Carbon Flux Prediction Under Low-Data Regime Using Meta-Learning

Juan Nathaniel

December 23, 2021

1 Introduction

Terrestrial ecosystem serves as an important climate regulator with carbon and energy being exchanged at the atmosphere constantly. As a persistent and large carbon sink, terrestrial ecosystem removes approximately 30% of the fossil-fuel CO_2 generated globally [4]. However, this estimate varies greatly across geographical regions and time due to the high variability of photosynthesis and the uncertainty of flux measurements.

Climate data, including flux observation, tends to be imbalanced across climate and geographical regions, especially in the tropics. Even if a good amount of data exists, the quality may be compromised due to noisy environmental conditions or issues with instruments. These issues make learning from data challenging, and a laborious amount of effort needs to be spent to clean and fix the data.

Nonetheless, numerous data-driven deep learning methods have been proposed to transfer the learning of climate observations in data-rich regions/zones to data-scarce areas, such as the tropics. However, many require a separate process of learning and transferring. Only until recently, the idea of learning how to learn is brought forward. The concept is inspired from our own ability to learn: we are able to generalize and adaptively apply our learnings to a new domain quickly. For instance, if one is well-versed in a particular language, then one is highly likely to adapt to a new dialect quickly with sparse learning data points.

This report, therefore, attempts to adaptively and quickly learn features of carbon flux data from data-rich zones and applies these learnings to a relatively new domain (ie. data-poor regions). Essentially, we are exploring the idea of meta-learning, or the ability to learn how to learn efficiently.

2 Data

2.1 FLUXNET data

For carbon flux data, I'm going to use FLUXNET2015. The dataset consists of either half-hourly and hourly measurements across 500 sites. FLUXNET2015 uses eddy covariance methods to measure carbon dioxide, water vapor, and energy exchanges between the atmosphere and terrestrial ecosystems. The dataset has multiple improvements over its predecessor (eg. FLUXNET Marconi Dataset, 2000 or FLUXNET LaThuile Dataset, 2007) including better monitoring of on-the-ground stations, improved uncertainty quantification methods, and gap-filling strategies by using re-analysis products.

In addition to measuring carbon and energy fluxes, FLUXNET2015 captures useful meteorological and eco-hydrological variables including air temperature, incoming and outgoing shortwave radiation, vapor pressure deficit, atmospheric pressure, precipitation, wind speed, and soil water content.

The distribution of FLUXNET stations is illustrated in Figure 1. I superimpose with Köppen classification map [1] to identify climate characteristics of each station. The three categories I'm using include the tropical, arid, and temperate stations. This classification would be useful when defining a *Class* as described in Section 3.1.1

2.2 MODIS satellite data

Leaf Area Index (LAI) and the fraction of photosynthetically active radiation (fPAR) are key variables that show strong physical relationship with terrestrial carbon flux. LAI is defined as the one-sided green leaf area per unit ground area in a broadleaf canopy setting (or half of that in coniferous canopy). On the other hand, fPAR as the term suggests, describes photosynthetic activities. Both variables are commonly used to infer the rate of photosynthesis, evapotranspiration, and gross/net primary production - all key components to carbon cycle.

