Forecasting Electricity Production from Photovoltaic Solar Panels using Elastic Net Regularization

Master Thesis in Applied and Computational Mathematics

Benjamin Mekki Widerøe

Department of Mathematics
University of Bergen
Spring 2019

Acknowledgements

First, I would like to thank my supervisors, Tor Sørevik and Nello Blaser for all the help they have provided while I was writing this thesis. Second, I would like to thank my fellow student at the University of Bergen, Karoline Lekve, for the collaboration on this project. I would also like to express my gratitude to Einar Eimhjellen, former student at the University of Bergen, who introduced me to the topic of solar energy with his thesis in 2018. Finally I would like to thank my friends and family for providing motivational help in the form of food and entertainment.

Abstract

As the number of solar power plants grows throughout the world, being able to predict the power output becomes more essential than ever. Such predictions could either be built on top of weather predictions, or they could rely on historical data. In this thesis we use the latter approach. However, any data that a source provides is certain to contain both errors and missing data, which will affect the prediction of any forecast technique that is employed. The aim of this thesis is to be able to forecast the power production of a solar plant one hour into the future. In order to do this we need quality data, so we will also look at the data quality of data sets, and see how we can aim to improve erroneous data. We will use this improved data when forecasting, and compare the results to our forecasting on the original data to see if the results improve.

Contents

List of Figures

List of Tables

Preface

This thesis is the first part of a joint project with fellow student Karoline Lekve, where we both aim to solve different parts of the same problem. Because of this, we chose to collaborate on the introduction.

Chapter 1

Introduction

The goal of this thesis is to find accurate methods for forecasting power production of solar plants. We have several data sets that we will examine. The data sets will be preprocessed and evaluated for inaccuracies and missing values, and different methods will be applied in order to predict future values of power. The historical data will need to be relatively large to make accurate predictions, so we will be needing enough data to split into training and testing parts.

First we need to look at why it is important to predict power production. Renewable energy is gathering a strong foothold as power sources around the world and demand will seemingly only increase over time. Any company that wishes to produce renewable power has an interest in accurate predictions of power production.

In a hypothetical future where we are heavily reliant on renewable energy sources, we need tools to ensure that the power grid of any city, country or continent remains stable, even when the production from renewable energy fluctuates. For this reason, we must use forecasting to predict output of solar power in order to accurately balance the grid with alternative energy sources. Knowing in advance the approximate power production from a solar plant gives us the opportunity of increasing or decreasing production from other energy sources. Some forms of power production have longer ”wind-up-times” than others (see Table 1.1), so when implemented in real-time scenarios, we may have to adjust the forecast time window if required.

	Nuclear	Hard coal	Lignite	CCG	PS
Start-up time 'warm'(h)	na	5-7	2-8	2-4	< 0.1
Ramp rate (%/min)	0	0.6-4	0.6-6	0.8-6	15-25
Minimal possible load (%)	100	40-60	40-60	40-50	5-6

Table 1.1: Start up time, ramp rate and minimal possible load for different energy sources. CCG=Combined cycle gas, PS=Pumped storage. Table taken from [J. Antonanzas et al 2016].

Here start-up time is the time needed for the production to begin, ramp rate is the highest possible relative increase in production (e.g. if it takes 5 minutes for the machine to go from 0-100%, the ramp rate is 20%/min). Pumped storage refers to a form of gravitational energy storage, used for load balancing, where excess energy is used to pump water into a basin high above the ground, which can be released whenever more power is needed on the grid. We see that some of the sources have long start up times, and therefore need a larger forecast window than others.

When considering prediction of power production from solar panels, there are three main strategies that can be used for analysis and prediction.

- Statistical analysis of electricity production
- Physical representation of production
- Hybrid

Statistical analysis is evaluating historical data of a variable, and using this as the sole basis of our prediction. Time series and machine learning algorithms fall into this category. One of the perks of this method is the simplicity of it - we only need at least one variable to predict, and it requires no understanding of how solar power works. It is however limiting, and fails to consider weather, which the power production of a solar plant is highly dependent on. Also, our basis for new information is always past information, and for weather values this will not always be sufficient. It also means that if our data is inadequate, our results will be too.

The other method - physical based representation, is representing the new values as a combination of physical values and weather phenomenon - looking at irradiance, weather data (mainly cloud cover and temperature), time of day and physical placement of solar panel. This would be an improvement of our previous method as the data is represented based on weather values which are more likely to be correct when modelled from physical observations and not as a pattern of historical information. This however means we will have to construct our own models and equations for the values and we always run the risk of not knowing how reliable our data is. If we do run into extreme weather conditions we will however be able to predict power production reflecting this (unlike patterns in historical data which will not help capture extreme weather conditions in their predictions.)

A third option is a hybrid of these two methods, a combination of historical data and weather values. These theses will only focus on historical data and not look at any pure physical representations of the production or hybrid methods.

Of the different methods that are used for prediction, the hybrid option is usually the most accurate, while statistical is a close second. [1]

For accurate predictions, a large amount of good quality data is necessary. This project can be split into two parts - quality check and improvement of data, and forecasting using different methods. This thesis will look into the data, evaluate missing/incorrect measurements and aim to correct for inadequacies, and then use machine learning methods to forecast using both the preprocessed data, and the original data, and compare the results. The second thesis in this project will use some of the preprocessed data to forecast values - first with the help of time series analysis and then machine learning methods.

Chapter 2

Data sets

2.1 Bari, Italy

The Bari data set is a subset of a larger data set referred to as "PVItaly", which originally consists of data from 18 plants in Italy. The original dataset features many different plants on different altidues and with different azimuths (angle with respect to the ground), but the specific sets we use consists of hourly data from three solar power plants in Bari, Italy. The plants are in relatively close proximity to each other. We chose to use these data sets because they featured more variables than any of the other free data sets we were considering. More details on the larger data set can be found in [Ceci et al].

The data we work with was recorded between January 1^{st} and December 31^{st} 2012. It consists of hourly measurements from 02:00 to 20:00 each day. These 19 measurements each day for 366 days (2012 was a leap year) give us 6954 data points for each measured variable. The variables in the data are:

Irradiance The irradiation from the sun, measured at the location of the plant. Unit for solar irradiance is watt per square meter (W/m^2).

Power The output of the power plant, measured in kilowatts (kW).

Weather temperature Air temperature at the location of the power plant. Measured in Celcuis (°C).

Plant temperature Temperature of the solar panels. Measured in Celcius (°C).

Cloud cover Dimensionless continuous variable between 0 and 1, where 0 represents clear sky, and 1 total cloud cover.

(Relative) Humidity is the ratio of the partial pressure of water vapor to the equilibrium vapor pressure of water at a given temperature. Usually represented in percentage from 0% to 100%, here it is given as a continuous variable between 0 and 1

Time of day Time of day of each measurement. Integer ranging from 2 to 20 each day.

Dew point The dew point is the temperature to which air must be cooled to become saturated with water vapour. Measured in Celcius (°C).

Day ID ID of the day, 1 corresponds to January 1^{st}, 366 to December 31^{st} etc.

Pressure Air pressure, measured in Pascal or Newton per square meter ($Pa = N/m^2$)

Wind bearing Direction of wind. Measured in degrees from 0 to 360.

Wind speed Wind speed, measured in meters per second (m/s).

Chapter 3

Data quality

Real-world data often come with problems. Whether it is malfunctioning equipment or human error, there will be missing values and other errors in any non-generated data. There are some key differences between these two types of errors that we need to take in account.

Missing data occur in all real-world data sets, as neither humans nor machines are perfect. Missing values of a variable are usually denoted by some value that is impossible for the variable to attain, e.g. negative irradiance or power, any temperature reaching -300 °C or simply "not a number"[NaN]. Alternatively, a good method is to impute a value that is extremely unrealistic, e.g. -50 °C in Bari, Italy. For this reason, imputed missing values are often easy to spot, as long as they are consistently noted throughout the set.

This is where problems occur in the Bari data set as the missing data seems to be mostly replaced by 0, which is a problem in itself since many of the variables, such as irradiance, will realistically be 0 at many hours through the day. In addition to this, it does not seem like the choice of representing missing data as 0 is done consistently throughout the set, as we will see in Sections 3.3 and 3.4.

Imprecise/erroneous data often occur in data sets as well. This is usually a result of malfunctioning equipment or imprecise measurements. This also includes non-zero measurements where the true value is zero, which we need to take into account when identifying missing values later in this chapter. These errors are much more difficult to identify in a data set than missing values, as they may appear within the expected range of data.

In our sets it is hard to quantify exactly what constitutes missing and erroneous data in all cases, because data that appears erroneous may simply be imputed missing data. We may mislabel errors as missing data and vice versa, but all points are treated the same in our imputation algorithms in Chapter 4.

Errors and missing data may only be a problem if the frequency exceeds a point where it will affect the accuracy and reliability of a model. A small number of errors in a data set may simply be discounted as noise.

In this chapter we will look at the quality of the five variables in our data set that we consider most influential in the power production - irradiance, weather temperature, plant temperature, cloud cover and humidity. We will also look at our target variable, power, which is what we aim to forecast. We will argue for a reliable criteria for classifying a point as missing data or as an error, and look at the frequency and distributions of these errors in the sets.

True and false positives and negatives

In this paper, we characterize non-erroneous points as ”positives”, and erroneous points (missing values, imprecise or in any way erroneous data) as ”negatives”. ”True” and ”False” refer to the labelling of these points. If we falsely label a non-erroneous point as an error, it is a false negative, and vice versa.

	Non-erroneous value	Erroneous value
Not labeled as error	**True Positive**	**False Positive**
Labeled as error	**False Negative**	**True Negative**

Table 3.1: True and false positives and negatives

3.1 Irradiance

When we talk about solar panels, what is usually meant is photovoltaic [PV] solar panels. These panels convert energy from the sun (irradiance) into a flow of electrons, through what is known as the photovoltaic effect. Therefore we know that if we have 0 irradiance, it is physically impossible to generate power from the photovoltaic effect. This means that in any point where we have power but no irradiance, or vice versa, one of the two variables must be wrong. We need to figure out which of the two variables are incorrect.

The goal is to not label points where we have no irradiance, and very low (but non-zero) power, as errors in irradiance. This is because the early morning will have close to no irradiance, and the corresponding power may be very low, but may still be within the margin of error for the instrument measuring power.

A reasonable assumption is that the instrument measuring power has a certain accuracy, say α, which means that with x% certainty, a measurement, P, will represent an exact value that lies in the interval $[P-\alpha, P+\alpha]$. This means that if our measurement $P > \alpha$ we can with x% certainty say that the exact value for P is greater than zero, in which case the irradiance should also be greater than zero. Our problem is that we do not know the value of α, and we will therefore try different values, which we call our threshold.

The possible problems that we may face is a high number of *false negatives* and *false positives* in our classification. If a point were to be identified as a false positive, it will remain the same value as it was originally, and will therefore not decrease the "accuracy" of the data set. On the other hand, a false negative may distort the data by removing non-erroneous data and replace it by a value that may only be less accurate.

We now want to specify what a reasonable threshold for power should be. To get an overview of the impact of the threshold value, we will first implement three thresholds, TH= $\{20, 50, 100\}$, interpret the result, and see if there are any possible problems that may occur with any of the thresholds we have chosen. See Table 3.2 and Figure 3.1.

	TH=20	TH=50	TH=100
Set 1	264 (3.8%)	202 (2.9 %)	150 (2.16%)
Set 2	872 (12.54%)	730 (10.5%)	629 (9.05%)
Set 3	30 (0.43%)	28 (0.4%)	24 (0.35%)

Table 3.2: Number of erroneous values in irradiance

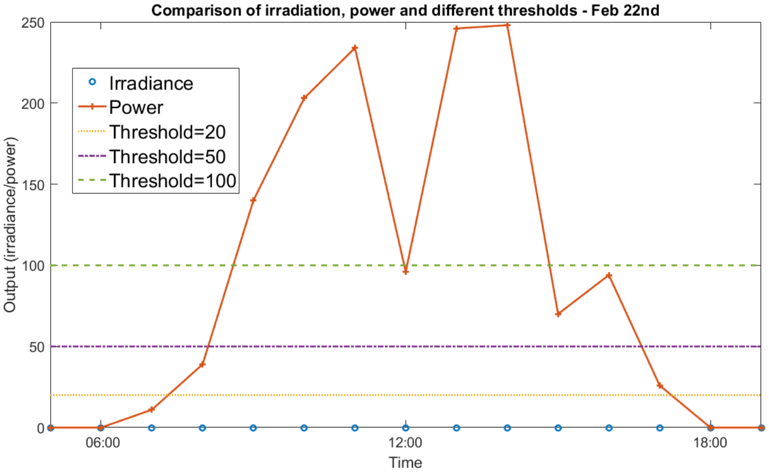

Figure 3.1: Irradiance and power from 05:00 to 19:00 Feb 22nd - set 1, together with the three thresholds. A day with relatively low power production overall, which illustrates the threshold issue well.

