Lappeenranta-Lahti University of Technology LUT
School of Engineering Science
Computational Engineering and Technical Physics
Computer Vision and Pattern Recognition

Mickaël Denni

DEEP FORECASTING OF ELECTRICITY CONSUMPTION AND PRODUCTION

Master's Thesis

Examiners: Professor Lasse Lensu
Professor Galina Malykhina

Supervisors: D.Sc. (Tech.) Toni Kuronen
Associate Professor Arto Kaarna
Professor Lasse Lensu

ABSTRACT

Lappeenranta-Lahti University of Technology LUT
School of Engineering Science
Computational Engineering and Technical Physics
Computer Vision and Pattern Recognition

Mickaël Denni

Deep forecasting of electricity consumption and production

Master's Thesis

2020

52 pages, 33 figures, 7 tables.

Examiners: Professor Lasse Lensu
Professor Galina Malykhina

Keywords: deep forecasting, time series analysis, electricity production, electricity consumption, recurrent neural network, long short-term memory

Buildings stock represented 41% of the final European energy consumption in 2016. It is, therefore, critical to improve their energy efficiency. Forecasting the electricity consumption of buildings would help users in saving energy, as it can support energy efficiency and reveal building system faults. The focus of this thesis is to apply machine learning for forecasting the electricity consumption and production. The data includes electricity consumption, solar power generation and weather times series that were previously collected hourly in the campus of Lappeenranta-Lahti University of Technology. It is analysed over a 36-hour prediction horizon with a state-of-the-art model, called the Interpretable Multi-Variable Long Short-Term Memory Neural Network, that was selected among the best forecasting models. An accurate forecast with a MAAPE of 2.45% is achieved for the electricity consumption data. Forecasting the solar power generation data was less successful, reaching a MAAPE of 55.86% at best.

PREFACE

First of all, I would like to thank my supervisors, D.Sc. (Tech.) Toni Kuronen, Associate Professor Arto Kaarna and Professor Lasse Lensu for the continuous support and professional help they provided me to move forward.

Thanks to DIGI-USER, a LUT research platform, that proposes the study of the thesis that is a work included in their activities. Thanks to our collaborators in LUT LES for helping with the data of the thesis. Thanks to all the staff of the Computer Vision and Pattern Recognition Laboratory for assisting me for technical aspects. A more global thanks to LUT University for affording such an amazing environment for studying.

Finally, thanks to my parents for their encouragement and the chance they gave me to study abroad. Thanks to all my friends in Lappeenranta with whom I had great time all year long.

Lappeenranta, May 25, 2020

Mickaël Denni

CONTENTS

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
ANN	Artificial Neural Network
BNN	Bayesian Neural Network
CART	Classification and Regression Tree
CEC	Constant Error Carousel
CNN	Convolutional Neural Network
DA-RNN	Dual-Stage Attention-Based Recurrent Neural Network
DNN	Deep Neural Network
FMI	Finnish Meteorological Institute
GP	Gaussian Processes
GRNN	Generalized Regression Neural Network
GRU	Gated Recurrent Unit
IMV-LSTM	Interpretable Multi-Variable Long Short-Term Memory
KNN	K-Nearest Neighbors
LSTM	Long Short-Term Memory
LUT University	Lappeenranta-Lahti University of Technology
MAAPE	Mean Arctangent Absolute Percentage Error
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
NN	Neural Network
RBFNN	Radial Basis Function Neural Network
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SVM	Support Vector Machine

1 INTRODUCTION

1.1 Background

According to Odyssee Mure, a project supported under the Intelligent Energy Europe Programme of the European Commission, buildings accounted for 41% of the European final energy consumption in 2016 [1]. Finland had the highest average annual consumption per m^2 in Europe with around 265 $\mathrm{kW\,h/m^2}$. Therefore, building energy use is a key element to work on in order to reduce the reliance on global energy.

The forecasting of buildings energy balance plays an important role in saving energy, as it can support the evaluation of energy efficiency and reveal building system faults. Since the early 1990s, several tools have been developed to forecast buildings energy use. They can be divided into two main types of methods [2]. On the one hand, there are the thermodynamics-based models that aim to model the thermal behavior of the concerned buildings by using thermodynamic equations. On the other hand, there are the models based on Artificial Intelligence (AI) that statistically predict energy use. This is called deep forecasting, that means using deep learning for forecasting.

With regard to the use of thermodynamics-based models, a lot of different simulation tools exist to model the thermal behavior of buildings [3]. They are really suitable for energy systems commissioning because they do not need huge amount of data. However, in order to use them, all the physical features of the buildings have to be known, such as dimensions, wall materials, heating system, air conditioning system or behaviour of occupants for instance. Getting all the information for large and complex buildings and designing a model that reflects reality is a enormous task that needs a lot of time and money.

AI-based models require less detailed physical information. There is no need to know the different relationships that could exist between the exogenous variables, i.e., the independent variables that affect a model without being affected by it, such as the outdoor temperature or the sunshine [4]. It is true that extensive data are necessary to train AI-based models in order to get reliable results. However, they are relatively easy to establish and their approach has gained a lot of popularity in recent years because they provide promising prediction accuracy once the model is well trained.

In this thesis, the input variables of a model are time series. It means they are sequences

taken at successive equally spaced time steps. A global approach to the thesis is exposed in Figure 1 to clarify what is the data and what is the main goal. The data for the thesis have been previously collected in the campus of Lappeenranta-Lahti University of Technology (LUT University). They are time series of the electricity consumption, photovoltaic power generation and weather. Radiation data collected from the meteorological station of Kotka is conjointly used for forecasting the solar power generation. In LUT University, buildings are additionally using district heating. This system consumes energy as it distributes the heat through a system of insulated pipes. However, it does not require any electricity, that is why it is not taken into consideration in the data.

No physical information about the concerned building is considered in the data. That is why this thesis will focus on AI-based methods. The forecasting of the electricity use will be assessed for a 36-hour horizon in order to match with the maximum number of hours in advance power producers and consumers can submit their bids to the European market, declaring the supplied or demanded electricity quantities and the corresponding price [5].

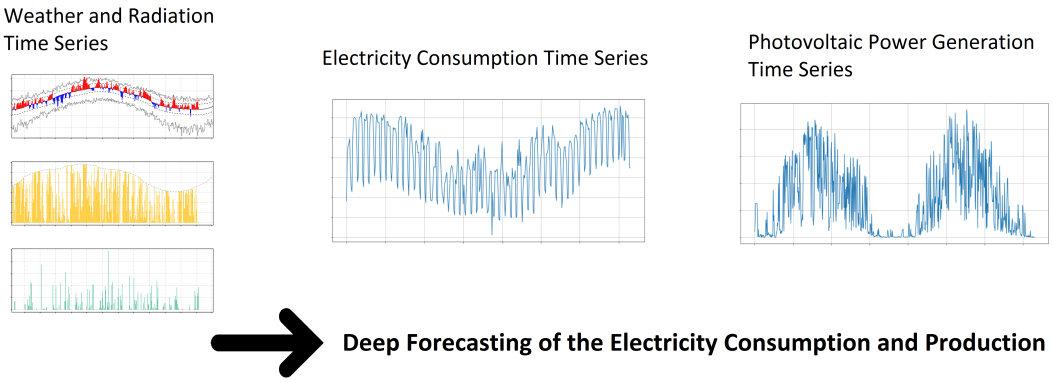


Figure 1. The collected data and the forecasting goal of the thesis.

1.2 Objectives and delimitations

The previously collected data about the campus of LUT University is used to predict the energy production and consumption of the campus using AI-based methods. The aim is to perform a quantitative evaluation of the forecasting performance as well as investigate the relevance of input variables. The objectives are as follows:

- Study and develop a deep learning model to forecast electricity production and consumption for a horizon of 36 hours.

- Study and develop ways to investigate the relevance of the input variables for the models.
- Analyze the effect of the forecasting horizon for the models.

The models tested in this thesis is optimized to get the best results with a specific dataset. They can be tuned for another building with different data, but they should not be taken as an incontestable reference better than other models. In fact, AI-based models do not rely only on a physical interpretation. As a consequence, they need to be adapted and re-trained once changes are made to the building system.

1.3 Structure of the thesis

This Master's Thesis consists of six chapters. Chapter 2 is an overview of common approaches which can be applied to forecasting time-series data. Chapter 3 describes the model that has been chosen. Chapter 4 is related to the experiments and the results. Chapter 5 handles the discussions about research which has been made. Chapter 6 is the conclusion.

2 FORECASTING OF TIME-SERIES DATA

According to the M3 time series forecasting competition that happened in 2008, the best AI-based methods for time series can be divided into the regression-based methods and the Artificial Neural Networks (ANNs), know more simply as Neural Networks (NNs) [6]. ANNs are the most popular model and they have a multitude of variants, which are most of the time combined. That is why ANNs are divided into two categories: the traditional NNs that consist of only three layers of neurons and the Deep Neural Networks (DNNs) that have more layers.

2.1 Regression-based methods

Regression-based methods are a set of statistical processes for estimating the relationships between a dependent variable (the desired output) and one or more independent variables (the inputs). Therefore, they are widely used for forecasting as it is easy to approximate a predicted value when the relationship function has been optimized [2].

The K-Nearest Neighbors (KNN) is a simple regression method that is based on the target outputs of the K nearest neighbors of the given query point [7]. To determine how to evaluate the distance between the points, the user has to set up a reference distance, such as the Euclidean distance which is reliable for low-dimensional problems. For each query point, the outputs of its K nearest neighbors are calculated. The output of the query point is then approximated by the output that is mostly represented upon its neighbors. The parameter K is set by the user. It has to be large enough to give a smooth fit but not too large to avoid a high bias. Variants of the KNN can lead to better results in certain cases. For instance, in case of disparity between the distances, the distance weighted KNN is more appropriate because it gives more weight to the points which are nearby to the query point and less weight to the points which are farther away. In turn, the cosine distance-based KNN is more appropriate to consider for high-dimensional problems because it gives a more balanced significance to the different dimensions by reducing the impact of the magnitude of the vectors.

Classification and Regression Trees (CARTs) are based on a hierarchical tree-like partition of the input space [6]. The tree consists of internal decision nodes depending on the values of specific variables and terminal leaves. The path of a test sample is determined through a series of tests on each node, starting from the root node till reaching a leaf. The

approximation output of the query point corresponds to the output associated with the final leaf. That is why CARTs are limited when the individuals present a similar performance because it is harder to divide samples into branches. But in case of good results, the CART is an excellent method to detect the most relevant variables of the problem because they are involved in the first nodes of the tree near the root node.

A Support Vector Machine (SVM) is a model where the individuals are represented as points in space that are mapped in order that the categories of individuals can be divided by a gap [2]. The individuals are then predicted to belong to a category based on the side of the gap on which they fall. The originality of SVM is that the more the gap is wide, the more it authorizes an error for classifying the individuals. The output is then defined in a range and not limited to only one value, such as in the KNN and the CART. The drawbacks of the SVM are the determination of a measure of the similarity between points in the feature space (called the kernel function) and the difficulty of adjusting parameters which are not physically interpretive. The advantage of the SVM is that it is based on a minimization problem that leads to a reasonable amount of training.

Gaussian Processes (GP) is a method of interpolation for which the individuals are modeled by a Gaussian process i.e. a distribution over functions with a continuous domain [8]. It is a nonparametric method which means it will consider every possible function that matches the data, from the simple linear regression to regressions with way more parameters. The goal is to find the most consistent functions that describes the data. The prediction equation is obtained by standard handling using Bayes rules.

The main draw-back of regression-based methods is that they rely on a predefined nonlinear form hypothesis, such as the distance between points in the KNN or the kernel function in the SVM, so they may not be able to capture the true underlying nonlinear relationship appropriately. ANNs are more adapted to understand nonlinear data.

2.2 Traditional Neural Networks

An ANN, sometimes called a multi-layer perceptron, is a computational approach inspired by the way animal brains work [6]. An ANN consist of artificial neurons placed in layers. The neurons of the layers located next to each other are connected. Each connection is weighted by a value from 0 to 1 which controls how the signal is transmitted from one neuron to another. ANNs have huge training faculties and there is no starting hypothesis to impose compared with the regression-based methods. The main disadvantage of an ANN

is the "black box" between the input layer and the output layer that prevents physical interpretation. A traditional NN is an ANN that has only three layers in total: the input layer, one hidden layer and the output layer. Traditional NNs are the simpler version of ANNs. They are easier to parametrize and are appropriate for not too large datasets. The network output y is defined as

$$y = \nu_0 + \sum_{j=1}^{n} \nu_j g(\omega_j^T x') \tag{1}$$

where n is the number of neurons on the output layer, x' is the input vector x augmented with 1, ω_j is the weight vector for the j^{th} hidden neuron and ν_j is the weight for the j^{th} output neuron. Function g is called the activation function: it is a squashing function that keeps the weights between -1 and 1 (or sometimes between 0 and 1).

A Bayesian Neural Network (BNN) is a traditional NN based on a Bayesian probabilistic formulation [6]. The idea of the BNN is to impose to the network weights to follow an a priori distribution. This distribution is appropriate for reaching smoother results for low complexity models. The predictions takes into account both the a priori distributions and the accuracy of the observed data by redefining the objective function J as

$$J = \alpha E_D + (1 - \alpha) E_W \tag{2}$$

where E_D is the sum of the square errors in the outputs and E_W is the sum of the squares of the network parameters i.e. the n elements of the weight vector previously named as ω_j in the Equation 1. α is a regularization coefficient, such as $0 \leq \alpha \leq 1$, that controls how impactful the a priori distributions should be.

A Radial Basis Function Neural Network (RBFNN) is an advanced traditional NN where the activation function is the Gaussian function, which is a localized function [6]. That means it is only defined by a center position and a width parameter. RBFNN works with an algorithm that determines the appropriated number of neurons so as the user does not have to choose it. The algorithm adds neurons to the hidden layer until it meets a specific mean squared error goal. It is appropriate to consider RBFNN when the adequate architecture of the network is not known.

