**Development of a General Climate Model Correction Algorithm using a Random Forest on a Reforecast-Reanalysis Comparison**

**Amsterdam University College**

**Bachelor Thesis**

**Tim F. Holthuijsen**

**May 2021**

**Supervisor: Bart Verheggen**

**Reader: Breandán Ó Nualláin**

**Word Count:**

**Appendix Word Count:**

**Abstract**

*Multiple adjustment approaches are examined, and their pros and cons are discussed. Eventually, an all-encompassing combination of these approaches is proposed as the optimal form of blackbox climate model adjustment.*

**Introduction**

Climate models try to understand, simulate and calculate the physics that cause our climate to behave the way it does. While we have a good idea of the underlying laws that hold true in physics, it is very hard to estimate how and how often these physics behave in nature, especially in a system as large and complex as earth’s climate. And although they are skillful, these models therefore make mistakes. Either through incorrect parameterizations of physical phenomena or insufficient knowledge of when they occur, the physical simulation of climate systems is not perfect. And that is why this model has an advantage; it does not use the laws of physics. Rather than trying to simulate everything that’s going on in earth’s climate, this model can make significantly better climate predictions by simply analyzing where contemporary climate models are wrong.

While it is common to use machine learning in climate model tuning (sources), it is predominantly used for optimizing the functionality of specific parameterizations (sources).

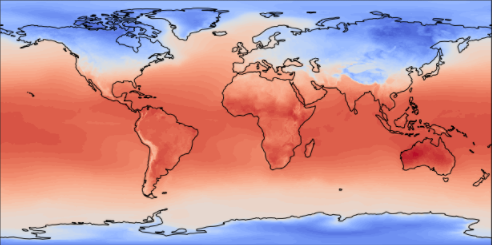
Contrast: Generally climate model adjustment is based on one specific model and one or a couple of specific physical climate variables.

Dueben & Bauer (2018) argue that with contemporary advancements in Machine Learning (ML) and Neural Networks (NN), it may be possible to develop algorithmic models without any understanding of climate physics that could make better climate predictions than a climate model could. This paper goes one step further, by using modern ML techniques to adjust climate model predictions and improve their accuracy, rather than having the ML model make the predictions from scratch. This solution offers the best of both worlds, since it uses a climate model that understands physical processes as a baseline, and then attempts to improve this baseline using the computational power and efficiency of ML techniques.

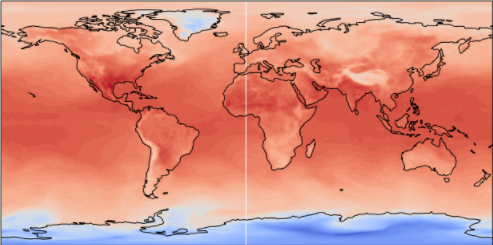
**Methodology**

*Data Processing*

In order to start the data analysis, all data first needs to be converted into the same format so that it can be compared. In order to get all data in the same shape, we start of by converting all of the climate data we have obtained into the same filetype, for which we choose the Network Common Data Form (NetCDF). This filetype is a convenient solution for geographic climate analysis variating over time, and both the ERA5 Reanalysis (Copernicus Climate Institute, 2019) and the GEFS reforecast (Earth System Research Laboratory, 2019) can be converted to this filetype without causing any data loss. Then, to display the climate data that has just been created, the NetCDF files are loaded into python and plotted on a global map with continental outlines, using the World Geodetic System 1984 (WGS84) coordinate system.

*Fig 1: Initial Reanalysis Data*

*Fig 2: Initial Reforecast data*



As becomes apparent from the images shown above, the Reanalysis and Reforecast data initially differ quite a bit. While we are in fact searching for discrepancies between these two datasets, this initial difference cannot be attributed to errors in the physical climate model. Rather, the current difference in these two datasets comes from the fact that both of them are build up in a different way, using different time units, span from and until different moments in time, and use a different resolution. In order to make the datasets comparable, all of this needs to be converted to exactly the same format. For the spatial resolution, this is relatively straightforward: since the resolution of the reanalysis is exactly 4 times higher than that of the reforecast, it is sufficient to simply reduce the reanalysis resolution to the exact extent of the reforecast dataset, which is a longitude of 360 degrees by a latitude of 180 degrees.

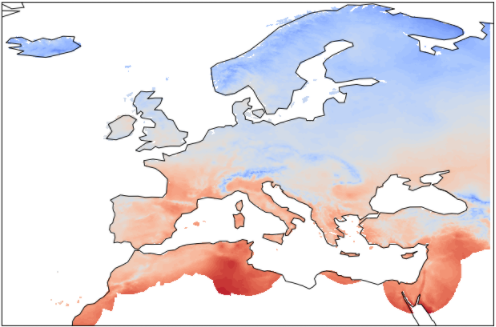
While getting the spatial resolution to align is rather simple, getting the time units of these different datasets to match up will prove to be more complicated. The first problem that needs to be solved is the fact that both datasets use a different unit to represent time. Specifically, there is exactly a 100-year difference between the way in which both datasets represent time. Keeping this difference in mind, we calculate that that the reforecast dataset starts on the first of June 1985 and ends on the 16th of September 2020, while the Reanalysis goes from the first of January 1980 until the first of August 2020. The different start dates can also be recognized in the fact that the initial reanalysis data (figure 1) seems a lot colder (blue) than the initial reforecast data (figure 2): the reanalysis starts in cold January, whereas the reforecast begins in warmer June. To solve this problem of mismatching timeframes, we clip the extent of both datasets to span from the first of June 1985, until the first of August 2020. We lose a little bit of data in this way, but it ensures that both datasets are comparable over the entire extent of the analysis. An additional problem is that the reforecast contains climate data for every day of the month, whereas the reanalysis only contains data for the first day of very month. The reforecast data is reduced to match the timesteps of the reanalysis, meaning the eventual data for our analysis will span across every first day of each month between June 1985 and August 2020.

