PROBABILISTIC MODELS FOR ONE-DAY AHEAD SOLAR IRRADIANCE FORECASTING IN RENEWABLE ENERGY APPLICATIONS ON OAHU

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 SCIENCE

MAY 2016

By

Carlos Vincius Andrade Silva

Thesis Committee:

Lipyeow Lim, Chairperson
Duane Stevens
Rick Kazman

Keywords: solar forecasting, probability models, hawaii, 1-Day Ahead Forecast

Success is going from failure to failure without losing enthusiasm.

Winston Churchill

ACKNOWLEDGMENTS

There are several people who without their continuous support, this thesis would not be made possible. I am thankful to Dr. Rick Kazman, who brought me from Brazil to the University of Hawaii at Manoa as a Masters student, and provided me with a stable source of funding throughout my studies. Dr. Kazman, whom I met still as an undergraduate as an exchange student in United States has been to me a mentor, a friend and a father since I left home. I am also in debt to Dr. Lipyeow Lim and Dr. Duane Stevens, who through over an year and a half spent many hours worth of inspiring discussion. I also appreciate the patience and availability of Dr. Lipyeow Lim throughout the whole work when I needed his advice outside the usual meeting time. I am thankful to Dr. Edoardo Biagioni, which as a graduate chair offered me many times room to ask for advice, some of which even on holidays.

I am thankful to my now fiancee Kathryn L. Paradis, who spent several nights hearing about solar forecasting to help me clarify unclear parts of my work, and for being always there for me. I am thankful to my family in Brazil, who constantly reached me to offer any support I may need, and the family whom I live with Hawaii, Lee and Alton Slater, an 82 year young lady and 79 athletic great grandpa, who I had the pleasure to share a house I could call my second home.

ABSTRACT

In order to produce energy, the Hawaiian Islands rely heavily on oil and oil products to fuel their power plants, leading to high electricity costs that help make renewable energy economically competitive, such as solar energy. Solar energy production, however, introduces a new dimension of uncertainty to meet energy load with supply due to climate conditions: We are not guaranteed to have sufficient solar irradiance available the next day to generate the necessary amount of energy for households and businesses. Forecasting 1-day ahead solar forecasting would then be helpful to know how much energy from other sources are necessary to be produced, in order to compensate the lack of solar energy for the following day.

To address the solar irradiance forecasting need, in this thesis we investigate probabilistic models for one-day ahead solar irradiance forecasting. Namely, we investigate how the use of past solar irradiance and other weather variables (e.g. relative humidity, pressure, temperature, etc.) using one or more sites can influence the accuracy of 1-day ahead solar forecasts. We also discuss how different parameters and limitations encountered throughout the usage of our probability models for solar forecasting influence the forecasts. To address the limitations, we present an entropy based probability model.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Context and Motivation

In order to produce energy, the Hawaiian Islands rely heavily on oil and oil products to fuel their power plants. On Oahu island, large, centrally located coal- and oil- fire power plants often run 24 hours a day to provide through electric grids a steady source of power to households and businesses throughout the day [1].

Electric grids are branching networks, with the main branches being feeder lines, which feed electricity to smaller distribution lines that run house-to-house and business-to-business. Electric grids also do not store energy, and **energy production (*supply*) and consumption (*load*) differences should be kept to a minimum to minimize costs for electric companies**.

The Hawaiian islands present unique challenges for minimizing costs of energy production. For instance, when compared to mainland power grids, island power grids are self-contained and isolated, so they have no neighboring grids to turn for support, for example, to purchase or sell energy when energy production and consumption are not matched. All power plants also have limits on how quickly they can ramp up and down (i.e. start or stop producing energy). **These limitations imply that forecasts are not only necessary for *consumption* (demand) in order to be met by *production*, but also that forecasting errors should be kept to a minimum)**. Failure to minimize the differences can lead to wasted energy or a blackout.

Grid operators are in charge of maintaining the balance between production and consumption, by taking into account demand forecasts, and by keeping power reserves that can be drawn on during emergencies to compensate for not meeting energy supply and demand. Relying on reserves, however, will also incur additional costs, so grid operators should keep their usage to a minimum. In this sense, forecasting errors, reflected on the decisions of the grid operators, have an associated cost, when energy from reserves are used.

Due to the Hawaiian Islands relying heavily on oil and oil products, it is observed high electricity costs that help make renewable energy economically competitive. This led to an increasing adoption of grid-linked **photovoltaic energy generation** systems (i.e. energy produced from solar irradiance instead of coal and oil) at the residential and commercial scale.

Solar energy production, however, introduces a new dimension of uncertainty for grid operators. **Whereas oil and oil products were readily available for energy production, solar energy production is subject to climate conditions**. This means that aside from energy demand forecasts, **a consequent concern for the grid operator will be to know *ahead of time* the solar energy availability. In case there is no energy availability or only partially, oil reserves are then used, incurring additional costs**.

An essential component to forecast solar energy production is **forecasting solar irradiance** (measured in Watts per square meter, W/m^2) in order for energy grid operators to manage the variability in renewable energy supply, which has been subject of previous research work in the scientific literature.

1.2 Previous Work

Current state-of-the-art solar irradiance forecasting methods can be divided into two broad categories: *physical models* and *data mining methods*. *Physical models* uses mathematical equations to describe the physics and dynamics of the atmosphere. High performance computing systems and numerical methods are then used to simulate the physical models forward in time for forecasting. *Data mining methods* are typically based on extracting statistics from historical data in order to forecast solar irradiance. Examples include neural networks [6], and auto regressive methods [5]. When using *data mining methods*, we also need to decide on (1) *time granularity* (e.g. hour,day), (2) *resolution and location* (e.g. Martin et al. [3]; Moreno et al. [4] uses global irradiance while Wang et al. [6] included land-based solar irradiance), and (3) *data transformations* (e.g. calculating derivatives or using meteorology based composition of weather variables related to solar irradiance [6]).

1.3 Problem Statement

In this thesis we address a particular instance of the solar irradiance forecasting problem. Given historical and current time series data of weather variables (most notably, solar irradiance) from land-based sensors, we would like to forecast the solar irradiance for the brightest daylight hours (0800-1700) of the next day given the previous day's data (also between 0800 to 1700 hours) at a given station's site at Oahu.

As an example, consider the solar irradiance at Schofield Barracks on the island of Oahu (in Hawai‘i) between February 12th and 15th, 2014 as shown in Fig. 1.1. At 1700 on Feb. 13, 2014, we would like to forecast the solar irradiance at 0800, 0900, ..., 1700 on Feb. 14, 2014. In this particular case, an accurate forecast would be very useful for energy grid operators to plan for conventional oil-based generation for the next day (Feb. 14) which happened to be overcast.

We can elaborate the example further: Instead of using the previous days solar irradiance, can we use other weather variable (e.g. relative humidity, wind speed and direction, temperature, pressure, precipitation)? Could we use data collected at one site to forecast a different site at the island? In posing different data usages and set-ups, what we seek here is to minimize the forecasting error. The bigger the forecasting error, the higher will be the difference between supply and demand, and therefore more costs will be incurred to the company.

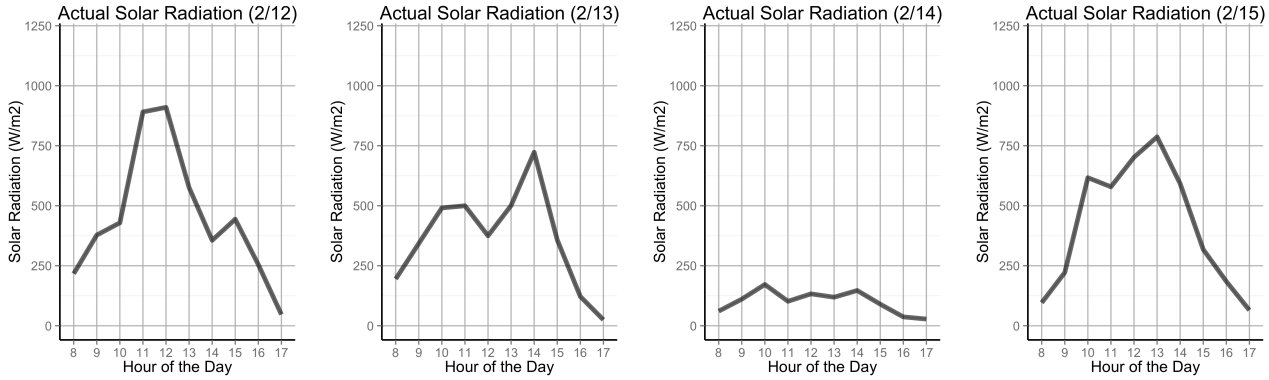


Figure 1.1: Land-based observations of solar irradiance (W/m^2) at Schofield Barracks from 02/12/2014 to 2/15/2014 between 0800 and 1700.

In order to study how well data mining methods perform for the solar irradiance forecasting problem, we chose publicly available weather observation data from land-based weather stations on the island of Oahu in the state of Hawai'i, USA to build, tune and test our data mining methods.

The remainder of the thesis is divided in chapters as follows: In **Chapter 2**, we investigate if we can perform solar forecasting using only solar irradiance data, i.e., solar models. We also define in this chapter the data mining methods, forecasting error equation and associated limitations of the methods that will be used throughout the thesis.

In **Chapter 3** we lift the restriction of only using solar data by using other weather variables past data (i.e. relative humidity, temperature, etc.) to forecast solar irradiance. Our interest is observing how the other weather variables combined or not with past solar data influence our forecasts. When considering all data available throughout several years, another question is if other external factors may influence the correlations between past and future. We investigate two of those potential external factors: Seasons, and El Nino and La Nina.

Up to Chapter 3 we only consider the effect of past data from the intended site of solar forecast. In **Chapter 4** we investigate if past data from different sites can influence our forecasts. Specifically, we are interested in observing if the conditions observed on a different site may correlate to solar irradiance in the future (e.g. due to cloud motion).

Finally, in **Chapter 5** we discuss the limitations observed in the previous chapters and introduce two new data mining methods in order to adress them. **Chapter 6** discuss our final conclusions and results.

CHAPTER 2
TEMPORAL EFFECT ON SOLAR FORECASTING

2.1 Introduction

In this Chapter, we are interested in forecasting 1-day ahead solar irradiance using past solar irradiance. We will introduce the data mining methods used and the forecasting error equation, in order to compare the forecasting of the different methods.

Since *data mining methods* extract statistics from *historical* data, what data is provided to the data mining method (i.e. days, months or years) can lead to different models' performance (i.e. higher or lower forecasting error). It makes sense then that we first investigate the temporal effect on the learned models, i.e., how using more or fewer data affect the forecasting error. Should we use all the data or just a few years suffice? Do our conclusions of temporal effect vary across sites?

To answer these questions, we will first introduce the land-based solar irradiation dataset on Section 2.2. Section 2.3 then presents the transformations we use on the data for some of the data mining methods. Finally, Section 2.4 describe the forecasting error equation and answers through experiments how the number of data and choice of data mining method affect the forecasting error.

2.2 MesoWest Dataset

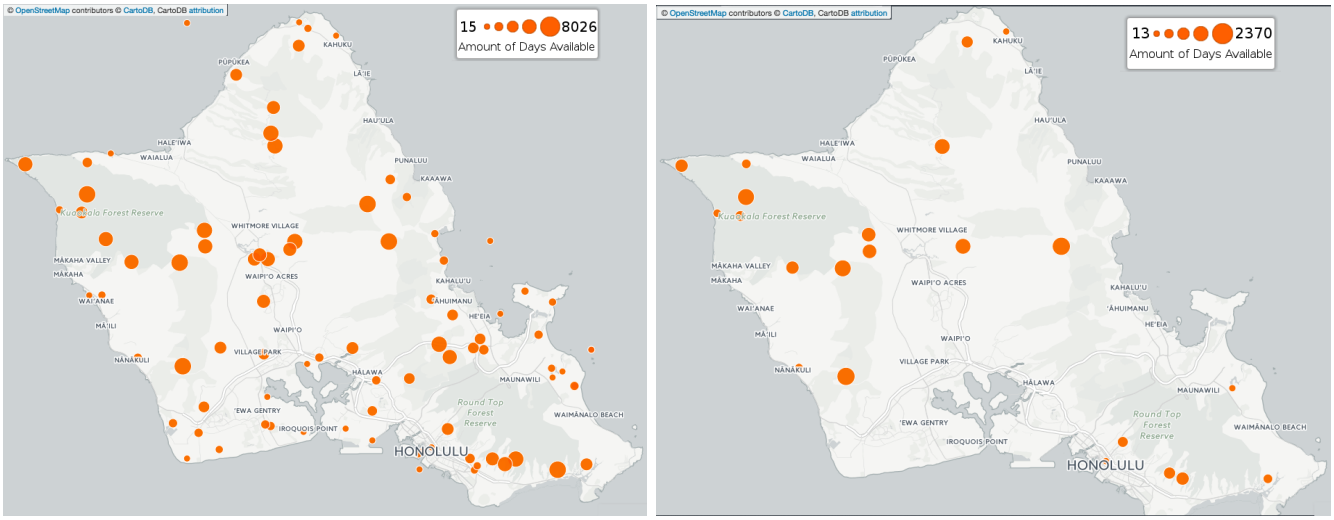
The MesoWest[1] dataset consists of current and archived weather observations across the United States. The observations are sampled hourly or every fixed number of minutes by sensors of different kinds (e.g. pyranometer for solar irradiation), and one or more of these sensors are located in each station at different locations on Oahu island. Each station has also a day of activation (sensors started sampling) and deactivation (sensors stopped sampling).

Figure 2.1 shows all stations available on Oahu island. Since we are interested in solar irradiance data across different sites, it is also important to observe which stations have a solar sensor, as shown on Figure 3.5b. Considering stations with only solar sensors filters out a considerable number of sites on Oahu Island (88 to 22 stations), however, they are still distributed well enough across the island, which is of our interest for better generalizing our conclusions.

Figure 2.2 shows the days on which all hourly samples are available between 0800h and 1700h (otherwise it is considered a missing day). We can see that some stations contain no solar irradiation available, such as AU956 and E5473 (first two lines on the plot).

Figure 2.3 illustrates some of the inconsistencies observed on the solar sensor of the Schofield Barracks station (SCBH1). **Sample problems:** (1) From this figure, we can already observe by white boxes how missing data at a daily granularity appears inconsistent over the years. Despite

[1] `http://mesowest.utah.edu`



(a) All Stations on Oahu Island

(b) Only Solar Stations

Figure 2.1: Stations on Oahu Island. Each bubble is sized according to how many days were sampled by one or more station's sensors between January 2002 and October 2015. No data at any station is available before this time range. The October 2015 upper boundary is the last day we acquired the data. (a) shows all the 88 stations that exist on Oahu island. (b) displays the only 22 stations that contain a sensor sampling solar irradiation, the main sensor of our interest.

SCBH1 being considered activated since 2002, the year of 2010 has no weather observations. (2) A second and more subtle problem occurs with existing *but incorrect* measures. Incorrect measures occurs when a sensor reports a value, however it appears inconsistent upon inspection. For example, we can observe 2013 solar irradiance is considerable low throughout most of the year, which seems inconsistent with the solar irradiation intensity on other years. The temperature for this particular year is also similar to the other years, which is inconsistent with the low number of solar irradiation. **Other Characteristics:** It is also clear from the plot Hawaii's 2 seasons impact over solar irradiation (end and start of the year), and the overall variance of solar irradiance across the year (e.g. 2015 has lighter red colors throughout the year, indicating lower solar irradiance). The effect of season will be discussed on Chapter 2 when we investigate the weather variables effect.

In summary, we can observe that data can present problems. **Missing data** can be observed either at the hourly level or at the daily level (i.e. all hours solar irradiation are missing). Even if the data is not missing, it can still be **incorrect**. To discern incorrect data from correct data, we look for inconsistencies across different sensors of the same site, for example, observing low solar irradiation and high temperatures at one year, while the remaining years having either both high solar irradiation and temperature or low high solar irradiation and temperature.

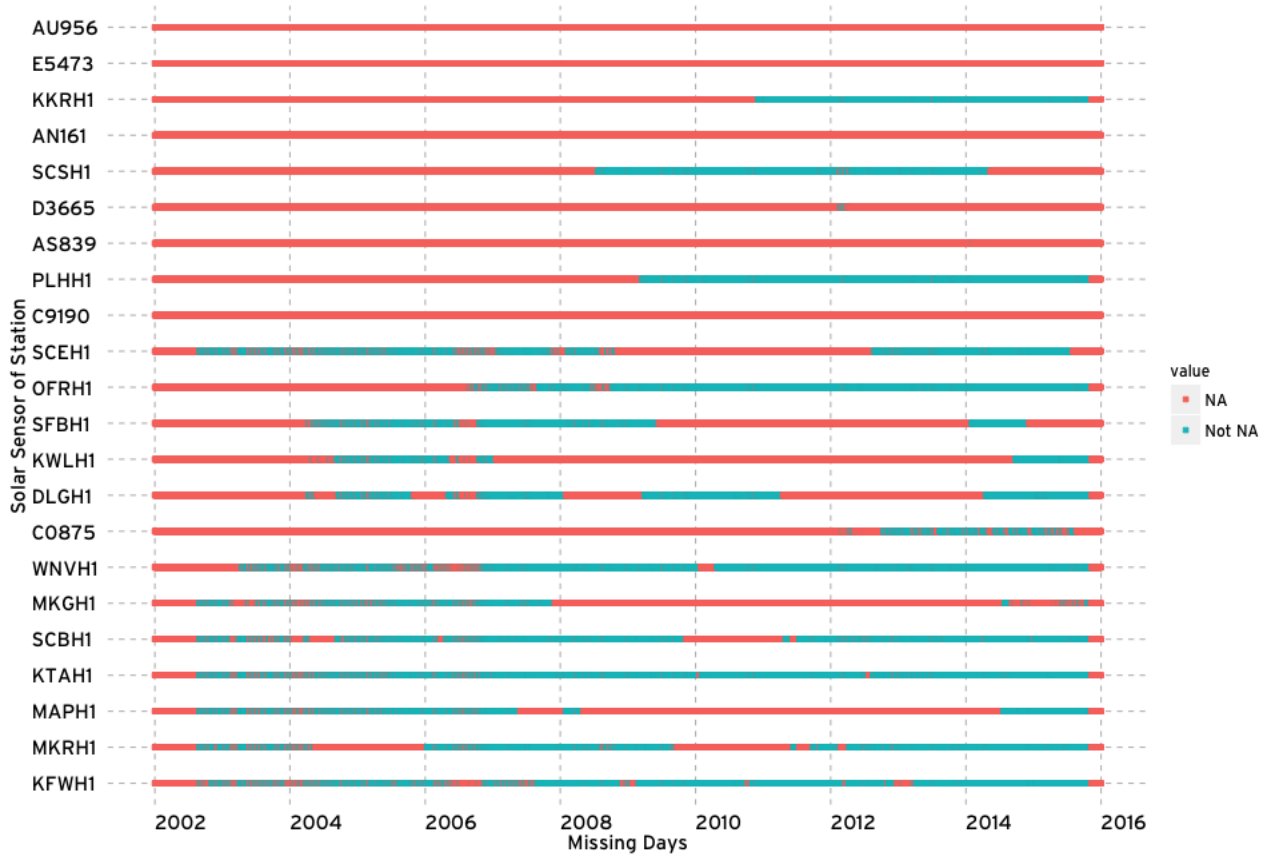


Figure 2.2: Missing Data per Day on Stations with Solar Sensor (see Figure 3.5b). A day is considered missing (NA) if at least one hourly sample between 0800h and 1700h is not available for the given day.

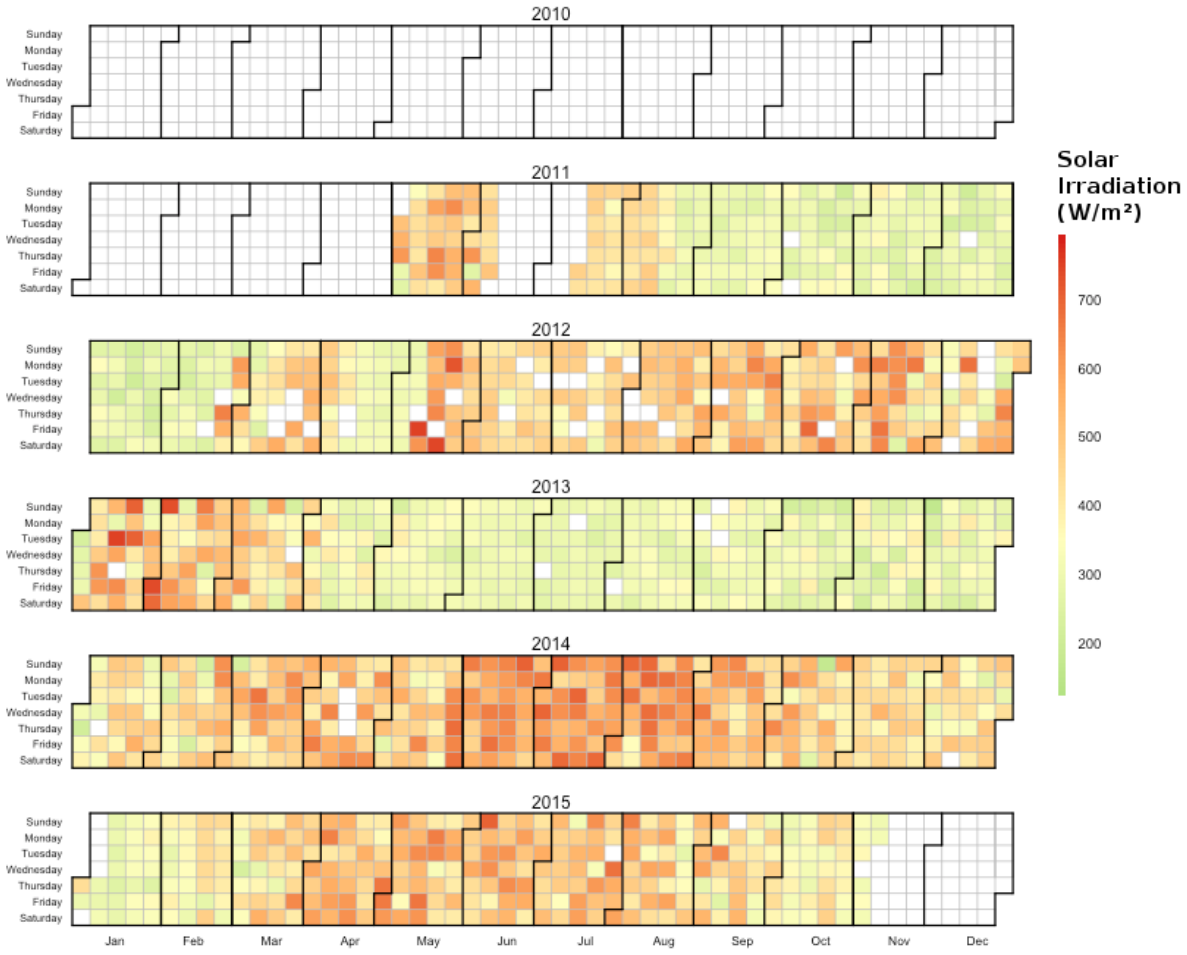


Figure 2.3: Solar Sensor Data of Schofield Barracks (SCBH1) located at the center of the island. Each day is represented by a box, and it's color is the sampled solar irradiation (W/m^2) daily. Since the sensor samples solar irradiation hourly instead of daily, we average all the hourly samples of a given day that are available to plot. If a day does not have a single sample, then the box is uncolored.

2.3 Data Pre-Processing

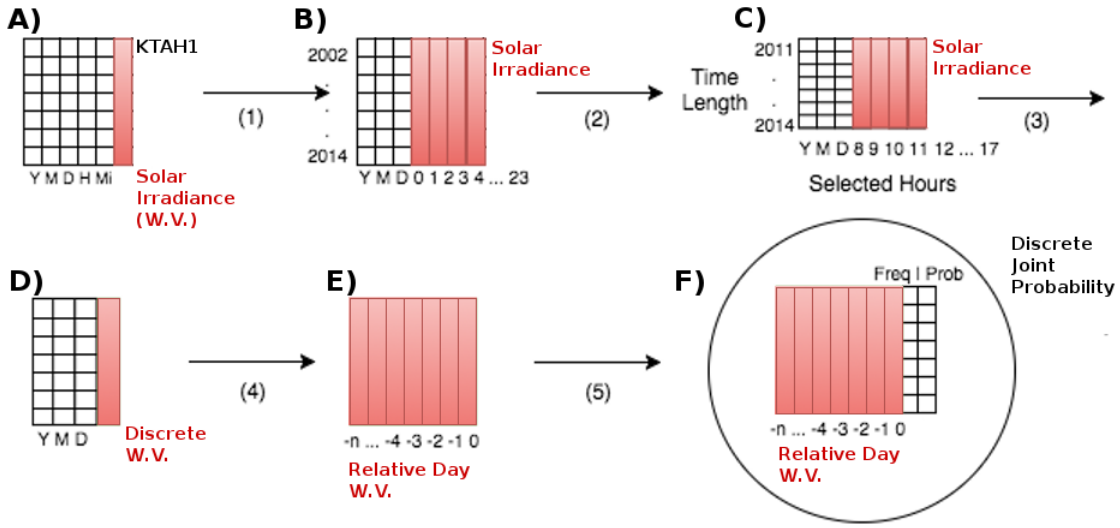


Figure 2.4: Data Pre-processing Pipeline for Solar Irradiation Weather Variable.

As we mentioned on the previous section, the stations collect weather observations through sensors, sampling every hour or minute level (only C0875 on the later case). In this section and the next we will introduce our **data pre-processing pipeline**, as shown on Figure 2.4, and how it is used by the data mining methods. To illustrate how the data is transformed and used from raw data to forecast, we will focus the pipeline on the probability model.

In Figure 2.4, Table (a) represents all the sensors data available at any given station. The timestamp extends up to the minute level, and the remaining columns are associated to each sensor, or weather variable. For example, the column highlighted indicates the solar irradiation sensor data, and each element is the weather observation sampled hourly or aggregated to hour by averaging. Since on this Chapter we are only interested on using solar irradiance data, our pipeline will focus on pre-processing and forecasting this weather variable alone. In further Chapters we will revisit this pipeline and show how we include other analysis. The next steps, which we will discuss in more details afterwards, are as follows:

- **Step (1) - Weather Variable Selection:** Takes (a) as input and aggregates the data from minute granularity to hour granularity, now exhibiting the solar irradiation at every hour from 0h to 23h as **columns**, resulting in Table (b).

