GEOPHYSICAL RESEARCH LETTERS

Supporting Information for“Subseasonal Forecasts of Opportunity Identified by an Explainable Neural Network”

Kirsten J. Mayer [1]and Elizabeth A. Barnes [1]

[1]Department of Atmospheric Science, Colorado State University, Fort Collins, CO, USA

Contents of this file

Introduction Here we provide information about the choice of lead day for the predictand and a more detailed description of artificial neural networks (ANNs), layerwise relevance propagation (LRP), and k-means clustering. In addition, we include composite z500 figures for both positive and negative correct predictions, a timeseries of z500 anomaly predictions, as well as a confusion matrix and a table of additional skill metrics for all and the 10% most confident predictions.

Text S1: Reasoning behind Prediction of Lead Day 22

Previous research has shown that MJO impacts on the North Atlantic Oscillation occur approximately 5-15 days following phases 2-3 and 6-7 (Lin et al. 2009; Cassou 2008). Henderson et al. (2016) show that MJO impacts over the North Atlantic are statistically significant out to 20 days. In addition, Barnes et al. (2019) illustrate a causal connection between the MJO and NAO on the order of 15-20 days; however, they hypothesize that the MJO may still impact the NAO after the 20 days due to the autocorrelation of the NAO and MJO. Therefore, we evaluated the ANN on a variety of leads from 5-28 days. We found that the network performed well across leads within week 3 (days 15-21), but started to decrease in skill after lead day 22. A lead of 22 days is, therefore, used for our analysis, as it was one of the later leads with higher skill. While daily anomalies are used here, the ANN can also be used to predict a smoothed z500 anomaly (e.g. 7-day running mean anomalies). We find that the network performs similarly well for both weekly and daily anomalies, and therefore, use daily anomalies for this analysis.

Text S2: Artificial Neural Networks (ANNs)

In this analysis, we use an artificial neural network (ANN) as a tool for subseasonal forecast of opportunity identification where Figure 1 shows the ANN architecture used for this analysis. The architecture includes an input layer (teal and brown nodes) and is followed by two hidden layers (grey nodes) and an output layer (red and blue nodes). The network is tasked to predict the sign of the geopotential height at 500hPa (z500) at a point in the North Atlantic (40°N, 325°E, white 'X' in Figure S1) given tropical OLR anomalies. The input layer receives vectorized OLR anomalies so that each input node represents an OLR anomaly from a single grid point. The output layer returns two values, one in each output node, where the nodes represent the sign of the z500 anomaly. The node with the larger value signifies the predicted sign of the z500 anomaly.

The network architecture is set up so that each node in a layer receives a value from the preceding layer. The value of a single node in a layer is calculated through a weighted sum of the incoming values in the preceding layer with an added bias (equation 1).

$$z_j = \sum_i w_{ij} x_i + b \tag{1}$$

In equation 1, j denotes the node for the value being calculated in a given layer and i denotes a node from the preceding layer. Therefore, w_{ij} signifies the weight connecting the ith and jth node and x_i represents the value of node i. b denotes the added bias term. A nonlinear transformation is then applied to z_j (equation 2). For this analysis, the Rectified Linear Unit (ReLU; equation 2) is used as the nonlinear activation function.

$$f(z_j) = max(0, z_j) \tag{2}$$

Both equation 1 and 2 are repeated for each node in the layer, which results in a single value ($f(z_j)$) for each node. These new calculated values are then be passed to the following layer and the process continues. At the final layer, a softmax activation function is applied:

$$\tilde{y_i} = \frac{exp(x_i)}{\sum_j exp(x_j)} \tag{3}$$

where x_i represents the presoftmax value for output node i , the denomenator is the sum of the exponential of all the presoftmax output values, and $\tilde{y_i}$ represents the predicted output value for the ith output node. This function converts the raw values in the output layer into values that sum to one. By doing so, the output values then represent an estimation of likelihood that an input belongs to a particular category. We refer to this estimation of likelihood as "model confidence". A confident prediction will, therefore, have a value closer to one.

The architecture used here is often referred to as a fully-connected ANN since all the nodes from one layer are connected to all the nodes in the next layer. We have used the simplest ANN architecture that provided a relatively high accuracy since this set-up is sufficient for this application (two hidden layers). Additional information on ANNs can be found in Nielsen (2015) or Goodfellow et al. (2016).

