Seasonal Forecasting Hackathon Summary

Ankur Mahesh

June 26, 2020

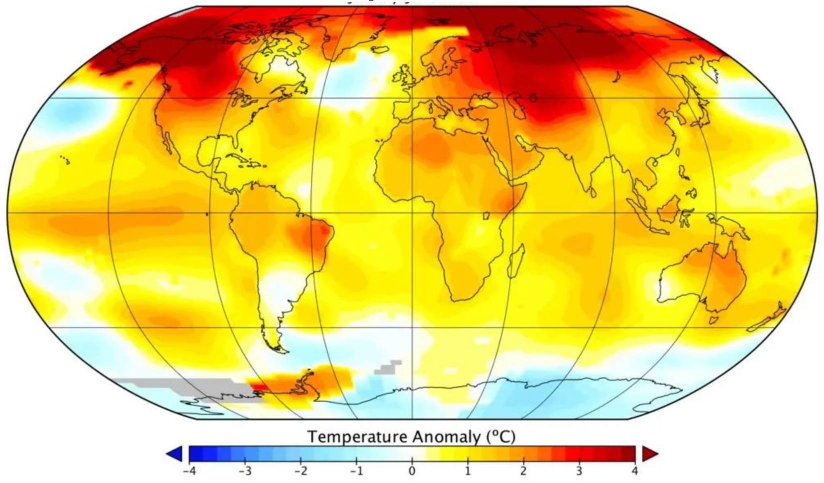
This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

Overview of the hackathon

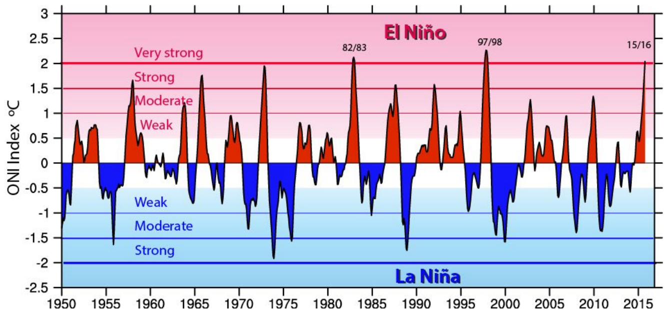
- El Niño is a cycle of warm and cold temperatures in the equatorial Pacific that affects seasonal weather
- It is measured by the *Niño3.4 Index:* rolling 3-month average of sea surface temperatures in the equatorial Pacific

- We will forecast El Niño 1-6 months ahead of time with machine learning
- How do we learn effectively from using the variety of ***data sources*** and **machine learning** models?
- How does predictability of El Niño compare to that of land temperatures?

Predictor Data: surface temperature



Target Data:



Most Common Questions

What considerations go into making a **train/ test** split?

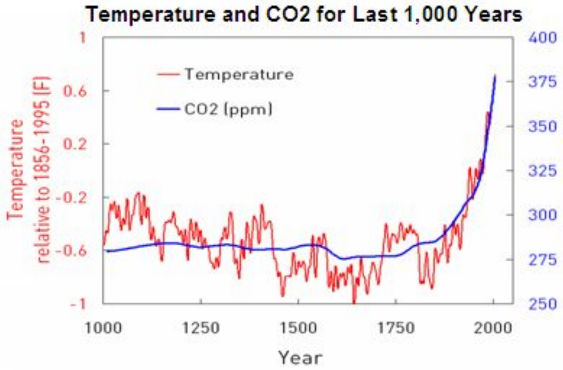
- Anthropogenic warming trend
- Autocorrelation between consecutive months
- Climate models offer another source of training data
- More uncertainty with observations pre-1979

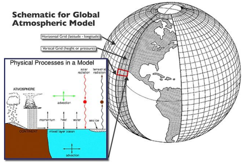
Which machine learning models leverage the **spatial** nature of the input?

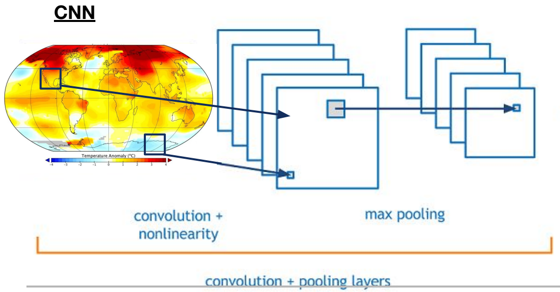
- The input is a global **grid**: latitude by longitude
- Convolutional neural networks use this spatial organization of grid cells
- Other ML methods treat each grid cell independently

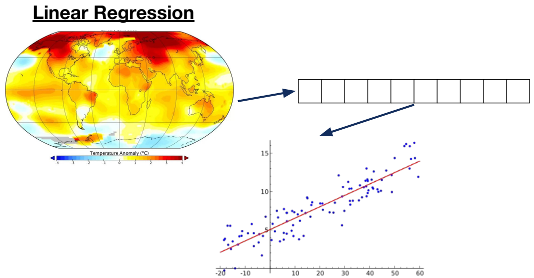
How can I define a custom neural network architecture? How can I learn from time series?

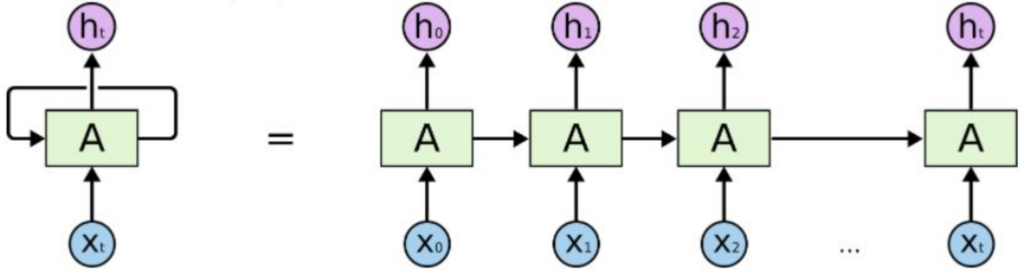
- PyTorch requirement: you need to know the size of the extracted features from the convolutional layers in order to define the fully connected layers











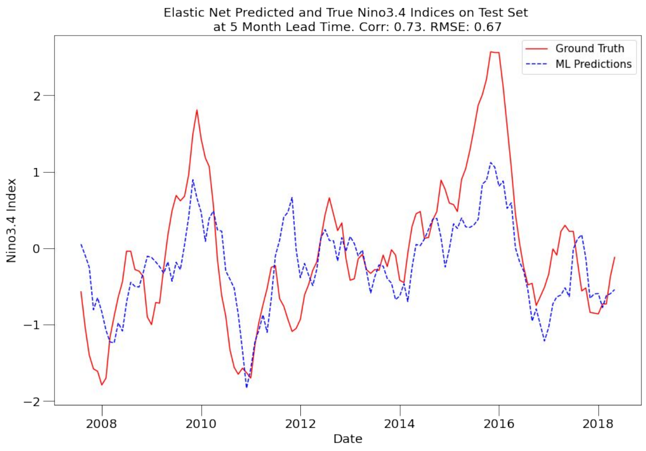
El Niño Seasonal Forecasting

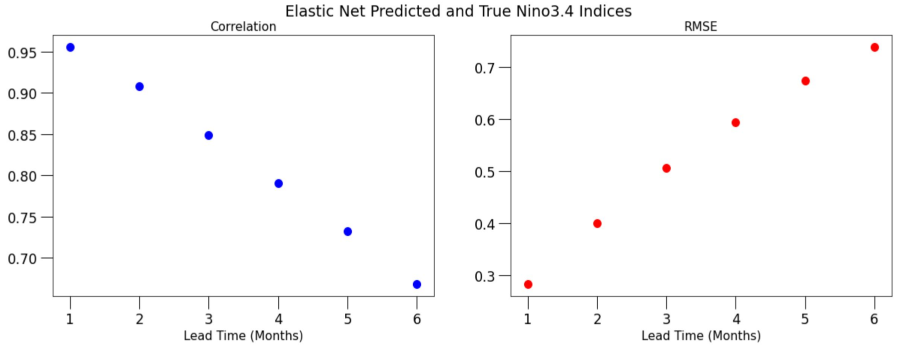
Best model

Team #6: Piyush Garg, Yuanyuan Xu, K S S Sai Srujan, Sarah Kanee, Xian Wu

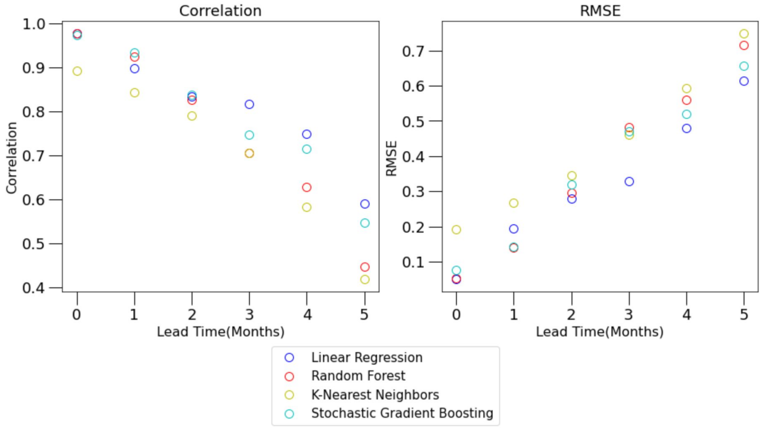
Summary of models applied:

- Linear Regression
- Support Vector Machines
- Decision Tree
- Random Forest
- K-Nearest Neighbors
- Stochastic Gradient Boosting
- Elastic net (with L1 and L2 regularization)
- CNNs

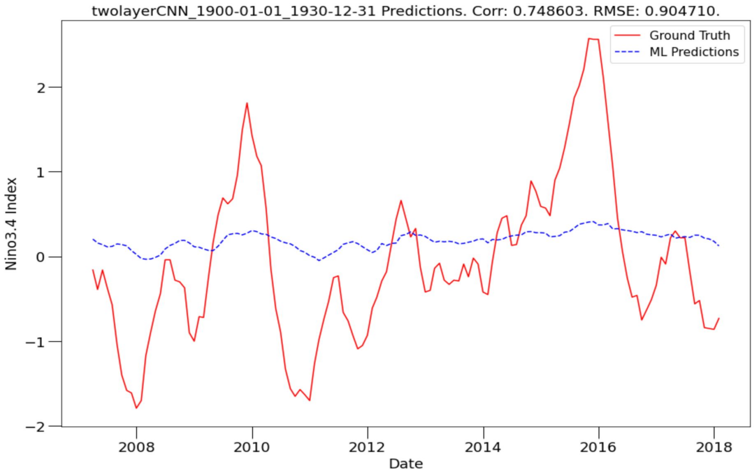


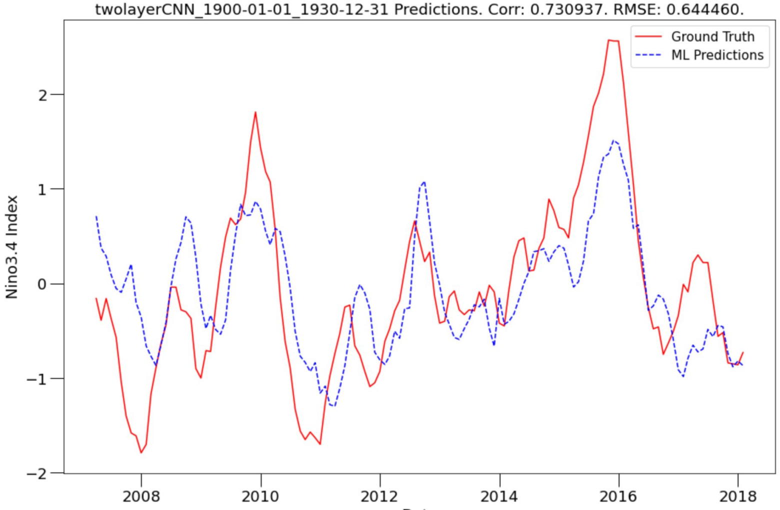


#2 performance of various models

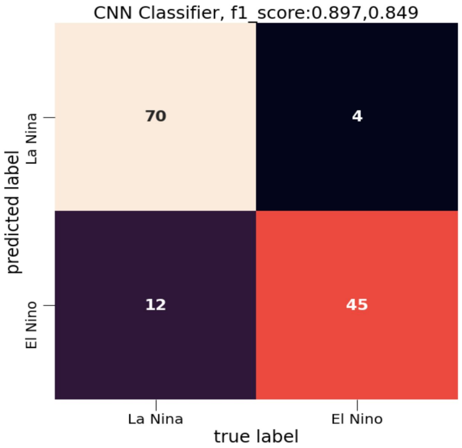


#3 Non-detrended(top panel) Vs detrended data(bottom panel)