From Figure 3.1 it seems clear that most of the points in question (irradiance is zero, power is non-zero), will be correctly labelled as errors in the peak hours of the day as long as the threshold is not too large. The data in Figure 3.1 does illustrate a common day with relatively low power production, the days we are most likely to label false negatives. We can see that choosing threshold larger than 20 will omit several points in which the power production is growing substantially, and it therefore seems reasonable to set the threshold TH=20.

Another important factor is the distribution of errors. We can see from Figure 3.2 that in set 1 and 2, we have a large number of zeros clustered in the late winter/early spring part of the year, while set 3 has a few erroneous points scattered throughout the set.

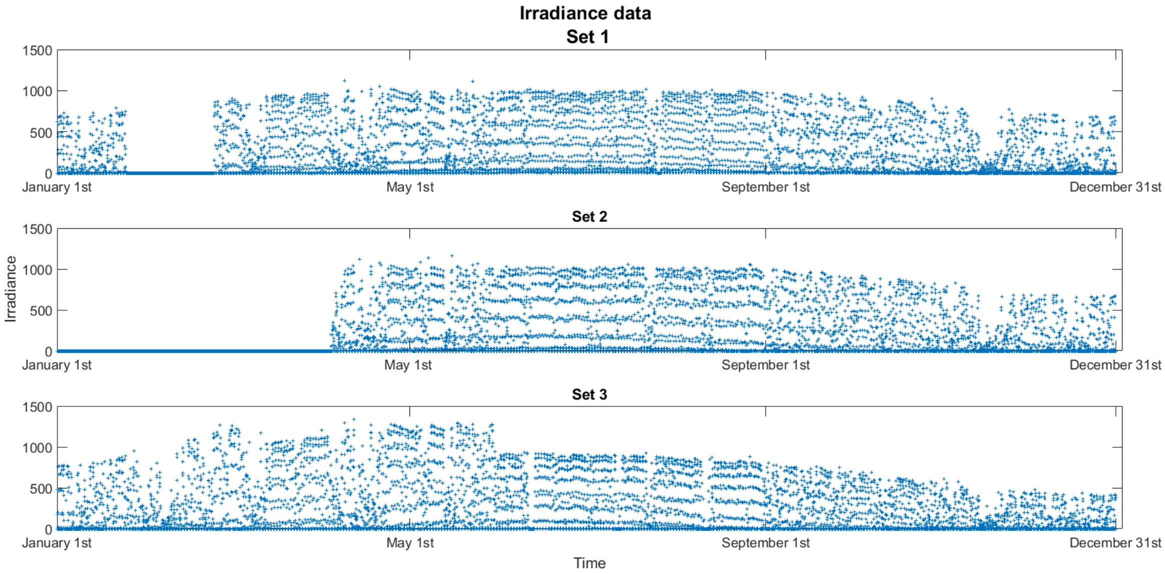

Figure 3.2: Irradiance throughout the year, all sets.

3.2 Power

As we mentioned in the beginning of Section 3.1, power mostly depends on irradiance, as it is physically impossible for us to have power when we have no irradiance. We therefore utilize the error criteria from the previous section, by labelling any data point where we have no power while irradiance exceeds a certain threshold, as an error. We implement the same TH-values as in the previous section, of which results can be seen in Table 3.3.

	TH=20	TH=50	TH=100
Set 1	7 (0.1%)	7 (0.1 %)	7 (0.1%)
Set 2	16 (0.23%)	13 (0.19%)	12 (0.17%)
Set 3	26 (0.37%)	16 (0.23%)	8 (0.12%)

Table 3.3: Number of erroneous values in power

Just as we argued in the previous section with regards to irradiance, the possible mislabelling (particularly false negatives) will most likely occur in the early mornings and late evenings, when the irradiance is low, but non-zero, and the power is equal to zero. However, as we can see from Table 3.3, even at the smallest TH we still have less than 0.5% errors in all sets. These points are distributed relatively evenly throughout the sets, and are not clustered at any time.

The low frequency of errors leads us to conclude our best choice of action here is to not modify anything, and let the few erroneous points remain in the set.

3.3 Weather Temperature

There seems to be inconsistency in how missing data is registered in the weather temperature data, as missing data is incoded by a value of either 0, -1 or -2 degrees. In this case it does not seem like true measurements should fall within the range $\{-2, -1, 0\}$, with the exception of February in set 3. This can be seen from the sets themselves in Figure 3.4.

This makes it less likely to mislabel imputed missing data point, since no true measurement should be 0 °C or colder, and so we choose to label any value for which the weather temperature $wT \leq 0$ as an error.

There are three key observations we can make by inspecting the data. The first of which appears when we count the number of values $wT^k \leq 0$.

Set	# of $wT \leq 0$	% of total # of values
1	2928	42.11%
2	2928	42.11%
3	2928	42.11%

Table 3.4: Number of values in weather temperature less than or equal to zero

From Table 3.4, we notice all sets have the exact same number of missing values. The reason for this is not known, but it may be the result of a computer or coding error. It should be noted that the indices k of the missing data are not *equal* across the sets, but the sets share exactly 760 erroneous indices (i.e. data points k where $wT^k \leq 0$ in all sets). To investigate this further, we look at the correlation coefficients of the sets, displayed in Table 3.5. These results signify a strong statistical relationship in the weather temperature between the sets. They may be data from the same source that is adjusted slightly between sets based on the location of the plants, or there may be another reason for this strong relationship that we do not know.

	Set 1	Set 2	Set 3
Set 1	1	0.9495	0.9392
Set 2		1	0.9288
Set 3			1

Table 3.5: Correlation coefficients of weather temperature between sets

The second observation is specific for set 1, as it seems that the distribution of missing values has a distinct pattern in this particular set. It seems that they mostly appear in the early morning throughout the year, from 02:00 to 09:00, which can be seen in Figure 3.3. No such pattern appears in other sets. It is hard to tell why this specific phenomena occurs, but it may be the result of equipment shutting down during the night and remain off until people return in the morning.

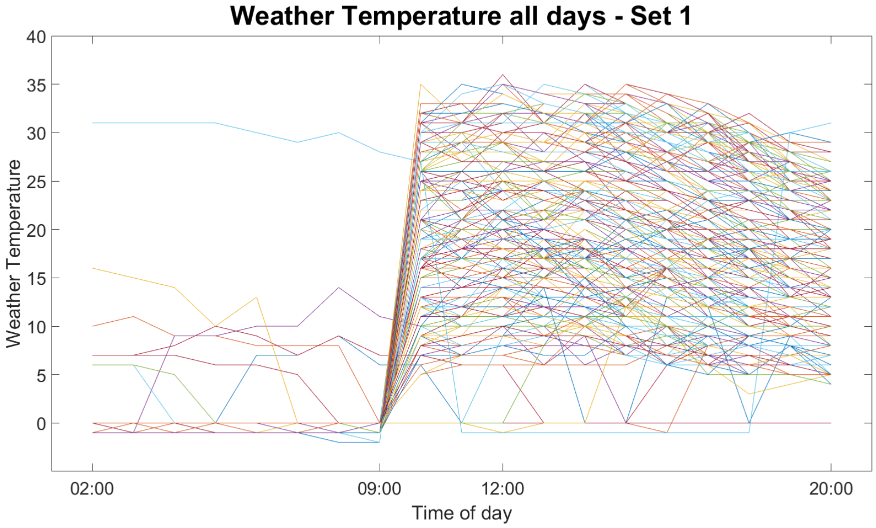

Figure 3.3: Weather temperature for all days in set 1. In this case, the majority of the erroneous points occur between 02:00 and 09:00

The third observation is on the general distribution of missing values throughout the set. As we can see from Figure 3.4, the erroneous values are relatively evenly spread out through the year.

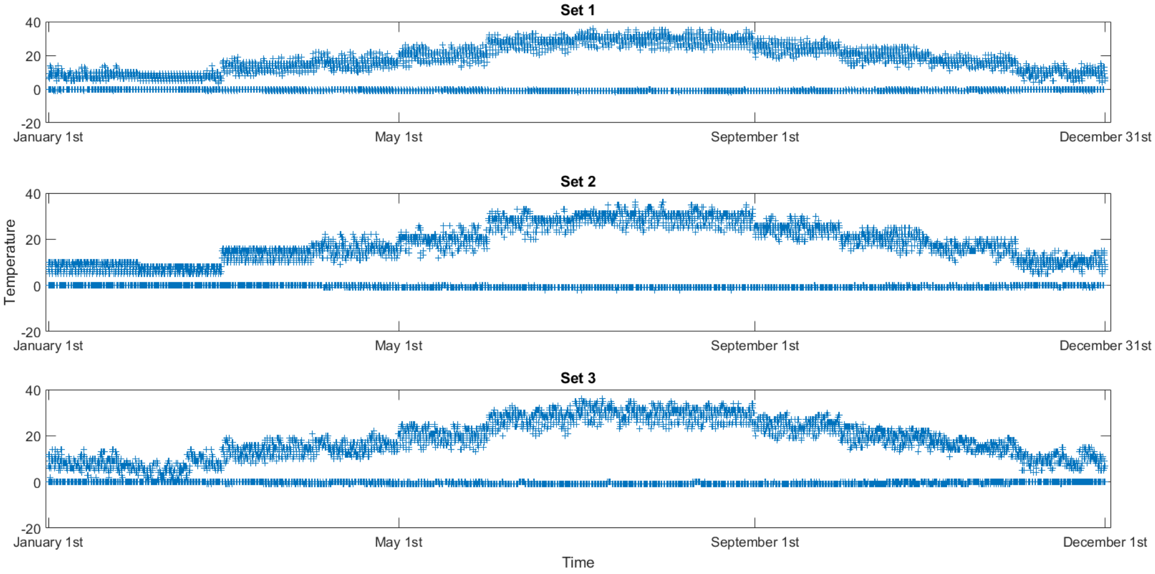

Figure 3.4: Weather temperature through the year for all sets. We can see that the erroneous values are spread our through the year in all sets.

3.4 Plant Temperature

In the plant temperature, it seems that most of our problems come from missing data points which have been imputed as 0, -1 or -2 degrees, just as in weather temperature. However, identifying these missing values is not as simple as in weather temperature in certain parts of the year. This is because it seems from the data, to be realistic for the plant temperature to approach zero or negative values in the winter months. This makes it more likely for us to obtain false negatives.

Set	# of $pT \leq 0$	% of total # of values
1	627	9.02%
2	1804	25.94%
3	194	2.79%

Table 3.6: Number of values in plant temperature less than or equal to zero

As we see from Table 3.6, there is a large difference in the number of data that we assume to be imputed missing data. This difference can also be seen in Figure 3.5. From the plots, we can see a clusters of errors in sets 1 and 2, while set 3 only has a few which are relatively evenly distributed throughout the year.

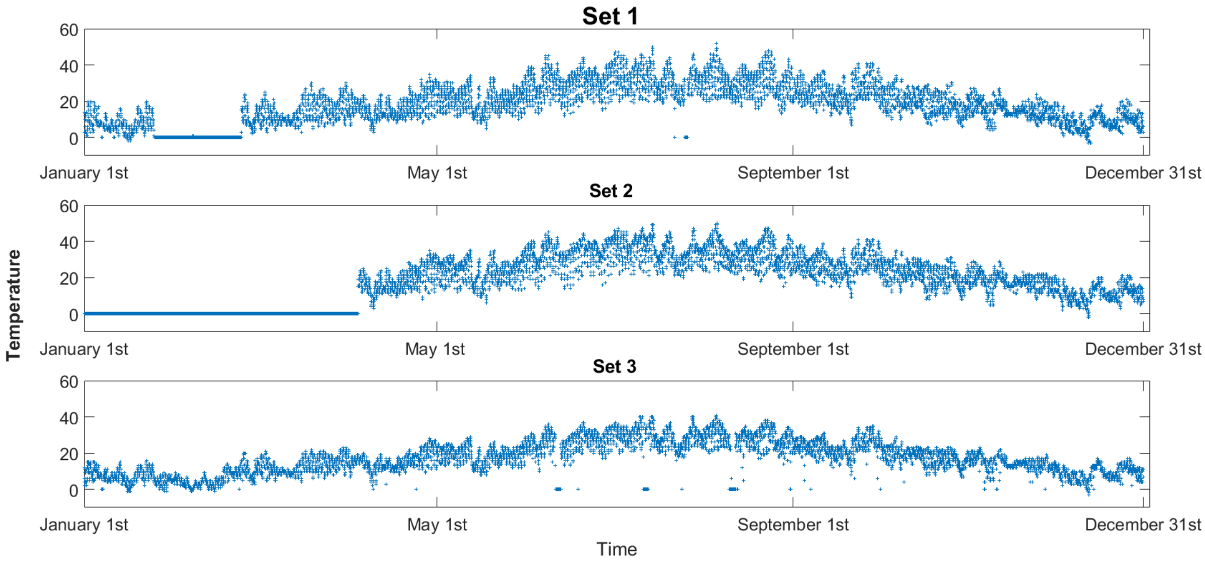

Figure 3.5: Plant temperature through the year for all sets. Errors appear clustered for set 1 and 2.