- **Step (2) - Data Filtering:** Takes (b) as input and and filter both **rows and columns**. We filter **rows** to select the number of years to create a train and test dataset. We filter **columns** to use only the hours between 8h and 17h inclusive, the brightest hours of the day. After the filtering step, we obtain Table (c).

Tables (a) to (c) contain the common pre-processing pipeline for all data mining methods. The bottom 3 Tables are the pre-processing steps for the probability data mining methods.

- **Step (3) - Discretization:** Solar irradiation is obtained through the solar sensor in W/m^2, a numeric quantity. Probability models requires the data to be countable. The discretization step takes as input Table (c) and reduces the solar irradiation intensity columns from 8h to 17h (10 columns). to a single *discrete weather variable column* representative of the 10, which is now countable and results in table (d).

- **Step (4) - Tuple Extraction:** Step (4) take as input Table (d) and applies a sliding window over the *discrete weather variable column*, extracting a set of tuples associated to a number **w** of consecutive days specified a priori. The set of all tuples extracted constitutes Table (e).

- **Step (5) - Discrete Probability Distribution Estimation:** The tuples Table (e) can have repetitions. The repetitions are then counted and removed, resulting in a frequency table. Normalizing the frequencies results also in a probability column, the discrete probability distribution estimated from the data.

We will now discuss each step and intermediate table in detail.

A data mining method always follow a **train** and **test** set-up. During the **training** stage, data is input to the data mining method so it can learn any statistics that exist within the data, resulting in a learned model of the data. The choice of the data mining method also reflects our assumptions of the patterns' statistics that will be computed from the data. For example, a linear regression method will assume the patterns are linear. During the **test** stage, a model that has been **trained** will be used for forecasting data which *was not used for training* (e.g. the year ahead). For example, a data mining method can be **trained** using solar irradiation data from 2012 and 2013, and be **tested** by forecasting 2014 data.

In order for the model to be evaluated during the **testing** stage, some measure of performance, or **forecasting error** must be defined.The **train** and **test** set-up also closely relates to an operation set-up, where some data is available for training (i.e. past solar irradiation) and some data will be of interest to forecast (i.e. "future" solar irradiation). As time moves forward, the future solar irradiation which is intended to be forecast will become known, and, therefore, can be used to evaluate the model forecasting error as a **test** dataset.

To more visually present the data pre-processing for the models, and how the models use this data for forecasting, we will use a table metaphor referring to Figure 2.4. Figure 2.5's elements, for example, presents in more detail table (b) of Figure 2.4. The table contains two types of columns: **Timestamp** and **Hourly Observations**. The **timestamp** columns identify the date which the row was sampled by a sensor. The **hourly observation** columns are the sampled weather variable (e.g. solar irradiation) value at the given hour.

For training the models of our work, only the first and fifth day observations on Figure 2.5 would be considered, since the other days contain missing hours. Note that in the **train** and **test** set-up we would have two of these tables, e.g. one containing 2012 and 2013 data for training the model, and another table containing 2014 for testing the model. This split of train and test occurs at step (2) of Figure 2.4 when we select the time length of each table for 2012 and 2013 for training, and 2014 for test.

We will discuss how the training table is used next, and how the test table is used on the following sub-section.

	Timestamp			**Hourly Observations**			
	YEAR	**MON**	**DAY**	**8**	**9**	**...**	**17**
1	2003	1	1	322	507	...	7
2	2003	1	2	131	236	...	
4	2003	1	4			...	
5	2003	1	5	100	120	...	7

Figure 2.5: An example of solar irradiation observations over 5 days as obtained from a solar irradiation sensor of a station. The table also is representative of the missing data problems: The first and fifth day contain all observations between 0800 and 1700. The second day has a missing observation at 1700h. The third day observation was never reported by the sensor. The fourth day was reported, but has no solar observation between 0800-1700h. This Table is represented in the pipeline of Figure 2.4 as Table (b).

There are different ways in which we can create data mining *models* from training data to forecast 1-day ahead between 0800h and 1700h. For example, starting from Table (c) on the pre-processing pipeline of Figure 2.4 we can create one linear regression model for each next-day hour by training it only with solar irradiation observations from 0800h of 2012 and 2013, and have it forecast only 0800h of all days of 2014. However the model would only learn the changes of 8am at every day. A more interesting model is training a linear regression model using the solar irradiation from 0800-1700 (a total of 10 coefficients would be estimated), to forecast the next day 0800h. If we now similarly create one model for 0900h, 1000h, ... up to 1700h for a total of 10 models (one for each hour), using the 10 hours from the preceding day, we can use the 10 models to obtain our forecasts of 2014.

For a *discrete* probability model, which requires as input a discrete probability distribution, the extra steps (3), (4) and (5) of Figure 2.4 are required to estimate the discrete probability distribution from the Table on Figure 2.5, namely making the daily solar irradiation countable. The discretization step (4) can be done, for example, using *binning* or *k-means clustering*.

Both methods will result in a new *cluster id column*, to which each day is represented by, as shown on Figure 2.6 and Table (d) on the pipeline Figure 2.4. The values a cluster id can take are subject to the parameter k, the number of clusters we specify a-priori to the algorithm (or

	Timestamp			Hourly Observations				Cluster ID
	YEAR	**MON**	**DAY**	**8**	**9**	**...**	**17**	
1	2003	1	1	322	507	...	7	5
5	2003	1	5	100	120	...	7	1

Figure 2.6: Table from Figure 2.5 after applying *binning* or *k-means clustering*. Notice only the first and fifth day are kept, since the others contain missing data problems. In the pipeline 2.4, this table is represented by table (d).

analogously, the number of bins). For example, if $k = 5$, then the cluster id column can only take a value from 1,2,3,4 or 5. Each cluster id also has a *centroid*. A *centroid* contains solar irradiation for the hours between 0800-1700h and how these values are calculated depend on the method used. Following our example from Figure 2.6, on Figure 2.7 we can observe the centroids to which they could be mapped.

Cluster ID	Hourly Observations			
	8	**9**	**...**	**17**
5	322	507	...	7
4	231	436	...	2
3	150	300	...	1
2	100	150	...	3
1	95	137	...	0

Figure 2.7: An example of a centroids table.

We will call the process of applying either *k-means* or *binning* **discretization**. We **discretized** the data from Table 2.5, obtaining a new *cluster id column* as shown on Table 2.6, and the associated *centroids* for each cluster id, as shown on Table 2.7.

Since a cluster id is nothing but a mapping from the 10 hours solar irradiation of each day to a single value, we can now estimate a discrete probability distribution using the cluster ids of every day, therefore *training* the probability model, forecast the cluster ids for 2014, and finally re-map them to hourly solar irradiation to calculate the forecasting error.

If we examine Tables 2.6 and 2.7, we can imagine different rows will contains the same cluster id. However, a cluster id only contains one associated centroid. We say that the hourly observations were **mapped** to cluster ids. Again, recall this is required for the discrete probability distribution estimation, and therefore training the probability models. Since the probability model will take as input cluster ids, it will also forecast cluster ids for each desired day. In order to obtain the solar irradiation hourly, we can use the centroids from Table 2.7 to replace the cluster id of each forecasted day, i.e. **map** from cluster-id to hourly observations.

Notice this **mapping** step will incur some forecasting error to the overall forecast. This error

is *not* associated to the probability model, rather, the forecasting error of the probability model depends solely on whether the correct cluster id is forecast or not. **For probability models**, therefore, the **forecasting error** contains two components: The forecasting error associated to the **clustering mapping**, and a forecasting error to **wrong cluster id forecasts**. When we discuss the forecasting error equation, we will also define how to assess the error separately for the two components.

We now discuss in more detail the difference between *binning* and *k-means* to generate the *cluster id column* and how the *centroids* are calculated for the **training** data. We conclude the section explaining how the *cluster id column* and *centroids* are generated for the **test** data.

2.3.1 K-Means

K-means is a clustering process that organizes points (rows in our previous tables) into clusters (groups), where the cluster elements are *similar*, and similarity is defined by a similarity function. If we were only considering 2 hours instead of 10 (0800h to 1700h), we could for example plot each day solar irradiation in 2D space. K-means would then group the points who are closest to each other, while attempting to maximize the distance between the points from the other groups. Since days are represented by 10 hours, the clustering algorithm then performs the clustering operation in 10 dimensions. There are multiple ways to define the *similarity* function, and here we chose the euclidian distance (Eq. 2.1).

$$d(p,q) = d(q,p) = \sqrt{(q_8 - p_8)^2 + (q_9 - p_9)^2 + \cdots + (q_{17} - p_{17})^2} \quad = \sqrt{\sum_{i=8}^{17}(q_i - p_i)^2}. \qquad (2.1)$$

Where **p** and **q** are a row's hourly observation as shown on Figure 2.6, and p_i or q_i is a given hour solar irradiation between 0800-1700h.

K-means also requires the number **k** of **clusters** to be specified a priori. Quantitatively, we would like to choose a **k** that minimizes the distance of the rows' hourly observation within each cluster and maximize their distance between the clusters (i.e. similar solar irradiation rows are grouped together). Qualitatively we would like the clusters to be meaningful (e.g. on Figure 2.8 it is expected that peaks of solar irradiance can occur early, later, and also be absent).

We observed for k=5 that the centroids were the shapes with most distinct and therefore chose this value for k. Every cluster value on Figure 2.6 will therefore be either 1,2,3,4 or 5. Once the rows are mapped to each cluster by successively comparing one with another using Equation 2.1, each centroid will be the mean of all the rows that have the same id, as shown on figure 2.8.

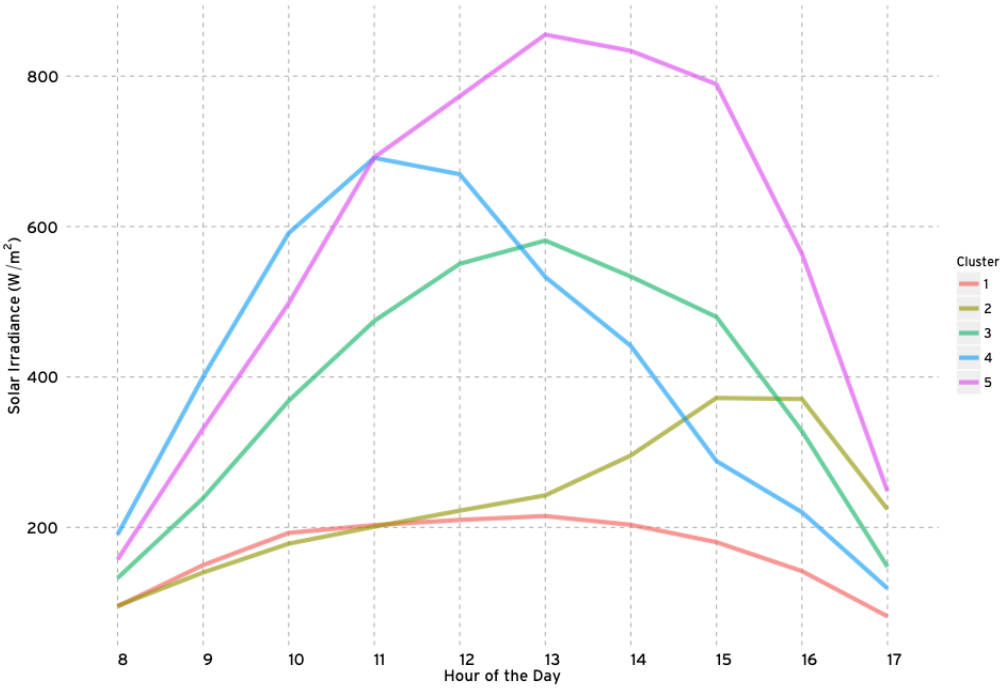


Figure 2.8: SCBH1 Centroids. Each daily sample of Table 2.5 is aggregated and discretized to one of the k=5 clusters obtained by the K-Means process.

2.3.2 Binning

An alternative way to group the rows is **binning**, commonly used to create histogram plots. To perform binning on Figure 2.6, we **average** the solar irradiance of all hours of each day. Visually, we are collapsing the rows of Figure 2.6 into a new column of hourly averages. Since typically the solar irradiation curve from 0800-1700 is similar to a bell curve with one peak, the average effectively groups the rows on the peak between 0800h to 1700h occurs. Considering all values of the new averages column and taking it's minimum and maximum, we split all the averages into **5 bins** as shown on Figure 2.9. Note also that after binning, we have performed a transformation similar to k-means clustering, serving as an alternative method.

Comparing both Figures 2.9 and 2.8 we can visually observe how for the same number of clusters/bins both processes aggregates data differently. In essence, while the binning method represents all values with the same hours, the clustering method allow each hour to be different.

2.3.3 Clustering and Binning Test Data

On a data mining forecast set-up, clustering and binning are applied differently between **training** and **testing**. During **training**, both clustering and binning will be performed as described on the previous sections. This is emphasized in Figure 2.10, where the train table contains the frequency

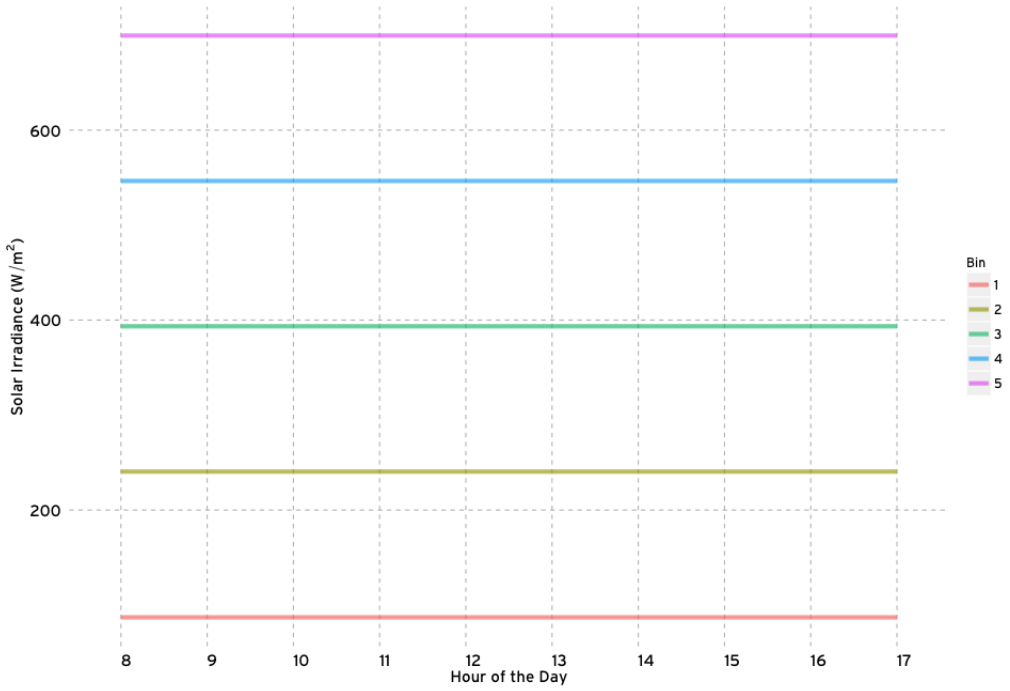


Figure 2.9: SCBH1 Bins. Each daily sample of Table 2.5 is aggregated and discretized to one of the k=5 bins obtained by the binning process.

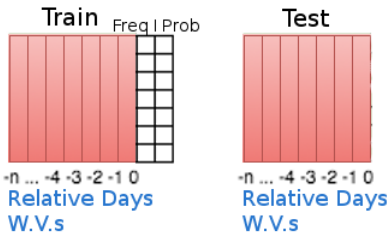


Figure 2.10: Train and Test Tables.

and the estimated discrete probability required by the probability models. The test table however only requires the tuple extractions step of the pipeline in Figure 2.4, and does not include the final step to estimate the probability distribution.

A second difference occurs at Step 3 in how the data is **discretized**. Specifically, for the **test** we do **not** add the **test data** to the **training** data and re-cluster or re-bin all the rows, as this would be a too expensive computation for every new row. Instead, for each new row available on every new day, we compare it to the clusters or bins centroids of the **training data** and label the new row cluster id with the one that is the most similar. Visually, a new row will also be of the form on Figure 2.5, which is also represented by a line on Figures 2.8 and 2.9. Using the euclidian distance Equation 2.1, we can compare the new row against all the 5 lines, obtaining a number (the distance, or similarity) for each cluster or bin (depending on which method was chosen), and finally label the new row with the one of lowest distance. This process is repeated for all the test dataset rows to cluster or bin them for the data mining model.

The cluster/binning methods here will be used with some of the data mining models described next. On the experiment section 2.5, we will explicitly note which has been used on the models by prefix the forecasting method acronym with KM for K-Means or BIN for Binning. If the forecasting model does not have a prefix, then it does not use them.

2.4 Data Mining Methods

In the previous section we described how the data can be pre-processed for training a data mining model. Here we detail how the data is used by the methods.

2.4.1 Probability Model

With the obtained cluster id column for the training and test dataset, which results at Table (d) of the pipeline in Figure 2.4, we now discuss how the discrete probability distribution is estimated for the probability model, and then used for forecasting the test dataset.

We will consider now the tuple extraction step (4). Consider a moving window of size $\mathbf{w}$ which starts at the top $\mathbf{w}$ elements of the cluster id column in Figure 2.6. For instance if $\mathbf{w} = 2$, then it contains the tuple (5,1) initially. For the sake of the example, consider the table contains another 2 days, for a total of 4 days, and let the cluster id column associated to these 4 days be 5,1,5,1. The window, moving forward one day at a time, will move as follows:

$$\underline{5,1},5,1$$
$$5,\underline{1,5},1$$
$$5,1,\underline{5,1}$$

If we consider all the tuples contained for every time the window moves forward, then we would have the following tuples:

(5,1)
(1,5)
(5,1)

Or 2 tuples (5,1) and 1 tuple (1,5). Note that the elements of these tuples are also indexed by time, if we observe figure 2.6. Furthermore, according to Figure 2.6, remember the first tuple (5,1) is not from consecutive days, and in this case would be discarded from the following steps to train the model. The set of all tuples constitutes Table (e) in the pipeline.

Effectively, this procedure allows us to count across the whole training dataset cluster id, performed by step (5) in the pipeline, given a number of consecutive days **w** represented by the window size, how many times a sequence of clusters (or bins) occurred throughout the training dataset, therefore generating a **frequency table**. Normalizing (dividing all counts by the highest count value of the entire table) the counts of the **frequency table** results in a discrete joint probability distribution, resulting on the final table (f) of the pipeline.

We now define S_t to be the discrete random variable for the row on Figure 2.6 at a particular date t. Oftentimes, we will use relative dates in the subscript of S_t instead of the actual date such as $t = 20030101$. So for instance, the three tuples $(s_{20030101}, s_{20030105}), (s_{20030105}, s_{20030106}), (s_{20030106}, s_{20030107})$ could be represented as (s_{t-1}, s_t).

Given a window size w, we construct a joint probability distribution as

$$P(S_t, S_{t-1}, \ldots, S_{t-w+1}), \tag{2.2}$$

and use the following prediction function to forecast S_t on the test dataset cluster id column, given that the previous $(w-1)$ rows are $\langle s_1, s_2, \ldots, s_{w-1} \rangle$,

$$\hat{s} = \begin{cases} \arg\max_s P(S_t{=}s | S_{t-1}{=}s_1, S_{t-2}{=}s_2, \ldots, S_{t-w+1}{=}s_{w-1}) \text{ if } P(S_1, S_2 \neq 0) \\ \arg\max_s P(S_t = s) \text{ else} \end{cases} \tag{2.3}$$

To understand how the prediction function works, suppose the tuple $(s_{t-1} = 1, s_t = 2)$ has the highest frequency among $(s_{t-1} = 1, s_t = 1)$, $(s_{t-1} = 1, s_t = 3)$, $(s_{t-1} = 1, s_t = 4)$ and $(s_{t-1}1, s_t = 5)$. Upon obtaining a tuple $(s_{t-1} = 1, s_t = \mathbf{3})$ of the test dataset, the model would guess, wrongly in this case, $(s_{t-1} = 1, s_t = \mathbf{2})$. If **w=3**, then the tuples would be of the form (s_{t-2}, s_{t-1}, s_t), for both training and test dataset and we would use s_{t-2}, s_{t-1} of the to forecast s_t analogously. More formally, we call $(S_{t-1}, \ldots, S_{t-w+1})$ the **predictor variables**, given the **dependent variable** S_t.

Last, since the model requires part of the tuple counts to forecast a test dataset, it is possible this part never occurred on the training dataset, and therefore the model would not be able to choose. More generally, For the days when the previous $(w-1)$-day sequence $\langle s_1, s_2, \ldots, s_{w-1}\rangle$ does not occur in the training data at all, the conditional distribution does not exist for that sequence and we return the most frequent tuple's s_t of the entire training dataset (using the prior distribution) as the prediction.

2.4.2 Naive Bayes Classifier

An alternative to using the joint distribution with different window size w is to apply the Naive Bayes assumption, i.e., the predictor variables $(S_{t-1}, \ldots, S_{t-w+1})$ are independent given the dependent variable. The row on the date t can then be predicted using

$$\hat{s} = \arg\max_{s} P(S_t{=}s) \prod_{i=1}^{w-1} P(S_{t-i}|S_t{=}s). \tag{2.4}$$

Observe that the Naive Bayes assumption is used to factorize the full conditional distribution into a product of lower-order joint distributions ($P(S_{t-i}|S_t{=}s)$'s). For small data sets, estimating the lower-order distributions would often be more accurate than estimating the full joint distribution due to the missing forecasts.

2.4.3 Random Forecasts and Most Frequent Forecasts

To have a baseline to compare these 3 models, we also considered two prediction functions: Random and Most Frequent. The random prediction function forecasts randomly a cluster id between 1 and k for every day of 2014. Comparing the other models error to this one for example gives us an idea how better we are forecasting compared to throwing a dice of k faces.

The Most Frequent Forecast prediction function was introduced earlier on the probability model on the **else** portion, when the probability model is unable to find a tuple from the training data that match the predictor variables. This means that on the extreme case where the probability model is unable to forecast any day, it's prediction function is reduced to the Most Frequent prediction.

2.4.4 Linear Regression

Given that the solar irradiance time series is continuous, another possible data mining method is to use linear regression. In this case, we do not require turning the data countable, sufficing to pre-process the data up to Table (c) in the pipeline, and therefore not requiring **discretization**. Let each data point in the solar irradiance dataset (train or test) be denoted by S_t, where t denotes the timestamp at the granularity of hours. To forecast the solar radiation of each hour 1-day ahead, we create a separate *linear regression model* for each hour of the next day using all hours of the

previous (consecutive) w days. For example, the model for predicting $S_{20140214.0900}$ with $w = 2$ (1 day before) would be,

$$\begin{aligned} S_{20140214.0900} &= c_1 \cdot S_{20140213.1700} \\ &\quad + c_2 \cdot S_{20140213.1600} + c_3 \cdot S_{20140213.1500} \\ &\quad + \ldots + c_{10} \cdot S_{20140213.0800} + c_{11}. \end{aligned} \tag{2.5}$$

Hence, there would be 10 linear regression models for each hour between 0800 and 1700. We use the standard least squares algorithm to obtain the coefficients c_i for each model.

Observe that across all methods we are faced with the choice of $\mathbf{w}$, i.e., how many consecutive days. This will also be discussed next on one of the experiments.

2.5 Experiments

We will now address the following questions:

- *Experiment 1*: Is there any significant difference on forecasting error using different data mining methods for solar forecasting?
- *Experiment 2*: Does using more past years of solar irradiance decrease the solar forecasting error?
- *Experiment 3*: Can we observe any relationship between the number of past years used and the data mining methods ability to forecast?

In order to do so, we will define here what we mean by **forecasting error**.

2.5.1 Forecasting Error

As noted previously, for the probability model and the naive bayes classifier, we need to estimate the discrete probability distribution, and therefore make the data countable. In this case, the forecasting error can be observed as a contribution from **clustering mapping** and the **cluster id forecast**. The former is subject to the choice of k and the clustering method (k-means or binning). The later is subject to the number of consecutive days w and the prediction function used. Altogether the mapping and forecast will result in a **forecasting error**. Let's first consider the forecasting error, i.e., the contribution of **both** steps on the final error.

We measure the accuracy of our forecasting models by measuring the mean absolute error (MAE) of each daylight hour being forecasted (0800-1700). Given a testing dataset, with hourly

time points (daylight hours only) indexed from 0 to n, MAE is defined as

$$\text{MAE} = \frac{1}{n}\sum_{t=0}^{n}|s_t - \hat{s}_t|. \tag{2.6}$$

To evaluate the performance of the models, we split the dataset into a training dataset (earlier years) and a testing data set (2014 data). The MAE is measured on the testing data set. We did not use cross validation because the use of earlier years for training and later years for test better mimic operational usage of such forecasts. Since the probabilistic models predicts the cluster ids for the next day, the hourly solar irradiance representation of the clusters, i.e., the centroids, as exemplified in Figures 2.8 and 2.9, is used to compute the MAE. Note that the maximum solar irradiance during the sunniest hour on the sunniest day is approximately 1200 W/m^2.

Since the linear regression don't use a discretization step, the forecasting error is the same as the prediction error.

Clustering Mapping Error. The clustering error can be calculated also using the MAE Equation 2.6. To obtain the **clustering mapping error**, we simple map the **test dataset of 2014** to cluster ids, then re-map them to hourly solar irradiation *using the centroids*. If we now subtract the actual hourly solar irradiation from the ones obtained using the centroids, we will obtain the **clustering mapping error**. This is equivalent to having a probability model forecasting correctly all the cluster id of 2014. The **forecasting error** in this set-up would then be associated to the clustering step only.

Cluster ID Forecast Error. The cluster id forecast error can be calculated in two ways: Either hourly as done previously, or by simple observing if the cluster id forecasts matches the test data cluster id. On the former case, if we subtract the **clustering error** from the **forecasting error**, we can obtain the **Cluster ID Forecast Error *hourly***. The hourly version of the Cluster ID Forecast error is useful to observe when comparing the models how much of the total error the mapping and cluster id forecast contribute.

Missing Forecasts. Our last metric to assess the performance of the probability models are counting missing forecasts, since part of the forecasting can be due to switching to the *else* statement on Equation 2.3.

