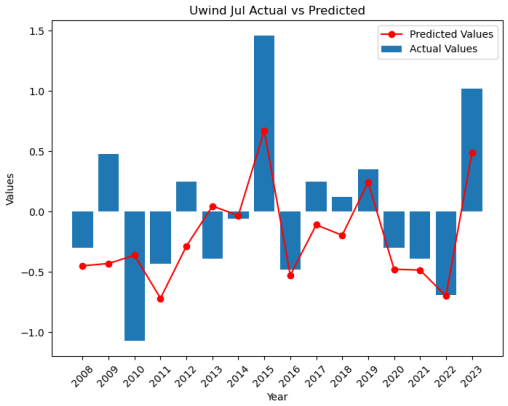

Figure 11.28: Predicted vs Actual

11.15 August Bagging

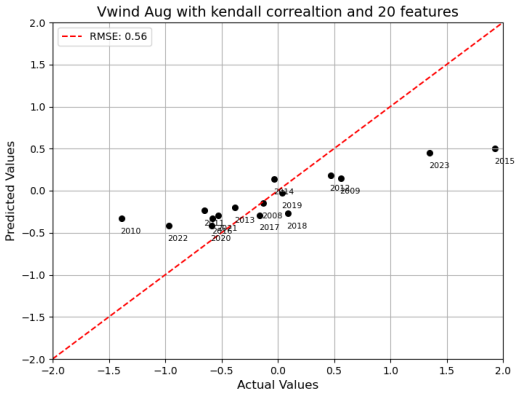

Figure 11.29: Variations in the obs and pred

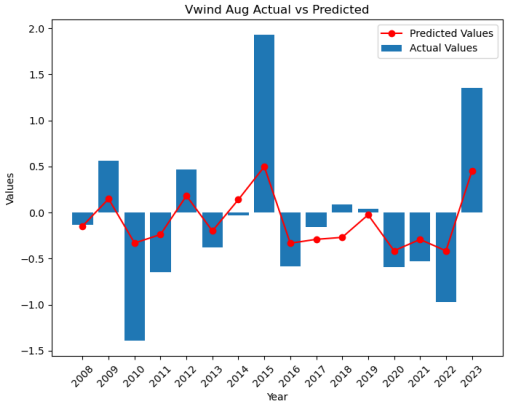

Figure 11.30: Predicted vs Actual

11.16 August Boosting

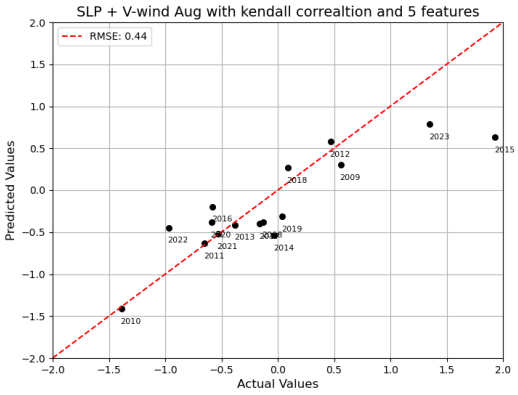

Figure 11.31: Variations in the obs and pred

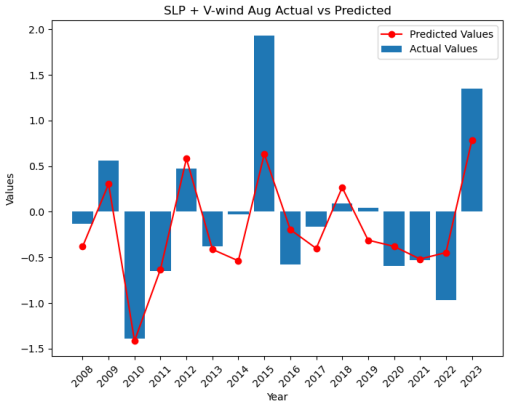

Figure 11.32: Predicted vs Actual

11.17 September Bagging

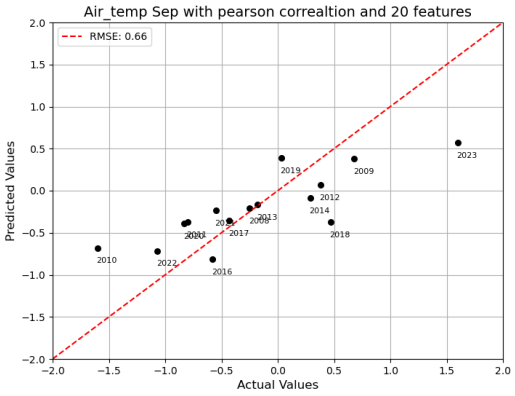

Figure 11.33: Variations in the obs and pred

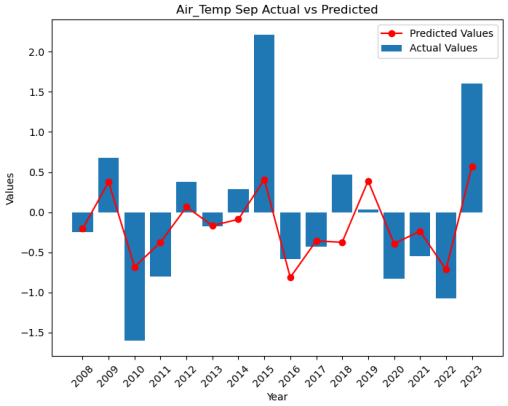

Figure 11.34: Predicted vs Actual

11.18 September Boosting

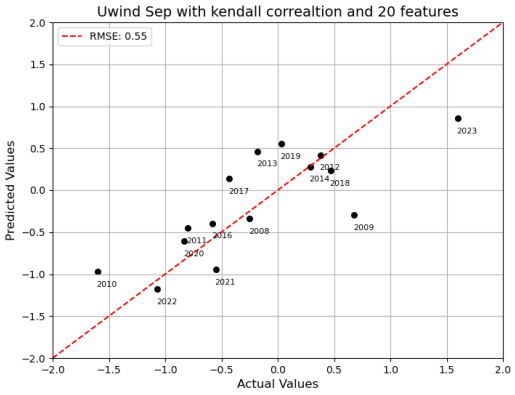

Figure 11.35: Variations in the obs and pred

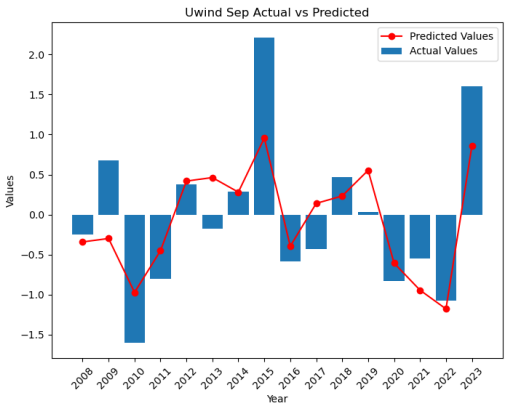

Figure 11.36: Predicted vs Actual

In the figures presented above, a comparative analysis is provided between the Bagging and Boosting models with performance metric as RMSE for the months of June, July, August, and the JJAS (June-July-August-September) average.

11.19 Discussion

Regardless of the performance metric used—whether it is correlation for the MAM period or RMSE for the JJAS period—Boosting consistently outperformed Bagging in capturing the peaks in the testing data.

Chapter 12

