MACHINE LEARNING BASED STATISTICAL DOWNSCALING FOR RAINFALL ON HAWAIIAN ISLANDS

A THESIS SUBMITTED TO THE GRADUATE DIVISION OF THE UNIVERSITY OF HAWAIʻI AT MĀNOA IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

COMPUTER SCIENCES

DEC 2022

By

Yusuke M. Hatanaka

Thesis Committee:

Peter Sadowski, Chairperson
Mahdi Belcaid
Thomas Giambelluca

ABSTRACT

Long-term rainfall prediction on Hawaiian islands in the scale of up to decades is a crucial task for water resource management. The current physics based climate models only produce coarse outputs, which are not suitable to the islands due to high rainfall gradient. Statistical downscaling is a method of learning a model to perform super-resolution on weather and climate variables; predicting local weather and climate from coarse resolution variables. This project focuses on rainfall data, and aims at building a framework for statistical downscaling using historical reanalysis data in coarse resolution.

Statistical downscaling is typically done using linear regression models. Here we test the use of machine learning methods such as decision trees and neural networks, which are underutilized for this application. Given a set of coarse inputs, non-linear machine learning models are trained to make rainfall predictions.

In this study, we compare machine learning methods for statistical downscaling on a large historical dataset for Hawai'i's rainfall. In Chapter 2, the dataset used for this project is explained. In Chapter 3, explanations on each method are provided. Chapter 4 iterates the result on feature selection and experiment on site-specific models. It also has a followup on the site-specific experiment, where the effect of sample size on machine learning methods is examined.

Our results show that neural networks are able to improve upon linear regression prediction. However, while this is true in aggregate, there are some cases where linear regression is superior to neural networks, typically when there is not much data.

Overall, this project provides a demonstration of the capabilities and limitations of non-linear machine learning methods, establishing the initial milestone on improvement on statistical downscaling research to follow.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Long-term rainfall prediction on the Hawaiian islands is a crucial task for water resource management as the livelihood of the people living on the islands is highly dependent on rainfall. In fact, 99% of domestic water is from groundwater, which comes from rainfall and cloud-water interception [5]. However, how the changing climate will affect the water resources is still unknown. Climate scientists study the effects of climate change using General Circulation Models (GCMs). These are long-running simulations of global climate variables. However, they are very expensive simulations and done in a coarse manner that doesn't capture local micro-climates of places like Hawai'i. The mountainous topography of the islands change the weather in ways that are unaccounted for by the GCMs. This causes high rainfall gradients [7], resulting in micro-climates across the islands. Fig 1.1 shows mean annual rainfall of Hawai'i, with measurements taken from 100+ stations and smoothed for visualization. Even within the same island, there is a great diversity in the amount of rainfall. Estimating the future rainfall at high resolution is necessary for accurately predicting the future water budget.

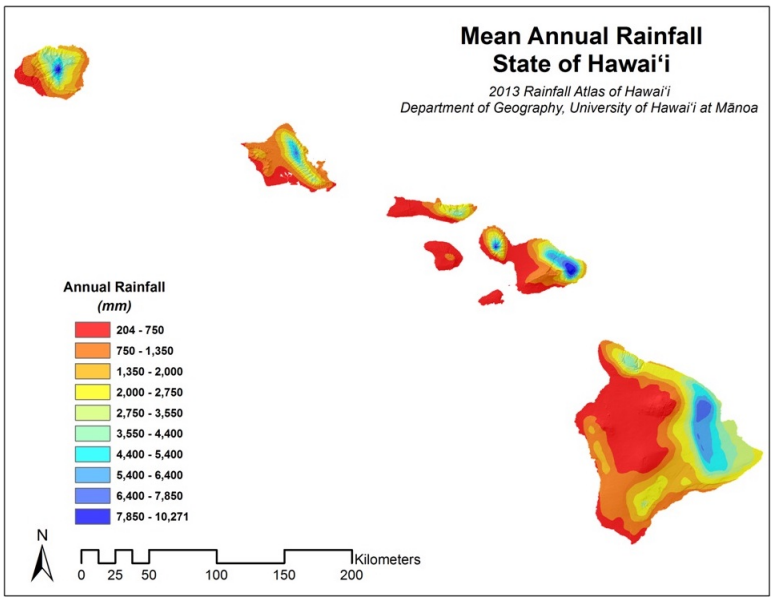

Figure 1.1: Mean annual rainfall of Hawai'i [4]. Hawai'i exhibits high rainfall gradients, requiring predictions at high resolution, as these patterns cannot be captured by GCM.

Statistical downscaling is a method of making predictions at high resolution based on coarse resolution inputs. Linear statistical models have been applied to climate data over Hawai'i, but exploration of non-linear models still has not extensively been conducted. Recent studies have seen advances in machine learning methods such as neural networks and tree methods such as XGBoost,

which can effectively learn non-linear and thus complicated representation of the data. In this study, such non-linear algorithms are applied to rainfall prediction over Hawai‘i in order to assess their effectiveness. Random forest, XGBoost, and neural networks are tested as non-linear methods while being compared against linear regression models as the performance baseline.