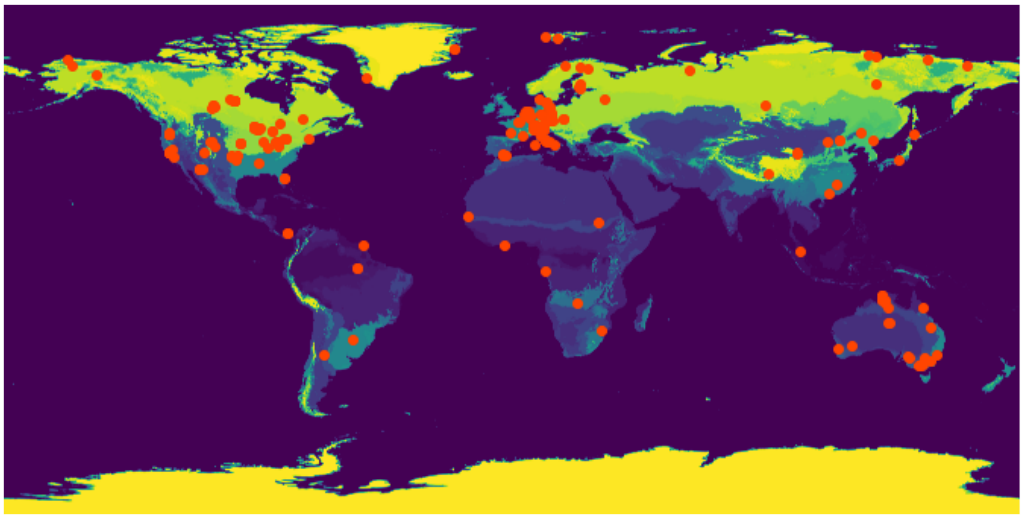

Figure 1: Distribution of FLUXNET stations against Köppen climate classification

One method to obtain LAI and fPAR is through remote sensing data, such as Sentinel-2 or Moderate Resolution Imaging Spectroradiometer (MODIS). For this report, I'm going to use MODIS to extract both variables across time and for each FLUXNET2015 station.

2.3 Data processing

The target variable, y, that I'm interested in would be the carbon flux as captured by the gross primary production (GPP) variable in FLUXNET2015. The independent variables that I'm considering will include LAI, fPAR, air temperature, incoming and ourgoing shortwave radiation, vapor pressure deficit, atmospheric pressure, precipitation, wind speed, and soil water content. Thereafter, some preproccessing steps are done:

1. Replacing the value -9999 with NULL to ease further processing
2. Removing gap-filled data where the quality score is above 2 (1 indicates the most accurate, less uncertain gap-filled value)
3. Applying a backfill method (ie. using the next available observation to fill in the current gap)
4. Performing a z-normalization procedure for all variables, where:

$$z - normalized\ X = \frac{X - E(X)}{\sigma_X}$$

where $E(X)$ is the expected value for the variable X, and σ_X is the standard deviation of said variable X.

3 Methodology

3.1 Models

3.1.1 Meta-learning

I'm using an optimizer-based meta-learning approach adapted from Model-Agnostic Meta Learning (MAML) algorithm [2]. The framework for meta-learning is similar to a canonical deep learning approach with salient differences. As illustrated in Figure 2, instead of having the conventional training and testing phases, we have the meta-training and meta-testing phases. In addition, instead of refering to training and testing data, we have the support and query sets respectively. The difference stems from the fact that in meta-learning, a finetuning (or training in machine learning context) needs to be done during both meta-training and meta-testing phase on the support set before inference could be done on the query set. The term finetuning is used as the support set normally has very few data (eg. less than 1% of the whole available dataset); and is often termed few-shot learning.

One might wonder, *training during meta-testing phase*? This is where the term Class as illustrated in Figure 2 falls in. The key idea behind meta-learning is in its ability to extract important features during meta-training phase and apply these learnings, in conjunction with finetuning given a minimal size of support set, for one or more previously unseen classes.

This idea of meta-learning or *learning how to learn* can be achieved by implementing two training loops: the meta-learning inner loop and the adaptation outer loop. They are briefly described as follows:

1. *Inner meta-learning loop*: for each of the n classes in meta-training dataset (eg. stations in data-rich non-tropical regions), we are iteratively making predictions using the support set, comparing against ground-truth, computing the loss and gradient, before proposing a new set of model parameters, θ'.

2. *Outer adaptation loop*: once the inner loop finishes, we update the initial parameters, θ, by making prediction on all classes using the query set and θ', and computing the loss and gradient.

In meta-testing phase, only the inner meta-learning loop happens but without the gradient descent computation.

The psuedocode for meta-training is described in Algorithm 1, adapting from Model-Agnostic Meta Learning (MAML) [2].

