Overcoming SPB: Improving Predictability of ENSO Events

Joseph Karpinski, Alexandra Mendoza, Sean Lydon, and David McCurdy

June 13, 2024

Abstract

El Niño Southern Oscillation (ENSO) events are the most well-known interannual climate signal and have many effects on global climate that are important to be able to predict. These events account for variability in precipitation and stronger weather events during an El Niño period and thus influence important fields like agriculture, outdoor entertainment, infrastructure, and watershed levels. However, due to the chaotic nature of atmosphere and ocean interactions in early spring known as the Spring Predictability Barrier (SPB), the ability to predict these oscillations and replicate the dynamics using data before this period is very difficult. In order to assist the prediction and ability to prepare for shifts we aim to utilize common variables to study these events along with sea surface salinity. Furthermore, we utilize salinity, pressure, and temperature data and predict the ENSO index before the SPB. To do this we have developed a convolutional neural network that will accept our pre-SPB data in a specific structure and output a prediction for the next several months. By training our data in a rolling fashion the model was run through several configurations of the same neural net with slightly changed parameters. However, even with varied parameters the model produced random, non-decreasing loss and such was unable to learn from the features presented. Given these results, we show that the use of sea surface salinity to improve prediction past the SPB is unproven. However, based on initial data analysis there is still promise to the hypotheses and thus needs further study.

1 Introduction

The El Niño Southern Oscillation (ENSO) is the most powerful interannual climate signal which influences regional climates, and extreme weather events across the globe [Cai et al., 2021]. It's characterized as self-organizing quasi-periodic patterns in ocean-atmosphere dynamics over the equatorial Pacific. This oscillation consists of an irregular warm and cold period known as El Niños and La Ninas, respectively, which vary from the long-term mean state of the system. These ENSO events have been observed to be a key factor in assessing future risk of anthropogenic climate change [Le, 2023]. Forecasting ENSO events has been of interest to scientists, and governments since the 1970s (NOAA, 2024). The methodology scientists have employed to develop accurate forecasting of ENSO events is diverse. However, using sophisticated data analysis techniques such as Empirical Orthogonal Functions (EOFs)–developed by pioneering atmospheric physicist Edward Lorenz in the 1950s–has proven to be an effective technique in ENSO forecasting when performed on key climate variables such as sea surface temperature (SST) and sea level pressure (SLP). While these methods of ENSO forecasting are effective, they fail to produce reliable forecasts with data sourced before the Spring Predictability Barrier (SPB). The SPB is a well-documented phenomenon in ENSO prediction that is characterized by a stark reduction in ENSO forecast accuracy before boreal spring [Jin et al., 2023]. Insight into how to reduce the SPB and achieve accurate forecasting of ENSO events before the SPB is of much interest in the field, and a recent study–which incorporated Sea Surface Salinity (SSS) into a linear dynamic model–found a significant reduction of the SPB [Pang et al., 2023]. This suggests that key changes to upper ocean dynamics occur when accounting for salinity in the buoyancy fluxes of the model and that these subtle dynamics play a key role in contributing to the complexity that creates the SPB. Here we take an interest in how accounting for SSS when performing EOF analysis on climate data can reduce the SPB in a predictive neural network. With our predictive model, we will assess the improvement in SPB reduction from a forecast using only traditional climate variables of SST and SLP, and analyze predictive performance when only using SSS.

2 Methods

2.1 Data Formatting

The data that we have procured for studying ENSO prediction is a combination of three datasets: sea surface temperature and sea surface pressure from the NCEP-NCAR reanalysis made by NOAA's Physical Sciences Laboratory, sea surface salinity from the European Space Agency, and MVI ENSO index data from the National Center of Atmospheric Research (NCAR). The temperature, salinity, and pressure variables within the datasets contain values for the entire globe with a temperature and pressure grid size of 2.5 x 2.5 degrees and an SSS grid size of 1.5 x 15 degrees. The SST and the SSP data were recorded monthly and have a time span of 55 years from January 1969 to March 2024. The SSS data was recorded every 2 weeks and therefore needed to be modified to fit the same time step as the other data by averaging every 2 week period. The temporal span however was far smaller than that of the temperature and pressure data with only about 120 months' worth of data. Thus, in order to have a full range of variables for each time point the temporal spans for pressure and temperature were cut down to match the salinity data.