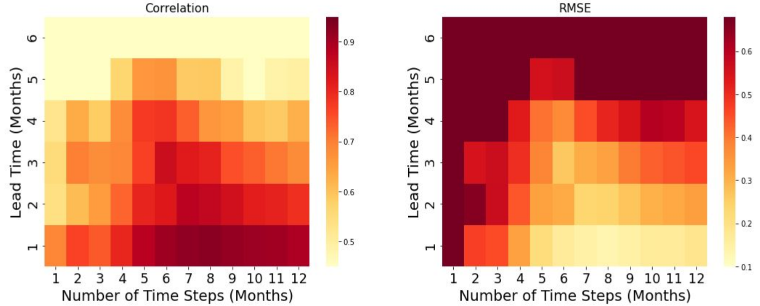


#4 CNN classifier with an Hit rate of 87.78%

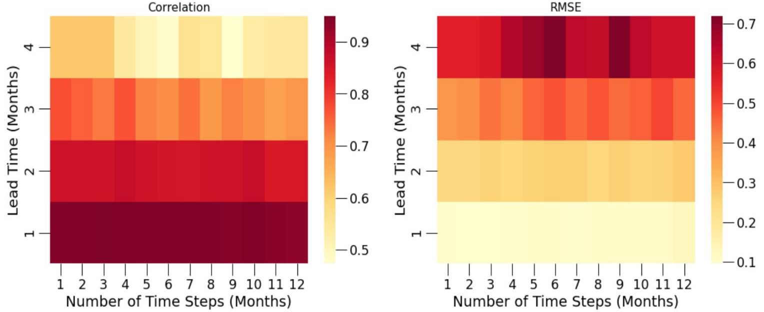


#7 Prediction skill as a function of various models (LR,RF,CNN)

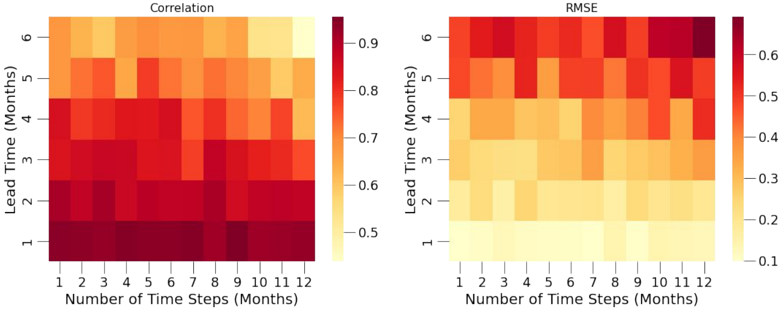
Linear Regression Model



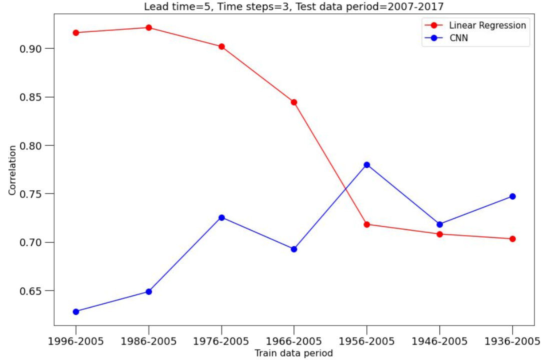
Random Forest



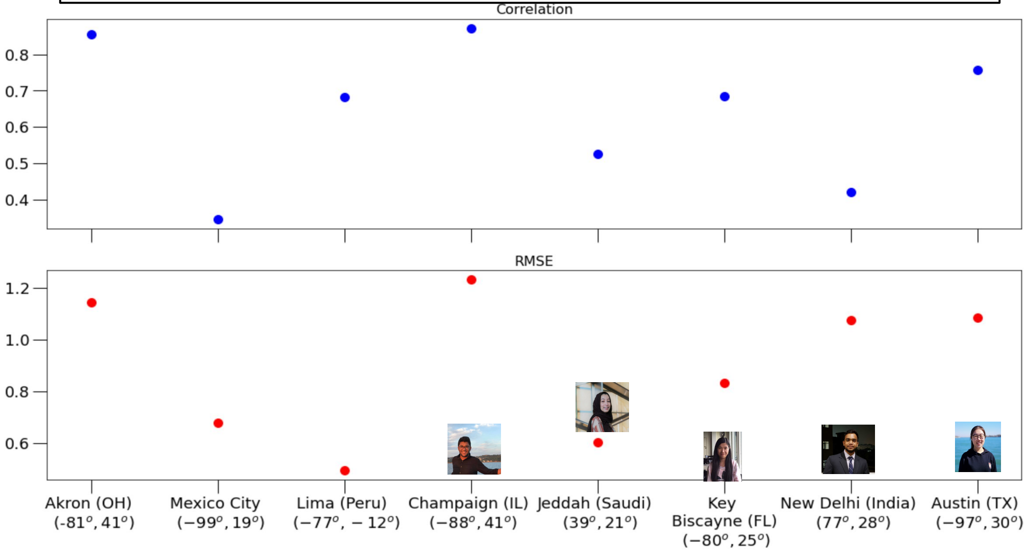
2-Layer Convolution Neural Network



11 Influence of data quality



9 Linear Regression Predicted 2m Air Temperatures with Global Temps as Predictors (2007-2017)



Team 66: <El Niño>

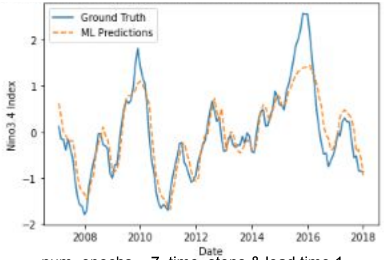
Team Member: **Deborah Khider, Tse-Chun Chen, Connor Aghili, Cora Frederick**

Applications to Exercise 5- 7

Methods

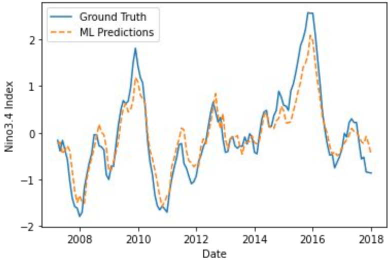
- Two Layer CNN Notebook Default
- **Auto-sklearn** & **Auto-Pytorch**: Both automatically searches for the ‘right’ MLing algorithm for a new MLing dataset and optimizes hyperparameters!

Two Layer CNN: Corr: 0.92 RMSE 0.37



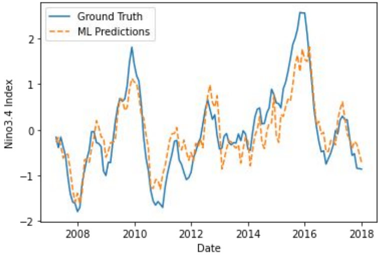
num_epochs = 7, time_steps & lead time 1
Train GCMS 1860-2100

Auto-PyTorch: Corr 0.96 RMSE 0.12



Time_steps = 3, Leat time =1
Train: 1960-2005

Auto-Sklearn: Corr 0.91 RMSE 0.18

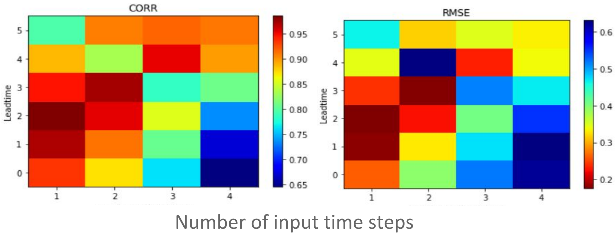


Time_steps = 3, Leat time =1
Train: 1960-2005

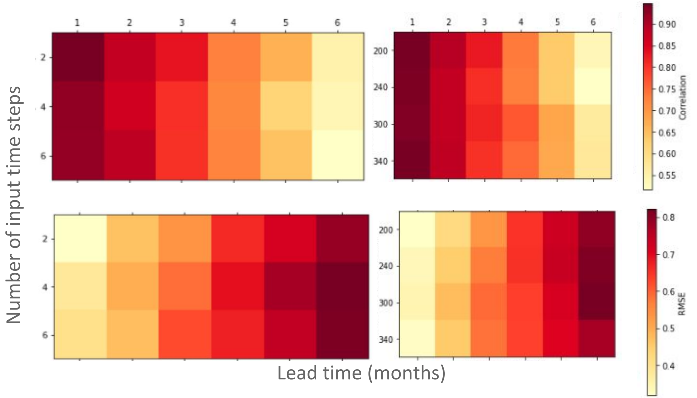
Major Takeaways:

➢ ***Auto-PyTorch compared to Two Layer CNN does extremely well!***

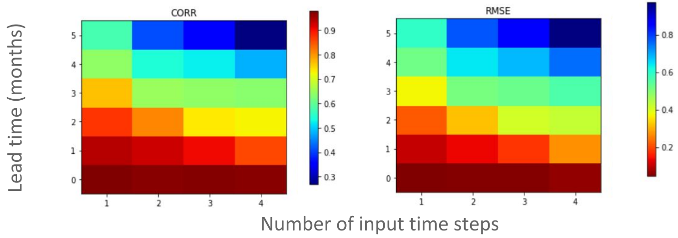
➢ ***And more inputs is beneficial only if you’re doing predictions for longer leadtime***



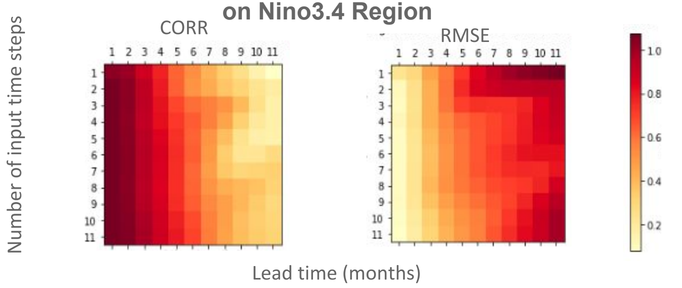
Two Layer CNN



Auto-Sklearn



Linear Regression with era5 dataset on Nino3.4 Region



Team 66: <El Niño>

Team Member: Deborah Khider, Tse-Chun Chen, Connor Aghili, Cora Frederick

Applications to Exercise 8

Methods: CNNs and Auto-Pytorch

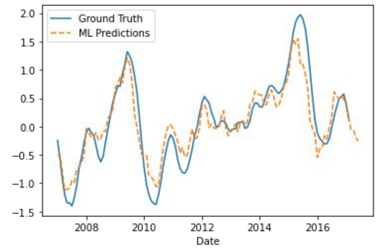
An issue which comes up with MLing and NNs is Optimization

- **How many layers**
  - Performed better with less layer
- **How long to train**
  - Increasing the training period helped with making predictions at long lead times
- **Train Data Is Extremely Important**
  - where it's provided from such as observations or models affect accuracy
- **Other notable variables**
  - Lead time
  - Time step

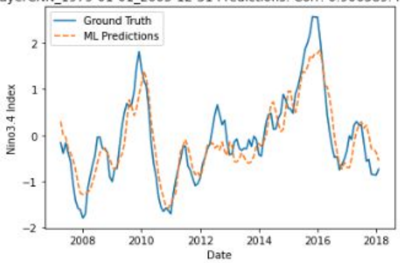
	AutoPyTorch	2-CNN	1-CNN
Corr	0.94	0.906	0.85
RMS	0.29	0.40	0.4

different data used and parameterization per model

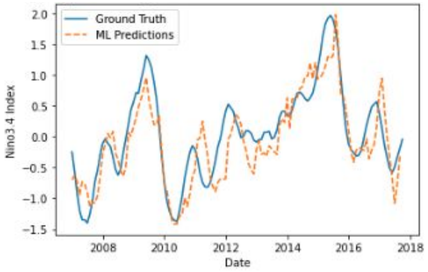
Auto-Pytorch Regression Predicted and True Nino3.4 Indices on Test Set at 5 Month Lead Time. Corr: 0.94. RMSE: 0.29



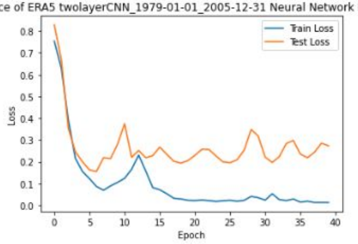
ERA5 twolayerCNN_1979-01-01_2005-12-31 Predictions. Corr: 0.906389. RMSE: 0.404233.