COMPARATIVE ANALYSIS OF PERFORMANCE METRICS

In this chapter, we compare the performance metrics of correlation and RMSE for both Bagging and Boosting techniques. Our analysis reveals that different metrics favor different techniques:

By comparing these performance metrics, we gain a nuanced understanding of the strengths and weaknesses of each technique, providing valuable insights into their suitability for different predictive tasks.

12.1 Monthly Correlation and RMSE Values for MAM

Table 12.1: Monthly Correlation and RMSE Values for MAM

Month	**Bagging Correlation**	**Boosting Correlation**	**RMSE Bagging**	**RMSE Boosting**
March	0.93	0.94	0.32	0.31
April	0.91	0.88	0.38	0.32
May	0.75	0.85	0.31	0.33
MAM avg	0.92	0.92	0.39	0.35

12.2 Monthly Correlation and RMSE Values for JJAS

Table 12.2: Monthly Correlation and RMSE Values for JJAS

Month	**Bagging Correlation**	**Boosting Correlation**	**RMSE Bagging**	**RMSE Boosting**
June (J)	0.81	0.76	0.42	0.37
July (J)	0.88	0.83	0.43	0.44
August (A)	0.91	0.88	0.51	0.44
September (S)	0.90	0.85	0.66	0.55
JJAS avg	0.87	0.83	0.49	0.42

12.3 Correlation

Bagging demonstrates superior performance when evaluated using correlation metrics. This suggests that Bagging is more effective in capturing the linear relationships between the predicted and actual values.

