Forecasting Final Presentation

By Gordon Li, Jaimie Obatake,
Brieanna Sundberg, and Austin Tasato

Overview:

- Project Background and Motivation
- Objectives
- Block Diagram
- Algorithms
- Results
- Problems and Solutions
- Final Status
- Future Improvements

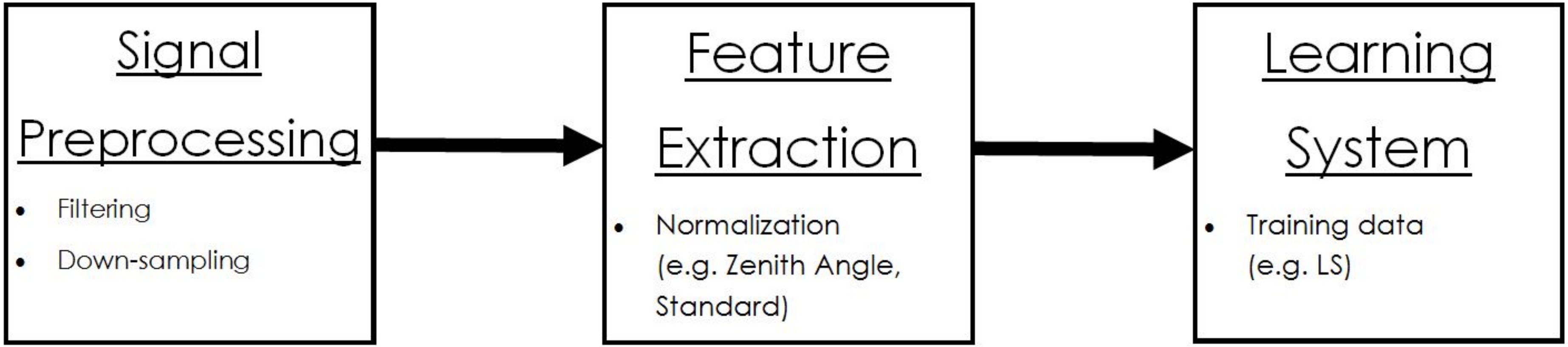
Project Background & Motivation:

- Want to integrate photovoltaic (PV) into the UHM campus
  - Determine optimal location of PV placement
- Difficulties of making this change?
  - Intermittency
  - Match power generation and consumption
    - Black and brown outs

Objectives:

- Visualize and analyze annual solar irradiance data with 3-D Plots
- Predict solar irradiance with offline algorithm
- Compare and visualize normalization methods with zenith angle and standard normalization
- Compare and visualize taps of varying data sets
- Compare and visualize error of varying data sets

Block Diagram:



Preprocessing: Error Detection and Correction

`check_samples()`

- Checking for continuous time samples & filling in missing data
- **“Why is 10:00AM and 10:01 is missing?”**
- Corrects by adding in missing samples, and deleting duplicates

`check_errors()`

- Checking & replacing erroneous solar irradiance data
- **“Why is there a spike that’s 10x the surrounding data at 9:59AM?”**
- Replaces irradiance samples with previous value

`check_90()`

- Checking & replacing zenith angles greater than 87° with 87°
- **“Why is there a spike at this point while the surface around is relatively flat?”**
- Prevents normalization from causing spikes in the data due to the division of a very small number

cos(90°) ≈ 0

Preprocessing: Data Selection and Averaging

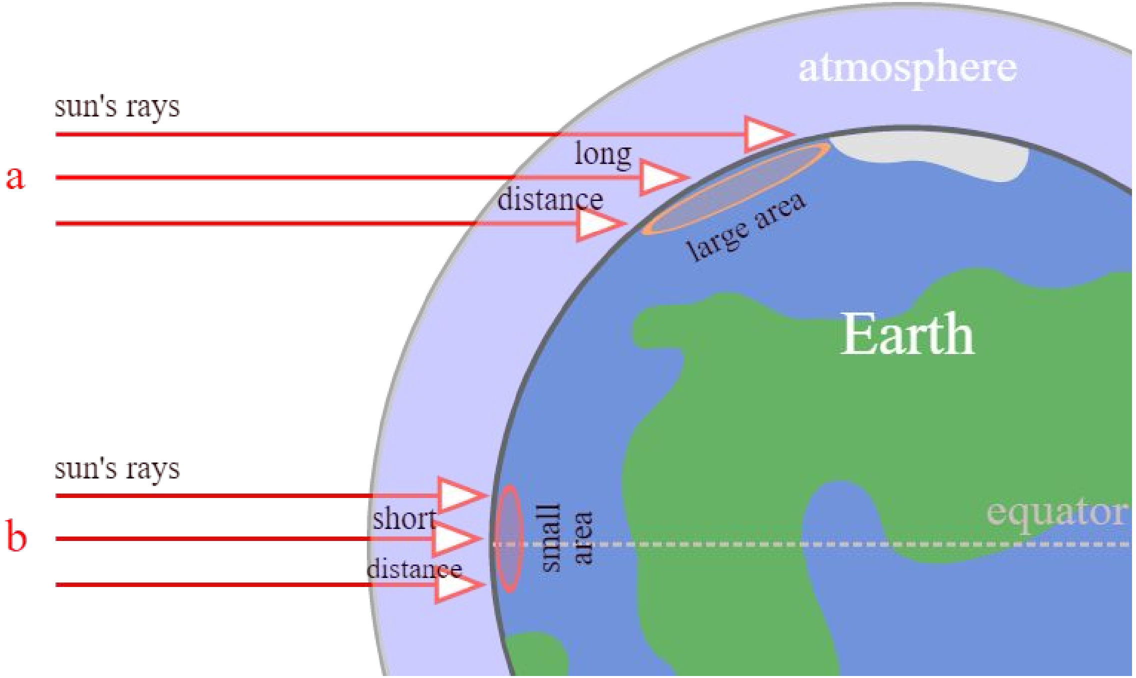
- Choose interval of day for solar irradiance
  - Select times to include relevant hours
  - Hours before sunrise and after sunset do not need to be predicted
- "Simplifying" data to a succession of averages
  - Take 5 minutes and average the solar irradiance of each minute
  - Helps to get smoother solar irradiance data