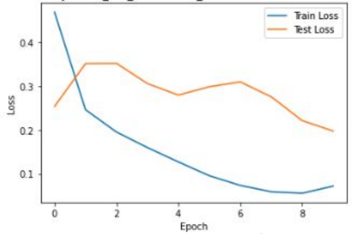
OnelayerCNN_MPI_1860-01-01_2200-12-31 Predictions. Corr: 0.859326. RMSE: 0.405592.



Performance of ERA5 twolayerCNN_1979-01-01_2005-12-31 Neural Network During Training



Performance of twolayerCNN_MPI_1860-01-01_2200-12-31 Neural Network During Training



A Great way to visualize Performance of Data is with Train and Test Loss Curves

Team 38: <El Niño>

Team members:

Iacopo, Suso Peña-Izquierdo, Aheli Das, Pedro Llanillo

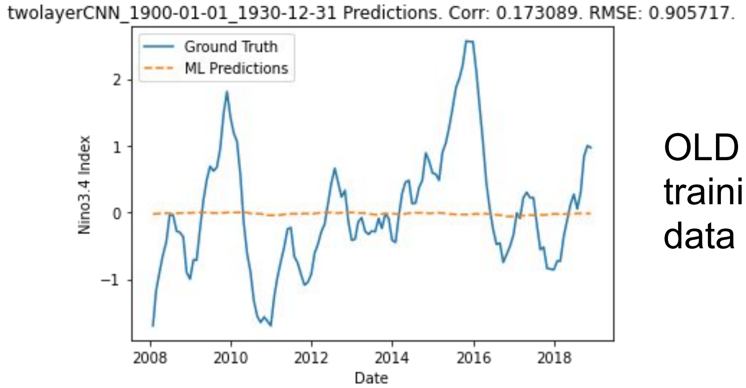
Climate model = MPI
Train period = 1960-2005
Lead time = 5
Num input time steps = 6

MODELS	CORRELATION	RMSE
CNN	**0.59**	**0.6**
CNN + LSTM	0.55	0.64
CNN (Conv3D)	**0.48**	**0.72**

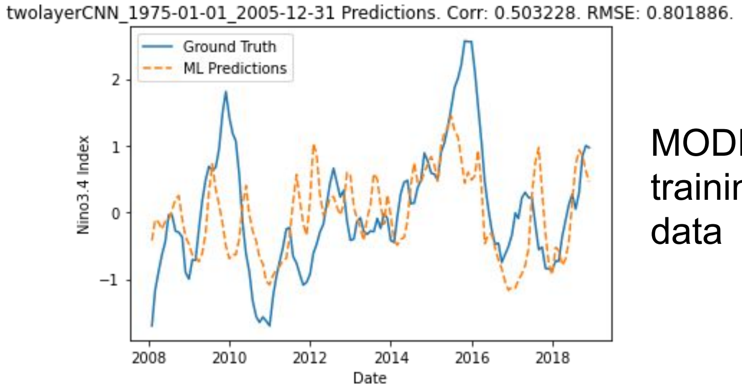
Conclusions:

- Selection of training set region and temporal period appears as important features
- Temporal dimension does not seem to be very relevant
- Important overfitting always present. Simpler models best.

Team 38: <El Niño>



OLD training data

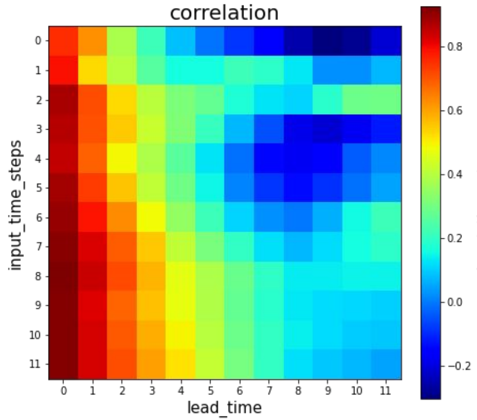


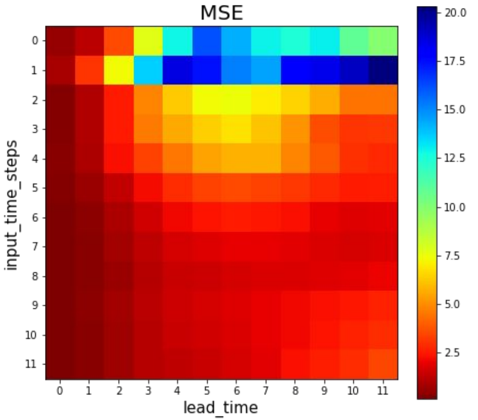
MODERN training data

Training with OLD (low quality) data has a clear impact on performance

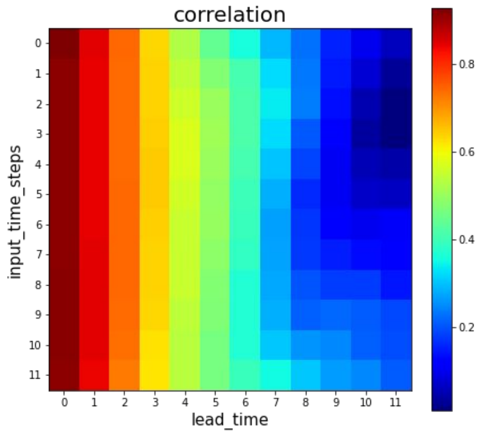
Team 38: <El Niño>

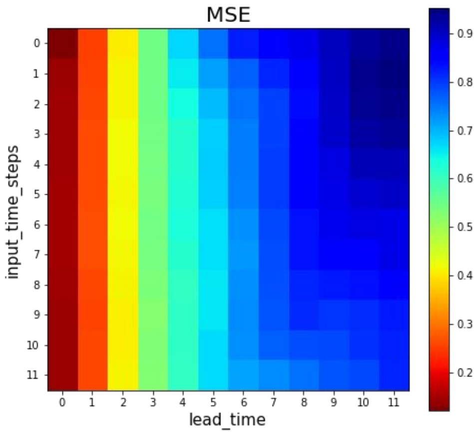
Linear regression



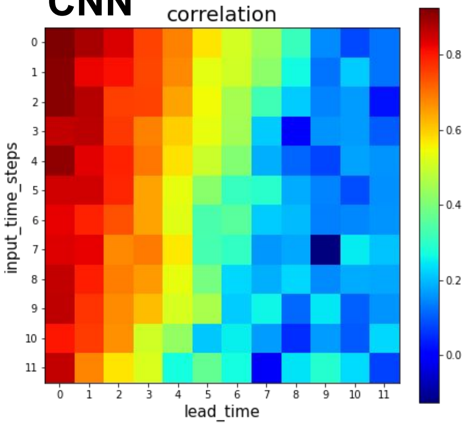


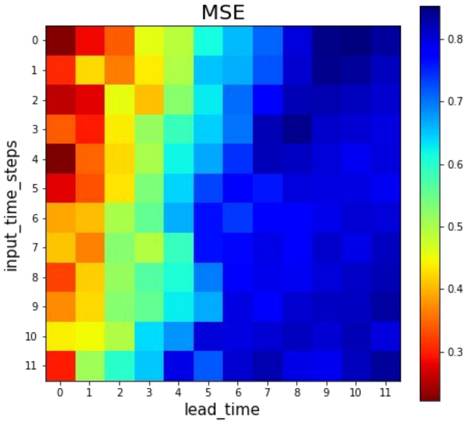
Random forest regression





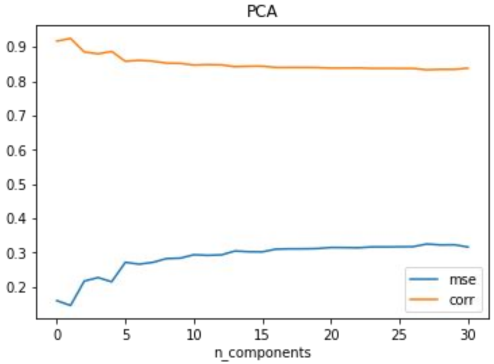
CNN



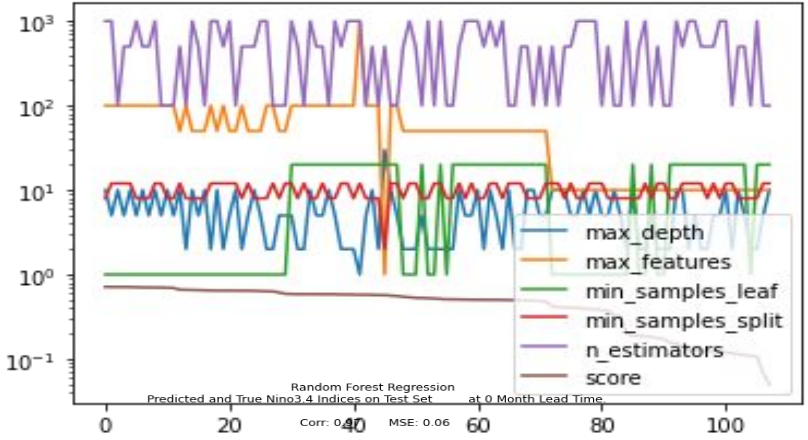


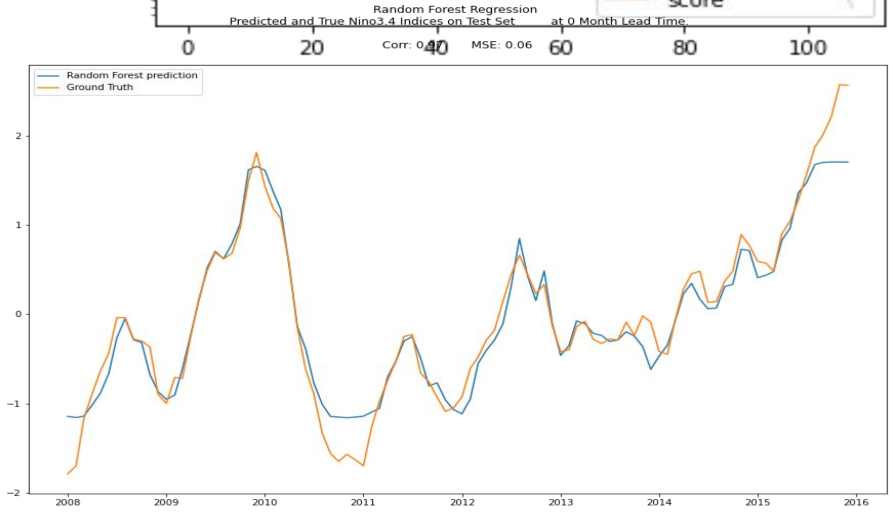
Team 38: <El Niño>

Changing PCA number of components affects the performance of linear regression



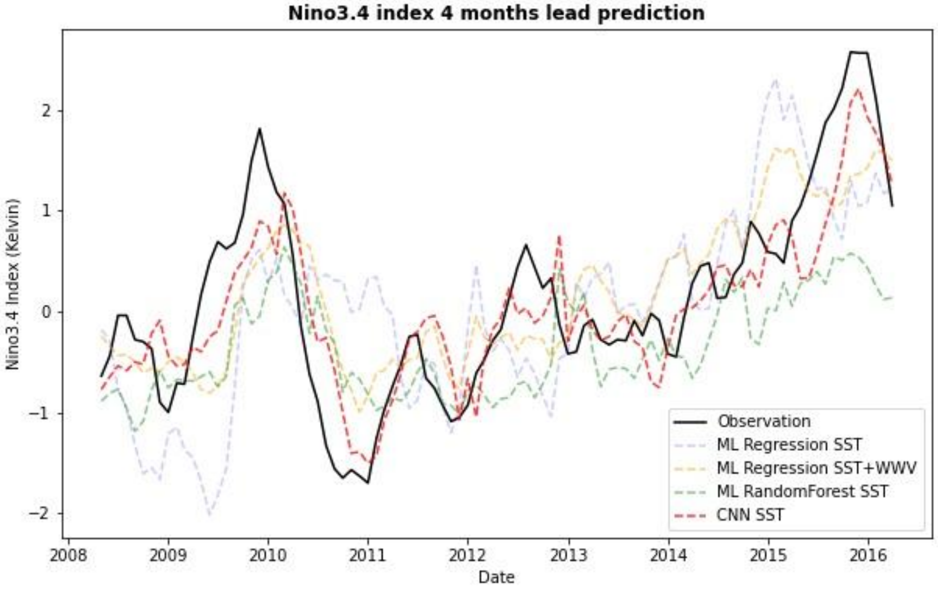
Random forest good performance but it takes long time to tune the parameter