CHAPTER 2
DATASET

2.1 Rainfall Data

The rainfall data includes monthly rainfall data collected from over 2,000 rain gauges across Hawai‘i up to 2012, after which gap filling was applied to fill the missing data [4]. As explained in the next section, the reanalysis data is available starting 1948. Therefore, only the subset of the rainfall data collected between 1948 and 2012 is used for this study, which means that the maximum number of rainfall observations that a single station could have is 780. This resulted in 1992 unique weather stations from Kauai, Oahu, Molokai, Maui, Kahoolawe, and Hawai‘i. Observations are monthly rainfall for each station in inches, accompanied with latitude and longitude coordinates and elevation of the weather station. Total of 865,537 month-station samples are available, of which 52% (453,285 samples) are gap-filled data. Of all 1992 stations between 1948 and 2012, there are 78 stations where all of rainfall data is obtained by gap filling. Although 780 is the maximum number of rainfall samples per station, the rainfall dataset has a significant number of missing data. Figure 2.1 shows the histogram of the number of samples available per station, of the entire dataset and of the actual observations. If we exclude the gap-filled data, only a few stations have more than 600 samples per station.

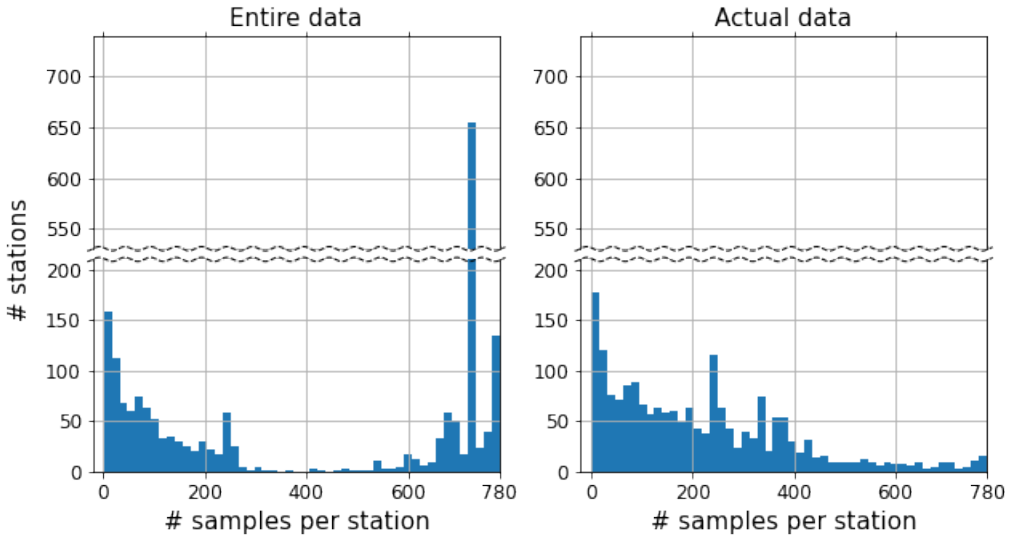

Figure 2.1: Number of samples available per station. *left*: histogram of the entire dataset. *right*: histogram of the subset of the data excluding the gap-filled data. A large portion of the rainfall data is gap-filled.

2.2 Reanalysis Data

Climate data has been measured across the globe for a long time, but their format and geographical configurations are not consistent. For example, different instruments would record different climate variables at different periods of times. Reanalysis data is an attempt to impose geographical and temporal consistency in climate variable measurements. Taking actual observations across the globe, climate variables are re-calculated using physics based climate models to end up in measurements at consistent timestamps and geographical grid.

In order to make rainfall predictions on specific weather stations, 16 of commonly used reanalysis variables are used. These variables are suggested by the climate science collaborators, including but not limited to air temperature at different levels, air pressure, humidity, surface temperature, etc. The complete list is provided in Table B.1 in Appendix. The dataset is published by National Centers for Environmental Prediction (NCEP)/National Center for Atmospheric Research (NCAR) [6]. Reanalysis data consists of monthly mean observations on 2.5° by 2.5° grid across the globe since 1948. Unless explicitly specified, the model input is the closest observation from each station per each of the 16 reanalysis variables. Therefore, a vector of length 16 describes the state of the atmosphere for each data sample. As shown in the figure 2.2, such resolution is not appropriate to capture local climate patterns, as the whole island of Oahu is explained by a single cell. Therefore, machine learning models take such coarse observations and try to make rainfall predictions on each weather station.

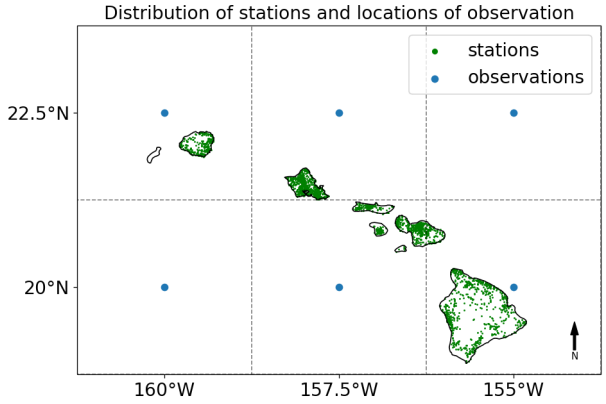

Figure 2.2: Distribution of weather stations and locations of observations from reanalysis data. While weather stations are densely distributed across the islands, the reanalysis measurements are coarsely distributed.