12.4 RMSE

Boosting excels when evaluated using RMSE. This indicates that Boosting is more adept at minimizing the overall prediction error, particularly in capturing the peaks and variations in the data.

Chapter 13

EVALUATING MODEL PERFORMANCE WITH F1-SCORE

In the previous chapter, we explored the performance of Bagging and Boosting ensemble methods using correlation and RMSE metrics. This chapter delves deeper into the analysis by employing the F1-score metric. The F1-score offers a balanced view of a model's performance, considering both precision (ability to identify true positives) and recall (ability to capture all actual positives).

13.1 Comparison for MAM

	Metric	MAM	MAR	APR	MAY
0	Correlation	0.92	0.94	0.91	0.85
1	Sensitivity	0.50	0.83	0.50	0.14
2	Specificity	1.00	0.80	1.00	1.00
3	Precision	1.00	0.71	1.00	1.00
4	Negative Predictive Rate	0.67	0.89	0.67	0.60
5	Accuracy	0.75	0.81	0.75	0.62
6	F1 Score	0.67	0.77	0.67	0.25

Figure 13.1: Correlation for MAM

	Metric	MAM	MARCH	APRIL	MAY
0	RMSE	0.31	0.32	0.31	0.35
1	Sensitivity	0.50	0.83	0.75	0.50
2	Specificity	1.00	0.90	1.00	1.00
3	Precision	1.00	0.83	1.00	1.00
4	Negative Predictive Rate	0.67	0.90	0.80	0.67
5	Accuracy	0.75	0.88	0.88	0.75
6	F1 Score	0.67	0.83	0.86	0.67

Figure 13.2: RMSE for MAM

13.2 Comparison for JJAS

	Metric	JJAS	JUN	JUL	AUG	SEP
0	Correlation	0.87	0.81	0.88	0.91	0.90
1	Sensitivity	0.25	0.25	0.43	0.67	0.29
2	Specificity	1.00	1.00	0.89	0.90	1.00
3	Precision	1.00	1.00	0.75	0.80	1.00
4	Negative Predictive Rate	0.57	0.57	0.67	0.82	0.64
5	Accuracy	0.62	0.62	0.69	0.81	0.69
6	F1 Score	0.40	0.40	0.55	0.73	0.44

Figure 13.3: Correlation for JJAS

	Metric	JJAS	JUNE	JULY	AUGUST	SEPTEMBER
0	RMSE	0.42	0.37	0.43	0.44	0.55
1	Sensitivity	0.62	0.38	0.43	0.83	0.86
2	Specificity	0.88	0.88	1.00	1.00	0.78
3	Precision	0.83	0.75	1.00	1.00	0.75
4	Negative Predictive Rate	0.70	0.58	0.69	0.91	0.88
5	Accuracy	0.75	0.62	0.75	0.94	0.81
6	F1 Score	0.71	0.50	0.60	0.91	0.80

Figure 13.4: RMSE for JJAS

Here, we focus on the months considering both bagging and boosting that exhibited the highest correlation and lowest RMSE based on the results from the previous chapter. This targeted approach allows for a more in-depth comparison of F1-score performance between the two ensemble methods on these specific months.

13.3 Discussions

The results suggest that RMSE Metric achieved a higher F1-score in these critical months. This indicates that RMSE excelled in both identifying true positive values (precision) and capturing all actual positive values (recall). This balanced performance of RMSE strengthens the case for RMSE as the preferred permformance metric for our specific prediction task.

Chapter 14

PREDICTING ENSO INDICES FOR 2024 (MARCH-SEPTEMBER)

This chapter outlines the application of the previously developed ensemble models (Bagging and Boosting) for predicting ENSO (El Niño-Southern Oscillation) indices for the months of March to September in 2024. Here, we delve into the specific steps undertaken to generate these predictions.

14.1 Data Used

The data employed for prediction encompasses the period from March 2023 to February 2024 for the months of March-April-May (MAM) average and June 2023 to February 2024 for the June-July-August-September (JJAS) average. This timeframe reflects the most recent data available from the National Centers for Environmental Prediction (NCEP) at the time of prediction.

we leveraged preprocessed data by incorporating trained long-term monthly anomalies and their corresponding normalized values. Additionally, we employed the weights obtained from trained auto-encoders for predictive modeling. Moreover, predictor sets, derived during the training phase, were utilized for forecasting the indices of 2024.