2.5.2 How different data mining methods perform solar forecasting using only solar data?

Experiment Setup. To compare the different models we introduced across the different sites, we used the years of 2012 and 2013 to train and the year of 2014 to evaluate them. To select the stations, we used Figure 2.11 to identify which out of the 22 stations containing a solar sensor had data for 2012, 2013 and 2014.

To complement our analysis beyond missing data to *incorrect data*, we used calendar plots (such

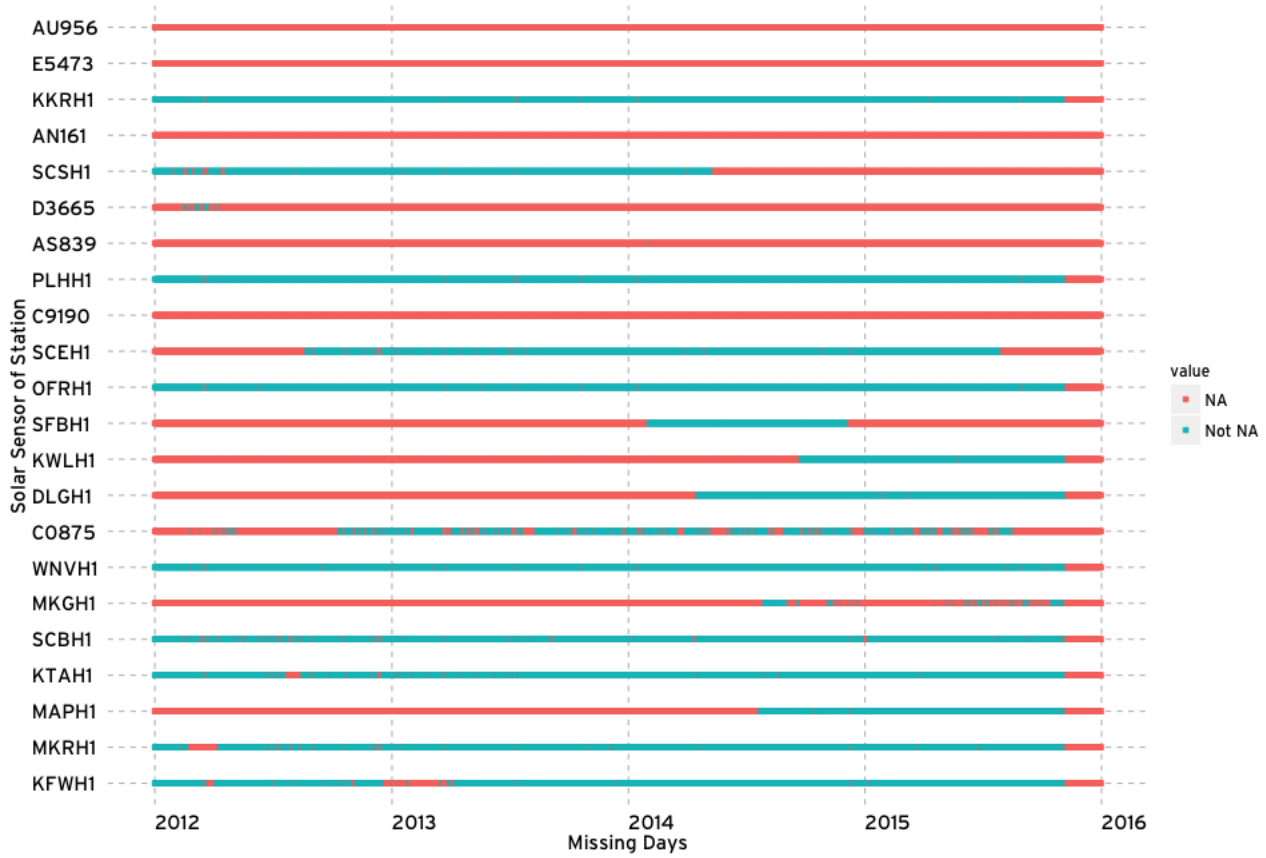


Figure 2.11: Missing Data per Day on Stations with Solar Sensor after 2012 (see figure 3.5b for the location of all stations over the map). A day is considering missing (NA) if at least one hourly sample between 0800h and 1700h is not available for the given day.

as in Figure 2.3) to inspect each one of the stations for quality issues (the calendar plots of each station are available on the supplemental material). We first removed the stations containing no data, and compared each year on the calendar plot for every station to verify data inconsistencies. Years that contained strange solar irradiation intensities, as seen in 2013 on Figure 2.3, were considered missing data for the entire year, and therefore were either not used or clearly highlighted on the error plots on the following experiments.

We also considered as missing data years such as 2011 in which less than half of the data was not available in an unbalanced manner (i.e. only the end or the start of the year was available). By using this criteria, we ensure the trained models will not learn from incorrect data, or bias towards one part of the year alone. After applying this criteria, we obtained Table 2.1 which highlight all the years in which we consider data available.

Station	**Available Years**
KKRH1	2011-2014
PLHH1	2009-2014
SCEH1	2003-2008;2013-2014
C0875	2013-2014
SCBH1	2005-2009
KTAH1	2003-2014
MKRH1	2003;2006-2009;2012-2014
KFWH1	2007-2014
OFRH1	2007-2008;2010-2011;2014
WNVH1	2003-2005, 2007-2009, 2012-2014

Table 2.1: Correct or with sufficient number of data years on stations.

Results. Figure 2.12 compares the forecasting error for all the models across all different sites.

We can note the probability model combined with K-means (kmPM) has, overall, lower or equal forecasting error to the other models, except for two of the stations KTAH1 and PLHH1, in which Linear Regression (LR) has lower error.

We can also compare how each model performs on different sites. Observing the maps on Figures 2.14, 2.15, we can see the forecasting error of all the models *relative to the site location* appears consistent across all the models. Comparing the 4 model maps to the standard deviation of the solar irradiance of each site on figure 2.13, we can see the forecasting error fluctuations seem to also associated to the standard deviation (the circles have similar sizes across all the models and standard deviation).

Comparing both maps of Figure 2.13, the variation, relative to the other sites, appears to be consistent between train and test, aside from KFWH1 and SCEH1. This may be due to SCEH1 only contain one year of data available (2013), according to Table 2.1, making the statistic more vulnerable to small variations. In KFWH1, the increase on the standard deviation may be due

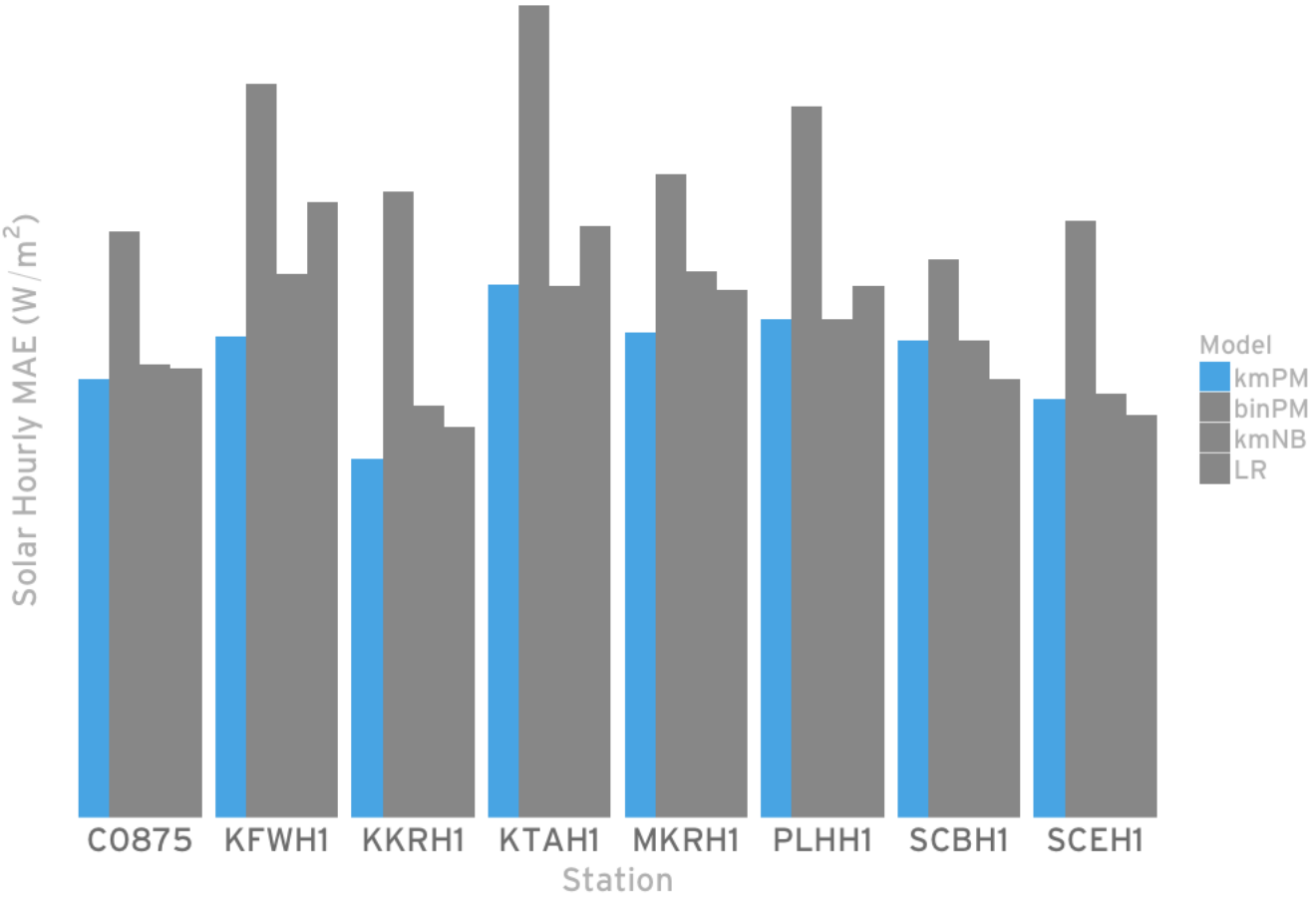
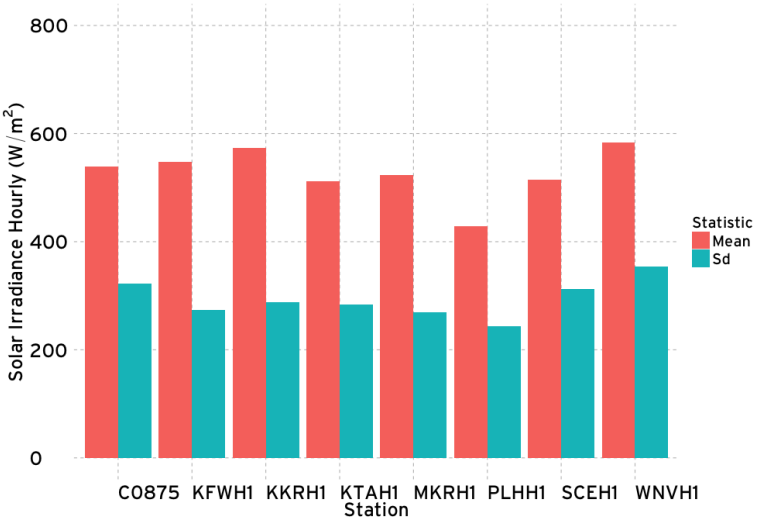
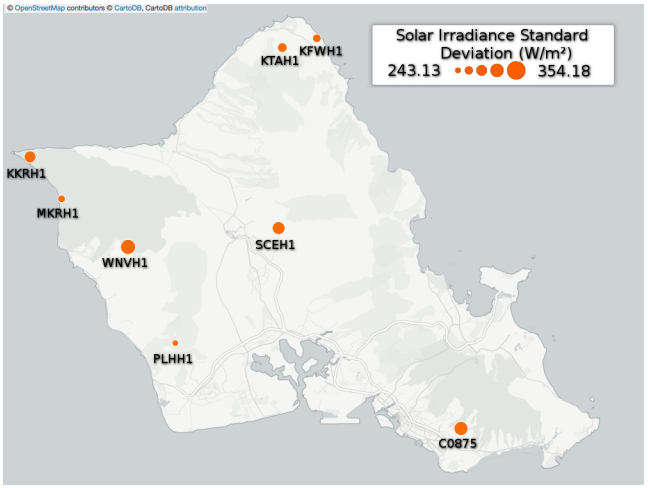


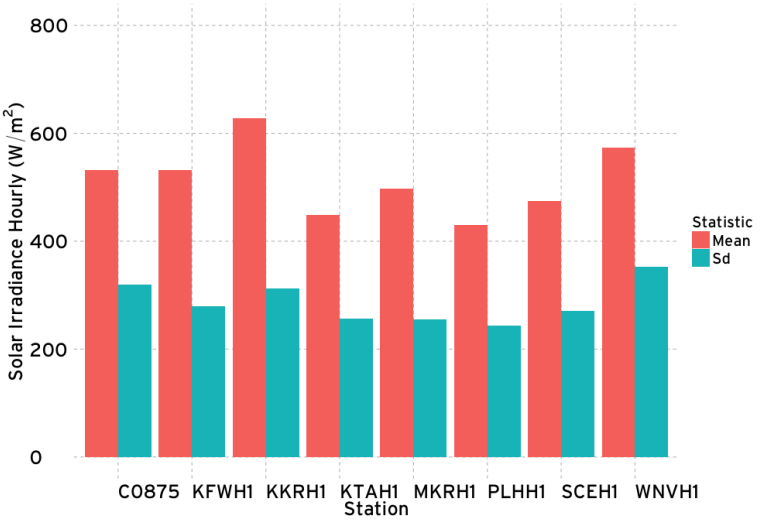
Figure 2.12: Solar Hourly MAE (W/m^2) for the four different models. All models used only the previous day (w=2), training on 2012 and 2013 to forecast 2014.



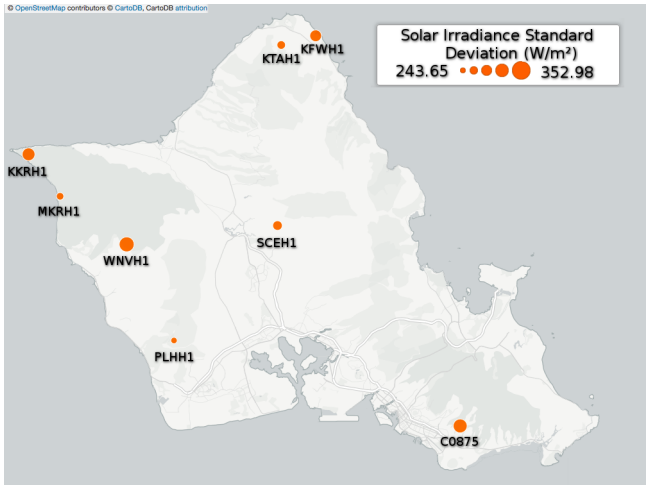
(a) Training(2012,2013)



(b) Training (2012,2013)



(c) Test (2014)



(d) Test (2014)

Figure 2.13: Solar Irradiance Standard Deviation for Train (2012, 2013), Test Data (2014) (W/m^2) for selected stations.

to the years of 2003, 2004 and 2005 containing potentially incorrect data of 0 solar irradiation. This emphasizes the need of proper data evaluation when building the data mining models for the experiments and interpreting their results. We can also note that the solar irradiation varies to a small number across the sites, which we will compare to the forecasting errors next.

The clustering error contribution for the forecasting error on kmPM shown on Figure 2.15 was on average 0.61%, with a standard deviation (SD) of 5%. across all sites. The cluster id prediction function of kmPM forecasted correctly across all sites 39% on average, and 7% SD, which was higher than both the most frequent centroid prediction function (Mean=32%,SD=6%) and the random prediction function (Mean=23%,SD=9%). Finally, out of the 365 days of 2014 cluster id predictions, all but C0875 had only between 2 and 6 missed forecasts, whereas C0875 had 108 missed forecasts, over a quarter of the total forecasts (29%)!

Conclusions. The answer to our question becomes clear on Figure 2.12: The K-Means probability model overall had a better performance than the other models. Due to this, the remainder of the thesis will focus on the kmPM. We also observed that the forecasting error appears to correlate to the variance on the maps. The kmPM performed better than the random function and most frequent centroid, however the forecasting error components, cluster mapping error and cluster id prediction, could still improve, given less than half of the predictions are done correctly, and over half of the forecasting error is due to clustering on this set-up. Finally, we observed that C0875 has a considerable number of missing forecasts when compared to the other site, due to only contain 1 year of training data. It makes sense therefore, that our next question investigate the effect of the number of years on missing forecasts, as they reduce the probability model to the most frequent centroid prediction, which as has been shown, has higher forecasting error.

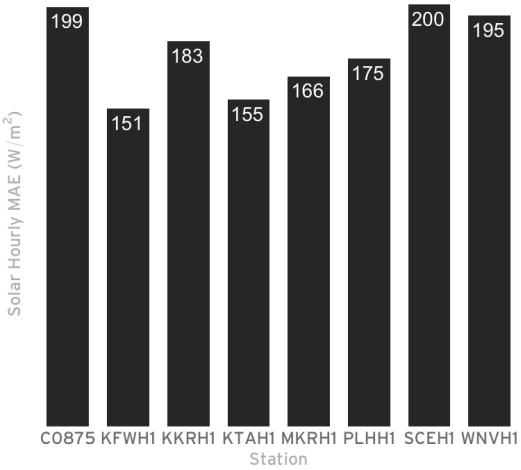
2.5.3 How different amount of years influence solar forecasting?

Experiment Setup. To verify the impact of adding more years affect solar forecasting across different sites, we chose KTAH1 and KFWH1 stations, which contain the highest number of consecutive years, as shown previously on Table 2.1.

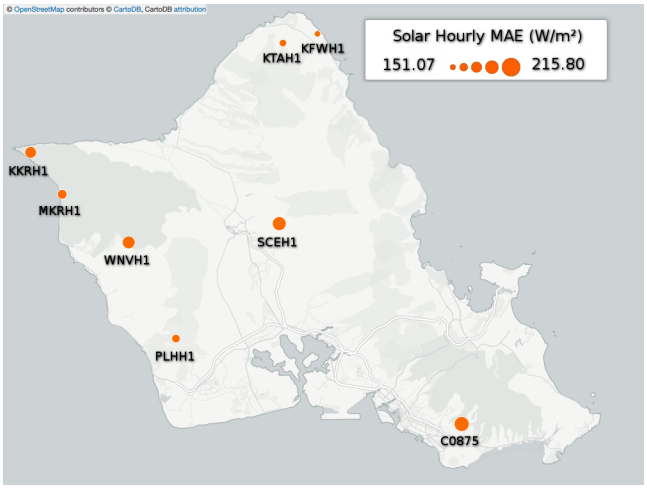
We constructed a set of kmPM models for KTAH1, and another set for KFWH1, such that each set contained 10 models with increasing number of years, starting on 1 year training (2013). For example, one model contained 1 year of training data (2013), another model 2 years (2013,2012), etc up to one model containing 10 years (2013-2004) to forecast 2014.

Results. Figure 2.16 shows the forecasting error as years are added to the probability model with K-Means. By observing both stations, we conclude adding more than 2 years does not improve significantly the forecasting error, since by observing the plot bars they vary only slightly or by 50 W/m^2 only on KTAH1.

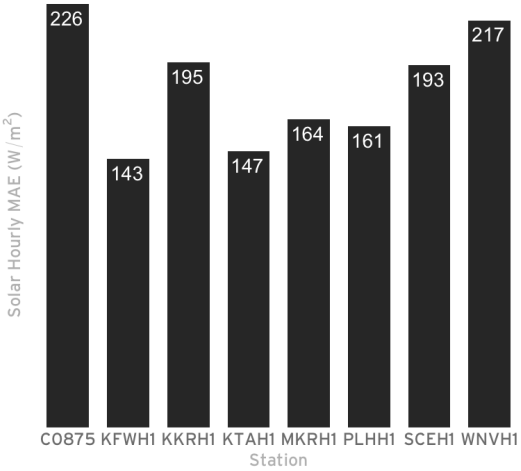
We can again decouple the forecasting error to observe the clustering and prediction contributions. On KFWH1, the clustering error contribution mean remained the same at 62%. The



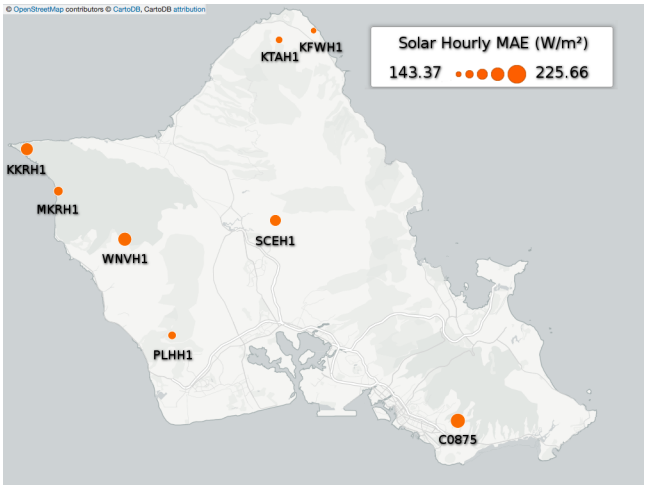
(a) Naive Bayes



(b) Naive Bayes

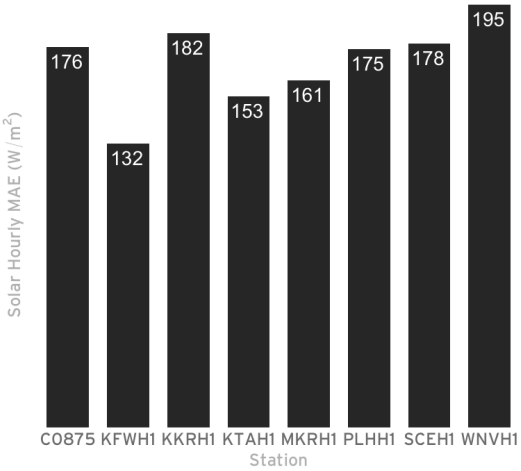


(c) Linear Regression

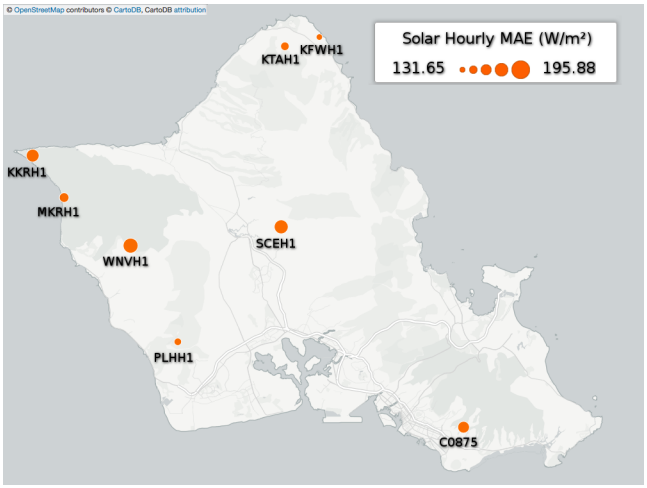


(d) Linear Regression

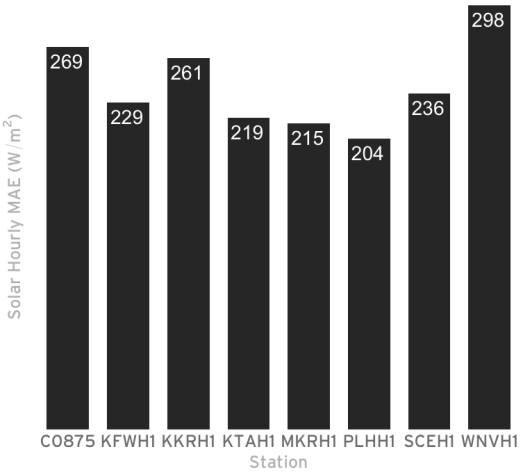
Figure 2.14: Forecasting Error on Different Sites. All models used only the previous day (w=2), training on 2012 and 2013 to forecast 2014.



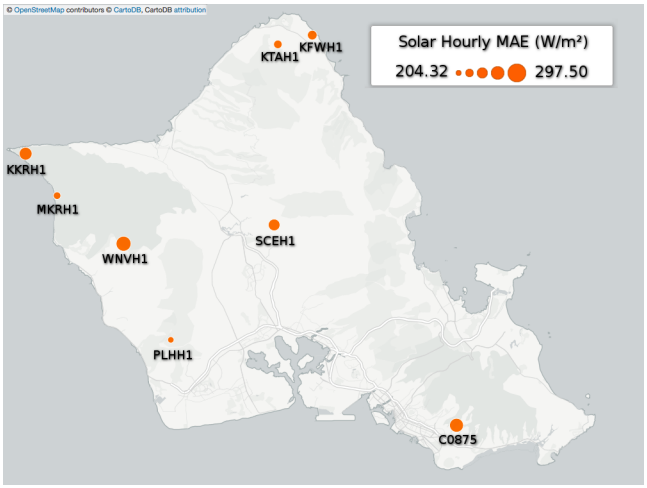
(a) K-Means Probability Model



(b) K-Means Probability Model

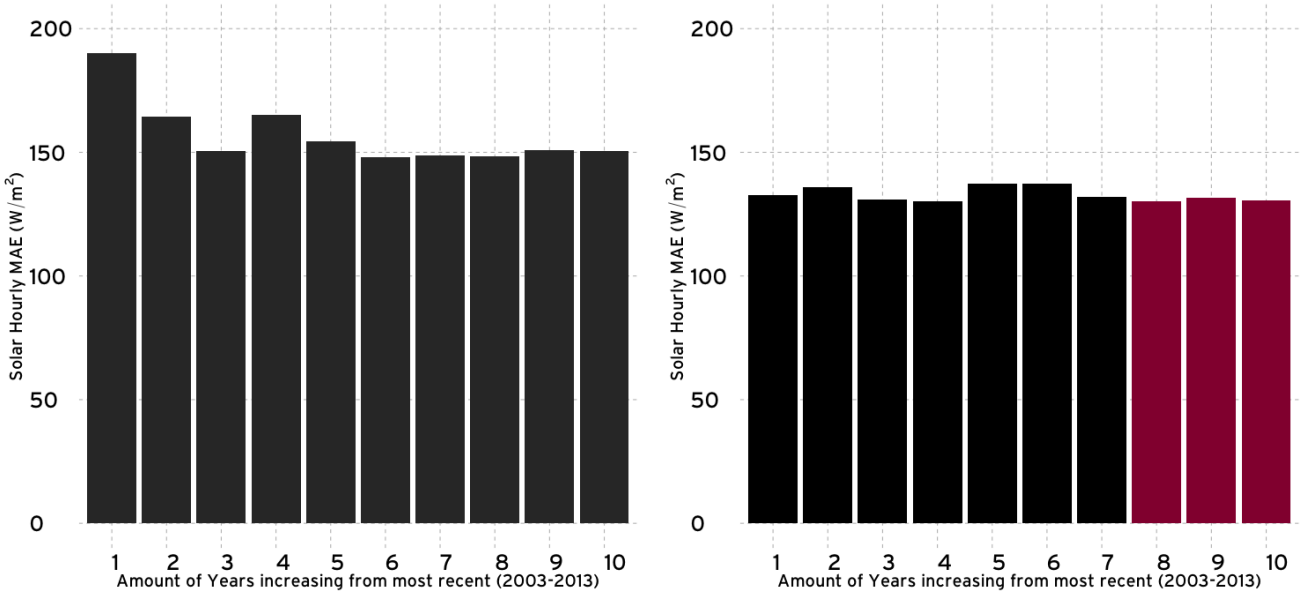


(c) Binning Probability Model



(d) Binning Probability Model

Figure 2.15: Probability Model Forecasting Error on Different Sites. All models used only the previous day (w=2), training on 2012 and 2013 to forecast 2014.