Normalization with Zenith Angle

- This eliminates the effects of:
  - Time of day
  - Seasons
  - Geographical location
- Projecting solar irradiance an hour ahead using future zenith angle an hour ahead

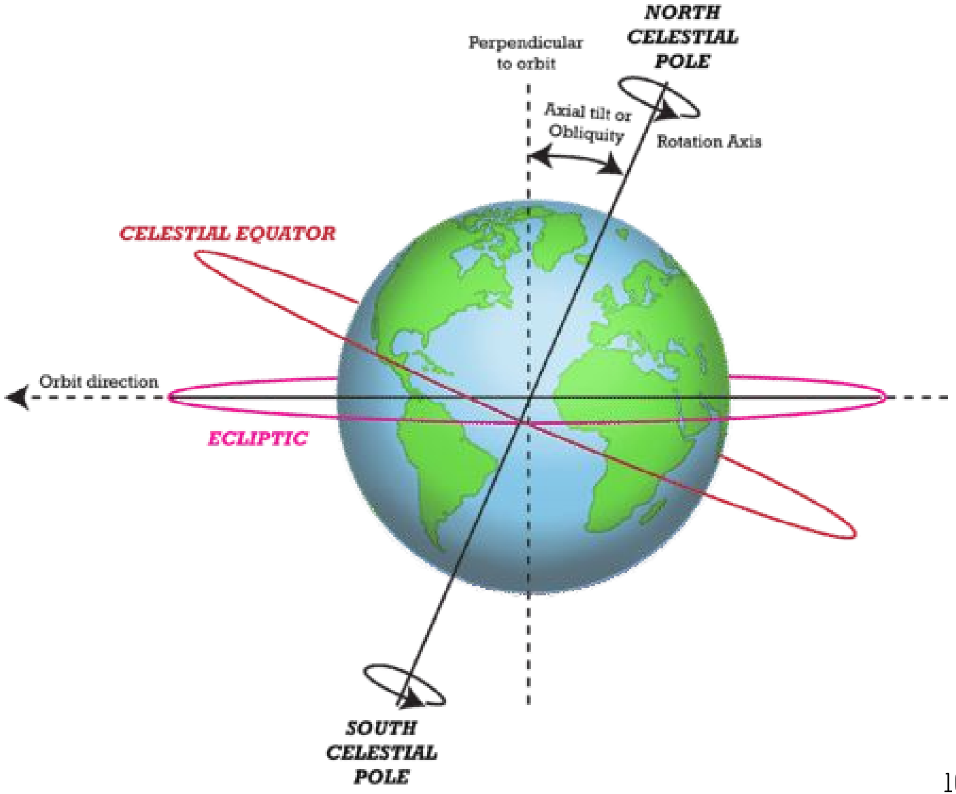
Zenith Angle: Geographical Location

- Distance solar irradiation travels through the atmosphere affects amount of irradiance that reaches the earth due to scattering, reflection, and absorption.
- Changes in latitude affect the surface area the irradiance is projected on.



Zenith Angle: Time of Day and Seasonal Effects

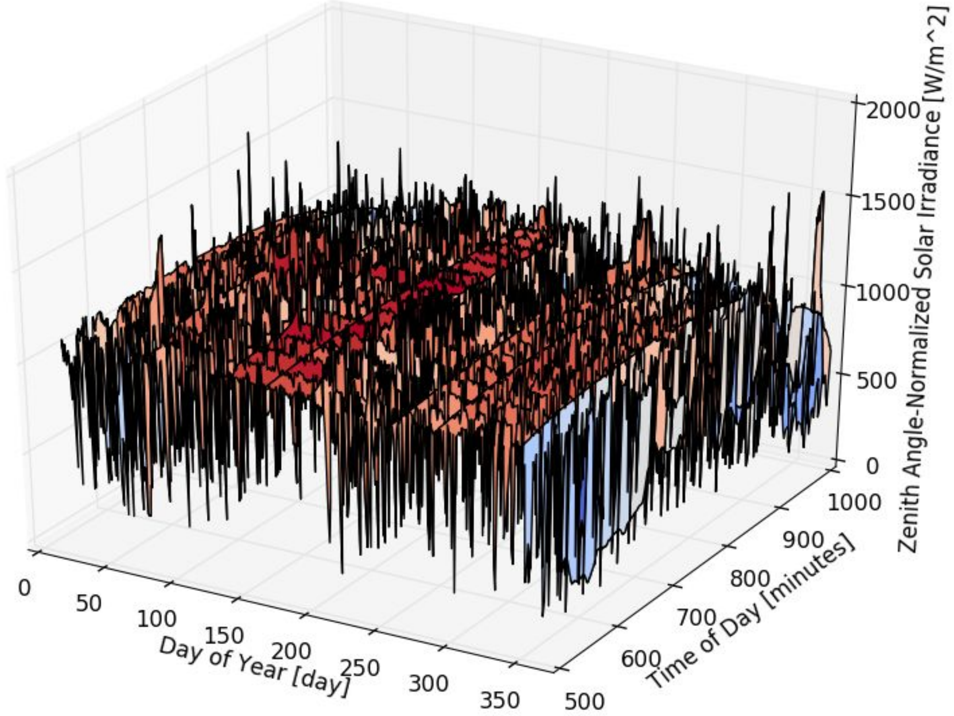
- Tilt of earth affects how much sunlight is in each season, and affects the time at which the sun rises and sets
- Solar ray’s angle of incidence changes throughout the day



