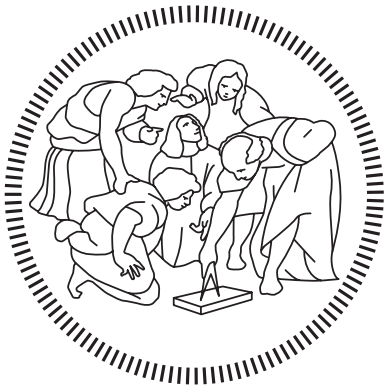
Politecnico di Milano

SCHOOL OF INDUSTRIAL AND INFORMATION ENGINEERING
Master of Science – Automation and Control Engineering



Short-Term Load Forecasting of Non-Residential Building with hybrid LSTM and ARX model

Supervisor
Lorenzo Mario FAGIANO

Co-Supervisor
Marco LAURICELLA

Candidate
Francesco LAEZZA – 905157

Academic Year 2020 – 2021

Acknowledgements

La prima persona che voglio ringraziare è mio fratello, Alessandro, per avermi spinto e dato la forza di andare avanti nonostante le difficoltà, per esserci sempre stato nei momenti felici e per avermi fatto da guida nei momenti bui, senza di lui il raggiungimento di questo obiettivo non sarebbe stato possibile. Mia mamma e mio papà, Anna e Luigi, per l'incredibile amore, forza e supporto che mi hanno continuatamente trasmesso sin dalla mia partenza. Li ringrazio per avermi sempre spinto ad uscire dalla mia confort zone per provare nuove esperienze, è grazie a loro se sono riuscito in tutto ciò.
Un immenso grazie va al prof. Fagiano e all'Ing. Lauricella, supervisore e co-supervisore per il mio progetto di tesi. Voglio ringraziarli per l'immensa disponibilità e dal punto di vista professinale e dal punto di vista umano mostrata durante tutto il periodo di analisi e di sintesi. Tutte le persone che mi hanno accompagnato in questo percorso accademico da vicino e da lontano. I miei amici di giù: Mario e Alessandro. I miei amici di giù e di su: Cræk e Marco. I ragazzi del QG: Nino, Luca, Macchia, Ciccio, Nino, Luca. I miei compagni di corso: Gianluca, Pasquale, Francesco, Alberto. La 17esima board di BEST: Chiara, Ersy, Fabs, Teo e Jeh. Ciccio. Tutta BEST Milano perchè mi ha permesso di vivere anni ed eventi incredibili, che non dimenticherò mia nella vita. Infine un ringraziamento va a tutti coloro hanno condiviso anche solo una risata o un'incazzatura con me durante questi lunghi anni a Milano ed al Poli.

Abstract

In the current fast-changing industrial environment, the effective and efficient prediction of electricity demand of factories and buildings is a critical issue especially in the early stage. In fact, the lack of data at the beginning of the system's life makes it difficult to take reliable decisions. This is why in the past 60 years researchers tried to invest their time in looking for a good solution. In this thesis a new prediction approach for Short-Load-Term-Forecasting (STLF) using a small amount of data (e.g. 2 weeks) is developed to predict one-day-ahead load of the following week. The numerical validation is based on real data collected by an Italian company located in Bergamo. The final two-stage prediction model employs Long-Short-Term-Memory (LSTM) based Neural Network and an Autoregressive with Exogenous Input (ARX) linear model. The former catches the non-linear complex dynamics and the latter captures the remaining linear dynamics. The proposed model is integrated with weather data and Fictitious Input (FI). Neural Network and intuitive Naive approaches for weather data are developed and the model with the overall best prediction capabilities is considered. A literature review is performed to understand the state-of-the-art prior to the quantitative studies of the proposed approach considering previous thesis on this specific dataset. Furthermore, two coefficients are used for comparative analysis. MAPE coefficient determines forecasting performances and R^2 coefficient is employed for fitting performances. The new model's MAPE and R^2 are compared to state-of-the-art ARX and with the single LSTM model results. The proposed method demonstrates superior one-day-ahead forecasting performances with respect to the other models. This conclusion suggests the proposed model should be further developed to better exploit all the characteristics and furtherly improve the forecast. One of the most relevant advantages is the possibility to exploit previous research to compute guaranteed error bounds on the linear residual prediction.

Sommario

Nell'attuale contesto industriale in rapida evoluzione, la predizione efficace ed efficiente del carico elettrico di industrie ed edifici specialmente nella fase di avvio è un problema critico. La mancanza di dati all'inizio dell'utilizzo rende difficile poter prendere decisioni affidabili. Per questo motivo, da oltre 60 anni i ricercatori investono il loro tempo nel trovare buone soluzioni. Pertanto, in questa tesi viene proposto un nuovo approccio a due stadi per la predizione del consumo elettrico a breve termine (STLF), utilizzando una piccola quantità di dati (e.g. 2 settimane) al fine di predirne giorno per giorno il carico della settimana successiva. L'analisi numerica si basa su dati reali raccolti da una fabbrica italiana nel Bergamasco. Il modello di predizione finale usa dapprima un rete neurale basata su Long-Short-Term-Memory (LSTM) e poi un modello Autoregressivo con Input Esogeno (ARX). Il primo viene utilizzato con lo scopo di predire la dinamica non lineare, mentre il secondo viene utilizzato per la predizione delle rimanenti dinamiche. Il modello proposto è integrato con dati meteorologici e con un input fittizio (FI). In più vengono proposti dei modelli per la predizione dei dati meteorologici utilizzando soluzioni basate su reti neurali e approcci intuitivi. Una revisione della letteratura per conoscere lo stato dell'arte è eseguita prima dell'analisi numerica considerando precedenti lavori che già trattano e analizzano questi dati. Per determinare la capacità predittiva del modello viene utilizzato il coefficiente MAPE, mentre per determinare la capacità di fitting dei dati viene usato il coefficiente R^2. Il MAPE e l'R^2 del nuovo modello sono paragonati a quelli ottenuti con un modello ARX come proposto in un precedente lavoro e con lo stesso modello LSTM. I risultati dimostrano che il metodo proposto possiede maggiori capacità di predizione rispetto agli altri modelli. Ciò suggerisce che nuovi studi dovrebbero essere condotti basandosi su questa soluzione per sfruttare meglio tutte le caratteristiche e migliorare ulteriormente la capacità predittiva. Un importante vantaggio di questa soluzione è la possibilità di applicare precedenti risultati di ricerca per calcolare errori di accuratezza garantiti del residuo derivante del modello lineare Autoregressivo.

Contents

Acknowledgements **iii**

Abstract **v**

Sommario **vii**

Sommario **vii**

Contents **x**

List of Figures **xii**

List of Tables **xiv**

1 Introduction and literature review **1**
1.1 Background 1
1.2 Literature review 2
1.2.1 Classification of forecasting approaches 2
1.2.2 Summary 3
1.3 Brief description of the work 4

2 Problem overview **7**
2.1 Dataset description 7
2.1.1 Load dataset 7
2.1.2 Weather dataset 8
2.1.3 Time input 8
2.2 Problem formulation 10
2.3 Dataset correlation analysis 11
2.3.1 Weather-load data correlation 11
2.3.2 Load data characteristics correlation 13
2.4 Dataset pre-processing 13
2.4.1 Missing data 13
2.4.2 Weather dataset adaptation 15
2.4.3 Normalization 15
2.4.4 Data correlation results 15

3 Weather prediction model **21**
3.1 Temperature forecasting 21
3.1.1 Naive approach 22

3.1.2 Feed-forward Neural Network 23
3.2 Radiation forecasting 23

4 Load forecasting model **25**
4.1 Final prediction model 25
4.2 LSTM unit 26
4.3 Time input 27
4.4 Non linear prediction model 29
4.5 Linear prediction model for residual 31

5 Implementation and experiments **33**
5.1 Evaluation indices 33
5.2 Test set definition 33
5.3 Result and comparative evaluation 34
5.3.1 Exogenous input result 36
5.3.2 ARX model for load 36
5.3.3 ARX model for residual 36
5.3.4 Combined models 37

Bibliography **48**

List of Figures

Figure 2.1 Example of a full three weeks electricity measurement from Mon 4 March 2019 to Sun 24 March 2019. 8
Figure 2.2 Example of a full three weeks weather measurement from Mon 4 March 2019 to Sun 24 March 2019. From top to bottom temperature, global radiation, relative humidity, average wind speed and maximum wind speed 9
Figure 2.3 Block diagram of the final proposed prediction approach. At first, the weather prediction is run with a proper model. Then the load prediction routine starts: first data are fed into LSTM block that provides the non-linear prediction. Afterwards the obtained residual is used to identify an ARX model that will predict the future residual. The final forecasted time series will be the sum of the LSTM and ARX predictions 10
Figure 2.4 Scatter plots of weather variables vs load power. Up-left: Temperature. Up-right: Global Radiation. Middle-left: Relative Humidity. Middle-right: Maximum wind speed. Bottom: Average wind speed. . 12
Figure 2.5 Scatter plots of load average power. Left: average value of day i with respect to day $i-1$. Right: average value of day i with respect to day $i-7$. Up: value with respect to the previous day. Middle: value of working days with respect to the previous working day. Bottom: value of weekends with respect to the previous weekend 14

Figure 3.1 Naive forecasting approach for temperature (above) and global radiation (below). From temperature plot it is possible to see that the model better behave when the time series evolution has slow dynamic and has a regular trend like in the predicted Saturday and Sunday. On the other hand the prediction has worse accuracy when the temperature curve of the known days has an irregular shape; like in the firsts predicted day of the time series. While for radiation the two models perfectly behave when the curve of two subsequent day are identical. 23
Figure 3.2 Forecasting approach for temperature (above) and global radiation (below) with feed-forward neural network 24
Figure 3.3 Time based method forecasting for radiation 24

Figure 4.1 Information flow of the proposed prediction set-up. Input data gathers all the weather and time inputs. 25
Figure 4.2 LSTM Cell 26
Figure 4.3 Diagram of a general LSTM layer 27

Figure 4.4 Flow of information of the proposed LSTM-based non-linear prediction 28
Figure 4.5 Fictitious Input (FI) time series example. From top to bottom: example of a daily and weekly periodic FI signal, Fictitious Input signal deriving from ARX identification for load prediction and Fictitious Input signal deriving from ARX identification for residual prediction . 29

Figure 5.1 Prediction result for week no. 2 with real weather data input. In blue observed data and in red prediction data. From top to bottom: prediction with LSTM model, residual prediction with ARX model, prediction with the combination of LSTM and ARX model, load prediction with ARX model and temperature and radiation. In green is highlighted the validation set and the green plot shows how MAPE varies day by day 38
Figure 5.2 Prediction result for week no. 9 with real weather data input. In blue observed data and in red prediction data. From top to bottom: prediction with LSTM model, residual prediction with ARX model, prediction with the combination of LSTM and ARX model, load prediction with ARX model and temperature and radiation. In green is highlighted the validation set and the green plot shows how MAPE varies day by day 39
Figure 5.3 Prediction result for week no. 6 with "First Sample" model weather input. In blue observed data and in red prediction data. From top to bottom: prediction with LSTM model, residual prediction with ARX model, prediction with the combination of LSTM and ARX model, load prediction with ARX model and prediction of temperature and radiation. In green is highlighted the validation set and the green plot shows how MAPE varies day by day 40
Figure 5.4 Prediction result for week no. 8 with "First Sample" model weather input. In blue observed data and in red prediction data. From top to bottom: prediction with LSTM model, residual prediction with ARX model, prediction with the combination of LSTM and ARX model, load prediction with ARX model and prediction of temperature and radiation. In green is highlighted the validation set and the green plot shows how MAPE varies day by day 41

List of Tables

Table 2.1 Test result with LSTM NN only. Average MAPE values over 7 simulations of load forecasting performance for different weather input. The first value represent the average MAPE while the value between brackets is the standard deviation of the distribution. In **bold** are highlighted the value with the lower average MAPE. T, R and H respectively state whether temperature, radiation and humidity input signal is used. From left to right the columns include: model without artificial input, model with *Daytype1* input, model with *Daytype2* input, model with *Time* input, model with a combination of *Daytype1* and *Time* input, model with a combination of *Daytype2* and *Time* input, model with a Fictitious Input with only daily and weekly periodic signal, model with Fictitious Input with all harmonics 16

