Degree Project
Level: Master’s

Household’s energy consumption and production forecasting: A Multi-step ahead forecast strategies comparison.

Author: Gonzalo Martín-Roldán Villanueva.
Supervisor: Mark Dougherty, Jonathan Atkinson.
Examiner: Siril Yella
Subject/main field of study: Microdata Analysis
Course code: MI4001
Credits: 30 ECTS
Date of examination:

At Dalarna University it is possible to publish the student thesis in full text in DiVA. The publishing is open access, which means the work will be freely accessible to read and download on the internet. This will significantly increase the dissemination and visibility of the student thesis.
Open access is becoming the standard route for spreading scientific and academic information on the internet. Dalarna University recommends that both researchers as well as students publish their work open access.
I give my/we give our consent for full text publishing (freely accessible on the internet, open access):
Yes ☑
No ☐
Abstract:

In a changing global energy market where the decarbonization of the economy and the demand growth are pushing to look for new models away from the existing centralized non-renewable based grid. To do so, households have to take a ‘prosumer’ role; to help them take optimal actions is needed a multi-step ahead forecast of their expected energy production and consumption. In multi-step ahead forecasting there are different strategies to perform the forecast. The single-output: Recursive, Direct, DirRec, and the multi-output: MIMO and DIRMO. This thesis performs a comparison between the performance of the different strategies in a ‘prosumer’ household; using Artificial Neural Networks, Random Forest and K-Nearest Neighbours Regression to forecast both solar energy production and grid input. The results of this thesis indicates that the methodology proposed performs better than state of the art models in a more detailed household energy consumption dataset. They also indicate that the strategy and model of choice is problem dependent and a strategy selection step should be added to the forecasting methodology. Additionally, the performance of the Recursive strategy is always far from the best while the DIRMO strategy performs similarly. This makes the latter a suitable option for exploratory analysis.

Keywords: Multi-step, forecast, strategies, Recursive, Direct, DirRec, DIRMO, MIMO, Artificial Neural Networks, Random Forest, K-Nearest Neighbours Regression, MAPE, MAE
# Table of Contents:

## Chapter 1: Introduction
- 1.1 Background 
- 1.2 Purpose 
- 1.3 Literature review
  - 1.3.1 Deseasonalizing 
  - 1.3.2 Multi-step ahead forecasting strategies
    - 1.3.2.1 Recursive strategy 
    - 1.3.2.2 Direct strategy 
    - 1.3.2.3 DirRec strategy 
    - 1.3.2.4 MIMO strategy 
    - 1.3.2.5 DIRMO strategy 
  - 1.3.3 Models
    - 1.3.3.1 Artificial Neural Network 
    - 1.3.3.2 Random Forest 
    - 1.3.3.3 K-Nearest Neighbours Regression 
  - 1.3.4 Measures of performance 

## Chapter 2: Methodology
- 2.1 Data collection
  - 2.1.1 Energy feeds 
  - 2.1.2 Weather data 
- 2.2 Cleaning data
  - 2.2.1 Energy feeds 
  - 2.2.2 Weather data 
- 2.3 Aggregation 
- 2.4 Deseasonalization 
- 2.5 Feature creation 
- 2.6 Tuning parameters 

## Chapter 3: Results

## Chapter 4: Discussion

## Chapter 5: Conclusion

## References:
Chapter 1: Introduction

In the current global energy market things are changing; the increment of the climate change consciousness in an increasing global population, the decarbonization of the energy consumption, with the clear example of the electric vehicles, and the energy production with important investments in renewable energies, are some of the big challenges facing the existing non-renewable electrical grids. Along with the planning, renovation and optimization of the infrastructure, other actions must be taken to cope with the simultaneous decarbonization of the economy and the increasing pressure on the electric infrastructure. The decentralization of the energy production, where multiple small agents produce and consume their own energy, and the planning of the consumption based on forecasted availability is contemplated as a possible solution.

1.1 Background

The future of the energy system is a scenario where energy is produced from renewable sources and small or medium entities can consume, generate, store or trade their own energy. This scenario has institutional support (European Commission, 2016) as well as an available technology: smart grids (Nobel grid, 2017), smart meters and community sized renewable energy production technologies. To help the participants in the energy market to take optimal actions is required a one-day ahead energy system multi-step ahead forecast. Multi-step ahead forecast refers to the prediction from a given moment \( t \), to an horizon \( H \) into the future, composed by equal periods of time. One-day ahead means that this horizon \( H \) equals to the count that sums up to one day given the applications time granularity. This forecast opens the opportunity of implementing multiple applications (Clastres et al. 2010) like trading load differences, malfunction detection, optimizing battery charging and demand shifting schemes. In order to be aligned with the current standards of the energy industry the time granularity in this thesis is set to 30 minutes, that makes the horizon \( H \) equal to 48.
Multi-step ahead forecasting is known in the academic literature for its relevance in multiple fields such as economy (Marcellino et al. 2006), finance (Kodogiannis et al. 2002) or energy (Andalib & Atry, 2009). To extrapolate the future values of a given target variable different strategies are proposed. The oldest and most known is the recursive or iterated strategy, where a single model is used by repeatedly feeding it with its own forecasts, resulting in the accumulation of errors as the distance from the initial moment \( t \) increases. The direct strategy (Cox, 1961) trains one single model for each one of the steps until the horizon is reached, this avoids accumulation of errors but imposes time independence to the forecast. A combination of the previous two strategies is the DirRec strategy (Sorjamaa & Lendasse, 2006), where a model is trained for each one of the steps but adding to the time series the forecast of the previous steps. To keep the inherent structure of the data, the Multiple-input Multiple-output (MIMO) strategy, where a single model is used to forecast the desired time series, is proposed (Bontempi, 2008). Finally the Direct Multi-output (DirMO) strategy (Taieb et al, 2010) attempts to give flexibility to the MIMO strategy while preserving the time structure of the data.

1.2 Purpose

This thesis aims to explore the problem of one day ahead energy production and consumption forecasting on single households, so they can participate in the energy market and contribute to the current trend of energy production decentralization. Single households describes one house or flat that has at least one measuring device, called smart meter, that records the energy consumed in the whole property. If the household has some kind of energy production means (e.g solar panels), there are more than one device. These devices measure and keep record of the energy produced by such production means and the flow of energy between the household and the general energy grid.

The main purpose of this thesis is splitted in two objectives. First, this paper makes an evaluation of the performance on the given problem of the different multi-step ahead forecasting strategies with the same three models for each one of the
strategies. Second, this thesis evaluates the produced models’ capacity to develop possible practical applications.