In addition to the model architecture, there are also important parameters to specify for the training process. This includes the type of loss function, batch size, and number of epochs. The loss function estimates the accuracy of the predicted value to the actual value. For this example we use categorical cross entropy (equation 4) where $\tilde{y_i}$ is the predicted value of the ith node in the output layer and y_i is the actual value.

$$loss = -\sum_i y_i log(\tilde{y_i}) \tag{4}$$

This loss function assigns error to the ANN output so that larger errors are punished more than smaller errors due to the logarithmic transformation. The weights and biases of the neural network are updated using the gradient of the loss function through back propagation (a series of chain-rule operations). An incremental step, defined here by the Adam method (Kingma & Ba, 2014), is then taken in the direction of greatest decrease along the loss function, in attempt to minimize the loss.

In addition, we also use ridge regression (L_2 norm penalty) to limit the magnitude of the coefficients. The penalty forces the model to combine values from many grid points for each prediction. We apply this additional penalty because individual grid points on the globe are spatially correlated with nearby points.

The weights and biases are updated after each batch, a subset of the training data. A batch size of 256 is used. After the network iterates through the entire training dataset using a batch of 256 (an epoch), the process is repeated again for a defined number of epochs. In this analysis, we use 50 epochs, however, we apply early stopping (ending the training before 50 epochs) if the validation loss increases for 2 epochs in a row. This is done in order to reduce overfitting on the training data.

Text S3: Multinomial Logistic Regression

Multinomial logistic regression (MLR) is a form of logistic regression that can be used for a multi-class problem. Using ANN terminology, the MLR architecture can be described as an input layer and an output layer, where the output values are passed through the softmax activation function. The ANN architecture used for this analysis is similar, but also includes two hidden layers. These hidden layers in the ANN make the ANN more complex than MLR and able to account for additional nonlinearities. As ANN and MLR methods are similar to one another, we compare the accuracies between the two methods for reference. We find that the ANN and multinomial logistic regression models have similar accuracies for the validation data, but the

ANN performs much better (over 20% higher accuracy) on the testing data than MLR. However, regardless of accuracies, we use an ANN for this paper, instead of MLR, since an ANN makes the methods more generalizable to other more complex nonlinear systems.

Text S4: ANN Explainability - Layerwise Relevance Propagation

To understand how a trained network makes its prediction, explainability techniques can be used to extract and visualize what the network has learned. In this paper, we use an explainability technique known as layerwise relevance propagation (LRP; e.g. Bach et al. (2015); Montavon et al. (2019)). To apply LRP, a single sample of interest is initially passed through the trained network (with frozen weights) to obtain a prediction. Using the output values without the softmax activation, the output node with the highest value (the predicted category) is back-propagated through the network using the following rule

$$R_i = \sum_j \frac{a_i w_{ij}^{+} + max(0, b_j)}{\sum_i a_i w_{ij}^{+} + max(0, b_j)} R_j \tag{5}$$

where i denotes the node of the layer to which the relevance is being back-propagated to while j denotes the node of the layer in which the relevance is from. R_i is therefore, the relevance translated backward to the ith node and R_j is the relevance of the jth node. The weight connecting the ith and jth nodes is denoted as w_{ij}^{+} where the $+$ signifies that only the positive weights are used for back propagation. Lastly, a_i signifies the value of the ith node (post activation function) and b_j signifies the bias term of the jth node. The above relevance equation is for the LRP-$\alpha\beta$ method where $\alpha = 1$ and $\beta = 0$. This type of LRP method only propagates information associated with positive weights. In other words, only the information that positively contributed to the prediction is propagated backward.

For relevance back-propagation from the first hidden layer to the input layer, the following equation is used:

$$R_i = \sum_j \frac{w_{ij}^2}{\sum_i w_{ij}^2} R_j \tag{6}$$

At the input layer, the relevance values for each node can then be used to create a heatmap of relevance where more relevant nodes have larger values. This process is then repeated for every prediction of interest, resulting in a unique relevance heat map for each prediction. These maps show the relevant regions from the input sample that positively contributed to the prediction.

For more information on LRP as well as other neural network explainability techniques, see Toms, Barnes, and Ebert-Uphoff (2020) and McGovern et al. (2019).