By reducing the temporal and spatial resolutions of the datasets, a lot of data is lost. However, since the initial climate data was so abundant, this should not be a problem. In fact, the data we will be using consists of 180 x 360 cells per map, across a total of 422 moments in time, resulting in more than 27 million datapoints we can use for our analysis. This plethora of datapoints will prove to be more than enough to train and validate any possible machine learning model. The data is now in the perfect shape to be compared, spanning exactly the same dimensions, and still holds a vast amount of information.

*Data Validation*

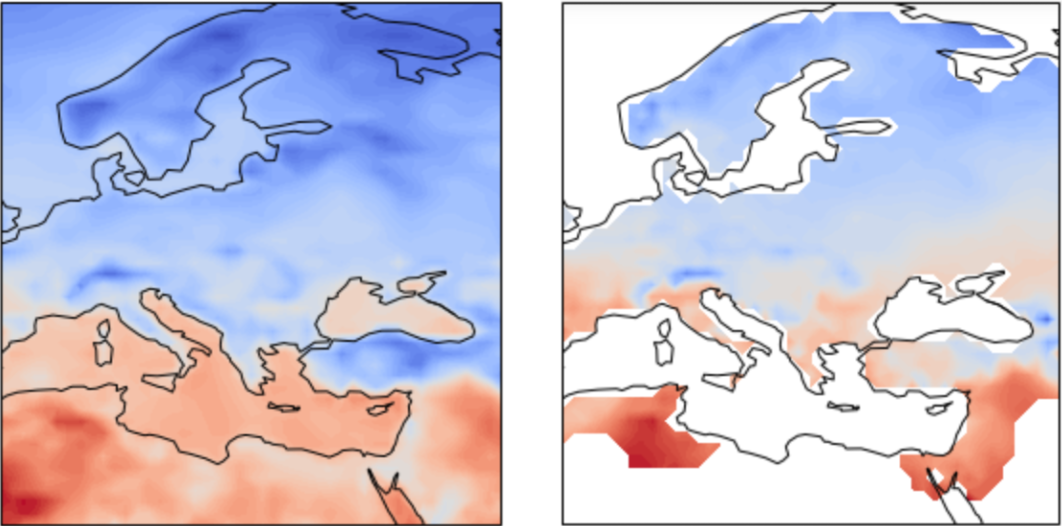
Before beginning to tune the reforecast model output to match the reanalysis more closely, it is important to ascertain that the reanalysis data is actually more accurate than the GEFS climate model outputs. It is known that reanalysis datasets always still contain some bias (source), so it is crucial to confirm that the reanalysis dataset is in fact closer to the real-world climate variables before using it as a test set. In order to validate the accuracy and reliability of both datasets, we use a dataset containing station measurements directly sourced from the European National Meteorological and Hydrological Services (NMHSs): the E-OBS meteorological data for Europe derived from in-situ observations dataset (Copernicus Climate Change Service, 2021). The temperature data in this dataset is derived directly from in-situ climate observations made by NMHSs stations and can therefore be seen as the closest we can possibly get to the real climate variables (source). We use this to test the accuracy of both the reanalysis and reforecast datasets against this ground truth baseline.

*Fig 3: In-situ validation set*



As becomes apparent from the image shown above, the validation set does not exactly match the extent of the other datasets, as the observations on which the dataset is based only span across Europe. Therefore, we clip the extent of the reforecast and reanalysis dataset to match this extent above Europe. Additionally, the 25.933 different measurement moments in the validation set are reduced to exactly the 422 moments we have chosen for our analysis. After matching the resolution and cropping the data a bit more, the climate model output and climate observations are shown side-by-side in figure 4. The validation set still contains some white spaces without observational data, but these regions will be ignored in the error estimation.

*Fig 4: Cropped Reforecast (left) and* *in-situ data (right)*

**

While the blue-red distribution across the two maps in figure 4 seems relatively similar, there are definitely some discrepancies between the predicted and the measured temperatures. In order to quantify these discrepancies, and all discrepancies from now onwards, we will use the Root Mean Square Error (RMSE) accuracy measurement. The RMSE gives a number as error score, representing the difference between two climate maps. If we have a set of N model-predicted climate variables Ŷ, and a set of N accurate climate variables Y, the the RMSE formula is defined as:

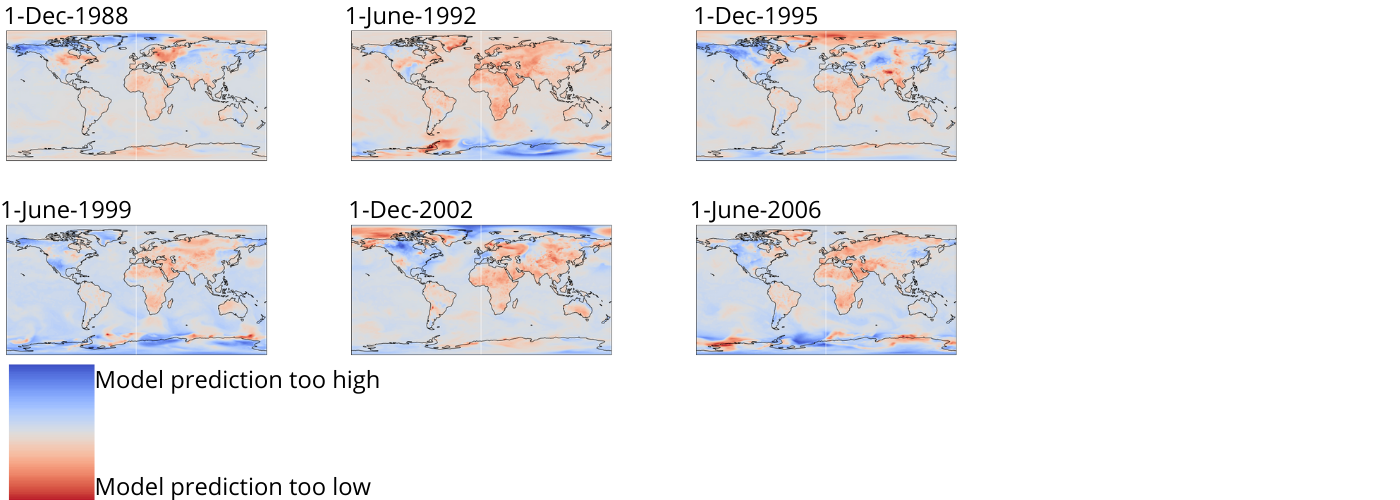
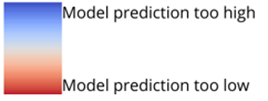
*Formula 1: RMSE*

For the inaccuracy of the reforecasted model output tested on the in-situ observations set, we calculate the RMSE score to be 3.22, indicating that there is a significant discrepancy between the climate model and the climate observations. When testing the Reanalysis’ accuracy on the in-situ set, its RMSE evaluates to 0.38: showing that the reanalysis set is substantially more accurate than the reforecast dataset, confirming the hypothesis that this reanalysis data is a relatively accurate and authoritative dataset suitable for climate model tuning. While it does not perform perfectly on the validation data and has an error score of 0.38, this is still an immense improvement over the reforecast’s error score of 3.22. This indicates that if we can tune our climate model to predict the reanalysis more accurately, it will be a more realistic estimation of the real-world climate as well.

To define a baseline RMSE which our models need to score below in order to be an accuracy improvement, we test how accurate the reforecast data is in predicting the reanalysis. Comparing the difference between all reanalysis maps and all reforecast maps, yields that the reforecast has an RMSE score of 2.90 on the reanalysis. This is an interesting result, as it is lower than the 3.22 RMSE which the reforecast attained for predicting the observational data, showing that the reforecast is in fact better at predicting the reanalysis data than it is at predicting the even more accurate observational climate data. This implies that some of the bias which is present in the reanalysis is also present in the reforecast, indicating that some of the same inaccurate parameterizations may have caused a similar bias in both datasets (Reanalysis bias source). If it has not been done before (reanalysis research), a good application of the final model developed in this paper may be to tune the reanalysis dataset to match the observational data more closely. This may help remove the RMSE of 0.38 still present in the reanalysis, reducing reanalysis bias. Furthermore, investigating how the model adjusts the reanalysis may give some clues as to which parameterizations this inherent bias originates from, allowing for these parameters to be tuned and the bias to be reduced or removed. Worthy of further investigation would also be whether this reanalysis bias reduction would also work in reducing the RMSE of the reforecast dataset. Noteworthy is that the reforecast’s accuracy difference between the two different test sets (reanalysis and observational) of 0.32 (3.22 – 2.90) is almost equal to the reanalysis’ inaccuracy of 0.38. This begs the question that if a method is developed for reducing this slight bias in the reanalysis, whether this method would work on reducing the reforecast bias by the same amount as well. This is a prime example of a potential application of the final product of this paper and may provide insights into both the origin of inherent model bias, as well as methods of reducing it and how the eventual climate correction algorithm can aid with this process.

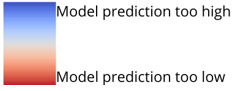
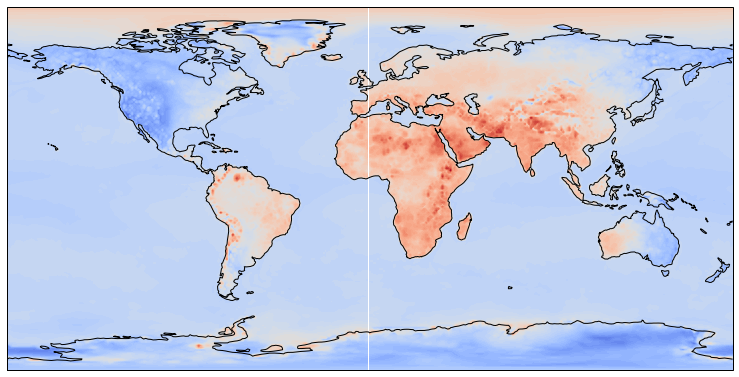
Regardless of the reanalysis’ slight inaccuracy, it is still deemed to be a valid test set, since it is vastly more accurate than the reforecasted climate data. Therefore, we will start quantifying the reforecasted climate model inaccuracy by subtracting the reforecast temperature maps from the reanalysis temperature maps, yielding global maps of climate model inaccuracy. From now on, model error will be defined as the reanalysis temperature minus the reforecast temperature in a given location. This means that a high model error value indicates that the prediction was too low and the model prediction should have been higher, and a negative error value indicates that the prediction should have been lower. This model error is then calculated globally for every moment in time across all 422 timeslots, creating 422 inaccuracy maps, which all show individual patterns of inaccuracy. The inaccuracy patterns found in these maps are all interesting, but also differ a lot across different moments in time. To give an indication of what these different patterns look like, a sample of six inaccuracy maps equally distributed across the chosen timeframe is displayed below.