To be able to make a proper comparison this thesis focuses on models that can support both single and multiple outputs in their implementation in the Python library scikit-learn. Therefore, single output models like autoregressive based methods or support vector regression are out of the scope of this thesis. This thesis is part of an open source project, therefore it needs to be as general as possible. This is achieved only by using an overall measurement of the house’s energy consumption rather than its appliances. Also, given the spatial dispersion of the obtained data and the future users only one household is studied at the time. Area based and small communities (a few dozens of households) are out of the scope of this thesis. Due to the size of the dataset and the computational time it takes to train the combination of 986 different models per feed, one energy and one grid feeds from the same household have been randomly selected. The study of the DirRec strategy, due to its computational cost gets out of the scope of this thesis.

1.3 Literature review

In literature there are a wide range of examples of Machine Learning applications for energy forecasting, generally focusing on load forecasting for energy companies. Hu et al. (2013) propose a FA-MA algorithm to find the parameters of a Support Vector Regression for one step ahead load forecasting of the Pennsylvania-New Jersey-Maryland power system. Mandal et al. (2006) proposes an Artificial Neural Network (NN) to produce an on one-to-six hours forecast the load of the Okinawa Electric Power Company with a direct strategy. Sorjamaa and Lendasse (2006) use the DirRec strategy to forecast the national load in Poland.

At a small and microgrid scale, Becali et al. (2004) propose a Self Organising Map Neural Network to produce a one day hourly forecast in a residential area of Portugal with a multi-input-multi-output strategy. Veit et al, (2014). did a benchmarking of the state-of-the-art forecast methods, 3 ARIMA based and one NN, of seven households with multiple energy measure sensors inputs. They
forecast to different horizons in a recursive forecast strategy.

1.3.1 Deseasonalizing

This section is to give theoretical background to the process performed in section 2.4.

Many forecasting studies and applications take a deseasonalization step in their methodologies because as Taieb et al. (2012) states, “Deseasonalization leads to consistently better results”. This is because of the seasonal component of the data affects the forecasting of time series; humans and nature don't behave in the same way at different points in time but there are specific behaviors that occur at certain moments of the day, like waking up at 7:00 am or going to work at 8:00 am. In addition, people do not behave in the same way in January as in August. Seasonality is the effect of those routines that govern our lives on a daily, weekly, monthly, annually basis or for any reasonable period given the nature of the time series.

The transformation explained here and used in this thesis is the multiplicative deseasonalizing technique where any observation can be decomposed in the following way:

\[ y_t = S_t \times T_t \times E_t \]

Where \( y_t \) is the observation in the moment \( t \), \( S_t \) is the seasonal effect of a given period of time, \( T_t \) is the trend of the series and \( E_t \) is the remaining error.

The deseasonalization of this paper is similar to the one found in Andrawis et al. (2011) but using the mean of the data instead of the median as a correcting parameter. The process used to correct the time series seasonality given a period of time \( p \) of the period \( P \), (e.g. half hour of the day, day of the week…), is the next one:

A. For every data point the moving average of \( P \) is computed. This variable is called \( \text{Avg } p \).
B. The series is then normalized as follows:

\[ z_t(i) = \frac{y_t(i)}{Avg_p} \]

By dividing the observations by the moving average of the period \( P \), any effect of that particular period \( P \) on the observation is corrected.

C. Then the seasonal average is computed for all the points that are the same period \( p \) in all different periods \( P \), (e.g. the same half hour for all days):

\[ s_p(i) = Mean(z_t(i)) \]

being \( s_p(i) \) the effect over the observation \( y_t \) the period of time \( p \) that contains it.

D. The deseasonalized series is obtained by computing:

\[ u_t^H(i) = \frac{y_t(i)}{s_p(i)} \]

Where \( u_t^H(i) \) represents the series corrected from its \( p \) time slot effect.

By following these steps it is corrected only the seasonal effect of one period of time \( p \), this means that in order to correct the seasonal effect of multiple periodicities, half hour of the day, day of the week … we need to repeat this process as many times as different periodicities we have.

1.3.2 Multi-step ahead forecasting strategies

Multi step ahead forecasting is the task of predicting up to the \( H^{th} \) value in the future \([y_{N+1}, \ldots, y_{N+H}]\) from a time series \([y_1, \ldots, y_N]\) with \( N \) observations, where \( H > 1 \) (Ben Taieb et al. 2012).

Nowadays there are five common approaches to long-term forecasting, which can be divided into two categories, single output strategies and Multiple output strategies.

**Single output strategies:**

These strategies, to estimate future values to a horizon \( H \), rely on models which forecast one value at the time and the iteration of the learning process or the
forecasting task. They include the Recursive, the Direct and the DirRec strategies. Figure 1 shows a representation of the single output strategies which will be explained in sections 1.3.2.1 to 1.3.2.3.

**Figure 1: Single-output multi-step ahead forecast strategies.**

*Multi-output strategies:*

These strategies attempt to preserve the complex structures of multiple contiguous values that underline the time series data, which cannot be addressed by single output strategies. These strategies produce a forecasted time series instead of a single value. In this category are the Multi-Input Multi-Output and Direct Multi-Output strategies. A Diagram of these strategies can be found in Figure 2 which is explained in sections 1.3.2.4 and 1.3.2.5.

1.3.2.1 Recursive strategy

This is the oldest approach of all strategies; it is also widespread in literature. This strategy consists in the training of one-step ahead forecasting model $g$:

$$y_{t+1} = g(y_t, ..., y_{t-d+1}) + w,$$

where $t \in \{d, ..., N-1\}$, $w$ is a white noise observation, and $d$ is the distance in the past we use as input in the model.
With this model an H-step-ahead prediction problem is tackled by iterating $H$ times a one-step-ahead predictor, where the forecast produced in the $i_{th}$ iteration is added to the time series and used to predict the value $y_{t+i}$,

$$i \in \{1, ..., H\}.$$

Therefore, to estimate the desired horizon the next $\hat{g}(\cdot)$ forecasting tasks are performed:

$$\hat{y}_{t+1} = \hat{g}(y_t, ..., y_{t-d+1})$$
$$\hat{y}_{t+2} = \hat{g}(\hat{y}_{t+1}, y_t, ..., y_{t-d+2})$$
$$\vdots$$
$$\hat{y}_{t+H} = \hat{g}(\hat{y}_{t+H-1}, \hat{y}_{t+H-2}, ..., \hat{y}_{t+1}, y_t, ..., y_{t-d+H})$$

Iterated methods may suffer from low performance due to the accumulation of errors in the forecasting process. This happens because the model is trained to perform one-step ahead forecasting and the outputs of the previous steps, with their biases, are used as input for the next steps. On the other hand, this strategy keeps the time dependence between the values.

1.3.2.2 Direct strategy

This strategy seeks to find the stochastic relationship between two observations on
distant periods of time. Therefore, to forecast a horizon $H$, as many $g_h$ models are trained over the time series $[y_1, ..., y_N]$ where:

$$y_{t+h} = g_h(y_t, ..., y_{t-d+1}) + w$$

with $t \in \{d, ..., N-H\}$, $h \in \{1, ..., H\}$.