Table 2.2 Test result with LSTM NN only. Average MAPE values along 7 simulations of load forecasting performance for new feature data input. The first value represent the average MAPE while the value between brackets is the standard deviation of the distribution. In **bold** are highlighted the value with the lower average MAPE. Max, Min and Avg respectively state whether the maximum, the minimum and the average load of the day before is used as input signal. From left to right the columns include: model with a combination of *Daytype2* and *Time* input and model with $FI_{1:14}$ input. T, R and H states whether temperature, radiation and humidity are used as input signal. 18

Table 2.3 Test result with LSTM NN only. Average MAPE values along 7 simulations of load forecasting performance for new feature data input. The first value represent the average MAPE while the value between brackets is the standard deviation of the distribution. In **bold** are highlighted the value with the lower average MAPE. Max, Min and Avg respectively state whether the maximum, the minimum and the average load of the same-type day before is used as input signal. From left to right the columns include: model with a combination of *Daytype2* and *Time* input and model with $FI_{1:14}$ input. T, R and H states whether temperature, radiation and humidity are used as input signal. 19

Table 3.1 MAPE values of one-week-ahead temperature and radiation forecasting. In **bold** are highlighted the value with the lower MAPE. FS state the *First Samples* method is used, Dos refers to the *Difference of Samples*, FFNN is the *Feed-forward Neural Network* and TM is the *Time method* developed only for radiation 21

Table 5.1 Dataset containing the nine 3-weeks-long time span on which the routine is tested. "Start" states the beginning, "End training" states the end of the model training dataset (e.g. 2 weeks ahead) and "End" states the date up to the prediction is executed (e.g. 3 weeks ahead) . 34

Table 5.2 MAPE and R^2 values of one-week-ahead load forecasting for the 9 of different prediction models prediction for weeks in Table 5.1. The exogenous input used are temperature, radiation and $FI_{1:14}$ for ARX model and temperature, radiation and $FI_{1,7}$ for LSTM model. In **bold** are highlighted the value with the lower MAPE and with the best fitting coefficient. From left to right: load prediction with LSTM only with *First samples* weather model, load prediction with ARX only with *First samples* weather model, load prediction with LSTM+ARX with *First samples* weather model and load prediction with LSTM+ARX with observed weather data 35

Chapter 1

Introduction and literature review

1.1 Background

It has been around a hundred years since power systems have been developing as self-managed system with a proper control routine and proper embedded communication. From simple solutions they are undergoing a disruptive revolution revolution [1] becoming Smart Grids [2] and part of a full interconnected world of smart objects, most importantly. The optimization of electricity demand has become a key point especially for non residential building to make it possible to maximize the efficiency, reliability and security of power grids. A large amount of methods and models are employed in load forecasting to increasingly improve the prediction performances [3, 4], going from simple linear model to Artificial Neural Networks. Also, different combination of such models and powerful pre-processing tools are employed [5, 6] in order to increase prediction precision. Around 20% and 40% of the total worldwide consumed energy comes from buildings. This means that having a good management and prediction of energy consumption could lead to significant savings for consumers, and a more efficient and sustainable energy system overall. Load forecasting has therefore become a central topic of this research area and even marginal improvements are fundamental for future achievements.

Load forecasting research could be divided into three main areas which differentiate each other on the basis of the future timespan the prediction is performed:

Short term Load Forecasting (STLF) has the aim of predicting the consumed electricity up to 1 week ahead by daily minimizing cost function and providing an hourly (or sub-hourly) based detail

Medium term Load Forecasting (MTLF) to predict load up to some years ahead. It is based on a weekly resolution and is mainly used for medium term planning of the production

Long term Load Forecasting (LTLF) is used to predict decades ahead of electricity consumption in order to make long term plans and plan public infrastructure expansions

This thesis is focussed on STLF using real load data recorded at an ABB office building located in Bergamo, Italy. The forecast is 24 hours ahead for the following week of

non-residential building with a resolution data of 15 minutes, having two weeks of data available.

1.2 Literature review

The literature review focuses on STLF with small sample dataset. In particular, papers based on non-residential buildings are considered. Electricity demand of commercial buildings has peculiar features [7]. In particular, the time series curve tends to be strongly periodic and stationary. Daily, weekly, monthly and seasonally periodic behaviour are present and they are mainly influenced by occupancy of the building, by weather parameters and by the type of the day (differentiating between working days and holidays). Studies on forecasting models have been going on for more than 60 years and different reviews [8] are present as well as precise reviews of Neural Network approaches [3] and general Artificial Intelligence (AI) approaches [9]. A big amount of research results are thus present in literature, going from basic linear models to more complex non-linear self-learning models.

1.2.1 Classification of forecasting approaches

Mathematical models

The first and simpler approaches involve statistical and mathematical methods. The popularity of such methods has increased year by year mainly for their capability of capturing periodicity in time series with a low computational effort, despite their simplicity. They need an explicit model that puts into relationship inputs and outputs. The first statistical models developed include Multiple Linear Regression (MLR), Kalman filtering and state-space models. A huge variety of methods are tested going from moving average (MA) to autoregressive (AR) models. Their combination and variants such as autoregressive with moving average (ARMA), autoregressive integrated moving average (ARIMA) [10], autoregressive with exogenous input (ARX), autoregressive with moving average and exogenous input (ARMAX) are tested in the analysed literature.
Among all the possible variety of models, the most effective one is the ARX model [9]. It can be fed with different inputs and combination of itself in order to empathize the periodicity of the load consumption curve trying to achieve always better results. Thanks to this conclusion, ARX model is the baseline of this thesis.

Artificial intelligence

Statistical methods have been subject of many of research studies making them well based methods with clear prediction capabilities. As mentioned above, statistical approaches are characterised by simplicity and good performances but this results also in the difficulty of catching intrinsic non-linearities between the time series and input data. As a result, new more complex models and their combination and slight modification are taken into consideration. Many research results are present utilising Gaussian processes [11–14], Support Vector Machine (SVM) [15, 16], Regression Trees [7, 17], Deep Learning approaches [18–20] hybrid models with data fuzzification

logic [21–23]. A common result deriving from all these approaches is the presence of a large amount of local minima which the gradient descent routine could converge to. This means not exploiting the model at its maximum and resulting in less accurate forecasting performances. To deal with this, advanced smart algorithms that exploit more than one initialization vectors [24, 25] help with falling into the global minimum. The most promising model seems to be Neural Network approach, especially Deep Learning Approaches involving Recurrent Neural Network.
Long Short Term Memory (LSTM) networks have gained popularity in a large number of application fields including load forecasting prediction. It is an innovative type of Recurrent Neural Network and is widely used for the topic research of this thesis. It has been tested in combination with other techniques involving Convolutional Neural Network [19], decomposition of time series [26], Empirical Mode Decomposition (EMD) and XGoost Regression Tree [7]. Its big capability of avoiding vanishing gradient problem that affects most of Recurrent Neural Network is a key additional advantage that gave LSTM great notoriety.

Preprocessing and limited data

Pre-processing of data could help in improving forecasting performances [27]. Different pre-processing methods are present in literature already from early 1960s. Simple time based algorithms such as Seasonal Trend decomposition based on Loess (STL) [26, 28] and frequency based algorithm like Empirical Mode Decomposition (EMD) [7] show improvements in many cases in the final load prediction performance. Clustering of data is a well-based technique along with the most powerful generation [29] and augmentation [30] of data. This new techniques is going towards a very big revolution thanks to Artificial Intelligence and, in particular, Neural Network. Augmentation and generation of data has the aim of providing new uncorrelated data to the training dataset of the forecasting model in order to increase the information provided to reach better results. There are basic and advanced approaches [30] for data augmentation. The former generate new data by simple modification of the original one like flipping, warping, perturbing etc. The latter exploit decomposition and modelling approaches and also self-learning and generative techniques, being Generative Adversarial Network (GAN) the best known and advanced, developed in 2014. A key point is the availability of a limited quantity of training data. This issue is faced in literature exploiting techniques like Transfer Learning [31] which consists in exploiting an already trained network. The big drawback of this technique is to find a similar Neural Network. More solutions to small dataset problem are generally faced trying to find the best model set-up, as tried in Lin *et al.*, Huang *et al.*, Yuan *et al.* [32–34].

1.2.2 Summary

Since there are enough results for basic statistical and mathematical based method, the thesis is focused on AI-based algorithms, specifically on LSTM Neural Network. This type of models must be tested carefully. The prediction capability of these models is strongly dependent on the dataset and easily leads to overfitting of data [35]. Recently the research on STLF models has taken two different directions:

- Using increasingly more complex models in order to improve forecasting performances by modelling high non-linearities at a cost of an high computation demand
- Developing new techniques to exploit simple models that require less computational effort and that are easily embeddable in industrial environment

Another important issue that must be carefully taken into consideration is the over-fitting. The forecasting capability of complex models like the ones just described strongly depend on the dataset they are tested on. Moreover, lots of statistical decisions are made during training phase, making the model behave differently even when exactly the same dataset is used. This is a typical sign of to over-fitting of data, due to the high number of parameters to be estimated. The over-fitting is the situation in which the model not only approximates the dynamic of the data but fits also the noise that is constantly present. This situation is translated into a low forecasting performance. There are different solutions to avoid over-fitting. The most intuitive solution is surely reducing the number of parameters. Moreover, drop-out layer and statistical zeroing parameters techniques are employed for Neural Networks.

1.3 Brief description of the work

This thesis research on a new forecasting algorithms for load consumption for non-residential buildings. The preliminary step of the algorithm is the prediction of weather variables used as exogenous input. Then, the load prediction is carried on with a strong non-linear LSTM Neural Network model using Adam optimizer. It is one of the most efficient and innovative optimizer that exploits a big set of tuning parameters to obtain the best results. The result of this prediction is then fed into an ARX model that is identified through *Simulation Error Method* (SEM) [37, 38] to obtain the most accurate model at the lower possible computational request. In general, in order to avoid luckily/unluckily chosen initial conditions, the identification routines are performed multiple times and the mean and standard deviation of the result are considered. The key part of using a liner model is the exploitation of previous result from [36]. The authors were able to derive guaranteed error bounds using a Set Membership (SM) approach. Both models are integrated with Fictitious Input (FI) signal that marks time passing. Thanks to this input, the models intrinsically catch periodicities of the energy consumption being able to perform a sort of frequency decomposition of the time series without resorting to external routines.
Finally, the performances of the hybrid model are compared through fitting (R^2) and prediction (MAPE) coefficients, with classical ARX model and with the LSTM model without the further linear residual correction. As a result, an outstanding improvement in the performance is verified suggesting the right direction this thesis has taken.
To sum up, the main contributions of this thesis are:

- Pre-processing of experimental load and weather dataset to avoid unreliable data and to check cross dependency among the available time series

- Developing an effective and computationally cheap algorithm for weather forecasting
- Developing a two-stage hybrid prediction system to predict the load consumed. The first stage is made by a non-linear model able to predict complex dynamics and the second stage employs a linear model to catch the remaining dynamics and on which is possible to compute guaranteed error bounds
- Clearly comparing the performances of the proposed model with the state-of-the-art

Chapter 2

Problem overview

The problem addressed here refers to the 7-day-ahead forecasting for energy consumption of a non-residential building. The forecasting is intended to be performed on the basis of the two previous weeks of data. Energy consumption data are taken from the three-floor ABB office building located in Bergamo, Italy. The building hosts around 200 employees and includes elevators, electric cooling and heating systems, lighting systems, kitchen, UPS, fridges, computers, printers, electric car charging station and different electronic laboratories and electronic tools. Apart from load consumption time series, some local meteorological data are available, in particular temperature, solar radiation, relative humidity and average and maximum wind speed.