Even though several non-erroneous data points are ≤ 0, we would argue that labelling all points that are less than or equal to zero as errors may be an adequate solution. Firstly, since the number of possible mislabelled points is so low (set 1 and 2 clearly has a large number of missing values, whilst set 3 has less than 3% values less than or equal to zero). Secondly, the times where it seems realistic for the plant temperature to be zero or negative, all the sets behave similarly. This property may help us not distorting the data, as we will see in Chapter 4.

3.5 Cloud Cover

Cloud cover is not as straightforward to measure compared to other variables like weather temperature, power etc. What we *can* define is that we need a maximum (complete) cloud cover and minimum (clear sky). This is expressed as a continuous variable that ranges between 0, minimum, and 1, maximum.

In the Bari data set, we have some trouble assessing the criteria set for measuring cloud cover. What constitutes a ”complete cloud cover” or a ”mid-range” (as in corresponds to the value 0.5 or near it) cloud cover we do not know.

When we process our data, one particular feature sticks out. There seems to be an over-representation of the value 0.310, displayed in Table 3.7. We are not sure about the reason for this, but it may be related to the mean of the values in the data set, which can be seen in the fourth column. This may be the value they use to replace missing values in the set, or it may be a result of some other action upon the data.

Set	# of cc=0.310	% of total # of values	Mean value ∀ cc ≠0.310
1	2716	39.06%	0.3060
2	2919	41.98%	0.3038
3	2655	38.18%	0.3083

Table 3.7: Frequency of the value 0.310 in cloud cover and mean of all other values.

As stated earlier, our knowledge of cloud cover is limited, but what we do know is that the cloud cover is related to the relative humidity. We can look at the correlation when we include or exclude the 0.310 values, and see if the correlation grows substantially when the suspected errors are excluded. As we can see from Table 3.8, the correlation between variables grows when we exclude the dubious values in question.

Set	Corr. including 0.310-values	Corr. excluding 0.310-values
1	0.3677	0.4499
2	0.3502	0.4331
3	0.3762	0.4574

Table 3.8: Correlation coefficients of cloud cover and humidity

Now that we know that the value 0.310 distorts the data somewhat by reducing the correlation with the relative humidity, it seems that some of these values may be missing values imputed by a value approximately equal to the mean of the set. Some of the values are most likely correct data, but it is hard to tell the difference. Thus, we must choose to either assume they are all erroneous or that they are all correct. We conclude that most of them are imputed missing values, and that we therefore classify them all as erroneous.

3.6 Relative Humidity

As stated in Chapter 2, relative humidity is the ratio of partial pressure of water vapor to the equilibrium vapor pressure of water at a given temperature. Relative humidity depends on temperature and the pressure of the system of interest. The reason we look at relative humidity is because of its inherent relationship with cloud cover.

While cloud cover is hard to measure, as discussed in the previous section, relative humidity is measured simply by using a hygrometer, which make the measurements more reliable than cloud cover. The amount of power a solar panel produces is highly impacted by the cloud cover, which is again dependent on the relative humidity. A relative humidity of 100% indicates a complete cloud cover and rain.

However, our lack of general knowledge of the physical processes behind relative humidity makes it very hard to identify errors in the humidity data, although this does not indicate that there are none. We do not know what a realistic rate of change in humidity is, which could have been an error indication. In addition to this, the data itself does not seem riddled with errors like many of the other variables in our data, in that there are no clusters (”holes”), i.e. a span of time where we have constant humidity, or other visual indicators of errors.

Therefore, we will neither classify any errors, nor aim to improve these data in any way, since we might only distort the data.

Chapter 4

Data imputation

Chapter 3 described the errors and missing values, and so we now seek to improve the data in any way we can. We will use the information gathered in Chapter 3, and attempt to formulate methods for replacing missing or erroneous data. Our goal is to improve the quality of the data so that when we attempt to forecast, the preprocessing of data improves our results, compared to results based on the original data.

We will not be looking at cloud cover or humidity in this chapter. For cloud cover we have labelled what we assume are imputed missing values, but we do not have the knowledge needed to replace any of these missing values with more valid measurements. For humidity we failed to provide any meaningful criteria for identifying missing or erroneous data. We will also not look to impute any power data, since the number of errors were so low for this variable, as discussed in Section 3.2, but the variable will be utilized in Section 4.2.

Irradiance, weather temperature and plant temperature had different types of errors. The plant temperature data from set 1 and 2 had large "gaps", several weeks of errors in a row, and relatively few errors scattered throughout in all sets. The errors in irradiance were similar to plant temperature, while the weather temperature had errors distributed evenly throughout the year for all sets, with errors distributed evenly through all days for set 2 and 3, but clustered in the morning for all days in set 1.

Thus, we have two types of error distributions for these three variables, clustered and scattered. We also know that the values of these three variables rely on different factors. We therefore take three different approaches in order to improve the data.

In Section 4.2 we will look at the imputation of irradiance, and how we can replace erroneous values in one variable by converting the value of another related variable. In Section 4.3 we will look at the weather temperature and construct a day-and-year periodic regression problem, from which we can replace missing values. Finally, in Section 4.4 we will look at the plant temperature of all three sets and construct a replacement algorithm in which we may replace missing values by extracting values from different sets.

4.1 Preliminaries

Before describing the algorithms utilized for the different variables, we will define some terms that will be used, and establish the notation used in the different sections.

4.1.1 Standard deviation

Standard deviation, often represented as σ, is a measure that is used to quantify the amount of variation in a set of values. If the standard deviation is small, it indicates that values in the data set are generally close to the mean of the set, while a large standard deviation indicates that values are spread out over a wider range, further from the mean.

4.1.2 Regression and interpolation

Regression is trying to find a function $y(x)$ that fits data sampled from that function with errors. **Interpolation** is when we estimate intermediate values for a function, usually used when data is accurate but intermediate values are missing. A solution to a general regression problem can be expressed as,

$$\begin{array}{lcr} & y(x) = \sum_{j=0}^{k} a_j \phi_j(x), & \\ \text{where} & \{\phi_j(x)\}_{j=0}^{k} & \text{forms a linearly independent basis.} \end{array}$$

Here $\phi_j(x)$ are the functions that acts upon x, and a_j are the scalar coefficients that we wish to estimate, which correspond to the $j = \{1, 2, ..., k\}$ functions ϕ_j. The choice of ϕ is dependent on the problem we wish to solve. We get the optimal values of a_j by minimizing the residual r,

$$r = \arg\min_{a_j} \sum_{i=0}^{n} \left[\hat{y}(x_i) - \sum_{j=0}^{k} a_j \phi_j(x_i) \right]^2 \tag{4.1}$$

Where $\hat{y}(x_i)$ are the true measurements and n is the number of data points.

4.1.3 Multiple imputation

The purpose of multiple imputation is to generate possible values for missing values, thus creating several "complete" sets of data. The number of imputations is decided by the user, and the combined results of these imputations are usually more accurate than single imputation methods. If we do i imputations for all values we seek to impute, we get i complete data sets.

Random number

When we do multiple imputations of a variable, we will use a random number ξ to sample from the range of expected values. This random number is taken from a normal distribution with mean 0 and a standard deviation $\sigma = 1$. We will then multiply this random number by the standard deviation of the value that we seek to impute. For each value we impute, we generate a new random number ξ.

4.1.4 Notation

Sets:

$$\begin{aligned} K &= \{1, 2, ..., 6954\} && \text{Indices of all data} \\ S &= \{1, 2, 3\} && \text{Indices of sets (plant 1,2 and 3)} \end{aligned}$$

$$\begin{aligned} \hat{K}_{irr} &\subseteq K && \text{Indices of non-erroneous irradiance data} \\ \hat{K}_{pw} &\subseteq K && \text{Indices of non-erroneous power data} \\ \hat{K}_{IP} = \hat{K}_{irr} &\cap \hat{K}_{pw} && \text{Indices in which neither irradiance nor power are erroneous} \end{aligned}$$

$$\begin{aligned} \hat{K}_{wT} &\subseteq K && \text{Indices of non-erroneous weather temperature data} \\ \hat{K}_{pT} &\subseteq K && \text{k-indices of non-erroneous plant temperature data} \\ \hat{S}_{pT} &\subseteq K && \text{s-indices of non-erroneous plant temperature data} \\ \hat{K}_{wT,pT} &\subseteq K && \text{Indices in which neither weather- nor plant temperature are erroneous} \end{aligned}$$

Data:

$$\begin{aligned} I^k &= && \text{Irradiance data,} && k \in K \\ P^k &= && \text{Power data,} && k \in K \\ t^k &= && \text{Time of day,} && k \in K \\ wT_k &= && \text{Weather temperature,} && k \in K \\ pT^{k,s} &= && \text{Plant temperature} && k \in K, s \in S \end{aligned}$$

Variables:

$$e^k_{irr} = \begin{cases} 1, \text{if } k \notin \hat{K}_{irr}, \\ 0, \text{if } k \in \hat{K}_{irr} \end{cases}$$

Binary error vector for irradiance, where 1 corresponds to data point k being erroneous.

$$e_{pw}^{k} = \begin{cases} 1, \text{if } k \notin \hat{K}_{pw}, \\ 0, \text{if } k \in \hat{K}_{pw} \end{cases}$$

Binary error vector for power, where 1 corresponds to data point k being erroneous.

$$e_{wT}^{k} = \begin{cases} 1, \text{for } k \notin \hat{K}_{wt}, \\ 0, \text{ otherwise} \end{cases}$$

Binary error vector for weather temperature, where 1 corresponds to data point k being erroneous.

$$e_{pT}^{k,s} = \begin{cases} 1, \text{for } (k,s) \notin (\hat{K}_{pT}, \hat{S}_{pT}) \\ 0, \text{ otherwise} \end{cases}$$

Binary error vector for plant temperature, where 1 corresponds to data point (k,s) being erroneous.

$$R^{k,s} = \begin{cases} 1, & pT^{k,s} \text{ has been replaced} \\ -1, & pT^{k,s} \text{ cannot be replaced by data from any set} \\ 0, & \text{otherwise} \end{cases} \qquad k \in K, s \in S$$

Replacement matrix for plant temperature

4.2 Irradiation and Power - Converting values of one variable to another variable

In this Section we will describe the replacement algorithm for irradiance, but first we need to investigate the relation between irradiance and power for a solar panel.

4.2.1 The relation between irradiation and power

The power of a solar panel is mostly relying on irradiance from the sun. Other factors such as cloud cover can limit the amount of power we get by reducing the irradiance, whilst the temperature of the panels themselves may reduce the power output without affecting the irradiance. From [6.2.1 in Duffie & Beckman 2013] we have an equation that describes the energy output P of a collector of area A, where U is the heat transfer coefficient.

$$P = A\left[I - U(pT - wT)\right] \tag{4.2}$$

We wish to simplify this equation, because the three variables in the equation that are needed in order to estimate the power production are not reliable in all sets. Weather temperature has the same number of erroneous points through all sets and is therefore equally unreliable throughout, see Table 3.4, while the number of errors in irradiance and plant temperature vary between sets, where set 3 has the least amount of errors and set 2 has the most for both variables.

When we have unreliable irradiance and temperature measurements at times, any estimation that relies upon these values are highly questionable. We will therefore exclude any index $k \notin \hat{K}_{wt,pT}$ when looking at the relation between the two variable, plant and weather temperature, and exclude any index $k \notin \hat{K}_{irr}$ when we look at the magnitude of the irradiance.

We would first like to get an impression of the magnitude of $(pT - wT)$, and so we estimate the arithmetic mean and standard deviation of the difference between the two variables, which can be seen in Table 4.1.

Set	Mean value μ	Standard deviation σ
1	1.4	5.3
2	2.6	5.9
3	-1.1	4.4

Table 4.1: The mean of the difference between plant temperature and weather temperature

From the results in Table 4.1 we see that the difference between the two variables wT and pT are relatively close to ± 1 for set 1 and 3, the sets with the least number of errors, and therefore arguably the best estimates. The standard deviation is around 5 for all sets, which is not very large, but also relatively far from zero. This means that most values of $(pT - wT)$ should fall within ± 10.

We will now look at the difference between U and I^k. If we can show that $I^k >> U$, we may conclude that $I^k >> U(pT^k - wT^k)$, and simplify accordingly.

The heat transfer coefficient U is not known to us from the data set, but in several examples in [Duffie & Beckman (e.g. 6.4.1 and 6.4.3)], they describe a collector with a heat transfer coefficient of approximately $7W/M^2$°C. In other similar examples, irradiance is many times larger than U, but we want to compare it to the irradiance data from the set.