To estimate the horizon $H$, each of the computed models are used to forecast their $h^{th}$ step,

$$\hat{y}_{t+H} = \hat{g}_h(y_t, ..., y_{t-d+1})$$

This strategy has the benefit that it does not use forecast values to predict future values, therefore the problem of accumulation of errors does not apply to this strategy. On the other hand, the fact that $H$ independent models are trained, does not match the nature of a time series where complex dependencies over time between the observations are often present.

1.3.2.3 DirRec strategy

The DirRec strategy (Sorjamaa & Lendasse, 2006) is a combination of the Direct and Recursive strategies. It achieves the objective of forecasting a horizon $H$ by computing $h$ models $g_h$ where,

$$y_{t+h} = g_h(y_t, ..., y_{t-d+1}) + w$$

with $t \in \{d, ..., N-H\}$, $h \in \{1, ..., H\}$.

To estimate the horizon $H$, the $g_h$ learned models are used as follows,

$$\hat{y}_{t+1} = \hat{g}_h(y_t, ..., y_{t-d+1})$$

$$\hat{y}_{t+H} = \hat{g}_h(\hat{y}_{t+H-1}, ..., \hat{y}_{t+1}, y_t, ..., y_{t-d+1})$$

This way, the possible stochastic relations between distant points in time is addressed while preserving the complex relations within the time series. In other words, tailored models are learned for every step instead of relying on only one model predicting one-step ahead. On the other hand, this strategy uses estimated values which can bring the problem of accumulating errors. Despite this, early
results show contradictory results of its performance in comparison with the Direct and Recursive strategies. (Sorjamaa & Lendasse, 2006. Taieb et al., 2012).

1.3.2.4 MIMO strategy

The Multi Input Multi Output (MIMO) strategy only learns from the series \([y_1, ..., y_N]\), one model \(G\) where the output is a time series that comprehends the whole desired horizon \(H\),

\[
[y_{t+H}, ..., y_{t+1}] = G(y_t, ..., y_{t-d+1}) + ,
\]

with \(t \in \{d, ..., N-H\}\), \(F : \mathbb{R}^d \rightarrow \mathbb{R}^H\) is a vector valued function, and \(\epsilon \in \mathbb{R}^H\) is a noise vector with a covariance that is not necessarily diagonal.

The forecast is performed by a single estimation task in the next way,

\[
[\hat{y}_{t+H}, ..., \hat{y}_{t+1}] = \hat{G}(y_t, ..., y_{t-d+1}).
\]

With the MIMO strategy the stochastic relation between the estimated values can be preserved. Avoiding to induce independence between periods unlike the Direct strategy as well as the accumulation of errors found in the Recursive strategy. On the negative side this approach might lack flexibility as it restricts forecast of all the steps to one single time structure.

1.3.2.5 DIRMO strategy

To capture the model flexibility of the Direct strategy and the complex output relations of the MIMO strategy, the DIRMO strategy (Taieb et al., 2009) produces \(n\) forecasts blocks of size \(s\), where \(s \in \{1, ..., H\}\), \(n = \frac{H}{s}\).

This strategy learns \(n\) models \(G_p\) from the series \([y_1, ..., y_N]\) where

\[
[y_{t+ps}, ..., y_{t+(p-1)s+1}] = G_p(y_t, ..., y_{t-d+1}) +
\]

with \(t \in \{d, ..., N-H\}\), \(p \in \{1, ..., n\}\) and \(G_p : \mathbb{R}^d \rightarrow \mathbb{R}^s\) is a vector valued function if \(s>1\).

The forecast is performed by a \(n\) estimation task in the next way,
By partitioning the output vector, it gives the MIMO strategy flexibility over the output structures but it introduces conditional independence over the forecast blocks. Additionally, a new parameter $s$ has to be estimated.

1.3.3 Models

Multiple different models have been used in the forecasting literature, SVR is used by Hu et al. (2013), different ARIMA and a NN is used by Veit et al. (2014).... An artificial neural network (NN), a Random Forest (RandFor) and K-Nearest Neighbours Regression (KNNR) algorithms are used in this thesis because of different reasons. First, the different models to be trained have to support multi-output in their implementation in the python library scikit-learn 0.18.1 as it is the library used to develop this thesis. Second, the selection of the NN is due to its use in previous energy forecasting work (Mandal et al. 2006, Veit et al, 2014). Third, the selection of the KNNR algorithm is due to its similarity with the lazy learning algorithm proposed by Bontempi (2008) and Taieb et al. (2012); in their papers this lazy learning algorithm shows better results for multi-output strategies than the single-output ones. Finally, Random Forest has been selected because of its application to wind energy production multi-step ahead forecast (Fugon, 2008) that can give a reasonable method to forecast weather dependent energy production means.

1.3.3.1 Artificial Neural Network

Artificial Neural Networks (NN) have been used in energy forecast in various occasions (Jebaraj & Iniyan, 2006, Veil et al. 2014).

A NN is a supervised algorithm that maps a function $f : \mathbb{R}^n \rightarrow \mathbb{R}^o$. Where the output $y$ of the function is the result of the processing of an activation function $\varphi(\cdot)$ over the sum of the dot product between the observation’s features and the neurons weights. Expressed in a mathematical form is formulated as:

$$ [\hat{y}_{t+p^s}, ..., \hat{y}_{t+(p-1)s+1}] = \hat{G}_p(y_t, ..., y_{t-d+1}). $$
\[ d_{j}^{l+1} = \sum_{i=1}^{m} \varphi(W_{j|i}^{l}x_{i} + b_{j}), \]

being \( d_{j}^{l+1} \) the \( j^{th} \) neuron in the layer \( l+1 \), \( l \in \{0, ..., \text{number of layers}\} \). \( x_{i} \) is the \( i^{th} \) input in the layer \( l \). \( W_{j|i}^{l} \) is the weights that relate all the \( i \) inputs in the \( l^{th} \) layer to the output neuron \( j \).

This feed forward network has been trained with the Adaptive Moment Estimation (Adam) algorithm (P. Kingma and Lei Ba, 2015). This algorithm is a gradient based stochastic optimization algorithm, that uses the first and second momentum of the gradient to compute adaptive learning rates. The authors have proved better and faster convergence of the algorithm in comparison to others as SGD or AdaGrad.

1.3.3.2 Random Forest

A Random Forest (Breiman, 2001) Regression (Liaw and Wiener, 2002) is an ensemble learning method that uses \( n \) independent regression trees by training them to bootstrap samples of the dataset. The algorithm is based in regression trees where a regression is constructed by iteratively selecting the variables that offer the greatest information gain. Regression trees tend to overfit the training data, to avoid the problems of robustness random forest performs bootstrapping of trees on the dataset.