*Fig 5: Climate inaccuracy across the years: a 6-map sample*

While some common patterns can be recognized in these maps, it is difficult to estimate the prevalence of a specific pattern across the years by simply looking at the different maps. In order to quantify the climate inaccuracy across the years, we use all the climate data from the 422 different maps and average their error. This yields one, all-encompassing climate inaccuracy map, which shows the average inaccuracy on each place by the GEFS, from 1985 until 2020.

*Fig 6: Average global inaccuracy*

******

Since this map averages all inaccuracies across the years, it is a good representation of the most prevalent patterns of inaccuracy, which become rather apparent by looking at this map. Specifically, model inaccuracy seems to manifest itself differently between different continents but appears to remain relatively consistent within these continents. Additionally, all water regions on this map have approximately the same blue colour, indicating that water is consistently predicted too high by the same amount. The South Pole also has its own blue colour which seems rather consistent with Greenland and the Russian Far East, implying that maybe land use types such as ice-covered regions and water may affect climate model inaccuracy in a rather consistent way. An alternative explanation for this pattern could be that warmer regions are generally predicted too cold, whereas colder regions are generally predicted too warm. This hypothesis seems to agree rather well with the data shown in figure 6, but different land use types could explain this phenomenon just as well. Which of these variables is more important for predicting model inaccuracy will be explored further in the results section of this paper. Figure 6 shows a large number of interesting patterns which could point to different origins of inaccuracy and warrants a large amount of further research to completely understand. For the sake of this analysis however, we are mostly interested in how we can use these patterns to create more accurate climate model predictions. A first use of this average model error set may be to simply subtract the average error from all temperature predictions in their respective location. If the model on average predicts a certain place two degrees too high, it may improve the accuracy if we simply subtract 2 degrees from all predictions in that specific location. Before we start making any modification to the data however, we need to split the climate data into a train and a test set, in order to prevent polluting the training data with the test data.

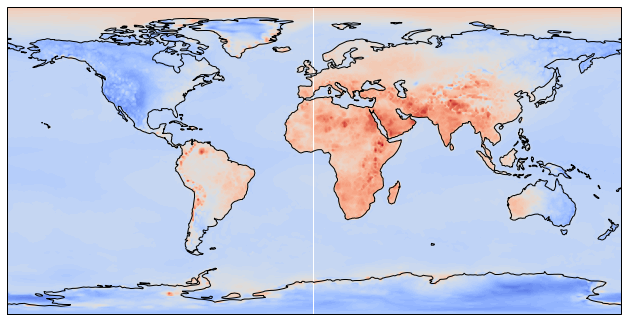
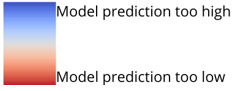
*Creating a training and test set*

When training our model, it is important that the data on which the model is trained and the data on which it is tested are kept separate. This prevents overfitting the model on the training data and creates a more realistic model accuracy assessment since the final version of the model is tested on a test dataset it has never seen before. Research has been conducted towards the ideal Train/Test ratio, (Pawluszek-Filipiak & Borkowski, 2020; Rácz et al., 2021) but has proven to be very dependent on the type of data used (Pawluszek-Filipiak & Borkowski, 2020). Regardless, Train/Test ratios of either 70%/30% or 80%/20% have proven to perform very well, specifically across large datasets (Rácz et al., 2021) like the one that is currently being used. For our Train/Test split, we choose to allocate 75% of the data to the training set, and the other 25% to the test set used for validating the model after training it. Thus, we shuffle all of the data, and randomly separate the dataset into a test and a training set.

*Average Error Subtraction*

As shown in figure 6, a map of average global model error gives a rather straightforward and understandable indication of average inaccuracy patterns. A logical conclusion to be made from these patterns of model error is that if the average model error was to be subtracted from all model temperature predictions, the average error might be reduced. In order to test this assumption, another average model error map is created (figure 9), using only the training set as its sample. This average error in the training set is then subtracted from the test set, and the accuracy improvement of this method is determined.

*Fig 9: Average global inaccuracy in the training set. Note: This is almost identical to figure 6*

**

The initial model RMSE was 2.90. After subtracting the average model error from all datapoints in their respective locations, the RMSE becomes 2.51. This is already a significant improvement, especially given the fact that this error subtraction is quite a simple solution, without the application of any sophisticated statistical techniques. For the application of such statistical or ML models, it may prove beneficial to first add more data which our models can use to understand and predict the complex patterns of inaccuracy.