Normalization with Zenith Angle Normalization

- Zenith angle normalization:

$$\frac{X}{cos(\Theta)}$$



Normalization with Standard Normalization

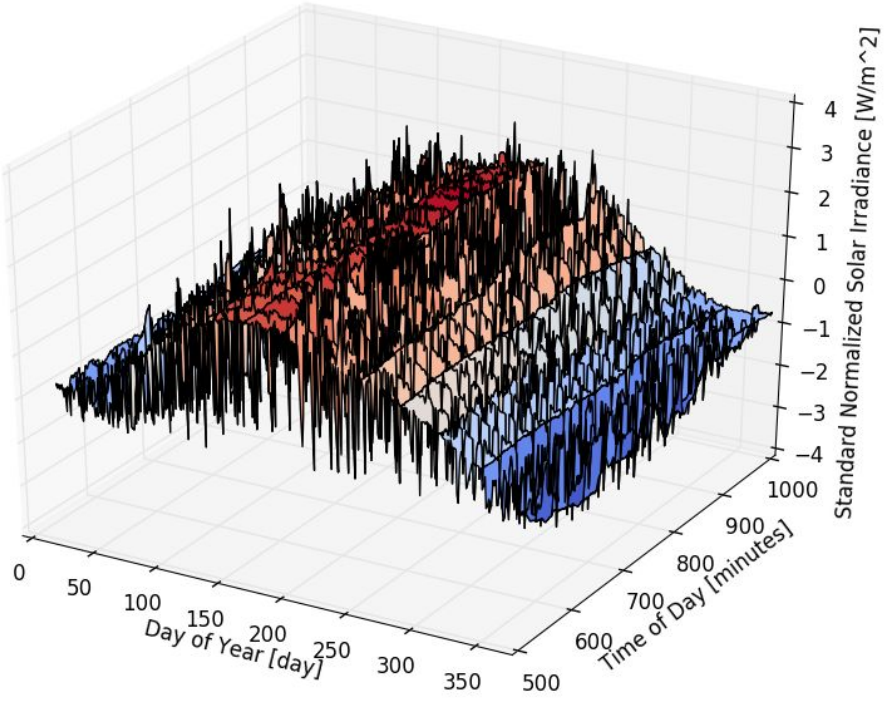
- Standard normalization:

$$\frac{X-\mu}{\sigma}$$

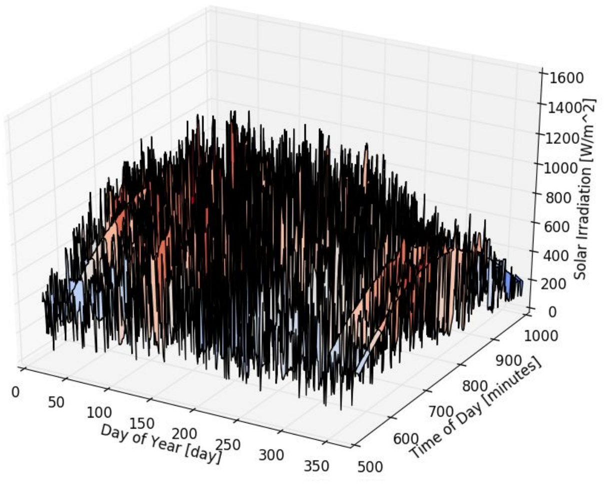
where X is the input, μ is the mean, and σ is the standard deviation

- Because it doesn't take into account the effects of season...

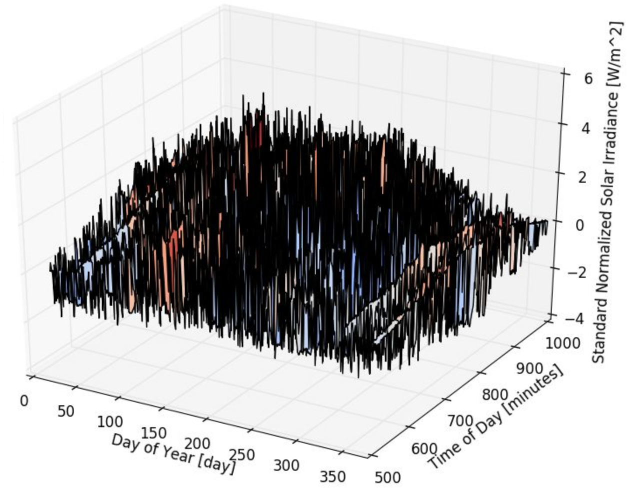
  It's not flat (against day of year)!