The algorithm works as follows:

1. Selects \( n \) bootstrap samples of the data.
2. Fits a regression tree on each sample, where at each node the split is made by choosing the best feature, the one which reduces the impurity of the model, over a random sample of \( m \) features.
3. Averages the predictions performed by the \( n \) trees.

Random Forest have been used in the forecasting literature to predict the short term energy generation on wind farms (Fugon, 2008) and for one step ahead forecasting of energy production in a solar farm (Zamo et al, 2014).
1.3.3.3 K-Nearest Neighbours Regression

K-Nearest Neighbours Regression (Altman, 1992) is a non parametric lazy learning algorithm that performs predictions by averaging the closest k-nearest observations. Being \( k \) a parameter that indicates the number of neighbours to be averaged. This algorithms delays all the computations until the prediction is required. Its algorithms is described as follows.

1. A new observation is given to predict.
2. The original dataset is orderd based on the distance to this observation.
3. The prediction is performed by computing the weighted average of the K-Nearest neighbors to the desired observation.

1.3.4 Measures of performance

To evaluate the performance of the models, three measures are used.

R squared states a comparison between the squared error of the forecast and the squared error of the mean.

\[
R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}
\]

It might not be the best option to evaluate the performance of models forecasting time series due to the inherent variability of these. Despite this, it is used to select the models in the parameters tuning phase. This is because its robustness, as it is defined for all \( \sum (y_i - \bar{y})^2 \neq 0 \).

The measure to compare the different strategies is the Mean Absolute Percentage Error (MAPE). It is computed as follows,

\[
MAPE(\%) = \frac{1}{N} \sum \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100
\]

This measure evaluates the forecast based on the error it produces as a proportion of the actual value. This feature helps to evaluate the quality of the forecast on the
distance to the actual value. It is extensively used in the forecasting literature and helps with the comparison of the results among papers. This measure has as downside that it is not defined for $y_i = 0$ and that positive errors have a greater effect on the measure than the negative ones, Bao et al. (2013).

Finally, the Mean Absolute Error, defined as,

$$ MAE(\%) = \frac{1}{N} \sum_i |y_i - \hat{y}_i| $$

is used to evaluate the viability of the possible applications as they are only evaluated in relation to the energy disparity between the actual amount needed and the amount forecasted.
Chapter 2: Methodology

This section describes the processes carried on the data to generate the comparison between strategies. Figure 3 describes the processes used which are detailed in the following subsections.

The technology used is based on python 3.5.3, with special mention to the libraries pandas 0.19.2 for data structures and transformation and scikit-learn 0.18.1 for model implementation. The implementation of the strategy forecasting selection is an open sourced created by the author of this thesis (Martín-Roldán, 2017).

![Figure 3: Methodology process.](image)

2.1 Data collection

2.1.1 Energy feeds

This study has been created around volunteer provided data from the Emon CMS community (Emoncms, 2017). The volunteers signed up by filling a public form
where their email and feed’s information was collected. No personal information was collected, nonetheless, they were informed their data can be deleted and retrieved under request. The data collected from this form includes an API reading key for the Emon CMS platform that works as an individual identifier, an email address, their physical address or/and coordinates location to obtain weather data. The locations are spread around the United Kingdom and France.

The data was collected over long periods of time by the devices installed in the volunteers’ houses, stored in the EmonCMS servers and accessed in bulk the day 15 May 2017 and it has a volume of 4.8GB for all the 10 feeds. Its granularity is 10 seconds for each feed. The distribution of the feeds are:

- 4 feeds that measure energy house consumption.
- 3 feeds that measure solar production.
- 3 feeds that measure grid input in the houses that have solar production.

In cases where the individual setting does not have solar power production, the House Consumption equals the input grid.

Due to the high volume of data, only two feeds are used on this thesis, one from a house without a solar array, and the solar feed and grid input feed from a house with an array of solar panels. Table 1 shows the feed’s descriptive statistics.

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Grid input</th>
<th>Solar production</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total observations</td>
<td>20.925.334</td>
<td>17.265.237</td>
</tr>
<tr>
<td>Non NaN observations</td>
<td>16.291.910</td>
<td>14.010.510</td>
</tr>
<tr>
<td>Proportion of NaN</td>
<td>22%</td>
<td>18%</td>
</tr>
<tr>
<td>mean</td>
<td>405,2718 Watts</td>
<td>241,69 Watts</td>
</tr>
<tr>
<td>std</td>
<td>611,0746 Watts</td>
<td>489,8369 Watts</td>
</tr>
<tr>
<td>min</td>
<td>-29.733 Watts</td>
<td>-32.652 Watts</td>
</tr>
<tr>
<td>max</td>
<td>46.377 Watts</td>
<td>24.852 Watts</td>
</tr>
</tbody>
</table>

It’s worth mentioning that the retrieved data measurement is the average power in watts over the 10 seconds period. Therefore, if we want to obtain the energy consumed, the data must be multiplied by the time passed in the averaged period,
here 10 seconds. The result of this transformation is:

\[ \text{Energy}_{10\text{ Seconds}} = \text{Watts} \times 10 \text{ Seconds} \]

The retrieved data has a high grade of heterogeneity due to its volunteer component and the 'do it yourself' character of the Emon CMS community. Among other things, the data feeds have different lengths, as shown in Table 1, depending on when the owners set up their measurement system. In addition, its configuration can change over time or the feeds may have long periods of empty data due to technical issues or as the users upgrade their set ups. A description of the feeds’ working and non working periods after the feed aggregation described in section 2.2 can be found in Table 2. This heterogeneity will have a big impact in the suitability of the data for machine learning use; this is addressed in the cleaning NaN section.

Table 2: Grid input and solar production feeds working and NaN periods description. The feeds have a granularity of 15 minutes. For each feed the number of working periods and NaN periods are the same but the duration of the working periods is greater than the NaN periods.

<table>
<thead>
<tr>
<th></th>
<th>Grid input</th>
<th>Solar production</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Working periods</td>
<td>NaN periods</td>
</tr>
<tr>
<td>Number of periods</td>
<td>1528</td>
<td>1528</td>
</tr>
<tr>
<td>Average time duration</td>
<td>1 days 10:07:58</td>
<td>03:23:10</td>
</tr>
<tr>
<td>Duration time</td>
<td>4 days 23:35:26</td>
<td>16:01:27</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimum time period</td>
<td>00:14:00</td>
<td>00:14:00</td>
</tr>
<tr>
<td>Maximum time period</td>
<td>66 days 19:29:00</td>
<td>13 days 23:14:00</td>
</tr>
</tbody>
</table>

2.1.2 Weather data

The Weather Underground API is used to obtain weather data for individual locations (Weather Underground, 2017). The service gathers public weather stations data at a constant basis of 30 min granularity. The variables gathered are:
Numerical; *Temperature in Celsius, Pressure in millibars* and *Humidity,* the *wind direction* in grades, *wind speed in meters,* the volume of *precipitations* in cm³ and *Dewpoint Temperature in Celsius.* The categorical variable *General Conditions.* Dummy variables to describe the type of precipitation; *fog, hail, rain, snow* and *thunder.*

For each location a weather data series is accessed; the period obtained goes from the time when its respective feed has the first values until the 5\textsuperscript{th} of May of 2017. This is the day when the data is accessed from Weather Underground. The data has been collected and stored for its public access from the METAR weather stations. At the moment of access and for all 10 household feeds the size of the data equals 1.2 GB.