A Generalized Regression Neural Network (GRNN) is a variant of a RBFNN [6]. It uses the nonparametric regression into the neural networks. Every trainig sample describes one specific neuron. The prediction y for a given sample x is obtained by calculating the mean of the other samples that are in the neighbourhood of x. So the output y of one

specific x is defined as

$$y(x) = \frac{\sum_{j=1}^{n} y(x_j) K(x, x_j)}{\sum_{j=1}^{n} K(x, x_j)} \tag{3}$$

where n is the number of neurons on the current layer, $y(x_j)$ is the target output of the input x_j and K is the Gaussian kernel function.

2.3 Deep Neural Networks

The main problem with the regression-based methods and traditional NNs is that they can only handle static data features, i.e., the input features of predefined dimensionality [9]. In this way, reliable information to extract from time series are limited. A relationship between the inputs and the output is found but it does not vary through time. Any temporal feature is taken into account to describe the output. To overcome this problem, DNNs have been made to automatically detect deeper features that could be dynamic, and to sometimes map them into a more separable space. DNNs reduce in this way the need for preprocessing. Moreover, time series are extensive data from which it is extremely hard to select features. Even though their structure can be complex to establish, DNNs have better training faculty than traditional NNs. However, they might not be required for relatively simple forecasting.

Convolutional Neural Networks (CNNs) are DNNs that have achieved tremendous success in image recognition and speech recognition in the past few years [10], but they can be applied for time series [11]. The architecture of a CNN is represented in Figure 2. A CNN usually consists of two parts. First, convolution and pooling layers are alternatively used to extract deep features. The objective of the convolution operation is to extract the high-level features like gradient orientation in images for instance. These feature maps are divided into equal-length segments. After each convolutional layer, there is a pooling layer in order to represent every segment by its average or maximum value. Reducing the size of the output bands helps to decrease the variability of the hidden activations. Intending to classify, the features are then connected to fully-connected layers, which are in principle similar to a standard ANN.

Recurrent Neural Networks (RNNs) turn out to be a variant of DNNs really flexible in detecting temporal dynamics [12]. A RNN is an ANN where neurons are connected so as to create a directed graph. Figure 3 shows a basic RNN architecture with a delay line and unfolded in time for two time steps. A RNN can be seen as one standard network repeated several times. That is why, even if its architecture with a delay line looks like a

traditional NN, a RNN is a DNN as it can be unfolded. There is one drawback however: standard RNNs suffer from the problem of vanishing gradient which means that they cannot capture long-term dependencies [13].

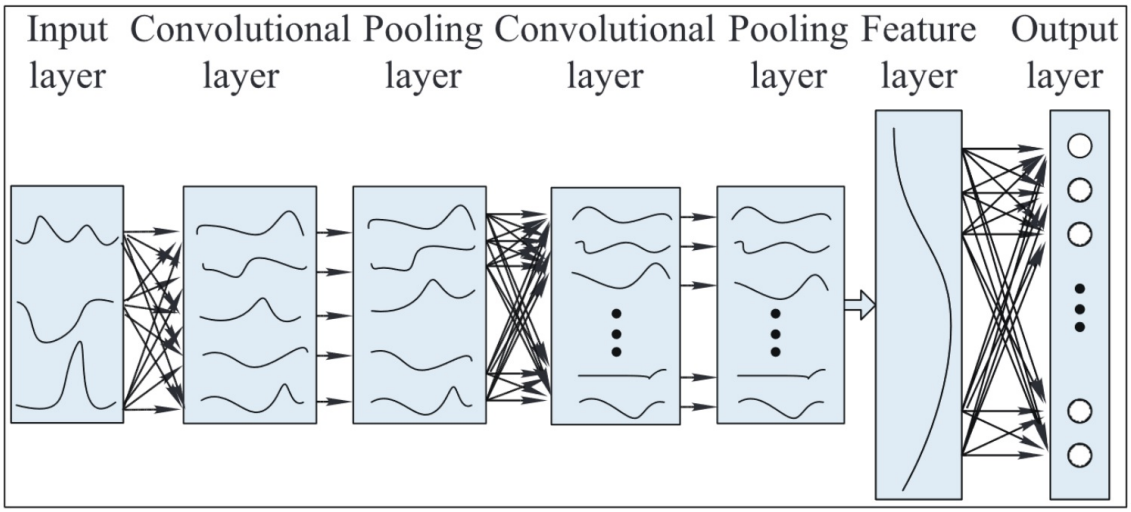


Figure 2. The architecture of a convolutional neural network for three-variate time series classification [11].

A Long Short-Term Memory (LSTM) is a RNN architecture that is specially developed to avoid the long-term dependency problem [14]. It aims at keeping information for long periods of time. A LSTM block is represented in Figure 4. In addition to the input and the output, a LSTM unit has a cell unit and three separated gates to avoid input weight conflicts. The input, forget and output gates control respectively how the input, the already saved information and the output are considered. The gates use a sigmoid activation function, and the input and cell state is usually transformed by the hyperbolic tangent, another activation function. The gating mechanism can hold information for long durations, however a few LSTMs do not have a forget gate and instead add an unchanged cell state (e.g. a recurrent connection with a constant weight of 1). This addition is called the Constant Error Carousel (CEC) because it solves the training problem of vanishing and exploding gradients. In networks that contain a forget gate, the CEC may be reset by the forget gate. The addition of the CEC allows for the LSTM to learn long-term relationships while mitigating the risks of prolonged testing. Exploring deeper the structure of LSTM can help in the interpretation of the exogenous variables and the capture of their dynamics [15].

The LSTM RNNs are often used in an Encoder–Decoder mode [17]. An encoder transcribes the input sequence as a vector which is then decoded, i.e., transformed in an output sequence. By doing this, the length of the output sequence can be different from

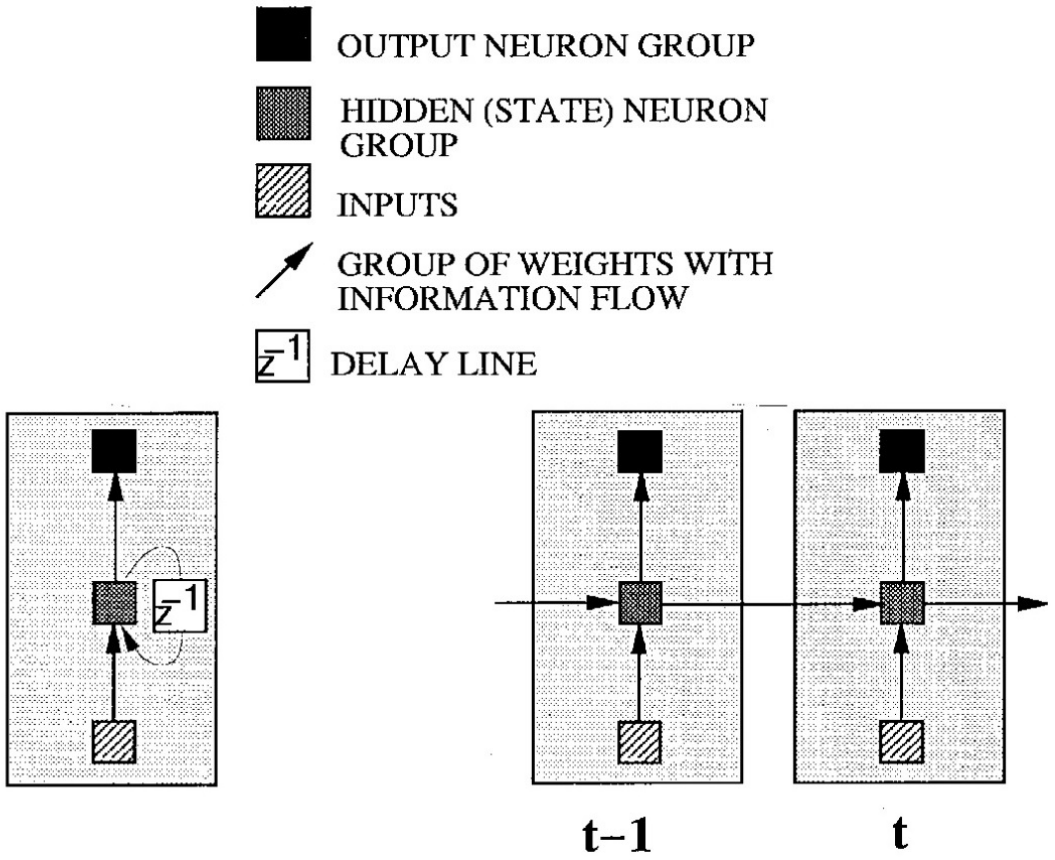


Figure 3. The structure of a standard recurrent neural network shown with a delay line on the left and unfolded in time for two time steps on the right [12].

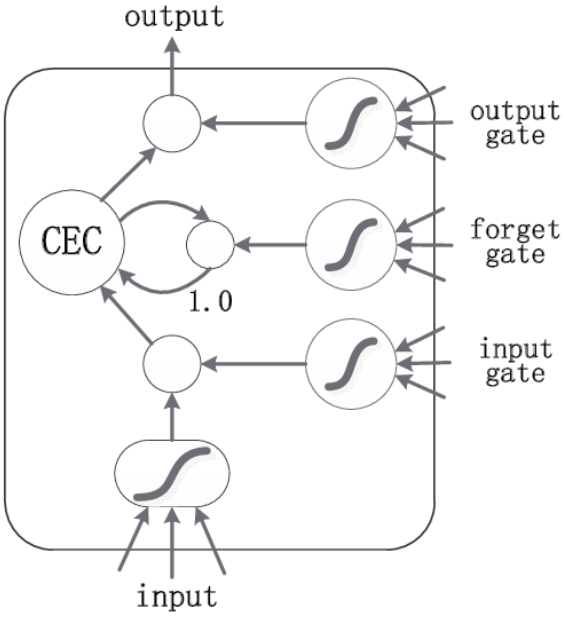


Figure 4. The architecture of a LSTM block with a constant error carousel (CEC) [16].

the length of the input sequence. This system is widely used in machine translation to avoid word-by-word translation. Encoder-decoders can in the same way be used with

Gated Recurrent Units (GRUs), which are a lighter version of LSTMs, as it lacks an output gate [17]. GRUs have been shown to exhibit even better performance than LSTMs on certain smaller datasets. Although, they have to be used carefully as they are less adapted than LSTMs for specific situations. For instance, they cannot perform unbounded counting, while LSTMs can. In addition, encoder-decoders can deteriorate rapidly as the length of the input sequence increases. So, carrying out a time series problem with an encoder-decoder has to be done with caution.

What makes the different types of ANNs attractive is that they can be combined. In 2017, researchers in the University of California developed a Dual-Stage Attention-Based Recurrent Neural Network (DA-RNN) with integrated LSTMs and GRUs for time series prediction [18]. The particularity of the DA-RNN is the input attention implemented in the encoder and the temporal attention implemented in the decoder. Attention mechanisms in ANNs are made to impact the way inputs are treated. In the case of the DA-RNN, attention weights have been inserted inside the input vector so the encoder can selectively focus on certain driving series rather than treating all of them equally. In the same way, attention weights have been added to encoder hidden states so the decoder can adaptively select the relevant ones across the time steps.

In 2019, a similar network, called the Interpretable Multi-Variable Long Short-Term Memory (IMV-LSTM), was developed. Inspired by the DA-RNN, the IMV-LSTM focuses on controlling the importance of the variables and their time steps to improve the forecasting performance [15]. The main idea of the IMV-LSTM is to handle a hidden state matrix permanently updated so as to extract information from every input time series and save it in the different elements of the matrix. In this way, the contribution of each time series can be distinguished. Once they are associated with their relative time series, the extracted features are then used as inputs for attention layers similar to the DA-RNN to evaluate the relevance of the time series.

A CNN can be combined with LSTMs [19]. It was shown in 2005 that this advanced architecture provides a 4 to 6% relative improvement in the word error rate in speech recognition over a standard LSTM RNN. Such improvement seems to be possible for forecasting time series.

2.4 Applications for forecasting the electricity use

In 2017, M. Dahl et al. developed a regression-based model for weather prediction [20]. The chosen model was a simple autoregressive forecast model. It was applied to the operation of three heat exchanger stations in order to optimize the control of the temperature. Such a control can avoid unneeded supply of heat. This model has been chosen because it was "simple and intuitive" and only "decent forecast performance" was needed. The model's performance was evaluated through a 10-fold cross-validation period. Tested with 25 weather forecasts, they achieved a Mean Absolute Percentage Error (MAPE) of 5.7%, which was enough for them to select the best heat exchanger station.

Although, the LSTM RNN is the most popular model for building electricity use forecasting by the fact that it is involved in a lot of recent articles about this subject [21–23]. In 2017, it was compared with its rival deep learning methods in short-term (in one hour time-step) residential load forecasting [21]. The data was more precisely a public set of real residential smart meter data. The following average MAPEs correspond to 2 time steps (2 hour-ahead) prediction, but the LSTM RNN was proven to be better for 6 time steps and 12 time steps prediction. LSTM RNNs had 8.18% of average MAPE, while MAPE minimization and empirical mean which are regression-based methods had respectively 34.91% and 32.54%, standard back-propagation NNs had 8.37%, KNNs had 15.37%, extreme learning machines had 33.68% and input selection scheme combined with a hybrid forecasting framework had 20.43%.

The DA-RNN and IMV-LSTM, developed respectively in 2017 and 2019, were trained on several reference data and compared to each other. One of the training was about forecasting the energy production of a photo-voltaic power plant in Italy with 9 additional input weather time series. The IMV-LSTM outperformed the DA-RNN and the other forecasting models for all the different data sets notably because of an advanced and relevant adjustments of the exogenous variables.