2.2.1 Linear Interpolation

One way of mitigating the effect of coarse resolution observation is to fill the gap with linear interpolation. This increases the resolution of observation in the form of simple approximation.

In order to incorporate observations from around the islands, observations ranging in 152.5°W to 162.5°W and 15°N to 25°N are used for computing the linear interpolation. Two different resolutions are tested; one that splits the entire region in 50 by 50 cells, and another that splits the same region in 100 by 100 cells. Figure 2.3 compares the resolutions of original observation, 50 by 50, and 100 by 100 interpolation.

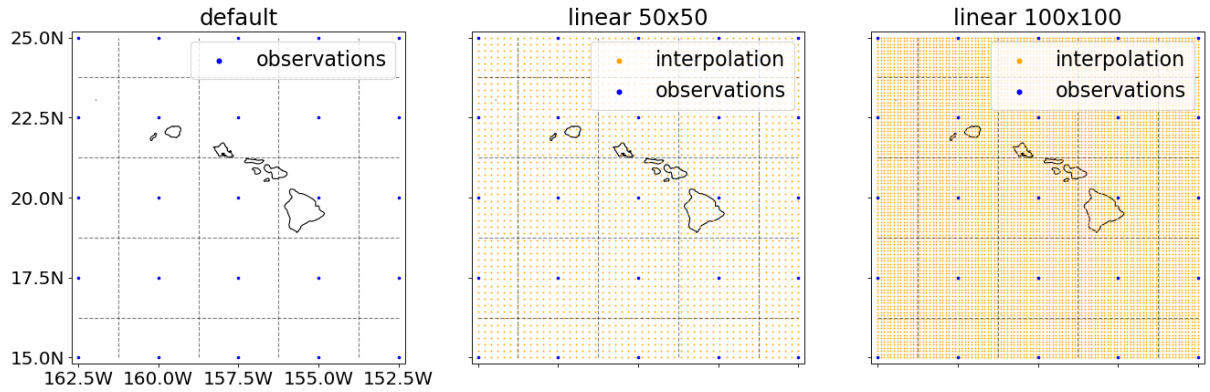

Figure 2.3: Comparison of resolution of reanalysis data. *left*: original resolution. *middle*: linear interpolation in 50 by 50 resolution. *right*: linear interpolation in 100 by 100 resolution.

The model input for each station is now the closest data obtained from linear interpolation instead of the original value in the coarse grid. As a result, stations that are physically close to each other are guaranteed to possess similar valued climate variables, which was not the case with coarse resolution. As shown in Figure 2.4 *left*, if two stations are next to each other but their locations are such that one is in a grid cell and the other in another grid cell, the reanalysis input for each station with be very different despite their physical closeness. As shown in the *middle* and *right* plots, such effect is mitigated by interpolation.

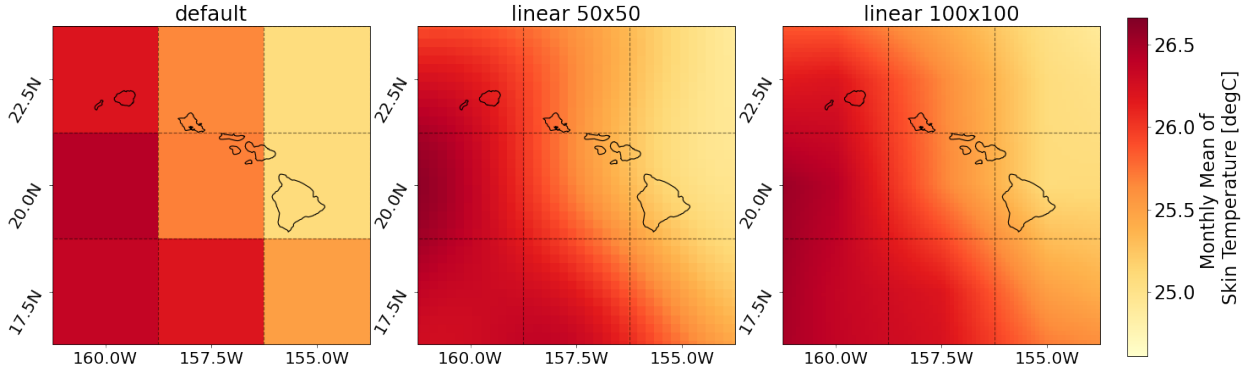

Figure 2.4: Monthly mean of skin temperature. *left*: original values as given in NCAR/NCEP reanalysis data. *middle*: 50 by 50 interpolation. *right*: 100 by 100 interpolation. By linearly interpolating the coarse input, input values to the models among stations are smoothed out.

CHAPTER 3
METHODS

The performance comparison is evaluated via four types of methods: linear regression, random forest and XGBoost, both of which are types of tree methods, and neural networks. Linear regression serves as the performance baseline. Evaluation is made on Root Mean Squared Error (RMSE) between ground truth rainfall observation vs. the prediction made by machine learning models.