The granularity of the data is one observation every 30 minutes therefore no aggregation or disaggregation has to be performed over the weather data.

This dataset can be considered quite clean as most of the locations have complete data over the stated time period. In some of the locations the data has a small time misalignment (some minutes) with the periods of time used for the feeds. Some have also minor gaps in the data with a maximum gap of 2 hours.

### 2.2 Cleaning data

#### 2.2.1 Energy feeds

As previously mentioned the data has missing information, due to multiple factors.

First of all, the devices have dropouts of observations now and then, with a random distribution and a span shorter than 30 minutes. By imputing the average of the 30 minutes period the posterior change of granularity will not be affected.

Second, there are other missing data periods among the series that last more than 30 minutes. These appear due to human activity, when the owners of the devices change or upgrade the setup or due to serious malfunctioning of the devices. These periods are more complicated to deal with as there are seasonal effects in the data that will be missing. These missing values cannot be just removed from
the dataset as the data has a temporal component that makes the observations make sense only in the context of their historical data.

The size of the missing data is variable between feeds being some of the feeds free from empty observations. As it is shown in Table 1 the size of empty data as a proportion of the total size in the feeds used in this paper equals to 22% and 18%, being two of the feeds with the biggest proportion.

In initial runs an approach of deleting the previous section of a time series with the presence of gaps of bigger than a week was discarded. For some feeds this approach deleted years worth of data, leaving really short time series, as the gaps were located in the middle or at the end of the series.

Different techniques can be taken to interpolate the missing data but all of them have a trade off between the different temporal components and the bias generated.

A Monte Carlo method inspired experiment (Paxton et al.2001) is used to select the best interpolation method to create the smallest bias while interpolating missing data. The experiment work as follows:

First it takes random days in the series where there is no missing data on any of its points. Second, it interpolates the series using the different methods subject to study. Third, it computes the Mean Absolute Percentage Error (MAPE) (see section 2.7) with the original data and the interpolation. Forth, steps 1 to 3 are repeated 4.000 times, which by the Law of the Large numbers can be assumed as an unbiased estimator of the expected MAPE of a given interpolation method. Finally, it computes the average of the MAPE obtained by the different interpolation methods.

The interpolation method that has obtained the smallest result during this experiment is chosen to interpolate the data.

While designing the experiment, Paxton et al. (2001) set a maximum sample of 1000 experiment repetitions. To ensure that enough sample are taken and thanks to its relatively low computing cost, in this thesis the experiment performs the same
task 4000 times. Even though this experiment provides the method that gives the smallest bias on certain points, it does not take in account how these methods affects the overall accuracy of the different models. The selected interpolation method might not be the best to improve overall accuracy. Therefore it would be interesting to study the impact of the different methods over the models’ forecasting capabilities.

The methods to be tested are: the implementation of scikit-learn of linear interpolation and quadratic interpolation, imputing the previous week’s value for a given moment of time, imputing week’s moving average of a time slot and imputing the week’s moving median over the time slot.

Imputing the week’s moving average over the selected feeds gives the smallest MAPE and therefore is the one chosen to interpolate the data. Using this the energy data is finally clean of gaps but this method’s side effect is the loosing of the data’s seasonality.

2.2.2 Weather data

The treatment of the weather data has been carried out by aligning the time series to the previous feeds observation and in the case of interpolating. This alignment is made as follows:

If the minute in the weather data is comprehended between 0 and 30 minutes, excluding these values, then the minute is set equal to 0. If the minute registered for the observation is greater than 30 but smaller than 60, this is corrected and set to 30.

In the case where the observations for a given time are missing, the missing values are filled by imputing the next valid observation over the empty period. Again as in the case of the energy feeds there is a trade off between the time component of the data and the bias introduced while interpolating. In this case no analysis of the impact of the interpolation method over the data has been carried out.
2.3 Aggregation

As mentioned in section 2.4.1.1, the data in the feeds represents the instant power in watts * second over 10 seconds periods where it is assumed to be constant. The desired output to estimate is the energy measured in kilowatts hour over 30 minutes periods. The transformation performed in the data is described by the next equation:

\[
Energy_{30min} = \sum_{i=1}^{180} p_i \times \frac{\Delta t}{3600 \times 1000}
\]

Where \( Energy_{30min} \) is in kilowatts * hour, \( p_i \) is the power measure on Watts per second and \( \Delta t \) is the length of the period in seconds. This aggregation leaves the variable \( Energy_{30min} \) in a range of (0, 3). The small range of the variables and the deterioration of the accuracy in early experiments, gives no arguments for the normalization of the series.

2.4 Deseasonalization

Given the nature of the data short periods of time have to be considered to find out seasonality. The fact that in our daily lives we perform the same activities at the same time of the day makes that small periods like half an hour subject to be tested for seasonality. The same periods in other applications would make no sense.

For this thesis the periods of time proposed to be corrected from seasonality are: the half hour of the day, the day of the week, the month of the year and the quarter of the year.

To be sure of the necessity of a seasonal transformation, one way ANOVA tests are performed on the series, one for each type of possible seasonal effect. The tests have as null hypothesis that all the means of the data grouped by 30 minutes time slot, day of the week, month or quarter are equal. This hypothesis is rejected for all the four levels of aggregation.

The deseasonalization process is done by following the process described in the section 1.3.1. The effects to correct are the half an hour of the day, the day of the
week, the month of the year and the quarter of year, therefore the
deseasonalization process is performed four times. Resulting in the time series
\( u_t^Q(i) \) which is corrected from all seasonal effects. This time series is the one
in the following steps. The final step of forecasting is to reintroduce the seasonal
effect in the series.

### 2.5 Feature creation

To be able to feed the machine learning algorithms a transformation of the feeds is
carried out. The original time series \( y_N \) of size \( N \), is converted into a matrix
\( F_{(N-H-d+1) \times d} \) where each one of the rows is an individual time series of size \( d \)
with a displacement of one period from the previous row. Also the response matrix
\( R_{(N-H-d+1) \times H} \) is created. Each one of its observations are the \( H \) future values to
their corresponding observation in the F matrix.