2.1 Dataset description

2.1.1 Load dataset

Data are recorded by various sensors spread around the building. The sensors record the absorbed electricity in [W] in different points and for different floors, at a frequency of one sample every 15 minutes. They store the maximum, the minimum and the average value obtained in the last quarter of hour as well as real, reactive and apparent electric power of elevators, air-conditioning, heaters, printers, car charger, ventilation system, servers and others. Data are collected through ABB API (Application Programming Interface). The API is able to deliver no more than one day of electric data. For this, to efficiently gather all the values, Python scripts have been written. The script creates a single **.json* database which is then elaborated by a second Python script to convert it into a more manageable **.csv* database. Afterwards, data are fed into *Microsoft Excel* to clean the dividers among columns with the aim of using a good formatted database to give to *MATLAB*. *MATLAB* is the chosen application for the implementation of the algorithms. Along with it some Add-Ons are used, in particular Deep Learning Toolbox for the training of Neural Networks. To keep the analysis as much general as possible the real average power consumption of the entire building is considered. Load data are available from 06/02/2017 up until 17/03/2020 but in between there are vectors of missing samples, mainly due to maintenance and setting operations.
As a result of the sampling frequency, each day has a total of 96 samples, one every

15 minutes. As mentioned above, the aim is to forecast the next 7 days of electricity load having 14 days of available data. This means that the training dataset is formed by 1344 samples and the test dataset 672 samples. From now on a general observed load data will be expressed as:

$$\tilde{y}(i,j) = c, j = 1, \ldots, 96 \tag{2.1}$$

where j indicates the moment within the day, i refers to the day considered and c is the mean electricity load consumed in the previous quarter of hour. An example of three-week-long load time series is shown in Figure 2.1.

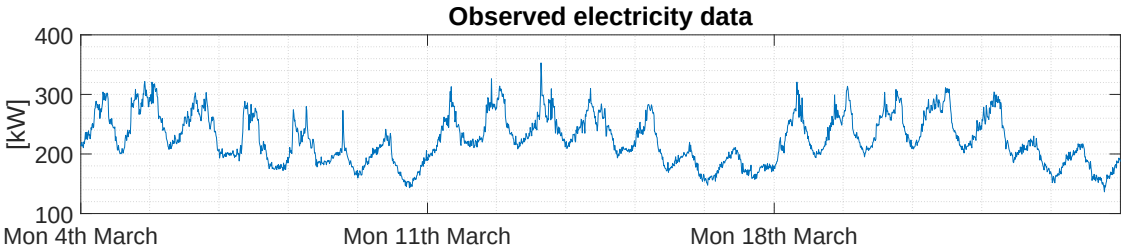


Figure 2.1. Example of a full three weeks electricity measurement from Mon 4 March 2019 to Sun 24 March 2019.

2.1.2 Weather dataset

Some weather data from a close-by meteorological station is available. Compared to load data, weather data are recorded every 10 minutes. 5 different time series describing the trend of weather are available: temperature [°C], the average global radiation [$\mathrm{W\,m^{-2}}$], the average humidity [%], the average wind speed [$\mathrm{m\,s^{-1}}$] and the peak wind speed [$\mathrm{m\,s^{-1}}$] of the past ten minutes. In order to match the same sampling frequency between weather and load consumption data, a linear interpolation described in Section 2.4.2 is performed. Weather data is available between 01/02/2017 and 15/04/2019 and, just like load, some unreliable samples of data are present. In Figure 2.2 a three-week long example of weather data is depicted. From top to bottom the plots represent temperature, global radiation, relative humidity, average wind speed and maximum wind speed evolution.

2.1.3 Time input

The passing of time is an intrinsic feature not described by the data at our disposal. This characteristic must be integrated [39] since it increases the final forecasting performances. For our analysis different type of such an input and their combinations are tested after being described in Section 4.3. The type of time series used are named *Daytype1*, *Daytype2*, *Time* and *Fictitious Input (FI)*. They exploits different characteristics and are able to describe time evolution in different ways.

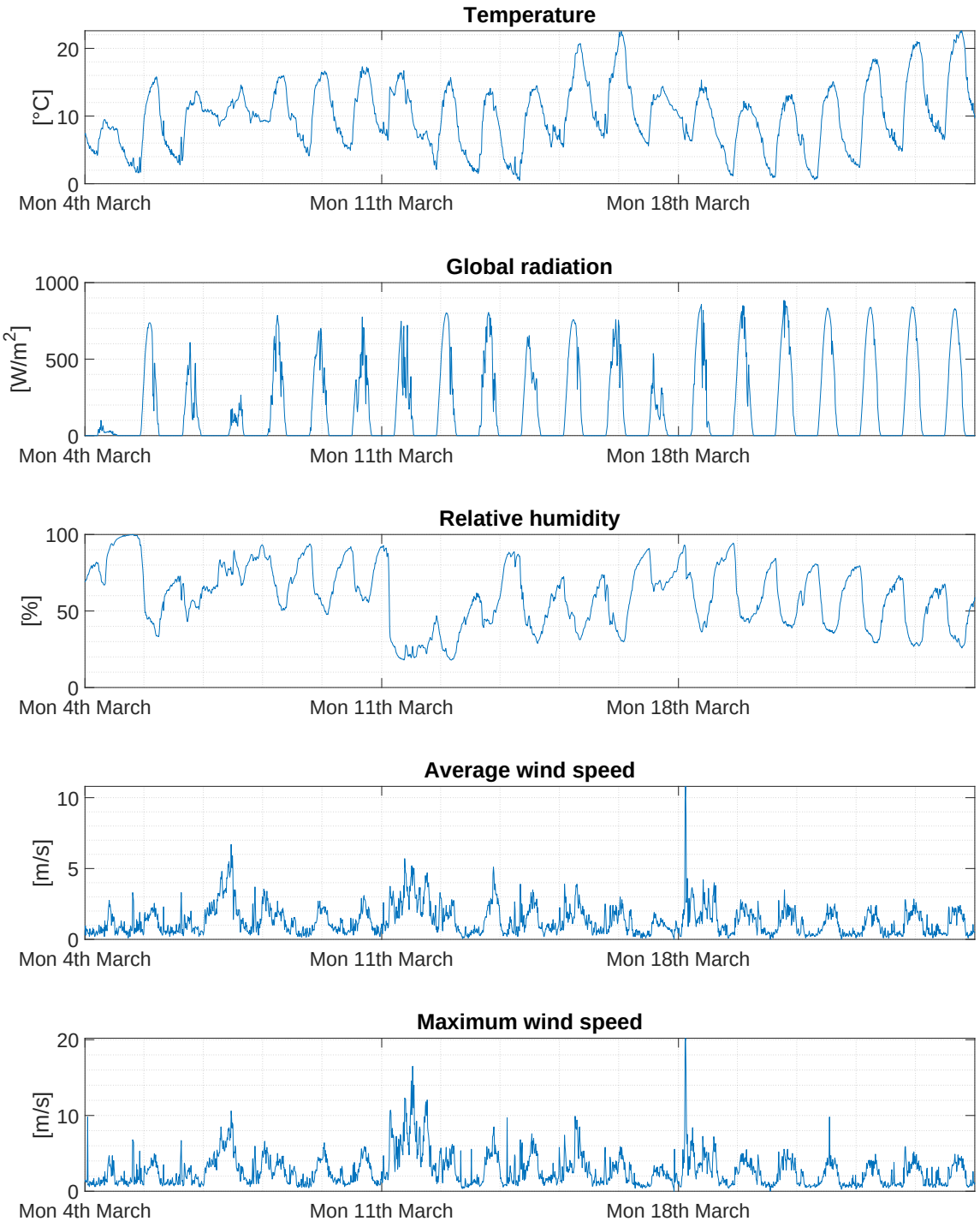


Figure 2.2. Example of a full three weeks weather measurement from Mon 4 March 2019 to Sun 24 March 2019. From top to bottom temperature, global radiation, relative humidity, average wind speed and maximum wind speed

2.2 Problem formulation

The final forecasting routine is carried out with a combination of non-linear and linear models in order to catch different dynamics in the time series. Preliminary the weather model are used to perform weather time series prediction subsequently used in the load consumption models. At first the non-linear model, made by Long Short Term Memory (LSTM) Neural Network, is used to capture strong non-linear behaviour and cross dynamics among inputs and the time series to be predicted. Afterwards, the residual obtained is used for the identification of an ARX model that will learn the residual between the neural network and the real process. Lastly, the resulting load prediction will be the sum of the LSTM and ARX results. A complete block diagram representing all the information flow is shown in Figure 2.3.

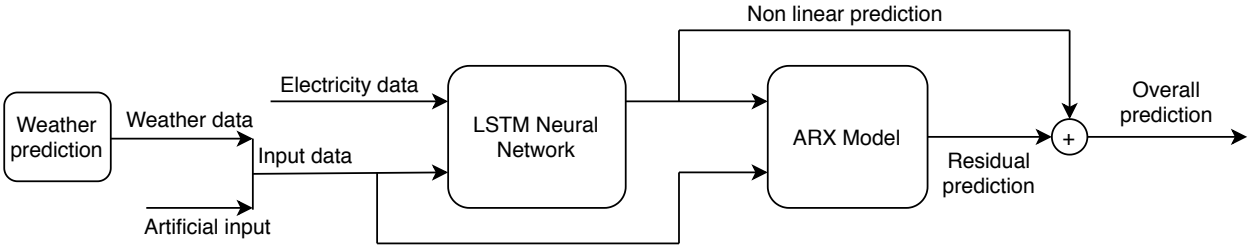


Figure 2.3. Block diagram of the final proposed prediction approach. At first, the weather prediction is run with a proper model. Then the load prediction routine starts: first data are fed into LSTM block that provides the non-linear prediction. Afterwards the obtained residual is used to identify an ARX model that will predict the future residual. The final forecasted time series will be the sum of the LSTM and ARX predictions

In both cases, used time input is customized according to the type of model considered. LSTM model will be fed with simpler time input data since its performances does not significantly increase with more complex input. On the other hand, ARX model, which is simpler and with fewer parameters than LSTM, will be fed with the complete time passing input since its performance has a remarkable improvement.
A key point of the analysis will be the study of a prediction approach for the weather input that considerably influences the final load prediction. The considered models are:

LSTM Neural Network the same model used for load prediction is employed but after few attempts this solution is left due to low prediction accuracy caused by the overfitting of the model

Feed-forward Neural Network inspired by [40]. A one layer Feed-forward Neural Network followed by a moving average smoothing filter with the aim of predicting day by day weather data is employed and described in Section 3.1.2

ARX model to capture the periodicity of such a series, in the end ARX model will not be used since it gives lower forecasting performances compared to the other models

Naive approaches two methods based on clear periodicity (see Figure 2.2) of weather time series are developed. The approach is described in Section 3.1.1 and derives from graphical observation and consideration of the time series.

After a specific analysis of their forecasting performances, weather models are tested together with the overall model. The result will be then compared to the one obtained with the real weather curve as input.
The final load prediction performances are compared to the prediction obtained using only LSTM Neural Network model and ARX model. At the beginning, preliminary analyses of the data are executed. These analyses will be performed in Chapter 2.3 and are necessary to understand which kind of correlation could be exploited. Different preprocessing techniques are tested without a real improvement in the overall results. In particular decomposition of the input time series both in time domain like STL [28] and in frequency domain like EMD has been tried. Data augmentation performed with simple time and frequency domain techniques and with decomposition method [30] does not lead to better forecasting performances. Different non-linear models are used embedding Convolutional Neural Network before the LSTM Neural Network with the aim of better catching symmetries and features [19].

2.3 Dataset correlation analysis

2.3.1 Weather-load data correlation

Before feeding all the weather data into the models a proper way to proceed is to further investigate whether weather input gives additional information and can help in improving forecasting performances or not. Scatter plots of weather data vs load data can help in that. In Figure 2.4 is possible to see a clear non linear dependence of load with respect to temperature. Precisely the load consumption increases for high and low value of temperature. This dependency could be explained in the usage of heating/cooling systems when the ambient temperature is too high or too low. Concerning the other weather variables, no clear dependency is predictable from the plots but cross dependency are not captured by this kind of analysis. Said that, different combinations of weather input are tested in simple LSTM models with the aim of understanding if their usage improves the forecasting performance. The plotted data are taken from the last year of available data, utilizing 16 uniformly distributed samples per day to catch the dependencies keeping the image clear and readable.