(a) KTAH1

(b) KFWH1

Figure 2.16: Solar Forecasting Hourly MAE to forecast 2014. Years are added from most recent to more distant years (e.g. 1 = 2013, 2 = 2013,2012, etc). For KFWH1, we observed a similar problem as SCBH1 on figure 2.3 for 2006 and 2005, and, therefore, the number of years for 8, 9 and 10 (colored in red) may be biased.

prediction function increased beyond half correct forecasts for the maximum number of years used as training data (52%), with mean = 48% and SD = 3% across all number of years. Again, the function also outperform the random (Mean=21%) and most frequent centroid (Mean=34%) prediction functions. On KTAH1, the results were worse, with the maximum number of correct forecasts dropping to 41% when all the data is used, and same clustering error contribution.

Conclusions. Despite the forecasting error varying slightly, we observed the probability model did benefit from increased number of data. For the forecasting error to remain similar on KFWH1 despite about 20% more correct forecasts, the clustering error must be compensating negatively the error. This reveals a trade-off faced on using the discrete probability models due to the estimated discrete probability distribution on the discretization step: More data translates to more tuples for the probability model to learn the transition from days, whereas it means more variance the centroids must represent, which increases the error. If we decrease the number of data instead, favoring the centroids accounting for less variability, then there will be less tuples observed by the probability model, which after a certain point will also lead to missing forecasts (since due to insufficient data, a certain combination of consecutive days of size w may never been observed on the train data). More missing forecasts then lead to higher error. This means both the parameters k and w are important on balancing this error trade-off.

2.5.4 How the amount of years relate to the model ability to forecast?

Experiment Set-up. Up to this point, the models only used the previous day, i.e. **w=2**. For probability models, the choice of **w** directly impact on the number of distinct tuples that will be counted (as **w** also indicates the number of elements of the tuple). Specifically, the number of potential max number of unique tuples (since not all values may occur in the data) is k^w, as each element of the tuple can potentially take a value between 1 and k, and **w** will define how many of them there are in the tuple. The same is true for the test dataset after being clustered. If both the training and test dataset now have a larger number of unique possible combinations, the chance they match will likely decrease, leading to missing forecasts.

Therefore, while considering that a model that take into account more days could be better than one that just consider the previous day, we must be careful not to increase the number of unique combinations to a point the kmPM model will not be able to find an exact match to the test tuple, and default to choosing the most frequent centroid of the training dataset due to missing forecasts. Adding more years, however, should help with this problem, as more tuples will be available. Understanding the relationship of more tuples versus considering more consecutive days **w** is the purpose of the next experiment. In this section we extend the previous experiment on the impact of the number of years on the forecasting error by adding in another variable, the number of consecutive days **w**. KTAH1 is used since it contains the most number of years.

Results and Conclusions. We start motivating this experiment with figure 2.17 which high-

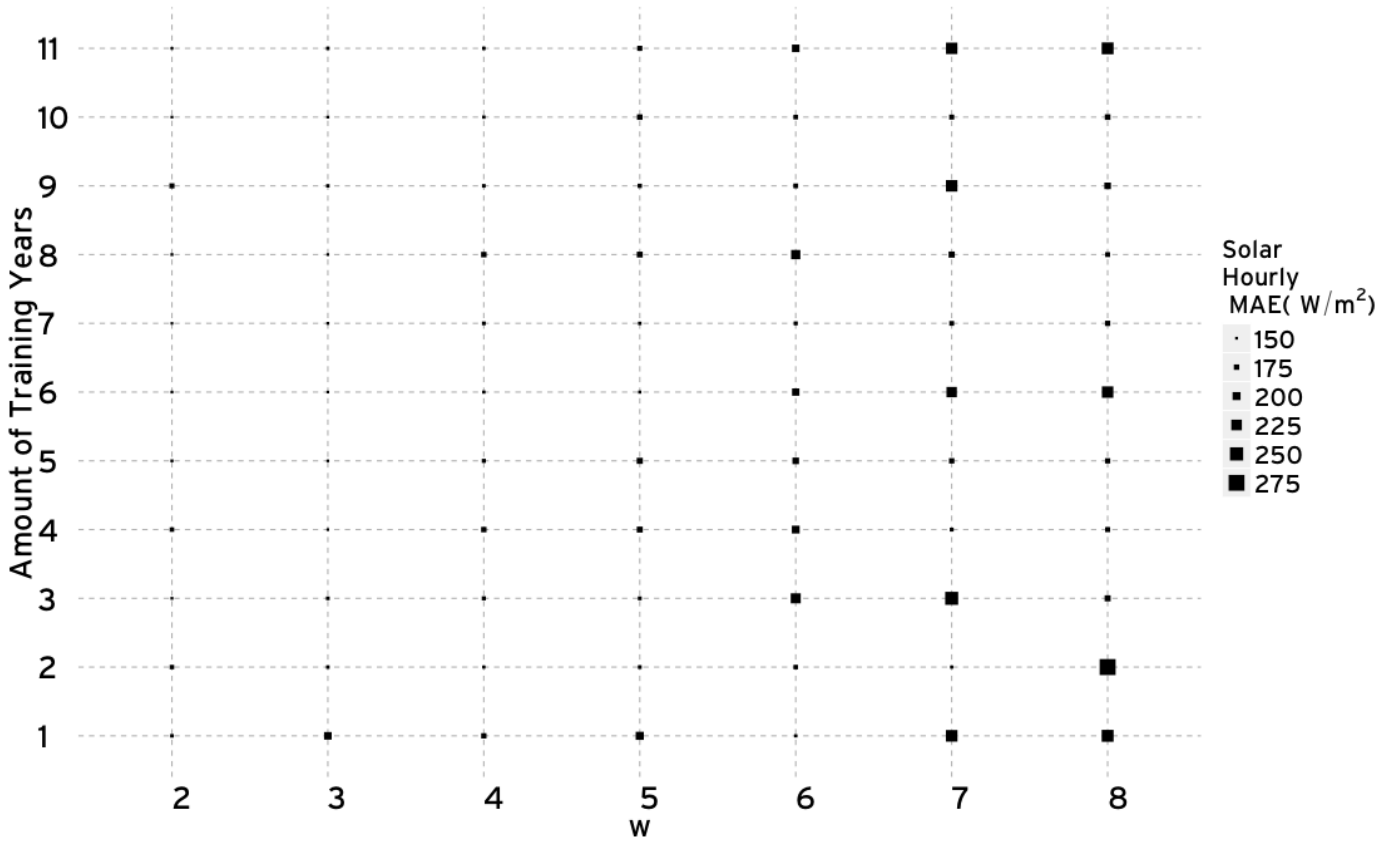


Figure 2.17: Relationship between number of years of KTAH1 versus the number of consecutive days. Each point on the plot correspond to a trained model resulted from the parameters of the Y and X axis. The size of the dot indicates the forecasting error.

light the relation of our interest. By observing this figure, we can already confirm that increasing the number of consecutive days **w** may lead to higher errors, even when compared to a smaller **w**. To be absolute sure this error relates also to missing forecasts, we can observe the same figure by replacing the error value by the missing number of forecasts days of 2014, as seen in Figure 2.18, which confirm the relationship.

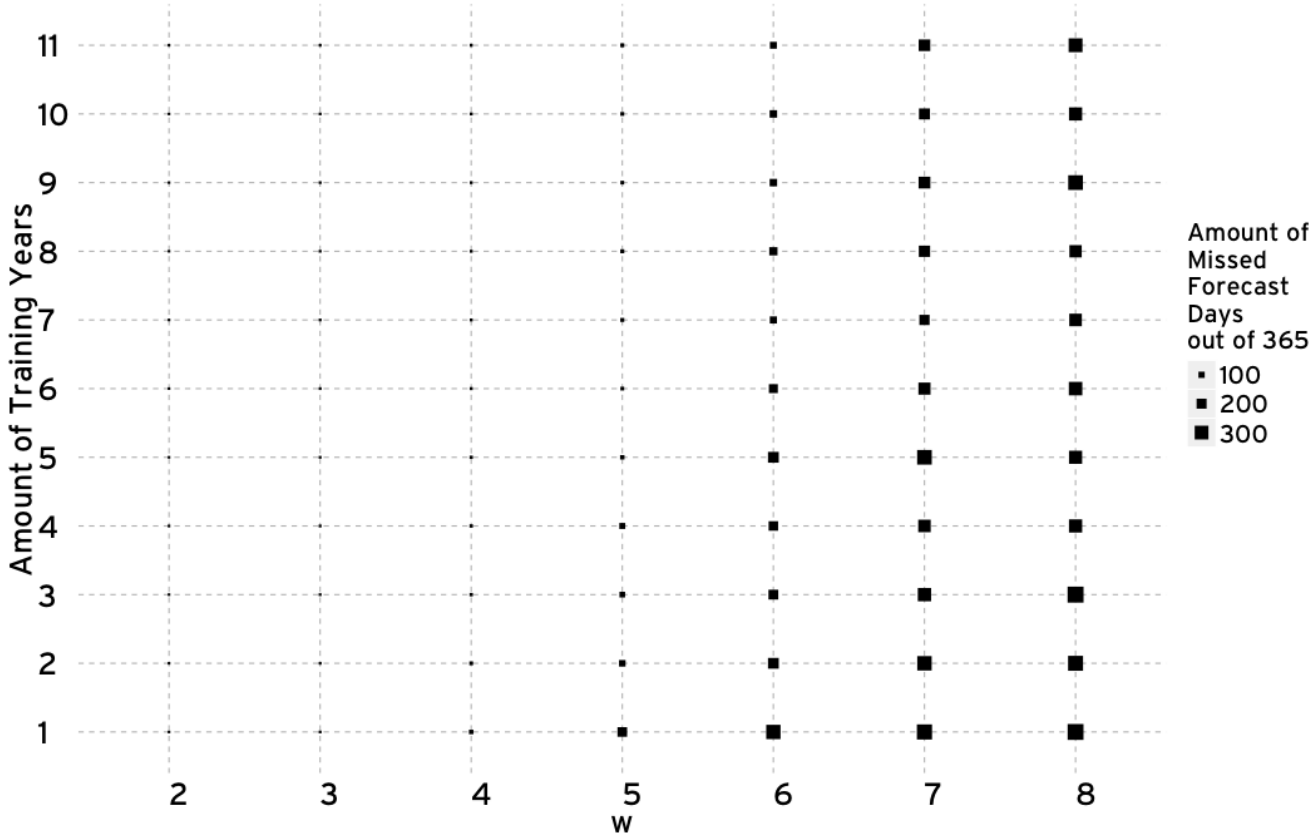


Figure 2.18: Relationship between number of years of KTAH1 versus number of consecutive days. Each point on the plot correspond to a trained model resulted from the parameters of the Y and X axis. **The size of a point indicates the number of days that the model could not forecast, and used the most frequent cluster of the training data instead.**

A follow-up question is to observe in more details the forecasting error fluctuation encoded on the smaller points, which is shown on Figure 2.19. Again, we can observe that the darker colors (higher values of **w**) are transitioning from bottom to top, as the points also increased in size from left to right on Figure 2.17. We can also observe there is an oscillating behavior of the forecasting error as more years are added. To observe the evolution of a single **w**, we re-color the same plot with the most distinct colors as shown on Figure 2.20.

If we focus on the "baseless triangles" on each line, we can quickly note that between 7 and 10

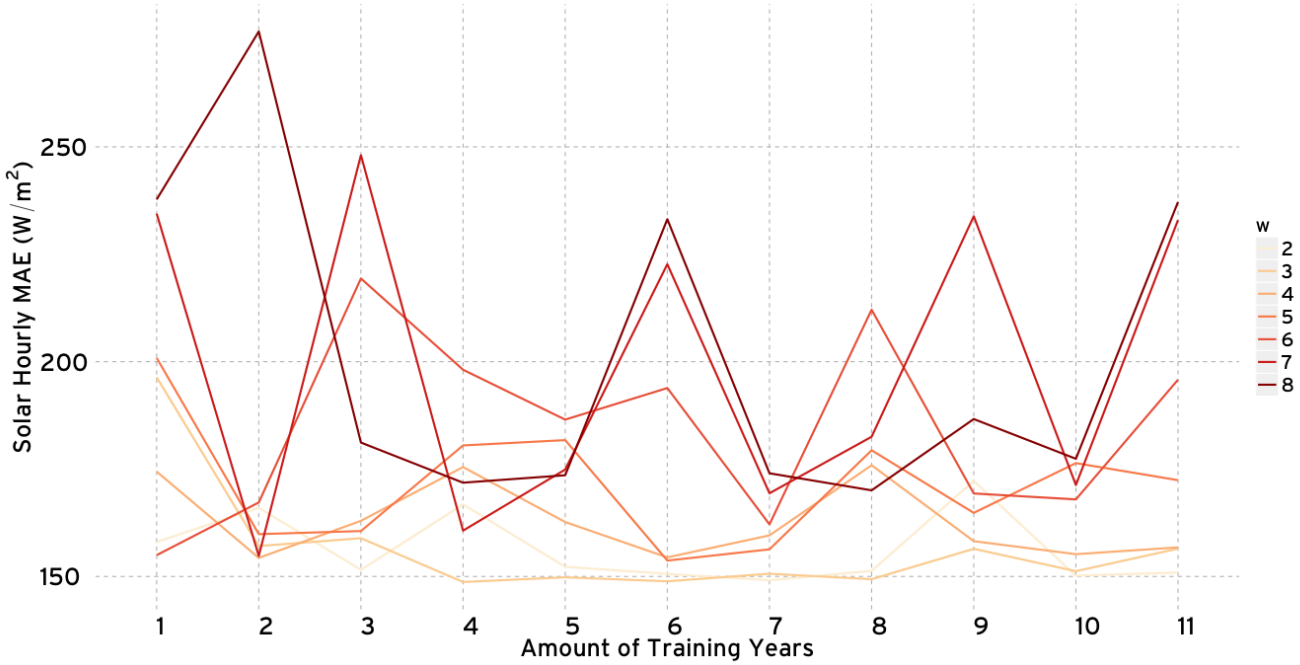


Figure 2.19: Relationship between number of years of KTAH1 versus the number of consecutive days. Each w from figure 2.17 is now represented by a line, and the transition observer from lower **w** values to higher **w** values is encoded on the color gradient (higher w's have darker colors). **It is important to note the y axis is a non-zero baseline so we can observe the small transitions in error.**

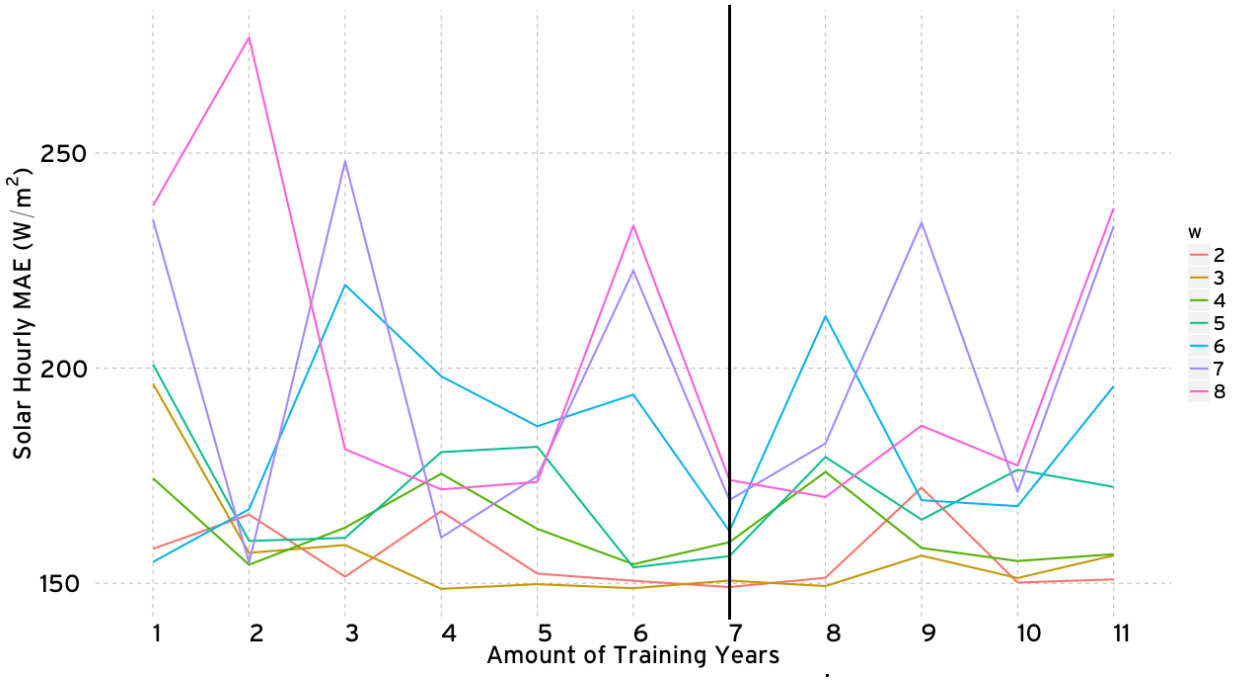


Figure 2.20: Figure 2.19 with distinct colors. We can more easily observe how each **w** line evolves as more years are added.

years of data the number of consecutive days all have a sudden increase and decrease of forecasting error, however there is a more clear distinction of these oscillations between years 1 and 7. Figures 2.21, 2.22 and 2.23 highlight the number of consecutive days that have similar forecasting error oscillations as more years are added.

Aside from the oscillation with 2 years, we can observe that as the **w** increases on the three figures, the oscillation occurs only when more data is provided. Before drawing further conclusions, it is interesting to observe how the missing data is associated to this same kind of visualization, as shown on Figure 2.24.

We can observe for **w=7** and **w=8** almost all the prediction is made solely using the most frequent centroid. The same is also true for **w=5** if less than 6 years is provided. Noteworthy, adding more years is particularly more meaningful for **w=5** and **w=4**, however comparing Figures 2.24, 2.21, 2.22 and 2.23 makes it clear lower **w** not only provide lower errors but more stable results (less error oscillation as more years are added).

We can therefore note that the shift of oscillations to higher number of years on higher **w** may be associated to the most frequent centroid being continuously used for forecasting, but this alone does not appear sufficiently convincing to justify the oscillation as more years are added. Another assumption would be that other weather variables may be influencing this oscillation, or other phenomena such as seasons, El Nino or La Nina. This will be the subject of the following Chapter.

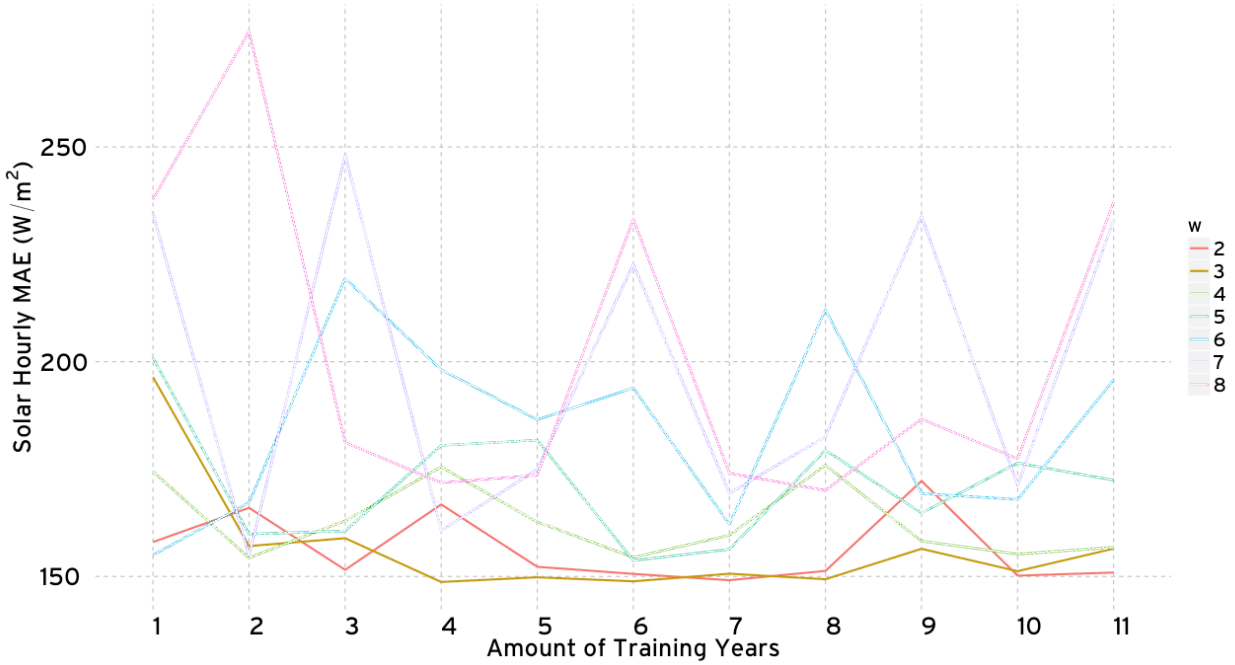


Figure 2.21: Figure 2.19 with distinct colors and w=2 and w=3 highlighted.

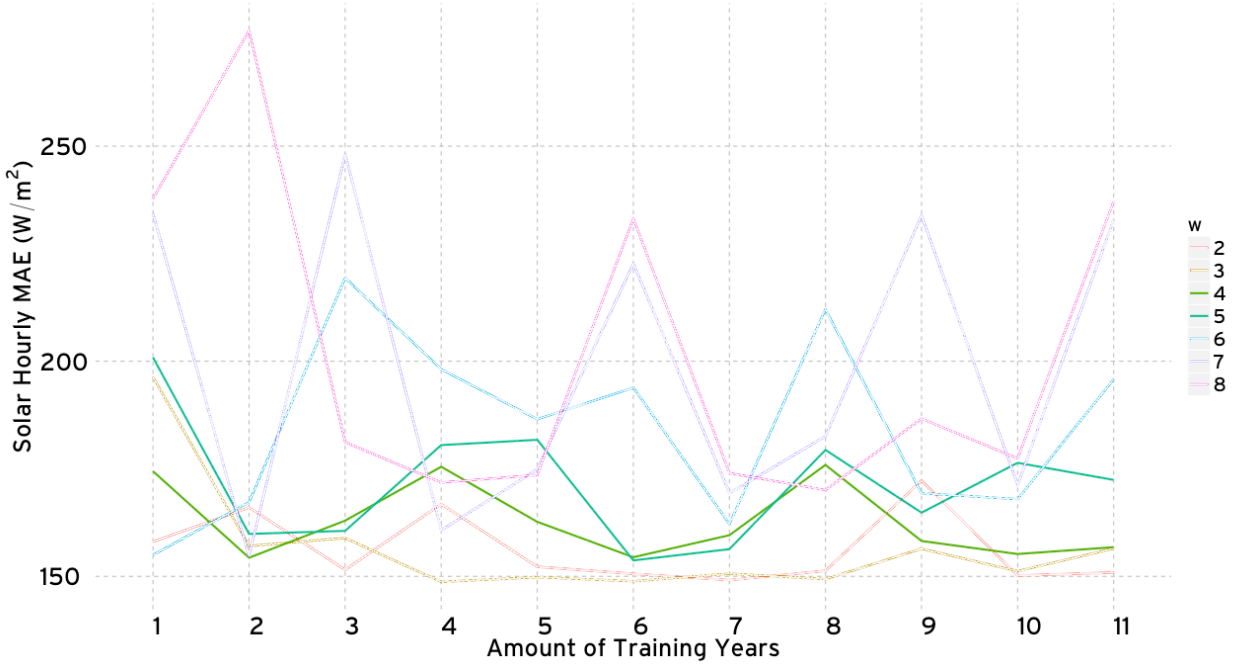


Figure 2.22: Figure 2.19 with distinct colors and w=4 and w=5 highlighted.