In 2020, ForecastNet was developed, trying to join forces of CNNs and LSTMs [24]. It uses a deep feed-forward architecture for multi-step ahead forecasting. It was tested for 10 various databases and compared to regression-based models and standard LSTM RNNs. It turned out to outperform the other models for 6 of the 10 databases. In particular, it performed best for 3 out of 4 databases about electricity consumption or production. Unfortunately, no comparison with DA-RNN or IMV-LSTM was done.

3 DEEP FORECASTING OF THE ELECTRICITY PRODUCTION AND CONSUMPTION

This section is based on the literature review and describes more specifically the model that is implemented in this thesis. As presented in 2.3 and 2.4, LSTM RNNs appear to be really appropriate for forecasting the energy consumption and production. It is, therefore, natural to focus on state-of-the-art LSTM RNNs. The model that has been chosen is the IMV-LSTM, firstly introduced in [15]. It turned out to outperform the other LSTM RNNs.

3.1 Structure of the model

What makes the IMV-LSTM more robust than other LSTM RNNs is that it implements novel concept to bring to the fore the most useful part of the data. The main idea of the IMV-LSTM is to handle a hidden state matrix and establish associated update scheme so as to extract information from every input time series and save it in the different elements of the matrix. What is called hidden states is more precisely the inputs and outputs that come in and out the LSTM units. The IMV-LSTM implements its attention mechanisms within the LSTM units to directly infer on the features extracted from them.

A simplified structure of the IMV-LSTM is represented in Figure 5. The input matrix at time step t is defined as $\mathbf{X}_t = \{x_t^1, x_t^2\}$. The hidden state matrix at time step t is defined as $\widetilde{\mathbf{h}}_t = [\mathbf{h}_t^1, \ldots, \mathbf{h}_t^N]$. The hidden update matrix at time step t is defined as $\widetilde{\mathbf{j}}_t = [\mathbf{j}_t^1, \mathbf{j}_t^2]$. Then, the input-to-hidden and hidden-to-hidden transitions are respectively denoted by $\mathbf{U}_j$ and $\mathbf{W}_j$ [15].

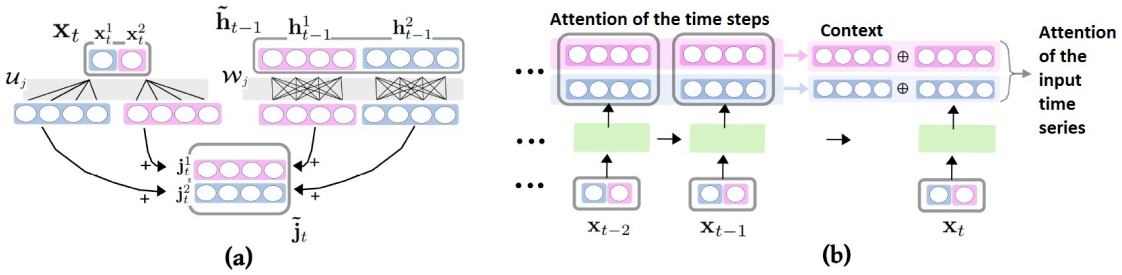


Figure 5. Simplified structure of the IMV-LSTM for a input data of two variables represented as purple and blue colors and a hidden state matrix of 4 dimensions per variable represented as circles. Illustration (a) describes how the hidden updates $\widetilde{\mathbf{j}}_t$ are carried out. Illustration (b) describes the implementation of attention mechanisms (modified from [15]).

3.2 Hyper parameters of the model

As its name indicates, the input of the IMV-LSTM is multivariate. So, several exogenous variables can be added to the inputs to improve the forecast. The output is though a single target time series. Two hyper parameters, inherent to time series forecasting, characterize how the data has been arranged:

- The window size is the number of backward time steps that are considered for forecasting.
- The prediction horizon is the number of forward time steps that are forecasted.

Let us understand more precisely how the input samples are formed from the data. The data is considered to be a T by N matrix, where T is the number of time steps and N is the number of time series corresponding to the exogenous variables. The window size and the prediction horizon are respectively set to strictly positive values w and h.
For each time step t such as $w - 1 < t < T - h$, a sample s_t is built. The sample construction is illustrated in Figure 6. The sample s_t is made up of an input matrix and an output vector. The input matrix contains the w backward time steps to time step t for every variable. The output vector consists of the h forward time steps to time step t. This is the true value of the target time series associated with the sample. Once all the samples linked to the time steps are built, the division into training, validation and testing sets can proceed.

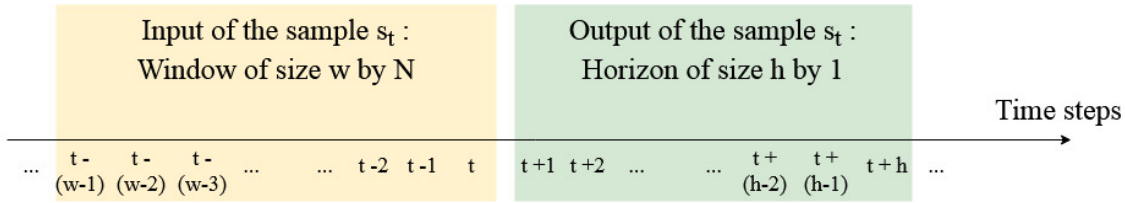


Figure 6. Illustration of the construction of the samples from the data.

As it is mentioned in 1.1, the ideal prediction horizon value for the thesis is 36 hours in order to match with the bids of the European electricity market [5]. That is why, the experiments will focus on forecasting 36-hour horizons. Concerning the window size, the most suitable value cannot be determined theoretically. Several tests will be performed to determine it.

The IMV-LSTM has additional hyper parameters that are common to most of the deep learning models: the learning rate, the validation patience and the maximal number of epochs. The learning rate controls how the network learns at every iteration. The patience defines the maximal number of epochs that can be done without any improvement in the training. The maximal number of epochs controls the iterations that can be done until the training stopped. The learning rate and the validation patience have to be chosen side by side so that the learning of the training is simultaneously smooth and efficient. Setting the maximal number of epochs high enough is better in order to make the network stopped because of the validation patience. In this way, the network learns as much as it can.

3.3 Attention mechanisms

The main strength of the IMV-LSTM is the attention mechanisms implemented in it to control the variables and their time steps. An attention mechanism aims at giving unequal weights to the variables after evaluating their utility in the forecasting goal. Considering the attention mechanism weights after training the network can determine which data seem to be more relevant for the network. But, the main reason why the network alters the contribution of the input time series is to increase the performance of the forecasting. In the IMV-LSTM, two attention mechanisms have been added in order to work with the LSTM units, corresponding to two ways of considering the relevance of the data.

The first attention mechanism evaluates the relevance of the N input time series. A lot of different input time series are taken into consideration but they may be more or less useful for the forecast. It is crucial for the network to know which exogenous variables are more relevant. An example of a potential conclusion from this attention mechanism could be: The outdoor temperature is the variable that the network takes more into account to build the forecast.

The relevance of the h time steps of the window can as well be manipulated for every variable. The data set in the thesis is quite large. Training the network with a high window size may be adequate to extract the most useful information. However, all the time steps may not be considered equally: the second attention mechanism should in this way nuance how they are weighted. An example of a potential conclusion could be: The data that are measured 3 days ago from the forecasting period are less taken into consideration by the network.

4 EXPERIMENTS AND RESULTS

4.1 Data description

The data consist of time series with a time step of one hour. The target time series are the electricity consumption and the solar power generation. There are 3 different categories of input time series:

- The weather data that consist of 11 sub-time series.
- The radiation data that consist of 3 sub-time series.
- The calendar data that consist of 6 sub-time series.

They have been collected from different sources as described below.

4.1.1 Electricity consumption and production data

Electricity consumption data and solar power generation data of the campus of LUT University have been collected from databases provided on Grafana which is an online interface that facilitates data visualization [25]. The electricity consumption data is not divided into sub-data: The provided time series is presented in Figure 7 and corresponds to the whole electricity consumption of the campus. Every time step value corresponds to the electricity consumption for the hour in kW. Numerous parts at the beginning of the data are not in harmony with the rest of the data. This may come from measurement problems. This issue will be investigated in the Experiment 1.

The electricity production data consist of solar power generation time series of five groups of panels implanted in the campus of LUT University. Their names are South Wall, West Wall, Carport, Flatroof and Single Panel. Two other groups of panels, named Fixed Installation and Tracker, were available on Grafana. However, their corresponding datasets ended in July 2018: It seems that they are not working anymore so they are not considered in this study.

The solar power generation data sets are presented in Figures 8 to 12. Every time step value corresponds to the solar power in kW generated in the hour. The last group Single

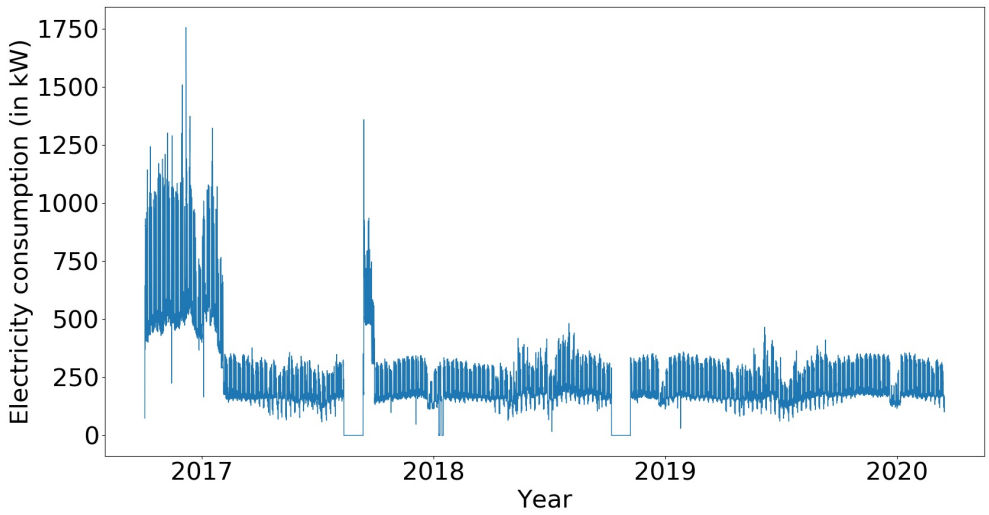


Figure 7. Electricity consumption data set.

Panel only corresponds to the electricity production of a single panel. That is why, its electricity production is lower than the other groups. All the solar power generation data sets have large periods with only zero values. These periods correspond to the dark period of winter when almost no solar power generation is feasible.

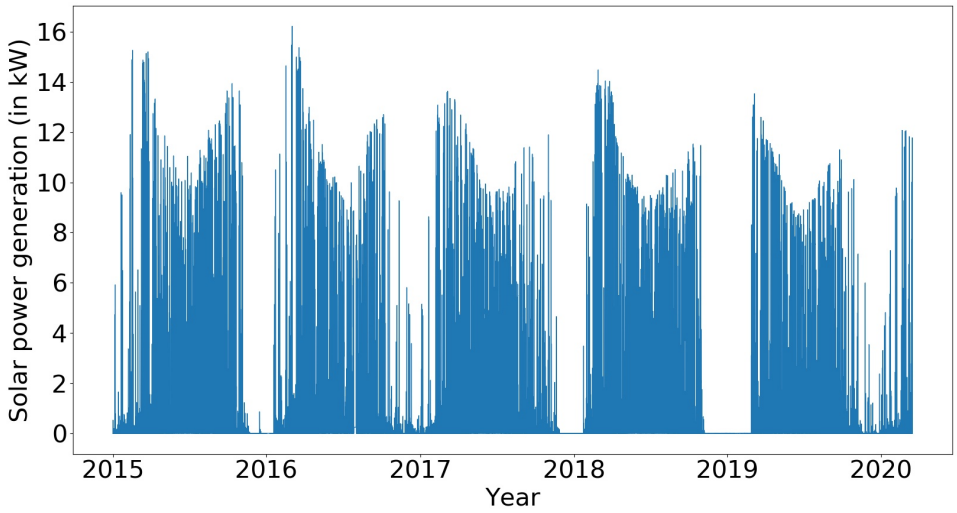


Figure 8. Solar power generation data set of South Wall.

The groups of panels are not oriented the same way and do not consist of the same number of panels. They may not be correlated the same way to the input time series. Thereby, the different electricity generation time series are studied separately. However, the whole

solar power generation is additionally studied, in order to determine the relevance of studying the different groups of panels separately. The whole solar power generation was obtained by summing the production of the five groups of panels. It is presented in Figure 13. The electricity production of the campus of LUT University reaches maximum values of approximately 130kW.

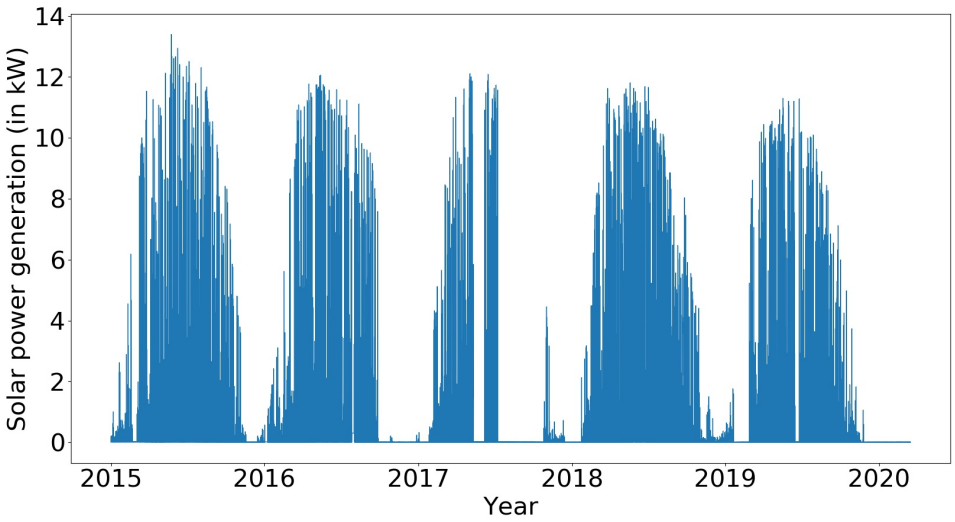


Figure 9. Solar power generation data set of West Wall.