The data for the global sea surface salinity, temperature, and pressure was also cut down to only have data about the equatorial Pacific, the area that is the most indicative of an ENSO event. This allows us to use less processing power and is more focused on the area pertaining to the study.

The salinity, temperature, pressure, and ENSO index data came from different sources so the first we thing we needed to do was merge them. We chose to merge data into a 4-dimensional numpy array, *data*, such that *data* $[i, j, k, l]$ = variable i at (time, latitude, longitude) index (j, k, l) where i indicates the variable according to

i	0	1	2	3
Variable	Salinity	Temperature	Pressure	ENSO index

Table 1: Variable indices of merged numpy array.

Note the ENSO indices are defined for the globe at each instance in time, that is

$$\text{data}[3, j, k_1, l_1] = \text{data}[3, j, k_2, l_2] \; \forall \; k_1, l_1, k_2, l_2. \tag{1}$$

The temporal and spatial range for the data selected by

1. Setting lower bound for time, latitude, and longitude equal to the maximums of time, latitude, and longitude among the variables

2. Setting upper bounds for time, latitude, and longitude equal to the minimums of time, latitude, and longitude among the variables

This still provided enough information: 120 temporal, 67 latitude, and 63 longitude records for each variable. After finding the range, we noticed that not all the spatial data aligned, so we used spatial points of the pressure data and interpolated spatial data of the other variables using nearest neighbor interpolation. Since the spatial data for the temperature and salinity is more dense than pressure, lots of it was dropped.

3 Model

3.1 Convolutional Neural Net

Our chosen method to study ENSO predictability is a Convolutional Neural Network (CNN). CNNs are traditionally used to predict images or time series. CNNs take input and condense them to determine the most important features and use those features to create a prediction. The biggest benefit of this model is its convolutional layers. The convolutional layers use a filter that slides over the input to determine the most important features and reduce the input. Our problem requires a multidimensional input and a 1-dimensional output, which can be easily implemented with a CNN. We named the CNN for this project ENSO_test_Model. ENSO_test_Model will take in our data in a four-dimensional structure and condense the data using a 3-dimensional convolutional filter. The model has 3 convolutional layers, 3 pooling layers, and 2 fully connected layers. The convolutional layers use 3-dimensional filters, each with the shape (4,3,3) to filter through the layer input and output a feature map. The feature map then goes through the pooling layers which compress the feature map by calculating the average. The output of the last pooling layer is then inputted into the fully connected layers, which resembles a feedforward neural network. In between the convolutional layers, pooling layers, and fully connected layers, there is a rectified linear activation function (ReLU). ReLU is a piecewise function defined as:

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$$

It further condenses the inputs. The last activation layer in the neural network is the hyperbolic Tan function, which condenses the output to values between (-1,1). Because the ENSO indices range from (-3,3), they can easily be condensed using the hyperbolic Tan function so that they can be used to test against outputs from the neural network. The structure of the CNN is:

1. First convolutional layer (implemented using torch.nn.Conv3d())
2. First ReLU layer (implemented using torch.nn.ReLU())
3. First Pooling layer (implemented using torch.nn.AvgPool3d())
4. Second convolutional layer (implemented using torch.nn.Conv3d())
5. Second ReLU layer (implemented using torch.nn.ReLU()
6. Second Pooling layer (implemented using torch.nn.AvgPool3d())
7. Third convolutional layer (implemented using torch.nn.Conv3d())
8. Third ReLU layer (implemented using torch.nn.ReLU())
9. Third Pooling layer (implemented using torch.nn.AvgPool3d())
10. First fully connected layer (implemented using torch.nn.Linear())
11. ReLU layer (implemented using torch.nn.ReLU())
12. Second fully connected layer (implemented using torch.nn.Linear())
13. Hyperbolic tangent activation layer (implemented using torch.nn.Tanh())