Text S5: K-Means Clustering

K-Means cluster analysis (Hartigan & Wong, 1979) is used to group the correct prediction LRP maps to further explore relevant regions for enhanced prediction skill. K-means clustering categorizes input data into a user specified number of groups. The method iteratively assigns the given data to centroids based on the minimum squared Euclidean distance, where each data point is assigned to the closest centroid. The centroids are moved to the center of their assigned data points after an iteration and then the process begins again, for a user specified number of iterations. The data points associated with each centroid are part of that centroid's cluster.

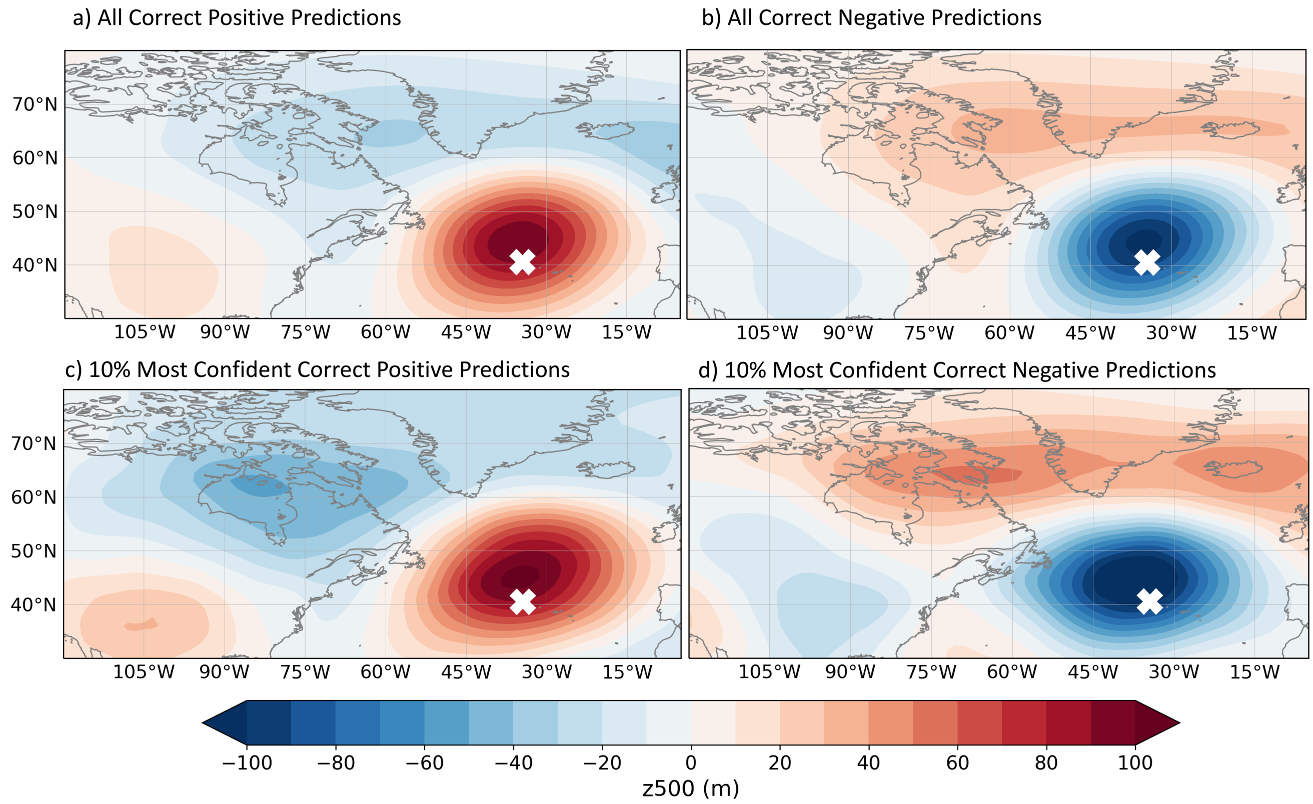

a) All Correct Positive Predictions
b) All Correct Negative Predictions
c) 10% Most Confident Correct Positive Predictions
d) 10% Most Confident Correct Negative Predictions
70°N
60°N
50°N
40°N
105°W
90°W
75°W
60°W
45°W
30°W
15°W
−100
−80
−60
−40
−20
0
20
40
60
80
100
z500 (m)

Figure S1. North Atlantic z500 composite: Composite of z500 anomalies for (a,b) all and the (c,d) 10% most confident predictions for correct (a,c) positive and (b,d) negative predictions. Shading represents the composite z500 anomalies and the white 'X' denotes the location of the ANN prediction over the North Atlantic (40°N, 325°E).