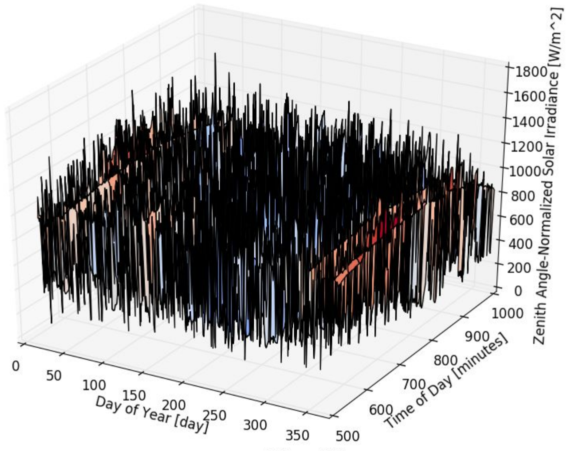
3D Plots : La Ola, Lanai



Not Normalized

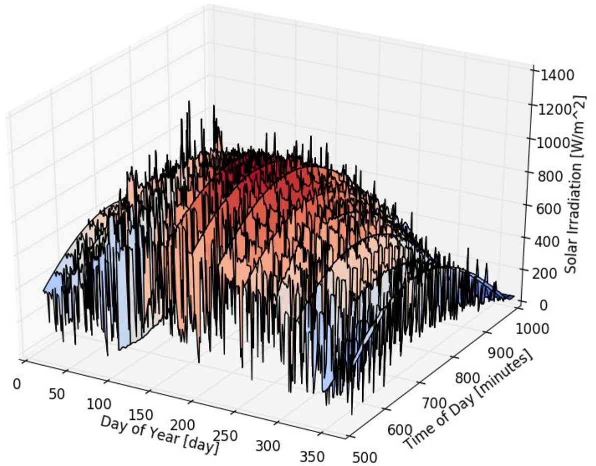


Standard Normalized

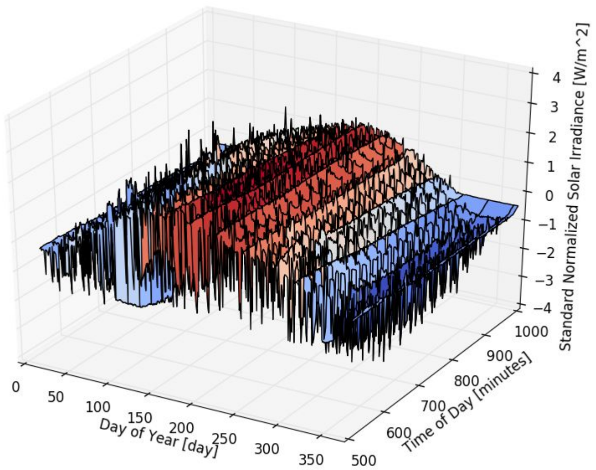


Zenith Normalized

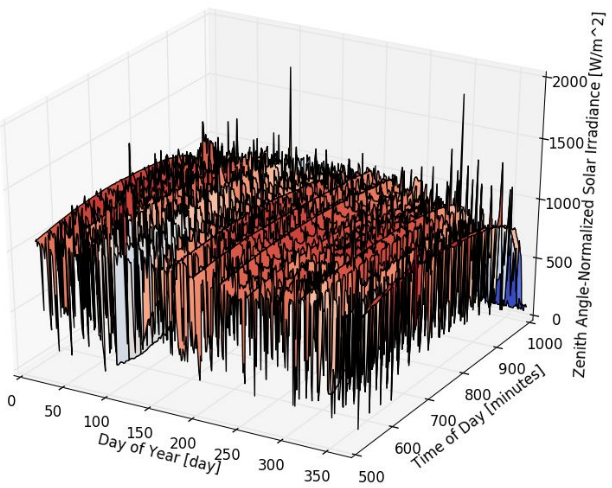
3D Plots : Los Angeles, California



Not Normalized

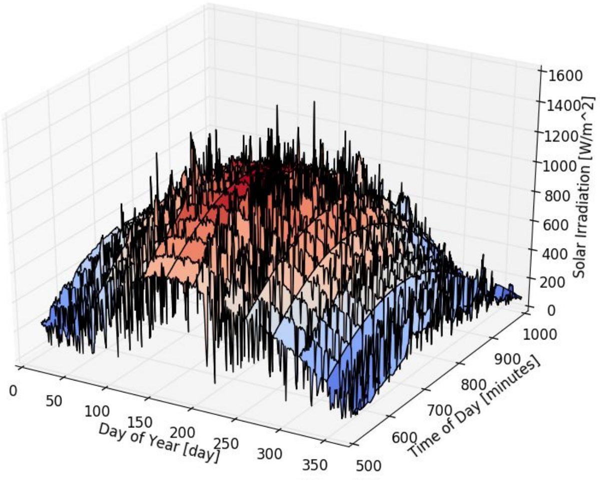


Standard Normalized

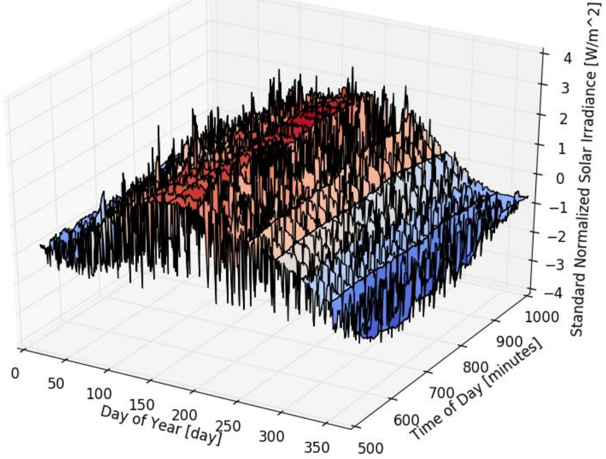


Zenith Normalized

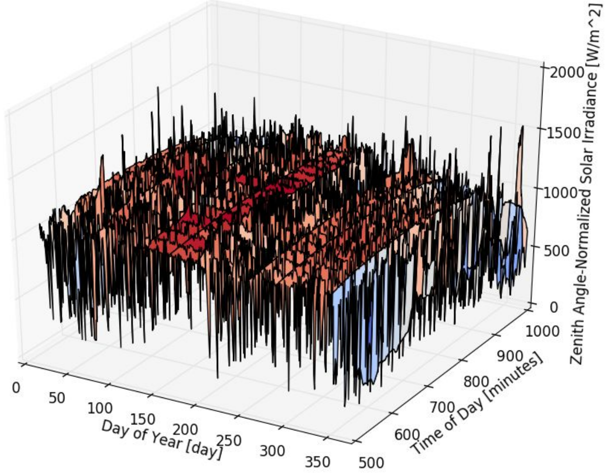
3D Plots : Milford, Utah



Not Normalized



Standard Normalized



Zenith Normalized

Least Squares Algorithm

- Batch algorithm
- 5 minute decimation
- Zenith angle normalization or standard normalization
- Create X-matrices (taps)
- Create W-vector (weights)
- Calculate **R**oot **M**ean **S**quare **E**rror (RMSE)
- Visualize results

Least Squares Algorithm

- Weights vector (W)
  - Determine weight by minimizing error