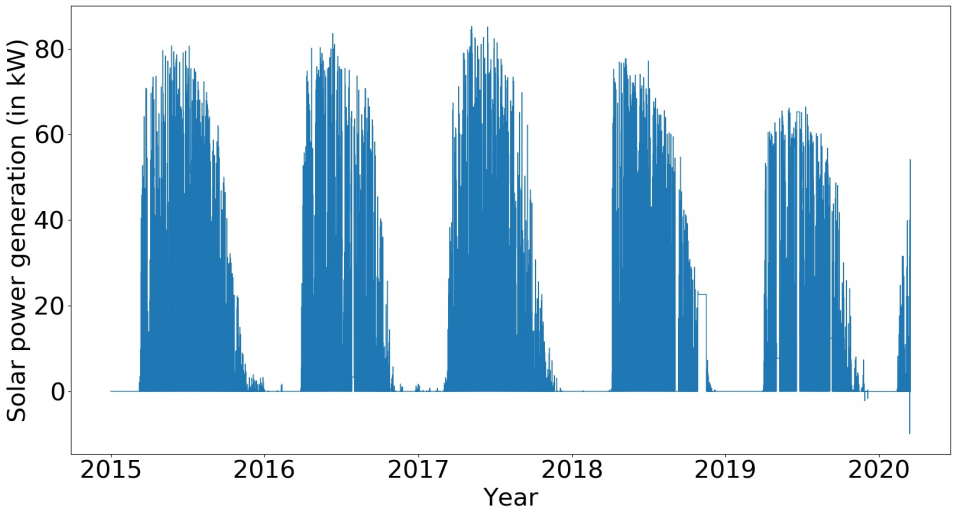


Figure 10. Solar power generation data set of Carport.

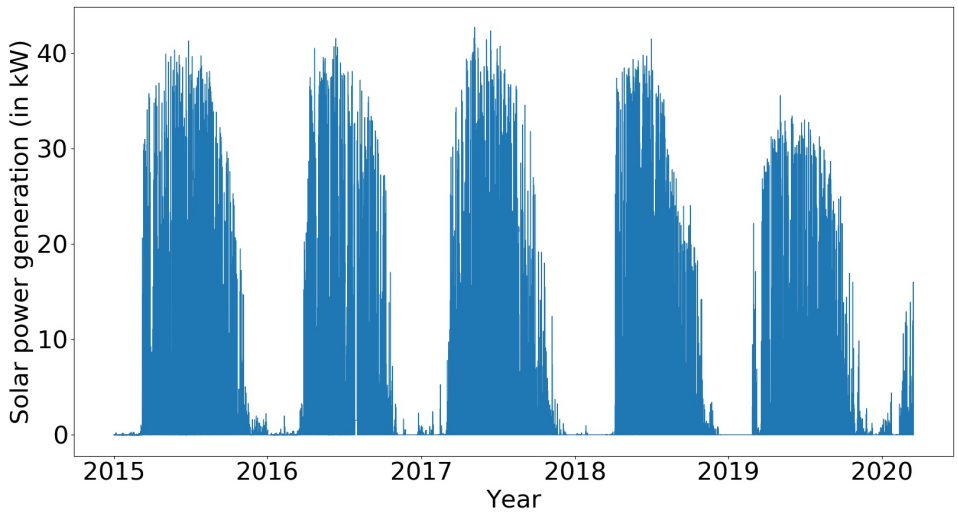


Figure 11. Solar power generation data set of Flatroof.

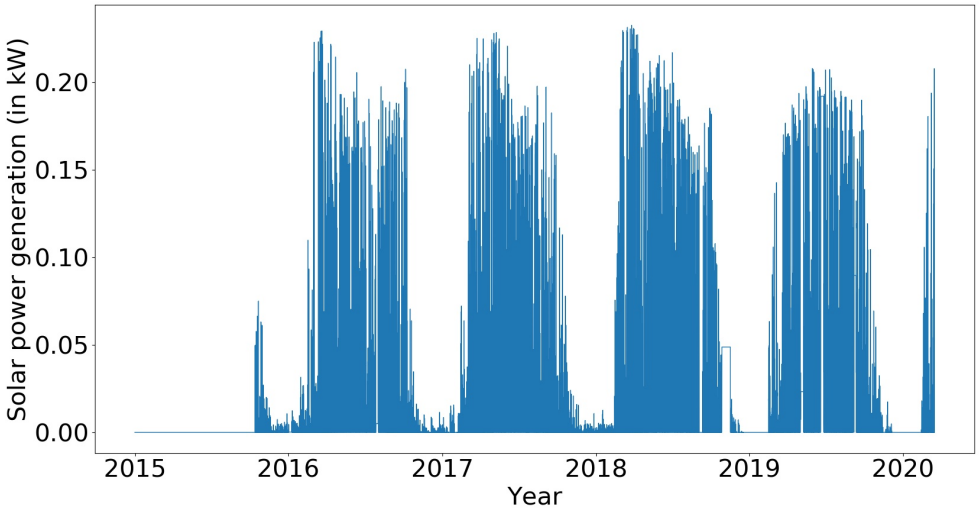


Figure 12. Solar power generation data set of Single Panel.

4.1.2 Weather data

Weather time series have been collected from the Finnish Meteorological Institute (FMI) [26]. They correspond to the weather data of the airport of Lappeenranta, which is located 6 kilometers from LUT University. The weather time series consist of 11 sub-time series which are cloud amount, pressure, relative humidity, precipitation intensity, snow depth, air temperature, dew-point temperature, horizontal visibility, wind direction, gust speed and wind speed. All of these variables are used for forecasting the electricity consumption and solar power generation.

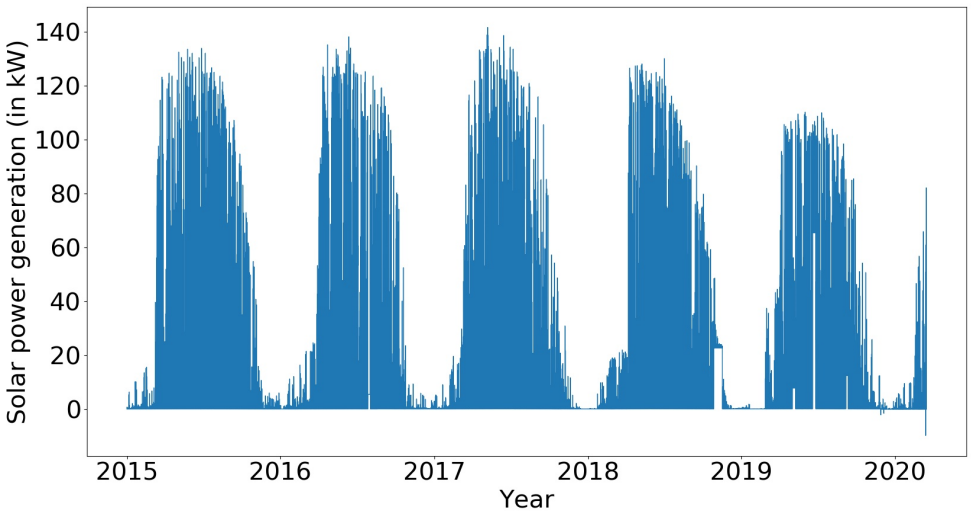


Figure 13. The whole solar power generation of LUT University.

4.1.3 Radiation data

Radiation observations are not available from the meteorological stations in Lappeenranta. However, adding radiation observations to the input time series is necessary because they should play an important role for forecasting the solar power generation. That is why, the radiation observations from Kotka station, which is 100km from Lappeenranta, are included in this study. This is the closest station from Lappeenranta that collects radiation data. As well as the weather data, the radiation data have been collected from the FMI [26]. They consist of 3 sub-time series which are diffuse radiation, global radiation and sunshine duration.

These time series are used only for forecasting the solar power generation. In principle, intense solar radiation may reduce and increase electricity consumption if electric heating or cooling is used. The radiation from the sun may conjointly warm up the buildings. However, this effect seem negligible compare to the influence of the weather data. The radiation data is, therefore, not considered for forecasting the electricity consumption.

4.1.4 Calendar data

Calendar data consist of the year, the month, the day and the hour. The initial calendar data do not represent well the cycles of time: for instance, the first hour of the day and the last hour of the day are perceived by the model to be 23 hours apart, whereas they

should be 1 hour apart. In order to convey the cyclical nature of the calendar data, they were transformed into cyclical time series using the sine and cosine functions. Such transformation enables a better use of the calendar data [27, 28]. A calendar series S is duplicated into two cyclical time series S_{sin} and S_{cos} as follows:

$$S_{\text{sin}} = \sin\left(2\pi \frac{S - S_{\text{min}}}{S_{\text{max}} - S_{\text{min}}}\right) \tag{4}$$

$$S_{\text{cos}} = \cos\left(2\pi \frac{S - S_{\text{min}}}{S_{\text{max}} - S_{\text{min}}}\right) \tag{5}$$

where S_{min} and S_{max} are the minimum and the maximum values of S.

The calendar time series S in Equation 4 and Equation 5 is one of the following:

- The hour of the day in order to represent the cycle of the days. The minimum value is 0 and the maximum value is 23.
- The day of the week in order to represent the cycle of the weeks. The minimum value is 0 and the maximum value is 6.
- The day of the year in order to represent the cycle of the years. The minimum value is 1 and the maximum value is 366.

In total, there are 6 sub-time series. The month of the year was removed from the calendar data, as it represents the cycle of the year such as the day of the year. The day of the year was preferred because it illustrates more precisely the fluctuation of the year using 366 different values instead of 12 values for the month. All of the calendar time series are used for forecasting the electricity consumption and solar power generation.

4.1.5 Splitting the data

The different target time series and the associated inputs are presented in Table 2. The weather data and the time data are taken as inputs when the electricity consumption is being forecasted. The radiation data is added to the inputs with the weather data and the time data when it concerns the forecasting of the solar power generation.

Before training it, the data is normalized. A new normalized time series S_{norm} is created

Table 2. The target time series of the thesis and their input data.

Target Time series	Weather data	Radiation data	Calendar data
Energy Consumption	Yes	No	Yes
Solar Power Generation	Yes	Yes	Yes

for every input time series S as follows:

$$S_{\text{norm}} = \frac{S}{max(|S|)} \tag{6}$$

Data is then organized into samples consisting of an input matrix and an output vector associated to every time step, as it was presented in 3.2. These samples are split into three sub-sets: the first 60% of the data is used for training, the next 20% becomes the validation set and the last 20% is for testing the model performance. The relative proportion of the training set can be considered relatively small. However, the data of the thesis spread out over a long period of time so 60% of the data is still amply enough to train the network.

4.2 Evaluation Criteria

In order to evaluate the performance of the trained model, three error metrics were chosen to measure the balance between the prediction output and the desired output. As a result of their popularity in several articles [18, 20, 21], the first two evaluation metrics are the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE).

The MAE measures the average magnitude of the errors without considering the direction. It is defined as:

$$MAE = \frac{1}{N}\sum_{t=1}^{N}|y_t - \hat{y}_t| \tag{7}$$

where N is the number of time steps of the prediction period, y_t and $\hat{y}_t$ are respectively the desired and the prediction output time series at time step t [29].

The RMSE measures the average magnitude of the errors, but gives more importance to large errors, as it squares the errors before averaging them [30]. It is defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2} \tag{8}$$

The MAE and the RMSE cannot be used to compare the results of two different sets of data because these evaluation metrics depend on the range of the data. One common evaluation metrics to get around the problem is the MAPE that is derived from the MAE [31]. The MAPE scaled to percentages is defined as:

$$MAPE = \frac{1}{N}\sum_{t=1}^{N}\left|\frac{y_t - \hat{y}_t}{y_t}\right| \times 100 \tag{9}$$

However, the MAPE is reliable only when the data does not contain zero or close-to-zero values. That is why, choosing the Mean Arctangent Absolute Percentage Error (MAAPE) for the third evaluation metric was preferred. In fact, the MAAPE has been specially developed for retaining the philosophy of the MAPE, while coping with the issue of zero and close-to-zero values [31]. The MAAPE scaled to percentages is defined as:

$$MAAPE = \frac{1}{N}\sum_{t=1}^{N}\arctan\left(\left|\frac{y_t - \hat{y}_t}{y_t}\right|\right) \times \frac{2}{\pi} \times 100. \tag{10}$$

4.3 Forecasting the electricity consumption

The Experiment 1 focuses on the forecasting of the electricity consumption. The objectives of the experiment are as follows:

- Evaluate how the window size is correlated with the prediction horizon and determine a suitable value for it.
- Examine the execution of the training runs and deduct potential improvements.
- Train the network on 36-hour prediction horizons and present the forecast results.

4.3.1 Description of the training runs

In order to evaluate the correlation between the window size and the prediction horizon, the prediction horizon of the training runs is first set to a specific value ahead instead on a full range. The aim is to observe if the most suitable window size value depends on the prediction horizon. The construction of the samples from the data that was presented in Figure 6 on page 19 is changed. The new sample construction is illustrated in Figure 14.

The output of a sample is a value at time step h instead of a vector of h elements. The value h will still be called the prediction horizon for these training runs but it has to be understood differently.