*Adding more training data*

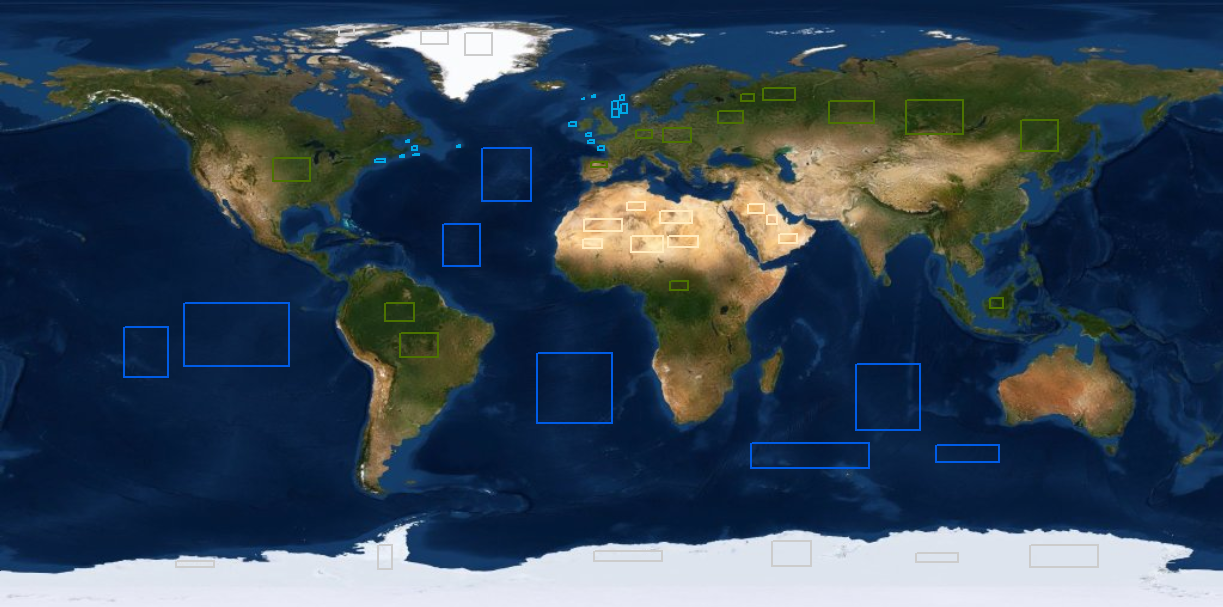
The data we currently have in the dataset still looks relatively meagre for making the more complex inaccuracy predictions that are needed to reach an even lower RMSE. Currently, one datapoint consists of its longitude, latitude, and time values, the reforecasted (model) temperature, the reanalysis (accurate) temperature, and the difference between these two temperatures (the error we are trying to predict). While this already offers a large amount of information about climate model inaccuracy, any machine learning model we train is likely to perform better if we give it more variables to train on. One condition for any data we add to the training and test sets is that it needs to be consistently available throughout time, since we want our error correction model to be applicable for future climate model predictions as well. Therefore, we cannot include variables such as measured humidity or solar radiation, since this data is not available for future model predictions. Something we can add however, is other climate variables forecasted by the GEFS. The Global Ensemble Forecast System attempts to simulate and quantify all of the physical processes occurring at any moment in the climate (source), and these other climate variables may prove helpful in our prediction of temperature inaccuracy. As such, we add the GEFS reforecasted Precipitation and Cloud Cover variables to the training and test sets and convert them to align with the spatial and temporal resolutions as previously described.

*Land use classification*

A less straightforward addition to our dataset relates back to the idea that land use may have an influence in determining temperature inaccuracy, as explained in the previous section. In order to test and quantify this hypothesis, it would be ideal to add a global land use type dataset into the analysis. However, land use data on a global scale in the required format is rather hard to come by. To solve this issue, we develop our own custom-made land use dataset on a global scale.

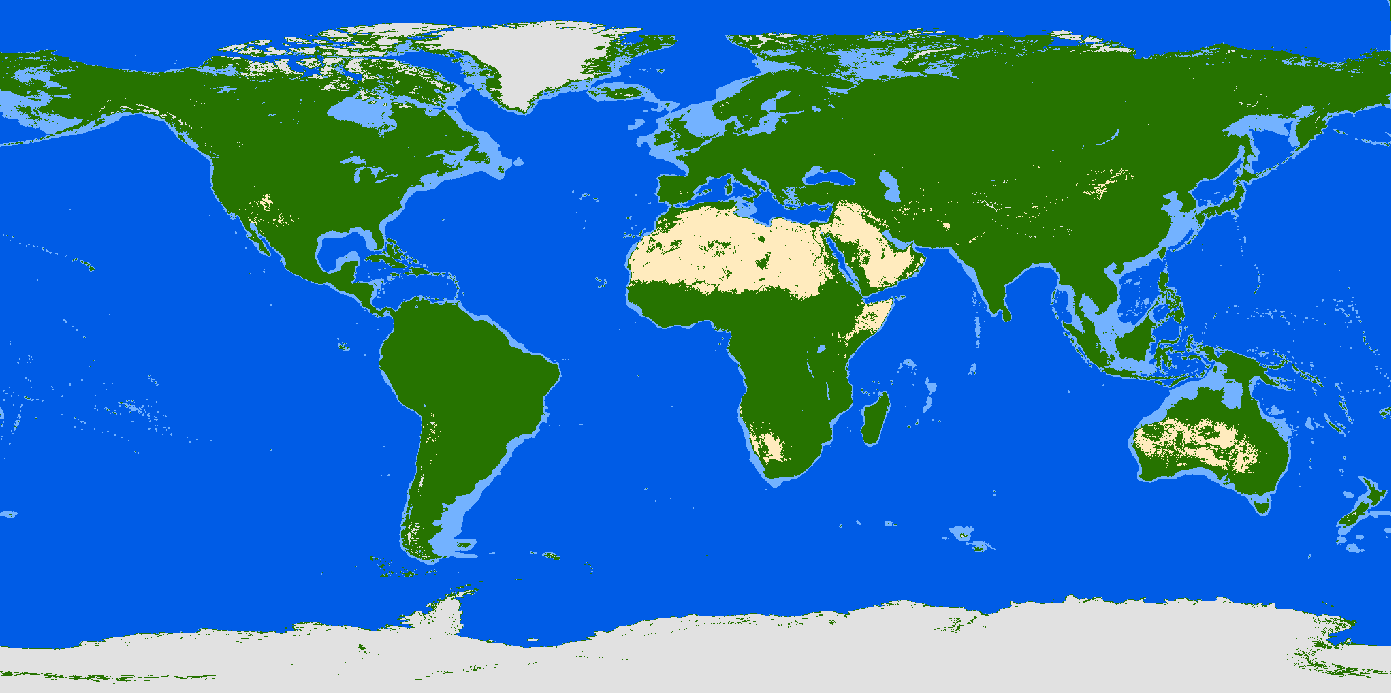
To start our own land use classification, we begin by georeferencing a high-resolution file of global satellite images (NASA, 2008) to a WGS84 coordinate system. After appending the appropriate coordinates to the global image, we use a form of supervised image classification in the GIS software ArcMap to begin the land use classification process. For this classification, we define 5 different types of global land use: water, ice, land, shallow water, and desert. For each of these classes, a number of different training samples are created on NASA’s georeferenced satellite map, as shown in figure 7.