For every feed that is parsed, five time-features are created. A numeric feature
called grouper which represents the 30 minutes period of the day when the
observation is taken. Its range is (0-47) respectively. Also computed is the day of
the week the data was taken, range (0-6), the month of the year with a range (1-12)
and the quarter of the year (1-4). In addition to this a dummy feature (0,1) states if
it is a working day or not.

Finally every row of the matrix F is merged with its corresponding weather data
for that period of time. In this operation the number of observations of the biggest
data sets is reduced to match the size of the smallest one. For example in this
thesis, where the available household feeds are available until the 15\(^{th}\) of May and
the weather data is gathered until the 5\(^{th}\) of May, the F matrix is reduced to the size
of weather data.

Afterwards the dataset is divided into train 70\%, test 20\% and validation 10\%.
This division is performed in a temporal fashion being the train data the oldest and
the validation the newest. The reason of it is that when forecasting, the most
important property is the capacity of the model to accurately forecast the future,
therefore is needed an approach that evaluates on the future rather than the past.
The validation division is added to be able to evaluate the model on unseen data rather than the data that has been used to select the best model.

2.6 Tuning parameters

There is at least one parameter in every model to be tuned. To select this parameters of the different models two searches are used. For the neural network a random search is used where 10 configurations of the network are randomly selected among the options of parameters. These parameters were selected after initial runs where the options where broader. The parameters to be tuned are:

- **Hidden_layer_sizes**: \{(600, 300), (600, 300, 100)\} are the two possible network configurations. One with two layers and 600 neurons in the first layer and 300 per second. The other has 3 layers with 600, 300 and 100 neurons respectively. This selection of neurons is an educated guess based on (How to choose the number of hidden layers and nodes in a feedforward neural network?, comp.ai.neural-nets FAQ, Part 3 of 7: GeneralizationSection - How many hidden layers should I use? and comp.ai.neural-nets FAQ, Part 3 of 7: GeneralizationSection - How many hidden units should I use?)

- **Activation functions**: \{'logistic', 'relu' (rectified linear unit)\}. Being \(relu = max(0, x)\).

- **'alpha' or regularization parameter**: \{1e-5, 1e-4, 1e-3\}.

- **'tolerance'**: \{1e-4, 1e-5\} which is the level which the network is considered to have converged to a solution.

For the Random Forest and KNNR the search was made with a grid of parameters. The Random Forest searches over a maximum depth of 30, 50 or 70 leafs in its trees. For the KNNR the search is made over the number of neighbours, (5, 20, 50 or 100) and their type of weighting, 'uniform' or 'distance'.


Chapter 3: Results

The results presented in this chapter were obtained from the computation of a total of 1,972 different models with an approximate total computing time of 90 hours.

The analysis of the performance of the different strategies and models has to be done both as a whole time series and individually for every forecasting step. As a whole time series in order to be able to find the best strategy and model to forecast to the desired horizon. Individually to observe the evolution of the error for further forecasting horizons and if some of the strategies perform better than others in some periods. This chapter is organized in the following way: First, an average MAPE(%), R squared and sum of MAE comparison is carried out in both feeds for all strategies and models. Second, for both feeds a model comparison within the strategies is presented. Finally, a comparison of strategies for the different models is exposed.

Table 3: Grid input average MAPE, R squared and sum of MAE over the forecasted horizon. RF = Random Forest, NN = Artificial Neural Network, KNNR = K- Nearest Neighbours Regression

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Model</th>
<th>AVG MAPE(%)</th>
<th>AVG R squared</th>
<th>SUM MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIRMO</td>
<td>RF</td>
<td>48.48</td>
<td>0.16</td>
<td>5.00</td>
</tr>
<tr>
<td></td>
<td>NN</td>
<td>51.15</td>
<td>0.18</td>
<td>5.14</td>
</tr>
<tr>
<td></td>
<td>KNNR</td>
<td>51.10</td>
<td>0.11</td>
<td>5.19</td>
</tr>
<tr>
<td>MIMO</td>
<td>RF</td>
<td>46.69</td>
<td>0.15</td>
<td>4.92</td>
</tr>
<tr>
<td></td>
<td>NN</td>
<td>46.45</td>
<td>0.14</td>
<td>4.87</td>
</tr>
<tr>
<td></td>
<td>KNNR</td>
<td>50.99</td>
<td>0.11</td>
<td>5.19</td>
</tr>
<tr>
<td>DIR</td>
<td>RF</td>
<td>56.44</td>
<td>0.11</td>
<td>5.44</td>
</tr>
<tr>
<td></td>
<td>NN</td>
<td>47.97</td>
<td>0.17</td>
<td>4.95</td>
</tr>
<tr>
<td></td>
<td>KNNR</td>
<td>51.18</td>
<td>0.11</td>
<td>5.20</td>
</tr>
<tr>
<td>REC</td>
<td>RF</td>
<td>115.73</td>
<td>-0.62</td>
<td>8.97</td>
</tr>
<tr>
<td></td>
<td>NN</td>
<td>69.21</td>
<td>-0.11</td>
<td>6.44</td>
</tr>
<tr>
<td></td>
<td>KNNR</td>
<td>65.78</td>
<td>-0.09</td>
<td>6.25</td>
</tr>
</tbody>
</table>

As shown in Table 3, the multi-output strategies make a better forecast of the grid input. Two of the models of the MIMO strategy have the smallest average MAPEs....
over all periods followed by the Direct strategy and the DIRMO strategy. All three Recursive strategy models perform worse than any other model for any other strategy.

Within strategies, the models to be selected based on the overall performance of the forecast are for the strategies DIRMO, MIMO and Direct or Random Forest or Neural network. Being the Recursive strategy the only one to have the K-Nearest Neighbors as a model of choice.

The values of the selection measure R squared are considered low, being negative for the Recursive strategy. This happens because the model’s estimated errors are greater than the errors produced by the average of the target variable. Also no relation is shown between the best results for R squared and the best results in the target variable. Also it can be perceived that MAPE and R squared have a positive correlation in general terms, showed by the deterioration of both measures while using the Recursive strategy.

Regarding the MAE, it shows a high positive correlation with the MAPE, being the smallest daily sum of absolute errors equal to 4.87 kwh. The average consumption over the period of 30 minutes equals to 0.21 kwh. Defining the average day as the average consumption in one period times the number of periods that compose a day, in this case 48. Therefore the energy consumed in the average day equals to 10.08 kwh. Therefore, the smallest MAE obtained by this forecast equals 48% to the average daily consumption.

The sum of MAE obtained for the solar energy production forecast is equal to 5 kwh. Taking in account that the average energy produced every half an hour is equal to 0.15kwh, this gives that the average day produces 7.19 kwh of energy. Therefore the average absolute error produced by this forecast is equal to 69% of the production of the average day.

In addition Table 4 shows an improvement in the model selection measure, R square, of Table 3. This improvement it does not translate into an improvement of the forecast quality measure MAPE. On the contrary, the latest shows values at
least 5 times higher than in Table 3.