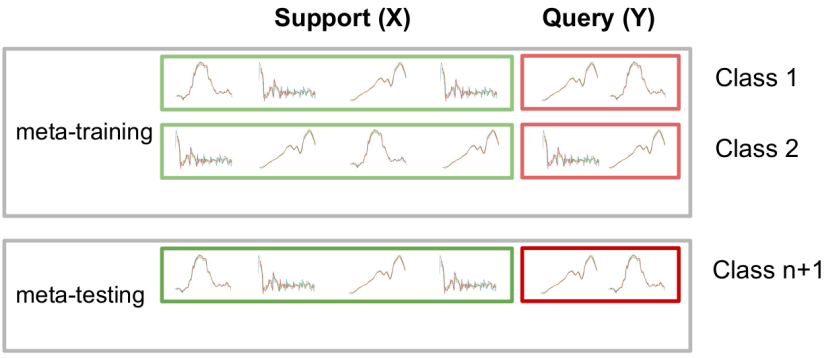

Figure 2: General data structure for meta-learing

Algorithm 1 MAML meta-training adapted approach

Require: α, β ▷ α and β are the models' and meta-learner learning rates

randomly initialize θ ▷ θ is the models initial parameters

while not done **do**

  Sample batch of Class $C_i \sim p(C)$

  for all C_i **do**

    Sample K datapoints $D = \{x^j, y^j\}$ from C_i ▷ support set

    Sample K datapoints $D' = \{x^j, y^j\}$ from C_i ▷ query set

    infer x^j and evaluate loss with y^j, $\mathcal{L}_{C_i}(f_\theta)$ from set D

    Compute the gradient, $\nabla_\theta \mathcal{L}_{C_i}(f_\theta)$

    Compute adapted parameters, $\theta' \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{C_i}(f_\theta)$ with gradient descent

  end for ▷ inner meta-learning loop is done

  Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{C_i \sim p(C)} \mathcal{L}_{C_i}(f_{\theta'})$ using D' ▷ outer adaptation loop is done

end while

3.1.2 Backbone models

For this project, we are considering three different backbone models for our meta-learning approach. The complete architecture description is summarized in Table 1:

1. Multi Layer Perceptron (MLP) or the feed-forward neural network: is a universal function approximator that can explain the non-linear relationship between variables well. MLP typically consists of 3 or more layers, activated by non-linear functions: the input, hidden, and output layer. In this report, we are going to use 3 hidden layers, each activated by a non-linear Leaky Rectified Linear Unit (LeakyReLU) activation function. The function outputs the positive values of inputs, with a small slope for negative values.

2. Long-Short Term Memory (LSTM): is capable of learning the long-term dependencies of sequential data such as timeseries. It is a type of recurrent neural network (RNN) and is widely popular among deep learning researchers and scientist for data with temporal elements to it. LSTM is first proposed in 1997 [3] to tackle the issue of vanishing or exploding gradient in RNN by introducing the idea of a forget gate. This method allows for greater contextual information to be inferred as longer sequences of data can now be learned.

3. Bi-directional LSTM (BiLSTM): is made up of two LSTM layers where the first layer takes input in the forward direction while the second layer takes input in the backward direction, both of which are then connected to the same output layer. In essence, for every point in the input sequence, BiLSTM has complete information of all other points preceeding or proceeding it. Similar to LSTM, BiLSTM allows for more data to pass into the network, enabling greater contextual insight. Many real-world problems require context in the past and in the future to make predictions in the present. For instance, language translation requires past and future contextual information to generate the best recommendation for the current word or sentence. Similarly in climate sciences, past observations are key to our understanding of the present and future. But future observation has also helped us to update and re-assimilate real dynamics into our models.

Table 1: Summary of architecture for the backbone models

	MLP	LSTM	BiLSTM
Architecture	FCN (18 x 128) LeakyReLU FCN (128 x 128) LeakyReLU FCN (128 x 64) LeakyReLU FCN (64 x 64) LeakyReLU FCN (64 x 1)	LSTM (18 x 64) LeakyReLU FCN (64 x 1)	BiLSTM (18 x 64) LeakyReLU FCN (128 x 1)

4 Results

4.1 Meta-learning preliminary evaluation

First, I'm going to analyze the performance of our backbone models with and without meta-learning. In this analysis, I'm going to use 48-hour timeseries data for the independent variables to predict the instantaneous carbon flux observation (GPP) at the 48th hour. As illustrated in Figure 3, models with meta-learning perform better than those without. Especially in the tropics where larger improvement in terms of reduced validation mean-squared error (MSE) is apparent.

I performed multiple model runs to get the average performance as summarized in Table 2. Overall, BiLSTM models perform better than LSTM and MLP, with MLP performing worse than the other two model architectures. In addition, GPP predictions in the tropics are more difficult than that in the arid or temperate stations, with higher validation MSE scores overall. In particular, the validation MSE for GPP prediction in the tropics *without* meta-learning approach ranges between 0.371 and 0.490 as compared to that in the arid (0.202-0.366) and in the temperate zone (0.208 - 0.217).