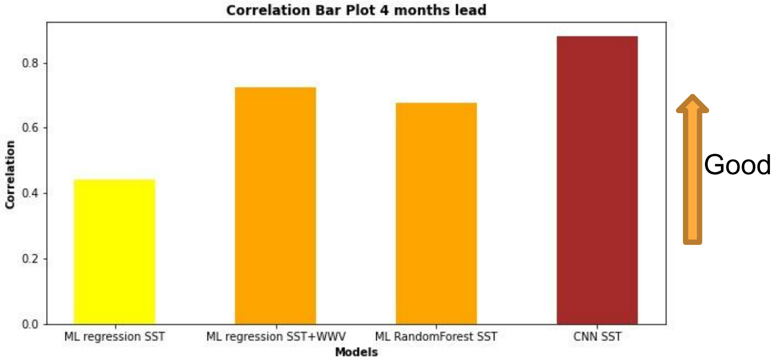


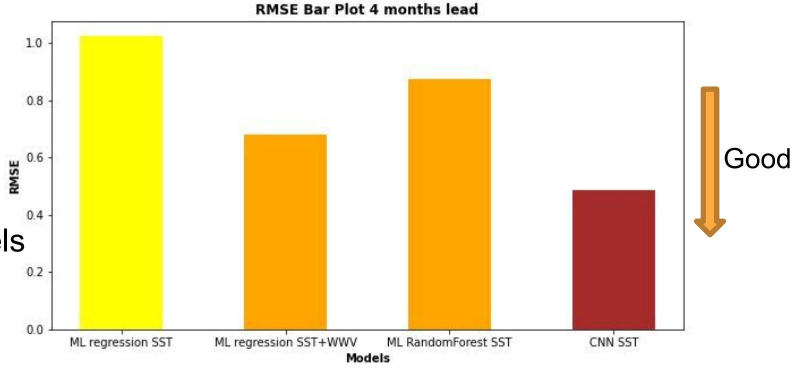


Team 1: Seasonal ENSO Forecasting (4 months lead)

Members: **Abdullah Al Fahad** (George Mason University), **Abisha Mary** (IIT Delhi), **Alka Singh** (NASA GSFC)







Nino3.4 index prediction >3 months lead is difficult. We explored ML/NN models with Sea Surface Temperature (SST) and **Warm Water Volume (WWV)** as predictor to forecast nino34 index 4 months ahead.

- WWV+SST increases the model predictability compared to just SST
- Convolutional Neural Network (CNN) worked best with high correlation (0.88) and least RMSE (0.48)

Team 34: El Niño

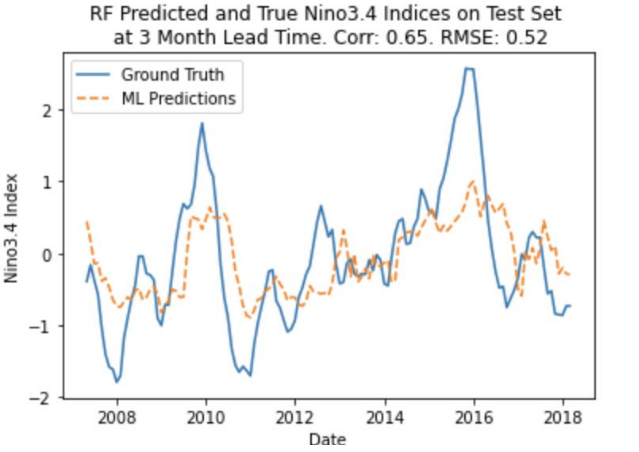
Team Members:

- Andrea Jenney
- Hannah Zanowski
- Chris Battisto
- Saicharan Vasala

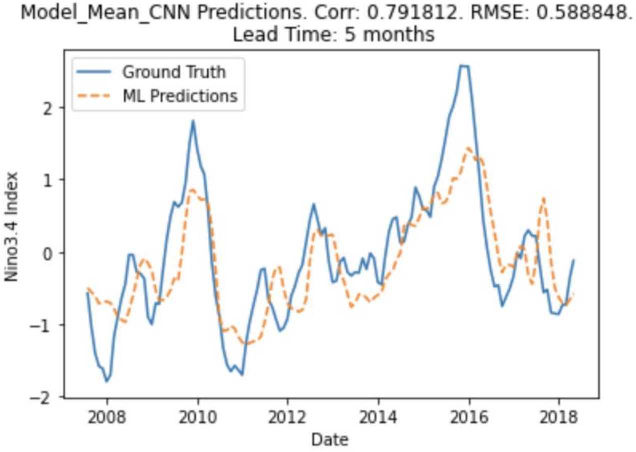
Methods:

- Linear regression and ridge regression, random forest, CNN, hyperparameter tuning/changes (dropout, batch sizes)

Some early attempts:



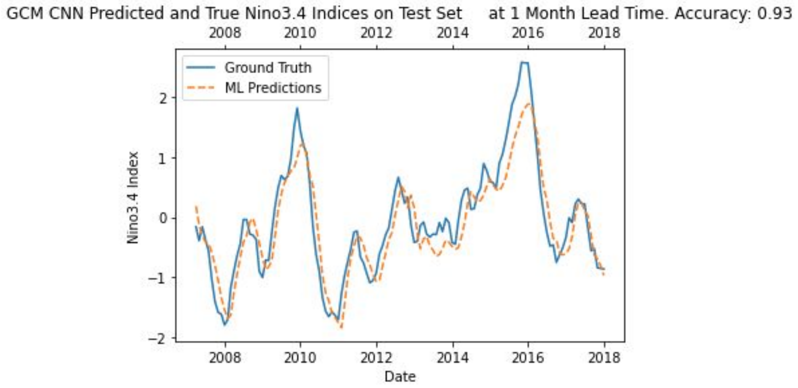
Hannah winged a Random Forest trained on 1990-2005 obs. It did not go well.

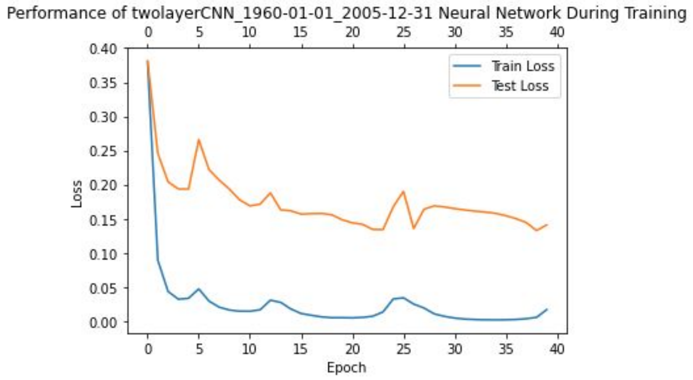


Two-Layer CNN trained on 200 yrs of ‘multi-model’ (CNRM+MPI) mean output

Team 34: El Niño

Our ‘final’ model: 1860-2015, lead_time=1, **added two dropout layers with p=0.5**





Dropout helped the model to train faster and slightly affect overfitting

Lessons learned/challenges:

- Loads of things impact model training such as data amount/quality, different lead times, hyperparameters...this was helpful to learn but hard to get a handle on what works and what doesn’t
- Beware of over/underfitting
- Gained some familiarity with new tools such as pytorch, etc
- ML is awesome! 5 days straight of zoom is not

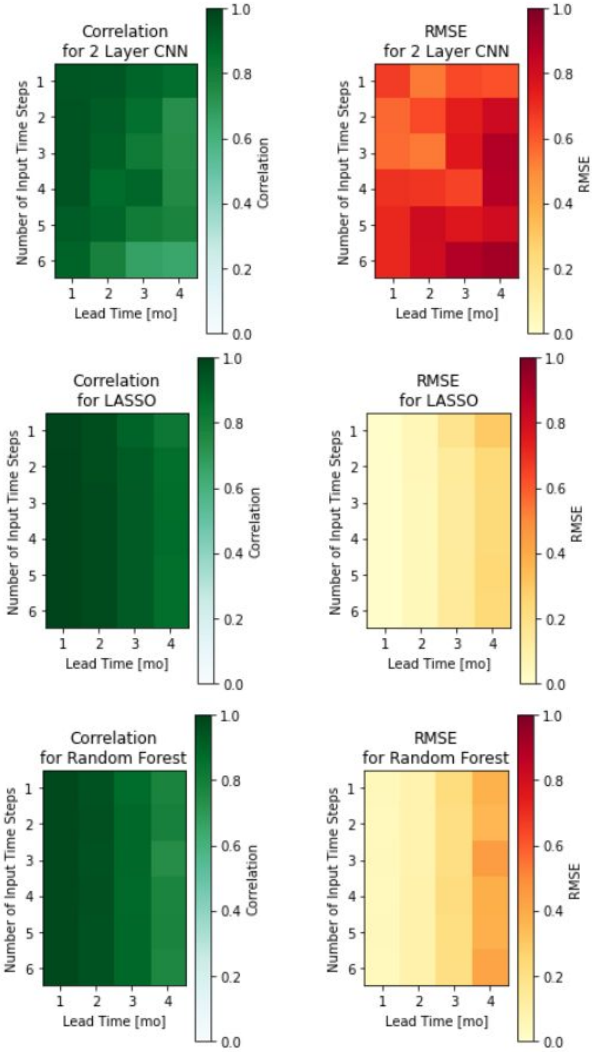
Team 13: El Niño

Team Members: Joyce Yang, Rahul Singh, Jun Zhang, Aaron Sweeney, Sopan K

Summary of methods tried: We ran *Linear Regression, LASSO, Random Forest, and CNN*. We learned that including training data prior to 1979 actually degraded the prediction, possibly due to increased observation uncertainty, decreased observation accuracy, various reconstruction technique for data making and/or greater sparseness of data (see figure on Slide 2 from NCEI).

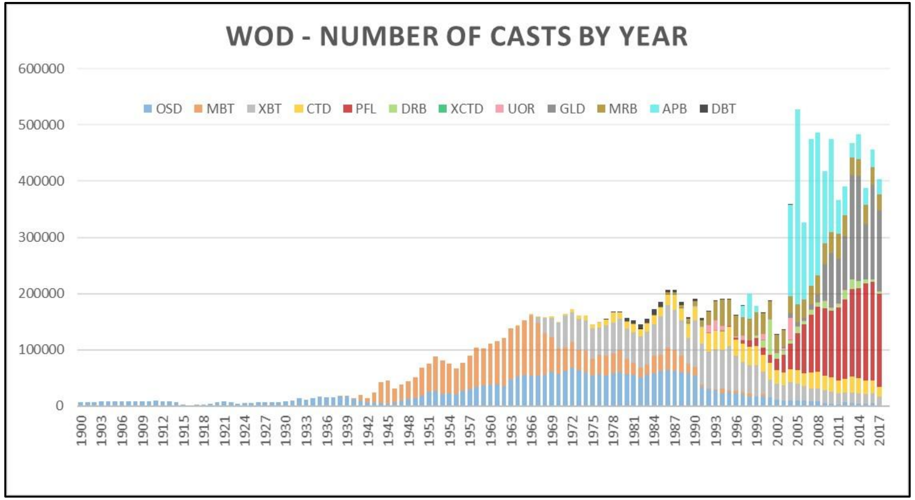
We verified that the *SST* outside Pacific region is not a good predictor for ML and CNN algorithms to predict Niño3.4 index with reasonable accuracy.