The values of I^k ranges from 0 to almost 1000 throughout the year. We do have a high frequency of 0-values throughout the set which are both errors and true measurements when the sun is not shining. The mean value of these non-zero elements in I^k is > 300 for all sets. If we compare this mean of irradiance μ_{irr} with U we get $\mu_{irr}/U = 300/7 \approx 43$. We know that most values of $(pT - wT)$ should fall within ± 10, and so we multiply the largets value, 10, by U, which gives us $\mu_{irr}/U * 10 = 300/70 \approx 4.3$. To test this in more detail, we computed I, U on some reasonable parameters, which indicated that I generally is 10 to 100 times larger than U.

Based on these arguments, we choose to simplify (4.2) to (4.3).

$$P = I * A, \tag{4.3}$$

The area of the panels, A is unknown to us, but we know that it should be constant since the area of the panels does not change throughout the year. We estimate the value of A for each set through linear regression, which also gives us the standard deviation σ.

While doing this we of course need to take account of the errors that occur when converting values from one variable to the other. We have substantially simplified (4.2), which means that we have introduced model errors ϵ^k. However, we argue that these model errors will not be very crucial when our result would, in any case, be highly impacted by data errors. When we account for the errors in (4.3), we have

$$\begin{aligned} P^k = A * I^k + \epsilon, &\quad \forall k \in K, \\ |\epsilon^k| < \epsilon, &\quad \epsilon \in \Re \end{aligned}$$

We will refer to this value of A as our conversion factor.

4.2.2 Replacement algorithm

We will now illustrate the replacement algorithm in pseudocode. We do ten imputations for each value in irradiance, and thereby attain $3 * 10$ data sets with different irradiance data.

Algorithm 1 Irradiance replacement algorithm

This is done one set at a time.

Input: A, P^k, I^k,e^k_{irr}, σ
Number of imputations: 10
Output: I^k_i
$k = \{1, 2, ..., 6954\}, i = \{1, 2, ..., 10\}$

ξ is a random number from a normal distribution with mean 0 and standard deviation 1.
σ is the standard deviation from our linear regression to estimate A.

```
for $k = 1, 2, ..., 6954$ do

    for $i = 1, 2, ..., 10$ do
        Generate $\xi$;
        $I^k_i = (1 - e^k_{irr})I^k + e^k_{irr}\frac{1}{A}(P^k + \xi\sigma)$ ;
    end for
end for
```

NOTE: We can not impute irradiance from power if the power is also erroneous at that point. However, from our definition of error, it is impossible for both power and irradiance to be erroneous in the same data point k

4.3 Weather Temperature - Regression and interpolation within a data set

If we were dealing with relatively isolated missing data points in the weather temperature, we could simply replace these points with values acquired from simple interpolation using the respective points neighbours. As the frequency of missing data is so high, we need a more thorough algorithm where we approximate the missing values by constructing a mathematical model that attempts to simulate the weather temperature.

As described in Chapter 2, we have 19 data points each day for 366 days, where the first measurement each day is at 02:00, and the last is at 20:00. Since we are constructing a model of the weather temperature, a periodic process, we need to construct our model on 24 hours a day. We then aim to solve the regression problem such that our model best fit the data we have. Our function $y(t)$ need to inhabit two different cycles, one daily temperature cycle and one yearly temperature cycle.

$$y(t) = a_0 + a_1 \sin(\frac{2\pi(t + d_1)}{24}) + a_2 \sin(\frac{2\pi(t + d_2)}{24 * 366}) \tag{4.4}$$

Where

t is the time in hours from 01:00 January 1st to 24:00 December 31st, $t \in \{1, 2, ..., 24 * 366\}$.

a_0 defines the yearly average temperature, i.e. the value of $y(t)$ when both sine terms are equal to zero.

a_1 is the daily amplitude, defined as the difference between the daily maximum and minimum temperatures.

a_2 defines the minima and maxima through the year, yearly temperature variance.

d_1, d_2 are the phase factors, which put the maximum and minimum at the right time of the day/year.

Approximations for these parameters could be found by solving a non-linear least squares, and in order to get the correct covariance estimates (a measurement of joint variability between variables), we should estimate all values simultaneously. However, due to a lack of time, we have chosen a more ad hoc approach, using the fact that the 5 parameters do represent specific physical phenomenon. All unknown parameters are physical features we have some knowledge about, which means that we can "manually" estimate it using our data.

We begin by estimating the yearly average temperature a_0, which we do by using wT from the data. We calculate the mean value of the weather temperature wT, excluding all erroneous points, and get the results displayed in Table 4.2.

	Maximum	Minimum	Mean value	Standard Deviation
Set 1	36	3	18.6	8.1
Set 2	36	5	18.7	8.1
Set 3	36	1	18.6	8.3

Table 4.2: Mean, maximum, minimum and standard deviation of weather temperature, all sets

From the mean values in Table 4.2 we get a good estimation for a_0. The mean values of the different sets are so similar we can use one value for all sets.

$$a_0 = 18.6 \tag{4.5}$$

We will now attempt to approximate the value of a_2, the yearly variance, and d_2, the parameter that adjusts what our model will set as the time of year when it is coldest and warmest. To do this, we first need to identify the coldest and the warmest parts of the year. This seems approximately equal for all sets. The coldest period is from mid January through February, while the warmest part of the year seems to be mid July into August. The sets have almost the same average temperature in these periods, approximately 7 °C in Jan/Feb and 29 °C in July/Aug, which can be seen in Table 4.3.

These results give us the variance through the year a_2. We look at the values from each set. We want the difference between the largest values in our $a_2 \sin(...)$ term to be $(\mu_{july/aug} - \mu_{jan/feb})/2, s = 1, 2, 3$. These values are again so similar, we choose only one for all sets.

$$a_2 = \begin{cases} 10.5, \text{for set 1} \\ 10.7, \text{for set 2} \\ 10.9, \text{for set 3} \end{cases} \approx 10.7 \tag{4.6}$$

We can also estimate the value of d_2 since we now know that the coldest day of the year should be in early February. A reasonable assumption is February 1^{st}, which would make the warmest day for our function August 3^{rd}, which also seems reasonable. Since our model is based on t (which represents hours from Jan 1^{st} 01:00 to December 31^{st} 24:00), we must find the t-value corresponding to the day exactly in the middle of these two days, which corresponds to May 1^{st}. We choose the time of day to be 12:00, arbitrarily, since this exact choice of hours does not have a large impact on the function as a whole. This gives us the value for $t = 24 * (31 + 29 + 31 + 30) + 12 = 2916$ such that

$$10.7\sin(\frac{2\pi(2916+d_2)}{24*366}) = 0 \quad \rightarrow 2916 + d_2 = 0$$
$$\rightarrow d_2 = -2916 \tag{4.7}$$

Now all that remain are the parameters corresponding to daily variations, a_1 and d_1. To estimate these parameters, we need to look at the temperature in smaller time frames than what we have done do far, and also see if this variance changes throughout the year. d_1 does not require as much calculations as a_1, so therefore we will begin by estimating d_1.

We want to adjust d_1 so that the peak values of $a_1\sin(\frac{2\pi(t+d_1)}{24})$ correspond with the warmest and coldest parts of the day. Therefore, we need to consider when the warmest and coldest parts of the day are. From the data, it seems like daily temperature peaks sometime between 12:00 and 18:00 each day. The actual peak may vary from day to day, but 15:00 seems like a fair estimate.

We cross check this with our data by extracting temperatures at time 15:00 through the year and comparing it to the temperature at all other times. The temperatures are fairly similar to those at time 14:00 and 16:00, but generally higher than all others. We therefore choose to set 15:00 as the peak hour of the day for temperature. This means that our daily sine-function has its roots at 09:00 and 21:00, and its minimal value will be at 03:00. We can now use this to estimate d_1.

$$a_1\sin(\frac{2\pi(9+d_1)}{24}) = 0 \quad \rightarrow 9 + d_1 = 0$$
$$\rightarrow d_1 = -9 \tag{4.8}$$

We estimate the mean temperature values each month, and look at the corresponding standard deviation. We plotted the mean temperature values and the values within one standard deviation for set 2 in Figure 4.3 to illustrate how this changes throughout a year. We use these values to estimate a daily variation in temperature. Since the standard deviation changes each month, we redefine $a_1 \rightarrow a_1^M$, (where M corresponds to each month) $M = \{1, 2, ..., 12\}$. We will use a_1^M to sample for our multiple imputations by utilizing the random value ξ acting upon a_1^M, where ξ is normally distributed with mean 0, as described in Section 4.1.3.

$$a_1^M = \xi\sigma^M \tag{4.9}$$

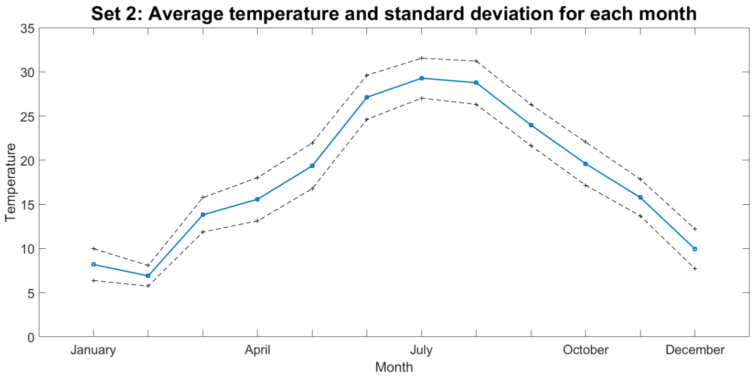

Figure 4.1: Monthly mean temperatures (blue line) and corresponding standard deviation (black dotted lines) for set 2

	Set 1: Mean	σ_M	**Set 2:** Mean	σ_M	**Set 3:** Mean	σ_M
February	7.4	1.3	6.9	1.2	6.6	2.9
July	29.2	2.5	29.3	2.3	29.6	2.8

Table 4.3: The mean values and standard deviations of non-erroneous elements in February and July.

We compute the standard deviation σ_M for each month and set, of which some results can be seen in Table 4.3. These values are used in our function $y(t)$ to ensure that the daily variations is correct.

From (4.5) through (4.9), we have our estimation for $y(t)$,

$$y(t) = 18.6 + \xi\sigma_M \sin(\frac{24\pi(t-9)}{24}) + 10.7\sin(\frac{24\pi(t-2916)}{24 * 366}) \tag{4.10}$$

Where σ_M, the standard deviation for month M is unique to each set. ξ is a random number from a normal distribution with mean 0 and standard deviation 1.

Just like in Section 4.2, we will use 10 imputations for each set, which means we will get 10 new weather temperature vectors, and thereby 10 new complete data sets.

Note: in the algorithm we slightly change the notation for $y(t)$ and define it discretely as y_i^t, where i represents one of the ten imputations.

Algorithm 2 Weather temperature replacement algorithm

This algorithm is run one set at a time.

Inputs: wT^k, σ_M, e_{wT}^k
Number of imputations: 10
Outputs: wT_i^k
$k = \{1, 2, ..., 6954\}, t = \{1, 2, ..., 8784\}, i = \{1, 2, ..., 10\}$

Using σ_M we construct a vector s_σ^t with values corresponding to σ_M for t within its respective month M
ξ is a random number from a normal distribution with mean 0 and standard deviation 1.

```
for $t = 1, 2, ..., 24 * 366$ do
    for $i = 1, 2, ..., 10$ do
        Generate $\xi$;
        $y_i^t = 18.6 + \xi s_\sigma^t \sin\left(\frac{2\pi(t-9)}{24}\right) + 10.7 \sin\left(\frac{2\pi(t-2916)}{24*366}\right)$;
    end for
end for
for $d = 1, 2, ..., 366$ do
    for $j = 1, 2, ..., 19$ do
        for $i = 1, 2, ..., 10$ do
            $Y_i^{(d-1)*19+j} = y_i^{((d-1)*24+j+1)}$ ;
        end for
    end for
end for
for $k = 1, 2, ..., 6954$ do
    for $i = 1, 2, ..., 10$ do
        $wT_i^k = wT^k(1 - e_{wT}^k) + Y_i^k e_{wT}^k$;
    end for
end for
```

4.4 Plant Temperature - Replacing data with corresponding data from different sets

As we saw in Section 3.4, the errors that occur in the plant temperature [PT] are very different from the errors in weather temperature [WT]. Since the PT will not necessarily behave in the same way as the WT due to the differences in heat conduction in crystalline silicone and air, our method of extracting and imputing yearly average temperature data in the previous section will not be a feasible solution for this problem.

In addition to this, there is a difference in the frequency and distribution of errors. While the erroneous values in WT are spread out relatively evenly throughout the year, errors in PT are mostly clustered in the first 100 days of the year and mainly occur in sets 1 and 2 which can be seen in Figure 3.5. These large holes in the dataset makes a regression approach near impossible and definitely highly inaccurate. We therefore choose to utilize a different method.