Meta-learning approach improves models performance across architectures and climate zones. The validation MSE in the tropics with meta-learning approach ranges between 0.228 and 0.293, compared to 0.371 and 0.490 without meta-learning. Similar trends are observed in the arid (0.162-0.223 with meta-learning *versus* 0.202-0.366 without meta-learning) and temperate (0.197-0.217

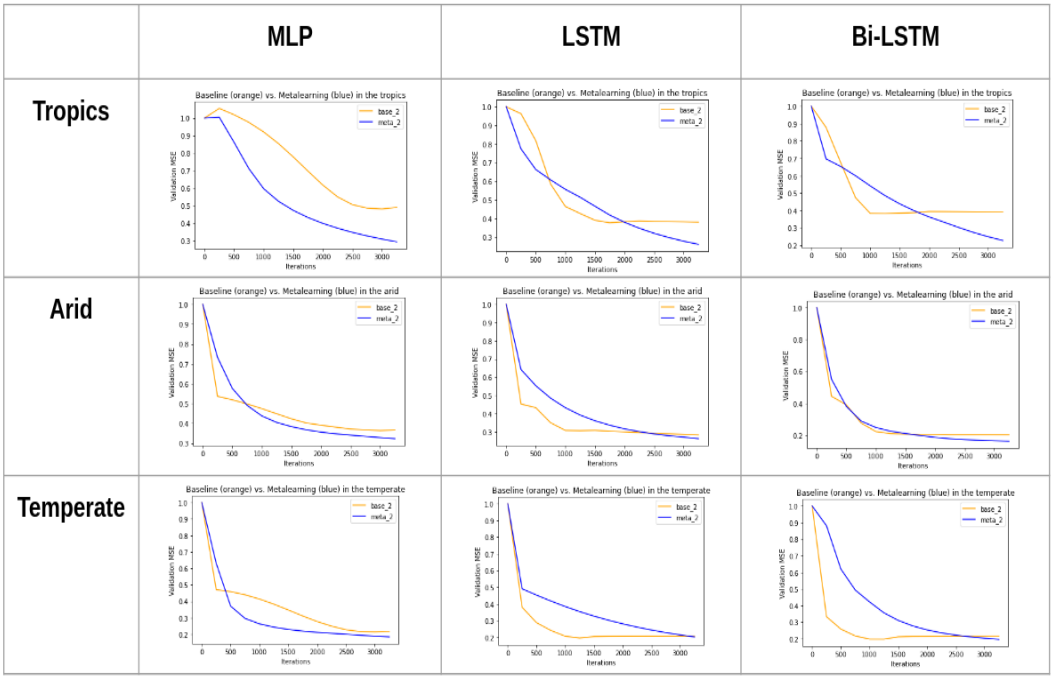

Figure 3: Convergence of models with and without meta-learning algorithm

with meta-learning *versus* 0.208-217 without meta-learning) regions.

Table 2: Meta-learning performance using validation MSE

	MLP		LSTM		BiLSTM	
	Base	Meta	Base	Meta	Base	Meta
Tropics	0.490 ± 0.05	0.293 ± 0.08	0.381 ± 0.04	0.261 ± 0.15	0.371 ± 0.02	0.228 ± 0.15
Arid	0.366 ± 0.05	0.223 ± 0.18	0.283 ± 0.08	0.262 ± 0.06	0.202 ± 0.02	0.162 ± 0.127
Temperate	0.217 ± 0.02	0.202 ± 0.25	0.208 ± 0.02	0.202 ± 0.25	0.217 ± 0.02	0.197 ± 0.13

However, one might notice the larger deviation across model runs for models with meta-learning. For instance, a deviation of ± 0.18, ± 0.15, and ± 0.13 are observed for meta-MLP in the arid, meta-LSTM in the tropics, and meta-BiLSTM in the temperate regions respectively. The deviation seems to be independent of climate zones and the underlying model architecture. One suspicion might be due to the increased stochasticity in our meta-learning approach, especially in the inner loop where new set of model parameters, θ',

is proposed for every class. In order to test this hypothesis, I'm going to run similar analysis but using varying sequence length. Specifically, I'm going to use 7 days, 30 days, and 60 days temporal scale to build the timeseries of our independent variables. The choice of length is to account for near, subseasonal, and seasonal pattern in climate science respectively.

4.2 Meta-learning under different temporal scale

Similar to the earlier section, I'm going to analyze the performance of our backbone models with and without meta-learning. In addition to using 48-hour timeseries data for the independent variables to predict the instantaneous carbon flux observation (GPP), I'm going to vary the length of our series at 7 days, 30 days, and 60 days to represent near, subseasonal, and seasonal cycles. Similar to our earlier analysis and as illustrated in Figure 4, models with meta-learning perform better than models without especially in the tropics.