We found that number of epochs does not improve the prediction accuracy (Ex. lower the loss)



Team 13: El Niño

Prior to satellite observations of *SST* in the 1980's, data were limited to ship-based observations.

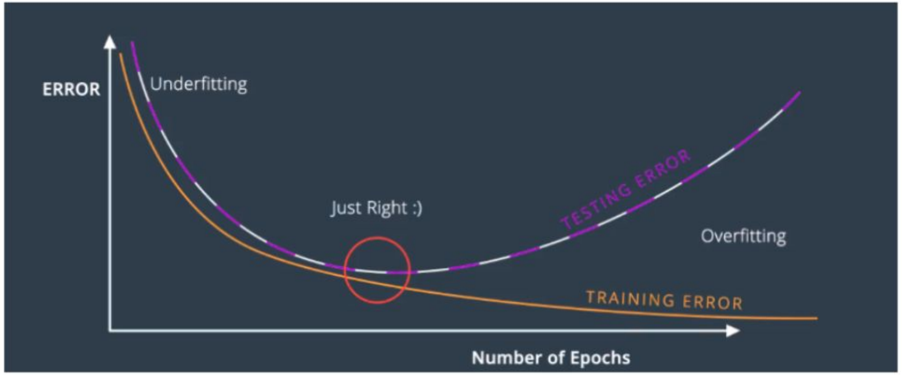


Team 13: El Niño

Lessons Learned:

1. Hyperparameters are as important as input data to train a ML model;
2. Necessary to do manual adjustments for overfitting/underfitting, such as to finetune the number of epochs, iterations and batch sizes. In future, could try to implement early stopping in the code;
3. More data is not necessarily better - tend to be overfitting.
4. Observed bias-variance tradeoff between training and testing.
5. It can be challenging to predict El Niño events with leading time longer than a year and half solely based on the *SST* datasets.

Challenge: not enough time/computing power in the future, would like to conduct a more comprehensive analysis of all the possible parameters to change (gap between training and testing data, GCM outputs vs. observational data, underfitting/overfitting, etc.)



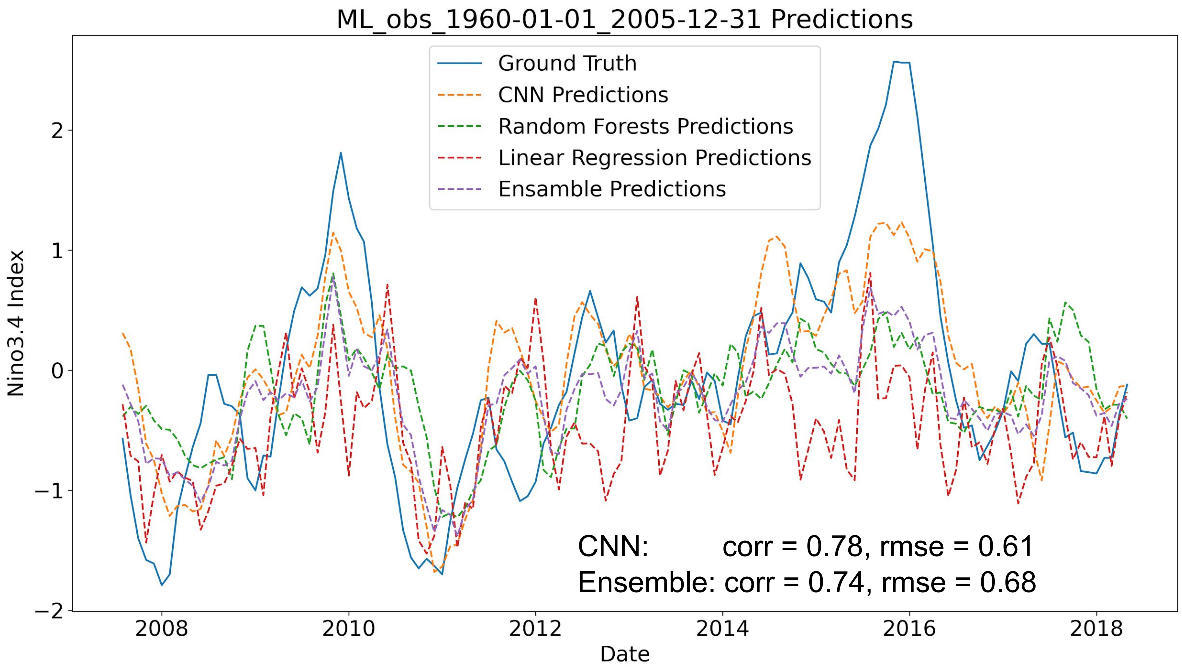
Team 20: <El Nino: Nino3.4 Index>

- Team Member Names:

  Qina Yan, Divyansh Chug*, Qinxue (Sharon) Gu, Varunesh Chandra, Abdellah Azdoud*

- Summary of methods tried
  - Two layer convolutional neural network
  - Random forests
  - Linear regression

- An visualization of the data
  - Examples of results: Lead time = 5, num_input_time_steps = 3



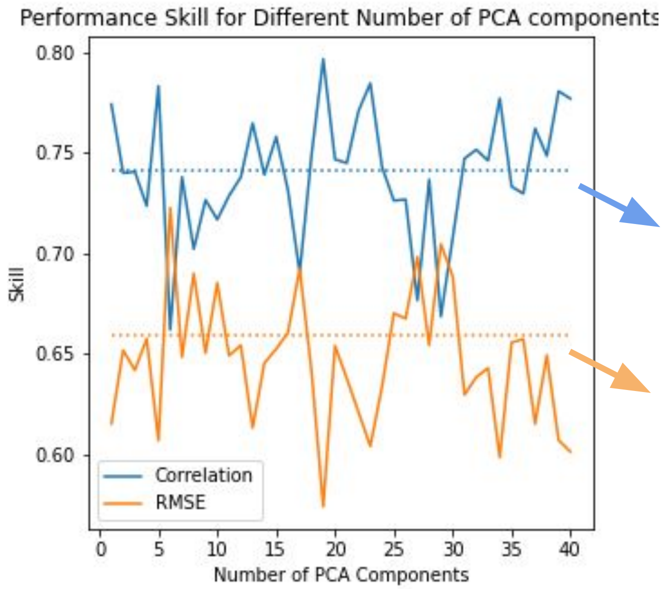
Team 20: <El Nino: Nino3.4 Index>

Qina Yan, Divyansh Chug*, Qinxue (Sharon) Gu, Varunesh Chandra, Abdellah Azdoud*

- Lessons learned/challenges
  - In general, if everything else is the same, the performances of different methods are rated:
    - For small lead time (i.e., 1): CNN ≅random forests > linear regression
    - For larger lead time (i.e., 5): CNN > random forests > linear regression
    - The training dataset with Pacific region provides better prediction than global dataset
  - The performance of CNN:
    - Overall: smaller learning rate provides better performance, but the size of convolution filters and the number of convolution filter do not show better performances than the default values
    - Larger size of the convolution filters provides better performance but numbers that are too large give worse performance, which is probably caused by overfitting
    - Higher number of convolution filters provides better performance but numbers that are too high give worse performance, which is probably caused by overfitting
    - Smaller number of weights in the fully connected layers would give better performance, and numbers that are too large gives worse performance, which is probably caused by overfitting

Team 20: <El Nino: Nino3.4 Index>

Impact of PCA



Solid Lines: With PCA

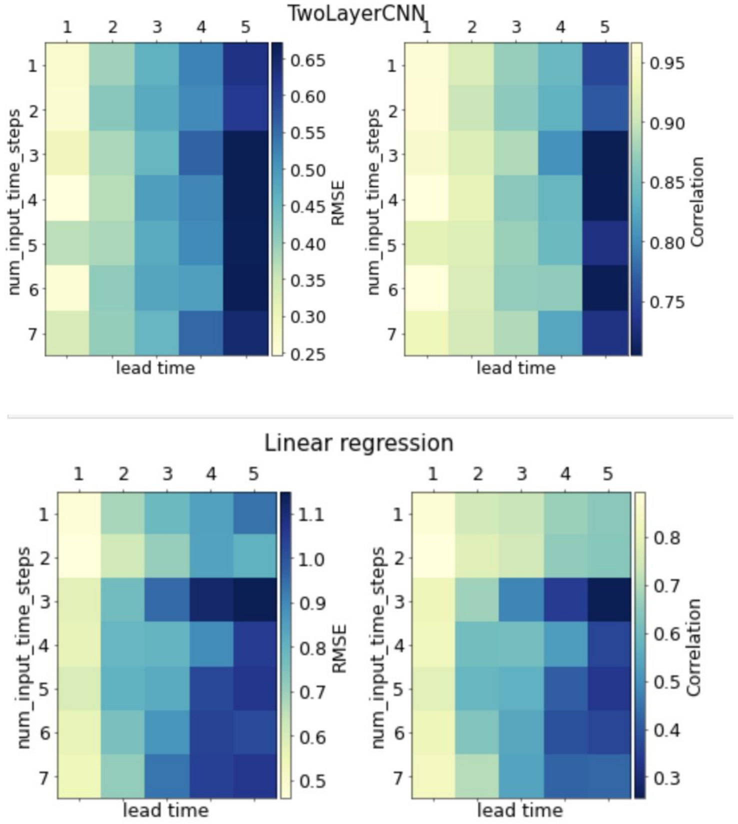
Dashed Line: Correlation without using PCA

Dashed Line: RMSE without using PCA

Experiments: Testing the performance of using different number of PCA components as well as not using PCA with lead_time=5, num_input_time_steps=3, predictor = global temperature, and default CNN parameters.

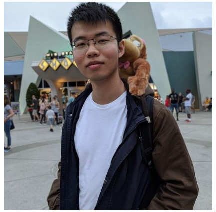
Results: The prediction is highly sensitive to the number of PCA components. 19 components are the best in this condition.

Impact of lead time and num_input_time_steps



Team 24: ENSO Forecasting

- Team members:
    - Da Fan
    - Hordur Helgason

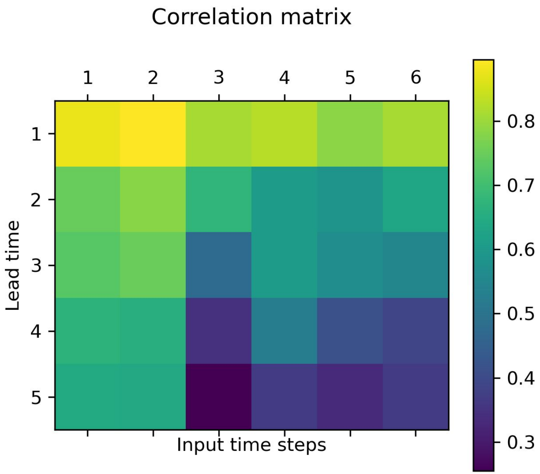



- Goal of the project: Use ML models to forecast the El Niño/ Southern Oscillation

- Training sets:
  - SST anomalies
  - GCM output
  - Temperature from ERA5

- ML models tested:
  - Linear regression
  - Random forest
  - Two layer CNN

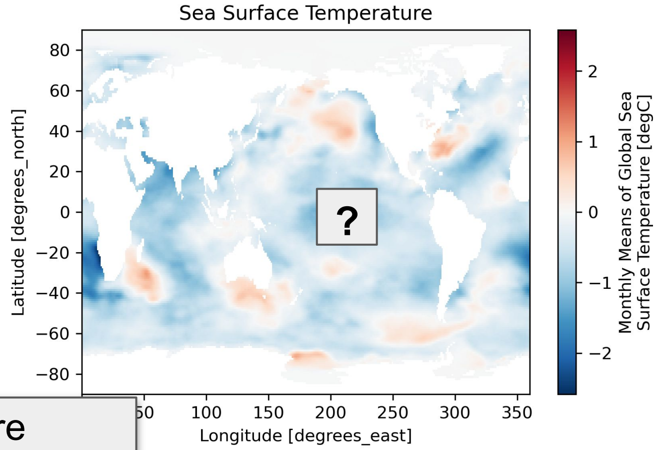
- Analysis example:
    - How does the model perform on different lead times?
    - How many months should we use as a predictor?