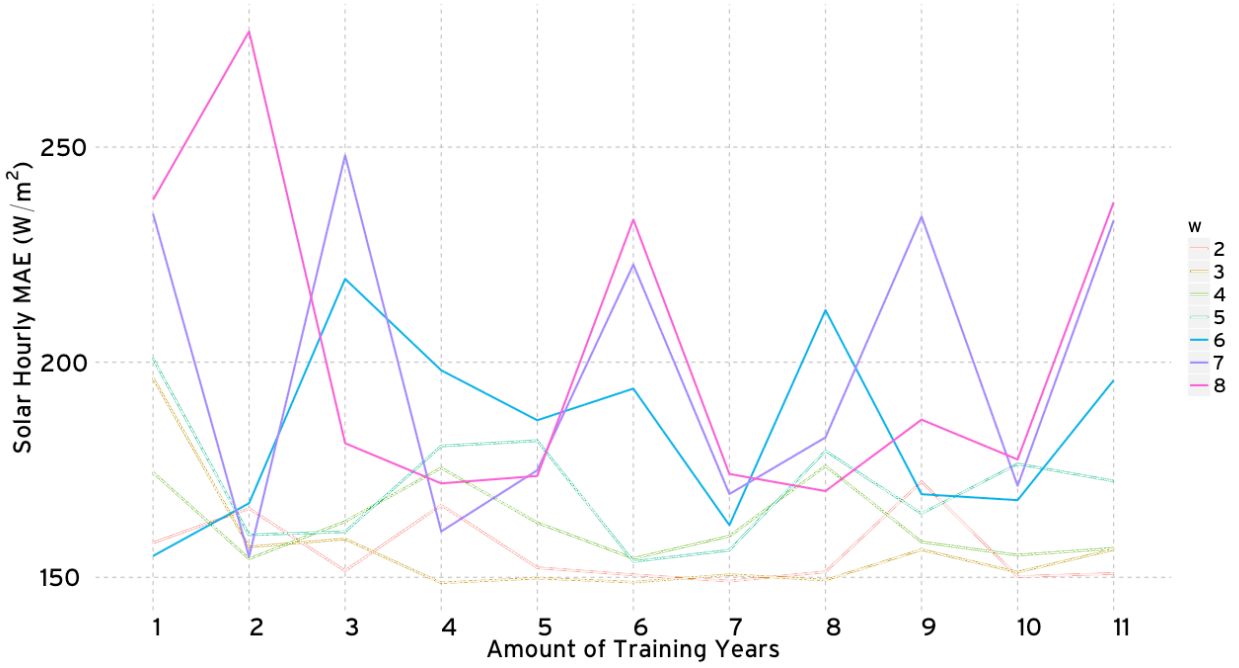


Figure 2.23: Figure 2.19 with distinct colors and w=6, w=7 and w=8 highlighted.

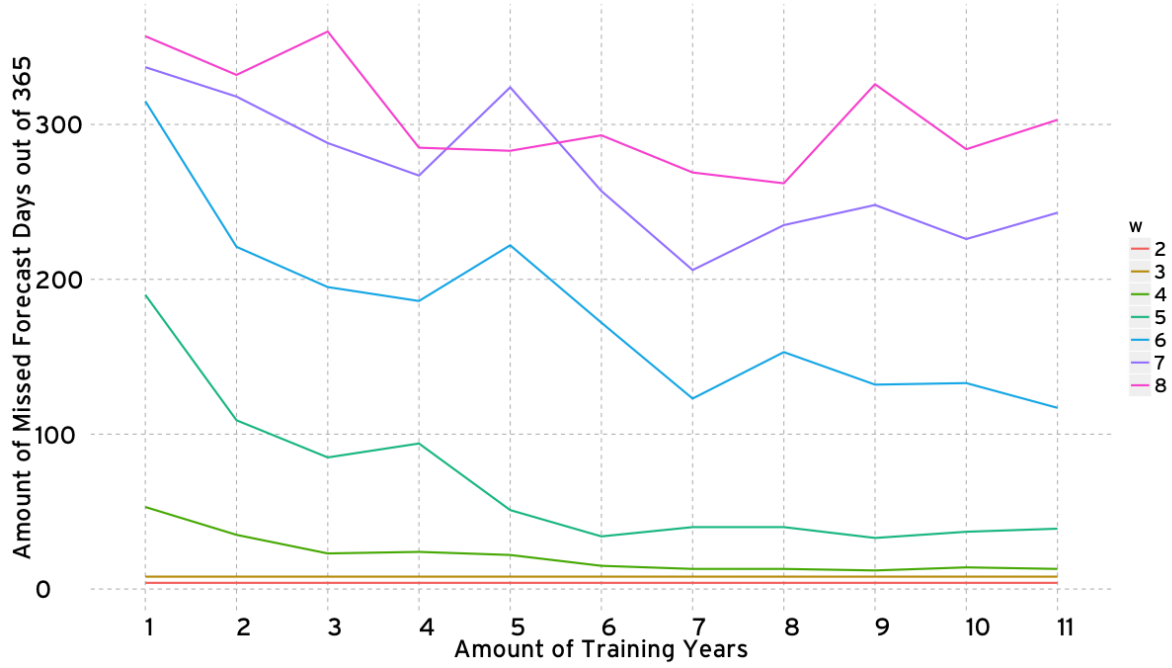


Figure 2.24: The number of missing forecasts on line plot representation. The error values (Y-axis) is replaced **by the missing days count.**

2.6 Conclusion

In this Chapter, we inquired the potential use of solar irradiation data freely available from sensors on Oahu island for solar forecasting using data mining methods, as an alternative to physical models. The usage of data mining models, however, require several decisions to be made *a priori*, which may influence the final solar forecasting performance, such as the number of clusters or bins **k**, the number of consecutive days **w**, the choice of the data mining method, and the number of years that should be provided. We concluded that simpler K-Means Probability Models had better performance than other methods (LR,binPM,kmNB), and also require less number of years of data to obtain the best forecasting error. This forecasting error was observed consistent with the number of solar irradiation standard deviation across different sites. We also decoupled the forecasting error to observe the clustering contribution and cluster id prediction. We noted the clustering contribution error accounted for over half of the forecasting error (60%), and the probability model had a improvement of 20% on correct forecasts when 10 years of data were used. However, the final *hourly* forecasting error remained with small differences, likely due to more years introducing more variability on the centroids, and therefore increasing the error despite the higher number of correct forecasts. Finally, the interplay of the consecutive number of days, provided number of years to the model, and resulting number of unique tuple combinations on missing forecasts also exhibit an oscillation behavior as more days and years are added, which may suggest influence of other weather variables or phenomena, serving as motivation for the following Chapter investigation.

CHAPTER 3
WEATHER VARIABLE EFFECT ON SOLAR FORECASTING

3.1 Introduction

Insofar we have only considered solar sensor data. However, other weather variables, such as relative humidity, pressure, precipitation, etc. can also be influenced by the weather and relate to fluctuations in solar irradiance. We investigate here, if the probability models can benefit from using this data with and without combining with solar sensor data to lower forecasting error. We can also consider time partitioning this data when training the models in respect to other phenomena such as seasons, el nino, and la nina. We investigate the effect of weather variables and time partitioning in our probability models in this chapter.

This Chapter is divided as follows: We will first explain how we extended the training and test-preprocessing described in the previous chapterto estimate discrete probability distributions of multiple weather variables in section 3.2. We will then discuss specific cases of this extension to account for time partition, and the wind direction weather variable, whose values have a different meaning than the other weather variables and consequently require a different clustering step to combine with wind speed.

The experiments of this section will then investigate if other weather variables are better individually (3.3.1) or interacting with solar irradiance for solar irradiation forecasting (3.3.2). We conclude the Chapter by investigating the effect of time partitioning (seasons and el nino and la nina) on experiment 3.3.3 and the presentation of our conclusions.

3.2 Training and Test Pre-processing Extension to Weather Variables

Let us now consider a slight different and more generalized scheme that allow us to estimate discrete joint probability distributions of any number of weather variables, as shown on Figure 3.1.

- **Step (1) - Weather Variable Selection:** Recall from the original pipeline each station contain several sensors. All the collected sensor data is shown in Table (a), each represented by a different column. Since we take into account now other sensors data beyond solar, we emphasize in our example two of the weather variables instead of only solar. The new column in purple here is associated to Relative Humidity (%). Again, using step (1) we select these two columns, aggregating from minute to hour samples and representing them as columns **in separate tables**. The sequence of steps 1 is the same and applied individually to Table (a), hence the steps (1a) and (1b) associated to each table.

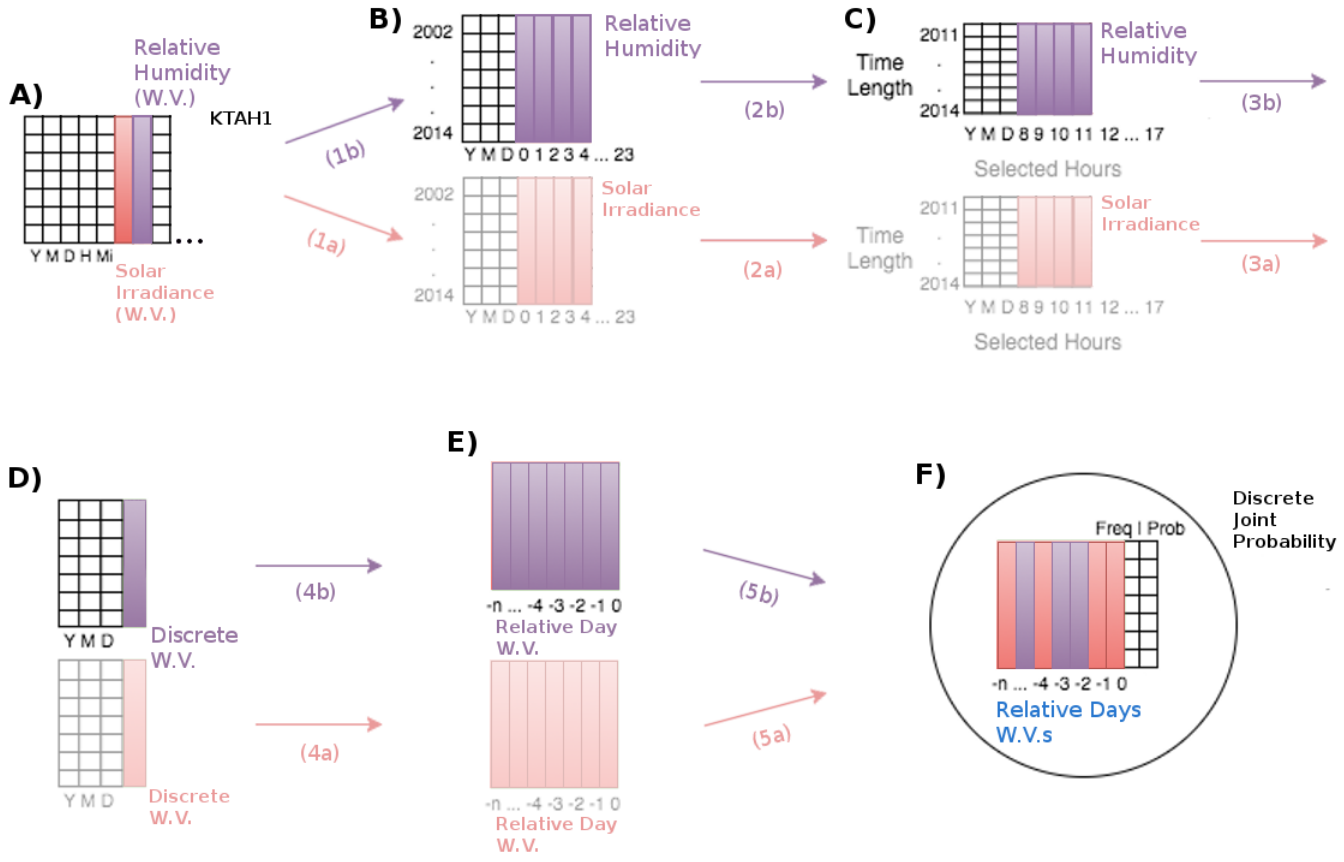


Figure 3.1: General Pipeline for any Weather Variable used on this Chapter.

- **Step (2) - Data Filtering, Step (3) - Discretization and Step (4) - Tuple Extraction:** The previous steps (2) and (4) happens the same way as previously described separately from each other from these two tables, resulting on tables of the form in (d), like the same previous pipeline. Since extracting tuples is equivalent to observe the data as **relative days** instead of the timestamp, step (3) will also guarantee the timestamps are aligned before extracting the tuples. All Tables (d) therefore will contain the same number of rows, for the entire time length chosen on step (2), the filtering step. Another required modification occurs in Step (4) to guarantee the alignment of timestamps, by specifying all tables to be generated using the highest number of consecutive days **w**. We will use an example shortly to illustrate the alignment.

- **Step (5) - Discrete Probability Distribution Estimation:** We need at this point to consider how to "merge" the different Tables (e). Since step (3) guarantee all the timestamps are aligned, we can just select the chosen relative days to train the model from the different Tables and set them "side-by-side" in a single table. Rows that contain dummy data are then filtered, guaranteeing the extracted tuples only contain consecutive days.

With the new pipeline, we can now consider a more general scheme to arrange the previous days in any order. Instead of creating models that use the previous 1 or more consecutive days, we could use a model which uses 2 days before to forecast, or a combination of the third day and the first. We can also consider a model who uses the relative humidity 2 days ago, while using the solar irradiation of 1 day ago.

Alignment example. In order to allow the model for any number of weather variables and relative days, step (4) will apply a different sliding window for every weather variable Table (d). In our example, two sliding windows (two **w** values) would be used since we have a table for solar irradiance cluster ids, and another for relative humidity cluster ids. If we would choose the model to have the relative humidity of the previous 2 days, and the solar irradiation of 1 day before, this would be equivalent of **w=2** and **w=1** respectively. Since we have 2 discrete weather variable table's, namely solar and temperature on our example, we will have 2 centroids tables associated to each of them. To guarantee alignment between all relative days of Tables (e), the highest value of **w** is used to specify all window sizes. If this would not be done, then 1 day ago for relative humidity (**w=2**) would not correspond to 1 day ago for solar irradiation (**w=1**).

Different clustering k's. The clustering step (3) of the *training tables* will cluster separately each table, and may have a different **k** for each table, and apply either k-means or binning to generate their respective cluster id columns on Table (d). A third pipeline is also possible to estimate probability distributions for multiple stations simultaneously for cross-site forecasting (i.e. train the model in one station, and forecast at another station), which will be discussed on the following Chapter.

In essence, Chapter 2 method to estimate probability distributions are equivalent to posing questions such as *"What will be the solar irradiation today, having observed 1 (or more) days before?"*. The more general procedure just described in turn allow for more general questions such as *"what is the solar irradiation today, having observed the relative humidity (RELH) 2 and 1 day before, the temperature (TMPF) 3 days before and the solar irradiation (SOLR) 1 days before?"*. Notice also that the discussion of maximum number of consecutive days **w** of the previous Chapter applies here, as every pair of weather variable and relative day will result in one element on the tuples extracted in the pipeline at Table (d) of Figure 3.1, impacting on the maximum number of combinations a tuple can have, leading to missing forecasts. Due to this, we delimit the discussion of the generalized model to tuples of maximum size 3.

We have described how to estimate discrete probability distributions with any number of weather variables, but we have yet to mention how the new general pipeline can also account for wind direction, seasons, el nino and la nina. This will be discussed on the next experiment sections.

3.3 Experiments

Our main interest on this experiment section is to investigate how weather variables, season, and el nino can affect solar irradiance forecasting. The first experiment will investigate the weather variables effect on forecasting of different sites. The second experiment will then observe how solar irradiation and other weather variables interact for solar forecasting. The third and last experiment will then investigate the impact of seasons, and el nino on all the previous weather variable models.

3.3.1 How other weather variables models perform when compared to solar forecasting models based on only solar data?

Set-up. We will first consider **w=2** km-Probability models using the previous day, i.e., *what is the solar irradiance today, given the previous day weather variable?*. The solar irradiance weather variable was discussed on the previous Chapter on the recent years experiment 2.5.2. We will replicate the experiment using the other weather variables: Temperature (Fahrenheit), Wind Speed and Direction (mph,degrees) Relative Humidity (%), Precipitation (inches) and, Pressure (inches).

Wind Vectors: Combining Wind Speed and Direction sensors. One of our weather variables, wind speed, is naturally a vector, and therefore must be used as a single weather variable with wind direction. In order to do so, we can observe wind direction also as a weather variable column on Figure 3.1 on (a), whose values range from 0 to 360 degrees. The only difference on the pipeline to this weather variable compared to the others occurs on step (3). Since we have the wind direction in degrees in *hours*, we average this value to obtain the average wind degree at a given day. Afterward, all the averaged degrees per day are binned on 16 bins of equal range (i.e. starting on 0 degrees and incrementing by 22.5 degrees each), to represent the wind-rose. Since this is the

only difference on the pipeline, we can consider the wind direction (DRCT) as being an element of a tuple, and as an extension of the questions which can be made with the model.

Results. Figure 3.2 highlights the previous Chapter experiment versus the new weather variable models. The blue bar indicates the forecasting error observed on the previous Chapter in the first experiment for km-PM. Each other grey bar is the error for the other models. Here we emphasize the previous model error for overall comparison. The following Figure **??** colors the bars differently for other comparisons.

We can see that the blue bar for solar irradiation is lower than the others, indicating solar is still a better model than using the other weather variables without solar irradiance to forecast, aside from C0875. C0875 case is most likely due to the low number of data provided for solar irradiation (2013 and 2014) as we mentioned on the last Chapter Table 2.1.

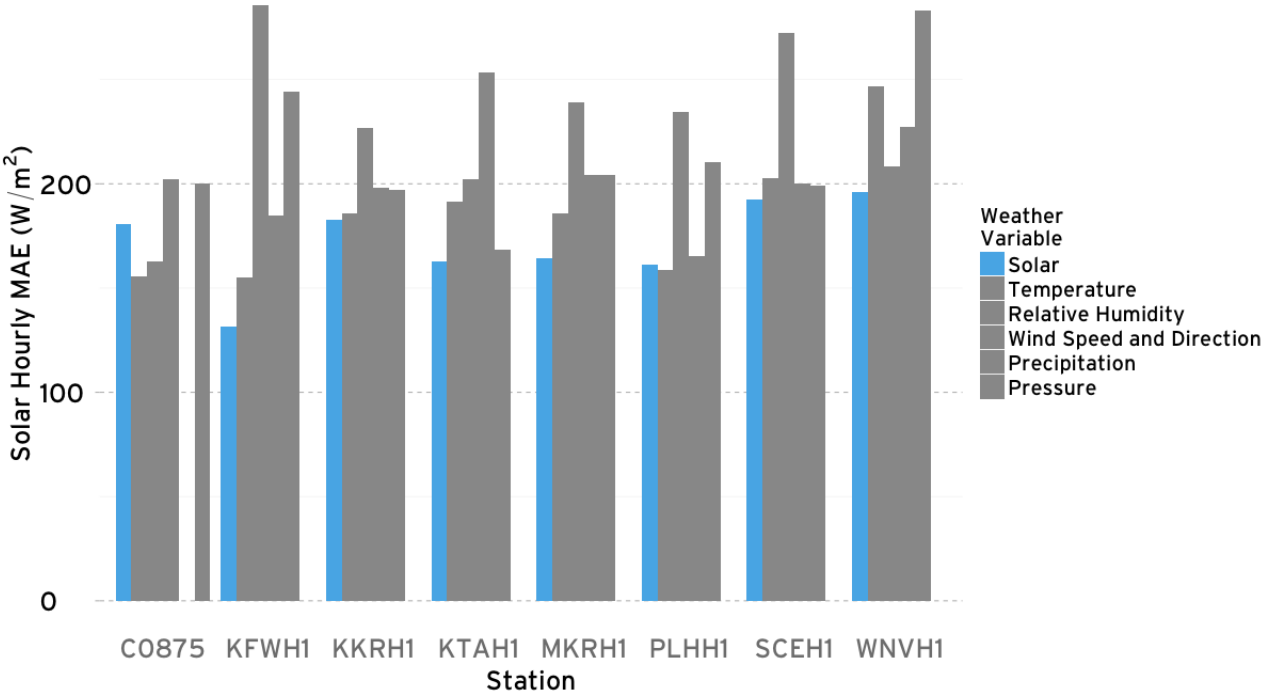


Figure 3.2: Solar irradiation model versus weather variable models. For C0875, the precipitation weather variable is not available.

It is also noteworthy to consider the number of missing forecasts of each model, as shown on Figure 3.3. We can observe that wind direction and speed models have the highest number of missing days. This may be due to the 16 direction partition used for the direction. We attempted using a lower number of directions, but this resulted on larger errors. This situation supports our decision to keep **w**, the number of consecutive days, reasonable low to observe the weather variables when they interact on the next experiment, since larger combinations would mostly lead

to inconclusive results due to missing forecasts. Recall missing forecasts are replaced by the most frequent centroid observed on the training data as a default.

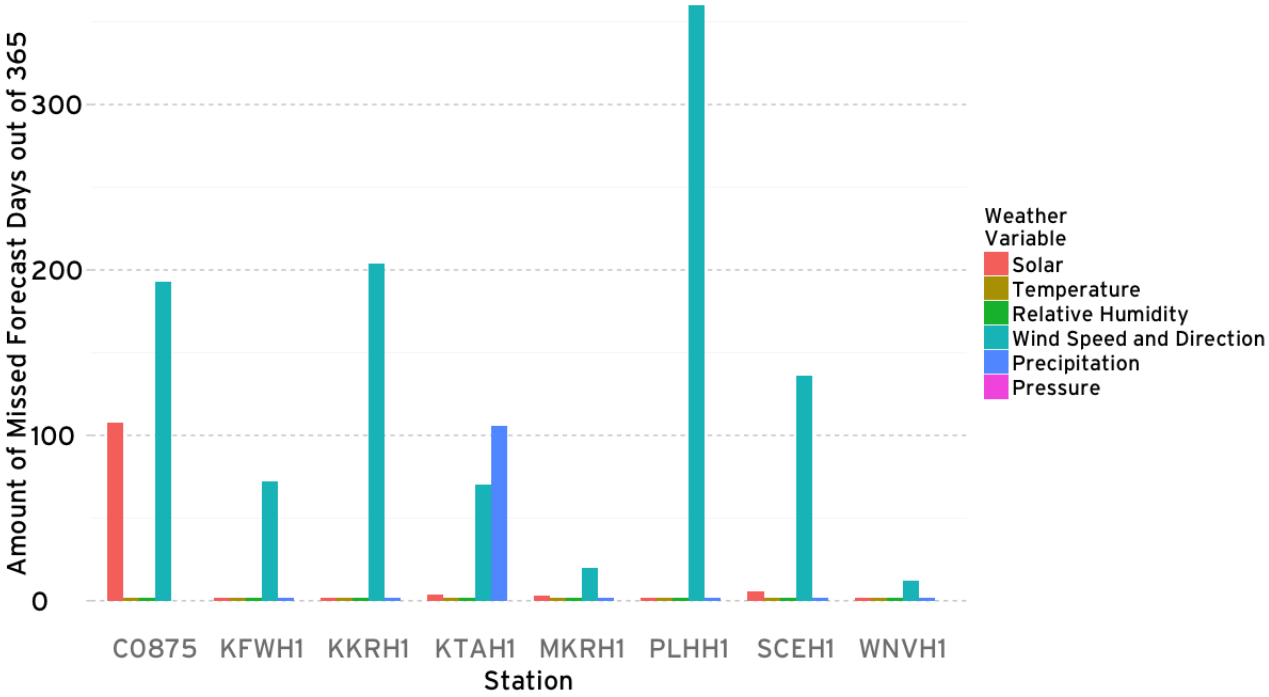


Figure 3.3: number of missing forecasts of 2014 using all weather variable models. For C0875, the precipitation weather variable is not available.

Conclusions. We conclude the model trained using only solar irradiance is still the lowest forecasting error model. We will investigate next if introducing the solar irradiance weather variable on each of the models presented in this section can improve the solar forecasting error.

3.3.2 How combining other weather variables with past solar data affect solar forecasting?

Set-up. In this section, our model represents the following question: *what is the solar irradiance today, given the previous day weather variable* ***and the previous day solar irradiation****?*. Therefore representing models up to **w=3**. We also added one extra question for comparison, which is *what is the solar irradiance today, given the previous first and second days solar irradiation?*.

Results. Figure 3.4 displays the observed forecasted error, highlighting the previous chapter solar irradiation errors.

We observe again that the model trained with only solar irradiance data is still superior or have insignificant error difference compared to the remaining models. Figure 3.5a associate a distinct color for each model for more detailed comparison. In this case, there is not much error difference

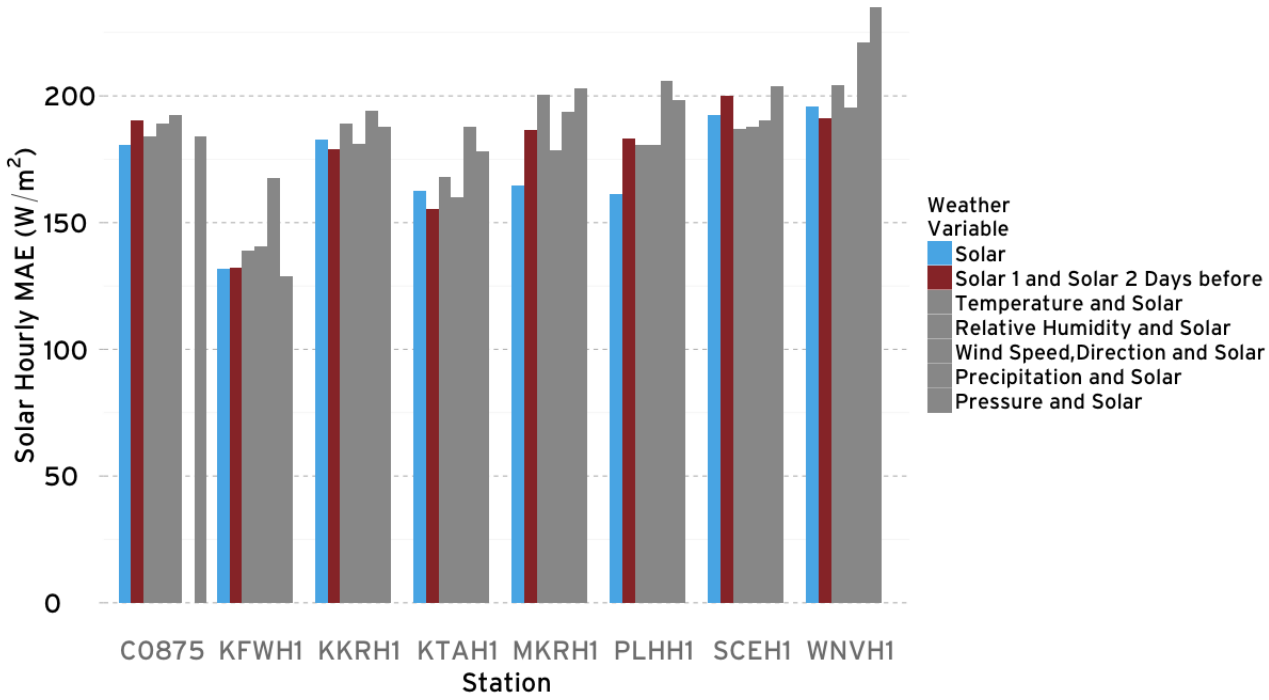


Figure 3.4: Solar Forecasting error of weather variable interaction models. Solar models are emphasized for comparison. For C0875, the precipitation weather variable is not available.

from the combined weather variable models. Finally we note on Figure **??** that the missing forecasts are much more visible due to a higher number of weather variables. They are particular high for some stations due to missing data on the training years. It is important to observe also that due to the nature of the probability model, and our criteria to only consider tuples without missing data, the final discrete probability distribution table will have potentially many more missing tuples than if the models were constructed separately.