Overall, models with meta-learning improves carbon flux prediction as compared to models without, across temporal scales. Models without meta-learning tend to be invariant to different sequence lengths. However, models with meta-learning seem to have better performance as the length of our timeseries increases. This suggests that meta-learning not only learns how to learn important predictors across different station classes, but also how each one relates in the temporal dimension.

I also performed multiple model runs to get the average performance as summarized in Table 3. Overall, BiLSTM models perform better than LSTM and MLP. Predictions in the tropics consistently show higher error across temporal scales. In addition, the large deviation of our meta-learning approach diminishes as the length of the timeseries is increased beyond 7 days. For instance, deviations of ± 0.18, ± 0.15, and ± 0.13 as observed in shorter sequence length for meta-MLP in the arid, meta-LSTM in the tropics, and meta-BiLSTM in the temperate are reduced to ± 0.09, ± 0.03, and ± 0.05 respectively in a longer timescale.

One final question worth exploring is how robust meta-learning is under uncertainty and extreme events. In order to answer this question, I'm going to perform a series of robustness tests as described and presented in the next section.

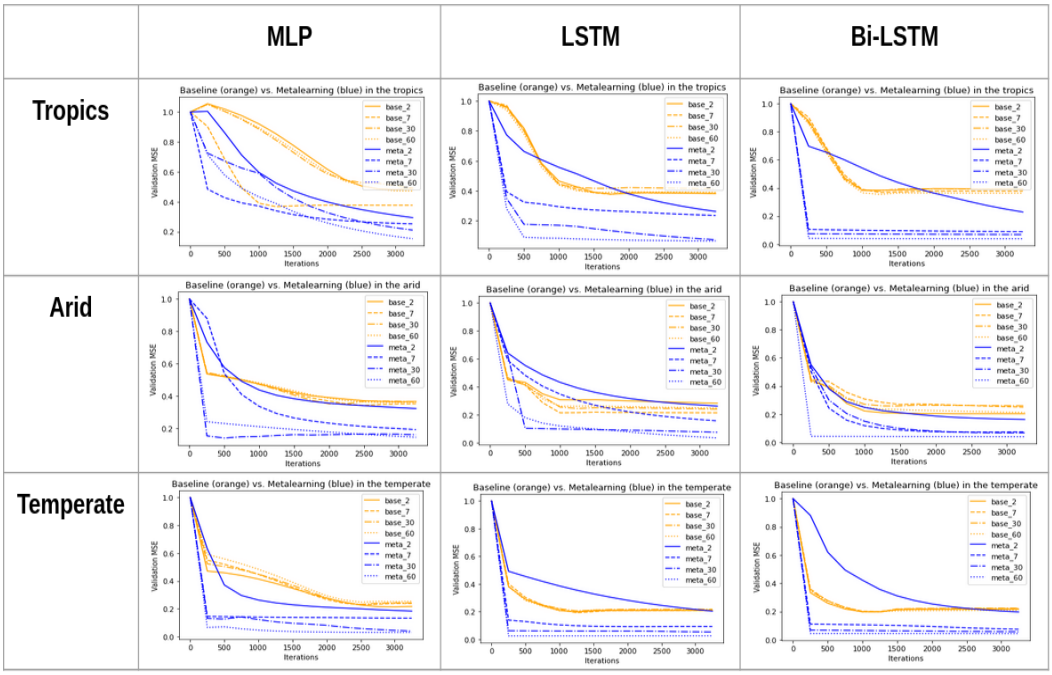

Figure 4: Convergence of meta-learning algorithm across different temporal scale

4.3 Meta-learning robustness under uncertainty and extremes

To carry out the experiments, I will use three different concepts to define uncertainty and extremes.

1. *Noisy data*: represents an actual climate where observation tends to be noisy. To generate noisy inputs, I'm going to compute the mean (μ) and standard deviation (σ) for each of the z-normalized feature, fit a normal distribution, sample random values, and add them up to the actual input variables. I'm going to vary the standard deviation when fitting the normal distribution by factor n, $n.\sigma$, where n ranges from 0.3 to 1.5 and incremented by 0.1. The greater the $n.\sigma$ is, the wider the normal distribution, and the noisier the variable becomes.