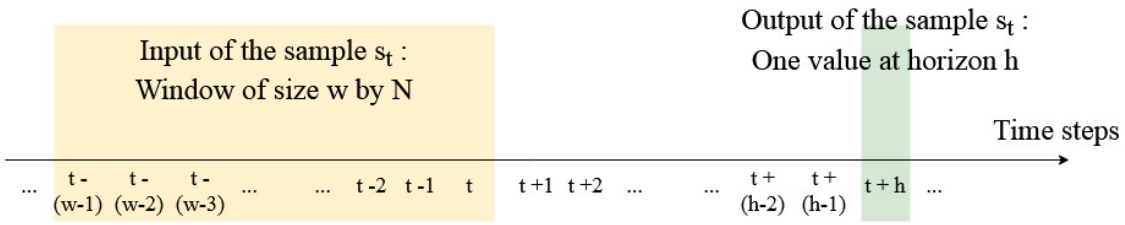


Figure 14. Illustration of the construction of the samples used to study the correlation between window size and prediction horizon.

The values of the parameters that are tested in the Experiment 1 are summarized in Table 3. Four different values are tested for the prediction horizon: 1, 12, 24 and 36 hours. As mentioned in 1.1, the horizon of 36 hours corresponds to the ideal value to match the bids of the electricity market.

Table 3. Parameter values tested in Experiment 1.

Parameter	Values tested
Prediction horizon	1, 12, 24, 36
Window size	12, 24, 36, 48, 60, 72, 84, 96, 108
Learning rate	0.001
Validation patience	35
Maximal number of epochs	1000

The range of values that is tested for the window size goes from 12 to 108 hours in 12-hour increments. Two tests are done for every combination of window size and prediction horizon values. The performance evaluation that is presented for one combination is the mean value of these tests.

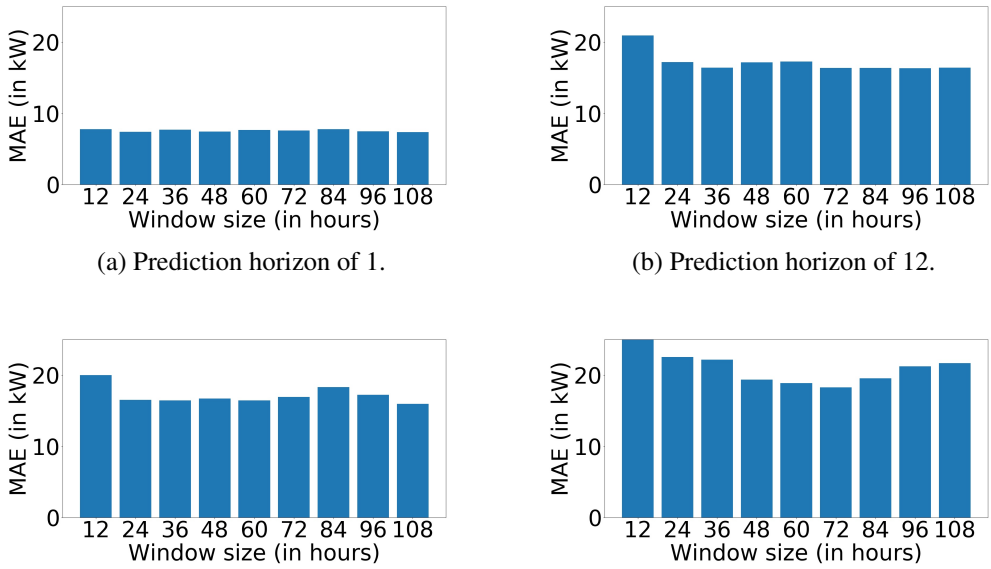
The learning rate, the validation patience and the maximal number of epochs are set to their initial values: the learning rate is 0.001, the validation patience is 35 epochs and the maximal number of epochs is 1000. If the training and the validation losses decrease smoothly and stop because of the validation patience, there is no reason to change these values.

Once the window size value is optimized, several training runs are performed for 36-hour horizons in order to evaluate the forecasting of the electricity consumption, as well as the

relevance of the input time series and their time steps. The best and worst forecasting samples are then presented in order to visualize the results.

4.3.2 Optimization of the window size

Figures 15 to 17 present the evolution of the MAE, the RMSE and the MAAPE depending on the window size for four different prediction horizon values. The error differences are more noticeable when the prediction horizon is higher. There is almost no fluctuation in the error measures for the tests done with a prediction horizon of 1 hour while there is a clear curve for the tests done with a prediction horizon of 36 hours. In addition, the gaps are more pronounced with the RMSE which may be due to the existence of large errors.



(a) Prediction horizon of 1.

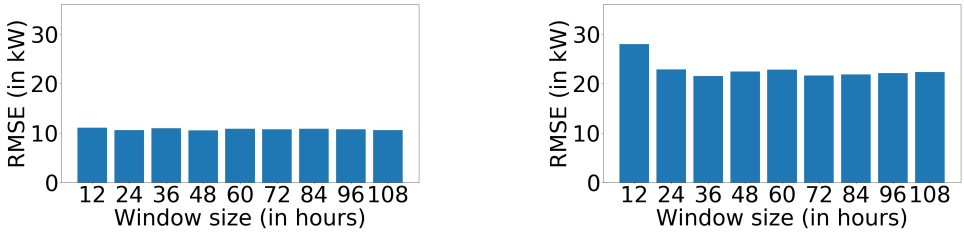
(b) Prediction horizon of 12.

(c) Prediction horizon of 24.

(d) Prediction horizon of 36.

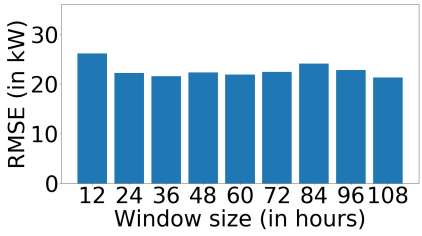
Figure 15. Evolution of the MAE depending on the window size.

The window size and the prediction horizon are correlated in the sense that the more the prediction horizon is large, the more the window size value impacts the performance. However, the most suitable window size value seems to be almost the same whatever the value of the prediction horizon is. For prediction horizon of 1, 12 and 36 hours, the lowest errors are achieved when the window size is set to 72 hours. The values with a window size of 60 hours and 108 hours are a little lower than the value with a window size of 72 hours only for the tests done with a prediction horizon of 24 hours. Therefore, the best window size value is 72 hours overall.

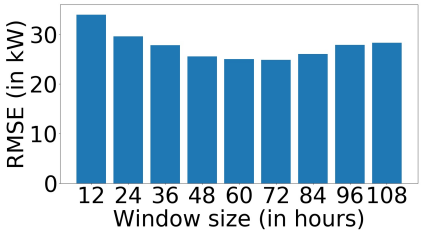


(a) Prediction horizon of 1.

(b) Prediction horizon of 12.

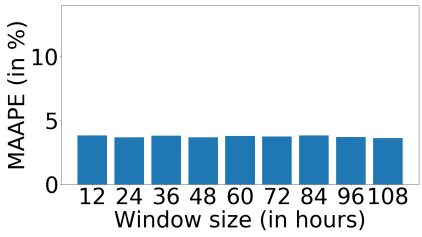


(c) Prediction horizon of 24.

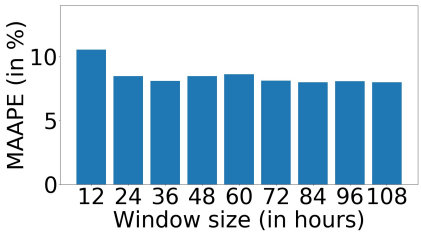


(d) Prediction horizon of 36.

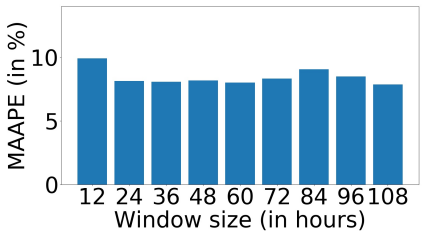
Figure 16. Evolution of the RMSE depending on the window size.



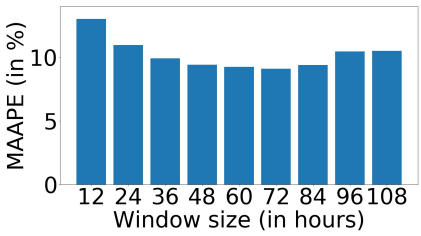
(a) Prediction horizon of 1.



(b) Prediction horizon of 12.



(c) Prediction horizon of 24.



(d) Prediction horizon of 36.

Figure 17. Evolution of the MAAPE depending on the window size.

The training and validation losses of the best test assessed for a combination of a prediction horizon of 36 hours and a window size of 72 hours is presented in Figure 18. The two curves decrease relatively smoothly and the training stopped automatically after 75 iterations. It seems that combining patience of 35 with learning rate of 0.001 turned out to be suitable to the network. However, the validation loss is under the training loss. Such thing may come from two possible problems:

- The training set has samples that are harder to detect than the validation set.
- The training, the validation and the testing sets may not be split appropriately.

If the two losses would fit each other, the model may be more reliable and efficient for various testing sets. That is why, this issue will be investigated in the second part of the experiment.

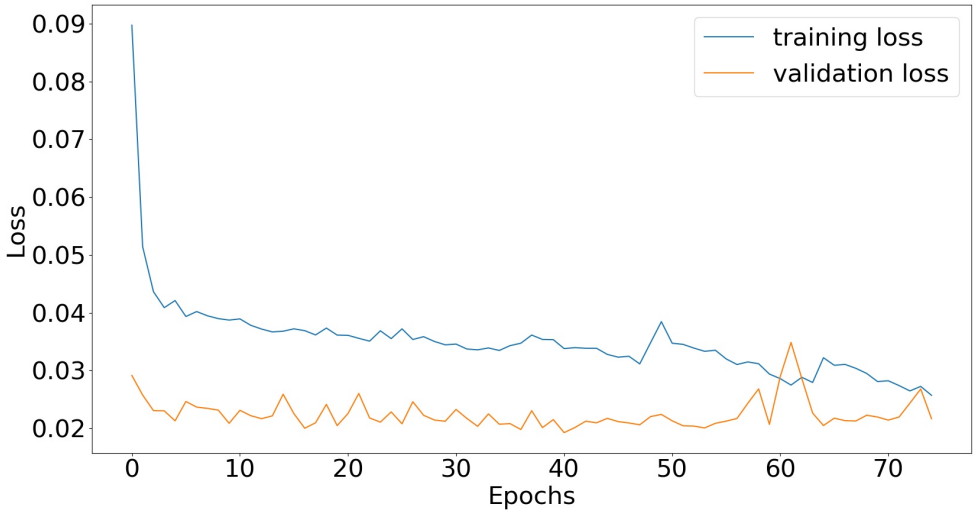


Figure 18. The training and validation losses for prediction horizon of 36 hours and window size of 72 hours.

4.3.3 Forecasting the electricity consumption for 36-hour horizons

According to the optimization of the window size, 72 hours was the most suitable value for the window size. Thereby, the window size of the next training runs is fixed at 72 hours.

Two main ideas were developed in order to improve the reliability of the forecast. The first one was to remove the invalid part of the data. Figure 7 on page 22 presents the electricity consumption data set that was used during the optimization of the window size. Numerous parts at the beginning of the data are not in harmony with the rest of the data. Such periods may potentially impair the training runs. The purpose of this study is not to detect outliers or abnormal periods but to improve the forecast potential of the

model. Thereby, the new data set that is used for the second part of the experiment do not have the invalid parts of the data anymore: the new data set is shown in Figure 19.

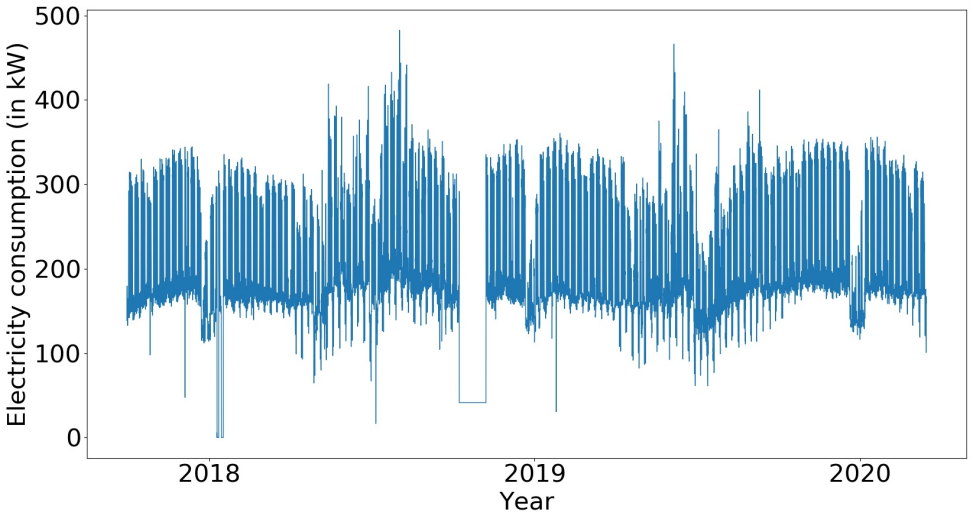


Figure 19. Reduced data set for the electricity consumption.

The second idea to improve the reliability of the results is to randomize the samples formed from the data before selecting the training, validation and testing sets. Previously, the samples of the three sets were taken chronologically. Now, they are taken randomly. The purpose of this randomization is to avoid potential inconsistencies between the three sets.

In the second part of the experiment, 10 training runs are performed over a 36-hour prediction horizon, with a window size of 72 hours. Half of the training runs is done without randomization of the data, and the other half is done with randomization. Table 4 and Table 5 represent the results of the 10 training runs. The values of the error measures have been calculated for the whole prediction horizon of 36 hours. The results without randomization achieved MAAPEs going from 4.51% to 5.48% while the results with randomization achieved MAAPEs going from 2.33% to 2.53%. Thus, the training with randomization are twice more accurate than those without randomization. Moreover, the training runs with randomization are way longer to train. They take on average 772 epochs compared to the training without randomization that last only 59 epochs. It may signify that when the data is not randomized, the validation set contains patterns that the network struggles to learn in the training set. In addition to enhance the reliability of the results, randomizing the sets improves the accuracy significantly. Randomizing the sets seem to be a better way of using the existing data. The results of Table 5 are further presented in

Figures 20 to 24.