3.1 Linear Regression

Linear regression is one of the simplest statistical models, where the predicted value is expressed as the sum of each input variable multiplied by its corresponding scalar values. Under the assumption that a quantity y_i depends on a set of d predictor variables, $\{x_i^1, x_i^2, \ldots, x_i^d\}$, the prediction $\hat{y_i}$ is expressed as

$$\hat{y_i} = \beta^0 + \beta^1 x_i^1 + \cdots + \beta^d x_i^d + \varepsilon$$

where ε follows a normal distribution with mean zero. Given a dataset of n samples $\{x_i^1, x_i^2, \ldots, x_i^d, y_i\}_{i=1}^n$ and corresponding predictions $\{\hat{y_i}\}_{i=1}^n$, the objective is to find β such that it minimizes the total sum of squares, i.e.,

$$\underset{\beta}{argmin} \left(\sum_{i=1}^{n} (y_i - \hat{y_i})^2 \right)$$

Because of its simplicity, linear regression serves as the baseline model.

3.2 Tree Methods

The basic component of tree methods is a decision tree. Starting from the root node, each sample is categorized by sequences of decisions, where each internal node is a binary decision and each edge of the tree represents agreement/disagreement on the decision. Therefore, each leaf node is a subset of the entire data. Random forest builds upon decision trees, where instead of a single tree making predictions, multiple trees are trained, each with a different subset of the training data [2]. The prediction is therefore ensemble from the trees, i.e., forest. XGBoost is also a variant of tree methods, where instead of taking ensembles of predictions, shallow trees with small number of depth, also called as 'weak learners', make sequential prediction, each updating the prediction made by previous weak learners [3]. In this paper, random forest and XGBoost are trained and tested for their performance.

3.3 Neural Networks

Neural networks are a set of computational units that execute sequential transformations of the input. A series of layers transform the input and pass onto the next layer, after all of which the prediction (output) is calculated. Each layer consists of a set of weights, which are used to transform the output from the previous layers. During training, those weight values are updated gradually. Neural networks are able to capture non-linear and complicated relationships between the input and the output. Though there exist numerous possible neural network architectures, this project focuses on simple feed-forward dense networks.

CHAPTER 4
RESULTS

4.1 Feature Selection

Our first set of experiments identify which input features are most useful for predicting rainfall. For each algorithm in linear regression, random forest, and XGBoost, a single model learns on the entire dataset from all of the stations. Latitude and the longitude coordinates are also included so that the models can identify which site each sample came from.

There are two types of experiments for feature selection: one that tests the effect of ablation of inputs, and the other that explores the variants of reanalysis data. In the ablation study, elevation and the seasonal indicator are tested for inclusion/exclusion. Seasonal indicator is a binary variable indicating whether the sample is observed in dry season (months from May to October) or wet season (November to April). This results in four variations of the input as described in Table 4.1: 1) *base*, which includes all of latitude and longitude coordinates, the seasonal indicator, and elevation. 2) *-both*, where the seasonal indicator and elevation are excluded from the *base*. 3) *-elevation*, where elevation is excluded from the *base*. 4) *-seasonal*, where the seasonal indicator is excluded from the *base*.

In the other experiment for exploring the variants of reanalysis input, three variants are tested as described in Table 4.1: linear interpolation with resolution of *50×50*, linear interpolation again but with resolution of *100×100*, and the *grid*. With the interpolation, all of the 16 reanalysis variables are linearly interpolated to replace the original coarse observations with the closest observations at higher resolutions. With the grid variant, the 16 reanalysis variables are given as a whole grid. The grid cells to be included are at (160°W, 22.5°N), (157.5°W, 22.5°N), (155°W, 22.5°N), (160°W, 20°N), (157.5°W, 20°N), (155°W, 20°N), which correspond to the blue dots in the Figure 2.2.

Hyperparameter search is conducted for each type of the input from both of the experiments, and for all of the methods. The dataset is split into train, validation, and test set chronologically. The training set consists of all of the data between 1948 and 1983, the validation set consists of years between 1984 and 1996, and the test set consists of years between 1997 and 2012. Boundary years are all inclusive. This results in 513,970 data in the training set, 172,508 data in the validation set, and 179,059 data in the test set, i.e., 59.3%, 19.9%, 20.7% split.

Hyperparameter tuning is done using random grid search in hyperparameter space to test 50 different combinations, using a python package for hyperparameter optimization, SHERPA [1]. The search space for XGBoost is the number of estimators in $\{100, 110, 120, ..., 300\}$, learning rate in $[0.05, 0.2]$, and max depth in $\{1, 2, ..., 10\}$. The search space for random forest is the number of estimators in $\{100, 110, 120, ..., 300\}$ and the minimum number of samples for a node to be split in $\{2, 3, ..., 6\}$. For every randomly chosen combination of hyperparameters, a model is trained on

the training set and performance on the validation set is used for choosing the best combination of hyperparameters. After the best hyperparameter is chosen, the final evaluation is made on the test set.