*Fig 7: Land Use Classification Training Samples*



After defining the training set for each different land use type, maximum likelihood image classification is used to assign each global coordinate to one of the 5 land use classes, as shown below:

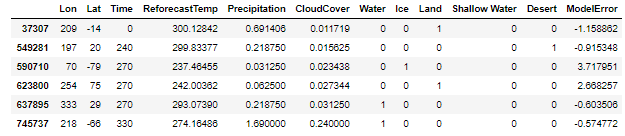
*Fig 8: Classified Global Land Use*



A confusion matrix accuracy assessment (Lina Yi & Guifeng Zhang, 2012) is conducted to calculate that the kappa accuracy coefficient for this classification is 0.91, which is above the generally accepted 0.81 threshold of “almost perfect agreement” (McHugh, 2012). Therefore, this land use classification dataset is determined to be sufficiently accurate to use in our analysis.

To implement the different land use classes into our training and test sets, 5 dummy variables are defined to represent the 5 different land use classes. Each of these dummy variables evaluates to 1 if that datapoint matches its corresponding land use class, and 0 otherwise. For example, if a coordinate’s land use is part of the ‘Ice’ class, then the ‘Ice’ dummy variable will equal 1, and all the other dummy variables will equal 0. These variables, in combination with the newly created Precipitation and Cloud Cover variables, are all appended to the existing datasets. All of this data is then used to create a huge Comma Separated Value (CSV) file that we will use to train our models, which is once again split into test and training sets.

*Table 1: an example of the training set CSV. ‘ModelError’ is the variable we are training to predict*

**

**Model Development**

*Linear Models*

Now that a plethora of training data has been obtained, we can start creating models that try to predict the climate model error based on this training data. The first statistical model we will use to try and predict this error is an Ordinary Least Squares (OLS) Multiple Linear Regression (MLR). The reason we start with a MLR model is that it is less of a “black box” model than more sophisticated ML algorithms, meaning that its coefficients can still be relatively well understood. This may provide us with relevant information about feature importance, and the suitability of a linear model for predicting climate model error. A MLR model attempts to quantify a linear relation between a dependent variable (model error) and a number of ßi variables, including a ß0 intercept, by minimizing the sum of squared errors (Nimon & Oswald, 2013). In general, a MLR model can be described as follows:

*Formula 2: General MLR*

Where ß0 is the intercept, Xi are the training variables, ßi are their corresponding coefficients, and ε is the error term. Fitting such a MLR model to predict the model error in our training set, the linear formula becomes:

*Formula 3: Fitted MLR model*

A number of interesting observations can be made from this linear relationship between the model error and the training variables. The first, striking observation is that the intercept for this MLR model starts off positive and rather high: at an initial error of 6.92, with almost all of the variables subtracting from this value. This indicates that a hypothetical datapoint with a high score in all training variables, (e.g., a high longitude, latitude, time, temperature, precipitation, and cloud cover value) would have a large, negative model error, whereas a datapoint with low variable scores would have a remarkably high model error. Important to keep in mind when analyzing these coefficients is that the land use variables (water, ice, land, shallow water, and desert) are in fact dummy variables; meaning that their relatively high values are not necessarily because of their importance, but rather because of their scale. With this in mind, it is still surprising how some land use classes appear way more important in determining model error than others. The land, water, and shallow water coefficients are all low values in the single digits, whereas the ice and desert coefficients (the most extreme climates) are equal to -2.23 and +1.92 respectively. This result is especially interesting given the fact that the temperature coefficient is negative, which seems to contradict the positive value for desert climates and the negative value for icy climates. These relatively high absolute coefficients for ice and desert areas could potentially indicate a large unreliability in climate model predictions for more extreme climates, but it could also point to the inability of a linear model to represent the complex patterns of climate model inaccuracy. In fact, our initial MLR model is calculated to have an adjusted-R2 value of 0.048, indicating that a linear model is not an effective way of representing the complex relationship between the different climate variables. Nevertheless, our MLR model did attain a p-value < 2.2E-16, giving us sufficient evidence to reject the null hypothesis and indicating that there is a statistically significant relationship between the different variables at the 99% confidence level. Moreover, using our trained model formula to predict the model errors of the test set and then subtracting these predicted errors from climate model temperatures reduces climate model RMSE from 2.90 to 2.48. This shows that the linear regression definitely has some skill in predicting model error, which proves hopeful for more complex non-linear models.

Before trying a drastically more complex approach however, we first import a number of Sckikit-Learn’s (Pedregosa et al., 2011) different regression models: Ridge, Lasso, and ElasticNet regression. We will not go as deep into their coefficient and test statistic analysis, as these regressions are simply different types of linear regression that attempt to solve some of the issues with OLS regression (Pedregosa et al., 2011). Rather, in the code provided as appendix with this paper, we import all of the aforementioned models into our working space, fit them on the training data, and use them to predict and reduce model inaccuracy as previously described. The accuracy reduction results provided each of the different models are shown below in table 2.