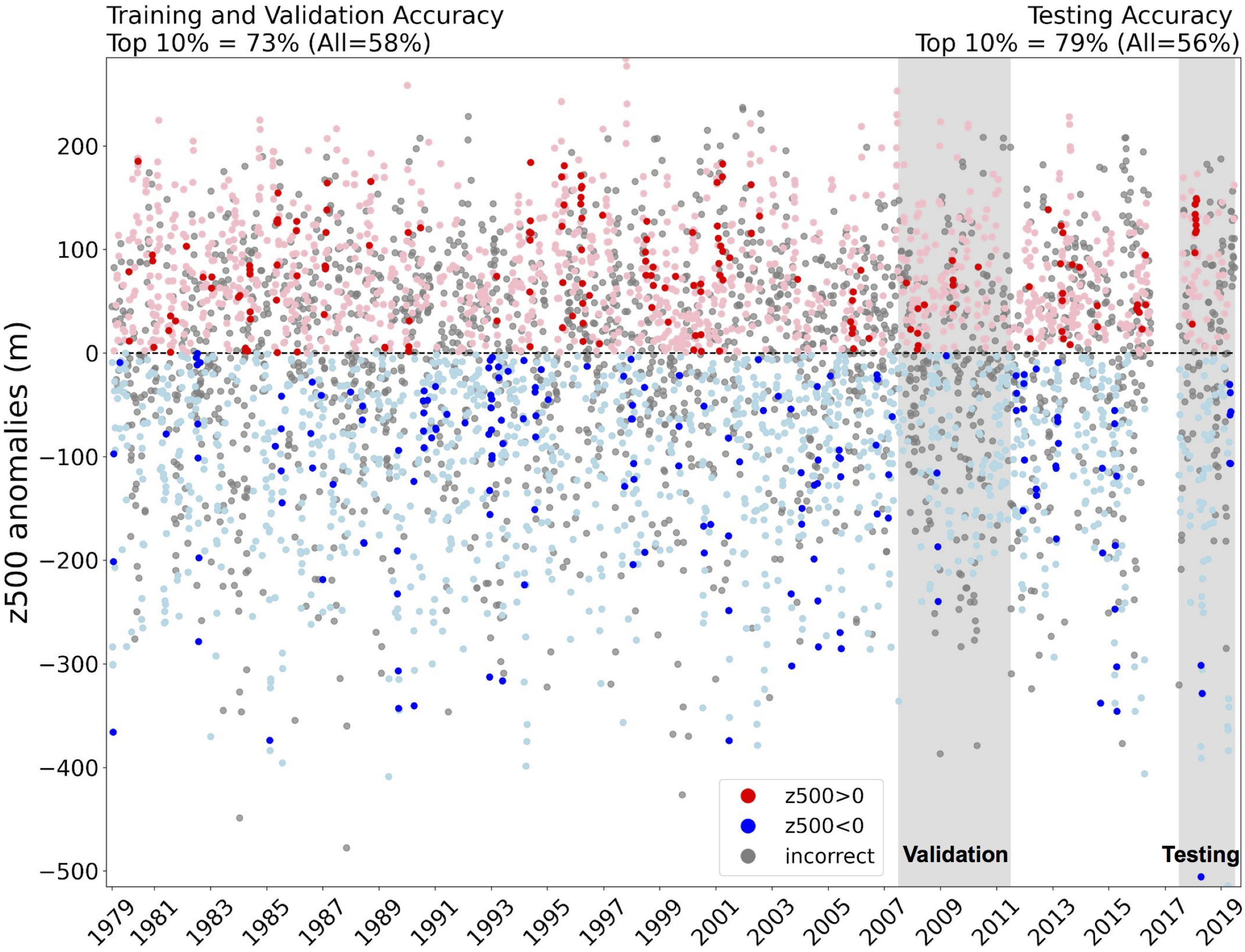


Figure S2. Timeseries of ANN z500 predictions: Timeseries of z500 anomalies shaded by the sign of the ANN predictions. Blue dots represent correct negative predictions, red dots represent correct positive predictions, and dark colored dots indicate forecasts of opportunities (i.e. 10% most confident predictions). Grey dots represent incorrect predictions. The vertical grey shading from 2007-2011 highlights the time period used for validation and the vertical grey shading from 2017-2019 highlights the time period used for testing. The accuracies for training and validation as well as testing data for forecasts of opportunities and all predictions are given in the top left and right, respectively.