The following Table 4.1 categorizes the input type by attributes used. In order to capture non-deterministic results by randomized training for random forest and XGBoost, each model is trained five times for the evaluation to calculate the mean and the standard deviation of Root Mean Squared Error (RMSE).

Table 4.1: Categorization of the model input. Total of seven variations of the input were tested.

Ablation Study					
Input Type	lat lon	seasonal	elevation	reanalysis	resolution
base	✓	✓	✓	closest	original
-both	✓	×	×	closest	original
-elevation	✓	✓	×	closest	original
-seasonal	✓	×	✓	closest	original
Reanalysis Variant					
Input Type	lat lon	seasonal	elevation	reanalysis	resolution
50x50	✓	✓	✓	closest	50x50
100x100	✓	✓	✓	closest	100x100
grid	✓	✓	✓	grid	original

The follwoing table 4.2 summarizes the RMSE obtained by this experiment. The best performance of the ablation study was observed with XGBoost when using both elevation and the seasonal indicator. Neither resolution of interpolation helped improve the performance, suggesting the original resolution is enough for this task. The fact that the RMSE increased compared to *base* suggests that interpolating the original reanalysis data is not only useless, but it introduces artifacts that negatively affect performance.

Providing reanalysis data as a grid had no strong effect over the *base* input type, either. Meanwhile, *grid* input increased the computation time significantly. This suggests that single input of the coarse reanalysis resolution is enough.

Table 4.2: Result of the study on feature selection. Statistics are obtained from five runs. Linear regression is run only once, as the prediction is deterministic. Removing some of the input variables did not yield improvement over the *base* input type. The performance increased when feeding reanalysis data as *grid* but the difference is not significance, while it increased the computation time.

Ablation Study			
Input Type	LR	RF	XGB
base	5.301	3.931 ± 0.006	$\mathbf{3.784 \pm 0.012}$
-seasonal	5.315	3.963 ± 0.002	3.818 ± 0.008
-elevation	5.319	3.959 ± 0.008	3.813 ± 0.019
-both	5.333	3.994 ± 0.011	3.868 ± 0.015
Reanalysis Variant			
Input Type	LR	RF	XGB
50x50	5.556	4.103 ± 0.010	4.113 ± 0.023
100x100	5.556	4.075 ± 0.005	4.077 ± 0.054
grid	5.667	3.991 ± 0.004	$\mathbf{3.762 \pm 0.021}$

4.2 Site-specific experiments

In the previous experiments, a single machine learning model fit to all of the data from all the stations. In this study, multiple models fit to each station using only the input data for that station, hence making them site-specific. Theoretically, there are as many machine learning models as the number of stations, which significantly reduces the number of training data for each model. Moreover, each model has a different number of data available. In order to compensate for the reduced number of training samples and thus variance of the performance, chronological five-fold cross validation is used to evaluate the performance. For each station, the entire data is chronologically split into five folds. The prediction on the first fold is made by training on the rest of the four folds. The prediction on the second fold is also done in the same manner, and so are for all the rest of the folds. This means any stations with less than five samples have to be excluded, which resulted in 1944 unique weather stations. Regarding the input features, since each model only receives data from a single station, lat lon coordinates and elevation become irrelevant. Therefore, XGBoost models are given 16 reanalysis variables in the original coarse resolution, along with the seasonal indicator. Comparison is made on linear regression and XGBoost.

The following figure 4.1 shows the result of such an experiment. When aggregated, the mean of RMSE with linear regression is 4.309, and 4.150 with XGBoost, so XGBoost outperforms linear regression. However, this is merely due to the fact that linear regression is particularly bad with a low number of samples. If we focus on the case where a station has more than 100 samples, linear regression outperforms XGBoost in the majority of stations. The issue with this approach is that

the XGBoost models are using the same hyperparameters across all the site-specific models. In the next experiment, we narrow down the number of stations so that we can choose appropriate hyperparameters for each site.

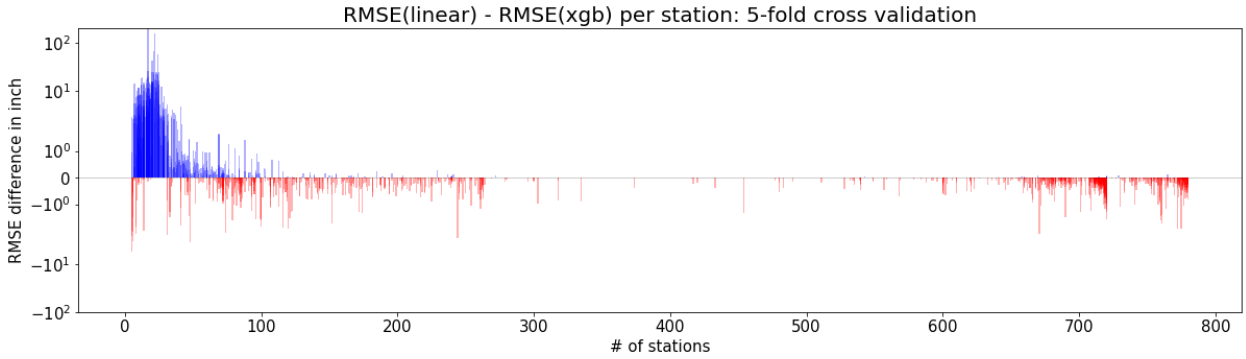

Figure 4.1: The comparison of performance between XGBoost and LR per station. Each bar corresponds to a station, where the positive height (blue) indicates XGBoost outperforms LR, and negative height (red) in the opposite case. XGBoost outperforms LR for the stations with low number of samples, but that is not the case with a higher number of samples.