Da Fan and Hordur Helgason

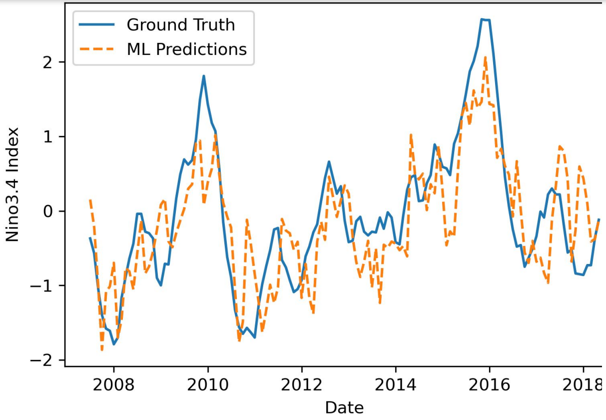
Team 24: ENSO Forecasting

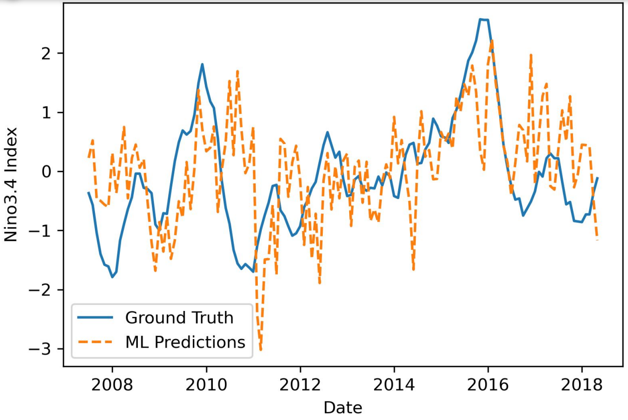
- The Nino3.4 index is a rolling 3-month average of equatorial Pacific temperatures.
  - Should the model predictors be the entire globe?
  - Or just a region in the equatorial Pacific?
- Data from the rest of the world does add value, but it also adds noise!



Global reanalysis temperature
Corr = 0.76, RMSE=0.39

Equatorial Pacific temperature
Corr = 0.34, RMSE=1.14





Our results:
Training ML models using temperature data from the entire globe yields better ENSO forecasts

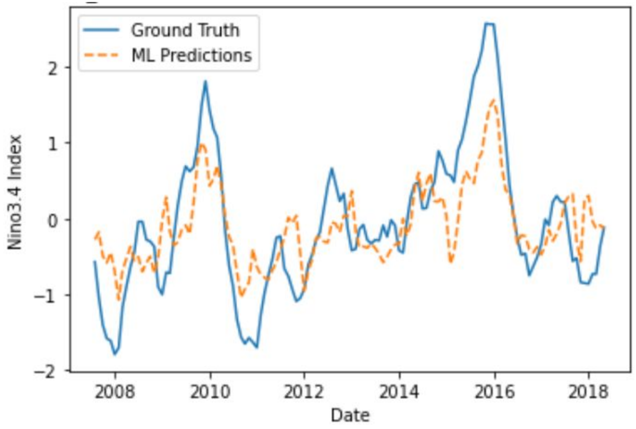
Da Fan and Hordur Helgason

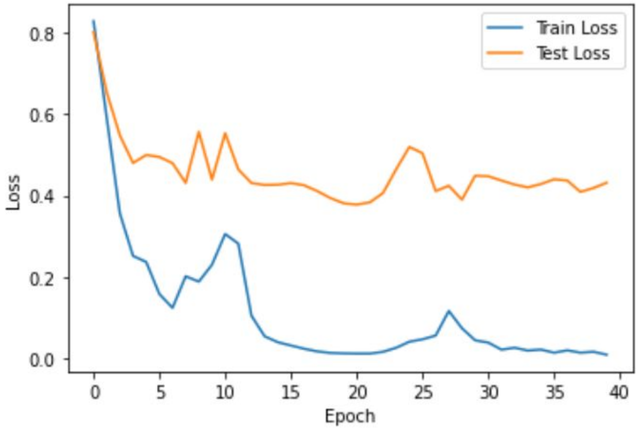
Team 24: ENSO Forecasting

CNN model setup:

- Training period: 1980-01-01 -- 2005-12-31
- SST data: Pacific region, global (Finally selected)
- Optimizer: SGD, RMSprop, Adam (Finally selected)
- Size of filter: (58464, 120), (120, 44), (44, 1)
- The size and depth of filter has a direct and complex impact on the speed of the error saturation
- Needs better calibration!

Correlation = 0.80



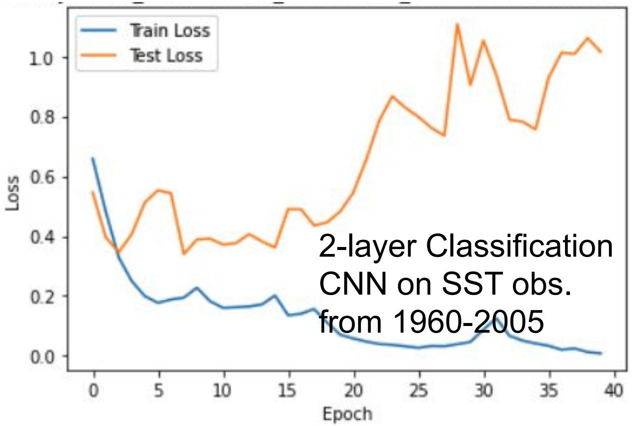


Team # 41: Seasonal Forecasting of ENSO

Team members: Hannah Horowitz, Margot Clyne, Nuo Chen, Sem Vijverberg*, Stephanie Knill*

- Summary of methods tried: linear regression; Random Forest; CNN; …...
- An visualization of the data
- Familiarize with Machine Learning debugging

Team # 41: Seasonal Forecasting



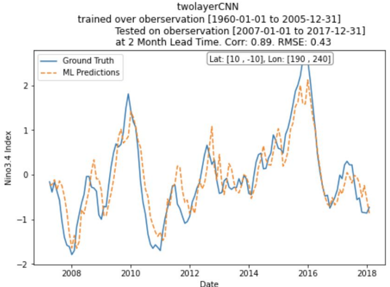
2-layer Classification CNN on SST obs. from 1960-2005

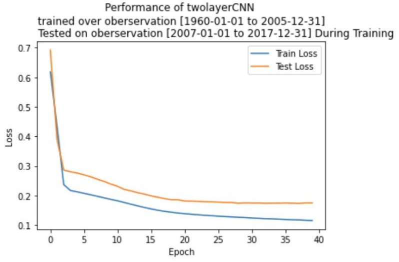
Lesson learned:

- Classification of Niño3.4 index as +/- performed worse than regression;
- ENSO predictability with linear regression performs bad at 3 months lead timeand beyond;
- A test size of 0.7 is a sweet spot
- Standardized training set doesn't perform better than unprocessed training set
- Convolutional Neural Network (CNN) usually performs better than linear regression
- Compare the performance of dataset over difference spatial region (e.g. tropics v.s. global) and time range (early 1900s v.s. late 1900s) help us confirm some fundamental knowledge.

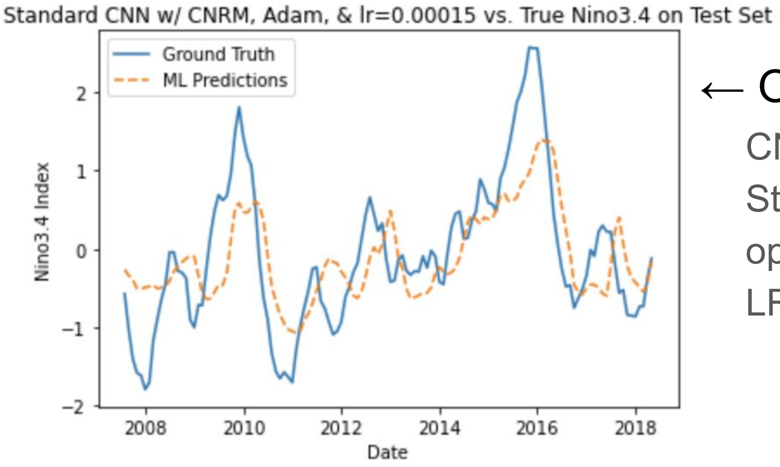
Challenges:

- Overfitting
- Closing the gap between train loss and test loss in CNN performance plot





Team 60: El Nino: Luis De la Fuente, Ian McGinnis, Brian Reed, Rochelle Worsnop



← Our best-performing model, RMSE ~0.63

CNRM data

Standard CNN network with 10 epochs

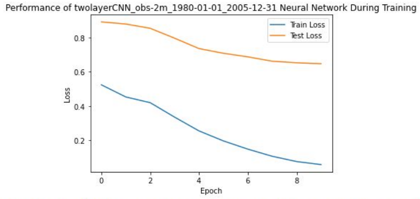
optimizer: SGD

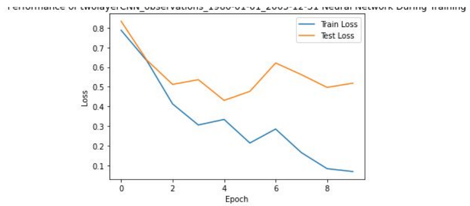
LR =1.5e-4

Additional findings →

t2m inputs (RMSE = 0.80)

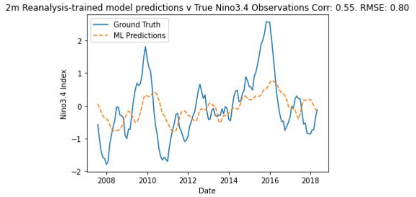
SST inputs (RMSE = 0.66)

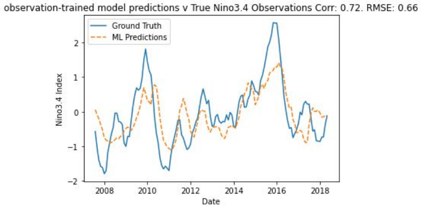




Average Training Losses: 0.6873032020604504; Average Testing Losses: 0.8733428

Average Training Losses: 0.6873032020604504; Average Testing Losses: 0.8733428807130881





Tested standard CNN network with inputs from observed SST & t2m reanalysis

SST inputs yielded better performance from the standard CNN network

Team 60: El Nino

Lessons Learned

The way we split the data for training, validation, and testing is critical especially for geosciences data where there may be memory of the system you’re trying to predict.

You need to be careful with the data you use (context matters) --- this is where expertise comes into play. A team of people with different expertise is helpful.

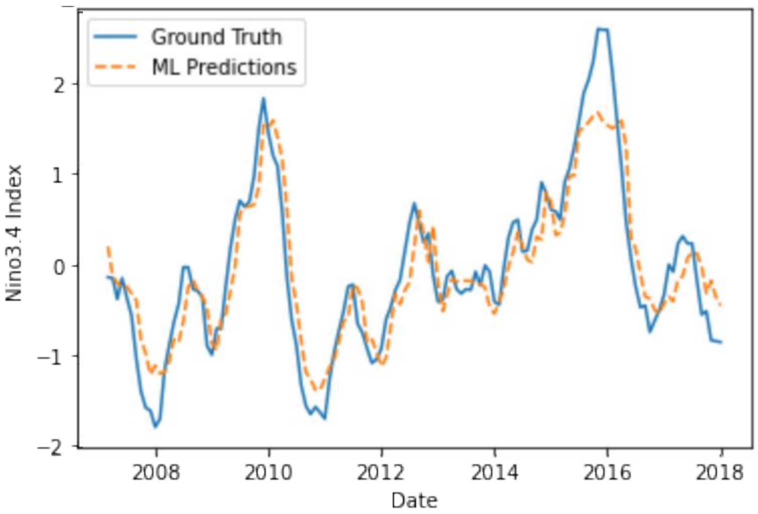
Bias in input can yield bias in output. Need to carefully consider the nature and biases of the data you are putting in. “Right model for the right reason”

Team 50: El Niño

Lead_time=1
num_input_time_steps=2
Methods are trained on observations.