Table 4. Error measures of five training runs done without randomization.

	MAE (kW)	RMSE (kW)	MAAPE	Epochs
Training 1	16.87	22.14	4.84%	57
Training 2	17.38	22.37	5.24%	55
Training 3	15.29	20.59	4.51%	73
Training 4	18.74	23.50	5.48%	50
Training 5	16.07	21.03	4.70%	62
Average	16.87 $\pm$ 1.18	21.93 $\pm$ 1.03	4.96 $\pm$ 0.35 %	59 $\pm$ 8

Table 5. Error measures of five training runs done with randomization.

	MAE (kW)	RMSE (kW)	MAAPE	Epochs
Training 1	5.65	8.65	2.53%	763
Training 2	5.50	8.16	2.44%	683
Training 3	5.67	9.08	2.33%	816
Training 4	5.57	8.41	2.50%	781
Training 5	5.73	8.71	2.46%	819
Average	5.62 $\pm$ 0.08	8.60 $\pm$ 0.31	2.45 $\pm$ 0.07 %	772 $\pm$ 49

The training and validation losses of Training 1 are represented in Figure 20. Removing the invalid part of the data and randomizing the sets lead to better viability: the training and validation losses are more appropriate than those of the first experiment.

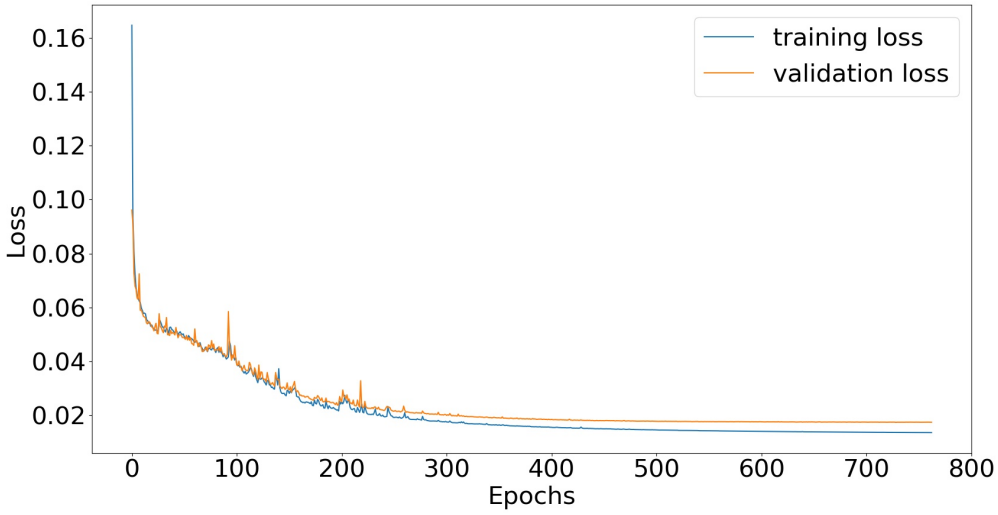


Figure 20. The training and the validation losses for the Training 1.

The best and worst 36-hour forecasting samples for the MAE over five training runs are illustrated in Figure 21 and Figure 22. The worst sample is obtained on a period where the true value of the electricity consumption varies on a large range of values, from 190 to 330 kW. In opposite, the best sample is obtained when it is less varying, from 145 to 165 kW. It seems coherent than the more there is variation into the time series to predict, the more it is hard to fit the right value.

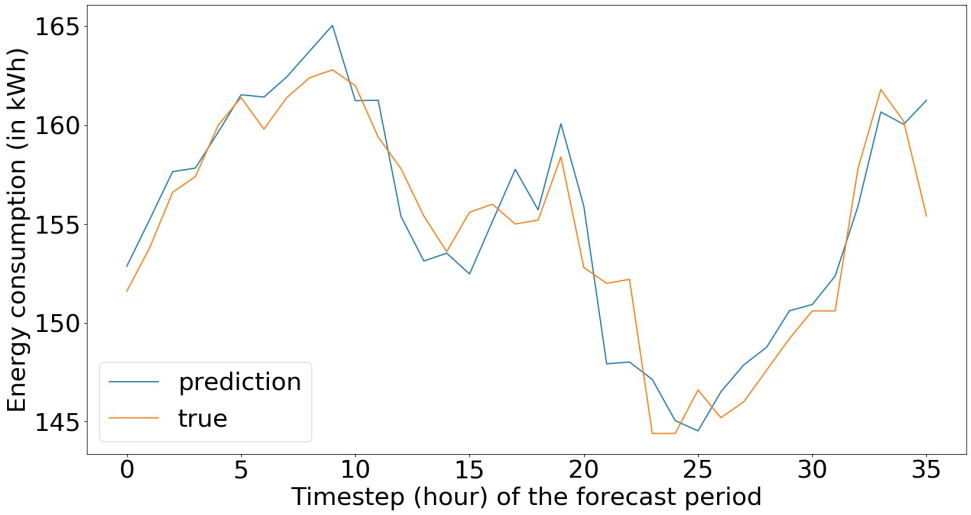


Figure 21. Best forecasting sample for MAE over five training runs: MAE = 1.69 kW.

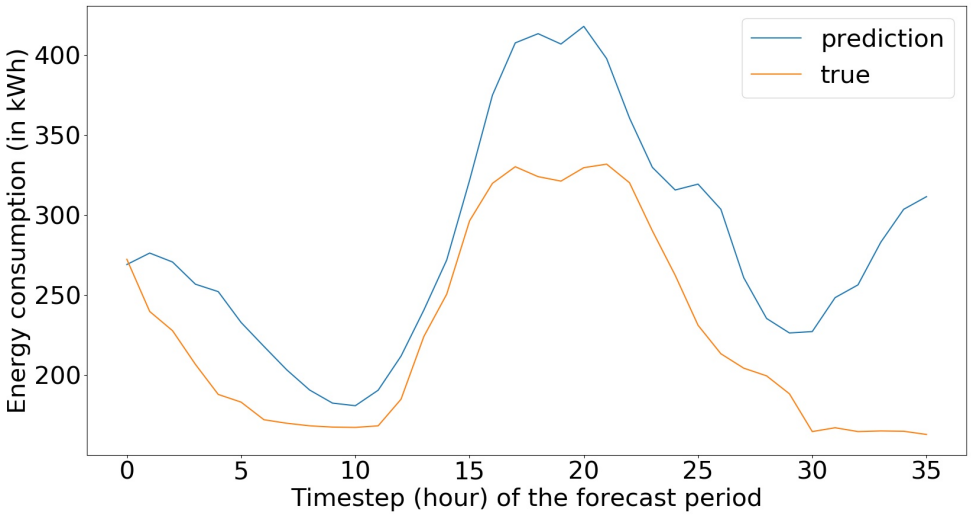
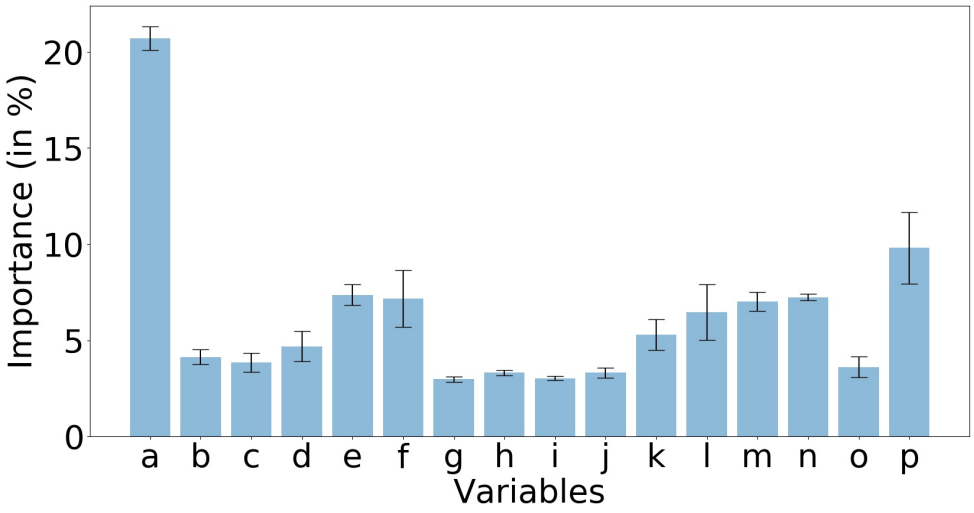


Figure 22. Worst forecasting sample for MAE over five training runs: MAE = 56.46 kW.

4.3.4 Relevance of the variables

As explained in 3, the network has two attention layers: the first one adjusts the importance of the variables and the second one adjusts the importance of the time steps of the window size for every variable.



Name of the variables

a	Energy	i	Gust speed
b	Cloud amount	j	Wind speed
c	Pressure	k	Cycle of the day (sin)
d	Relative humidity	l	Cycle of the day (cos)
e	Air temperature	m	Cycle of the week (sin)
f	Dew-point temperature	n	Cycle of the week (cos)
g	Horizontal visibility	o	Cycle of the year (sin)
h	Wind direction	p	Cycle of the year (cos)

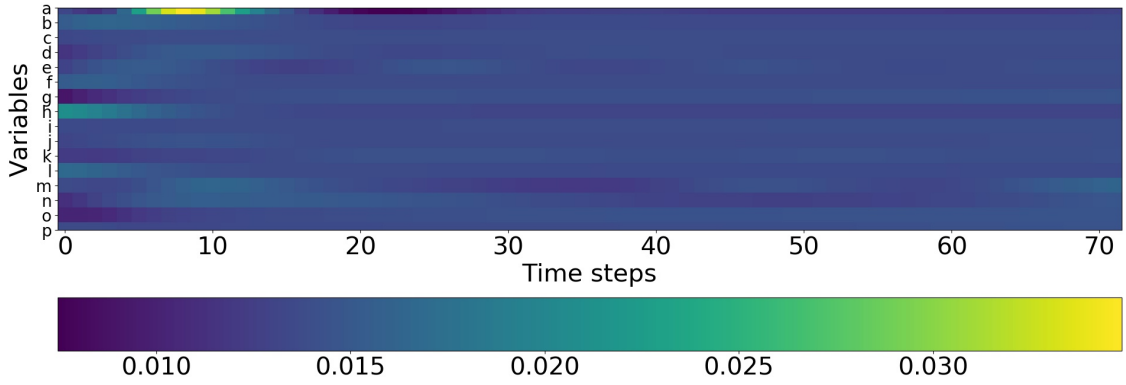
Figure 23. Relevance of the input variables with standard errors for a confidence interval of 95%.

The relevance of the variables inducted by the first attention layer is presented in Figure 23 as a bar graph where the mean and the standard deviation of the importance of the variables were calculated from the 5 training runs of Table 5. The standard deviation for every time series is relatively low. The maximum standard deviation, associated with the cosine time series of the cycle of the year, is around 2%. The five training runs seem therefore to learn the same way as there is no major difference in the importance they gave to the time series.

The variable that is the most considered is the energy consumption. This is totally rational

because it is the target time series. The variables that are then more considered are the cosine time series of the cycle of the year, the air temperature, the dew-point temperature and the cosine and sine time series of the cycle of the week. These variables should impact on the energy consumption so it is natural that they are more considered.

The variables that are less considered are those related to the wind (wind direction, gust speed and wind speed) and horizontal visibility. Even if they may help forecasting the energy consumption, once again it seems rational that they are not over-considered. The first attention mechanism plays globally a coherent role in adjusting the variables of the data.



Name of the variables

a	Energy	i	Gust speed
b	Cloud amount	j	Wind speed
c	Pressure	k	Cycle of the day (sin)
d	Relative humidity	l	Cycle of the day (cos)
e	Air temperature	m	Cycle of the week (sin)
f	Dew-point temperature	n	Cycle of the week (cos)
g	Horizontal visibility	o	Cycle of the year (sin)
h	Wind direction	p	Cycle of the year (cos)

Figure 24. Relevance of the time steps of the input variables for Training 3.

The relevance of the time steps of the window size for the best training run of Table 5 is illustrated in Figure 24. In the figure, time step t corresponds to the t^{th} time step of the window. Time step 0 is the furthest time step from the prediction horizon. The most apparent changes are for the wind speed (variable j in the graph) and for the sine time

series of the cycle of the year (variable o in the graph). For the wind speed, the first time steps of the window are dark blue, which means that they are less considered than the rest. For the sine time series of the cycle of the year, the first time steps are yellow which means that they are more considered. The reason why the network changed these values is not intuitively understandable but this is not inconsistent.

However, most of the rest of the graph is of color cyan corresponding to a value around 1.50%. This value matches with the initial value given to the importance of the time steps which is $1/72 = 1.39\%$. Thereby, the second attention layer has a minimal impact in adjusting the time steps for most of the input time series. Such result seem incoherent with the forecasting: it would have been logical that the closest time steps from the prediction horizon were more considered, in particular for energy which is the target time series.

4.4 Forecasting the solar power generation

The Experiment 2 focuses on the solar power generation. As explained in 4.1, the electricity production data consist of several sub-series depending on the group of panels they correspond. The solar power generation of the groups of panels are studied separately because they may not be correlated the same way to the input variables. But the whole solar power generation, that corresponds to the sum of all the solar power generation time series, is additionally studied and compared. The goals of the experiment are as follows:

- Find out the most suitable window size value for every solar power generation data set.
- Evaluate and compare the forecasting of the solar power generation of the five different groups of panels and the whole electricity production.
- Present the results for forecasting the solar power generation.