3.2 Model Training

To train our model we split the data into 70 percent training data, reserving the other 30 percent for testing the fully trained model. Because the ENSO index is an inter-annual climate signal, there is a need for a large amount of temporal data in order to generate reasonable output. However, salinity data is hard to come by due to it not being as prominent of an oceanic feature as temperature and pressure and we were only able to obtain data for the 10-year period January 2009 through December 2019. This only gives us approximately 120 data points to work with which is not much on the scale of convolutional neural nets. To remedy this lack of data we implemented a sliding cutoff for our training such that the training data is sectioned off into pieces of 18 months and then paired with the next 9 months afterward for validation of the model output. Then each of those slices of time are moved forward one month to create another batch of training data which is repeated until we reach the end of the data. In this way, we hope to train the model to capture the dynamics of the current training slice and learn to project the ENSO index for the next 9 months. By doing this we expand our number of training data sets from 12 to 66.

4 Results

4.1 PCA Cluster Analysis

The principle components of the climate data were analyzed in order to assess the effective clustering of data points based on their ENSO index value. The data was isolated by region, and an ENSO 'tolerance' level was implemented. This tolerance corresponds to the MVI ENSO index value that will be considered La Nina (red) or El Nino (blue). The default value used was ± 0.5, however changing this value resulted in greater or less clustering of the data, as can be seen in Fig. 2.

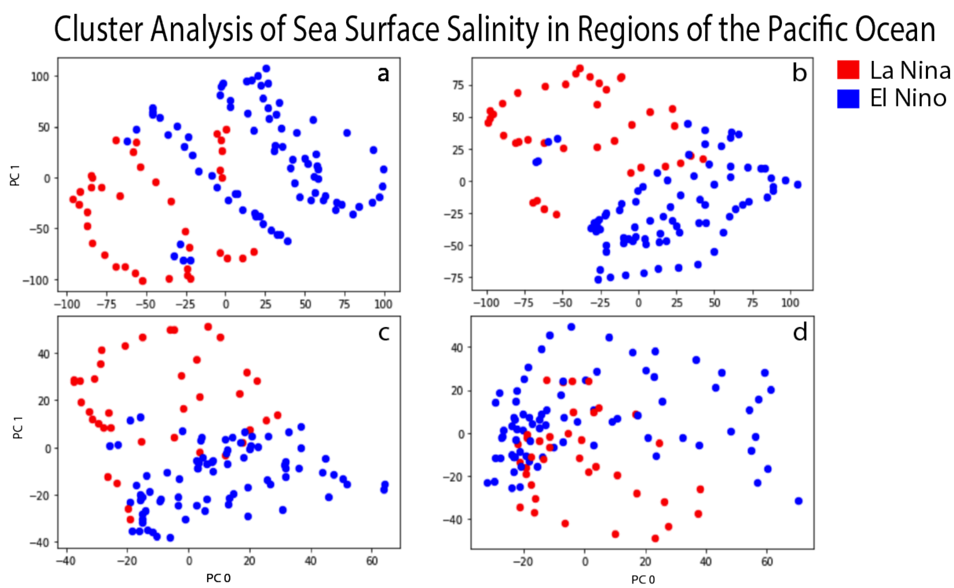

Figure 1: Scatter plot of 1st and 2nd principle components of sea surface salinity for four different regions: (a) Pacific, (b) Equatorial Pacific, (c) Western Equatorial Pacific, (d) Eastern Equatorial Pacific.