4.3 The effect of the sample size

As shown in the previous section, it is challenging to beat the performance of linear regression when it comes to site-specific models. In this study, we focus only on actual rainfall data, by excluding gap-filled data. This is to remove any possibility that the gap-filling method is inducing artifacts such that it favors the performance of downstream prediction with linear regression. The study focuses on the effect of sample sizes for site-specific models. Each model takes in samples from a single station, while the number of training data is different, i.e., in $\{50, 100, 150, ..., 500, 550\}$. To achieve this, gap-filled data is removed from the dataset first. Next, any stations with less than 750 samples are excluded for insufficient training data. This resulted in 24 unique stations. In order to achieve the best performance, hyperparameter tuning was done using 80% of the data for five-fold cross validation. The rest of 20% was set aside for the final performance evaluation. This split was made chronologically.

The search space for the hyperparameter tuning for XGBoost is the number of estimators in $\{100, 110, 120, ..., 300\}$, learning rate in $[0.001, 0.1]$, and the max depth in $\{1, 2, 3, ..., 10\}$. Randomly chosen 500 combinations of the hyperparameters were tested. For each of the random combinations, all of the 24 site-specific models are trained independently to evaluate the individual model's performance. After every combination is tested, each station obtains (possibly, but not necessarily) a different set of the best hyperparameters.

Since XGBoost had difficulty making comparable predictions to linear regression models, an-

other non-linear method, neural networks were introduced. This model consisted of three dense layers with selu activation, and the input and the output layers. Between each dense layer are dropout layers with the dropping rate of 0.5. L2 regularization is applied to each of the dense layers. Unlike linear regression and tree methods, input data has to be pre-processed for the best performance. All of the input values are scaled into values in $[0, 1]$, and the output was scaled into log space. Training was done using 20% of the training data for early stopping. The hyperparameter search was done for the number of units in each layer in $\{256, 257, ..., 1024\}$, learning rate in $[1 \times 10^{-5}, 1 \times 10^{-2}]$, and batch size in $\{64, 128, 192, 256, 512\}$. Randomly chosen 160 combinations of the hyperparameters were tested with five-fold cross validation on the 80% of the entire data. Again, 24 of site-specific models were independently trained to achieve site-specific hyperparameters.

After the best hyperparameters were obtained, all of the five folds used for hyperparameter tuning were now used as the training set. 24 independent models train on each site, and mean and standard deviation of ten runs was calculated on the held-out 20% of the data. The following figure 4.2 shows the aggregated result of all 24 models. In general, it is shown that more samples lead to better performance in any method. Neural networks are able to achieve lower RMSE than linear regression if provided more than 150 samples.

However, if we focus on the comparison on a station basis as shown in figure 4.3, we can observe that neural networks are still outperformed by linear regression in some stations.

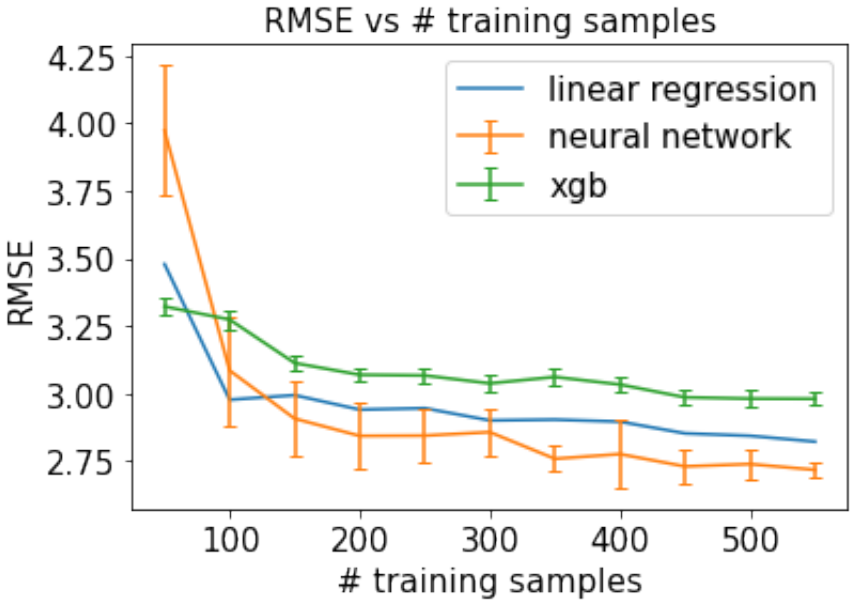

Figure 4.2: RMSE of the prediction with different numbers of training samples.