Since our plants are located in the vicinity of each other, we can replace missing values in one set with values from another plant, given that at least one of the data points from these two plants are non-erroneous. If the case is that only one of the three sets has an erroneous point at data point k, we can replace the erroneous value with the mean of these two non-erroneous values, so that we retain as much variance as possible.

Before we proceed with this method, we should inspect the relation between the plant temperature in the sets. We exclude any value that is erroneous in at least one of the sets, and look at the absolute difference between the values in the sets. We do not have any definition on what constitutes a "small difference", but we would assume it would be less than one standard deviation for (non-erroneous) values in each set. This standard deviation is 8 or greater for all sets, so we look at temperatures that are within ±8 °C of each other. Results are given in Table 4.4.

Choice of set	% of values ≤ 8 °C
$\lvert pT_1 - pT_2 \rvert$	87%
$\lvert pT_1 - pT_3 \rvert$	92%
$\lvert pT_2 - pT_3 \rvert$	87%

Table 4.4: Differences in non-erroneous plant temperature between sets. The percentage of the total number of values that fall within one standard deviation of each other.

From the results in Table 4.4, we see that the difference in values is small between sets, which indicates that this method is feasible and fairly accurate.

The first thing we need to do is to check if the point in question (k, s) is classified as an error, and at the same time we check to see if at least one of the two other sets have a non-erroneous point at that data point k. If there is at least one such point, we replace the missing pT value with the mean of the non-erroneous points. If the value of point k is missing in all sets s, we do nothing.

For notation, see section 4.1.4

Algorithm 3 Plant temperature replacement algorithm

We run this algorithm for all sets at the same time since we want to use values from all sets as replacements.

Inputs: $pT^{k,s}, e_{pt}^{k,s}$
Number of imputations: 1
Outputs: $pT^{k,s}$
$k = \{1, 2, ..., 6954\}, s = \{1, 2, 3\}$

```
  for $s = 1, 2, 3$ do
      for $k = 1, 2, ..., 6954$ do
          if $e_{pT}^{k,s} = 1 \wedge \sum_{m=1}^{3} e_{pT}^{k,s} < 3$ then
              $pT^{k,s} = \frac{1}{3-\sum_{s=1}^{3} e_{pt}^{k,s}} \left[ \sum_{s=1}^{3} (1 - e_{pt}^{k,s}) pT^{k,s} \right];$
              $R^{k,s} = 1;$
          else if $\sum_{s=1}^{3} e_{pT}^{k,s} = 3$ then
              $R^{k,s} = -1;$
          end if
      end for
  end for
```

Chapter 5

Forecasting: Machine learning methods

Now we have preprocessed the data in the hope that this improves the result of our forecasting techniques. In this chapter we will attempt to use Elastic Net Regularization in order to forecast the power production at time $k + 1$, given data at time k.

Choosing to use Elastic Net Regularization in order to predict power production may seem like a strange choice, considering that so many of our variables have a periodic nature (increases in the morning and decreases in the evening). The algorithm seeks to discover a linear relationship between the predictors (inputs) and targets (output), which means it can not describe these periodic elements in itself. We chose to use this algorithm for three main reasons.

Firstly, if the elastic net algorithm manages to forecast adequately one hour into the future, it may signify that there is some linear relationship between the predictors and the outcomes, even if there is no such physical relationship. If this were the case, it may indicate that there are ways of forecasting using linear methods.

Secondly, given that we have preprocessed what we deemed the most influential predictors for power, we wanted to use an algorithm where we could see the impact of these variables clearly.

Finally, there are ways to work around this linearity, which we will look at in more detail later in this Chapter.

Perhaps the most important factor in any machine learning method is the choice of predictors and outcomes. In addition to this, we have to choose the time frame for these predictors and outcomes. We use different combinations of all available data, which initially gives us 12 predictor variables, described in Chapter 2. We will also attempt to include error identification vectors for irradiance, weather temperature, plant temperature and cloud cover to see how this affects the outcome.

Our objective variable is power at time $k+1$, P^{k+1}, and we include power at time k, P^k as a predictor.

$$\begin{aligned} \text{Original power:} \quad P^k &= \{p_1, p_2, ..., p_{6954}\} \\ \text{Target variable: } P^{k+1} &= \{p_2, p_3, ..., p_{6954}\} \end{aligned}$$

We can of course not use data at time $k = 6954$ to predict power at time $k = 1$, so all data at time $k = 6954$, and power at time $k = 1$ must therefore be excluded.

In the following Sections, we shortly describe some key concepts in machine learning and how relevant they are to the problem we are attempting to solve.

5.0.1 Training, validation and test data

Before we start training any method, we need to first identify how we should split the data into training, testing and validation sets. Training data is the data a method uses to adjust its hyperparameters so that they best fit the training targets. Validation data is the data the method tests its accuracy and performance on in order to know if it should train more or not. Finally, the testing data is used to measure the performance of the final model.

We have 1 target variable and 12 predictor variables, with 6953 measurements in each of the three sets (we removed the first measurement of power and the last measurements for all predictors). If we had fewer measurements, we would maybe need to include data from all three sets in our training and testing data, but this does not appear to be necessary. We will use the standard split of training and validation data, which is 75% training and validation and 25% testing data, which we will describe in more detail in Section 5.0.4.

5.0.2 Bias and variance

Bias and variance are often mentioned in relation to each other, as both terms are descriptions of the quality of a model. The bias describes how well a model fits the data, where a large bias means it does not fit the data well. Variance refers to the amount of variation in a model. If a model has high variance, it may fit the data very well, but have very large variations in its values, and may fail to describe any trends in the data. An example would be a straight line fitted to some data, which will have no variance, but high bias since it does not fit the data well. A high degree polynomial (particularly if the degree of the polynomial is the same as the number of data points) fitted to the same data, may have no bias but very high variance. This example is illustrated in Figure 5.1.

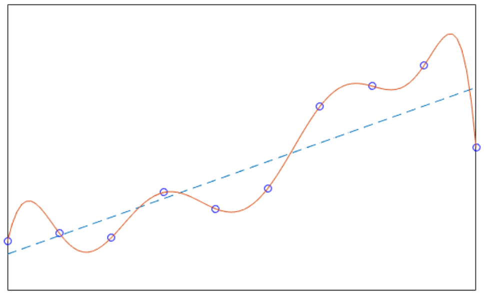

Figure 5.1: Illustration of bias and variance. The dotted line has very high bias and no variance, while the polynomial curve has almost no bias and very high variance.

Increasing the complexity of a model may reduce the bias, but increase variance. This is called the *bias-variance tradeoff.*

5.0.3 Generalization & Overfitting

Overfitting is when a we have a model that fits training and validation data with accuracy greater than on the testing data, and thereby fails to adequately describe any useful trends.

A model that does not overfit is a model that generalizes well. This is a concept that must be taken into account when implementing any form of machine learning techniques, and we must consider any possible issues that may result in overfitting. In our case, one possible issue is that the model may not use data from the entire year in training, and thereby not see the range of possible outcomes that our data has.

5.0.4 Cross-validation

Cross-validation is a technique for assessing how well a statistical model will generalize to a data set, and the goal is to test the models ability to predict from new data that was not used while training it, in order to detect problems such as overfitting. There are various methods of cross-validation used for different purposes, but we will only describe the ones we implement. Both these methods are non-exhaustive cross-validation, meaning they to not compute all ways to split a sample, which may be less accurate but less computationally expensive.

Holdout method is a method based on splitting a data set X randomly into two parts x_0 and x_1, usually training and testing sets, where the size of the sets can be decided by the user. Typically, the model is run several times with different holdout sets, and the results are averages of these multiple runs.

k-fold cross validation is a method of splitting the data into k equally sized disjoint subsets, and using a single subset as the validation data and the rest for training in each iteration. This is done over k iterations so that all data is used as validation exactly once.

NOTE: *The k in k-fold cross validation is not related to the k we have used to denote indices in our data.*

For our purposes, we will do as mentioned in Section 5.0.1, and split the data such that 75% of the data is in the training/validation set, and 25% is within the testing set, i.e. the holdout set. We will use 10-fold cross-validation on our training and validation data, and we will run the method several times with data from multiple imputations on both weather temperature and irradiance.

5.0.5 Shrinkage methods

In general problems where we have an input vector $X^T = (X_1, X_2, ..., X_p)$ and want to predict an output Y, we have a problem of the form

$$\begin{aligned} Y &= \textstyle\sum_{j=1}^{p} X_j\beta_j + \beta_0 + \epsilon \\ &= f(X) + \epsilon \end{aligned}$$

Where β_0, and $\beta = \{\beta_1, ..., \beta_j, ..., \beta_p\}$ are parameters related to the inputs X_j, and ϵ is an approximation error.

If we were to solve this problem by Least Squares regression, i.e. solving $\min_{\beta_0,\beta} \sum_{k=1}^{N} \left(y_k - \beta_0 - \sum_{j=1}^{p} X_{kj}\beta_j\right)^2$, we would retain all values in the input vector X and use them as predictors for our value Y. The impact of these predictors may be limited, but not zero.

However, all of the input data is not necessarily relevant to estimate an output, and in some cases, some of the predictor variables may only reduce the accuracy of the model. In these cases it may be be valid to only retain some predictors and discard others, achieving a better model, possibly with a lower prediction error than the full model. Some methods that utilize this are called *shrinkage methods*. Two of the most common shrinkage methods are called ridge regression and lasso regularization. [4]

Ridge regression imposes a penalty term on the size of the regression coefficients β_0, β by introducing a complexity parameter $\lambda \geq 0$. It then aims to minimize this penalized residual sum of squares, using the L_2 norm of β

$$\arg\min_{\beta_0,\beta} \left[\sum_{k=1}^{N}(y_k - \beta_0 - \sum_{j=1}^{p} X_{kj}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2\right] \tag{5.1}$$

The complexity parameter controls the amount of shrinkage, which means that as λ becomes larger, we have more shrinkage, i.e. the coefficients β_0, β_j shrink towards zero. The optimal solution to the problem is usually the least complex model (smallest λ) such that the model's error is within one standard deviation of the smallest error.[4]

Lasso regularization is a shrinkage method just like ridge regression, but they differ in that Lasso is defined by the L_1 norm

$$\arg\min_{\beta_0,\beta} \left[\frac{1}{2}\sum_{k=1}^{N}(y_k - \beta_0 - \sum_{j=1}^{p} X_{kj}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|\right] \tag{5.2}$$

The L_1 penalty term in Lasso makes the solutions nonlinear, and computing the lasso solution is a quadratic programming problem, although there exists algorithms that computes the solutions with the same computational cost as ridge regression.[4]

5.1 Elastic Net Regularization

Elastic net regularization [ENR] is a form of supervised learning regression, i.e. both data X^k and targets y^k are given to the algorithm for training and validation, and the algorithm's goal is to finely tune the hyperparameters corresponding to the data so that it fits its targets as well as possible. It is a linear combination of the Lasso regularization and ridge regression. ENR solves the problem

$$\arg min_{\beta_0,\beta} \left[\sum_{k=1}^{N} (y_k - \beta_0 - \sum_{j=1}^{p} X_{jk}\beta)^2 + \lambda B_\alpha(\beta) \right] \tag{5.3}$$

where

y_k is the power data at data point P_{k+1}, the target value.

X_{jk} is the value of variable $j = \{1, 2, ..., p\}$ at data point k.

N is the number of observations.

β_0 is a scalar parameter

β is a vector parameter of length p, where p is the number of different variables in X.

λ is a nonnegative regularization parameter.

In which the number of non-zero components of β decrease as λ increases. The problem involves both the L_1 and L_2 norm of β.

B_α is defined by

$$B_\alpha = (1-\alpha) \parallel \beta \parallel_2^2 + \alpha \parallel \beta \parallel_1 = \sum_{j=1}^{p} \left((1-\alpha)\beta_j^2 + \alpha|\beta_j| \right). \tag{5.4}$$

Where $\alpha \in [0, 1]$ is chosen by the user or by cross validation.

For $\alpha = 1$ elastic net is the same as lasso regularization, but for other values of α, our penalty term $B_\alpha(\beta)$ will interpolate between the L_1 norm and the squared L_2 norm of β. When $\alpha = 0$, the elastic net algorithm is equivalent to ridge regression.

Note: In MATLABs lasso/elastic net function, $\alpha = 0$ *is not a feasible value, but for elastic net in general,* $\alpha = 0$ *is a feasible value.*

5.2 Results

Our first attempt at using the elastic net to forecast was done on all data from the Bari data set, where irradiance, plant temperature and weather temperature were preprocessed as described in Chapter 4. We chose to test the method with three values for $\alpha = \{0.25, 0.5, 0.75\}$, using every combination of the 10 different imputations in weather temperature and irradiance, thereby running the algorithm 100 times for each α, for each data set.