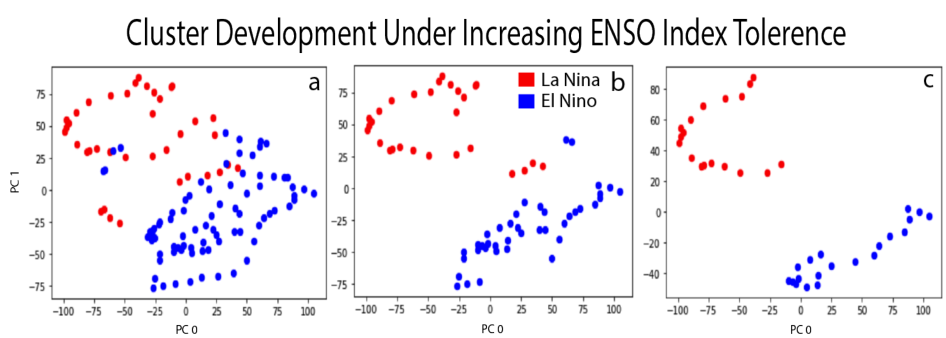

Figure 2: Scatter plot of 1st and 2nd principle components of sea surface salinity for three different ENSO tolerance levels: (a) 0.5, (b) 1, (c) 1.5.

4.2 CNN Training Loss

Multiple configurations of the same neural net were used to achieve a steadily decreasing loss during training however each resulted in a different cyclic pattern as shown below.

In the first run-through of the model, the neural network had the settings 9-month slider size, and a 9-month output for the ENSO index, and used the Adam optimizer. The loss for this setup was cyclical, with a new cycle about every 60 epochs, as shown below in Figure 3. Ideally, the loss would decay exponentially. A decay in the plot would depict the model as outputting values that are closer to the expected value, meaning that the model is learning. The cyclical pattern for the losses in this version of the model shows that the model is not learning, and therefore would not be helpful in solving our main question. Fortunately, many changes can be made to the model that can improve its predictability without needing to change the entire structure of it.

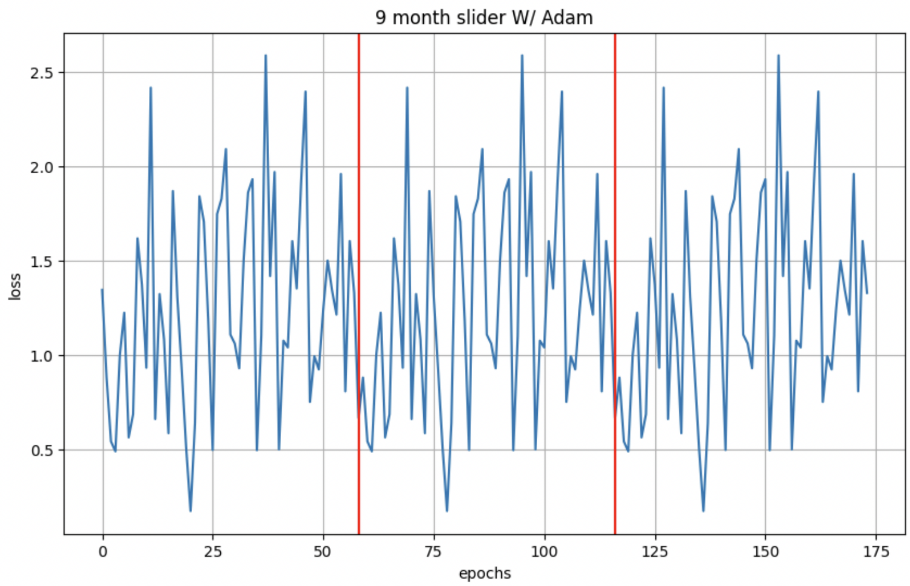

Figure 3: The plot of losses for the first 3 cycles through training data with a 3x3 convolutional grid and Adam optimizer. The output for this model is 9 months of prediction with an 18-month input. This shows a cyclical pattern after each run-through of data which is marked by vertical red lines. While the pattern does repeat, the loss varies wildly and does not seem to decrease as the model trains. This specific configuration of the model stays between 0.5 and 2.5 with a mean of 1.2358.