2. *Shifted data*: represents errors that are caused by irregular measuring devices. To generated shifted inputs, I'm going to compute the standard deviation (σ) for each of the z-normalized feature, and shift the input

Table 3: Meta-learning performance evaluation across temporal scales

	MLP		LSTM		BiLSTM	
	Base	Meta	Base	Meta	Base	Meta
Tropics (2-day)	0.490 ± 0.05	0.293 ± 0.08	0.381 ± 0.04	0.261 ± 0.15	0.371 ± 0.02	0.228 ± 0.15
Tropics (7-day)	0.477 ± 0.03	0.252 ± 0.03	0.384 ± 0.02	0.234 ± 0.12	0.377 ± 0.03	0.088 ± 0.05
Tropics (30-day)	0.491 ± 0.05	0.210 ± 0.03	0.413 ± 0.03	0.072 ± 0.03	0.392 ± 0.02	0.068 ± 0.08
Tropics (60-day)	0.471 ± 0.07	0.153 ± 0.14	0.389 ± 0.02	0.064 ± 0.04	0.363 ± 0.01	0.038 ± 0.04
Arid (2-day)	0.366 ± 0.05	0.223 ± 0.18	0.283 ± 0.08	0.262 ± 0.06	0.202 ± 0.02	0.162 ± 0.127
Arid (7-day)	0.353 ± 0.05	0.192 ± 0.26	0.214 ± 0.02	0.157 ± 0.08	0.205 ± 0.03	0.073 ± 0.04
Arid (30-day)	0.363 ± 0.03	0.160 ± 0.09	0.241 ± 0.08	0.076 ± 0.02	0.209 ± 0.08	0.067 ± 0.04
Arid (60-day)	0.364 ± 0.03	0.145 ± 0.11	0.248 ± 0.07	0.035 ± 0.02	0.211 ± 0.04	0.039 ± 0.03
Temperate (2-day)	0.217 ± 0.02	0.202 ± 0.25	0.208 ± 0.02	0.202 ± 0.25	0.217 ± 0.02	0.197 ± 0.13
Temperate (7-day)	0.237 ± 0.02	0.132 ± 0.15	0.208 ± 0.02	0.091 ± 0.08	0.220 ± 0.01	0.074 ± 0.07
Temperate (30-day)	0.242 ± 0.02	0.04 ± 0.05	0.216 ± 0.01	0.052 ± 0.03	0.224 ± 0.01	0.057 ± 0.09
Temperate (60-day)	0.254 ± 0.02	0.03 ± 0.05	0.213 ± 0.01	0.025 ± 0.02	0.208 ± 0.01	0.044 ± 0.05

data by $n.\sigma$, where n ranges from 0.3 to 1.5 and incremented by 0.1. The greater $n.\sigma$ is, the greater a shift each variable is undergoing.

3. *Extremes*: represents carbon fluxes that occur infrequently. This is done by keeping z-normalized flux observations that are larger than a threshold t with their corresponding input features. Here, t ranges between 1.0 and 2.0 and incremented by 0.1. A higher t represents a more extreme data being kept in our set.

The above analysis is going to be done using flux data in the tropics where predictions are the most difficult. A 48-hour timeseries data is used comparing both MLP and LSTM models with or without meta-learning approach. Figure 5, 6, and 7 illustrate the robustness test under noisy, shifted, and on extremes dataset respectively.

From the three figures, models with meta-learning approach (blue lines) perform better despite having its underlying input data altered. When I shift, add noise, or perform prediction on extreme fluxes, the percentage change in inference error is lower than models without meta-learning. The relative robustness for the prediction of extreme fluxes also shows promising sign for meta-learning approaches in generalizing learning. In all, meta-learning approach seems to improve the robustness of inference under few-shot learning environment.