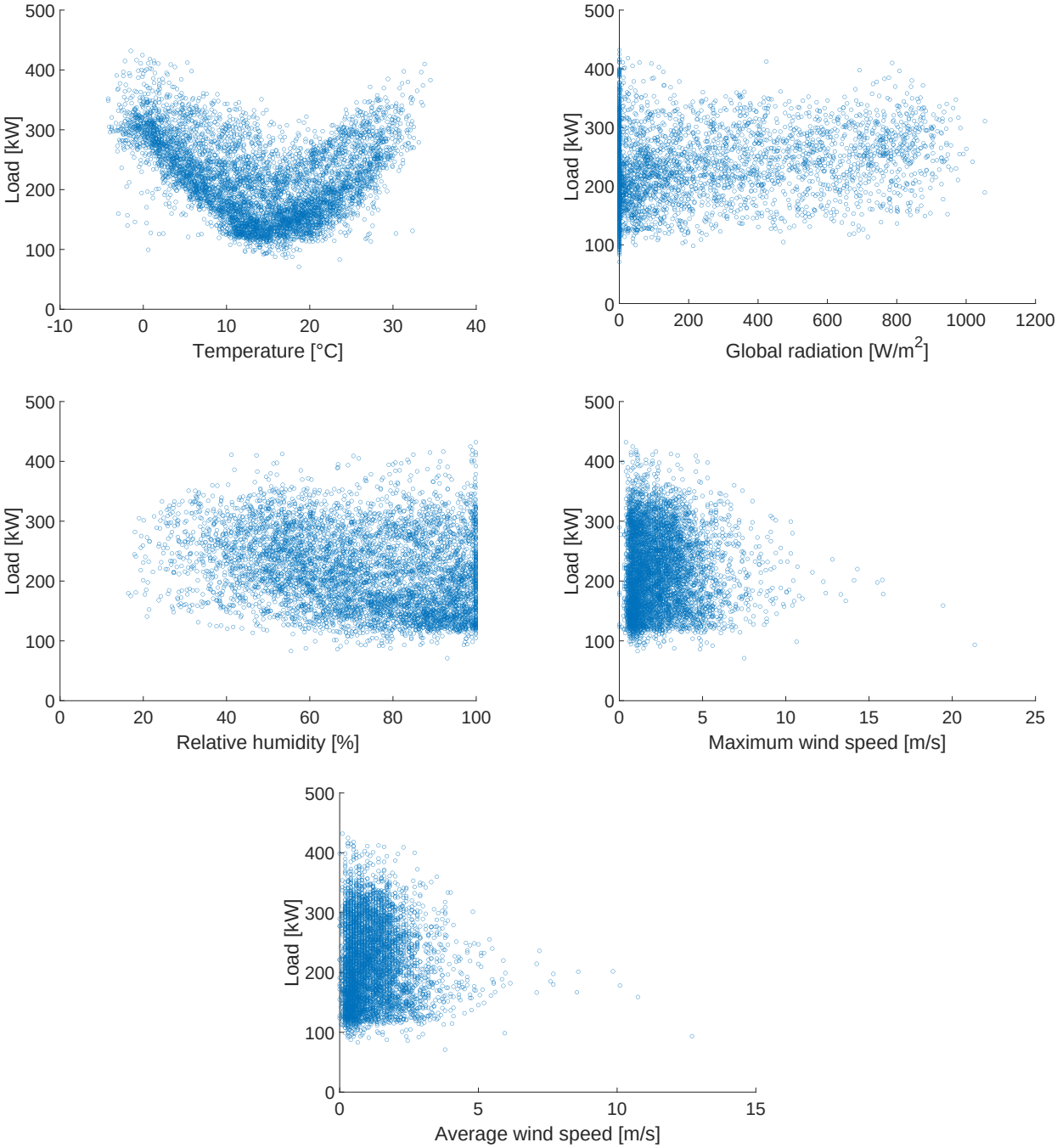


Figure 2.4. Scatter plots of weather variables vs load power. Up-left: Temperature. Up-right: Global Radiation. Middle-left: Relative Humidity. Middle-right: Maximum wind speed. Bottom: Average wind speed.

2.3.2 Load data characteristics correlation

Similarly to weather, it is possible to deeper investigate correlation among electricity power consumption data and some of its key features. In particular:

- $y_{i-1}^{max} = max(y(i-1,1),\ldots,y(i-1,96))$: peak load of the previous day
- $y_{i-7}^{max} = max(y(i-7,1),\ldots,y(i-7,96))$: peak load of the same day in the previous week
- $y_{i-1}^{min} = min(y(i-1,1),\ldots,y(i-1,96))$: minimum load of the previous day
- $y_{i-7}^{min} = min(y(i-7,1),\ldots,y(i-7,96))$: minimum load of the same day in the previous week
- $\overline{y}_{i-1} = \frac{1}{96}\sum_{j=1}^{96} y(i-1,j)$: average load of the previous day
- $\overline{y}_{i-7} = \frac{1}{96}\sum_{j=1}^{96} y(i-7,j)$: average load of the same day in the previous week

In Figure 2.5, as example, the scatter plots of $\overline{y}$ are reported being the most correlated data. Furthermore, it is possible to cluster the previous data between working and non working days in order to improve the correlation.

2.4 Dataset pre-processing

Load data are downloaded through ABB API, while weather data are collected from a close-by meteorological station. For both datasets there were missing samples along the whole dataset. In particular, for load data, as already mentioned, all the load value lower than 65 [kW] are not reliable, whereas for weather time series the missing values are presented as -999. In addiction, the two datasets are characterized by two different sampling frequencies. Load data are collected every 15 minutes while weather data every 10 minutes.

2.4.1 Missing data

Electricity data

No data cleaning is applied to the electricity time series. This choice is made in order to avoid running the prediction algorithm on hypnotized values that would surely lead to incorrect identification parameters and to untrustworthy forecasting performances. The choice of the training and testing datasets is crucial to obtain a reliable result. To do so, the starting and final dates that do not contain missing load dataset in between are computed and the datasets on which the model is run does not contain unreliable electricity load samples.

Weather data

Missing weather variables are, instead, simply linearly interpolated. Moreover, if the weather time series contains a vector of missing data longer than 8 hours it is not chosen as part of the testing dataset.

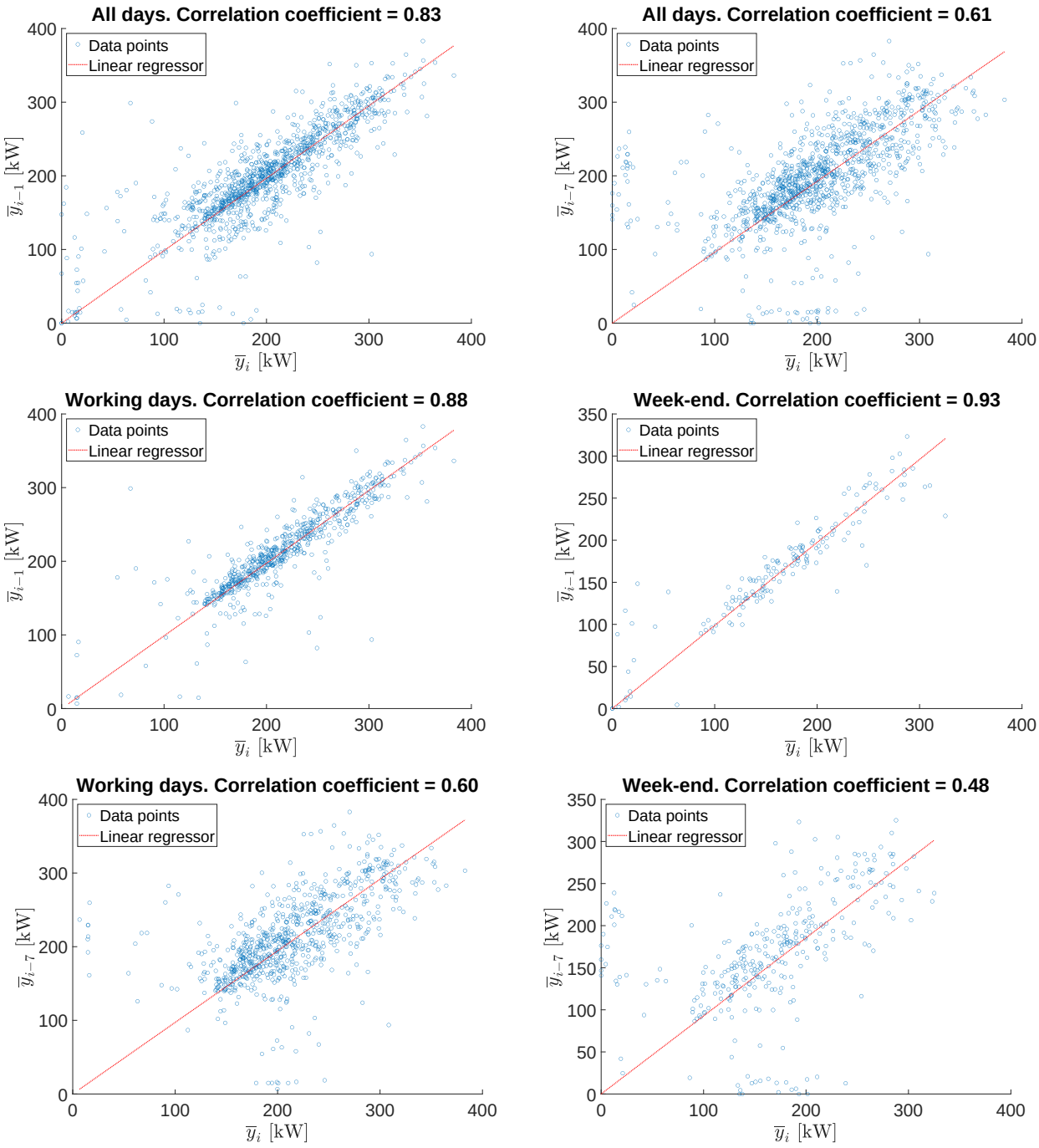


Figure 2.5. Scatter plots of load average power. Left: average value of day i with respect to day $i-1$. Right: average value of day i with respect to day $i-7$. Up: value with respect to the previous day. Middle: value of working days with respect to the previous working day. Bottom: value of weekends with respect to the previous weekend

2.4.2 Weather dataset adaptation

At first, all the meteorological data are adapted to the load data frequency. As the sampling time is small with respect to the dynamic change of weather, especially temperature and humidity, the missing meteorological data are retrieved by simple linear interpolation. Let's consider $W(i) = [W(i), \ldots, W(i+6)]$ $i = 1, 7, 13, 19, \ldots$ as a one-hour-long weather vector with 6 samples, the new vector $W_{new}(i) = [W_{new}(i), \ldots, W_{new}(i+6)]$ $i = 1, 7, 13, 19, \ldots$ sampled every 15 minutes is built as:

- $W_{new}(i) = W(i)$
- $W_{new}(i+1) = \frac{W(i+1)+W(i+2)}{2}$
- $W_{new}(i+2) = \frac{W(i+3)+W(i+4)}{2}$
- $W_{new}(i+3) = W(i+5)$

2.4.3 Normalization

All the data are normalised to zero mean μ and unitary standard deviation σ. This is performed before the identification of model parameters to prevent the wrong identification due to order of quantity issues. Both μ and σ are computed using the training dataset. The final normalized time series is computed as follows:

$$x_{norm} = \frac{x - \mu}{\sigma} \tag{2.2}$$

where x_{norm} is the normalized time series while x is the original one. At the end of the model parameters identification and the forecasting routine the obtained predicted time series is de-normalized using the same mean μ and standard deviation σ according to the straightforward inverse formula:

$$x = (x_{norm} * \sigma) + \mu \tag{2.3}$$

2.4.4 Data correlation results

Weather and time evolution data

Due to the fact that gradient descent procedures for neural network has some stochastically chosen parameters, the same identification and prediction procedures is repeated 7 times to have a more reliable result. Table 2.1 shows the result of the simulation. The first number represents the average MAPE obtained from the 7 simulation, while, between brackets, the standard deviation is shown. T, R, H indicates if temperature, global radiation and humidity are used, respectively. From MAPE values it is possible to state that the best combination of time type input is surely $DT2/Time$ but $FI_{1:14}$ is also tested. Therefore, only temperature and radiation are taken as input time series with the aim of keeping the model simple.