The questions we wish to answer in this section are:

- Is there one optimal value for α that results in the most accurate predictions for each or all sets?
- Are there large differences in error within sets with constant value of α?

Question two is particularly important, because if this is the case, it would indicate that we should implement more imputations than 10 in our multiple imputations, as this may improve accuracy even further.

Of course, the most essential question is whether or not the elastic net algorithm can be used to forecast power production from photovoltaic solar panels, which we will look further into in Chapter 6.

We will begin by looking at the results of implementation on preprocessed data, and evaluate the general accuracy on the different sets. We would expect irradiance and cloud cover, together with the time of day, to be the predictors with the largest impact on the power production, so we will also look at the impact of predictors, and see how they compare to our expectations.

Error estimates

The mean squared error for each iteration is calculated by approximating the data in our holdout set H.

$$MSE = \sum_{k \in H} \left[y_k - \beta_0 - \sum_{j=1}^{p} X_{jk} \beta_j \right]^2$$

Where y_k are the target values and X_{jk} our predictor variables from the holdout set H, and $\beta_0, \beta_j, j = \{1, ..., p\}$ are the minimized arguments from the elastic net algorithm. Since we do several runs for all choices of predictors and values of α, the error in the tables in the following sections are the mean of these mean squared errors.

5.2.1 Null model

Our results would be hard to interpret if we do not have anything to compare it to, which is why we include the null model. This model aims to predict power at time $k+1$ using only power at time k. The error here will give us an indication of whether or not our model improves as we include more predictors to the algorithm. We run the null model for the three values of α, of which results are displayed in Table 5.1. The model was run 10 times for each value of α for each set.

$*10^4$		Mean error
Set 1	$\alpha = 0.25$	1.5226
	$\alpha = 0.5$	1.5364
	$\alpha = 0.75$	1.5255
Set 2	$\alpha = 0.25$	1.7819
	$\alpha = 0.5$	1.7836
	$\alpha = 0.75$	1.7840
Set 3	$\alpha = 0.25$	2.2878
	$\alpha = 0.5$	2.2966
	$\alpha = 0.75$	2.2728

Table 5.1: Results of the null model for all sets and all values of α. All subsequent results should be more accurate than the null model.

5.2.2 Implementation on preprocessed data

We will now run the algorithm for our preprocessed data, including the error classification vectors in our prediction. All results are from 100 runs of the algorithm for each value of α for each set.

We will begin by looking at the results from set 1.

$*10^4$	Mean	Max MSE	Min MSE
$\alpha = 0.25$	1.3782	1.5268	1.2184
$\alpha = 0.5$	1.3530	1.5059	1.2090
$\alpha = 0.75$	1.3611	1.4889	1.2357

Table 5.2: Mean value of 100 mean squared errors for different values of α - Set 1 with error classifiers.

We see a clear improvement from the null model. We would also like to see the impact of the predictors, which we will do by looking at the values of the vectors β corresponding to each run of the algorithm. For set 1, the predictors acted similarly for all values of α, so we plot the predictors for all values in Figure 5.2.

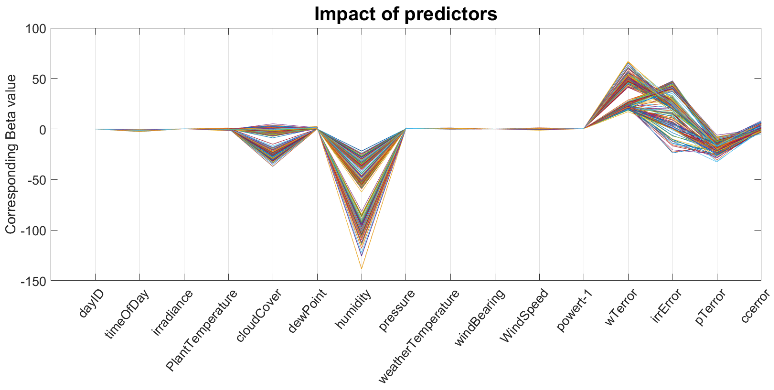

Figure 5.2: The impact of the predictors for set 1 for all values of α. Each line represents one of the 300 iterations upon the set.

It is clear from Figure 5.2 that the algorithm considers humidity to be the most influential variable in the set with regards to power production. The β value corresponding to humidity is negative, which indicates that when humidity at time k is high, the power at time $k+1$ will be lower. We also see an impact from the cloud cover, which affects the power production negatively in some iterations and positively in others.

An interesting observation is the impact of the error classification vectors. It seems that error in weather temperature at time k indicates higher power production at time $k+1$. Error in irradiance at time k has both negative and positive impact on power at time $k+1$, with the same values of α. Error in plant temperature has a negative impact, while error in cloud cover has next to no impact on the power. This has no physical explanation, of course, but could be an indication of a correlation as a result of our choice of error indication.

It seems like the values of the predictors appear in two distinct clusters. When we inspect this further, it seems that these clusters are results of the different imputations in weather temperature, not in irradiance. For some imputations in weather temperature, the β value corresponding to cloud cover increases negatively, which leads to the (absolute) β value corresponding to humidity decreasing. The variations in the impact of the error classification vectors do not seem to be the result of neither the impact of cloud cover and humidity, nor the choice of weather temperature or irradiance data. We note which weather temperature imputations that led to a greater impact of cloud cover, so that we may compare them to results from other sets.

We now look at the algorithms performance on set 2.

$*10^4$	Mean	Max MSE	Min MSE
$\alpha = 0.25$	1.6957	1.8607	1.5456
$\alpha = 0.5$	1.6981	1.8655	1.5501
$\alpha = 0.75$	1.6930	1.8917	1.5268

Table 5.3: Mean value of 100 mean squared errors for different values of α - Set 2 with error classifiers.

Yet again we see an improvement from the null model. The predictors in set 2 again behave similarly for all values of α, so we plot the coefficients for all iterations.

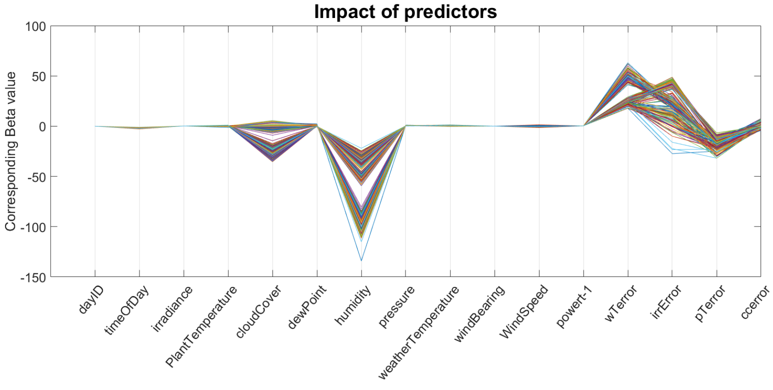

Figure 5.3: The impact of the predictors for set 2 for all values of α. Each line represents one of the 300 iterations upon the set.

Again the same trend appears. The model deems humidity highly influential, while the β-value corresponding to cloud cover changes from negative to positive between iterations. The clusters in this plot are the result of the same factors as for the set 1, different values for weather temperature which result in an occasional bigger negative impact of cloud cover, which leads to a lower reliance on humidity. The errors in weather temperature, irradiance and plant temperature exhibit the same features as for set 1, and yet again seem to vary arbitrarily.

We note that the imputations in weather temperature that led to an increased impact of cloud cover in set 2, were not the same as in set 1.

Lastly we look at set 3.

$*10^4$	Mean	Max MSE	Min MSE
$\alpha = 0.25$	2.1662	2.3374	1.9965
$\alpha = 0.5$	2.1647	2.3831	1.9672
$\alpha = 0.75$	2.1640	2.3406	1.9565

Table 5.4: Mean value of 100 mean squared errors for different values of α - Set 3 with error classifiers.

Again we see an improved accuracy compared to the null model. The predictors behave similarly to set 1 and 2, and has almost no variation between values of α, and we therefore again plot for all values of α.

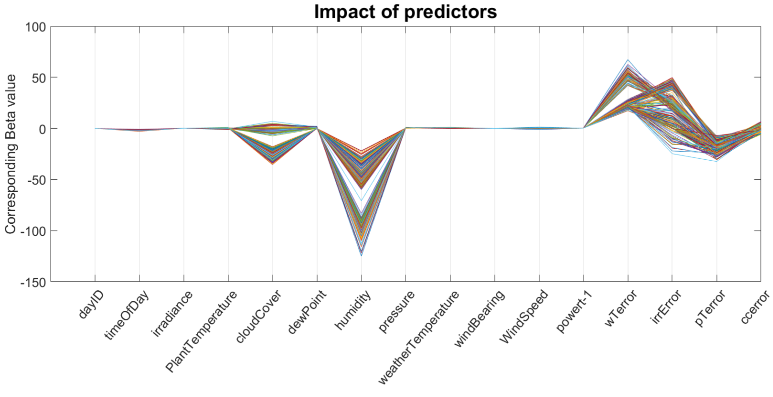

Figure 5.4: The impact of the predictors for set 3 for all values of α. Each line represents one of the 300 iterations upon the set.

We see the same trend appear, with humidity generally having a very negative impact on the power, while cloud cover varies from having negative to positive impact between iterations, seemingly distributed into two clusters. These clusters have the same features as in set 1 and 2, with higher impact of cloud cover leading to lower impact of humidity. The errors from weather temperature, irradiance and plant temperature behave similarly as for set 1 and 2, and vary arbitrarily between iterations.

The imputations in weather temperature that lead to a higher impact of cloud cover in set 3 were, again, not the same as neither set 1 nor set 2.

The large impact of the error classification vectors leads us to believe that we should exclude these in future predictions. Since the impact varies arbitrarily between iterations, it is likely that these data only 'confuses' the algorithm by introducing random correlations in the training and validation data that leads to weak performances on the testing data. It may also be the reason the clusters appear in the predictors. We therefore run the algorithm similarly as before, only excluding the errors.

This time we display all values for all sets in Table 5.5 to compress the results slightly.

$*10^4$		Mean	Max MSE	Min MSE
Set1	$\alpha = 0.25$	1.4139	1.5804	1.2573
	$\alpha = 0.5$	1.4132	1.5728	1.2447
	$\alpha = 0.75$	1.4109	1.5518	1.2483
Set2	$\alpha = 0.25$	1.7253	1.8834	1.5974
	$\alpha = 0.5$	1.7187	1.8515	1.5084
	$\alpha = 0.75$	1.7208	1.9580	1.5873
Set3	$\alpha = 0.25$	2.1749	2.4211	1.9729
	$\alpha = 0.5$	2.1620	2.3686	1.9838
	$\alpha = 0.75$	2.1791	2.3507	2.0209

Table 5.5: Mean value of 100 mean squared errors for different values of α - All sets, no error classifiers

The mean error increases for most sets, and values of α, but by a relatively small amount, and is still an improvement with respect to the null model.

As for the predictors, they behave similarly for all values of α, and so we plot predictor coefficients for all values of α in one Figure for each set.

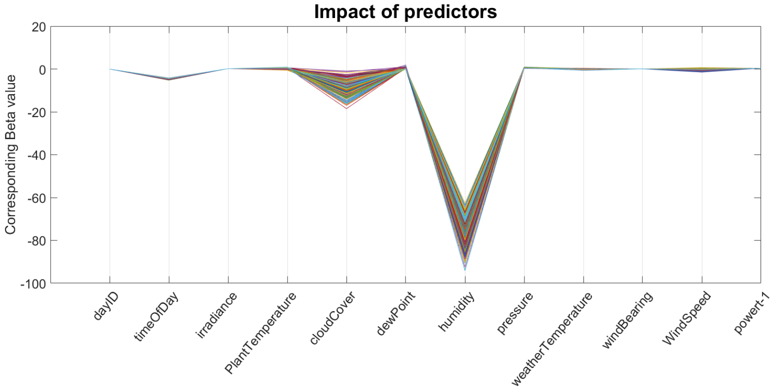

Figure 5.5: The impact of the predictors when we exclude error classification vectors, for all values of α in set 1. Each line represents one of the 300 iterations upon the set.

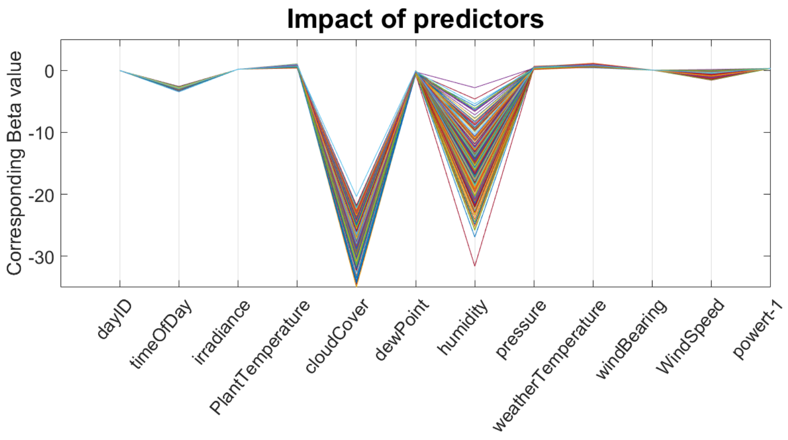

Figure 5.6: The impact of the predictors when we exclude error classification vectors, for all values of α in set 2. Each line represents one of the 300 iterations upon the set.