*Table 2: Error reduction by different regression types*

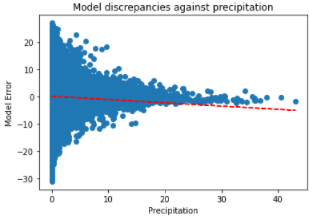
|  |  |
| --- | --- |
| **Regression type** | **RMSE** |
| Original Climate Model | 2.90 |
| Lasso Regression | 2.50 |
| Elastic-Net Regression | 2.50 |
| OLS Linear Regression | 2.48 |
| Ridge Regression | 2.47 |

From this table, it becomes apparent that all linear forms of regression are rather comparable in terms of RMSE reduction over this dataset. Only the ridge regression performs marginally better than the OLS regression, but nothing noteworthy. This seems to confirm our suspicion that linear models are inadequate to explain the complex patterns of climate model inaccuracy. Thus, we move on to more sophisticated deep-learning models.

*Deep-Learning Neural Networks*

*Random Forest*

**Results**



To the best of our knowledge, no random forest regression models have previously been used in climate model adjustment.

Data incest (find a nicer term for it)

in order to confirm that it

*Climate inaccuracy maps*

*Average climate inaccuracy: maybe we can use this to improve? Leads to->train/test*

*Train/Test split*

*Modeling*

***Linear Regression: make a nice formula and explain variables  
maybe temperature graphs?***

However,



Check Capstone proposals (x2) and ARW papers for things to write about and sources.

This model is a general model that can be reused. (mooier verwoorden, en vaker herhalen). Can be applied on any climate model’s outputs by changing the ModelData variable in the accompanying Python code. In this way, any climate model can generate its own tailor-made correction algorithm. The climate data that is to be inputted into ModelData needs to be in the shape of a NetCDF array, with the dimensions (Time, Longitude, Latitude) with the shape [422,360,180].

Alternatively, the data for this model can be found on github.com/timholthuijsen

Data Assimilation!

Search for (ecmwf) reanalysis papers

**Results**

*Average Model Error*

This map is really really interesting

*Table 2: a summary of all different models and their error*

|  |  |
| --- | --- |
| **Model** | **RMSE** |
| Original Climate Model | 2.90 |
| Average Error Reduction | 2.51 |
| Lasso Regression | 2.50 |
| Elastic-Net | 2.50 |
| OLS Linear Regression | 2.48 |
| Ridge Regression | 2.47 |
| Custom Neural Network | 2.33 |
| Multi-layer Perceptron Neural Network | 2.03 |
| Random Forest Regressor | **0.48** |
| Tuned Random Forest | **0.47** |

Has anyone ever done this before? Using Random Forest Regression to reduce climate model inaccuracy? It is wildly effective! How about Neural Networks? Has anyone done this?

this model can crude climate model prediction in combination with facts of our earth’s geography we know to be true, and turn it into more accurate climate predictions.

Write about Random Forest’s buildup and mechanism, and relate this to (contemporary) academic research! Maybe:  [Regression-Enhanced Random Forests](https://arxiv.org/pdf/1904.10416.pdf), and extrapolation problem. Extrapolation problem can be prevented by using Neural Networks, so we start using neural networks.

**Discussion**

Further research: include more climate variables for training!

Also: maybe more hyperparameter tuning and

Also: why is RF model so much better than all the other models?

Why is there such a big gap between the best and the second-best model?

ModelMaker.py

Appendix

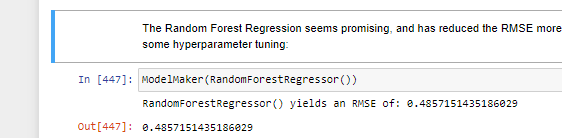
<https://github.com/timholthuijsen/ClimateModelCorrection>

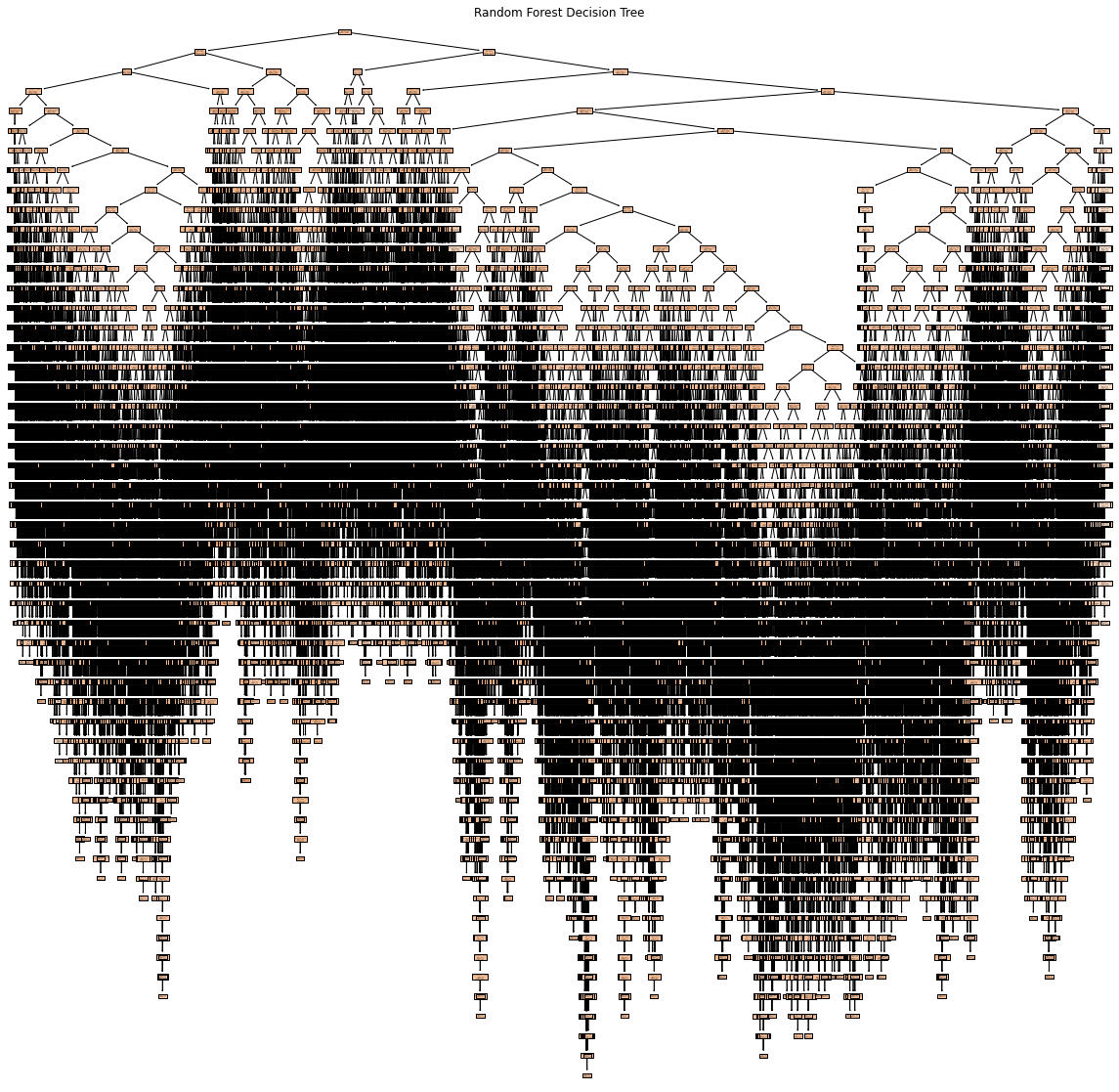
The reforecast and reanalysis datsets are too large to webhost, but can be found at x and y respectively.