The next method we attempted was increasing the slider window, changing the optimizer, and increasing the size of the convolutional filter. By increasing the size of the slider, there would be fewer inputs into the CNN, which may help with overfitting. The optimizer was changed from Adam to AdamW, which would generally give better results due to its ability to apply weight decay directly to parameters. And lastly, the increase in the filter would condense the inputs in each layer more than it would for a (4,3,3) filter.

Figure 4 shows the results from running this model variant. Once again, the plotted losses are cyclical, with cycles beginning about every 65 epochs. The cyclical pattern once again makes this model unfit for testing predictions. Moreover, there is more variation among the losses, which shows that this model is not better at predicting outputs. However, there is a positive change compared with the original model. The lower end of the range for loss is much closer to 0 than it was in the previous model variation. In the first model, there were only a few points under 0.5 loss, but in this new model, there are more points less than 0.5. Moreover, there are many points very close to 0 loss. This suggests that at least one of the changes made the model better. So for the next model variation, we decided to keep the optimizer the same.

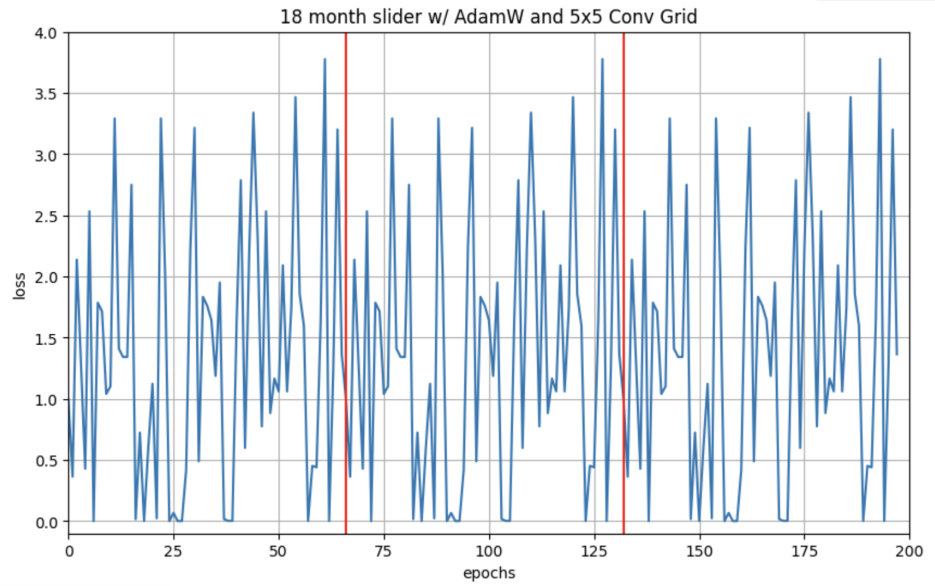

Figure 4: The plot of losses for the first 3 cycles through training data with dropout layers added in between each activation and pooling layer with a 5x5 convolutional grid and AdamW optimizer. The output for this model is a single month of prediction with an 18-month input. This shows a cyclical pattern after each run-through of data which is marked by vertical red lines. While the pattern does repeat, the loss varies wildly and does not seem to decrease as the model trains.

For this next variation, we implemented a 1-month slider window, a batchsize=1, and the AdamW optimizer. The 1-month slider was implemented to create more samples for the CNN to learn from. By using batches, the input data is randomized and split into different groups, which generally helps with loss. Unfortunately, once again, the calculated losses were cyclical, as shown in Figure 4. However, there were improvements from the previous variation. Similar to the previous variation, there were more losses close to 0. This pattern repeats for the next model variation.