This suggests that it is still challenging to outperform linear regression on all the stations. If the task is to predict the overall rainfall amount as the entirety, neural networks are able to

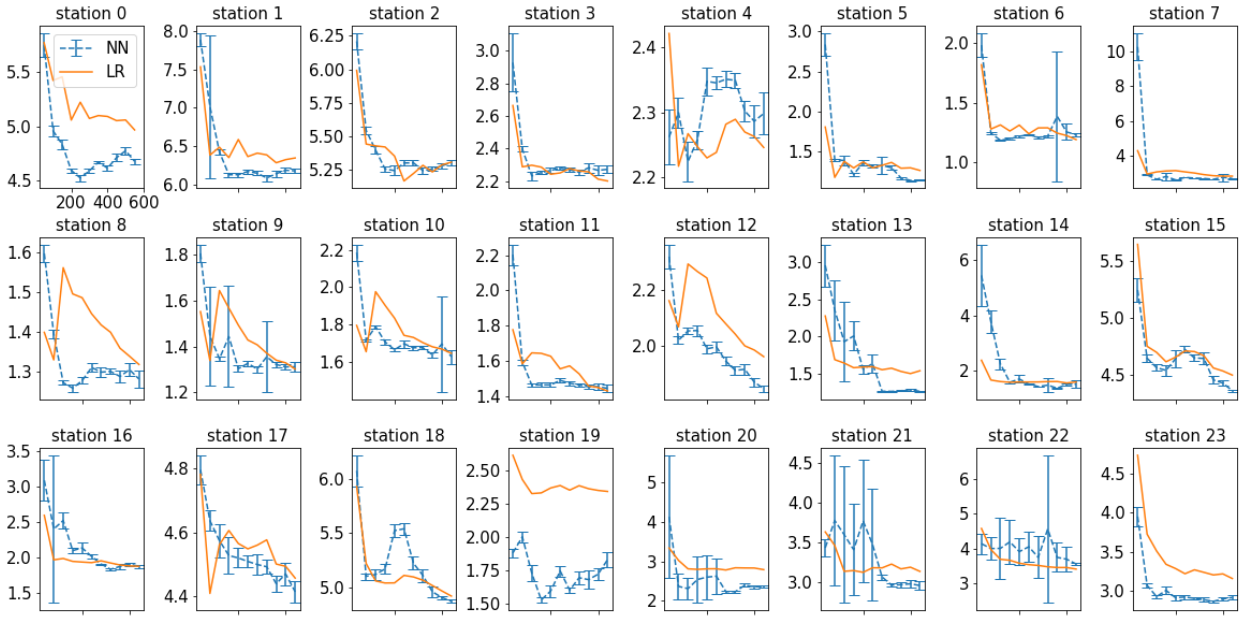

Figure 4.3: The comparison of performance between NN and LR per station. x-axis is the number of training samples and the y-axis is RMSE. At some stations, LR performs better but on average, NN performs better.

achieve better performance compared to linear regression. However, if the task is to achieve better prediction on a station basis, then linear regression is still effective in some cases.

Overall, the number of samples is a significant factor that affects the performance of the models. The observed trend is that the more training samples lead to better performance. However, because of the coarse temporal resolution of rainfall data (monthly observations) this encounters limitations on collecting more samples.

CHAPTER 5
DISCUSSION AND CONCLUSIONS

A series of experiments tested the performance of non-linear machine learning methods on climate data. Neural networks outperformed linear regression on this task, especially when provided with more sample sizes. We quantified an improvement of performance as the amount of historical data increased.

One of the challenges is to capture correlation between stations. In this study, all of the site-specific models were trained independently. However, correlation of prediction uncertainty is of crucial interest in some applications such as disaster control and state-wise rainfall mass prediction for water resource management. More research into spatial regression models that can capture uncertainty is needed. Our preliminary results on site-specific heteroskedastic regression models indicate capturing uncertainty is possible, but because we did not conduct a complete and thorough experiment on that, the result is not included in this paper.

Nonetheless, this project provides insight on the capability and limitation of non-linear methods, which encourages the future research on statistical downscaling.

APPENDIX A
HYPERPARAMETER SEARCH

The following tables show the space of hyperparameters search, and the best combination of hyperparameters chosen using cross validation.

A.1 Hyperparameter search for feature selection

Table A.1: Hyperparameter search for XGBoost

XGBoost		
#estimators	**learning rate**	**max depth**
{100,110,120,...,300}	[0.05, 0.2]	{1,2,...,10}

Table A.2: Best hyperparameters for XGBoost per each input type

XGBoost			
input type	**#estimators**	**learning rate**($\times 10^{-2}$)	**max depth**
base	290	11.6	9
-both	300	9.15	8
-elevation	240	5.52	9
-seasonal	170	10.6	8
50x50	190	6.34	9
100x100	240	8.52	8
grid	240	13.0	8

Table A.3: Hyperparameter search for random forest

Random Forest	
#estimators	**min # samples**
{100,110,120,...,300}	{2,3,...,6}

Table A.4: Best hyperparameters for random forest per each input type

Random Forest		
input type	**#estimators**	**min # samples**
base	180	4
-both	270	3
-elevation	250	2
-seasonal	100	5
50x50	270	3
100x100	200	2
grid	220	2