Table 4: Solar production average MAPE, R squared and sum of MAE over the forecasted horizon. RF = Random Forest, NN = Artificial Neural Network, KNNR = K-Nearest Neighbours Regression.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Model</th>
<th>AVG MAPE(%)</th>
<th>AVG R squared</th>
<th>SUM MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIMO</td>
<td>RF</td>
<td>509,1</td>
<td>0,5</td>
<td>5,3</td>
</tr>
<tr>
<td></td>
<td>NN</td>
<td>688,0</td>
<td>0,6</td>
<td>5,0</td>
</tr>
<tr>
<td></td>
<td>KNNR</td>
<td>1.400,4</td>
<td>0,3</td>
<td>6,9</td>
</tr>
<tr>
<td>DIRMO</td>
<td>RF</td>
<td>323,2</td>
<td>0,5</td>
<td>5,1</td>
</tr>
<tr>
<td></td>
<td>NN</td>
<td>682,3</td>
<td>0,6</td>
<td>5,2</td>
</tr>
<tr>
<td></td>
<td>KNNR</td>
<td>1.400,4</td>
<td>0,3</td>
<td>6,9</td>
</tr>
<tr>
<td>DIR</td>
<td>RF</td>
<td>309,7</td>
<td>0,5</td>
<td>5,1</td>
</tr>
<tr>
<td></td>
<td>NN</td>
<td>662,2</td>
<td>0,5</td>
<td>5,3</td>
</tr>
<tr>
<td></td>
<td>KNNR</td>
<td>1.419,3</td>
<td>0,3</td>
<td>6,9</td>
</tr>
<tr>
<td>REC</td>
<td>RF</td>
<td>5.402,6</td>
<td>-0,6</td>
<td>12,0</td>
</tr>
<tr>
<td></td>
<td>NN</td>
<td>4.249,1</td>
<td>-0,6</td>
<td>11,6</td>
</tr>
<tr>
<td></td>
<td>KNNR</td>
<td>4.097,7</td>
<td>-0,2</td>
<td>10,6</td>
</tr>
</tbody>
</table>

Figures 4.1 - 4.4 show the time evolution for all strategies and models of the forecast errors during the forecasting horizon of the grid input feed. Figures 5.1 - 5.4 show the time evolution for all strategies and models of the forecast errors during the forecasting horizon of the solar production feed. Figures 6.1 - 6.3 show the comparison in performance of the different models changing the strategy used to forecast the grid input feed. Finally figures 7.1 - 7.3 show the MAPE (%) variation of the same model using different strategies to forecast the solar production feed.

Figure 4.1 shows how the NN has a high variance in its MAPE(%) during the different periods ahead to forecast. The fact that multiple models have to be fit and some of them are not able to converge to a solution makes that even the model that has the best performance overall, shown in Table 3, for some of the periods (i.e. 4, 8 and 10.5 hours) has the worst performance of all models. Figure 4.2 is a clearer example of the lack of model convergence than Figure 4, it can be seen that the best model overall for the strategy, NN, has problems converging for some periods from 6.5 to 12 hours, going from having the best performance to the worst. This
period is larger because in the DIRMO strategy the model predicts 3 hours ahead at the time and two consecutive models have fail to converge. On contrast Figure 4.3 shows no convergence problems for any of its models because the MIMO strategy is fitting at once the models. Figure 4.4 shows the problem of recursive forecast, the accumulation of errors. This is shown by the growth of the MAPE (%) from levels similar to the other strategies to levels of error 3 times greater than the others. Finally it gets to levels of error similar to the initial ones; this initially unexpected reduction of the error is given by the fact that after a few periods ahead of forecast, the Recursive strategy gives an almost non variant forecast, while the variable’s pattern has a high variation over the day but similar values when the daily cycle is over thus reducing the distance with the forecasted values.

Figure 4.1 : Direct strategy MAPE% for all forecasting periods and models.

Figure 4.2: DIRMO strategy MAPE% for all forecasting periods and models.

Figure 4.3: MIMO strategy MAPE% for all forecasting periods and models.

Figure 4.4: Recursive strategy MAPE% for all forecasting periods and models.

Figure 4: Grid input models MAPE(%) comparison within strategies.
Figures 5.1 - 5.3 show that the Random Forest model outperforms the other for almost all periods. They also show the problem of the convergence of the Neural Network (5.1, 4.5 hours ahead and 5.2, 21 to 24 hours ahead). Figure 5.4 shows similar results to those described for Figure 4.4.

**Figure 5.1**: Direct strategy MAPE% for all forecasting periods and models.

**Figure 5.2**: DIRMO strategy MAPE% for all forecasting periods and models.

**Figure 5.3**: MIMO strategy MAPE% for all forecasting periods and models.

**Figure 5.4**: Recursive strategy MAPE% for all forecasting periods and models.

**Figure 5**: Solar production models MAPE(%) comparison within strategies.

Figure 6.1 - 6.3 provides a clear insight of the problem of error accumulation for the Recursive strategy for all models. In addition 6.2 shows the problem of the neural network convergence in the for the strategies Direct and DIRMO. Another relevant insight is Figure 6.3 is the close performance of the K-Nearest Neighbours
in all strategies but the Recursive one; in this case the MIMO strategy wins with a really low margin. This can also be seen in Figure 7.3.

Comparing Figures 7.1 and 7.2 shows that for solar production forecasting the variance of the MAPE (%) is greater for the Neural Network than for the Random Forest. In Figure 7.1 we can see an increase of the MAPE(%) over time for the strategies Direct, DIRMO and MIMO but such increment is not comparable to the error of the Recursive strategy.
Figure 7.1: Random Forest MAPE(%) comparison among strategies. To be able to distinguish patterns within the best performers a maximum limit of 1000 has been set for the vertical axis. This leaves out of the graph many values of the Recursive strategy, which has a maximum of 11750. The recursive strategy values can be found in Figure 5.4 under the label of RandFor.

Figure 7.2: NN MAPE(%) comparison among strategies. To be able to distinguish patterns within the best performers a maximum limit of 1600 has been set for the vertical axis. This leaves out of the graph many values of the Recursive strategy which has a maximum of 9700. The recursive strategy values can be found in Figure 5.4 under the label of NN.

Figure 7.3: K-Nearest Neighbours Regression MAPE(%) comparison among strategies.

Figure 7: Solar production forecast comparison among strategies. The performance of the same model is compared among the different strategies.
Chapter 4: Discussion

Table 3 shows a better performance of the Multi-Output strategies with a Neural Network model; in contrast Table 4 shows that Direct strategy using a Random Forest model generates a smaller error. These contradictory results might be explained by the nature of the data to be forecasted. While input grid is a highly behavioural dataset, made by people’s actions, solar production is almost a deterministic dataset where the randomness shown in the low scale of solar panels is missing in the general dynamic system of weather conditions. This makes the grid dataset more suitable to be analysed with Neural Networks over a period of time, being the network capable of capturing the complex time relations that happens within the feeds time series. On the other hand the linear dependency of the solar production on weather conditions may make a Direct strategy with a Random Forest model more suitable to generate forecast for this feed.