$$(XX^T)^{-1}XD$$

    - Error is difference in the desired (D) and predicted output (Y)

$$E = D - Y$$

- Root Mean Squared Error (RMSE)

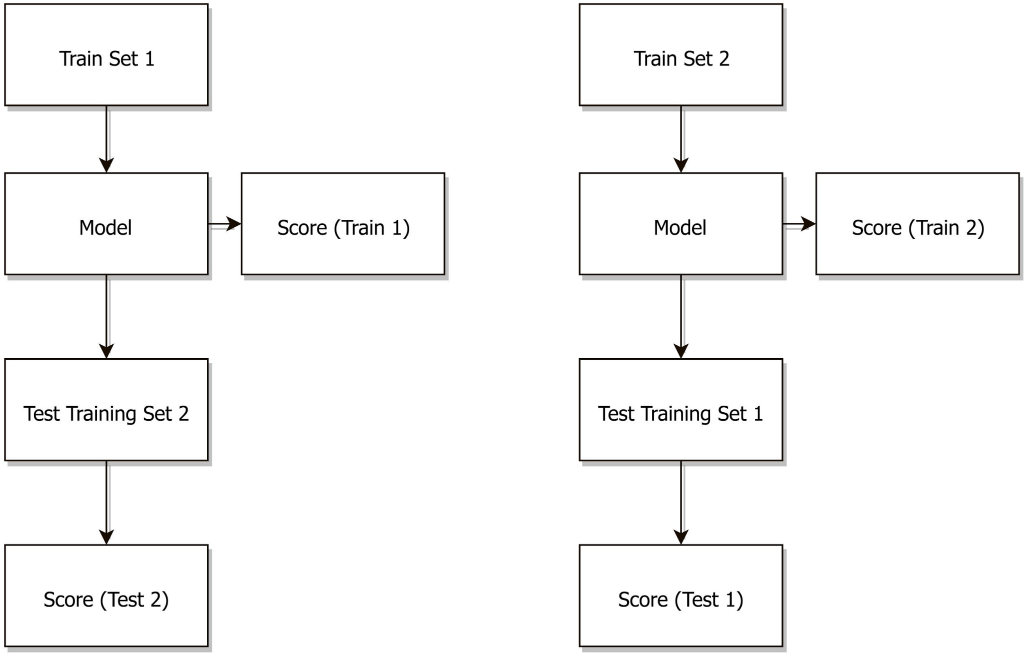
$$\sqrt{\frac{1}{m}\sum(D-Y)^2}$$

Tap Filters

- Using 1-10 taps
- Construct X-matrix with past data
  - Shifting “time window” 5-50 minutes into the past
- More taps = less error

TAP #																			
1	7:30	7:35	7:40	7:45	7:50	7:55	8:00	8:05	8:10	8:15	8:20	8:25	8:30						
2		7:30	7:35	7:40	7:45	7:50	7:55	8:00	8:05	8:10	8:15	8:20	8:25	8:30					
3			7:30	7:35	7:40	7:45	7:50	7:55	8:00	8:05	8:10	8:15	8:20	8:25	8:30				
4				7:30	7:35	7:40	7:45	7:50	7:55	8:00	8:05	8:10	8:15	8:20	8:25	8:30			
5					7:30	7:35	7:40	7:45	7:50	7:55	8:00	8:05	8:10	8:15	8:20	8:25	8:30		
6						7:30	7:35	7:40	7:45	7:50	7:55	8:00	8:05	8:10	8:15	8:20	8:25	8:30	
7							7:30	7:35	7:40	7:45	7:50	7:55	8:00	8:05	8:10	8:15	8:20	8:25	8:30