4.4.1 Description of the training runs

The values of the parameters that are tested in the Experiment 2 are summarized in Table 6. Six different window size values are tested for finding the most suitable one: 48, 60, 72, 84, 96 and 108 hours. Experiment 1 shows that the prediction horizon value did not affect the choice of the most suitable window size value. That is why, the training runs

are directly performed over a 36-hour prediction horizon, even for optimizing the window size value.

Table 6. Parameter values tested in the Experiment 2.

Parameter	Values tested
Prediction horizon	range of 36 hours
Window size	48, 60, 72, 84, 96, 108
Learning rate	0.001
Validation patience	35
Maximal number of epochs	1000

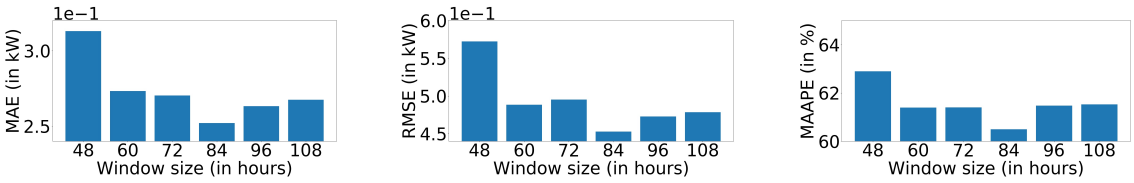
According to the positive results of randomizing the samples in Experiment 1, the samples for Experiment 2 are randomized before the division into training, validation and testing sets.

The performance for forecasting the solar power generation data sets are evaluated and compared, such as the relevance of the input time series. The best and worst 36-hour forecasting samples are illustrated.

4.4.2 Optimization of the window size

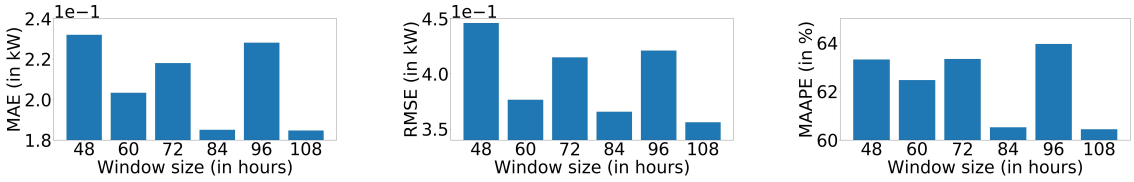
Figures 25 to 29 present the influence of the window size values for forecasting the five different groups of panels. The three metrics MAE, RMSE and MAAPE are considered. 84 hours seem to be the most suitable window size value for forecasting South Wall. A 106-hour window is better for forecasting West Wall. Concerning the forecasting of Carport, the most suitable window size value is not common to all the metrics. 108 hours seem to be appropriate with the MAE and the RMSE but 72 hours is a little better for the MAAPE. The MAE and the RMSE focus more on giving importance to the amplitude of the errors. That is why, the results are a little different. Forecasting Carport solar power generation with 108-hour window is chosen for further comparison between the groups of panels. Concerning the groups Flatroof and Single Panel, 96 hours is the best window size value.

Figure 30 presents the influence of the window size values for forecasting the whole solar power generation. The most suitable window size value for this forecast is 96 hours.



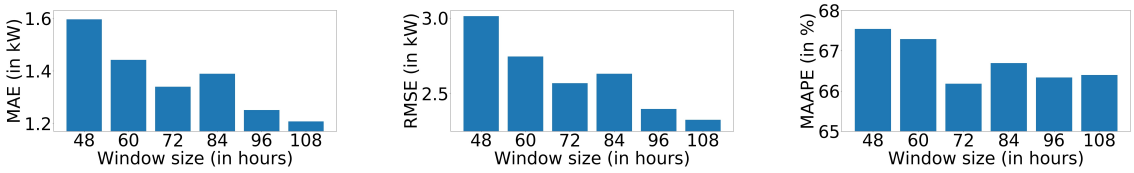
(a) MAE (b) RMSE (c) MAAPE

Figure 25. Influence of the window size values for forecasting the solar power generation of South Wall.



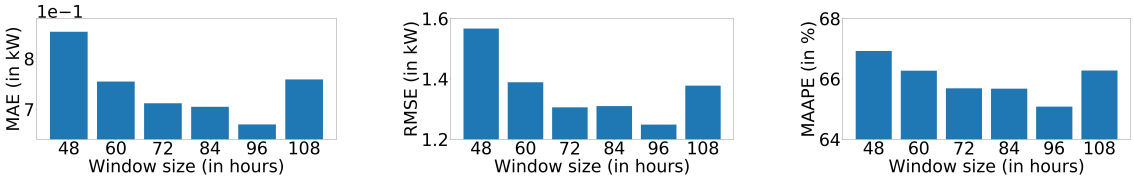
(a) MAE (b) RMSE (c) MAAPE

Figure 26. Influence of the window size values for forecasting the solar power generation of West Wall.



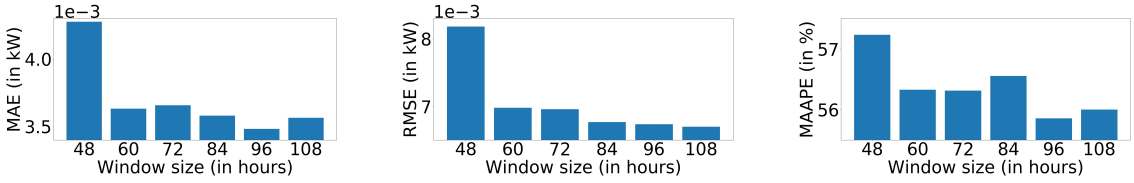
(a) MAE (b) RMSE (c) MAAPE

Figure 27. Influence of the window size values for forecasting the solar power generation of Carport.



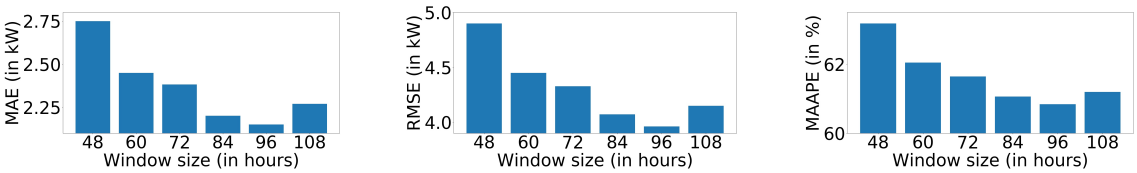
(a) MAE (b) RMSE (c) MAAPE

Figure 28. Influence of the window size values for forecasting the solar power generation of Flatroof.



(a) MAE (b) RMSE (c) MAAPE

Figure 29. Influence of the window size values for forecasting the solar power generation of Single Panel.



(a) MAE (b) RMSE (c) MAAPE

Figure 30. Influence of the window size values for forecasting the whole solar power generation.

4.4.3 Comparison of the forecasting of the groups of panels

Table 7 presents the errors of the best training runs for forecasting every solar power generation. The error metrics MAE and RMSE are included in the table to give a sense of the data range but they cannot be compared between different data sets. For instance, forecasting the solar power generation of Single Panel has lower MAE and RMSE because it corresponds to the production of one single solar panel. Its MAAPE is though comparable to those resulting from the forecast of the other groups of panels. The absolute peak value in $\mathrm{kW_p}$ for every group of panels is added to the table in order to give better understanding of the MAEs and the RMSEs.

The final MAAPEs are high: from 55.86% for Single Panel to 66.40% for Carport. Even though MAAPE works with zero values, it still increases if there are lots of zero values, which is the case for solar power generation data sets. If the true value is zero then the local MAAPE reaches the highest value possible which is $\pi/2$, whatever the predicted value is. That is why, MAAPEs are additionally calculated using a bias that was added to the true and predicted values before calculating the errors. The bias is set to 0.5% of the maximum peak values. It is chosen relatively low compared to the maximum values in order to impact only on the error calculation for the zero values. The MAAPEs with bias are way lower than the standard MAAPEs: they go from 26.91% for the whole production to 33.12% for South Wall. Thus the error metric impacts a lot how the performance is evaluated.

Forecasting the whole solar power generation in one single training run reaches a MAAPE of 60.84% and a MAAPE with bias of 26.91%. When the reference metric is the MAAPE with bias, the training run for forecasting the whole production outperforms all the training runs for forecasting the groups of panels separately.

The best and worst samples for the MAAPE with bias among the forecast of the solar power generation are presented in Figure 31 and Figure 32. The best sample is from West

Table 7. Comparison of the forecast errors for the five groups of panels and the whole production.

Data set	South Wall	West Wall	Carport	Flatroof	Single Panel	Whole Production
Window size	84	108	108	96	96	96
MAE in kW	0.252	0.185	1.208	0.670	0.003	2.151
RMSE in kW	0.453	0.356	2.326	1.249	0.007	3.962
Maximum peak values in $\mathrm{kW_p}$	18.4	18.4	108.0	51.5	0.25	196.6
MAAPE	60.50%	60.44%	66.40%	65.08%	55.86%	60.84%
Bias in kW (0.5% of maximum peak value)	0.092	0.092	0.540	0.258	0.001	0.983
MAAPE with bias	33.12%	27.98%	27.06%	30.01%	27.76%	26.91%

Wall data set. The worst one is from South Wall data set. Most of the best samples are predicted during dark periods, i.e., during nights or during winters. During these periods, there is no electricity to generate, which makes it easier to predict. Visualizing the forecast on these periods is not relevant, as the true value only consists of zero values. That is why, in order to get samples out of the dark periods, the MAAPE with bias was taken as the reference metric for selecting the best and worst samples.

The best forecasting sample perfectly fits with its associated true value. The corresponding period has production peaks that are relatively smooth and low, which makes it easier to forecast. That is why, the forecast is accurate for this sample. The prediction curve of the worst sample looks coherent with the true value curve but there is a lot of noise in the forecast. The associated period seem to correspond to a day where there is sun only for 4-5 hours. The rest of the sample is made of zero values. The main source of errors seem to come from the sudden peak of electricity production with which the network struggles to perfectly match. It is even more difficult for the network to forecast it because the peak is enclosed in a dark period.

4.4.4 Relevance of the variables

Figure 33 presents the relevance of the input time series for every best training run of the different groups of panels and the whole production, previously presented in Table 7. There are in total 19 input time series that was trained for forecasting the different data sets. Even though the importance for a variable can fluctuate a lot from one data set to

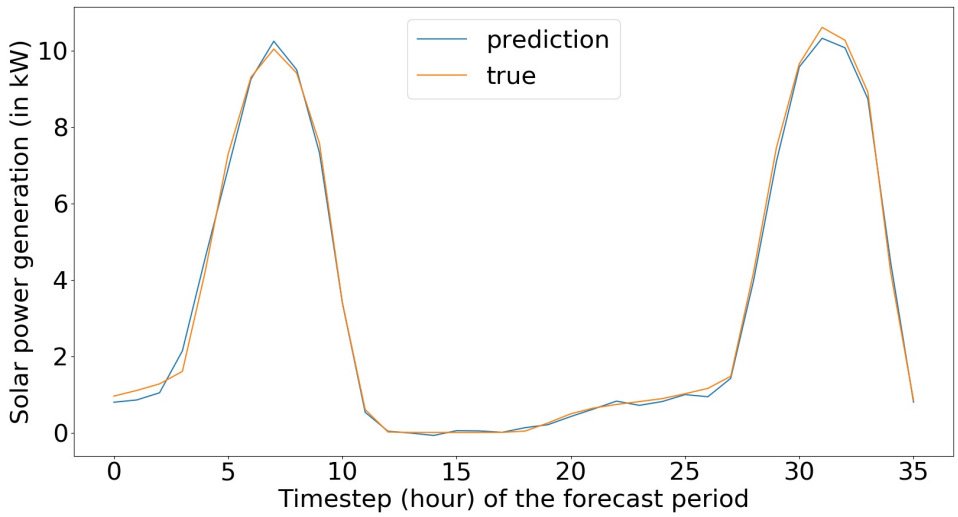


Figure 31. Best forecasting sample reached for West Wall: MAAPE with bias = 8.28%; MAAPE = 17.76%.

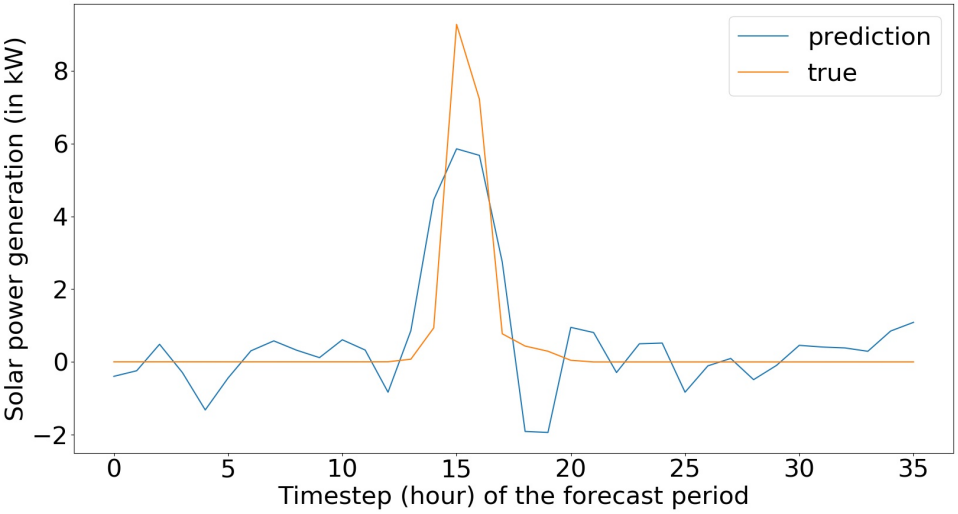


Figure 32. Worst forecasting sample reached for South Wall: MAAPE with bias = 78.72%; MAAPE = 92.99%.