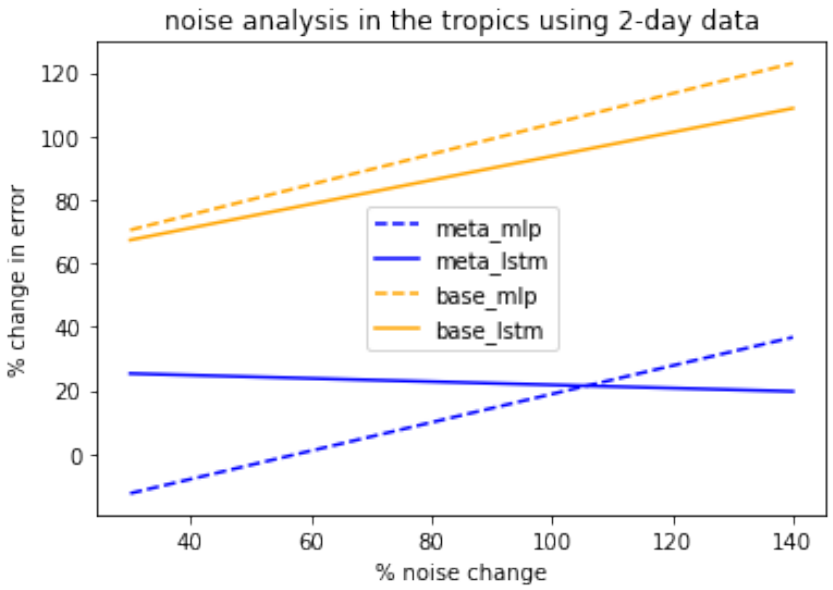

Figure 5: Evaluating the robustness of meta-learning approach under noisy dataset

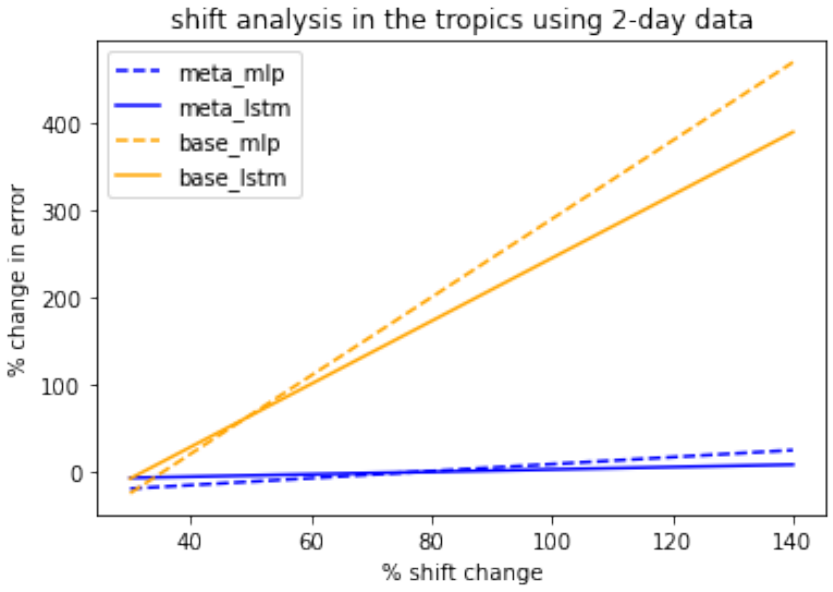

Figure 6: Evaluating the robustness of meta-learning approach under shifted dataset

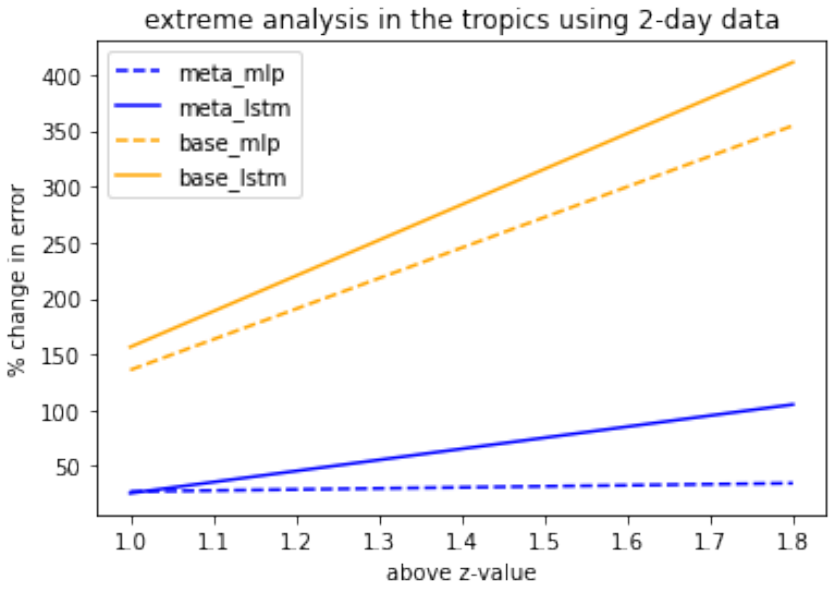

Figure 7: Evaluating the robustness of meta-learning approach for extreme dataset

4.4 Prediction Skill

In this section, I'm going to highlight the performance of models with and without meta-learning in predicting GPP flux. As observed in Figure 8, models with meta-learning is able to follow the trend of the actual observation closer than those without meta-learning. However, one limitation shared between the two approaches is in the inability to predict extreme cases as well (ie. for z-normalized GPP flux above 1.5). This might be caused by the limited amount of data that I use (1%) for training.