Results: Train and Test Error

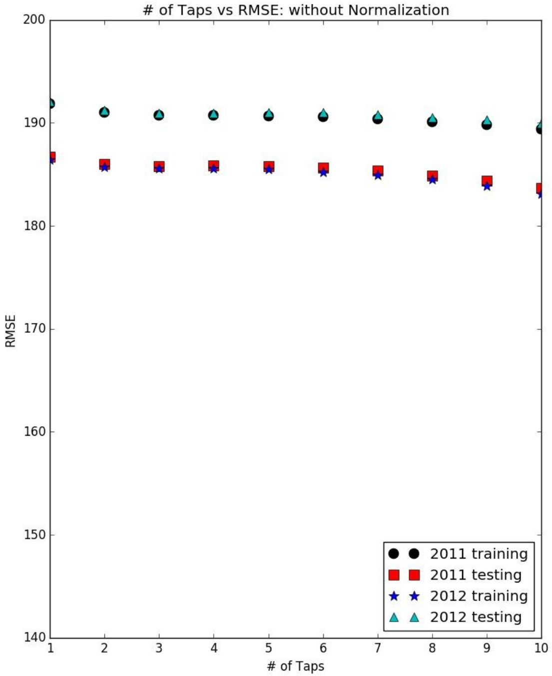


Train and Test for 2011 and 2012 Models

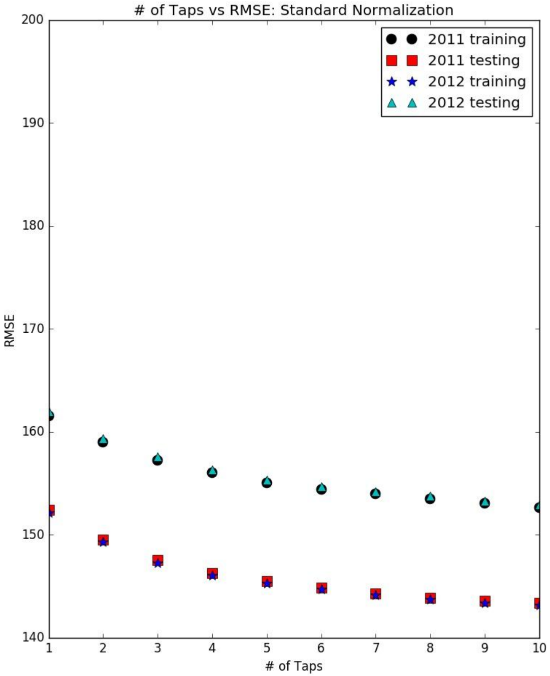
Average Train and Test Score to check correctness of model

$$\frac{Train\ 1 + Train\ 2}{2} \geq \frac{Test\ 1 + Test\ 2}{2}$$

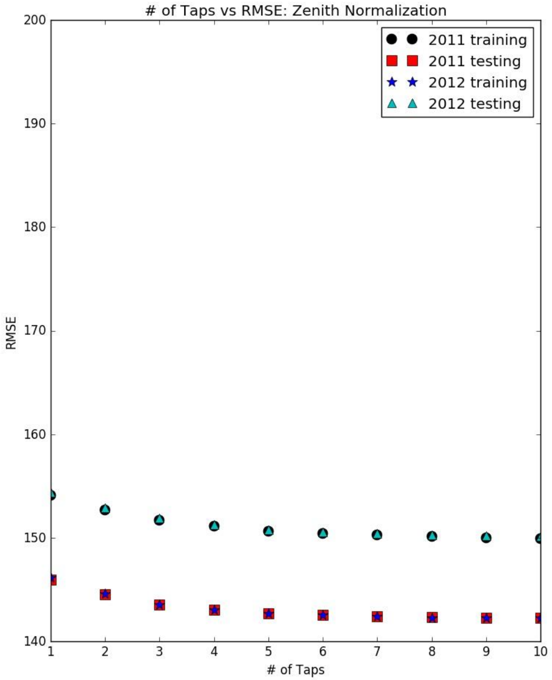
Milford, Utah Results: Taps vs RMSE Graphs