T	**R**	**H**	*No time data*	$DT1$	$DT2$	$Time$	$DT1/Time$	$DT2/Time$	$FI_{1,7}$	$FI_{1:14}$
–	–	–	17.45(0.75)	12.05(0.35)	68.08(54.40)	15.48(2.96)	7.08(0.65)	8.16(1.73)	**6.69(0.52)**	7.34(0.57)
✓	✓	✓	8.75(0.74)	5.53(0.54)	30.19(11.64)	12.04(2.67)	**4.95(0.56)**	5.11(0.14)	5.25(0.21)	5.68(0.27)
✓	–	–	10.87(1.20)	6.74(0.56)	27.51(24.15)	13.71(2.61)	5.71(1.32)	**5.06(0.35)**	5.16(0.24)	5.46(0.15)
–	✓	✓	14.34(2.86)	6.41(1.07)	43.12(17.57)	16.38(1.70)	8.75(1.41)	**6.67(1.05)**	30.58(23.59)	6.84(0.39)
✓	✓	–	9.79(0.94)	5.35(0.77)	12.35(14.76)	14.12(1.80)	5.38(0.65)	**5.14(0.73)**	5.47(0.75)	5.47(0.18)
✓	–	✓	11.17(2.10)	7.58(1.31)	25.43(18.34)	12.61(2.06)	**4.80(0.44)**	4.89(0.37)	5.48(0.28)	5.57(0.15)
–	✓	–	12.93(0.91)	5.89(0.46)	50.47(28.48)	14.72(0.80)	6.44(0.93)	7.17(0.50)	14.15(13.38)	**5.48(0.29)**
–	–	✓	16.43(1.08)	10.13(0.39)	45.31(34.44)	16.11(1.24)	7.16(1.58)	**6.51(0.47)**	17.43(27.37)	7.13(0.42)

Table 2.1. Test result with LSTM NN only. Average MAPE values over 7 simulations of load forecasting performance for different weather input. The first value represent the average MAPE while the value between brackets is the standard deviation of the distribution. In **bold** are highlighted the value with the lower average MAPE. T, R and H respectively state whether temperature, radiation and humidity input signal is used. From left to right the columns include: model without artificial input, model with *Daytype1* input, model with *Daytype2* input, model with *Time* input, model with a combination of *Daytype1* and *Time* input, model with a combination of *Daytype2* and *Time* input, model with a Fictitious Input with only daily and weekly periodic signal, model with Fictitious Input with all harmonics

Feature load data

Subsequently the same test is performed with the feature input described in Section 2.3.2. MAPEs by using the feature of the previous day are shown in Table 2.2, while the one obtained by clustering the feature of the days in working and non working group are shown in Table 2.3. Since this new set-up does not introduce new information and prediction performances are approximately the same, feature load data will not be used in the final model.

Max	**Min**	**Avg**	$DT2/Time$			$FI_{1:14}$		
			TRH	TR	TH	T	TR	R
✓	–	–	**4.74(0.55)**	14.51(23.67)	9.36(5.69)	5.92(0.09)	**5.50(0.20)**	6.59(0.27)
–	✓	–	5.58(0.62)	**4.81(0.71)**	7.47(1.82)	**5.40(0.11)**	5.59(0.27)	6.28(0.38)
–	–	✓	6.10(1.19)	**4.77(0.53)**	8.12(1.77)	**5.55(0.04)**	5.56(0.12)	6.11(0.10)
✓	✓	–	6.86(3.33)	**5.08(0.42)**	8.05(4.27)	5.85(0.22)	**5.43(0.17)**	7.17(0.31)
✓	–	✓	**5.16(0.52)**	5.30(0.48)	5.19(0.63)	6.10(0.28)	**5.29(0.11)**	7.26(0.27)
–	✓	✓	**6.41(1.33)**	30.23(40.33)	8.42(2.73)	**5.38(0.15)**	5.43(0.19)	6.53(0.13)
✓	✓	✓	8.05(5.35)	**5.30(0.45)**	9.65(2.17)	5.84(0.15)	**5.60(0.16)**	7.34(0.31)

Table 2.2. Test result with LSTM NN only. Average MAPE values along 7 simulations of load forecasting performance for new feature data input. The first value represent the average MAPE while the value between brackets is the standard deviation of the distribution. In **bold** are highlighted the value with the lower average MAPE. Max, Min and Avg respectively state whether the maximum, the minimum and the average load of the day before is used as input signal. From left to right the columns include: model with a combination of *Daytype2* and *Time* input and model with $FI_{1:14}$ input. T, R and H states whether temperature, radiation and humidity are used as input signal.

Max	**Min**	**Avg**	$DT2/Time$			$FI_{1:14}$		
			TRH	TR	TH	T	TR	R
✓	–	–	**8.81(0.79)**	13.97(6.19)	10.96(2.8)	8.42(0.65)	**7.60(0.50)**	10.32(1.14)
–	✓	–	7.23(1.16)	**6.28(0.97)**	8.21(2.38)	6.45(0.53)	**6.31(0.25)**	6.48(0.21)
–	–	✓	**5.75(0.71)**	5.75(0.90)	7.11(1.70)	7.11(0.10)	**6.33(0.13)**	7.05(0.16)
✓	✓	–	15.71(2.01)	24.34(2.930)	**15.26(3.30)**	8.89(0.52)	**8.87(0.73)**	10.19(0.96)
✓	–	✓	18.57(17.09)	17.73(16.55)	**11.51(2.10)**	8.64(0.38)	**7.87(0.65)**	9.55(0.72)
–	✓	✓	**10.57(2.28)**	20.49(27.88)	20.60(9.59)	6.75(0.50)	**6.67(0.19)**	6.72(0.27)
✓	✓	✓	**15.24(3.26)**	28.34(10.18)	15.40(2.77)	9.14(0.76)	**8.26(0.67)**	11.12(0.87)

Table 2.3. Test result with LSTM NN only. Average MAPE values along 7 simulations of load forecasting performance for new feature data input. The first value represent the average MAPE while the value between brackets is the standard deviation of the distribution. In **bold** are highlighted the value with the lower average MAPE. Max, Min and Avg respectively state whether the maximum, the minimum and the average load of the same-type day before is used as input signal. From left to right the columns include: model with a combination of *Daytype2* and *Time* input and model with $FI_{1:14}$ input. T, R and H states whether temperature, radiation and humidity are used as input signal.

Chapter 3

Weather prediction model

The importance of having a good forecasting model for weather variables is a key point discovered during the study conducted in this thesis. A big amount of time and different approaches have been tried to achieve good prediction performances [40]. Hereafter, the two most successful methodologies consisting of a Feed-forward neural network and a Naive intuitive approach are presented. The Naive approaches are formulates based on the periodicity these time series are affected. In Table 3.1 the MAPE values of the presented forecast approach are shown. It is clear that the Naive approaches has the best forecasting performances. Being easily implementable and computational efficient, the *First samples* approach is chosen for the overall electricity forecasting

Week	**Temperature**			**Radiation**			
	FS	DoS	$FFNN$	FS	DoS	$FFNN$	TM
1	**6.68**	7.94	225.57	40.90	40.90	100.30	**40.18**
2	**9.06**	9.17	78.81	**68.90**	**68.90**	361.38	91.21
3	9.45	**8.52**	174.15	**43.13**	**43.13**	126.71	87.40
4	**8.73**	8.97	134.65	**25.92**	**25.92**	95.25	45.45
5	**6.78**	8.54	126.27	13.32	13.32	71.49	**10.01**
6	**12.23**	13.48	68.99	**17.35**	**17.35**	71.96	28.98
7	**7.29**	7.97	574.98	**52.95**	**52.95**	558.07	218.25
8	**7.14**	8.60	437.05	**90.35**	**90.35**	212.25	187.31
9	**22.47**	32.87	235.48	**7.24**	**7.24**	29.90	15.99

Table 3.1. MAPE values of one-week-ahead temperature and radiation forecasting. In **bold** are highlighted the value with the lower MAPE. FS state the *First Samples* method is used, Dos refers to the *Difference of Samples*, FFNN is the *Feed-forward Neural Network* and TM is the *Time method* developed only for radiation

3.1 Temperature forecasting

Between temperature and radiation the first one turned out to be the most influencing weather input. Little discrepancies between observed and predicted temperature

values lead to worse forecasting performances of the overall model. This is why big attention must be given to the prediction routine of this exogenous input.

3.1.1 Naive approach

It is possible to observe the periodicity of the time series from the shape of temperature depicted in Figure 2.2. In particular, we can notice a strong daily periodic behaviour and the peak value reached during the day seems to depend on the first samples of the day itself or on the difference between the first and the last sample of the previous day. Based of this observation the, Naive approaches are formulated and two different solutions are tried. Supposing to denote with i the last day of available data, the two proposed method are formulated as follow:

***First samples* method** basically the temperature curve is tried to be predicted knowing the first n_{FS} samples of the day. Considering:

$$\begin{aligned} \tilde{Y}(i+1, n_{FS}) &= [y(i+1,1), y(i+1,2), \ldots, y(i+1, n_{FS})] \\ \tilde{Y}_{avg,i+1} &= \frac{1}{n_{FS}} \sum_{j=1}^{n_{FS}} y(i+1, j) \end{aligned} \tag{3.1}$$

where $\tilde{Y}_{avg}(i+1, n_{FS})$ is the average value of temperature of the first n_{FS} samples. The vector $\hat{Y}(i+1, \cdot)$ containing the temperature predicted curve of day $i+1$ is constructed as:

$$\hat{Y}(i+1, \cdot) = [\tilde{Y}(i+1, n_{FS})\ \tilde{Y}(i, 96 - n_{FS}) + \tilde{Y}_{avg,i+1}] \tag{3.2}$$

where:

$$\tilde{Y}(i, 96 - n_{FS}) = [y(i, n_{FS}+1), y(i, n_{FS}+2), \ldots, y(i, 96)]$$

Difference of samples on the other hand the temperature curve is tried to be predicted considering the difference of values between the last and the first sample of day i. More precisely let us consider:

$$d = y(i, 96) - y(i, 1)$$

the forecasted temperature values for day $i+1$ are:

$$\hat{Y}(i+1, \cdot) = \tilde{Y}(i, \cdot) + d \tag{3.3}$$

The prediction result applying this algorithms are shown in Figure 3.1.

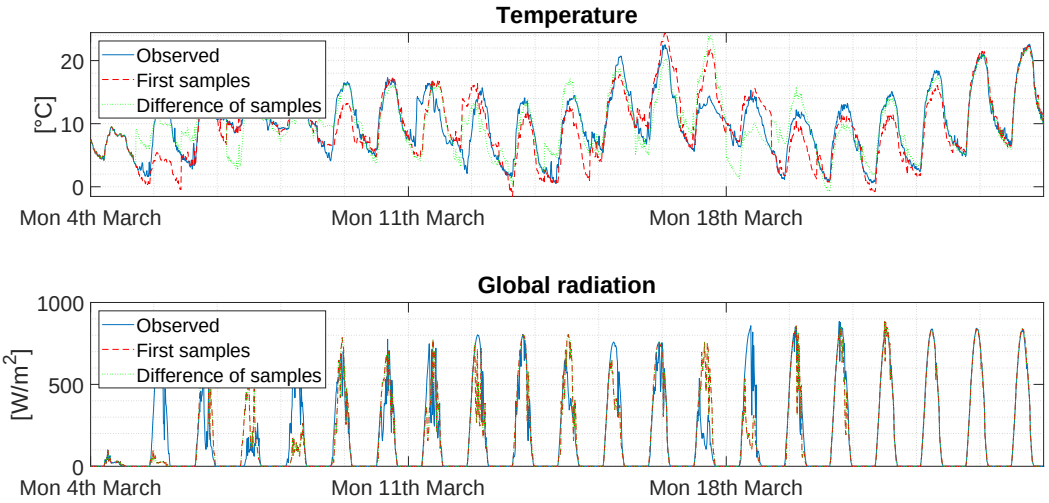


Figure 3.1. Naive forecasting approach for temperature (above) and global radiation (below). From temperature plot it is possible to see that the model better behave when the time series evolution has slow dynamic and has a regular trend like in the predicted Saturday and Sunday. On the other hand the prediction has worse accuracy when the temperature curve of the known days has an irregular shape; like in the firsts predicted day of the time series. While for radiation the two models perfectly behave when the curve of two subsequent day are identical.