Conclusions. In concluding this second experiment, we note that the use of weather variables insofar have not justified any benefit in comparison to the previous chapter results, indicating a solar irradiance alone is still better for the presented data mining probability model, and also taking into account the problems presented on the data. However, there may still be chance that if the model take into account seasons and el nino (both of which are time partitions), it may obtain better performance than solar irradiation alone. The next experiment will re-evaluate all models presented with and without time partitioning.

3.3.3 How are models in the previous sections influenced by external factors such as Seasons, El Nino and La Nina?

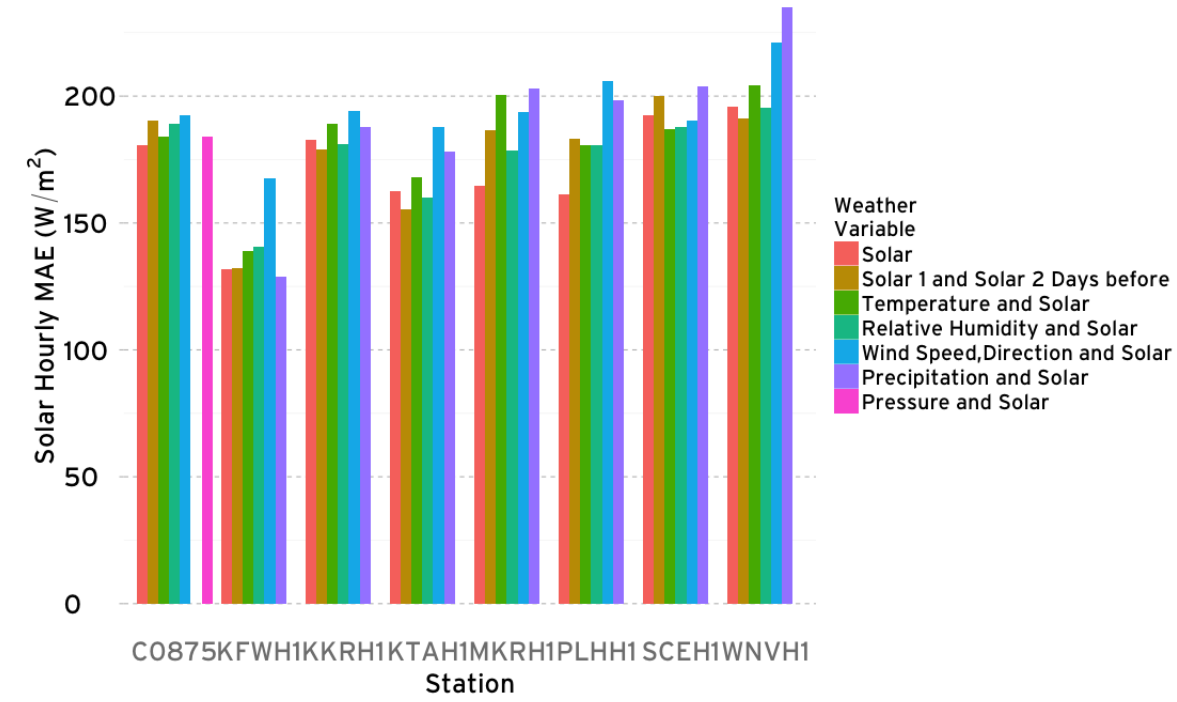
Set-up. Since the seasons partition the current dataset in either two or three set of tuples, leading to a higher number of missing forecast, we decided on using KTAH1 from 2003 to 2013 to train the model, and 2014 as test to verify this experiment. We created a new set of models with partition for el nino, season and without any time partition for all the models of the previous two experiments (i.e. without interactions and with interactions) to observe if any significant improvement on the forecast occurs. Before we explain the results, we will describe how we performed the time partition.

Time Partition: Seasons and El Nino and La Nina

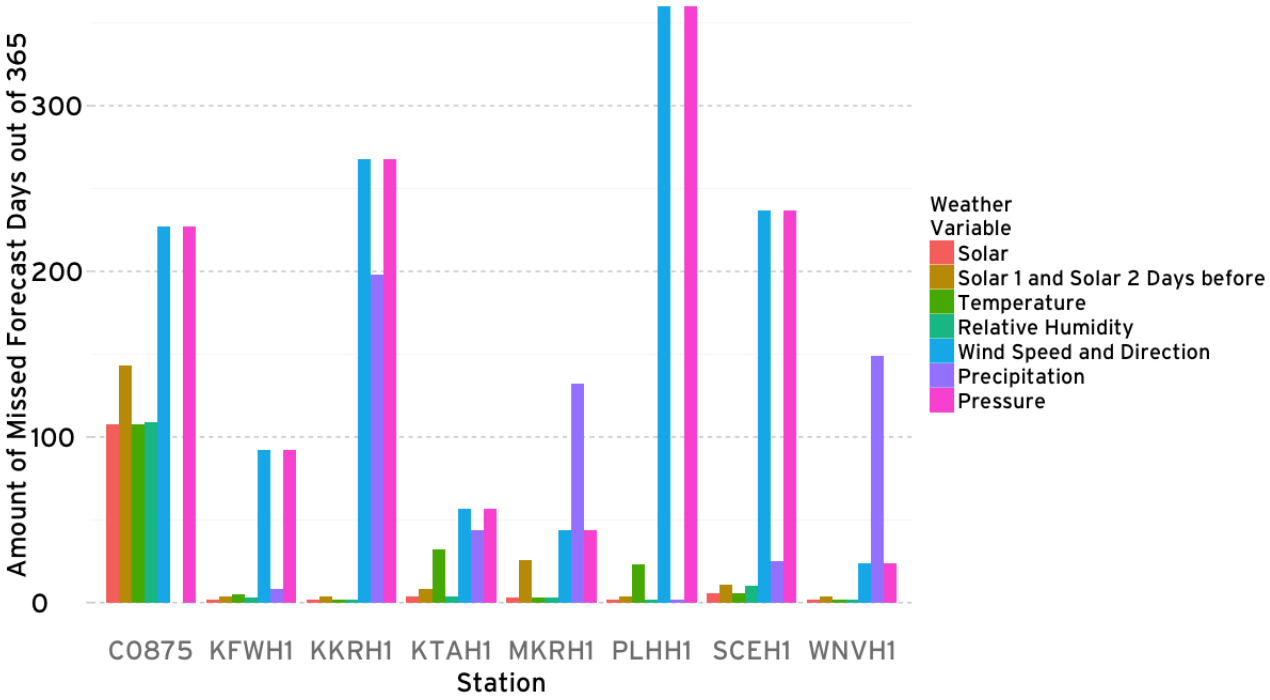
Seasons are one of the most visible characteristic of the solar irradiation observed on the start of Chapter 2 calendar plot, on Figure 2.3. The first and last 3 months of Hawaii at any given year are distinct from the 6 months in between, having lower number of solar irradiation, which characterizes Hawaii's 2 seasons, Summer and Winter. We may therefore be able to capture some of the variability due to seasonality by partitioning the data before estimating the probability distribution, and the kind of questions we can make using the probability model taking into account seasons.

Season Time Partition using the Pipeline. Season is a time partition, i.e., it is defined by the months. Let's consider again Table (a) in Figure 3.1. Months are defined in the second column, from left to right.

- **Step (1) - Weather Variable Selection:** Here, instead of selecting one of the usual weather variables, we select the month column for time partitioning.



(a) Solar Forecasting error of weather variable interaction models. For C0875, the precipitation weather variable is not available.



(b) number of missing forecasts of 2014 using all interactions weather variable models. For C0875, the precipitation weather variable is not available.

Figure 3.5: Stations on Oahu Island. Each bubble is sized according to how many days were sampled by one or more station's sensors between January 2002 and October 2015. No data at any station is available before this time range. The October 2015 upper boundary is the last day we acquired the data.

- **Step (2) - Data Filtering, Step (3) - Discretization:** These steps are skipped entirely, since the column of months are already "discretized", as months, and close to the form of Table (d). Here, we introduce a new column that serves as the discrete weather variable, namely the kind of season. Since in Hawaii we only have Summer and Winter, characterized respectively from April to September for Summer, and October to March for Winter, each month is labeled as "Summer" or "Winter" in the discrete weather variable column. Notice this means that for every day of a given month we will have the discrete weather variable assigned as either "Summer" or "Winter", since the granularity is still at the day level.

- **Step (4) - Tuple Extraction, Step (5) - Discrete Probability Distribution:** The remaining steps are the same as before: We guarantee alignment of the timestamps for the other Tables (e), and select the columns of the relative days we are interested to estimate the discrete probability distribution.

Our estimate probability distribution can now be used to make questions such as: *Given the season of 2 days before, and the solar irradiation 2 days ago what is the solar irradiance today?*. Although seasons for every day do not change as frequently as the discrete weather variables previously defined, and may impact less deciding wether we want the previous 1 day or n few days, using any one previous day will result on the extracted tuples contain one element indicating the season. Since we count repeated tuples, the estimated probability distribution will function as a season *partition*.

El Nino, La Nina, and Normal Months Time Partition using the Pipeline. For El Nino, La Nina and Normal months the process is similar. The only difference occurs at Step (3) - Discretization, on how we construct the discrete weather variable and assign the labels "El Nino", "La Nina" and "Normal" month instead of "Summer" and "Winter". In order to do so, we refer to NOAA's ENSO Table [1]. A month at any given year can be observed on the referred table as being of the color blue (La Nina), red (El Nino), or uncolored (Normal month). We use this same categorization to assign the labels for every day on Step (3), and repeat steps (4) and (5) with the el nino, la nina and normal month labels to categorize a day as being under the influence of one of the two phenomena. Again, although the phenomena will not change as frequent every day, using at least one prior day will result on the extracted tuple form a partition over the phenomena when the probabilities are estimated to train the model.

Results and Conclusions. The results are shown on Figure 3.6.Overall, season partitioning provided better models than without any time partitioning across all the models, decreasing the observed forecasting error from the previous experiments (Solar and Solar -1 and -2 Days are the same as highlighted on KTAH1 on Figure 3.4). We can also note compared to Figure 3.4 that introducing time partitioning led to improvement of the combined weather variable models. In

[1] `http://www.cpc.ncep.noaa.gov/products/analysis_monitoring/ensostuff/ensoyears.shtml`

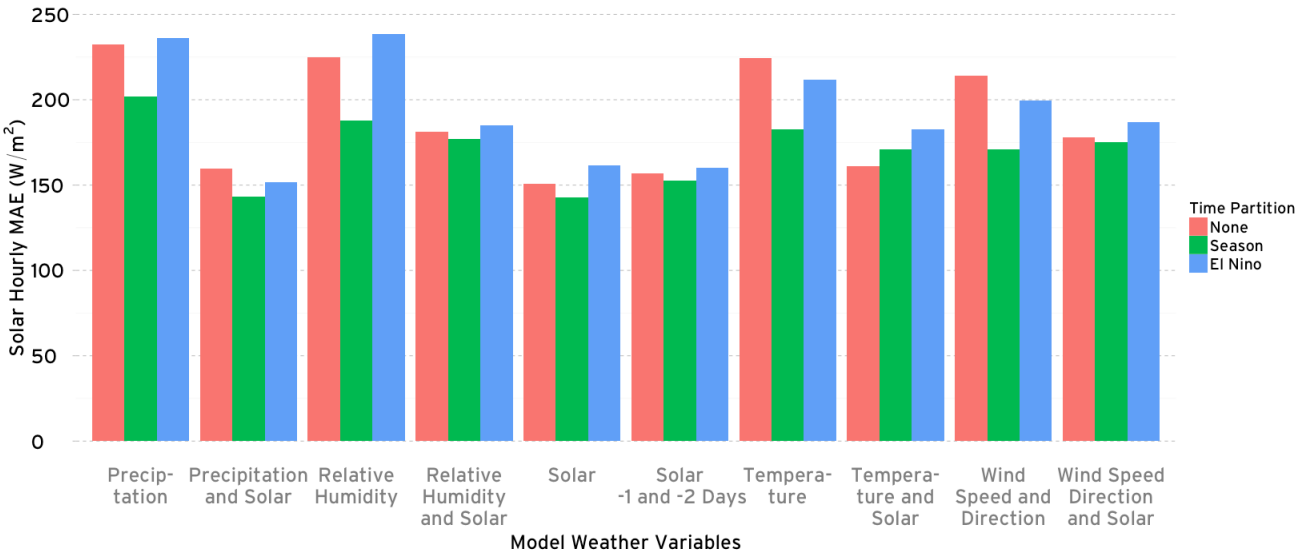


Figure 3.6: Season and El Nino effects on solar forecasting.

fact, the model of precipitation and solar is now equivalent in performance to using only solar irradiation.

3.4 Conclusion

We extended in this Chapter our K-Means probability model to account for other weather variables, including wind direction, and time partitioning, i.e., season and el nino, and la nina. Our first two experiments suggested solar models without using weather variables were preferable for forecasting, however the last experiment indicated that with time partitioning weather variables can also justify their use. We also observed during the second experiment that despite two stations being located closely and with similar wind direction, a difference in wind speed significantly changes the forecasting error of the same model. It seems sensible then to investigate the effect of spatiality on our probability models, which is the subject of the next Chapter.

CHAPTER 4
SPATIAL EFFECT ON SOLAR FORECASTING

4.1 Introduction

In this Chapter, our goal is to investigate if training our K-Means probability models with weather variables from a different site, instead of the same site as in the previous Chapters, can decrease the solar forecasting error. We can define more broadly the definition of our probability models as follows, now explicit parameterizing the station for each weather variable (WV):

$$WV_{k=Solar,t,Station=j}|WV_{k,t-i_1,Station=j}, WV_{k,t-i_2,Station=j}, \dots$$

where k is the chosen weather variable (e.g. temperature), $1 \leq i_1, i_2, \dots \leq n \in N$ are the chosen relative days to train the model (e.g. the day before, 3 days before), and j a given station. For the previous models, the station was fixed for all weather variables used, since we trained a model to forecast the same site.

An alternative to this set-up is to train the model using a different site's weather variables from the one we intend to forecast, i.e., **using a cross-site forecasting set-up**. There are two variants for cross-site forecasting set-up when training the models: In the first variant, we use the solar irradiance from the intended site and all weather variables but solar irradiance from the different site to forecast the intended site solar irradiance. For instance, suppose the intended site's station to forecast is KTAH1, and the different site is C0875, the model would be:

$$WV_{k=Solar,t,Station=KTAH1}|WV_{k,t_1,Station=C0875}, WV_{k,t_2,Station=C0875}, \dots$$

In the second variant, we use all weather variables from a different site (including solar irradiance instead of the intended site to forecast) to forecast the intended site solar irradiance. Using the same example, we would have the model as follows:

$$WV_{k=Solar,t,Station=C0875}|WV_{k,t_1,Station=C0875}, WV_{k,t_2,Station=C0875}, \dots$$

The former will have the model attempt to identify patterns between the intended site solar irradiance and the different site weather variables in the past to forecast the intended site solar irradiance. The later set-up will have the model attempt to identify patterns between solar and weather variables patterns in the past of the different site, to forecast the intended site solar irradiance. In the later case, the assumption is that training a model to forecast a different site will also make the model capable to forecast in the intended site due to external factors (e.g. cloud motion from the different site to the intended site).

We pose therefore the following two questions for our experiments:

- What are the effects in solar forecasting error using neighbor stations weather variables not including solar?
- What are the effects in solar forecasting error using neighbor stations weather variables **including** solar?

We will explain how the probability model is modified to enable cross-site forecasting in Section 4.2. Sections 4.3.1 and 4.3.2 answers the first and second questions.

4.2 Training and Test Pre-processing Extension to Cross-Site Forecasting

4.2.1 Cross-site Training with intended Station Solar Irradiance

Figure 4.1 shows the modified pipeline. We observe (a) now contains a new table from a different station, C0875, whose weather variable of interest is temperature. Steps 1 to 5 are performed on the same manner as before, since already from step (1c) the column representation on (b) as a table is the same as of the previous Chapters pipelines, regardless of the source Table on (a).

On (f) we can observe some of the relative day columns belong to both C0875 and KTAH1. This is an example of using *both* stations for cross-site forecasting. Note also that on (f) the *-0 column belongs to the solar irradiation weather variable KTAH1 table*. If we intended to forecast the solar irradiation from C0875, we could choose the solar irradiation w.v. column from the new C0875 table, perform steps (1) to (5) on this new column, and replace the -0 solar irradiation column from KTAH1 by the -0 solar irradiation column from C0875 on (f).

Both *test* and *training* table are distinct on the same way mentioned previously, as shown on Figure 4.2: K-Means will be applied for the training tables, while the test table will be mapped using the euclidian distance equation. As with the previous pipelines, the train and test Tables on (f) must be of the same form, i.e., they must consistently use the same relative days and weather variables from -n to -0.

4.2.2 Cross-Site Training with Different Station Solar Irradiance

Let us consider now a slight variation of this pipeline, as shown on Figure 4.3. The colors are associated to the weather variables described by the pipeline. Lighter colors are associated to KTAH1 while darker colors to the same weather variable but for C0875.

Suppose the -n to -0 columns of (f) from *training* are associated to one station, while the -n to -0 columns of (f) from *test* are associated to *another* station as shown on Figure 4.3. The -0 column on both tables are associated to their respective solar irradiation weather variable, and the columns -n to -1 are associated to their respective weather variables (which must be of the same kind and

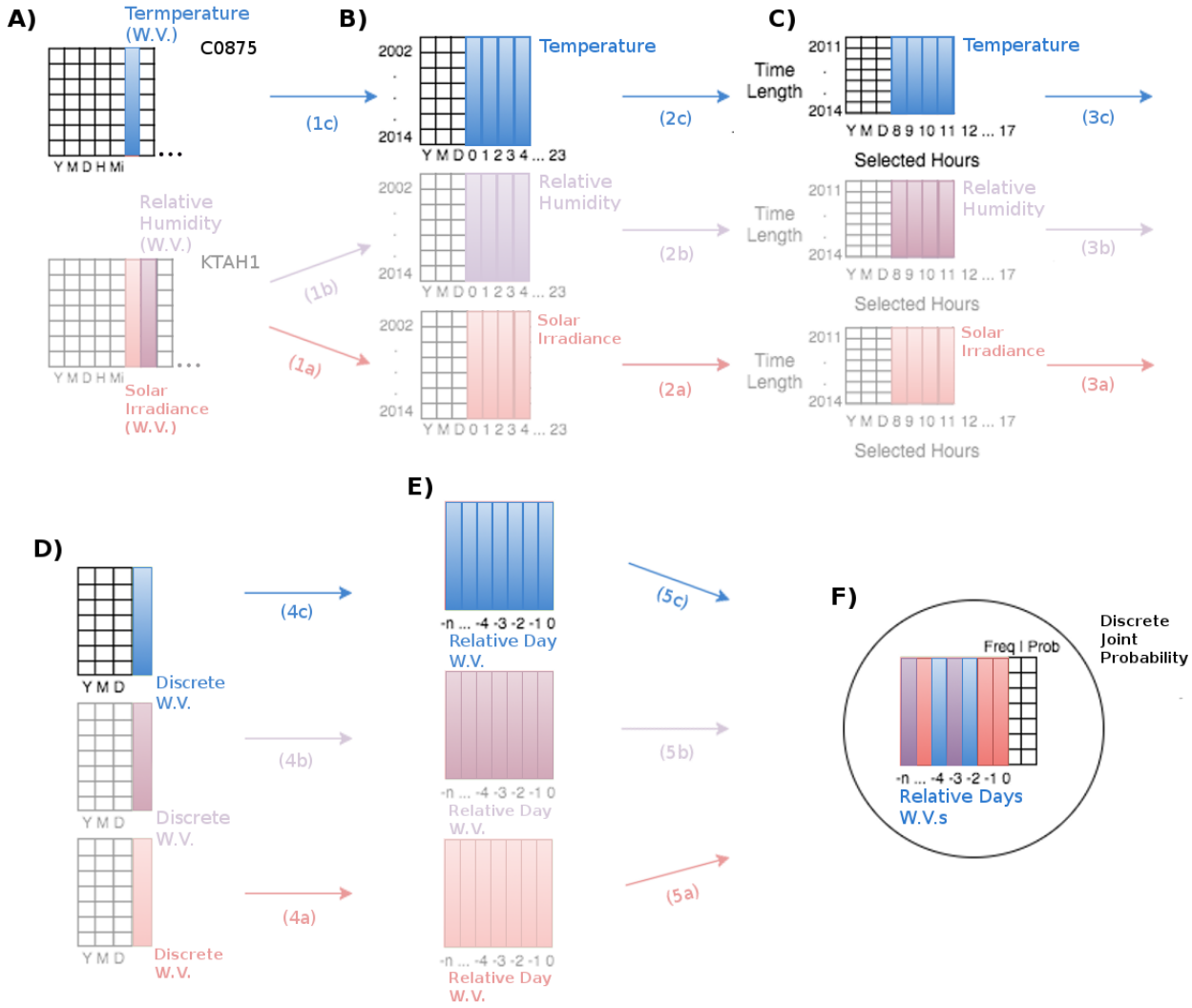


Figure 4.1: General Pipeline for any Weather Variable and station used on this chapter.

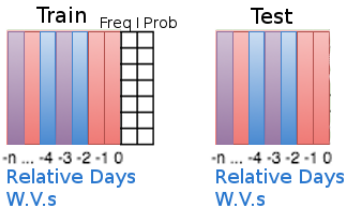


Figure 4.2: Train and Test table using two stations for training.

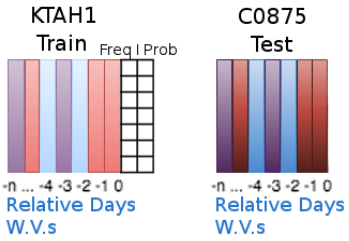


Figure 4.3: Train and Test table using one station for training. The colors are associated to the weather variables described by the pipeline. Lighter colors are associated to KTAH1 while darker colors to the same weather variable but for C0875.

relative day, otherwise the correspondence loses its meaning). This means that a probability model trained on this set-up would not learn anything of the intended station to forecast (test table). Also, when the model is attempting to forecast a given day from the test table, the days -n to -1 are not exactly the same, given the cluster id columns were generated from data of different stations, albeit the same weather variables and relative days. This constitutes the second set-up using only *one station* for training mentioned during the introduction.

4.3 Experiments

We will analyze the two variants described in the introduction in this experiment. Our overall goal is to investigate wether using weather variables from other sites decrease the solar forecasting error.

4.3.1 What are the effects in solar forecasting error using neighbor stations weather variables not including solar?

Set-up. Figure 4.4 shows stations which we could identify at least one nearby station. Out of the 88 stations available on the dataset, we show on the map only those which indicated at least 1600 days available of data.

Table 4.1 shows all the weather variable available on the neighbor stations. The available weather variables are Temperature (TMPF), Relative Humidity (RELH), Wind Speed (SKNT) and Direction (DRCT), Pressure (ALTI), and Precipitation (PREC).

Note also that not all weather variables are available on all stations. In order to be able to compare the results to previous experiments, we decided to use only the sensors that contain a suitable number of data for the years of 2012, 2013 and 2014, using the same criteria used to create Table 2.1 by inspecting the calendar plots of the neighbor station sensors. Table 4.2 shows all the weather variables which contain sufficient data for *2012, 2013 and 2014*.

C3005, MAPH1 and MKGH1 were completely removed since it only contains data between 2003

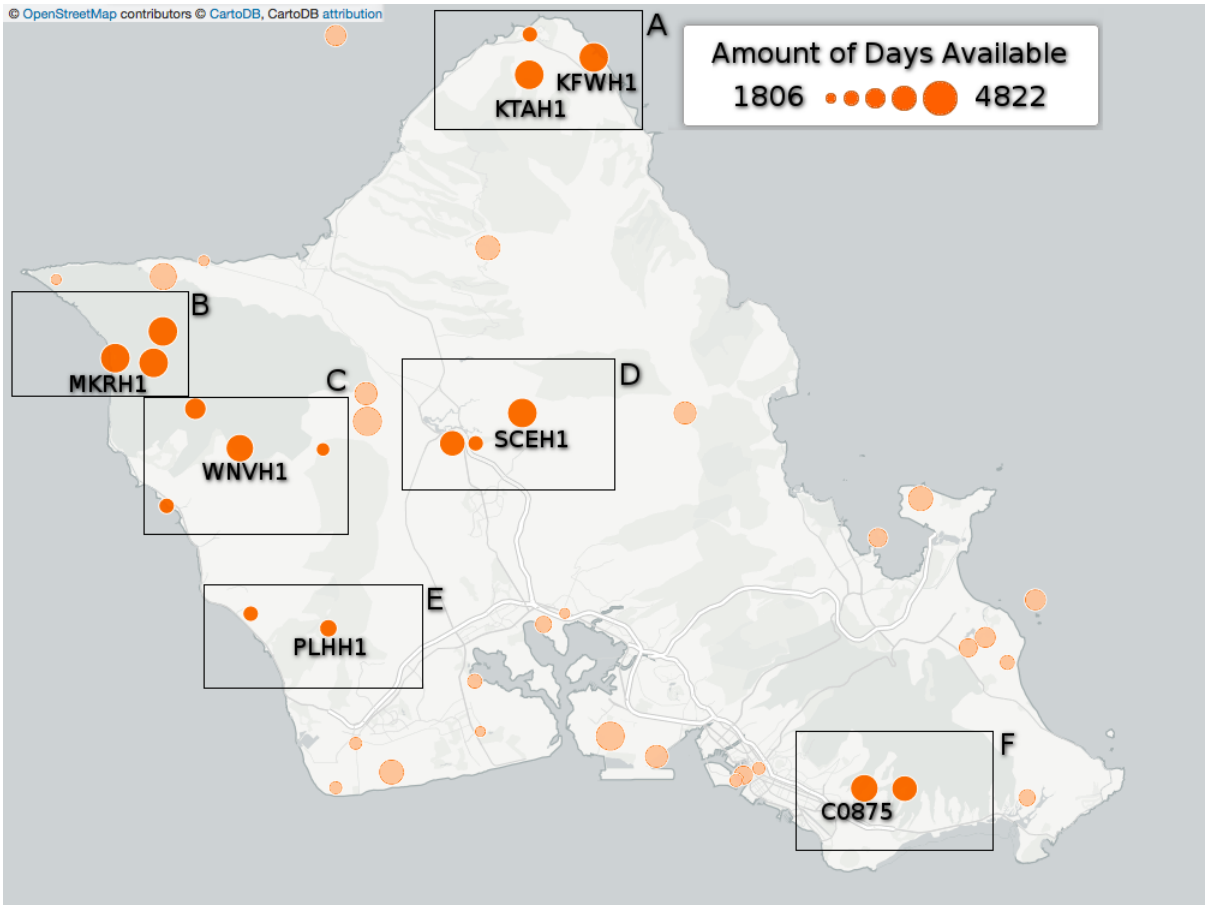


Figure 4.4: Selected Stations and Neighbors. Out of the 88 stations available, only those with at least 1600 days reported as available are shown. The actual number of data available on each sensor may be less due to *incorrect data*.