14.2 Bagging Predictions

Table 14.1: ENSO Index Predictions for 2024 (March-September) - Bagging

Month	Variables	Predictions	Lead	Columbia CS Prediction
March(M)	SLP+UWND	0.28	1	1.2(obs)
April(A)	SLP+UWND	0.08	2	0.8(obs)
May(M)	GPH	0.15	4	0.4
MAM avg	SLP+UWND	0.18	1	0.76
June(J)	AIR TEMP	-0.25	5	0.03
July(J)	UWND	-0.66	5	-0.26
August(A)	VWND	-0.06	10	-0.48
September(S)	AIR TEMP	-0.50	8	-0.62
JJAS avg	AIR TEMP	-0.14	4	-0.33

14.3 Boosting Predictions

Table 14.2: ENSO Index Predictions for 2024 (March-September) - Boosting

Month	Variables	Predictions	Lead	Columbia CS Prediction
March(M)	AIR TEMP	0.81	1	1.2(obs)
April(A)	UWND+VWND	0.00	2	0.8(obs)
May(M)	SLP+UWND	0.08	3	0.4
MAM avg	SLP+UWND	0.33	1	0.76
June(J)	SLP+VWND	-0.15	5	0.03
July(J)	UWND	-0.10	5	-0.26
August(A)	SLP+VWND	0.63	9	-0.48
September(S)	UWND	-0.56	8	-0.62
JJAS avg	UWND	-0.33	5	-0.33

14.4 Evaluation of 2024 ENSO Index Predictions

To assess the accuracy of our 2024 ENSO index predictions, we employed the forecasts generated by the International Research Institute for Climate and Society (IRI) at Columbia University as a benchmark reference. This approach allowed us to compare our model's performance with a well-established and widely used ENSO prediction system.

14.5 Discussions

Boosting exhibited promising performance in capturing the trends observed during 2024, suggesting its potential for real-time forecasting. Additionally, incorporating data till May 2024 might have yielded even more accurate predictions for the June-July-August-September (JJAS) period.

Chapter 15

CONCLUSION AND FUTURE WORK

15.1 Conclusion

In this study, we successfully validated the approach proposed by Saha and Nanjundiah (2020) [7] for predicting ENSO indices using deep learning. Our results exhibited broad agreement with their findings, lending further credence to the effectiveness of this method.

Furthermore, our evaluation of ensemble learning techniques revealed interesting insights. We observed that correlation emerged as a more suitable metric for assessing the performance of Bagging models, while F1 score and RMSE proved to be more effective for Boosting models. This suggests that the optimal choice of evaluation metric can be influenced by the specific ensemble method employed.

15.2 Future Research Directions

This study paves the way for further exploration of advanced ensemble methods and deep learning architectures for ENSO index prediction. Here are some promising avenues for future research:

1. **Stacked Encoders for Feature Selection:** Investigating the use of stacked encoders for feature selection prior to ensemble learning could potentially improve model performance by identifying the most informative features from the input data.

2. **Superensemble Prediction:** Exploring the integration of multiple ensemble models through a superensemble approach could potentially lead to even more robust and accurate predictions by combining the strengths of various ensemble methods.

3. **CNN-LSTM Models:** Implementing and evaluating Convolutional Neural Network (CNN)-Long Short-Term Memory (LSTM), known for their effectiveness in time series forecasting, could provide valuable insights into their suitability for ENSO prediction compared to the current models explored in this study.

BIBLIOGRAPHY

[1] Moumita Saha, Pabitra Mitra, and Ravi S Nanjundiah. Autoencoder-based identification of predictors of indian monsoon. *Meteorology and Atmospheric Physics*, 128:613–628, 2016.

[2] Yanan Guo, Xiaoqun Cao, Bainian Liu, and Kecheng Peng. El niño index prediction using deep learning with ensemble empirical mode decomposition. *Symmetry*, 12(6):893, 2020.

[3] Kalpesh Ravindra Patil, Takeshi Doi, Venkata Ratnam Jayanthi, and Swadhin Behera. Deep learning for skillful long-lead enso forecasts. *Frontiers in Climate*, 4:1058677, 2023.

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

[5] Gai-Ge Wang, Honglei Cheng, Yiming Zhang, and Hui Yu. Enso analysis and prediction using deep learning: A review. *Neurocomputing*, 520:216–229, 2023.

[6] Haoyu Wang, Shineng Hu, and Xiaofeng Li. An interpretable deep learning enso forecasting model. *Ocean-Land-Atmosphere Research*, 2:0012, 2023.

[7] Moumita Saha and Ravi S Nanjundiah. Prediction of the enso and equinoo indices during june–september using a deep learning method. *Meteorological Applications*, 27(1):e1826, 2020.