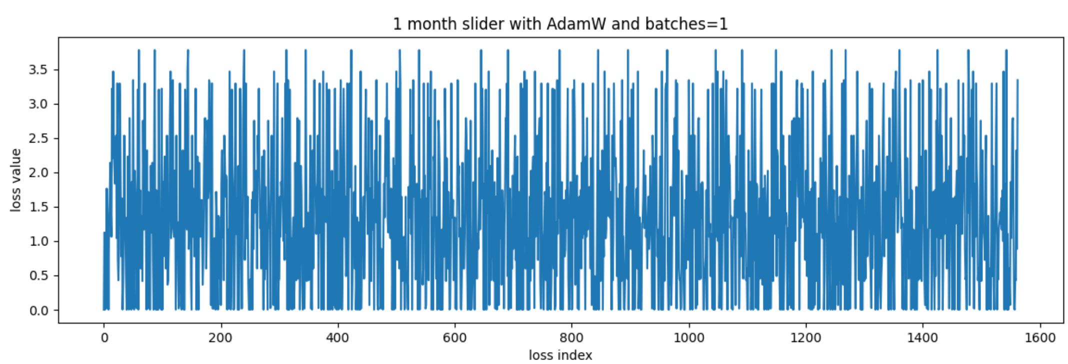

Figure 5: The plot of losses using the AdamW optimizer, setting the output to the ENSO index of 1 month and using batchsize=1

In this last variation we tested, we kept the 1-month slider and the AdamW optimizer, but then removed the batches and added dropout layers. The dropout layers, written after the convolutional layers, turn off nodes in the neural net based on a manually input probability p. Figure 5 below shows the results of running this model. The losses were still cyclical, however, the losses are centered closer to 0. The large spikes in the plot are more separated from the rest of the losses, which suggests that those losses are not as common as the previous models, which is an improvement.

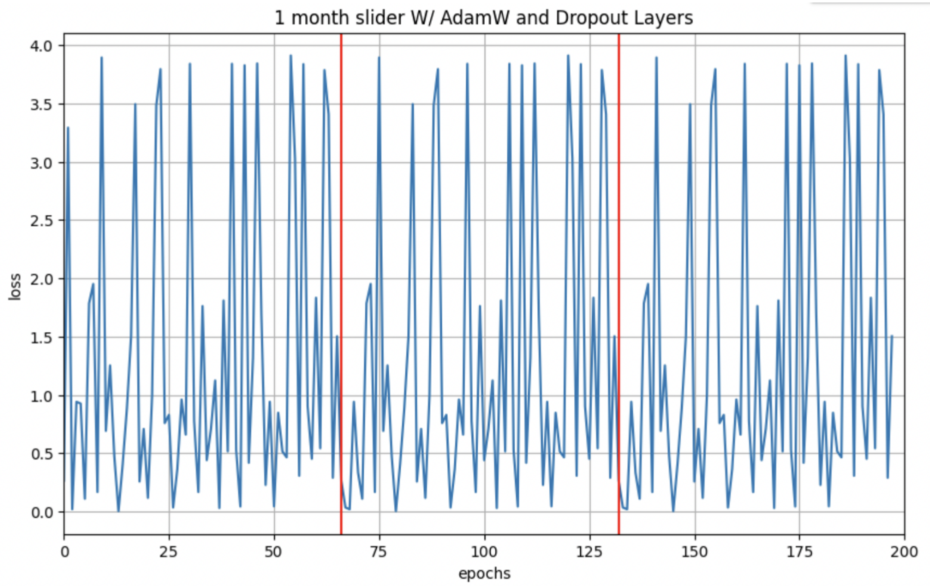

Figure 6: The plot of losses for the first 3 cycles through training data with dropout layers added in between each activation and pooling layer with a 3x3 convolutional grid and AdamW optimizer. The output for this model is a single month of prediction with an 18-month input. This shows a cyclical pattern after each run-through of data which is marked by vertical red lines. While the pattern does repeat, the loss varies wildly and does not seem to decrease as the model trains. This model however has a generally low loss amount with small spikes up to values around 3.