Mention version 2 of GEFS! In intro?

Something to keep in mind is that it is an improvement if our climate models are closer to our reanalyses, but that a reanalysis isn’t perfect either.

Something

****

****

**Bibliography**

Copernicus Climate Institute. (2019). ERA5 monthly averaged data on single levels from 1979 to present. Retrieved November 11, 2019, from <https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-single-levels-monthly-means?tab=overview>

Earth Systen Research Laboratory. (2019). Download GEFS Reforecast v2 Ensemble Data. Retrieved November 11, 2019, from <https://www.esrl.noaa.gov/psd/forecasts/reforecast2/download.html>

Pawluszek-Filipiak, K., & Borkowski, A. (2020). On the Importance of Train-Test Split Ratio of Datasets in Automatic Landslide Detection by Supervised Classification. *Remote Sensing (Basel, Switzerland)*, *12*(18), 3054-. <https://doi.org/10.3390/rs12183054>

Rácz, A., Bajusz, D., & Héberger, K. (2021). Effect of Dataset Size and Train/Test Split Ratios in QSAR/QSPR Multiclass Classification. *Molecules (Basel, Switzerland)*, *26*(4), 1111-. <https://doi.org/10.3390/molecules26041111>

NASA. (2008, June 1). *SVS: Draining the Oceans*. <https://svs.gsfc.nasa.gov/vis/a000000/a003400/a003487/>

Lina Yi & Guifeng Zhang. (2012). *Object-oriented remote sensing imagery classification accuracy assessment based on confusion matrix*. 1–8. <https://doi.org/10.1109/Geoinformatics.2012.6270271>

McHugh, M. L. (2012). Interrater reliability: The kappa statistic. *Biochemia Medica*, *22*(3), 276–282.

Nimon, K. F., & Oswald, F. L. (2013). Understanding the Results of Multiple Linear Regression: Beyond Standardized Regression Coefficients. *Organizational Research Methods*, *16*(4), 650–674. <https://doi.org/10.1177/1094428113493929>

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. <https://doi.org/10.5555/1953048.2078195>

Copernicus Climate Change Service. (2020). *E-OBS daily gridded meteorological data for Europe from 1950 to present derived from in-situ observations* [Data set]. ECMWF. Retrieved May 9, 2019 from <https://cds.climate.copernicus.eu/cdsapp#!/dataset/insitu-gridded-observations-europe?tab=form>

<https://doi.org/10.24381/CDS.151D3EC6>

**VERY GOOD PAPER! USE THIS ONE!**

Vrac, M. (2018). Multivariate bias adjustment of high-dimensional climate simulations: The Rank Resampling for Distributions and Dependences (R 2 D 2 ) bias correction. *Hydrology and Earth System Sciences*, *22*(6), 3175–3196. <https://doi.org/10.5194/hess-22-3175-2018>

**Uses machine learning to improve parameterizations:**

Couvreux, F., Hourdin, F., Williamson, D., Roehrig, R., Volodina, V., Villefranque, N., Rio, C., Audouin, O., Salter, J., Bazile, E., Brient, F., Favot, F., Honnert, R., Lefebvre, M.-P., Madeleine, J.-B., Rodier, Q., & Xu, W. (2021). Process‐Based Climate Model Development Harnessing Machine Learning: I. A Calibration Tool for Parameterization Improvement. *Journal of Advances in Modeling Earth Systems*, *13*(3), n/a. <https://doi.org/10.1029/2020MS002217>

**Can models that are based on deep learning and trained on atmospheric data compete with weather  
and climate models that are based on physical principles and the basic equations of motion?**

Dueben, P. D., & Bauer, P. (2018). Challenges and design choices for global weather and climate models based on machine learning. *Geoscientific Model Development*, *11*(10), 3999–4009. <https://doi.org/10.5194/gmd-11-3999-2018>

Tool for Parameterization Improvement. *Journal of Advances in Modeling Earth Systems*, *13*(3), n/a. <https://doi.org/10.1029/2020MS002217>