3.1.2 Feed-forward Neural Network

This prediction approach is inspired by [40], which considers a sampling period of 1 hour. The best prediction model for temperature is made by a Feed-forward neural network that takes as input the 24 observed values of day i and the forecasted maximum and minimum temperature value of the $i+1$ day (obtainable through a meteorological service). The obtained forecasted value are then smoothed through a third-order centred moving average filter. A similar model is used with some adaptation to better fit our forecasting framework. In particular, our model will be a feed-forward neural network with 1 layer of fully connected neurons with 96 input and 96 output. Due to the fact that no forecasted maximum and minimum values of the next days are available, this information will not be given to the model. In Figure 3.2 the results are depicted. The predicted load fits too much the training dataset leading to worse prediction performances than the previously described methods.
In Table 3.1 the MAPEs obtained with the described prediction method are shown. It is clear that the Naive approaches lead to superior results with respect to Feed-forward neural network. In particular, *First sample* has the best forecasting performances and will be employed for the proposed prediction method.

3.2 Radiation forecasting

As it is possible to see from Figure 2.2 also global radiation has a strong periodic behaviour. This suggests to use the same forecasting techniques proposed for temperature. Moreover, after making different simulations scaling the shape of radiation

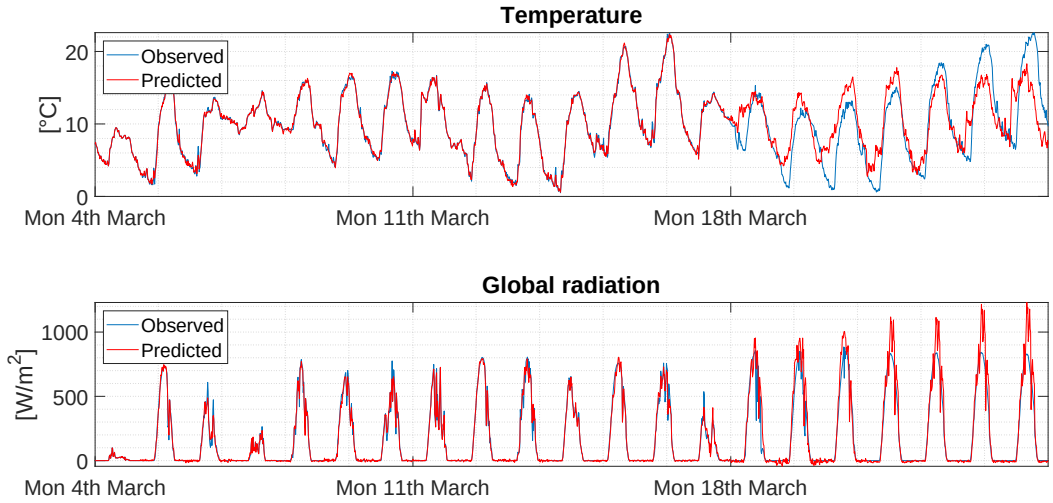


Figure 3.2. Forecasting approach for temperature (above) and global radiation (below) with feed-forward neural network

time series, prediction results suggest that the only information carried by this data is the beginning and the finishing of "solar" day. Any change in the peak does not affect forecasting error. In light of this the time based prediction method is here presented. Apart from that it is worth to say that *First samples* and *Difference of samples* for radiation prediction coincide since the first and last samples of each day are always equal to zero.

Time based method

Based on the relative motion of Earth with respect to Sun, lots of web services offer the opportunity to gather sunrise and sunset hours for any given date. According to this consideration, it is reasonable to assert that global radiation is zero anywhere not in between sunrise and sunset time. Looking at the shape of the data during the day it is possible to note that it is almost the same for all days, except from a cloudy or too sunny day. Furthermore, according to what has been said at the beginning of the section, knowing the precise peak occurring at that day has no value. Therefore the time based method approximates the global radiation with the average shape of the known days making it start at sunrise and finish at sunset hour. In Figure 3.3 an example is shown.

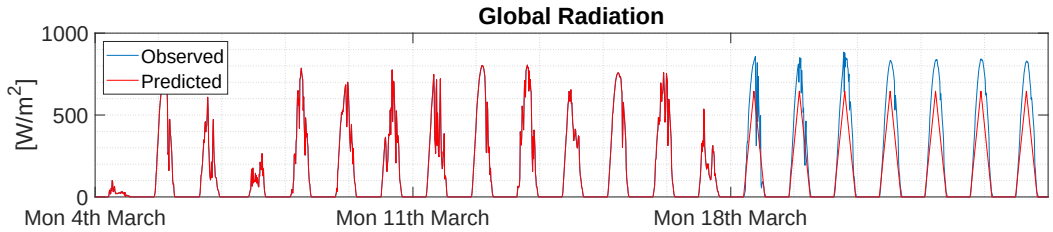


Figure 3.3. Time based method forecasting for radiation

Chapter 4

Load forecasting model

In this chapter the complete electricity forecasting routine is presented. Initially, the LSTM Neural Network based approach is used with the aim of approximating all the non-linear behaviours of the time series. The prediction residuals resulting from LSTM regression are then estimated by a linear ARX model. The final electricity power consumption forecast is given by the sum of the LSTM and ARX predictions.

4.1 Final prediction model

An anticipation of the final prediction model is here presented. The final prediction routine is a composition of two forecasting models. Figure 4.1 depicts a block diagram explaining the signals flow of the proposed method during the training phase. Input data is a vector containing both weather estimated as described in Chapter 3 and Fictitious Input and it is fed into LSTM and ARX. On the other hand, electricity data is used to train LSTM Neural Network that is therefore able to predict the future load curve. The residual time series coming from LSTM block is then used to identify an ARX linear model that will be able to predict the residuals for the time series. The LSTM block is responsible for the non-linear prediction of the signal and the ARX model aims to predict the linear residual dynamics.

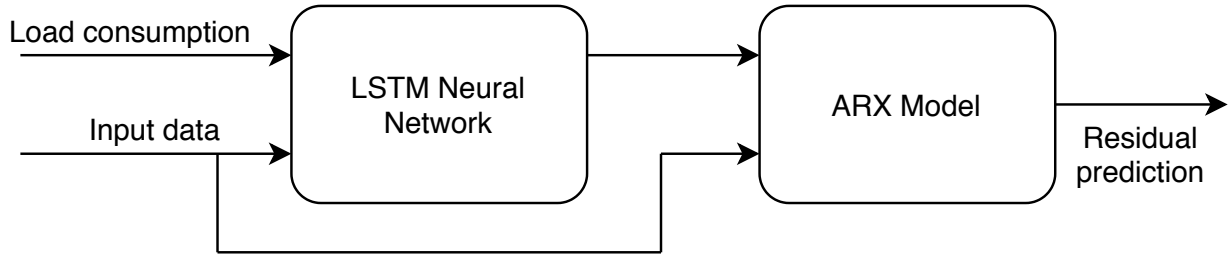


Figure 4.1. Information flow of the proposed prediction set-up. Input data gathers all the weather and time inputs.

where "Input data" contains both weather time series and Fictitious Input. Both LSTM and ARX uses the same input variables: temperature, radiation and Fictitious Input. Graphical results can be appreciated in figures 5.1 and 5.2 for real weather input and in figures 5.3 and 5.4 for model weather input.

4.2 LSTM unit

Proposed in late 90's, Long Short-Term Memory (LSTM) is an innovative architecture of Recurrent Neural Network. Its main advantage is that it solves the vanishing gradient problem that affected all the previous networks. LSTM unit is composed by a cell, an input gate, an output gate and a forget gate. The cell is the memory of the entire unit and the gates manage the flow of informations in and out the unit. It is a powerful tool because it can not only process single data point but also sequence of data. LSTM perfectly suits many artificial intelligence tasks such as processing electronic data (EDP), classification and prediction. An example of LSTM unit is shown in Figure 4.2.

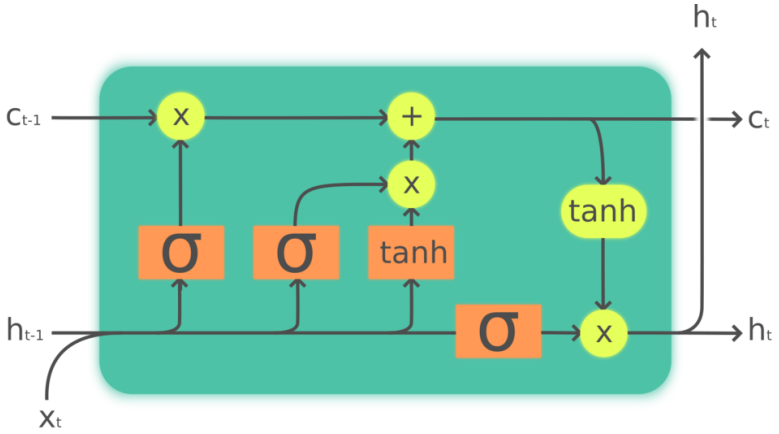


Figure 4.2. LSTM Cell

where x_t represents the input vector of the LSTM unit, h_t is the output or hidden state vector and c_t is the cell input activation vector. The green spots are point-wise operation while the orange box are the activation function, in particular σ is the logistic sigmoid function. The *input gate* has the role of controlling how much the new value will flow in the cell, *forget gate* controls how much information must remain in the cell in order to learn and *output gate* controls the extent to which the value of the cell is used to compute the output. The functions that puts into relationship the input and the output of a LSTM units are:

$$\begin{aligned} f_t &= \sigma_g\left(W_f x_t + U_f h_{t-1} + b_f\right) \\ i_t &= \sigma_g\left(W_i x_t + U_i h_t + b_i\right) \\ o_t &= \sigma_g\left(W_o x_t + U_o h_{t-1} + b_o\right) \\ \tilde{c}_t &= \sigma_c\left(W_c x_t + U_c h_{t-1} + b_c\right) \\ c_t &= f_t \circ c_t 1 + i_t \circ \tilde{c}_t \\ h_t &= o_t \circ \sigma_h\left(c_t\right) \end{aligned} \tag{4.1}$$

where the initial values are $c_0 = 0$ and $h_0 = 0$ and the operator $\circ$ denotes the Hadamard product (element-wise product). The subscript t indexes the time step. The combination of more units on the same layer and more layers lead to highly non-linear complex models whose number of parameters increases exponentially. An example of LSTM layer is shown in Figure 4.3. After performing simulations it comes

out that the best LSTM model for the aim of this study is a single-layer Neural Network with 96 hidden units followed by a drop-out layer with probability 50%. The drop-out layer has the function of avoid overfitting by randomly setting to zero the 50% of the LSTM layer parameters.

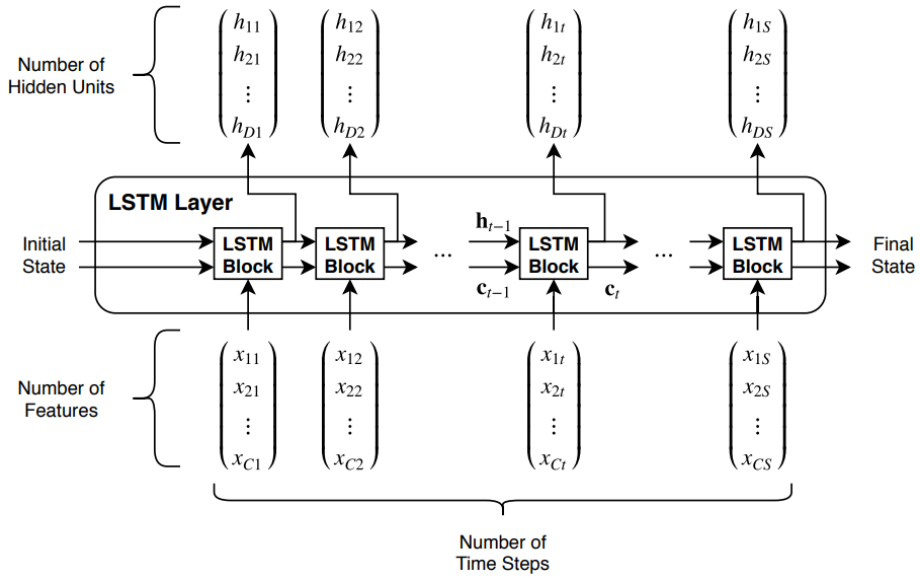


Figure 4.3. Diagram of a general LSTM layer

4.3 Time input

The evolution over time is an important information not directly carried by any of the exogenous input. Being load time series highly periodic, the introduction of a particular input that marks time passing lead to an increase of the forecasting performances of the model under study. Four of these types of time series and some of their combinations are tested and the performances are than evaluated. No prediction is needed for this input since it is all known a priori. Therefore, the models are composed by 2016 samples that cover the whole three weeks. The four models are:

Day type 1

DayType 1 is built in order to make simple distinction between working and non working days. For each working day a vector of zeros is used while for the non working day and for holidays a vector of ones is used.