The actual values of the results in Table 3 and Table 4, show that for the grid input feed, the best model gives an average error that can be considered acceptable given the low detail of the data feeds, only by measuring household energy consumption. On the other hand, the error given by the best model that forecast solar production, makes this forecast unreliable. It is interesting to observe that the R squared shows generally accepted low values for the Table 3 while relatively high values for Table 4. This suggests that R squared is not an optimum measure for model selection as it seems to have a negative relation with the objective performance measure MAPE(%).

While evaluating the best model to forecast grid input the average MAPE (%) points that the Neural Network with a MIMO strategy has the best performance overall. In addition to this, Figure 6.2 shows that over time the DIRMO and Direct strategies provide many of the smallest punctual errors in different periods of time, but the problem of non convergence of the Neural Network makes these strategies lose the best position. Therefore this methodology favours stability in the prediction over time and points out that a control on the convergence of the models
should be taken in account to improve accuracy. In addition, this convergence problem can be addressed by using an optimized hyperparameters search rather than a randomized search over a small grid of parameters.

The accuracy of the best model shown in Figure 4.3 is an improvement to the best results obtained in one of its two datasets by Veit et al, (2014) on a more detailed household energy consumption data. In addition, while Veit’s performance is reduced as the horizon gets further, the results provided in this thesis show a stable MAPE (%) over the whole forecasting horizon.

While studying the performance of the best solar energy feed forecast for any given model and strategy shown in Figure 7.1, it can be seen that its accuracy is not adequate for any given step ahead being the MAPE (%) greater than 100% from 1 hour ahead. This lack of accuracy might be due to the negative relation between the model selection measure R squared and the performance measure MAPE (%). This effect should be further studied and check the improvement on the model's performance while using another model selection measure.

Although the mixed MAPE (%) obtained can lead to the consideration of different business applications, the MAE obtained for both feeds, shown in Table 3 and Table 4, is big enough to make unfeasible the use of this methodology for such applications.

Figures 6.3 and 7.3 show that K-Nearest Neighbours performance is very close between the MIMO, DIRMO and Direct strategies but slightly better for the MIMO strategy for both feeds. This results are in line with those obtained by Taieb et al. 2012, but as the Figures 4 and 5 show; for both feeds the use of other models with different strategies, Neural Network with a MIMO strategy in the grid input feed and Random Forest with a Direct strategy in the solar production feed, outperforms the K-Nearest Network results. In the conclusion of their paper they propose multi-output strategies as primary strategies to be used. This thesis gives the intuition that the multi-step ahead forecasting strategy of choice is problem dependant and therefore a strategy selection step has to be taken in the forecasting
methodology.

The fact that in both datasets the DIRMO strategy is closer to the best solutions makes it a reasonable candidate for the initial exploring analysis. Its implementation allows to obtain the Direct strategy by setting the parameter $s = 1$ and the MIMO strategy by setting $s = H$. This means that we can explore the presence of the time structures on the data with a smaller cost than running a comparison between Direct and MIMO, by setting a comparison of a MIMO and two DIRMO with two different parameters $s$. Analyzing those results, if the DIRMO strategy with the smallest $s$ has the best performance, then a Direct strategy might be considered, in case that is the MIMO or the DIRMO with the biggest parameter $s$ the ones with the lowest MAPE(%) then no further analysis is needed.

In contrast to the previous paragraph the problem of error accumulation is noticeable in both datasets for all models using the Recursive strategy. This points out that this strategy is not suitable for periodic long term forecasting as the required for this thesis. Also given that the K-Nearest Neighbors algorithm is never the model of choice except for the Recursive strategy, shows that we can omit this model in the study of the given problem. Therefore, it is not recommended the inclusion of the Recursive strategy and K-Nearest Neighbors algorithm in the multi-step ahead forecast problem of energy consumption and production in households.

The results presented in Chapter 3 have room for improvement. Apart from the suggestions presented earlier in this chapter, improvement can be produced through the interpolation methods used to clean the data. In this thesis a simple selection is carried on in over a small array of simple interpolation methods, section 2.2. This selection is performed using a measurement taken from the data, but there is no analysis of the impact of the interpolation methods over the forecasting accuracy. Therefore future research might find worth to carry this analysis to improve the given results.
Chapter 5: Conclusion

The new contribution of this thesis is to address the problem of household energy production and consumption one-day ahead forecasting, by performing a multi-step ahead forecast comparison to find suitable models and strategies, to obtain the best possible results and to evaluate those results for different applications.

The forecasting accuracy obtained by the proposed methodology over the household energy consumption has a better performance that the state of the art forecasting models proposed by Veit et al. (2014) using a Recursive strategy for one of their datasets. This might be an indicator of the better performance of multi-step ahead strategies over the Recursive one. Although the results are satisfactory, the size of the error in terms of volume of energy given by the MAE makes this methodology unsuitable for other applications than for information purposes.

The discrepancy between the best performing strategy while forecasting energy consumption (MIMO) and energy production (Direct) leads to the conclusion that the forecasting strategy of choice is problem dependent. In consequence a forecasting selection step is recommended while developing new applications.

The results presented in Chapter 3 shows that the Recursive strategy performed the worst by far for both series, is also the most complex strategy to implement. Therefore its use is discouraged for multi step ahead forecasting purposes.

The fact that all the models of the DIRMO strategy were close to the top performers, suggests that it can be an adequate initial guess; it captures the data structure when it exists and provides flexibility when it does not. In addition, the implementation of this strategy allows to easily study the winning strategies, as just the parameter $s$ needs to be changed to obtain both the Direct and MIMO
strategies.

This thesis has many limitations given the size of the area of study and the time limits. Therefore, further research should focus on extending this analysis to new datasets in order to verify the robustness of the results. Also in order to tackle the convergence problem of the Neural Network for the Direct and DIRMO strategies, optimized hyperparameters search along with convergence control methods can be studied. Given the negative relation in performance of by the R squared and the MAPE, future research should study the impact of different selection model methods in the forecast performance. In addition an analysis of the impact of the different interpolation methods carried on Section 2.2 over the forecasting performance is recommended. The models used in this thesis were multiple-output implementations from the python library scikit-learn; consequently future research can focus on the use of other multi-output models. Finally, including the DirRec strategy in the comparison can lead to new results.
References:


EWEC, 6 p., 2008.

(https://github.com/GonxoMR/ML-energy-use)

How to choose the number of hidden layers and nodes in a feedforward neural network? Retrieved June 13, 2017, from