Station	Weather Variables
PHHI	ALTI,TMPF,RELH,SKNT,DRCT
C3005	ALTI,TMPF,RELH,SKNT,DRCT
MKHH1	PREC
HFO05	TMPF,SKNT,DRCT
HFO04	TMPF,SKNT,DRCT
HFO06	TMPF,SKNT,DRCT
D3665	ALTI,TMPF,RELH,SKNT,DRCT
SCSH1	TMPF,RELH,SKNT,DRCT,PREC
MAPH1	TMPF,RELH,SKNT,DRCT,PREC
MKGH1	TMPF,RELH,SKNT,DRCT,PREC

Table 4.1: Available weather variables per neighbor stations.

Station	Weather Variables
PHHI	ALTI,TMPF,RELH,SKNT,DRCT, WNUM
MKHH1	PREC
HFO05	SKNT,DRCT
HFO04	SKNT,DRCT
HFO06	SKNT,DRCT
D3665	ALTI,TMPF,RELH,SKNT,DRCT
SCSH1	TMPF,RELH,PREC

Table 4.2: Available weather variables per neighbor stations **which contain sufficient data for 2012,2013 and 2014**.

and 2009 and 2003 to 2007 respectively. This is reflected in Figure 4.5 by removing the groups B and F for the following experiments. Table 4.3 emphasizes which station's training and test tables will be used for training the model and forecasting the solar irradiation.

We did not analyze them further for the following experiments, since in the previous Chapter we observed wind speed and direction models had the highest errors.

Since SCEH1 only contains as neighbor station PHHI, we did not include it in our experiment. Contrary to other stations, PHHI is part of the federal NOAA stations. Stations of this network report weather and sky condition as separate data variables in order to provide an accurate description to pilots of existing weather conditions, cloud layer coverage and heights, etc, which provides a new set of weather variables. Also, the way NOAA distributes station data was originally built on the premise of "if you don't have something to report, don't report it", presumably as a means to reduce communications and parsing of the coded report itself, which can affect the sampling behavior of the station. For instance, by default the station reports hourly, but during active weather may sample more frequently. This will require additional care when handling this station data, and therefore was deferred to future work.

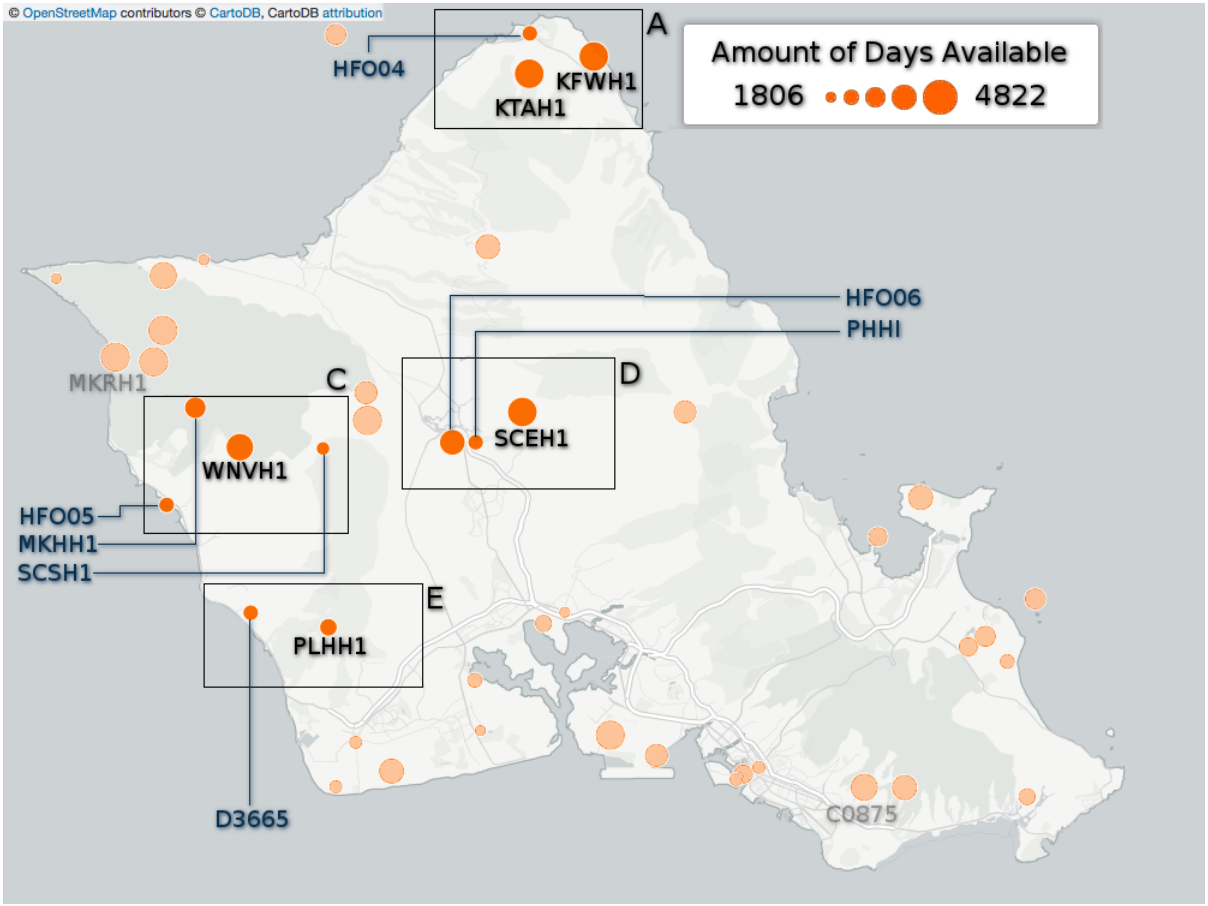


Figure 4.5: Selected Stations and Neighbors after data quality inspection and removing wind variables.

Test Station	Train Station
PLHH1	D3665 (North-West)
WNVH1	HFO05 (South-West), MKHH1 (North-West), SCSH1 (East)
SCEH1	PHHI (South-West), HFO06 (South-West)
KTAH1	KFWH1 (North), HFO04 (North-East)
KFWH1	KTAH1 (South-West), HFO04 (North-West)

Table 4.3: Train and Test station tables. The train station directions are relative to the test station which they will be used to forecast.

We will discuss each of the remaining stations results of the initial Table 4.3 separately, and then compare the overall performance of the cross-site forecasting.

PLHH1: 1 Neighbor Cross-Site Forecasting Without Solar

On PLHH1, our only neighbor station is D3665. Figure 4.6 compare the forecasting error of using the same station weather variables versus using the available weather variables from D3365 (and PLHH1 as the -0 solar irradiation weather variable).

We can see that using the relative humidity from the other station provide a slight improvement compared even to the solar irradiation model of the same station. If we recall the previous Chapter 3, we only observed a slight improvement using weather variables with interaction for training the models using season partitioning. Without the partition, solar models had the lowest forecasting error. We can also compare temperature, in which case forecasting from the same site had lower error.

WNVH1: 2 Neighbors Cross-Site Forecasting Without Solar

For WNVH1 on Figure 4.7, we again observed solar as having the lowest forecasting error, and cross-site forecasting presents for the same weather variables a similar error. Although we have not observed an improvement in the forecasting error, we can note the distances between the stations did not affect the error significantly (i.e. beyond 100 W/m^2).

KTAH1 and KFWH1: Cross-Site Forecasting with solar

KTAH1 and KFWH1 were discussed on the previous Chapter as these stations contained solar sensors, and therefore are the only pair of nearby stations we can observe if using other station solar data improves the model. From Figures 4.8 and 4.9 we can observe that aside from temperature in KTAH1 forecast, all weather variables had higher forecasting error by cross-site forecasting.

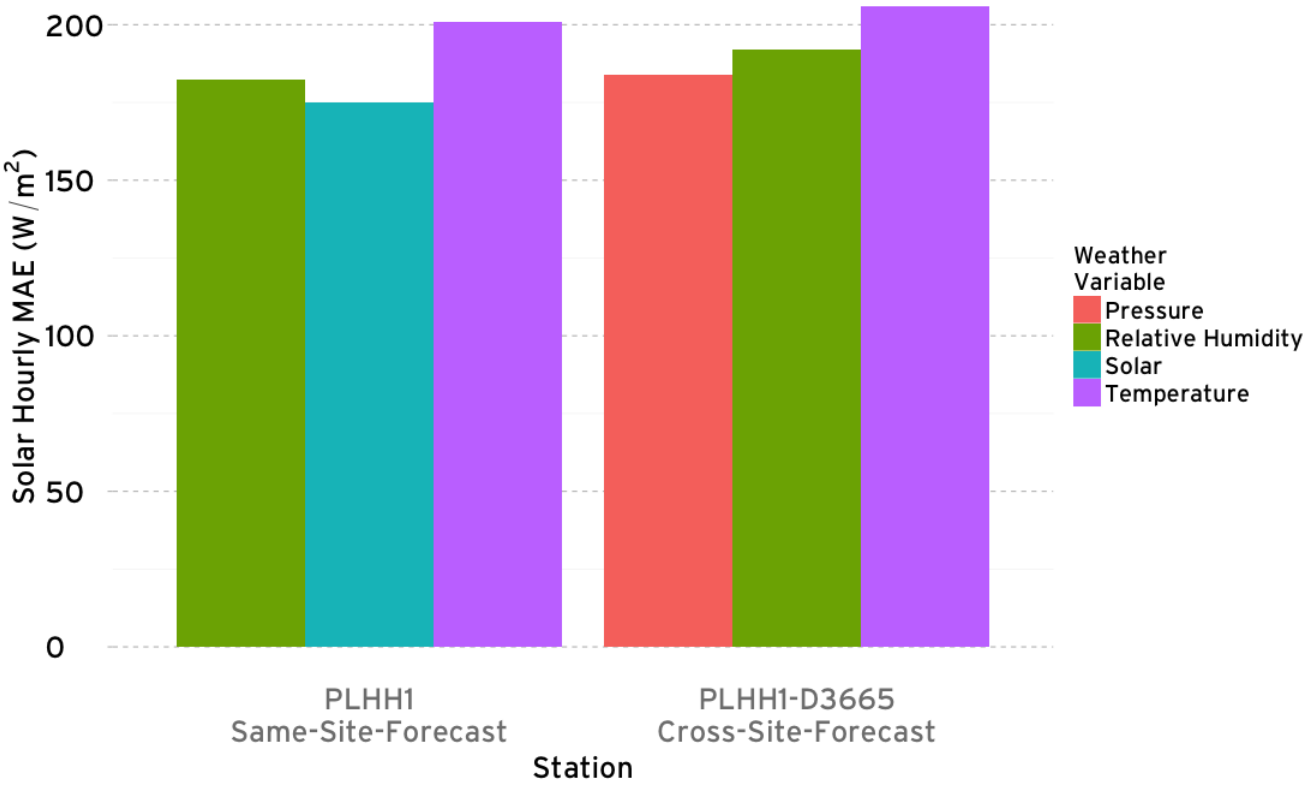


Figure 4.6: Forecast Error using the same station weather variables and cross-site forecasting with D3665.

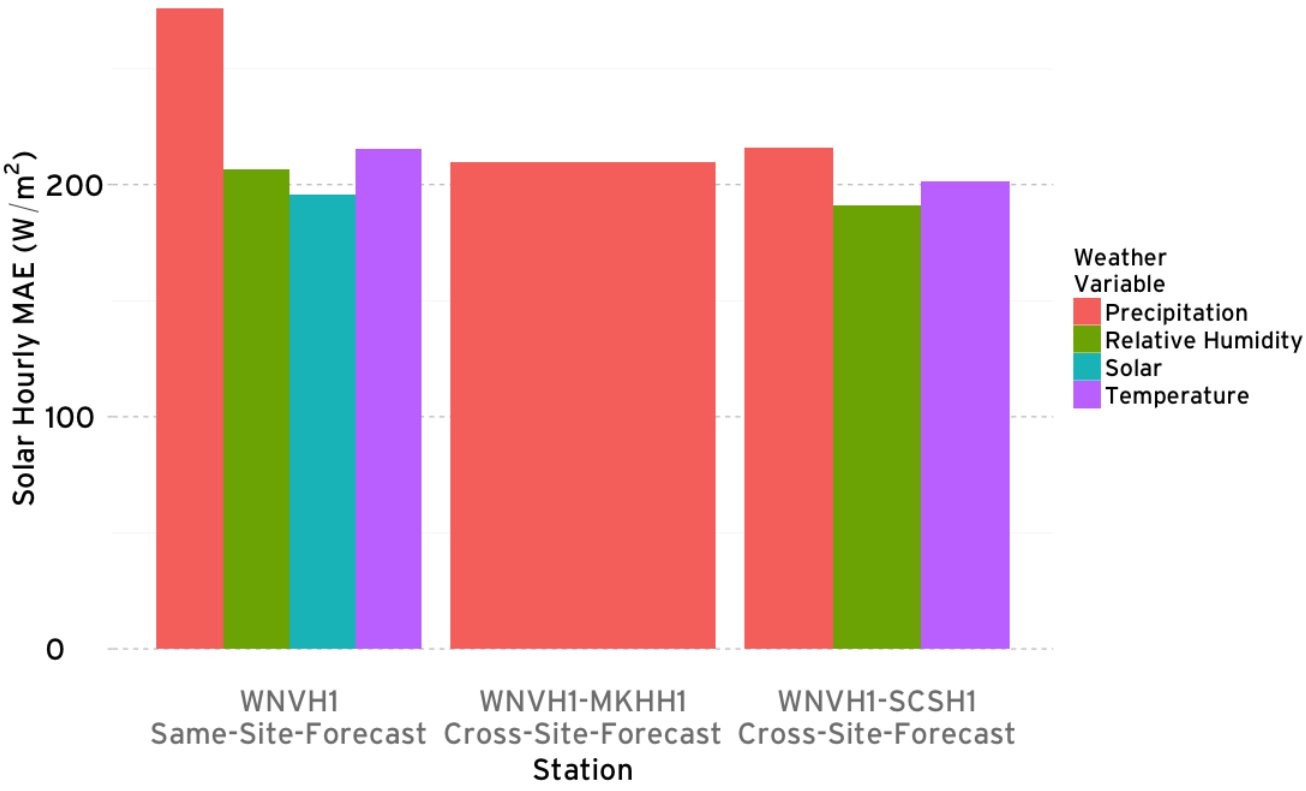


Figure 4.7: Forecast Error using the same station weather variables and cross-site forecasting with MKHH1 and SCSH1.

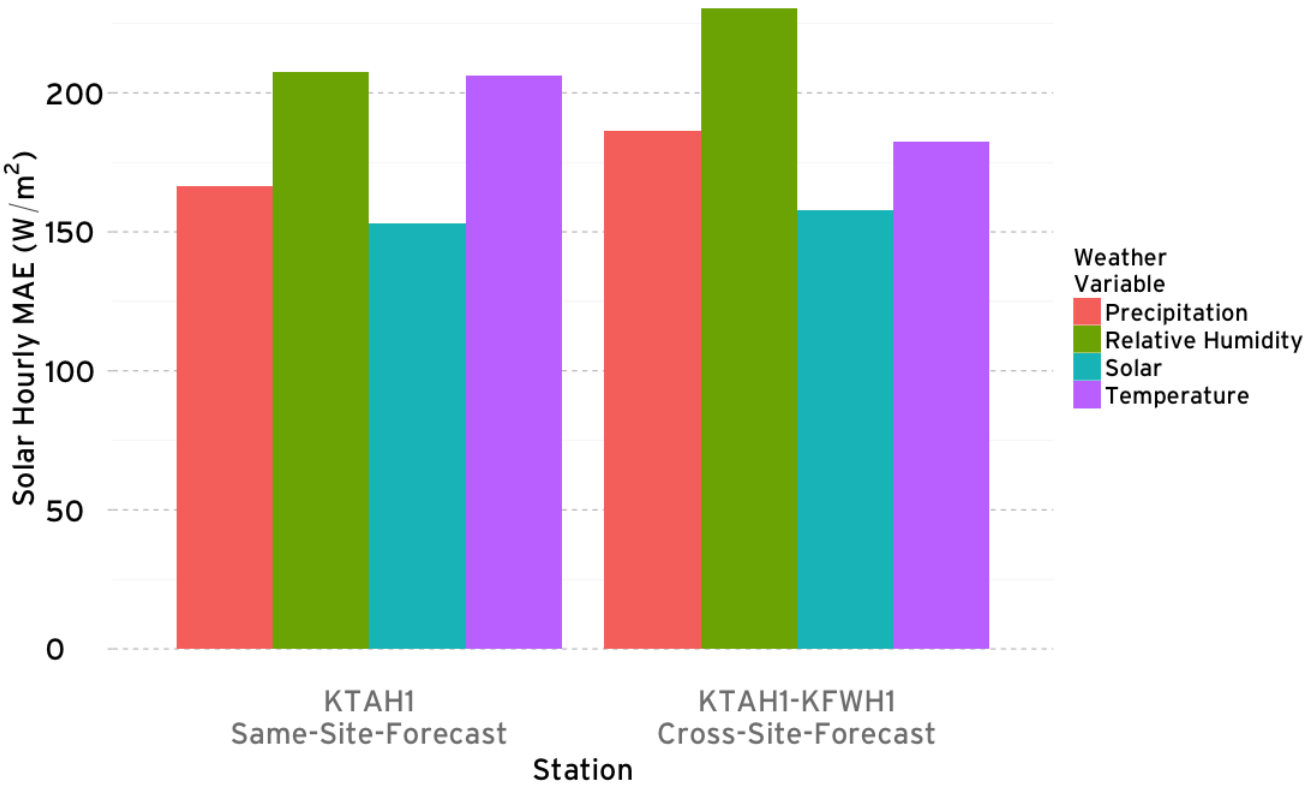


Figure 4.8: Forecast Error using the same station weather variables and cross-site forecasting with KTAH1 and KFWH1.

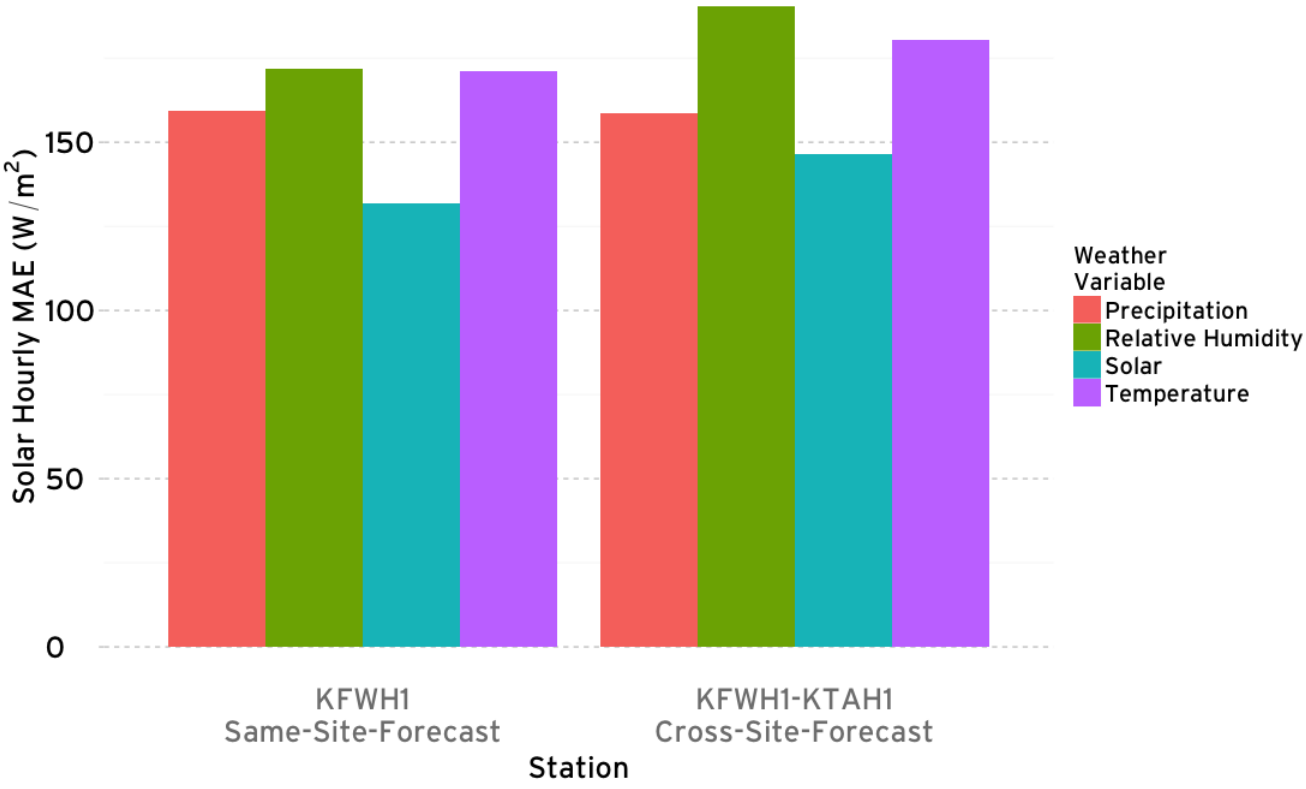


Figure 4.9: Forecast Error using the same station weather variables and cross-site forecasting with KFWH1 and KTAH1.

4.3.2 What are the effects in solar forecasting error using neighbor stations weather variables including solar

Set-up. For this experiment, the training dataset uses the solar irradiation data also from the other station, being trained therefore completely independently. In this case, this will require the other neighbor station to contain the solar variable, which reduces the number of stations we can consider to only KTAH1 and KWH1.

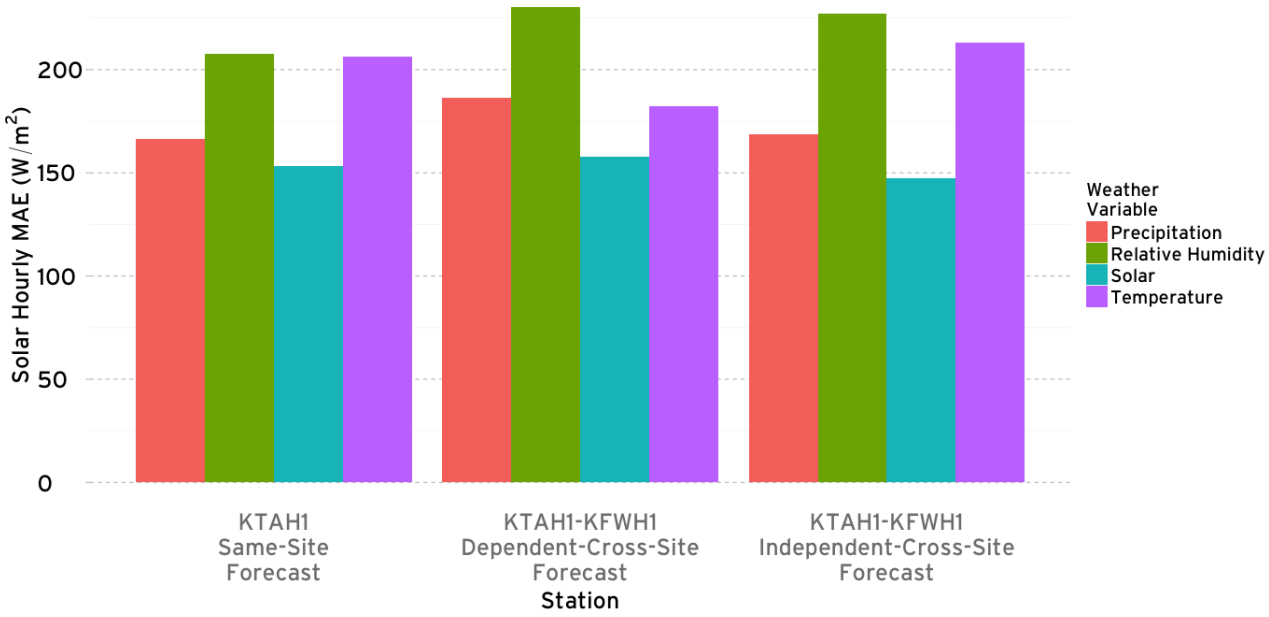


Figure 4.10: Forecast Error using the same and independent stations weather variables and cross-site forecasting with KFWH1 and KTAH1.

Result and Conclusions. From Figures 4.10 and 4.11, we can observe the forecasting errors for all weather variables vary only slightly for KTAH1 and KFWH1, but using the same station solar irradiance data still provides the lowest forecasting error.

4.4 Conclusions

In this Chapter, we investigated if using neighbor stations weather variables could improve the forecasting error. We concluded that a small difference in the forecasting error is observed, and using the same station solar data still provides the lowest forecasting error.