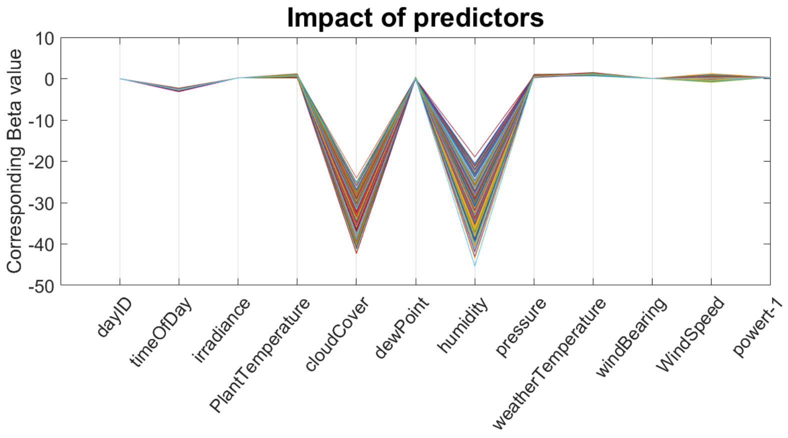

Figure 5.7: The impact of the predictors when we exclude error classification vectors, for all values of α in set 3. Each line represents one of the 300 iterations upon the set.

The most notable difference from the earlier predictions is that the clusters seem to have disappeared. Since the only change we have done is to remove the error classification vectors, this indicates that this was a good choice, since they seemed to only distort the results. We therefore choose to exclude error classification vectors from all predictions after this.

Another thing to note is that the models seems to rely slightly more on some predictors, 'timeOfDay' in particular. We should expect that time of day had a relatively high impact on the prediction, given that the highest power production is generally in the middle of the day. The reason behind this lack of impact is most likely the linear nature of the elastic net algorithm. The relationships between the predictors and the outcomes is linear, which means that a variable that ranges between 2 and 20 each day, with an expected peak from 12 to around 18, is not very helpful for the model itself.

We therefore introduce a transformed variable that interacts upon 'timeOfDay' (denoted here as t for simplicity) in order to more realistically simulate the daily variations in time. We define this term as $\tau(t) = \cos(2 * \pi * (t - 3)/24)$, where we choose to use $t - 3$ instead of just t in order for the peak (in this case negative) to be at 15:00 for each day. Whether the value is negative or positive does not matter, as long as the peaks agree with the peaks in power production.

We run the model again with the same data as the previous run, that is, excluding error classification vectors. We run it once for each combination of data from multiple imputations, which means 100 runs for each set, for every value of α.

$*10^4$		Mean	Max MSE	Min MSE
Set1	$\alpha = 0.25$	1.3990	1.5358	1.2637
	$\alpha = 0.5$	1.3847	1.5171	1.2351
	$\alpha = 0.75$	1.3807	1.5147	1.1386
Set2	$\alpha = 0.25$	1.6881	1.8479	1.5436
	$\alpha = 0.5$	1.6834	1.9120	1.5013
	$\alpha = 0.75$	1.6871	1.8754	1.5384
Set3	$\alpha = 0.25$	2.1276	2.3157	1.8968
	$\alpha = 0.5$	2.1315	2.3724	1.9379
	$\alpha = 0.75$	2.1333	2.3811	1.9934

Table 5.6: Mean value of 100 mean squared errors for different values of α, including our transformed variable τ - All sets

If we compare the results in Table 5.6 with our earlier results, we can see that the mean error for this version is lower for set 2 and 3 for all values of α, but higher for set 1 compared to the results in Table 5.2 (although it is an improvement from the results in Table 5.5). We will look more at the differences between results later, but first we want to see how the predictors changed from what we saw earlier.

Again, the values of α did not affect the predictors, so we plot for all values of α in Figures 5.8,5.9,5.10.

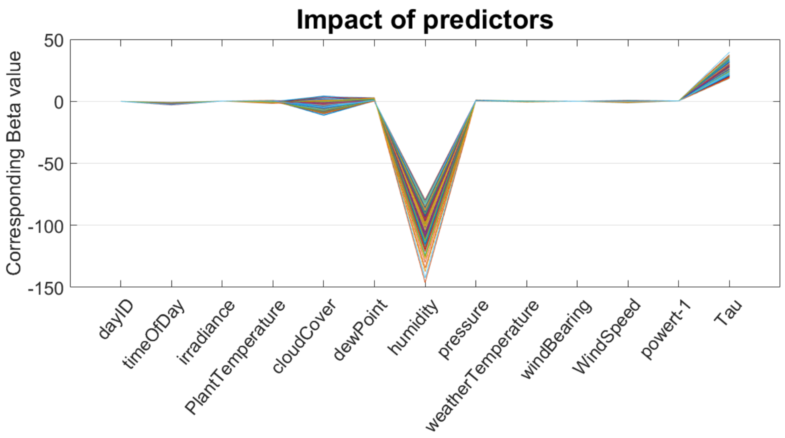

Figure 5.8: Predictors for set 1, no errors, with transformed variable τ

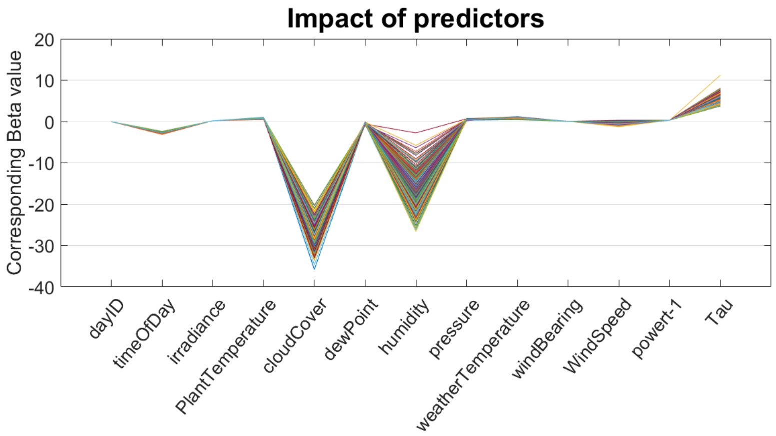

Figure 5.9: Predictors for set 2, no errors, with transformed variable τ

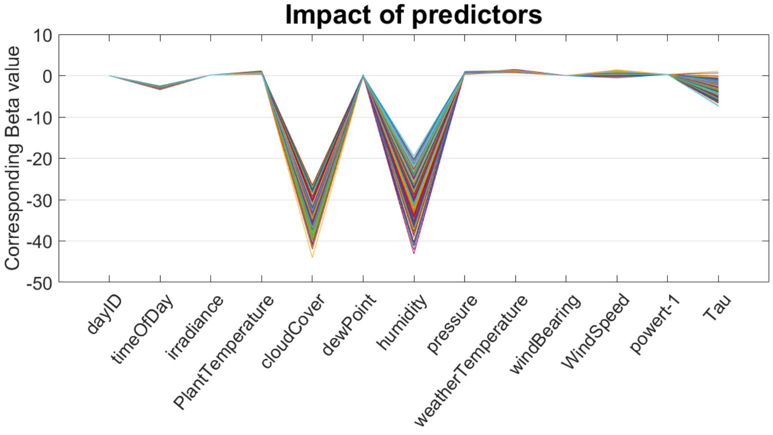

Figure 5.10: Predictors for set 3, excluding errors, with transformed variable τ

The clusters from Figures 5.2 - 5.4 have not reappeared, although the values of β corresponding to cloud cover and humidity do vary between iterations. This is a strong indication that these clusters were a result of the error classification vectors, which substantiates our choice of omitting these as predictors.

An interesting thing to note here is that only when applied to set 3 did τ have a negative corresponding β-value. In both set 1 and 2, the term is positive, which is not what we expected.

In general, including this transformed variable seems to improve the predictions somewhat. It seems variables that clearly separate the afternoon from the morning will improve the accuracy of the model. We therefore choose to construct a bigger data set by using interaction terms, i.e. products of variables in the set, as well as utilizing a new variable z, which we can use to separate the morning (02:00 to 11:00) from the afternoon (12:00 to 20:00), and have z interact upon the other predictors through multiplication. z is defined as,

$$z = \begin{cases} 1, & \text{for afternoon, i.e. corresponding time of day } \geq 12 \\ 0, & \text{otherwise.} \end{cases}$$

Again, we have our 12 predictors, which we will denote here simply as $x_1, x_2, ..., x_{12}$.

We construct our new data set by multiplying every combination of predictors, as well as multiplying our new variable z with every predictor, such that our new data set consist of our preprocessed data set,

$$[x_1, x_2, ..., x_{12}], \tag{5.5}$$

a set consisting of the products of all variables in (5.5) (interaction terms),

$$[x_1x_2, x_1x_3, ..., x_{11}x_{12}], \tag{5.6}$$

as well as the products of every variable in (5.5) and z (interaction terms),

$$[zx_1, zx_2, ..., zx_{12}]. \tag{5.7}$$

Combining (5.5), (5.6) and (5.7), we get our new data set X,

$$X = [x_1, x_2, ..., x_{12}, x_1x_2, x_1x_3, ..., x_{11}x_{12}, zx_1, ..., zx_{12}]$$

We then use this new data set as predictors for our algorithm.

These results, understandably, take a lot more time to attain than the previously described ones. The number of predictors has increased from 12 to 90, which increases the running time of the algorithm by a substantial amount. As a result of this, there was not time to run this set for every combination of weather temperature and irradiance data.

Therefore, we first ran the algorithm using only one imputation in irradiance, but using all ten imputations from weather temperature. We then found the imputation in weather temperature that returned the least mean squared error, and used this imputation in weather temperature for all ten imputations of irradiance. The mean error in Table 5.7 are the mean of the errors of these 20 runs of the algorithm.

We also only ran the algorithm for only one value of alpha for each set. The choice of α was based on which α returned the least mean squared error for the set in Table 5.6.

	Mean MSE	Max MSE	Min MSE
Set 1, $\alpha = 0.75$	1.0624	1.1148	0.9700
Set 2, $\alpha = 0.5$	1.5041	1.6761	1.3449
Set 3, $\alpha = 0.25$	1.9988	2.1026	1.8495

Table 5.7: MSE of all sets for our new data set with interaction terms.

The data in Table 5.7 shows a substantial improvement of all previous results for all sets. The running time of the algorithm increased by a lot for each iteration, but given the significant improvement in accuracy, this is by far the best model we have implemented. The mean error might increase if the model was run for all combinations of weather temperature and irradiance data, but most likely by a small amount.

As for the predictors, plots are hard to interpret due to the number of predictors. Additionally, the impact of the predictors were very similar between sets, so we choose to plot all predictors in one figure, and describe the predictors with the largest impact.

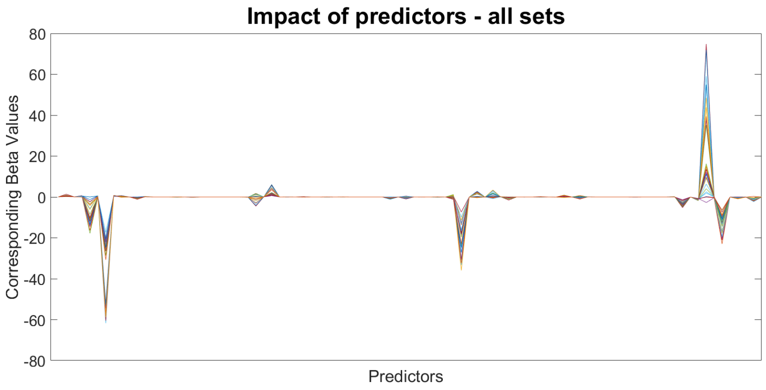

Figure 5.11: Predictors of preprocessed data, with interaction terms.

We again see both cloud cover and humidity have a big impact on the outcome, similar to our predictors in Figures 5.8 - 5.10. There are slight variations from one iteration to the next within sets, but almost no differences between sets. As for the interaction terms, around ten had slight impacts on all sets, with corresponding $|\beta| \leq 2$. We can see that three interaction terms clearly have larger impacts in all iterations, which were

- cloud cover × humidity *(large negative impact)*
- $z\times$ cloud cover *(large positive impact)*
- $z\times$ humidity *(relatively large negative impact)*

It is still clear that humidity and cloud cover are the most influential predictors for the power production. All models also relied slighly on 'timeOfDay', and the product of 'timeOfDay' and z. The prediction of power relies almost solely on these predictors, and the interaction between them and our variable z, which separates mornings from afternoons.