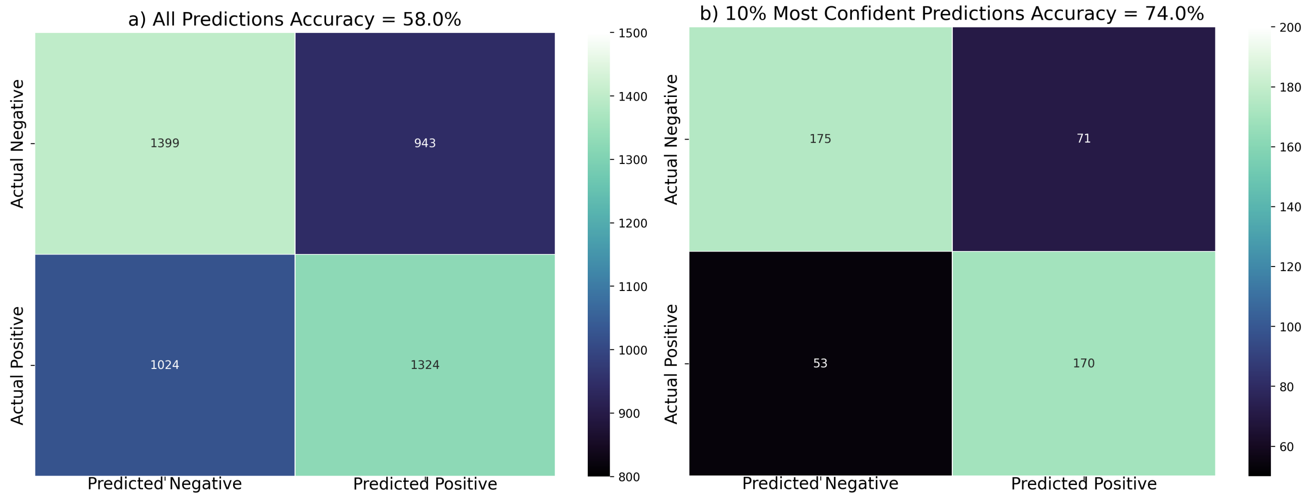
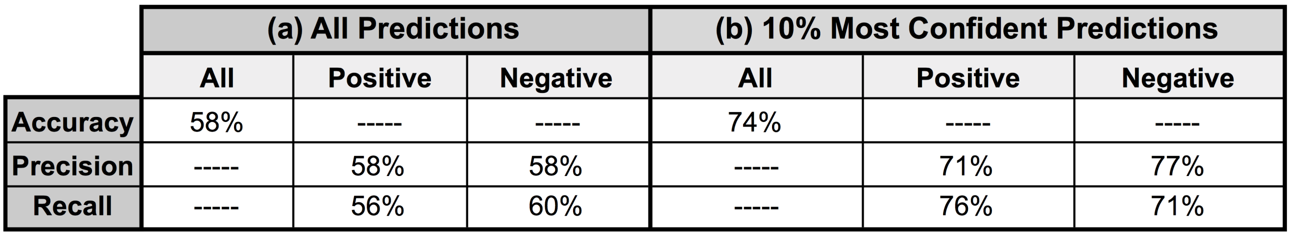


Figure S3. Confusion Matricies: Confusion matrix of training, validation, and testing data for (a) all predictions and (b) the 10% most confident predictions, where the accuracy is located at the top of each plot and the shading and the values inside each box represents the sample size for each category.



	(a) All Predictions			(b) 10% Most Confident Predictions		
	All	**Positive**	**Negative**	**All**	**Positive**	**Negative**
Accuracy	58%	-----	-----	74%	-----	-----
Precision	-----	58%	58%	-----	71%	77%
Recall	-----	56%	60%	-----	76%	71%

Table S1. Additional Skill Metrics: Table of accuracy, precision, and recall for (a) all predictions and (b) the 10% most confident predictions using training, validation, and testing data.