The most significant issue with our model is that it is not learning. With each variation, the model continued to produce cyclical patterns, showing that it did not improve predictions as the model trained. This means that the model, as it stands, is unable to be used to predict ENSO indices based on sea surface salinity, sea surface temperature, and sea surface pressure.

5 Conclusion

Through the use of data analysis and data reduction, several datasets containing sea surface pressure, sea surface temperature, and sea surface salinity were cleaned and analyzed using principal component analysis to find differences in correlation with the ENSO index. Then the data was interpolated and compiled into a single data structure such that the variables could be placed into channels for our neural net. A convolutional neural net was created to accept these fields along with the monthly ENSO index in an attempt to have it output a predicted ENSO index for 9 months following the input data. It was shown in the results section that the neural net was unable to train off the data and the losses would repeat cyclically with outputs not improving as the model trains.

The inability to predict ENSO events prior to the SPB is somewhat standard and these results are congruent with current research. While the principle component analysis shows that sea surface salinity has distinct features depending on the ENSO index, it is clear that the neural net is unable to detect them and thus is not able to predict. As shown in the results section, the cyclic result shows that as the batches are trained there is no learning taking place and thus the prediction is inaccurate. However, due to the chaotic nature of the phenomenon being studied, it is unclear as to whether this is an error inherent to the model or if this is due to the general inability to predict ENSO events.

6 Discussion

While these conclusions fall short of the intended goal of creating a strong predictive model, there is still much to discuss about both possible causes for the data shown above and future work on the same topic.

A significant obstacle in our study was the lack of long-term sea surface salinity data causing there to be a severe lack of data to use in training the convolutional neural net. With only ten years of data for a cycle that takes several years to repeat there is not a substantial amount of variability and history for the model to learn from. More data could allow for a more extensive training of the model which ideally would output much more accurate predictions.

For future research, we hope to obtain more extensive SSS data and train our model on more extensive and varied data. However, insights into the dynamics of the ENSO system can be observed in the context of the PCA analysis. We see in Fig. 1 the ENSO clustering is significantly reduced in panel d, while a strong cluster shows in panels a, b, and c. Since all of the SSS data contained in panels c and d are also contained in panels a and b, it suggests that the clusters we are seeing a most likely a result of SSS modes in the western equatorial Pacific and not in the eastern equatorial Pacific. The SPB occurs in April through June, roughly the same time as the East Asian monsoonal season over the western equatorial Pacific, and this suggests that precipitation and thus freshwater fluxes over this region play a key role in understanding the nuanced dynamics that lead to the SPB. While our data set was too sparse for our neural network to produce robust forecasting, this study achieved keen insight into the potential of SSS as an essential climate variable in ENSO forecasting through our PCA analysis, and our results lay a foundation for future studies to consider monsoonal conditions in the western Pacific and how understanding these monsoon events may give deep insight into reduction of the SPB.

Some caveats include only looking at monthly average values of our variables- if that, because we are unsure if salinity values are monthly average values or just measurements taken on the first day of each month due to the lack of metadata. In addition, using nearest neighbor interpolation may have skewed the data when merged causing inaccuracies that could be avoided with another interpolation method such as multidimensional cubic splines.

As for the CNN itself, ideally, the changes made to each variation would have been one at a time and then tested against other similar variations to find the best method. For example, running different models that have different optimizers and changing another aspect of the model from there. For future work, it would be beneficial to have access to a faster GPU to run the code. Those of us who ran the code on our laptops waited at least 30 minutes to run enough epochs to create results. With a faster GPU, we would be able to do as many tests as we would like for each model variation and be allowed to increase the number of epochs. A large number of epochs would also help the model learn.