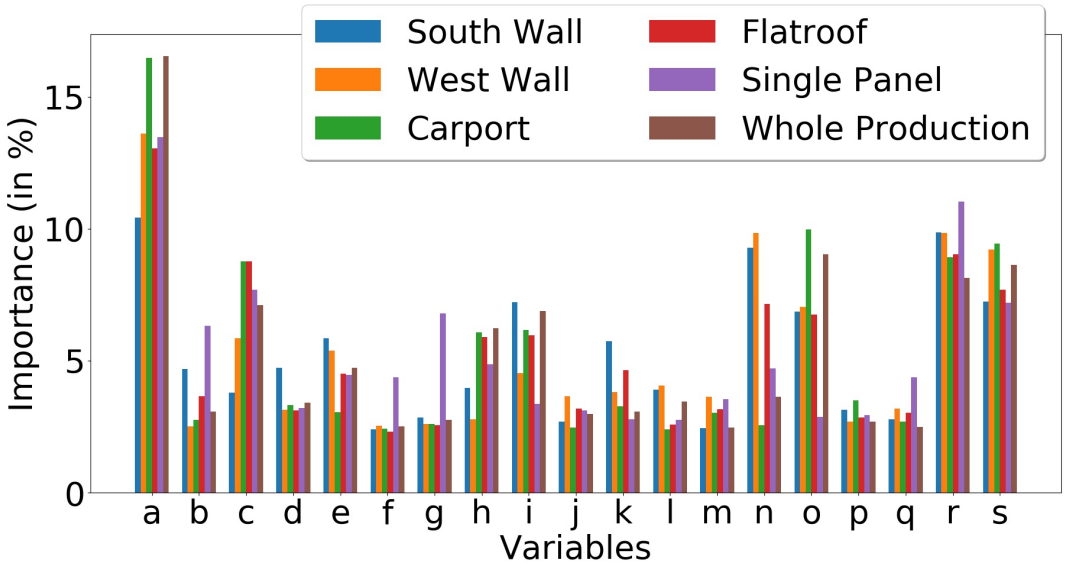
another, several general trends can be extracted from the figure concerning the relevance of the input variables for forecasting the solar power generation.

The target time series is always the input time series that is the most considered. This is rational, for the reason that it is the input time series that contains the more information for the forecast. Calendar data corresponding to the cycle of the year and to the cycle of the day are then a lot considered. The importance of the sine and cosine time series of the cycle of the year are respectively at least 8.2% and 7.2%, while the sine and cosine

time series of the cycle of the day have multiple values over 7%. The solar power generation depends on the season and the brightness, two information that can be potentially extracted from the cycle of the year and the cycle of the day. This is the reason why these variables are relevant for the training. The most considered time series among weather and radiation data are global radiation, dew-point temperature and air temperature which have most of their importance values over 5%. This is as well rational as the radiation and the temperature are two exogenous variables, connected to each other, that impacts the solar power generation.

The input variables diffuse radiation and sunshine radiation are unexpectedly not considered a lot. Diffuse radiation has only one importance value over 5% while all the importance values of sunshine radiation are under 5%. These variables should normally be more considered, as they directly represent the influence of the sun. The reason why these variables are not that much considered may be explained by the fact that the radiation data do not come from Lappeenranta, but from Kotka which is 100km from Lappeenranta. Thereby, they may not be in total accordance with the solar power generation data.

The relevance of the time steps for forecasting the solar power generation is not presented, as the adjustments of the network to nuance the time steps are insignificant.



Name of the variables

a	Solar power generation	k	Wind direction
b	Diffuse radiation	l	Gust speed
c	Global radiation	m	Wind speed
d	Sunshine duration	n	Cycle of the day (sin)
e	Cloud amount	o	Cycle of the day (cos)
f	Pressure	p	Cycle of the week (sin)
g	Relative humidity	q	Cycle of the week (cos)
h	Air temperature	r	Cycle of the year (sin)
i	Dew-point temperature	s	Cycle of the year (cos)
j	Horizontal visibility		

Figure 33. Relevance of the input variables for the five groups of panels and the whole production..

5 DISCUSSION

5.1 Current study

ANNs seem to be reliable models for time series forecasting, in particular the LSTM RNNs that are adapted to automatically extract dynamics of features from a large database of time series. The accuracy obtained for the electricity consumption data corresponds to the expectations based on the literature review. In fact, the best MAPEs of the other models popular for time series forecasting did not go under 5-6 %. Thereby, reaching an average MAAPE of 2.45% is totally acceptable, as it appears to be better than all the other models presented in the literature review. The forecasting of the electricity consumption data can thus be used for concrete forecasting.

Even if good results are achieved for the electricity consumption, the second attention mechanism to evaluate the importance of the time steps seem to be almost inoperative in opposite to the first attention mechanism to nuance the importance of the input time series. Understanding more deeply the reason why the second attention mechanism struggles to work and solving this issue would lead to better use of the window associated with the samples and, therefore, would lead to better forecasting. Furthermore, this issue may explain why in the Experiment 1 the training runs with larger window sizes than 72 hours did not give better results than the training run done with window size of 72 hours. A too large window size is counterproductive for the training run because the network does not succeed to nuance efficiently the importance of the time steps.

The forecasting of the solar power generation reaches a MAAPE of 55.86% at best. In order to limit the impact of zero values in the performance evaluation, a MAAPE with bias was additionally calculated. The best MAAPE with was thus 26.91% at best. However, such a MAAPE is still far behind expectations. If we compare it with the study of 2017 about forecasting residential load forecasting [21], a MAAPE of 26.91% is similar to the MAPEs of empirical mean and MAPE minimization, two former forecasting methods that work with basic concepts. It is far from the 8.18% of average MAPE for the LSTM RNNs.

In the study that introduces the IMV-LSTM [15], the model was though really accurate for forecasting the energy production of a photo-voltaic power plant in Italy with 9 additional input weather time series. Similar performance was, therefore, expected as the electricity production data sets of the thesis are in the same way solar power generation data sets

associated with weather data. That is why, the reason why the performance is mediocre may come from the data of the thesis. Two particular reasons may explain why forecasting the solar power generation is not efficient:

- The radiation data were from Kotka and not from Lappeenranta. Radiation data that would be more in accordance with the rest of the data could potentially improve the results. This is especially true given that the diffuse radiation and the sunshine radiation are two input variables that are unexpectedly not considered a lot by the network.
- The solar power generation data sets of the thesis seem to vary more arbitrarily than other data sets. For example, the electricity consumption seemed to be more regular and smoother.

Finally, training the solar power generation of the groups of panels separately was less efficient than training the whole solar power generation in one single training. With a MAAPE with bias of 26.91%, the training run for forecasting the whole production outperforms all the training runs for forecasting the groups of panels separately. The initial idea of training the solar power generation data sets separately was to take into account the fact that the groups of panels are not composed of the same number of panels and orientated the same way. The reliance on the input variables for every solar power generation is different so it should have been better to train the data sets separately. However, summing the whole production seems to give better benefit, as it reduces the influence of the noise that comes from the different groups of panels. The approach of training the whole production in one single run is, therefore, preferred.

5.2 Future work

As presented in 2.3, the CNN can be a good choice for forecasting, even more if it is combined with a RNN. The model that was tested is a state-of-the-art RNN, combining the most novel characteristics. However, no convolutional layer is included in its structure. One potential improvement could then be to combine the tested network with a CNN. The idea of this combination is to get additional deep features. A CNN may extract useful features that a LSTM RNN does not detect.

This thesis considered only solar electricity generation. Solar panels are the technological solutions that LUT University decided to rely on. However, a lot of other ways of elec-

tricity production exist such as wind turbines or dams. It would be interesting to study forecasting of other ways of electricity production.

6 CONCLUSION

In this thesis, the best forecasting models for time series were compared according to different articles. The LSTM RNNs represent the most state-of-the-art and adequate models to forecast time series data, in particular electricity consumption and production. The model that was chosen is the IMV-LSTM, developed in 2019, which implements novel concept to bring to the fore the most useful part of the data by using attention mechanisms.

The model was first tested with the electricity consumption data collected from the campus of LUT University over 36-hour prediction horizons. The average MAAPE achieved for forecasting the electricity consumption was 2.45%. The first attention layer plays a crucial role in achieving such results by nuancing the impact of the variables in an understandable way. The second attention layer, though, had trouble to clearly put forward the most useful time steps of the time series.

The different solar electricity sites collected from the campus of the LUT University, were then studied and modelled. The forecasting performance was less satisfactory, reaching MAAPE of 55.86% at best. These mediocre results seemed first to come from the complexity to evaluate a legitimate error because of the important number of zero values there are in the solar power generation data sets. In order to cope with this issue, a MAAPE with bias was calculated, reaching an error of 26.91% at best. However, this performance is still unsatisfactory. The problem for forecasting the solar power generation may come from the data itself, in particular the radiation data may be irrelevant because they have been collected from the meteorological station of Kotka that is located 100km from LUT University.

REFERENCES

[1] Odyssee-Mure. Energy efficiency trends in buildings. https://www.odyssee-mure.eu/publications/policy-brief/buildings-energy-efficiency-trends.html, June 2018. Online; accessed 19 January 2020.

[2] Aurélie Foucquier, Sylvain Robert, Frédéric Suard, Louis Stephan, and Arnaud Jay. State of the art in building modelling and energy performances prediction: A review. *Renewable and Sustainable Energy Reviews*, 23:272–288, 2013.

[3] Drury B. Crawley, Jon W. Hand, Michaël Kummert, and Brent T. Griffith. Contrasting the capabilities of building energy performance simulation programs. *Building and Environment*, 43(4):661–673, 2008.

[4] Zeyu Wang and Ravi S. Srinivasan. A review of artificial intelligence based building energy use prediction: Contrasting the capabilities of single and ensemble prediction models. *Renewable and Sustainable Energy Reviews*, 75:796–808, 2017.

[5] Søren Krohn, Poul-Erik Morthorst, and Shimon Awerbuch. *The economics of wind energy*. European Wind Energy Association, 2009.

[6] Nesreen Ahmed, Amir Atiya, Neamat Gayar, and Hisham El-Shishiny. An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, 29:594–621, 08 2010.

[7] Matthew Riemer, Aditya Vempaty, Flavio P. Calmon, Fenno F. Heath, Richard Hull, and Elham Khabiri. Correcting forecasts with multifactor neural attention. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, page 3010–3019. JMLR.org, 2016.

[8] Roger Frigola and Carl Edward Rasmussen. Integrated pre-processing for bayesian nonlinear system identification with gaussian processes. In *52nd IEEE Conference on Decision and Control*, pages 5371–5376. IEEE, 2013.

[9] Christopher M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, 1995.

[10] Tara N. Sainath, Abdel-rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran. Deep convolutional neural networks for LVCSR. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8614–8618. IEEE, 2013.

[11] Bendong Zhao, Huanzhang Lu, Shangfeng Chen, Junliang Liu, and Dongya Wu. Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics*, 28(1):162–169, 2017.

[12] Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[13] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[15] Tian Guo, Tao Lin, and Nino Antulov-Fantulin. Exploring interpretable LSTM neural networks over multi-variable data. In *Proceedings of the 36th International Conference on Machine Learning, PMLR*, pages 2494–2504, Long Beach, California, USA, 2019.

[16] Xin Wang, Yuanchao Liu, Cheng-Jie Sun, Baoxun Wang, and Xiaolong Wang. Predicting polarities of tweets by composing word embeddings with long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1343–1353, 2015.

[17] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[18] Yao Qin, Dongjin Song, Haifeng Cheng, Wei Cheng, Guofei Jiang, and Garrison W. Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, page 2627–2633. AAAI Press, 2017.

[19] Tara N. Sainath, Oriol Vinyals, Andrew Senior, and Haşim Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4580–4584. IEEE, 2015.

[20] Magnus Dahl, Adam Brun, and Gorm Andresen. Using ensemble weather predictions in district heating operation and load forecasting. *Applied Energy*, 193:455–465, 2017.

[21] Weicong Kong, Zhao Yang Dong, Youwei Jia, David J. Hill, Yan Xu, and Yuan Zhang. Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Transactions on Smart Grid*, 10(1):841–851, 2017.

[22] Aowabin Rahman, Vivek Srikumar, and Amanda D. Smith. Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks. *Applied Energy*, 212:372–385, 2018.

[23] Daniel L. Marino, Kasun Amarasinghe, and Milos Manic. Building energy load forecasting using deep neural networks. In *42nd Annual Conference of the IEEE Industrial Electronics Society (IES)*, pages 7046–7051. IEEE, 2016.

[24] Joel Janek Dabrowski, YiFan Zhang, and Ashfaqur Rahman. Forecastnet: A time-variant deep feed-forward neural network architecture for multi-step-ahead time-series forecasting. *arXiv preprint arXiv:2002.04155*, 2020.

[25] Grafana. The open observability platform. `https://grafana.com/`, 2020. Online; accessed 17 May 2020.

[26] Finnish Meteorological Institute. Download observations. `https://en.ilmatieteenlaitos.fi/download-observations#!/`, 2020. Online; accessed 17 May 2020.

[27] Ian London. Encoding cyclical continuous features - 24-hour time. `https://ianlondon.github.io/blog/encoding-cyclical-features-24hour-time/`, 2017. Online; accessed 17 May 2020.

[28] Andrich van Wyk. Encoding cyclical features for deep learning. `https://www.kaggle.com/avanwyk/encoding-cyclical-features-for-deep-learning`, 2018. Online; accessed 17 May 2020.

[29] Wikipedia. Mean absolute error. `https://en.wikipedia.org/wiki/Mean_absolute_error`, 2020. Online; accessed 17 May 2020.

[30] Wikipedia. Root-mean-square deviation. `https://en.wikipedia.org/wiki/Root-mean-square_deviation`, 2020. Online; accessed 17 May 2020.

[31] Sungil Kim and Heeyoung Kim. A new metric of absolute percentage error for intermittent demand forecasts. *International Journal of Forecasting*, 32(3):669–679, 2016.