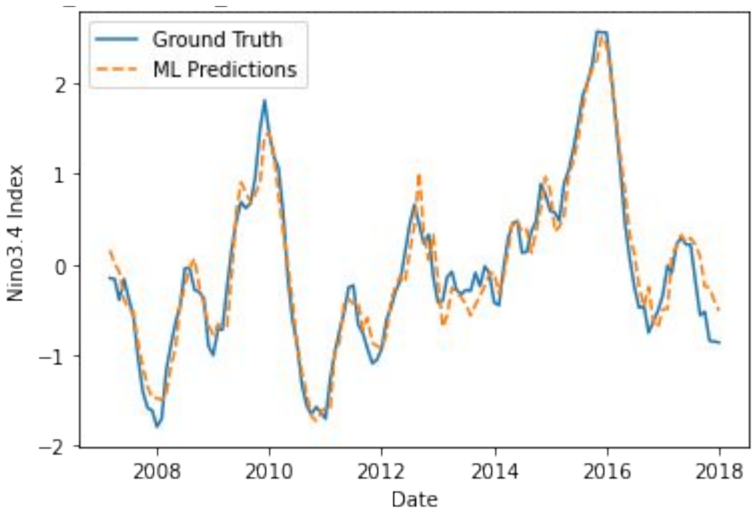
Random Forest with 20 branches

Corr: 93%, RMSE:0.32



Two-layer CNN

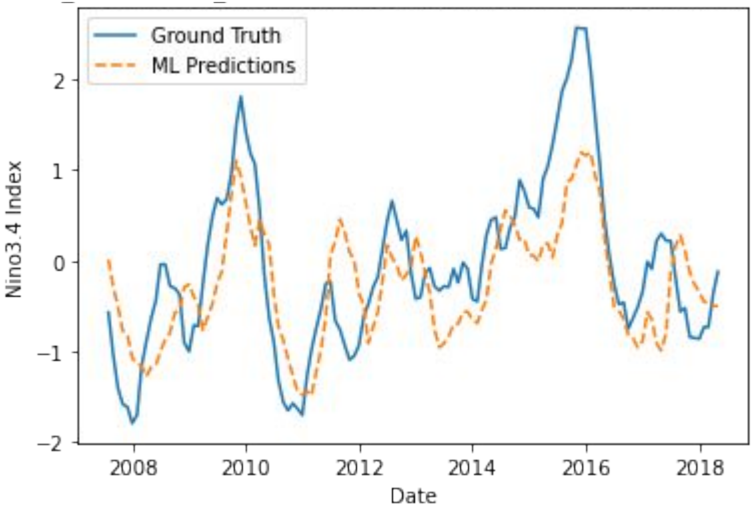
Corr: 97%, RMSE:0.24



Lead_time=5
num_input_time_steps=3

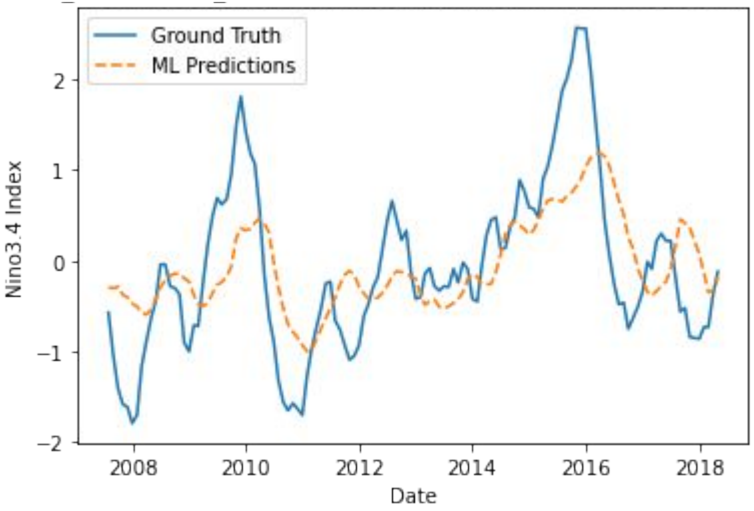
Two-layer CNN trained on Observation

Corr: 75%, RMSE:0.66



Two-layer CNN trained on CNRM

Corr: 68%, RMSE:0.71



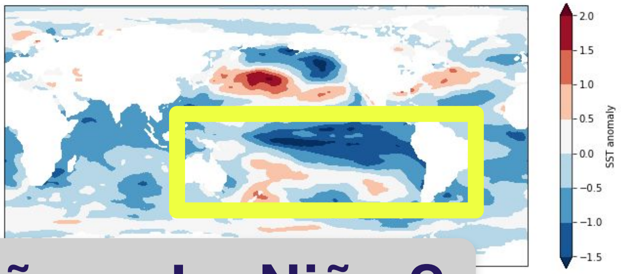
Team # 72: El Niño

Oscillation team

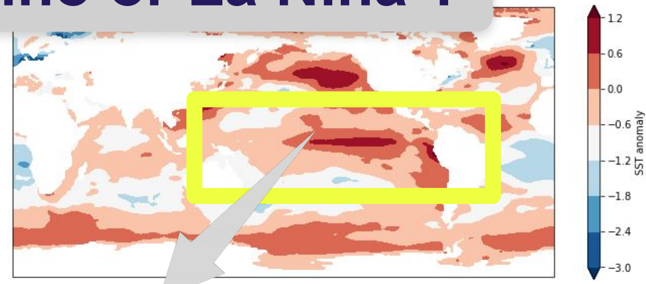
(since we switched projects and had only 2 days to work on this one)

Team Members:

- *Abram Farley*
- *Amanda Triplett*
- *Brayan Maurer Urbina Zenteno*
- *Gerardo André Rivera Tello*
- *Kasia Tokarska*
- *Xiaoning Wu*



El Niño or La Niña ?



Classification

CNN modified to binary classification

Can we predict it?

Neural Network (CNN)

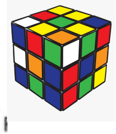
Linear Regression
(Regularized and not)

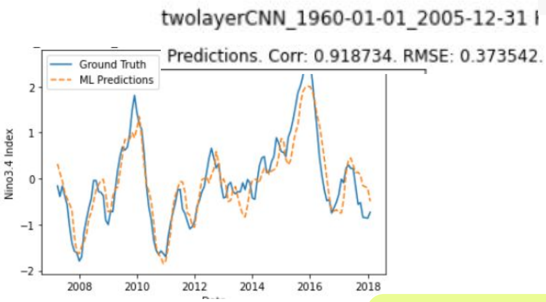
Random forest

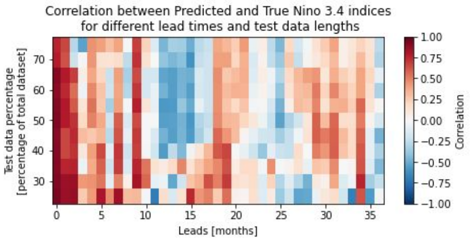
Team # 72: El Niño

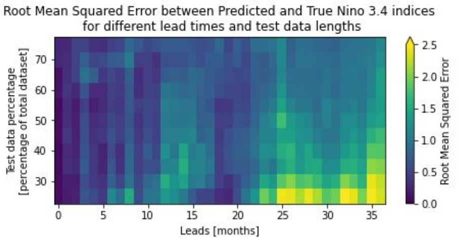
Exploring the Model ZOO

Convoluted Neural Network (CNN)

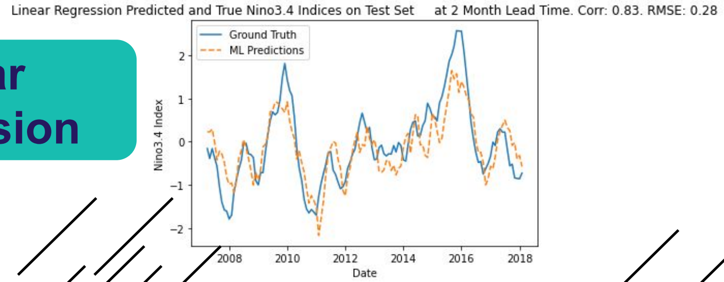




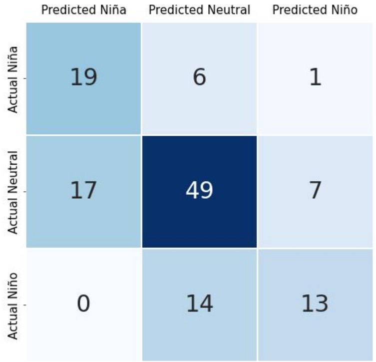


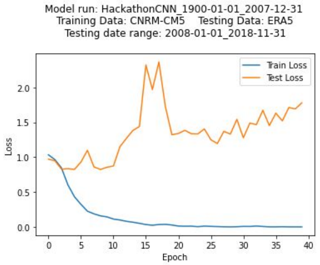


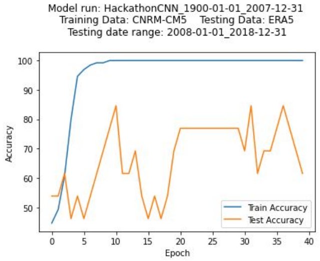
Linear Regression

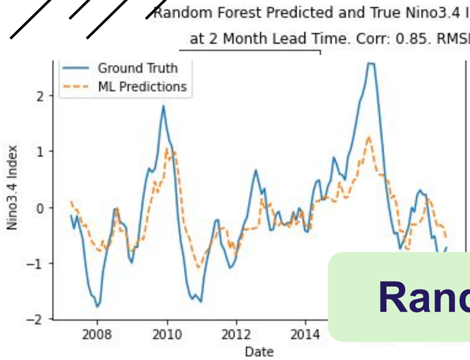


CNN modified to multiclass classification









Random forest

**Preliminary Results - subject to change*

** caveat: training periods may vary among the figures illustrating different methods, as it was a collective effort to assemble figures*

*** Images come from a google search - need proper attribution*

Team # 72: El Niño

Lessons learned and insights

- ❑ The "**ZOO**" of machine learning and deep learning models is **fascinating to explore**
- ❑ **Different evaluation** metrics need to be examined
- ❑ Sometimes **simpler linear methods** sometimes work surprisingly well…
- ❑ It is important to explore **hyperparameter space**

Importance of domain knowledge: examples closer to one's expertise are much easier to follow, interpret, and potentially gain new knowledge

Wishlist: visualization tools (e.g. "heatmap") **to help understanding the models** (e.g. why CNRM seems to do better than MPI)

- ❑ **Hackathon** was a great way to get started with machine and deep learning models
- ❑ **Slack team collaboration** was great despite different time zones
- ❑ **Thank you for this opportunity!**

Team 71: Seasonal Forecasting

Team Members:

- Zane Martin
- Alexandra Jahn
- Chris Wyburn-Powell
- Jamison Smith
- Alvaro Salazar*

Convolutional Neural Network (CNN) ENSO predictions

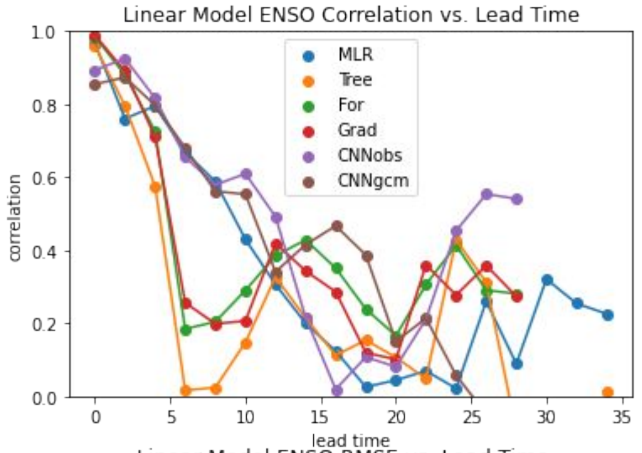
Trained several models to predict ENSO at various lead times.