Day type 2

Daytype 2 is built in order to make distinction for each day of the week. For all Mondays a vector on ones is used, for all Tuesdays a vector of twos is used and so on up to seven. For holiday, in that case, a vector of eights is used.

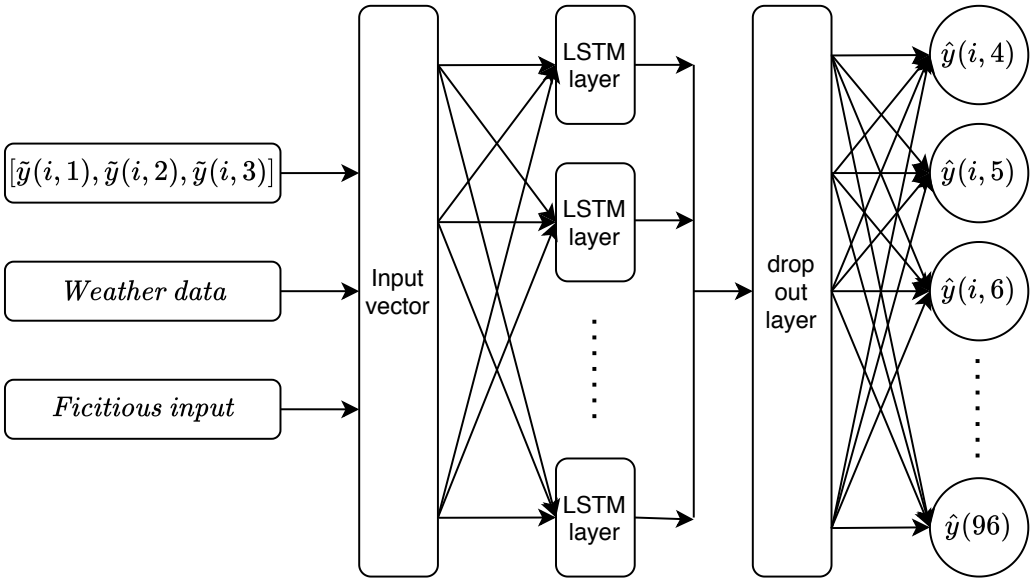


Figure 4.4. Flow of information of the proposed LSTM-based non-linear prediction

Time

Time vector is defined as a vector with values running from 1 to 96 for each time sample of the day and it is repeated 21 times. Its function is to empathize in which of the 96 discretized time interval of the day the model is working.

Fictitious input

From load plots in Figure 2.1 it is easy to notice different periodicity in the signal. From high frequency periodicity (e.g. daily) to low frequency periodicity (e.g. seasonally and yearly). Fictitious Input is an artificial signal made through the concatenation of sine and cosine waves built to catch all wanted periodicity. It is defined as:

$$u_{FI,i}(k) = \begin{bmatrix} \cos\left(\omega_1(k+96(i-1))\right) \\ \vdots \\ \cos\left(\omega_{n_\omega}(k+96(i-1))\right) \\ \sin\left(\omega_1(k+96(i-1))\right) \\ \vdots \\ \sin\left(\omega_{n_\omega}(k+96(i-1))\right) \end{bmatrix}, k = 1, \ldots, 96, i = 1, \ldots, 21$$

where n_ω is the number of considered frequencies a priori chosen and $\omega_j = 2\pi f_j$ is the corresponding harmonics. The parameter k runs from 1 to 96 since it expresses the evolution of the signal within each day, while index i runs from 1 to 21 so that a signal 3 weeks long is produced. Thus, the final matrix is $u_{FI} \in \mathbb{R}^{2n_\omega \cdot 2016}$. Being our train set 14 days long and our test set 7 days long, the presence of low frequency behaviours is excluded. Considering 4 samples every hour it is possible to define $f_j = \frac{j}{96 \cdot 7}$, than, choosing $n_\omega = 14$ and $j = 1, \ldots, 14$ the harmonics with periodicity between 7 days and 12 hours are considered. In Figure 4.5 is possible to appreciate some examples of Fictitious Input. The first plot on top shows two examples of FI

prior the identification and training phase, the example shows a daily and weekly periodic signal. Two FIs after the identification of the ARX model for load prediction is shown in the second plot. For both signals is possible to appreciate either a daily and a weekly periodicity that confirms our assumptions regarding the presence of multiple seasonality in the dataset. The third and last plot shows the FI signal after the identification for the ARX for residual prediction. This signal has harmonics with higher frequencies since lots of periodicity is caught already by LSTM and ARX tries to predict higher and higher frequencies.

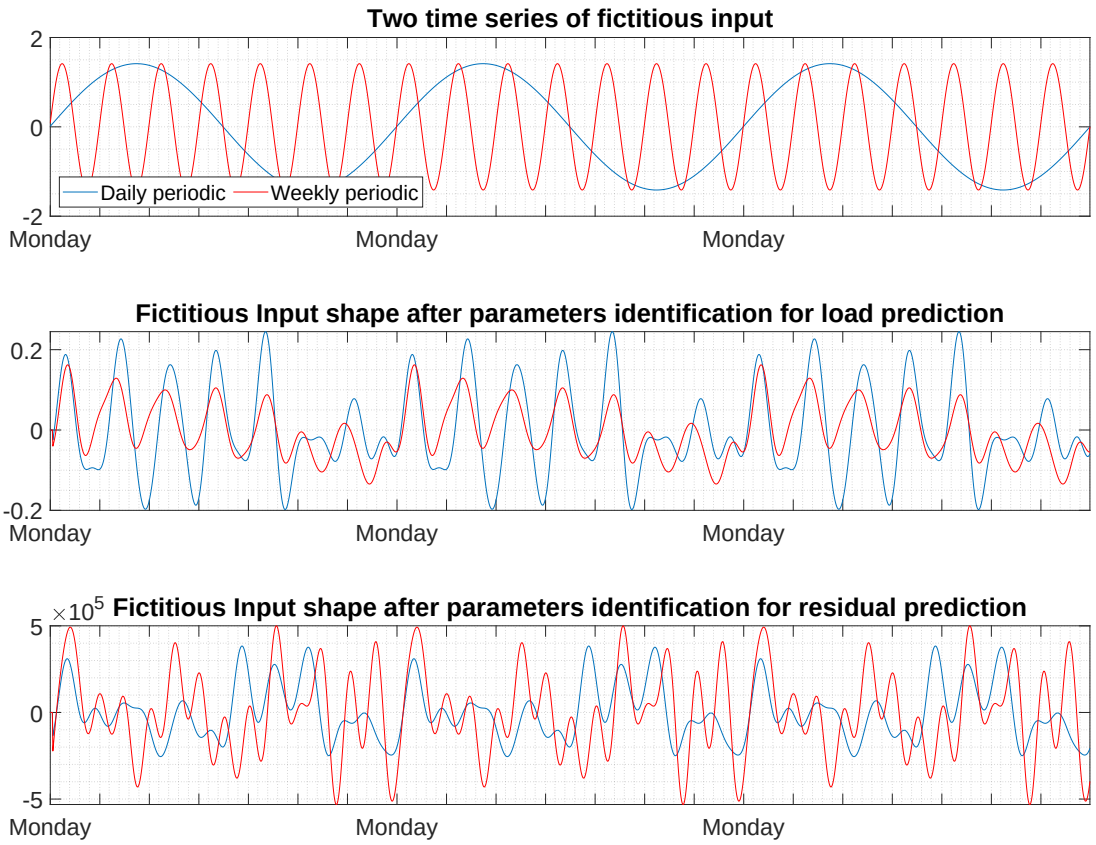


Figure 4.5. Fictitious Input (FI) time series example. From top to bottom: example of a daily and weekly periodic FI signal, Fictitious Input signal deriving from ARX identification for load prediction and Fictitious Input signal deriving from ARX identification for residual prediction

4.4 Non linear prediction model

As anticipated in Section 4.2 the chosen non-linear model is a Recurrent Neural Network with one layer made by 96 hidden units of LSTM cells. In order to best estimate the non-linear behaviour of the time series, different set-ups and configurations of the network are tested. As first attempt, a simple one-step ahead predictor through LSTM network is tried. The non-linear relation between input and output can be

expressed in the form:

$$\hat{y}(i+1) = f_{LSTM}(\tilde{y}(i), \theta_{LSTM}, x_{LSTM,0}, FI(i), \hat{W}(i)) \tag{4.2}$$

where $\hat{y}(i+1)$ is the one-step-ahead prediction, f_{LSTM} is the non-linear relationship given by the model, $\tilde{y}(i)$ is the observed value at time step i, θ is the LSTM parameters vector, $x_{LSTM,0}$ is the initial state of the cells, $FI(i)$ is the fictitious input and $\hat{W}(i)$ is the weather model input. Both the real observed value of weather and the predicted one are used with the purpose of understanding the effect of the weather model. Obviously for the validation phase $\tilde{y}(i)$ is substituted with $\hat{y}(i)$ because the observed value are not available. The multi-step-ahead prediction is obtained by iterating the one-step-head prediction model multiple times. Unfortunately, the results coming from this configuration are good only for some datasets. Moreover the model rapidly goes in over-fitting even with less LSTM cells and the prediction error increases rapidly meaning that a new configuration has to be considered.
Instead of reasoning time step by time step, for this new approach a day by day argument is tested. The main idea is to use the first $n_{L,y} = 3$ samples (occurring at 00:00, 00:15 and 00:30) of the day in order to predict the remaining 93. This time the non-linear relationship has slight modification from the previous one and can be expressed as:

$$\hat{Y}_j = f_{LSTM}(\tilde{Y}_{j,n_{L,y}}, \theta_{LSTM}, x_{LSTM,0}, FI_j(i), \hat{W}_j(i)) \tag{4.3}$$

in this case $\hat{Y}_j = [\hat{y}(4), \hat{y}(5), \ldots, \hat{y}(96)] \in \mathbb{R}^{93}$ is the vector containing the 93 values of the day j to be predicted, $\tilde{Y}_{j,n_{L,y}} = [\tilde{y}(1), \tilde{y}(2), \tilde{y}(3)] \in \mathbb{R}^3$ contains the first three values of the day j. Regarding $FI_j(i)$ and $\hat{W}_j(i)$ the whole values of the day are used and, as done with the first configuration, both observed weather value and predicted one are tried in order to understand the influences of the weather models. The optimization problem set-up is therefore to minimize the L2-norm loss function:

$$\hat{\theta} = \arg\min_{\theta} \sum_{j=1}^{14} \left(\tilde{Y}_j - \hat{Y}_j(\theta)\right)^2 \tag{4.4}$$

where the sum runs on the 14 training days. The vector $\tilde{Y}_j$ collects the observed values of day j and $\hat{Y}_j(\theta)$ gathers the predicted values of day j from the Neural Network. They take the following shape:

$$\tilde{Y}_j = [\tilde{y}_j(n_{L,y}+1), \ldots, \tilde{y}_j(96)]^T$$
$$\hat{Y}_j(\theta) = [\hat{y}_j(n_{L,y}+1) \ldots, \hat{y}_j(96)]^T$$

the LSTM Neural Network training is carried on with *Simulation Mode*. This means that at every iteration the values of vector $\hat{Y}_j(\theta)$ are computed by practically feeding the input into the Neural Network and collecting the outputs. Even though the training dataset for this configuration is quite small the model behave better, the fitting of the training dataset is good thus resulting in a more uniform residual that turns out to be more easily predictable with more clear dynamics left.