5.2.3 Implementation on original data

Before we proceed to finish analysing the results and aim to make any conclusions, we wish to see how the model performs on the original data.

Our hope is that the preprocessing of data has improved the accuracy, but we know that the preprocessing has given us model errors, which may have distorted the data more than the preprocessing has improved it. Additionally, we saw in the previous section that the predictors we preprocessed in Chapter 4 did not have the impact we originally thought, which indicates that the preprocessing of the data will not have a large impact on the accuracy of the model. We will again present the results relatively compactly.

In our original data we do not have the same number of data sets as the preprocessed data (as a result of the multiple imputations), but we run the algorithm ten times for each set, for each value of α, to utilize different holdout sets, and get a range of mean squared errors.

When we use the original data, we include both power at time k and, since it improved performance on the preprocessed data, the transformed variable τ. This means we have 13 predictors for these runs. The algorithm is run 10 times for each α and each set.

$*10^4$		Mean	Max MSE	Min MSE
Set1	$\alpha = 0.25$	1.3488	1.4768	1.2466
	$\alpha = 0.5$	1.3782	1.5124	1.2800
	$\alpha = 0.75$	1.3356	1.4076	1.1954
Set2	$\alpha = 0.25$	1.7582	1.8865	1.6425
	$\alpha = 0.5$	1.7269	1.7725	1.6246
	$\alpha = 0.75$	1.7253	1.8025	1.6197
Set3	$\alpha = 0.25$	2.1510	2.3217	1.9839
	$\alpha = 0.5$	2.1911	2.2848	2.0555
	$\alpha = 0.75$	2.1677	2.2641	2.0430

Table 5.8: Mean value of 10 mean squared errors for different values of α - All sets, original data with transformed variable τ

We can see from Table 5.8 that the algorithm performs relatively well on the original data, with the transformed variable τ included, compared to the results of implementation on the preprocessed data without error classifiers, with transformed variable τ. The impact of different predictors can be seen in Figures 5.12, 5.13, 5.14. Again, the differences between values of α are almost non-existent, so we plot all in one figure for each set.

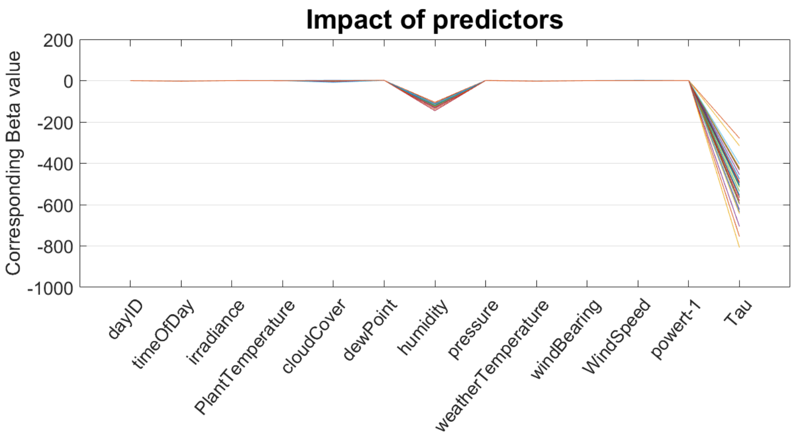

Figure 5.12: Predictors of original data with transformed variable τ on set 1. Ten iterations for each value of α.

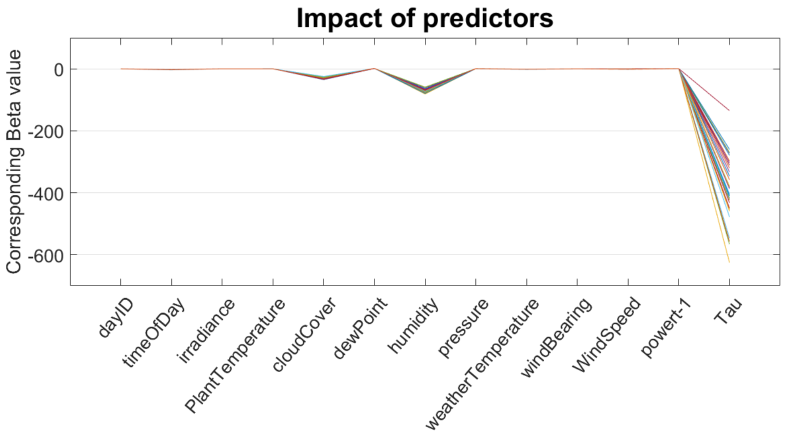

Figure 5.13: Predictors of original data with transformed variable τ on set 2. Ten iterations for each value of α.

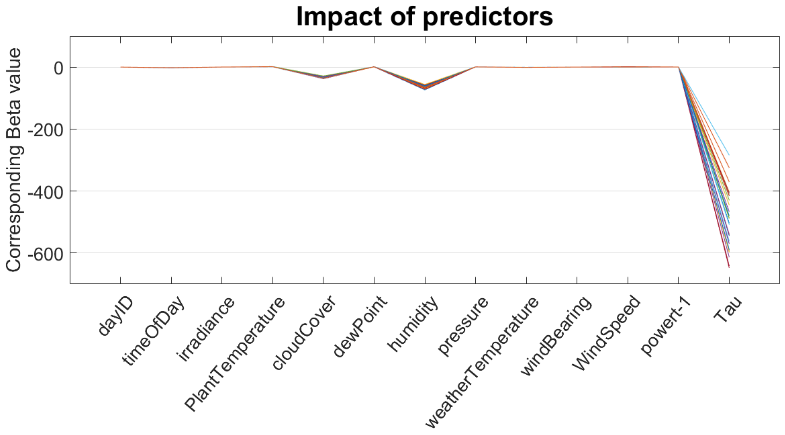

Figure 5.14: Predictors of original data with transformed variable τ on set 3. Ten iterations for each value of α.

In all three sets, the humidity still had a very high impact on the outcome, and cloud cover retained a relatively big impact, but the transformed variable τ had by far the largest (albeit negative) corresponding β-value for all sets.

Chapter 6

Comparison and analysis

In this chapter we will look further into our results and see if we can find any patterns or trends.

We will look at the results of all implementations of elastic net regularization on the different data sets. We have run the elastic net algorithm for the preprocessed, and the original data. The preprocessed data has been implemented with and without error classification vectors, and the transformed variable τ, and as a part of a larger data set with interaction terms. The original data has been implemented without the error classification vectors, with the transformed variable τ. We have also implemented the null model.

The (mean of) mean squared errors of all the different choices of input in the ENR are displayed in Table 6.1. **NE** denotes that errors are excluded from the model, and $\mathbf{w.}\tau$ notes that the transformed variable $\tau = \cos(2\pi(timeOfDay - 3)/24)$ has been included as a predictor. **Prep** refers to the preprocessed data set, while **OG** refers to the original data. **INTER** denotes that we use the expanded data set with interaction terms. **NULL** denotes the null model.

$*10^4$	$\alpha = 0.25$ MSE	$\alpha = 0.5$ MSE	$\alpha = 0.75$ MSE
Set 1 Prep	1.3782	1.3530	1.3611
Set 1 Prep (NE)	1.4139	1.4132	1.4109
Set 1 Prep (NE,w.τ)	1.3990	1.3847	1.3807
Set 1 INTER (NE)			**1.0624**
Set 1 OG (NE,w.τ)	1.3488	1.3782	1.3356
Set 1 NULL	1.5226	1.5364	1.5255
Set 2 Prep	1.6957	1.6981	1.6930
Set 2 Prep (NE)	1.7253	1.7187	1.7208
Set 2 Prep (NE,w.τ)	1.6881	1.6834	1.6871
Set 2 INTER (NE)		**1.5041**	
Set 2 OG (NE,w.τ)	1.7582	1.7269	1.7253
Set 2 NULL	1.7819	1.7836	1.7840
Set 3 Prep	2.1662	2.1647	2.1640
Set 3 Prep (NE)	2.1749	2.1620	2.1791
Set 3 Prep (NE,w.τ)	2.1276	2.1315	2.1333
Set 3 INTER (NE)	**1.9988**		
Set 3 OG (NE,w.τ)	2.1510	2.1911	2.1677
Set 3 NULL	2.2878	2.2966	2.2728

Table 6.1: Mean squared error of all variations of predictors with different α-values. All values are scaled down by 10^4. Least values in bold.

For all sets, the least error value was for the data set with interaction terms, which had errors that were significantly lower than for all other choices of data. There is a possibility of the mean error decreasing for other values of α, but this difference would most likely be marginal.

From the results, it does not seem like there is a clear, optimal choice for α. The values varied slightly, but not by significant amounts. The choice of data was a lot more influential for the results than the choice of α.

We see that the preprocessed data improved the accuracy for sets 2 and 3, but that the algorithm performed better on the original data for set 1. However, the differences between these results are small, most likely due to the fact that the variables we preprocessed did not have a large impact on the outcomes.

Including the error classifiers did not improve the model.

In general, forecasting solely based on regression is difficult. Particularly early mornings and late evenings, when we have a large rate of change in the power production, are difficult for the algorithm to pick up on. As illustrated in Figure 6.1, the algorithm predicts low - but not zero - production. This is one of the areas where the algorithm struggles the most.

In addition to this, the time gap between 20:00 and 02:00 makes it difficult for the algorithm to predict the next time step 6 hours into the future. This may have made our problem particularly difficult.

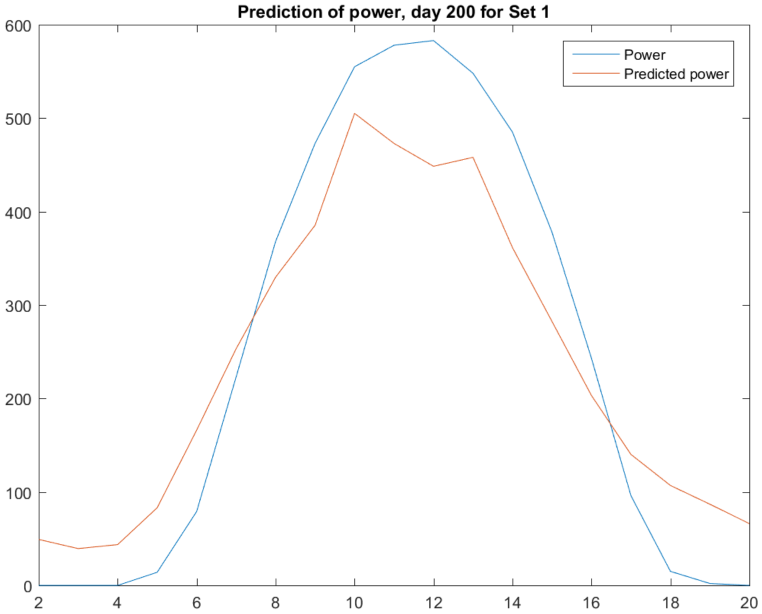

Figure 6.1: One day of forecasted power, July 19th, for set 1. Preprocessed data with errors as predictors. We chose to show this day in particular because it illustrates the problem of forecasting early mornings and late evenings.

It is clear from the results that if we were to forecast using Elastic Net Regularization (or any other linear model) using data similar to the data from Bari, it is crucial to include interaction terms. This action improves accuracy substantially, and is essential for giving a linear model a way to explain daily variations, even though it does increase the computation costs by a substantial amount.

Chapter 7

Conclusions

We have analysed the data from three plants in Bari, Italy, and preprocessed the data the way we deemed most reasonable. We have then applied the elastic net algorithm to this data, in order to predict power production one hour into the future.

Our results indicate that the elastic net algorithm is capable of forecasting power production in a one hour time frame, providing that we include interaction terms as our predictors.

We have not noticed any extreme weather conditions in the data, which makes it hard to say if the algorithm would handle that well or not.

In general, forecasting in the early morning and late evening is where the algorithm struggles the most.

Further improvements

There are several things that we could have improved further. Firstly, we could have applied our data to different machine learning methods and see if results improved, e.g. random forests or support vector machines. It would also be interesting to see the data implemented with even more interaction terms. Finally, implementing the algorithm on data with smaller time steps (half hour or fifteen minute measurements), may have improved the accuracy.

Bibliography

[1] J. Antonanzas et al, *Review of photovoltaic power forecasting*, Elsevier, 2016.

[2] Michelangelo Ceci et al, *Predictive Modeling of PV Energy Production: How to Set Up the Learning Task for a Better Prediction?*, IEEE, 2017.

[3] John A. Duffie & William A. Beckman, *Solar Engineering and Thermal Processes*, Fourth Edition, John Wiley & Sons, 2013

[4] Trevor Hastie, Robert Tibishirani & Jerome Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Second Edition, Springer 2009