Without Normalization

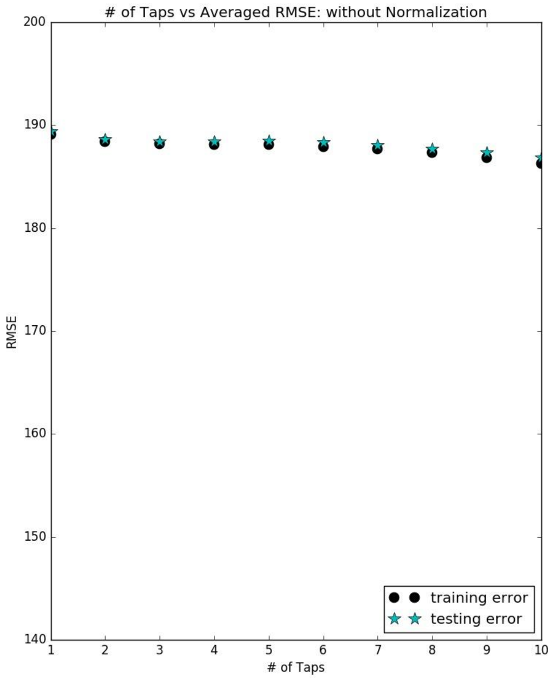


Standard Normalization

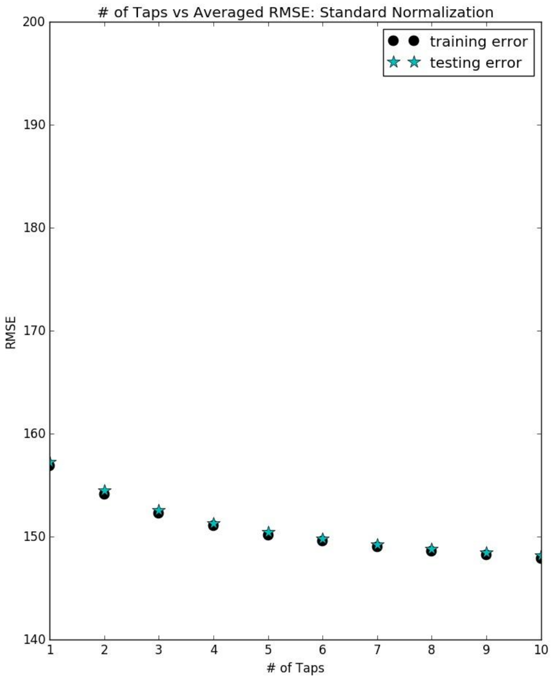


Zenith Angle Normalization

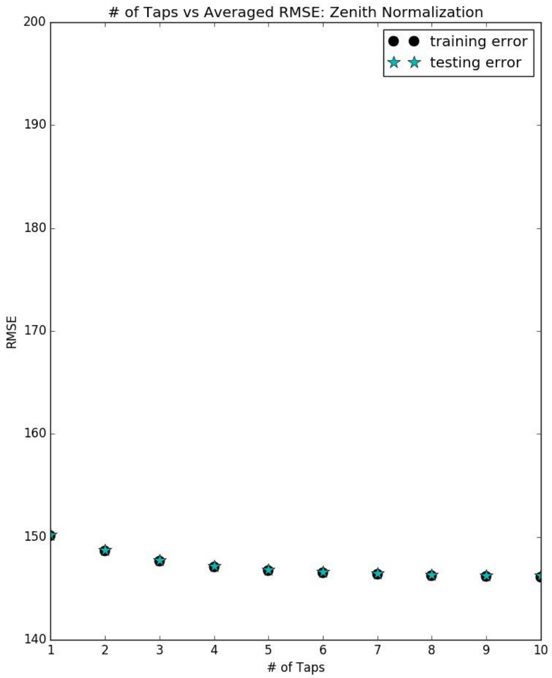
Milford, Utah Results: Averaged Taps vs RMSE Graphs