5 Conclusion

I have explored the use of meta-learning to enable deep learning models to learn efficiently how to learn from very limited and often spurious amount of data. We have explored how meta-learning could achieve better performance in previously unseen station classes. In particular, we classified FLUXNET stations based on their climate zones and perform meta-testing on each one of these zones. Our analysis found that performing GPP flux predictions in the tropics are the most challenging. In addition, I compared the performance of different deep learning models including MLP, LSTM, and BiLSTM.

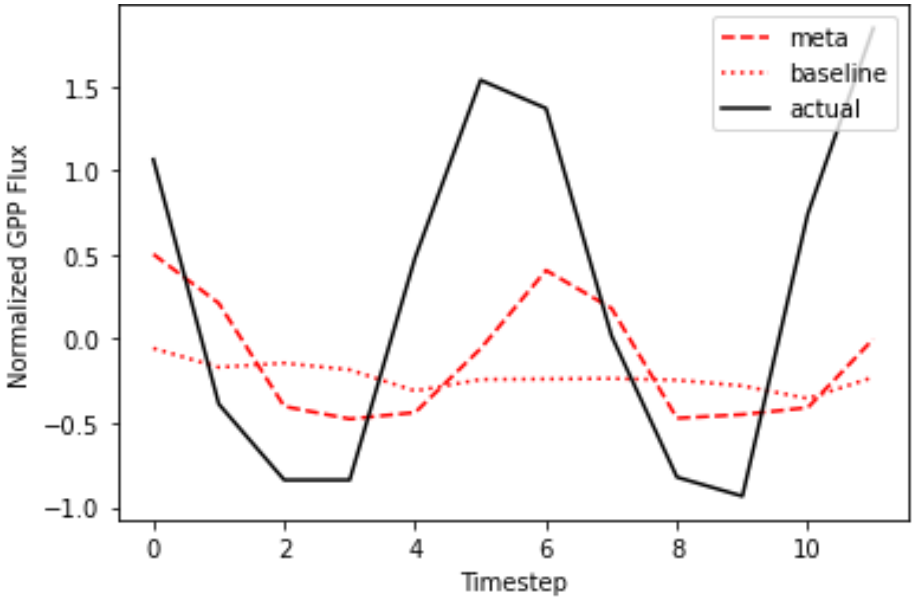

Figure 8: Prediction skill of models with and without meta-learning approach

As expected, BiLSTM outperforms LSTM which in turn fares better than MLP.

Furthermore, this report highlighted the effect of increasing temporal scales in improving the performance of meta-learning approach. This suggests that meta-learning not only learns important, shared features, but it captures temporal structure effectively. Given the improved performance of meta-learning approach, a series robustness test is performed. For this report, three tests are proposed: adding noise, shifting of the underlying input dataset, and performing predictions on extreme fluxes. As compared to models without meta-learning, those with the approach are more robust as measured by the lower percentage increase in error.

However, models with meta-learning does not seem to explain extremes as well despite performing better than models without. This, however, could be attributed to the low data regime that we have started out with. Overall, meta-learning seems to offer promising results especially in the climate sciences domain where data tends to be sparse, limited, and heavy-tailed.

References

[1] Hylke E Beck, Niklaus E Zimmermann, Tim R McVicar, Noemi Vergopolan, Alexis Berg, and Eric F Wood. Present and future köppen-geiger climate classification maps at 1-km resolution. *Scientific data*, 5(1):1–12, 2018.

[2] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[4] Corinne Le Quéré, Robbie M Andrew, Pierre Friedlingstein, Stephen Sitch, Judith Hauck, Julia Pongratz, Penelope A Pickers, Jan Ivar Korsbakken, Glen P Peters, Josep G Canadell, et al. Global carbon budget 2018. *Earth System Science Data*, 10(4):2141–2194, 2018.