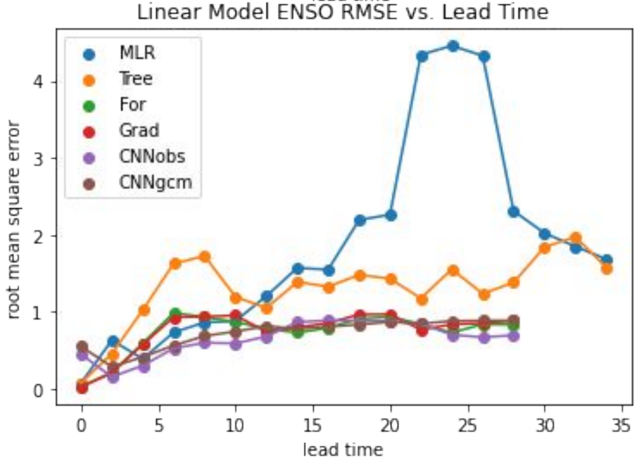
Training period: 1970-2000
Test period 2005-2015

Show correlation and RMSE as a function of lead time (months)

MLR - linear regression model
Tree - decision tree (max_depth=4)
For - random forest (30 trees,depth=6)
Grad - gradient boosted forest (30 iterations)
CNNobs - CNN trained on obs (15 epochs)
CNNgcm - CNN trained on CNRM gcm

We arrived at the parameters validating against 1930-1960, but didn't fully tune/explore the hyper-parameters; this is more a proof of concept!

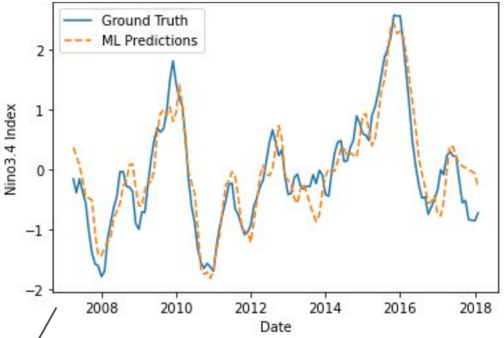




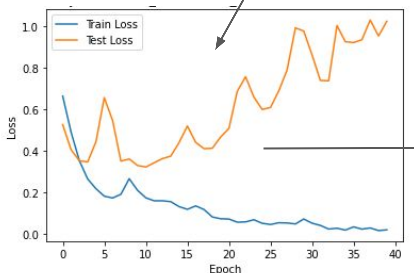
Team 71: CNN and classification ENSO predictions

1. Initial CNN prediction

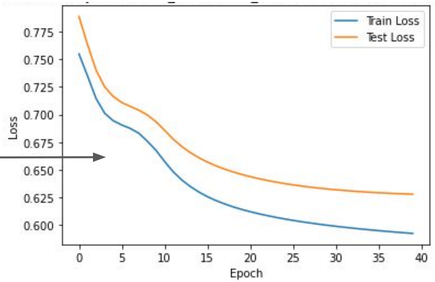
-Train: 1960-2005
-Lead time: 2 months
-r =0.92
-RMSE=0.38



Conclusion: CNN does better on regression ENSO prediction (r=0.92) than classification prediction (r=0.80)



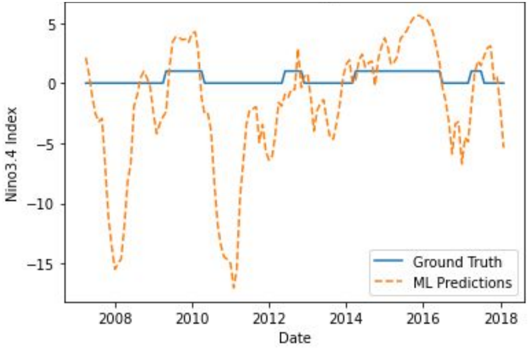
Loss functions initially showed overfitting



Improved, with longer training period (to 1930) and smaller learning rate

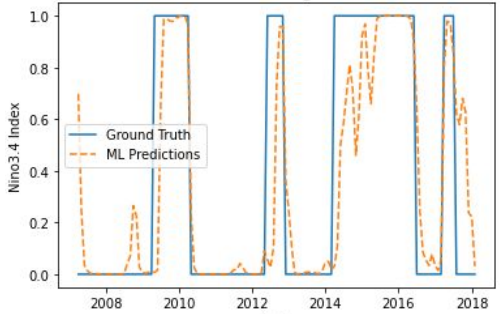
```
optimizer =
optim.Adam(net.parameters(), lr=1e-5)
```

2. Classification problem with CNN



-Train: 1960-2005-
-Lead time: 2 months
-r =0.62
-RMSE=5.5

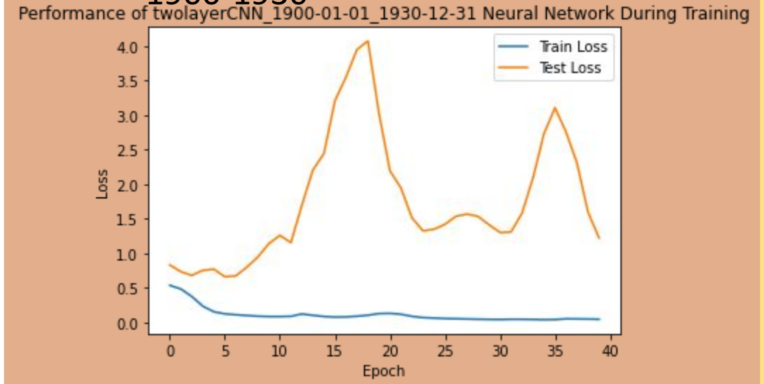
Longer training, smaller learning rate, and sigmoid in CNN: `x=(torch.sigmoid(x))`

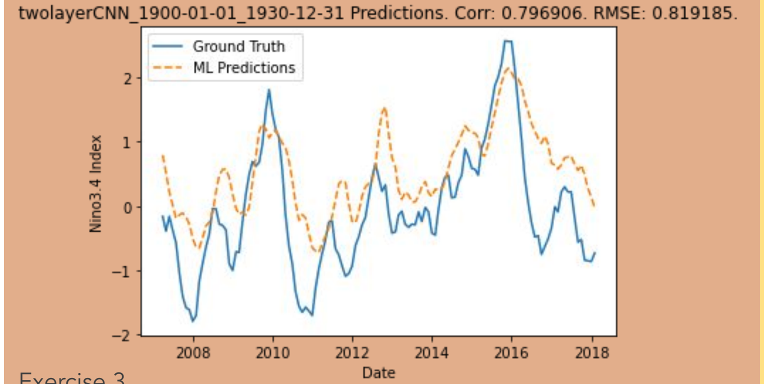


-Train: 1930-2005
-Lead time: 2 months
-r = 0.80
-RMSE=0.29

Team 20: How does time slicing (data quality) affect CNN performance?

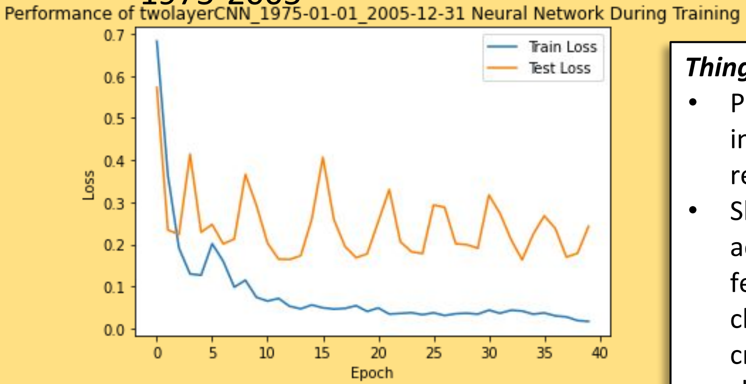
(A) Train on SST from 1900-1930

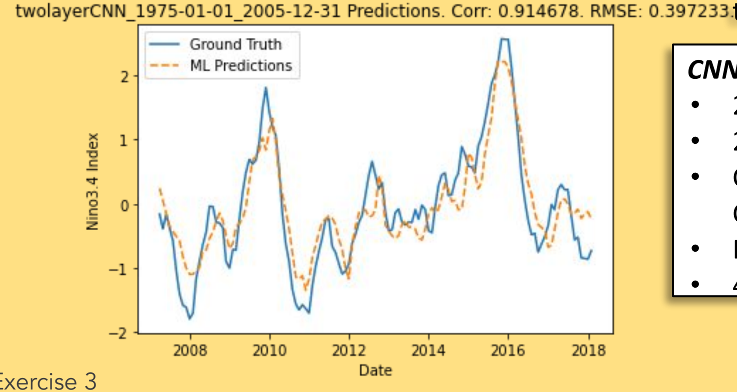




Exercise 3

(B) Train on SST from 1975-2005





Exercise 3

Things to note:

- Potential non-stationarity in SST and ENSO relationship with time.
- Should introduce additional climate features less sensitive to climate change and cross-validate while shuffling the data temporally.

CNN Architecture

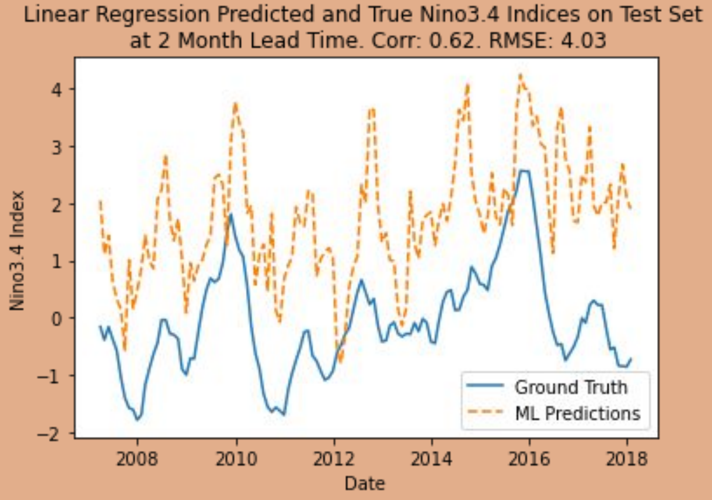
- 2-month input
- 2-month lead
- Conv2d | MaxPool2d | Conv2d | FC | FC | FC
- ReLU activation
- 40 epochs

Team 25

El Niño Hackathon
AI4ESS 2020

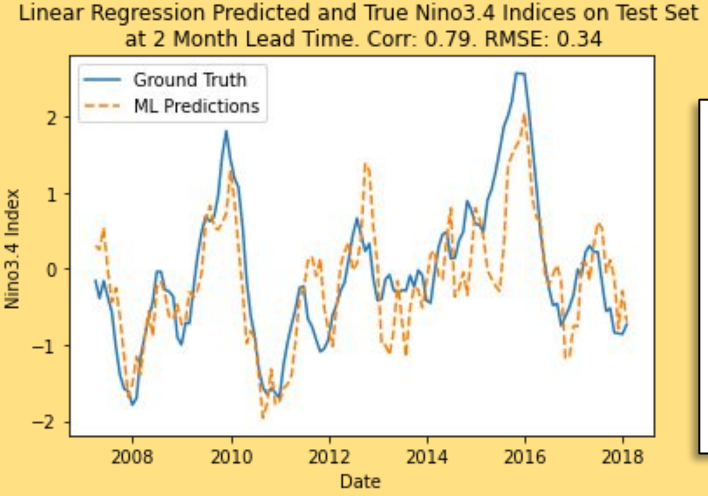
How does time slicing (data quality) affect Linear Regression performance?

(A) Train on SST from 1900-1930



Exercise 3

(B) Train on SST from 1975-2005



Exercise 3

Things to note:

- Training data from 1900-1930 performs poorly (R=0.62) compared to the more recent period (R=0.79).
- I wonder if scaling (A) training data would improve El Niño predictions.

Team 25

El Niño Hackathon

AI4ESS 2020