A.2 Hyperparameter search for assessing the effect of the sample size

Table A.5: Hyperparameter search for XGBoost

XGBoost		
#estimators	**learning rate**	**max depth**
{100,110,120,...,300}	[0.001, 0.1]	{1,2,...,10}

Table A.6: Best hyperparameters for XGBoost per each site

XGBoost			
station id	**#estimators**	**learning rate**($\times 10^{-2}$)	**max depth**
0	190	2.785	3
1	190	9.494	2
2	280	6.212	2
3	160	4.773	1
4	110	7.366	1
5	170	7.953	1
6	110	5.225	1
7	100	2.856	4
8	260	6.785	1
9	300	8.400	1
10	260	8.308	1
11	240	9.677	1
12	130	8.458	2
13	130	3.406	2
14	170	3.603	1
15	110	4.833	4
16	130	3.396	2
17	220	7.775	2
18	110	3.337	4
19	130	5.728	2
20	170	9.065	1
21	190	2.891	3
22	280	9.263	4
23	180	7.471	2

Table A.7: Hyperparameter search for neural networks

Neural Networks		
#units	**learning rate**	**batch site**
{256,257,...,1024}	$[1 \times 10^{-5}, 1 \times 10^{-2}]$	{64,128,192,256,512}

Table A.8: Best hyperparameters for neural networks per each site

neural networks			
station id	**#units**	**learning rate**($\times 10^{-3}$)	**batch size**
0	274	1.702	192
1	388	2.570	192
2	315	1.835	512
3	558	1.129	512
4	558	1.129	512
5	558	1.129	512
6	558	1.129	512
7	558	1.129	512
8	795	0.793	256
9	961	0.749	256
10	558	1.129	512
11	494	0.451	128
12	558	1.129	512
13	961	0.749	256
14	558	1.129	512
15	795	0.793	256
16	961	0.749	256
17	687	1.343	64
18	719	1.185	256
19	559	0.489	256
20	987	0.891	256
21	987	0.891	256
22	987	0.891	256
23	961	0.749	256

APPENDIX B
CLIMATE VARIABLES FROM REANALYSIS DATA

The following table provides the complete list of attributes of reanalysis data used for the experiments and the atmospheric trait they explain.

Table B.1: List of Attributes of Reanalysis Data

Attribute	**Atmospheric Trait**
Geopotential Height at 500hPa	Air pressure
Geopotential Height at 1000hPa	Air pressure
Air temperature difference 1000hPa minus 500hPa	Air temperature difference: atmospheric stability
Surface air temperature at 2m	Air temperature
Zonal moisture transport at 700hPa	Flux of moisture
Zonal moisture transport at 925hPa	Flux of moisture
Meridional moisture transport at 700hPa	Flux of moisture
Meridional moisture transport at 925hPa	Flux of moisture
Omega	Vertical velocity of the atmosphere
Specific humidity at 700hPa	Specific humidity
Specific humidity at 925hPa	Specific humidity
Precipitable water	Atmospheric moisture
Potential temperature difference between 850hPa and 1000hPa	Air temperature: atmospheric stability
Potential temperature difference between 500hPa and 1000hPa	Air temperature: atmospheric stability
Sea level pressure	Air movement: rising/sinking
Skin temperature	Surface temperature

BIBLIOGRAPHY

[1] Baldi, P., Collado, J., Hertel, L., Ott, J., and Sadowski, P. Sherpa: Robust hyperparameter optimization for machine lerning. arxiv:2005.04048v1[cs.LG], 2020.

[2] Brieman, L. Random forests. *Machine Learning*, 45:5–32, October 2001.

[3] Chen, T., and C. Guestin. Xgboost: A scalable tree boosting system. arxiv:1603.02754v3[cs.LG], 2016.

[4] Frazier, A. G., Giambelluca, T. W., Diaz, H. F. and Needham, H. L. Comparison of geostatistical approaches to spatially interpolate month-year rainfall for the Hawaiian Islands. *Int. J. Climatol.*, 36(3):1459–1470, 2016.

[5] Hawai'i Groundwater & Geothermal Resources Center web site. `<https://www.higp.hawaii.edu/hggrc/projects/geothermal-digital-collection/geothermal-collections/geothermal-topic-guides/water-quality-and-wells-hydrology/>`.

[6] Kalnay, E., M. Kanamitsu, R. Kistler, W. Collins, D. Deavan, L. Gandin, M. Iredell, S. Saha, G. White, J. Woollen, Y. Zhu, M. Chelliah, W. Ebisuzaki, W. Higgins, J. Janowiak, K. C. Mo, C. Ropelewski, J. Wang, A. Leetmaa, R. Reynolds, R. Jenne, and D. Joseph. The NCEP/NCAR 40-Year Reanalysis Project. *Bulletin of the American Meteorological Society*, 77(3):437–472, March 1996.

[7] Sanderson, M. *Prevailing trade winds: Weather and Climate in Hawai'i.* University of Hawai'i Press, Honolulu, Hawai'i, 1993.