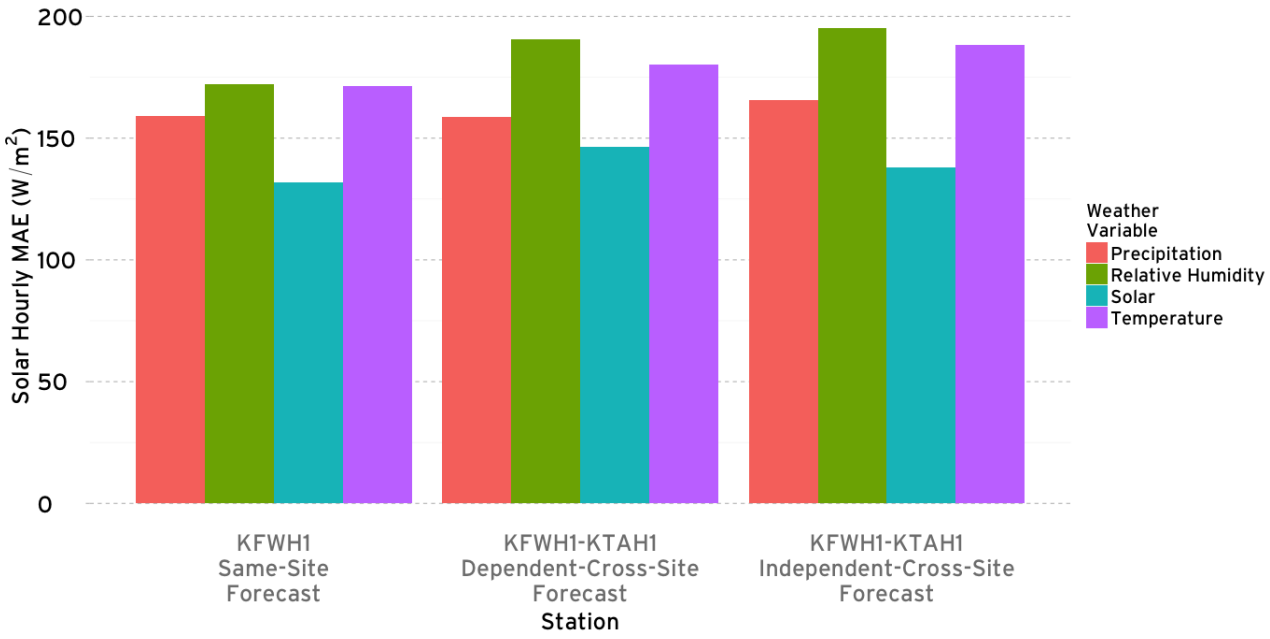


Figure 4.11: Forecast Error using the same and independent stations weather variables and cross-site forecasting with KFWH1 and KTAH1.

CHAPTER 5
CLUSTERING, CONSECUTIVE DAYS AND PREDICTION FUNCTIONS

5.1 Introduction

In the previous Chapters, we focused mostly in the choice of weather variables of our probability models, and the parameter **w**, the number of consecutive days used to extract tuples. In this Chapter, we will focus in the number of clusters **k** used by k-means to cluster the data to understand how it effect the solar forecasting error. We will also explore how the choice of **k** can influence the choice of **w** and the trade-off of the parameters in the final forecasting error.

A second limitation observed in the model was observed when investigating the choice of the number of consecutive days **w**, as more years of data were used to train the model varied the number of missing forecasts from less than 1% up to 99% of the entire forecast dataset. This limited the previous Chapter analysis to a very low number of consecutive days **w**. An alternative to use the most frequent centroid in replacement to the missed forecast on models that contain a high number of missing forecasts, is to simple use another model with different number of consecutive days **w**. For instance, if a model with **w=6** has 40% missing forecasts, then instead of replacing the 40% for the most frequent centroid, we could instead use for the missed forecast days, **w=5**. More generally, for any chosen model **w**, we are posed the question of what other model to choose that was able to forecast the given day. We will discuss some prediction functions in the second part of the Chapter.

5.2 Method

5.2.1 Clustering Error and the choice of k

As discussed in Chapter 2, we can observe the forecasting error as consisting of two parts: **the clustering error contribution**, and the **cluster id prediction error contribution**. The sum of both errors constitutes the forecasting error. Attempting to lower one error increases the other: Increasing the number of clusters **k** results on any one element of the tuples extracted to compute the estimated probability distribution will also have more possible values. For instance, suppose our chosen k-means probability model, in particular, the set-up using only the previous day solar irradiance data. If **k=2**, then any one tuple is of the form (i,j), where i and j can either take the values 1 or 2. As k increases, the number of possible combinations, and therefore distinct tuples also increases, up to a maximum of k^w tuples (the exact number will depend on the mapping from the actual data to cluster ids). Increasing the number of combinations is not ideal, however, since

it increase the likelihood of missing forecasts.

This is the same problem faced with the increased values of consecutive days **w**: Adding more elements will also increase the chances of unique combinations, which in turn lead to missing forecasts, as the chances that a certain tuple combination occurs in test and not in the training data-set increases. Therefore, we must always seek for a compromise between increasing **k** and increasing **w**, as increasing both would quickly lead to the whole test dataset being non-forecast due to missing tuples in the training dataset. We will investigate this trade-off empirically in our first experiment question.

5.2.2 Entropy, Support and Prediction Functions

We are still faced with one problem: if we have a set of values for w, we will have a set of different probabilistic models (one for each value of w). How do we choose which model to use (i.e. choose w)? It should be clear that w is associated with the question of how much history is needed to predict the profile for the next day. We investigate three approaches:

Fixed Choose the w that minimizes the prediction errors on the training data and use the same w for all the testing instances,

Entropy Given a testing instance, dynamically choose the w that minimizes the entropy of the posterior distribution (Eqn 2.3),

Support Similar to entropy, but further weight the entropy with support.

The fixed method was used in all the previous Chapters.

For the entropy method, we are given a testing instance which is a cluster id sequence $\langle s_1, s_2, \ldots, s_{w-1} \rangle$ and need to predict the cluster id on day i. Let $H(w)$ denote the entropy for the distribution,

$$P(S_t | S_{t-1}{=}s_1, S_{t-2}{=}s_2, \ldots, S_{t-w+1}{=}s_{w-1}). \tag{5.1}$$

The entropy method would choose w as

$$\hat{w} = \arg\min_{w} H(w). \tag{5.2}$$

Minimizing the entropy ensures that the model with the most skewed posterior distribution is chosen. Recall that a uniform distribution cannot distinguish between the five cluster ids. A skewed distribution, on the other hand, means that the historical data tends to favor a particular cluster id given the $w-1$ previous days' cluster ids.

One difficulty with the entropy method is that it is possible for a skewed distribution to be based on very few data points (giving us less confidence), while a less skewed distribution may be supported by a large portion of the data (giving us more confidence). Note as the window

size w increases, the number conditional variables $(S_{t-1}{=}s_1, S_{t-2}{=}s_2, \ldots, S_{t-w+1}{=}s_{w-1})$ increases, but the number of occurrences of each distinct sequence of $\langle s_1, s_2, \ldots, s_{w-1}\rangle$ decreases, resulting in lower support for that conditional distribution. To account for the support, we weight the entropy by the number of supporting data points for that distribution. The support method would choose w as

$$\hat{w} = \arg\min_{w} \frac{H(w)}{N(s_1, s_2, \ldots, s_{w-1})}, \tag{5.3}$$

where $N(s_1, s_2, \ldots, s_{w-1})$ denote the number of occurrences of the sequence $\langle s_1, s_2, \ldots, s_{w-1}\rangle$ in the data.

5.3 Experiments

5.3.1 How k effects the forecasting error?

Set-up. We used all stations which contain sufficient solar irradiation data between 2012 and 2014 inclusive, by training the kmPM models with 2012 and 2013 solar data and using 2014 as our testing dataset.

Results and Conclusions. Figure 5.1 shows the forecasting error for each chosen station (represented as a line) as we vary **k** in the X axis.

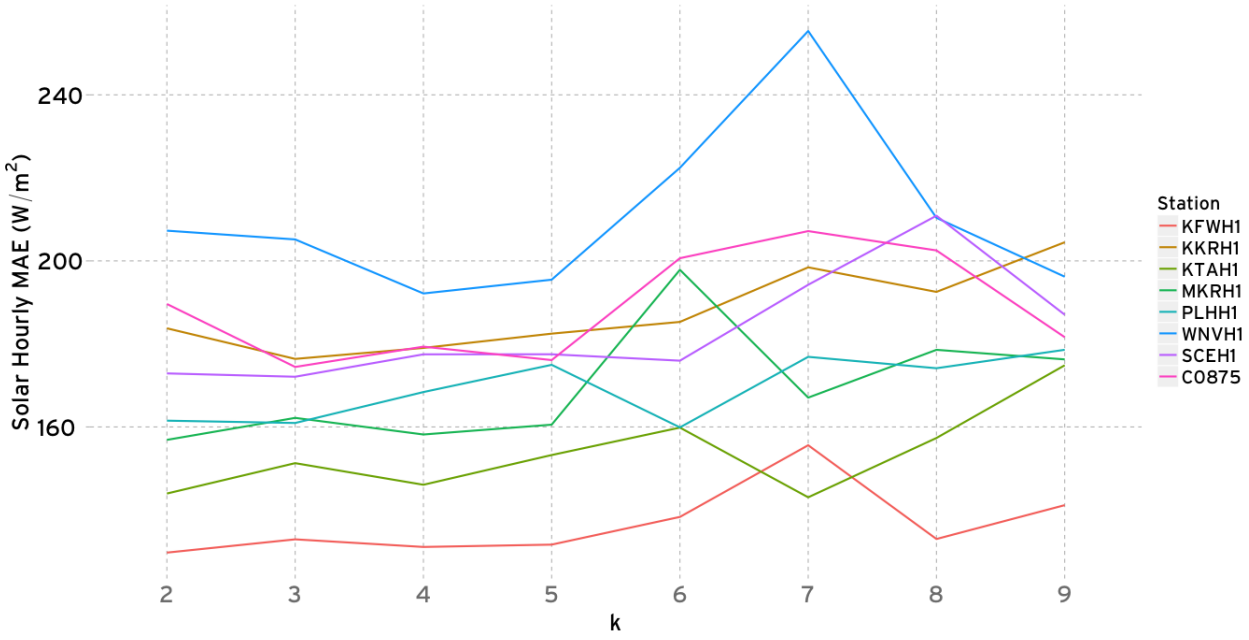


Figure 5.1: Forecasting error as k varies in different stations. Each station kmPM model was trained with solar irradiation data from 2012 and 2013 to forecast 2014.

We can observe that each line valley and peaks differ, indicating that the choice of k varies

depending on the station. Moreover, choices of k between 2 and 5 seems to overall present the lowest forecasting error, which justifies our preference for **k=5** quantitatively, on top of the qualitative interpretation presented in Chapter 2.

Figure 5.2 splits the forecasting error of Figure 5.1 in the the clustering error contribution (Figure 5.2a) and cluster id prediction error contribution (Figure 5.2b). Here, it is evident the trade-off between the number of consecutive days **w**, and the number of used clusters **k**.

Recall from Chapter 2 that the clustering error contribution in (W/m^2) is calculated by setting up the model in such way the cluster id predictions are always correct. Calculating the forecasting error in this set-up shows how much of the error is then associated to mapping the hours to centroids. The clustering error in (%) is then the clustering error in (W/m^2) divided by the forecasting error when the probability model can also make a wrong cluster id forecast, contributing further to the final forecasting error. The incorrect cluster id forecasts in (%) counts how many times the forecast was different from the discretized test dataset out of the 365 days of 2014.

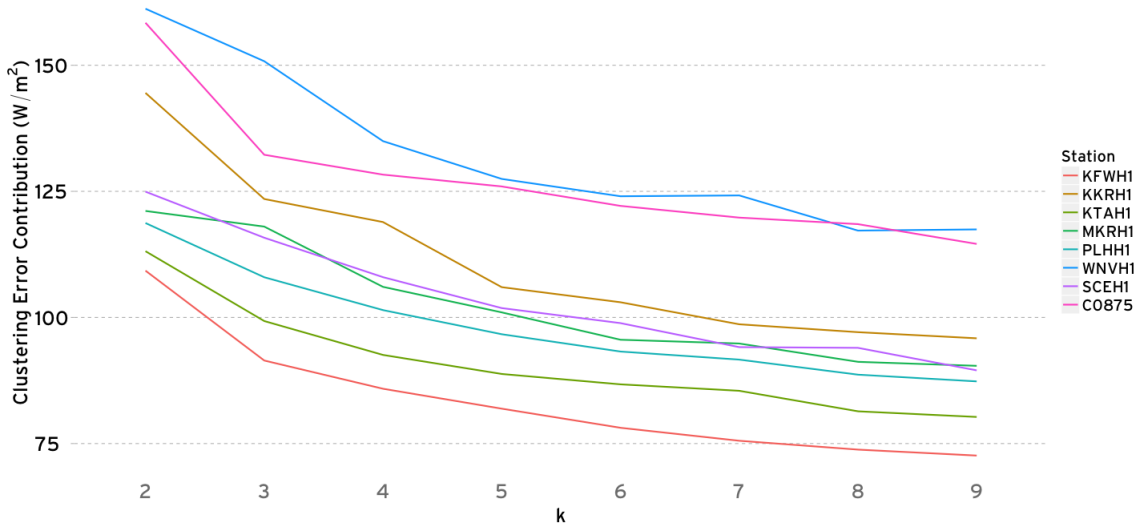
Putting together, we can see in Figure 5.2a, as we increase **k**, it's error contribution decreases, as the centroids more closely reflect the different solar irradiance of every day. However, this does not come without a cost: Increasing **k** increases the number of values any element of the tuple can take, leading in turn to a higher number of possible combinations (k^w), which leads the probability model to more incorrect forecasts. In the end, the decrease in forecasting error obtained adjusting one parameter is off-set by the other, as the error fluctuation reveals in Figure 5.1, revealing the trade-off underneath.

5.3.2 How the entropy and support entropy effect the forecasting error?

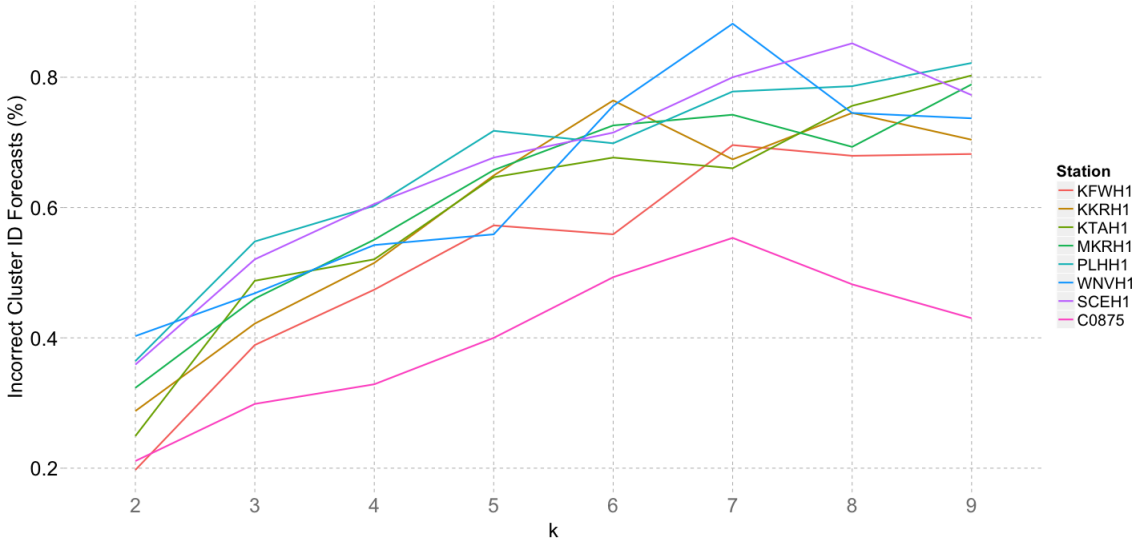
Set-up. For this experiment, we used only KTAH1 (which is the only station which contains 12 years of data), as from Chapter 2 we observed increased the number of consecutive days **w** quickly led to 100% missing forecasts. The years of 2003 to 2013 were used as training, while 2014 to forecast.

Results and Conclusions. From Figure 5.3a we can observe the entropy (E) and entropy support (ESpt) have a similar error to the fixed model using just the previous day (**w=2**). Given the similarity of the errors, one could be led to think the entropy and entropy support decided to "trust" the fixed model **w=2** forecasts more than the others, therefore reducing the use of different fixed models **w** to a **w=2** model. Although from Figure 5.3c this is in part the case for the entropy support prediction function, as out of the 365 days of 2014 it used **w=2** for 175 days (47%). For the entropy prediction function, however, the model correctly chose from the higher **w** values 5, 6 and 7 the centroids.

These results suggests that given enough data, using number of consecutive days can lead to better forecasts, however the number of missing forecasts ends up blocking better forecasts in the fixed model, as discussed in Chapter 2.

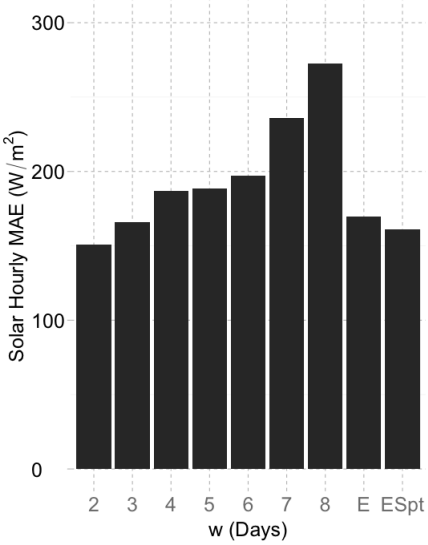


(a)

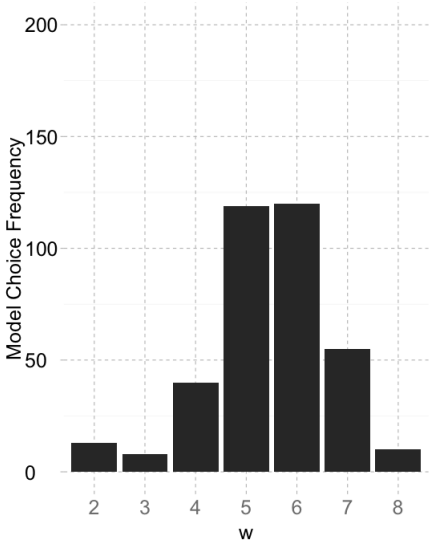


(b)

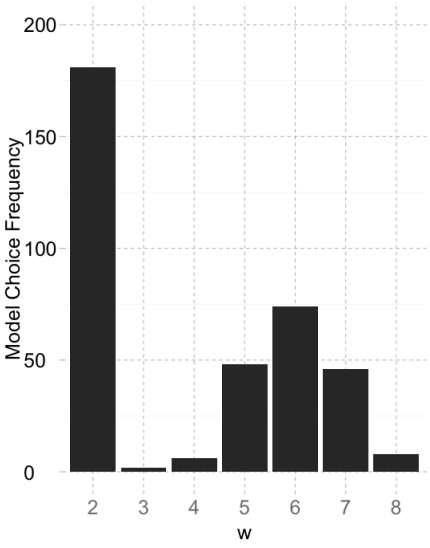
Figure 5.2: Forecasting error contributions for clustering and incorrect cluster ids for the forecast in the fixed set-up for different numbers of cluster id k.



(a) Forecasting Error using Fixed (previous Chapters prediction function), Entropy (E) and Support Entropy (ESpt)



(b) Model Choice Frequency for the Entropy Prediction Function



(c) Model Choice Frequency for the Support Entropy Prediction Function

Figure 5.3: Forecasting error for KTAH1 using 11 years of data (2003 to 2013) to Forecast 2014 with the 3 prediction functions. The model count frequency shows how the entropy and support entropy functions selected from the fixed model the forecasts.

5.4 Conclusions

In this Chapter, we investigated the effect of the number of clusters **k** in the forecasting error, and the trade-off between **k** and the number of consecutive days **w**. It became clear from the experiments, that the lowest forecasting error is not obtained by any fixed k, even using only solar irradiance on the day before to train the models. We also saw that as we attempt to increase **k** to the number of possible combinations increase, leading the cluster id forecast to have poorer performance, and therefore compensating back the lower forecasting error obtained by clustering.

In the second half of the Chapter, we investigated a more flexible approach to forecast using entropy and support entropy functions. Rather than choosing a fixed **w**, we deferred the decision for the model for every day forecasted based on the skewness of the probability distribution. We concluded that the model presented forecasting errors closer to the simplest model, i.e., using the day before solar irradiance, but however did not reduce the decision of the models solely to **w=2**, which may suggest given enough data, better forecasts could be achieved for higher values of consecutive days **w**.

CHAPTER 6
CONCLUSIONS AND FUTURE WORK

In this thesis, we investigated the use of data mining methods to address solar forecasting 1-day ahead. Throughout all the Chapters, we observed that using past solar irradiance data to forecast solar irradiance 1-day ahead presented the best results. In Chapter 2, we observed a combination of k-means with the probability model presented equal or lower forecasting error when compared to the other prediction and clustering data mining methods.

Already in Chapter 2, we also observed that three parameters were required to be decided a-priori: The **number of years** for training the models, the number of consecutive days **w** and the number of clusters **k**. The choice of parameters were bounded by the worst case scenario exponential increase of possible combinations of consecutive days, as k^w possible combinations increased the likelihood the model could not find a matching sequence of past days to forecast. Since most sites contained mostly 3 years of data, using **w=2** consecutive days, lead to the best results. Furthermore, in Chapter 5, we concluded that no ideal value of centroids **k** existed: Rather, **k** was too subject to the choice of the other parameters.

In Chapter 3, we considered how other weather variables could correlate to solar irradiance using the k-means probability model. Perhaps a bit surprising, no combination of weather variables with or without solar led to lower forecasting errors. In this case, it is not necessarily true that no underlying model containing a certain arrangement of weather variables would lead to best results: As with the previous Chapter 2, combining more weather variables would lead to an increase of possible combinations, again binding the model by the number of missing forecasts. This poses the question then if, provided enough data, one would be able to identifying an underlying model that would lead to a decrease in forecasting error and better predict solar irradiance 1-day ahead.

Still in Chapter 3 we investigated time partitions. Beyond other weather variables, temporal phenomena such as el nino, la nina and seasons could better capture the variance of past and future weather variables correlations. While observing this was true for seasons, el nino and la nina did not lead to better results.

In Chapter 4, we generalized the training of the models to other sites. Would a nearby site past weather variables correlate to our intended forecasting site? One intuition to posing this question would be cloud motion: Phenomena occurring in a nearby site could be occurring in the future in our forecasting station. While we did not observe lower forecasting errors, we observed similar performance throughout most of our weather variable models, suggesting a potential non-linear correlation of the different sites weather variables and future solar irradiance.

Finally in Chapter 5 we revisited the models and limitations identified in all Chapters to gain a better understanding of the limitation of the number of clusters **k** and the number of consecutive days **w**: Is it possible to address the missing forecasts, allowing for the investigation of more complex

models? We proposed then a set-up to group the several probability models by using an entropy function to decide based on the skewness of the estimated probability distribution and support, the number of observations used to estimate the distribution, which of several models to choose to forecast depending on the past days. Remarkably, the entropy prediction function supported our initial thoughts: The entropy probability model was able to obtain similar forecasting error to the simplest probability model, however choosing higher number of consecutive days $\mathbf{w}$. This suggests that when replacing the missing forecasts by other models forecasts with a more elaborated criteria rather than most frequent centroids, it may be possible to surpass the simplest of the models.

Whereas this thesis investigated multiple set-ups to investigate how the choice of weather variables, number of years, model parameters and it's limitations impacted the forecasting error, there are still several paths of improvement:

In chapter 2, although we systematically investigated the 88 stations available in Oahu island since 2002 up to the end of 2014, the final number of stations used throughout the experiments were still unfortunately low: Only 8 stations, when using 3 years of data or 2 stations when using 10 years of data were used in our experiments. The interpretation of our results could be different, if more stations would be available, or at least without the several issues encountered which greatly reduced the number of years that could be used. All stations, aside from C0875 which only contained 1 year worth of data, sampled hourly. If lower granularity of data would be available, multiple levels of clustering from second to days could be investigated to identify different clusters of solar irradiance.

Another unexplored path in chapter 2 are the compared models. All models used in this thesis assumed stationarity. Non stationary linear and probability models, i.e., which assume the patterns change over time, could perhaps yield better results. Hybrid models combining decision trees with probability models could also lead to different ways to map cluster ids to numeric solar irradiance measurements.

In chapter 3, the number of limited data again limited the exploration of more complex models by combining more weather variables and previous days. A point of extension here is the combination of weather variables and time partitioning before they are discretized: Weather variables can be clustered together, for example, by having a cluster id '2' indicate a combination of past solar irradiation and temperature altogether. For the time partitioning, instead of encoding el nino as a cluster id, we could partition the data before clustering, therefore impacting the centroids representation of each cluster.

In chapter 4, we only briefly explored the many different combinations of weather variables in different sites: Beyond neighbor stations, we could also attempt to study further away stations and multiple sites simultaneously. Beyond more distant stations, other data at different resolutions could also be incorporated to account for even further distances, such as satellite data. While preliminary investigation (not presented in this thesis), lead to much higher forecasting errors, we suspect again that data limitations impacted in the conclusion of our preliminary results.

In chapter 5, we only investigated groups of solar models. Virtually, however, we could use any combination of models present in all previous chapters: From same or different sites, to various set-up of consecutive days, weather variables and even satellite data.

BIBLIOGRAPHY

[1] Kevin Eber and David Corbus. Hawaii solar integration study: Executive summary. Technical report, National Renewable Energy Laboratory, June 2013.

[2] Laura M. Hinkelman. Differences between along-wind and cross-wind solar irradiance variability on small spatial scales. *Solar Energy*, 88:192 – 203, 2013.

[3] Luis Martn, Luis F. Zarzalejo, Jess Polo, Ana Navarro, Ruth Marchante, and Marco Cony. Prediction of global solar irradiance based on time series analysis: Application to solar thermal power plants energy production planning. *Solar Energy*, 84(10):1772 – 1781, 2010.

[4] A. Moreno-Munoz, J.J.G. De la Rosa, R. Posadillo, and V. Pallars. Short term forecasting of solar radiation. In *Industrial Electronics, 2008. ISIE 2008. IEEE International Symposium on*, pages 1537–1541, June 2008.

[5] Gordon Reikard. Predicting solar radiation at high resolutions: A comparison of time series forecasts. *Solar Energy*, 83(3):342 – 349, 2009.

[6] Fei Wang, Zengqiang Mi, Shi Su, and Hongshan Zhao. Short-term solar irradiance forecasting model based on artificial neural network using statistical feature parameters. *Energies*, pages 1355–1370, 2012.

[7] Dazhi Yang, Zhen Ye, Li Hong Idris Lim, and Zibo Dong. Very short term irradiance forecasting using the lasso. *Solar Energy*, 114:314 – 326, 2015.