7 Contributions

Sean Lydon

I cleaned and managed the sea surface pressure and sea surface temperature datasets by taking out missing data, slicing the geospatial data to be over the same area for the variables, and converting data to S-mode analysis. I also did a great deal of initial data analysis by finding the principal components of the data and plotting the data along them to find the separation between El Niño years and La Niña years. I also helped code the neural net and fixed errors in the code along with brainstorming ideas for how to train the CNN with the limited data we have. I also wrote large portions of the final report including the abstract, model training, conclusion,

Alexandra Mendoza

I researched different types of neural nets to use for our predictive model and looked into the general structure of convolutional neural networks. I constructed the CNN based on these findings and based on my meetings with Conrad Ainslie, our TA for AM 170B. I also coded the training loop for the model and wrote the code for the loss plots. And lastly, I helped create the sliding function that split the data into different-sized windows. As for the contents of this paper, I contributed to the model section, results section, and discussion section.

David McCurdy

I obtained and cleaned the sea surface salinity data by slicing specific regions, clearing Nans, and replacing with means and reformatting data for S-mode analysis. I obtained the MVI ENSO index data we used and converted it into an appropriate data file for our project. I then wrote a general function that performs PCA, and ICA analysis on a given data set and returns every 2-pair permutation for PC scatter plots and its numerical values. I also wrote a general function that takes in a data set and correlates every PC with the ENSO index and returns various plots. I also did background research in the field to find novel and relevant scientific questions for our group to address around ENSO prediction, sourced the papers in our project document and wrote the introduction for our project document, wrote the cluster analysis sections, and the climate dynamic discussion paragraph and made minor edits to various other parts of the document.

Joseph Karpinski

I cleaned, merged, and interpolated data since they originally came from different sources. I then made a netCDF object containing all the data with updated metadata. I explained this in more detail in the beginning of section 2 with some caveats in the last paragraph of section 6. I also made high-level neural net pseudo code to possibly use a basis in Alexandra's implementation shown below. All of this is available in my project folder on GitLab here.

```
#tin: time length of input to nn in number months(number of time indicies), try
tin=18
#tout: time length of output to nn in number months(number of time inicies), try
tout=9
#There are 120 time indicies from 0 to 119, where 0 indicates
#Jan 2010 and 119 idicates Dec 2019, I think we should test
#our nn in most recent times because I assume they would more
#similar to present climate
#ttest: time length of test in number months(number of time indicies),
#try the last 2 years, that is
ttest=24

if tout>ttest:
    raise ValueError('ttest must be >= tout to run at least one test')
if (tin+tout >= 120):
    raise ValueError('tin+tout must be <120 to run at least one test and one training session')
first_test_index=120-ttest-tin
nn.train(data[:,0:first_test_index,:,:])
#errors: shape=(ttest-tout+1,tout) such that errors[j,k] = the enso error of
#nn.eval(data[:,first_test_index+j:first_test_index+j+tin,:,:])[k] where
#len(nn.eval(data[:,first_test_index+j:first_test_index+j+tin,:,:]))=tout
#you might have to do some reshapeing to get errors how I described above
errors=nn.test(data[:,first_test_index:first_test_index-tout,:,:])
avg_error=np.mean(errors)
```

References

[Cai et al., 2021] Cai, W., Santoso, A., Collins, M., Dewitte, B., Karamperidou, C., Kug, J.-S., Lengaigne, M., McPhaden, M. J., Stuecker, M. F., Taschetto, A. S., et al. (2021). Changing el niño–southern oscillation in a warming climate. *Nature Reviews Earth & Environment*, 2(9):628–644.

[Jin et al., 2023] Jin, Y., Meng, X., Zhang, L., Zhao, Y., Cai, W., and Wu, L. (2023). The indian ocean weakens the enso spring predictability barrier: Role of the indian ocean basin and dipole modes. *Journal of Climate*, 36(24):8331–8345.

[Le, 2023] Le, T. (2023). Increased impact of the el niño–southern oscillation on global vegetation under future warming environment. *Scientific Reports*, 13(1):14459.

[Pang et al., 2023] Pang, Y., Jin, Y., Zhao, Y., Chen, X., Li, X., Liu, T., and Hu, J. (2023). Sea surface salinity strongly weakens enso spring predictability barrier. *Geophysical Research Letters*, 50(23):e2023GL106673.