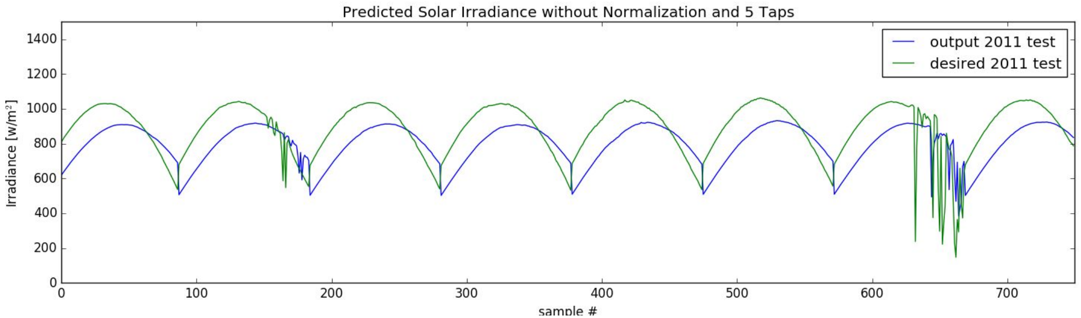
Without Normalization

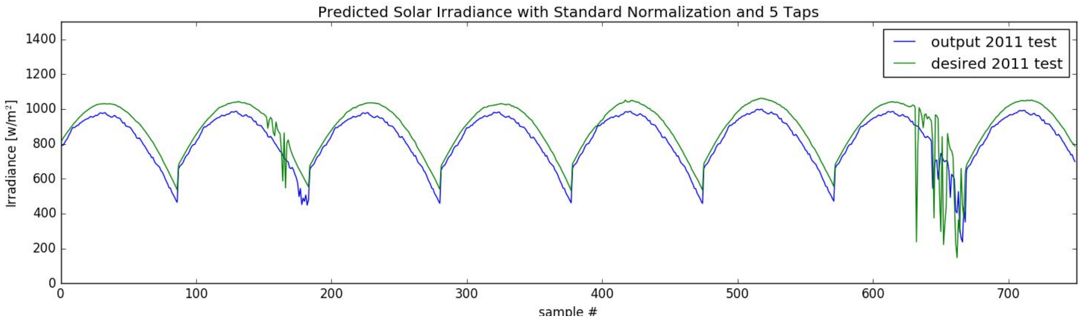


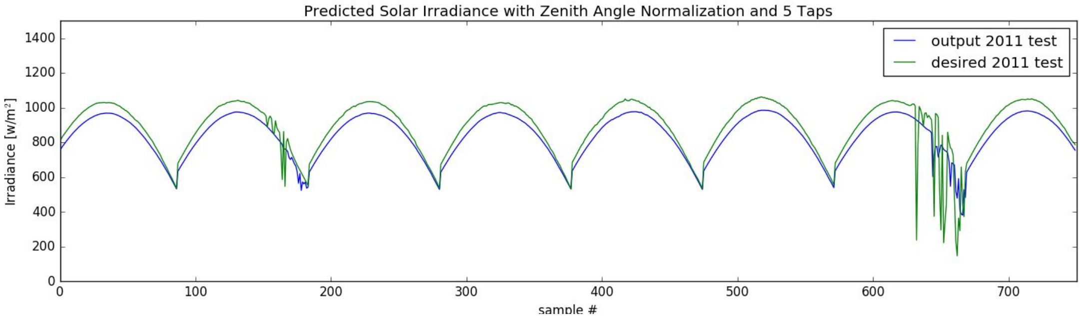
Standard Normalization



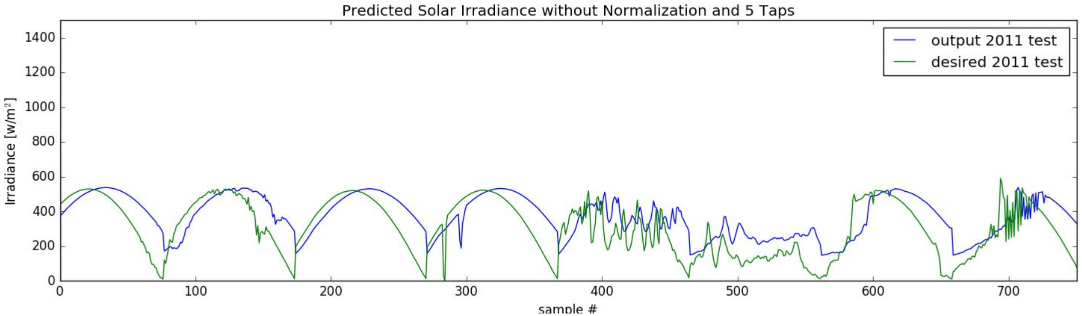
Zenith Angle Normalization

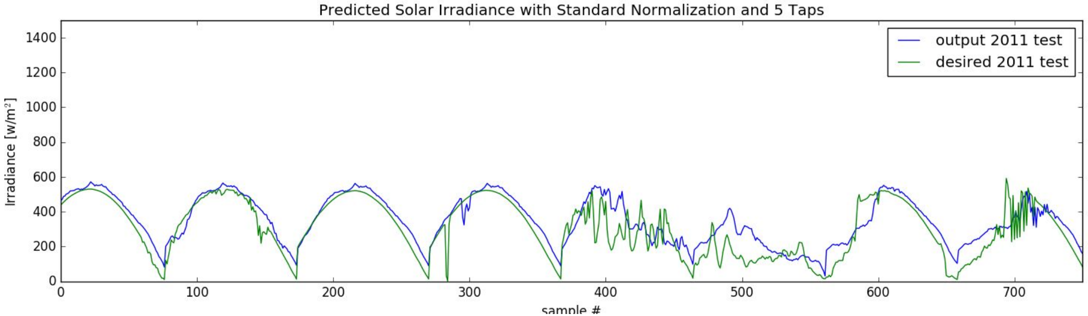


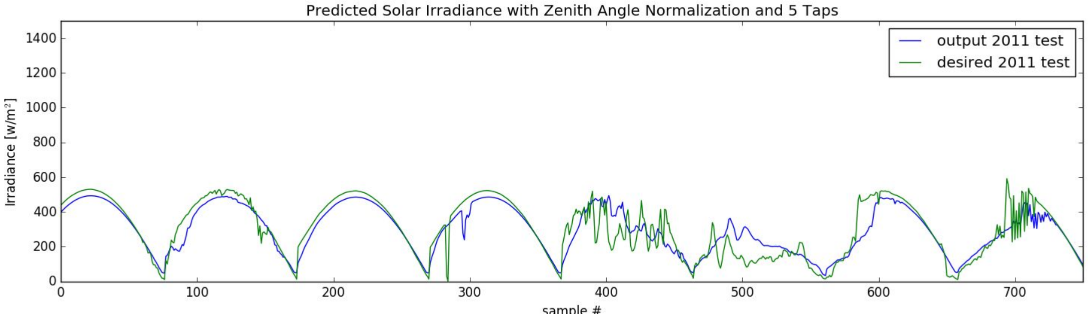




Milford, Utah Results: Predicted vs Actual Sunny Days







Milford, Utah Results: Predicted vs Actual Cloudy Days

Problems and Solutions:

- Zenith Angle Error
  - Solution: Implemented an error function to correct angles near 90°
- Continuity of Data Points
  - Solution: Implemented an error function to insert missing data points and removing duplicate data points
- Erroneous Data Points
  - Solution: Implemented an error function to correct erroneous data points with previous data points
- Incorrect RMSE Values
  - Solution: Incorrect usage of `numpy.flatten()` and `numpy.reshape()`

Final Status

- Implemented 3-D plots to visualize annual solar irradiance
- Developed error correction functions for missing and erroneous values of solar irradiance and zenith angle
- Learned about tap filters, zenith angle and standard normalization, and k-folds test-train framework
- Developed offline forecasting algorithm using linear regression

Future Improvements

- Predict power produced by solar irradiance
- Implement online algorithm
- Work with data collected from our weather boxes
- Work on documentation

Questions?