Adam optimizer

For the training of the LSTM Neural Network the Deep Learning Toolbox from MATLAB ® is used. Adaptive Moment Estimation (Adam) is employed as optimization algorithm [41]. It is an algorithm proposed in 2015 aiming at optimize stochastic objectives in high-dimensional parameter space and it is an evolution of classic Stochastic Gradient Descent. The classic Stochastic Gradient Descent update rule is:

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \tag{4.5}$$

that must be repeated until convergence, where J is a cost function. Adam algorithm is nothing else than an evolution of the update rule of Stochastic Gradient Descent, Adam update rule is:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \tag{4.6}$$

this equation is defined as the signal-to-noise (STN) ratio. When training phase is running, small values of STN mean that there is a lot of noise with respect to the signal, meaning a large uncertainty whether the direction of the first-order gradient point to the direction of the optimum or not. Since the EMA vectors are initialized to zero some little bias is necessary not to have impossible divisions.

4.5 Linear prediction model for residual

A further analysis carried on in this thesis is the attempt to predict the residuals coming from the non-linear prediction model with a linear one. The idea is that the non-linear model is capable of approximating all the non-linear dynamics while the linear ones are left to the linear model. This key analysis could lead to interesting scenarios and results. In particular, being our model linear, it is possible to compute guaranteed error bound by applying the Set Membership presented in [36]. The chosen linear model is an Autoregressive with Exogenous Input (ARX) model that relates the forecasted value of the residual time series to the past values of itself and to the exogenous driving input. The aforementioned model can be expressed as follow:

$$\hat{y}_i(k+1|n_{L,y}) = \varphi_i(k|n_{L,y})^T \theta, k = 0, \ldots, (95 - n_{A,y}) \tag{4.7}$$

where $\hat{y}(k+1|n_{A,y})$ is the (k+1)-step-ahead predictor, ^{T} denotes the transposition operation, $\varphi(k|n_{A,y}) \in \mathbb{R}^{n_{A,y}+(n_{A,u}n_{A,x})}$ is the regressor vector containing the past values of the autoregressive part and the exogenous input data and $\theta \in \mathbb{R}^{n_{A,y}+(n_{A,u}n_{A,x})}$ is the vector containing the parameters to be identified. $n_{A,y}$, $n_{A,u}$ and $n_{A,x}$ are respectively the autoregressive order, the input order (chosen equal for all input) and the number of input. It is possible to derive the multi-step-ahead predictor from the one-step-ahead one by recursively employing it.
Likewise LSTM prediction model, the prediction of the residual is computed day by day. Meaning that knowing the firsts samples of the day, the ARX model predicts

the remaining values of the residual. The regressor vector of the model 4.7 takes the following shape:

$$\varphi_i(k|n_{A,y}) = \left[Y_i^T(k|n_{A,y})\ U_i^T(k) \right]^T \tag{4.8}$$

the vector $Y_i^T(k|n_{A,y})$ is built differently whether $k \geq n_{A,y}$:

$$Y_i^T(k|n_{A,y}) = [\hat{y}_i(k|n_{A,y}), \hat{y}_i(k-1|n_{A,y}), \ldots, \hat{y}_i(k-n_{A,y}+1|n_{A,y})]^T$$

or $k < n_{A,y}$:

$$Y_i^T(k|n_{A,y}) = [\tilde{y}_i(k), \tilde{y}_i(k-1), \ldots, \tilde{y}_i(1)]^T$$

and vector U_i is made by:

$$U_i(k) = \begin{bmatrix} u_{1,i}(k), u_{1,i}(k-1), \ldots, u_{1,i}(k-n_{A,u}+1) \\ u_{2,i}(k), u_{2,i}(k-1), \ldots, u_{2,i}(k-n_{A,u}+1) \\ \vdots \\ u_{n_{A,x},i}(k), u_{n_{A,x},i}(k-1), \ldots, u_{n_{A,x},i}(k-n_{A,u}+1) \end{bmatrix}^T$$

Widely adopted and well-based techniques exist for the parameters identification procedure. In this case a *Simulation Error Method* (SEM) techniques is used that lead to better prediction accuracy. In order to estimate vector of parameter θ the identification routine minimizes the cost function:

$$\hat{\theta} = \arg\min_\theta \sum_{i=1}^{M} \left\| \tilde{Y}_i - \hat{Y}_i(\theta) \right\|_2^2$$

being $\tilde{Y}_i$ the vector of observed value belonging to the training set coherently built and $\hat{Y}_i(\theta)$ is the obtained value simulating the model to be identified. The two vectors take the following shape:

$$\begin{aligned} \tilde{Y}_i &= [\tilde{y}_i(1), \ldots, \tilde{y}_i(96)]^T \\ \hat{Y}_i(\theta) &= [\hat{y}_i\left(1 \mid n_y\right), \ldots, \hat{y}_i\left(96 - n_y \mid n_y\right)]^T \end{aligned}$$

where $\hat{y}_i\left(\cdot \mid n_y\right)$ is described in 4.7. The result is the vector of parameters $\hat{\theta} = \left[\theta_y^T\ \theta_x^T \right]^T$ that can be decomposed in the autoregressive parameters $\theta_y^T \in \mathbb{R}^{n_{A,y}}$ and the input parameters $\theta_x^T \in \mathbb{R}^{n_{A,u} n_{A,x}}$.

Chapter 5

Implementation and experiments

5.1 Evaluation indices

Two criteria are used to evaluate the goodness of the forecasting model under study: fitting performances and forecasting error. The forecasting error is quantified with $MAPE$ (Mean Average Prediction Error) which applies only to validation dataset. It is defined as:

$$MAPE = \frac{100}{N_v} \left| \frac{\tilde{y}(t) - \hat{y}(t)}{\tilde{y}(t)} \right| \tag{5.1}$$

where N_v is the number of samples of the validation dataset (eg. $N_v = 672$), $\tilde{y}(t)$ is the observed load value and $\hat{y}(t)$ is the load predicted value. Whereas the fitting performances of the prediction model is evaluated with R^2, defined as:

$$R^2 = 1 - \frac{\sum_{j=1}^{N_t} (\tilde{y}(j) - \hat{y}(j))^2}{\sum_{j=1}^{N_t} (\tilde{y}(j) - \bar{y})^2} \tag{5.2}$$

where N_t is the number of samples of the training dataset (eg. $N_t = 1344$) and:

$$\bar{y} = \tfrac{1}{N_t} \textstyle\sum_{j=1}^{N_t} \tilde{y}(j)$$

It is easy to infer that $0 \leq R^2 \leq 1$. The more R^2 is close to 1 the more the fitting to training dataset is good. It is worth saying that a good fitting in training not always leads to a low prediction error but could easily bring to over-fitting.

5.2 Test set definition

The overall evaluation is performed on 9 different 3-week-long time spans chosen among all the reliable dataset. The dataset is chosen according to these criteria: they do not contain any unreliable load data, they do not contain more than 8 hours of unreliable weather data and they do not contain a holiday during the working days. Table 5.1 shows the chosen dataset for numerical analysis:

	2017	2018					2019		
Week no.	1	2	3	4	5	6	7	8	9
Start	17/07	19/02	28/05	04/06	03/09	26/11	21/01	18/02	04/03
End training	31/07	05/03	11/06	18/06	17/09	10/12	04/02	04/03	18/03
End	07/08	12/03	18/06	25/06	24/09	17/12	11/02	11/03	25/03

Table 5.1. Dataset containing the nine 3-weeks-long time span on which the routine is tested. "Start" states the beginning, "End training" states the end of the model training dataset (e.g. 2 weeks ahead) and "End" states the date up to the prediction is executed (e.g. 3 weeks ahead)

5.3 Result and comparative evaluation

In this section the final results are presented. In particular a comparative evaluation is done among the proposed innovative model, the load prediction with only LSTM Neural Network and the load prediction made with only ARX model. For all the models the input vector is made by:

- Temperature
- Global Radiation
- Fictitious input
  - $FI_{1,7}$ for LSTM neural network. The Fictitious input contains only two harmonics corresponding to daily and weekly periodic signals
  - $FI_{1:14}$ for all the other identification. The Fictitious input contains all the harmonics obtained with $j = 1, \ldots, 14$ as described in Section 4.3

the different choice of the Fictitious Input is a consequence of the model type. The LSTM model has the same forecasting performances with the two Fictitious Input type therefore the simpler FI is used. On the other hand, ARX model has much poorer results when fed with $FI_{1,7}$ and, at a price of a longer identification phase, $FI_{1:14}$ is employed and the prediction is much more accurate. On the left, Table 5.2 shows the result for running models with "First Sample" weather prediction model. From left to right we can appreciate MAPE and R^2 of load prediction executed with LSTM, with ARX model identified directly on the load consumption data and with the LSTM+ARX proposed model. 6 weeks out of 9 experience an improvement in forecasting performances with the proposed model. The forecasting error is reduced up to 50% stating the superior forecasting performances of the hybrid model. The last column on the right shows the results obtained with the hybrid model using real weather as exogenous input. In many cases the forecasting error reached with weather model is not far from the one with real weather and, in week 6, the forecasting performance of the model with weather prediction is even better.

Week	**LSTM**		**ARX**		**LSTM+ARX**		**LSTM+ARX RW**	
	$MAPE$	R^2	$MAPE$	R^2	$MAPE$	R^2	$MAPE$	R^2
1	16.25	95.69%	**11.27**	96.27%	11.73	**97.87%**	8.05	98.89%
2	38.09	54.26%	29.75	77.91%	**20.66**	**85.22%**	8.31	96.47%
3	8.15	96.27%	9.07	91.30%	**8.11**	**97.09%**	6.54	97.42%
4	11.41	92.62%	10.10	88.90%	**9.07**	**94.75%**	6.43	97.65%
5	**4.89**	90.45%	9.05	93.96%	5.37	**96.62%**	4.02	98.05%
6	15.93	71.05%	11.21	82.49%	**9.84**	**92.13%**	10.98	96.04%
7	9.19	77.30%	9.97	79.23%	**7.95**	**88.82%**	6.71	95.70%
8	15.53	77.35%	9.28	84.90%	**7.97**	**89.02%**	7.32	95.42%
9	6.11	88.67%	**4.55**	91.39%	4.77	**95.91%**	4.12	96.24%

Table 5.2. MAPE and R^2 values of one-week-ahead load forecasting for the 9 of different prediction models prediction for weeks in Table 5.1. The exogenous input used are temperature, radiation and $FI_{1:14}$ for ARX model and temperature, radiation and $FI_{1,7}$ for LSTM model. In **bold** are highlighted the value with the lower MAPE and with the best fitting coefficient. From left to right: load prediction with LSTM only with *First samples* weather model, load prediction with ARX only with *First samples* weather model, load prediction with LSTM+ARX with *First samples* weather model and load prediction with LSTM+ARX with observed weather data

5.3.1 Exogenous input result

In a preliminary phase different combinations of exogenous input are tested to understand how to improve the forecasting performance of the final model. This tests are made with LSTM Neural Network only. The first analysis is carried out with different combinations of artificial input and weather type input. Since deep learning training phase involves a big amount of statistical based decisions, 7 training routines are run with the same input and with the same model. Table 2.1 shows the obtained results. The first value shows the average MAPE obtained with the 7 simulation, while the value in brackets shows the standard deviation of the distribution. On the column, from left to right it is possible to appreciate the results for different artificial input types: no artificial type input, only *Daytype1* input, only *Daytype2* input, only *Time* input, a combination of *Daytype1* and *Time*, a combination of *Daytype2* and *Time*, $FI_{1,7}$ with daily and weekly periodic harmonics and $FI_{1:14}$. The bold value indicates the best obtained MAPE. It is clear that the combination of *Daytype2* and *Time* has the highest forecasting performances but it is worth keeping to test also Fictitious Input to understand how it behaves with ARX and with the overall model, too.
The new tested exogenous input is the maximum, minimum and average load value of the previous day. Table 2.2 shows the average MAPE results and their standard deviations obtained over 7 simulations. On the columns, the used weather and artificial input are described. In **bold** is possible to appreciate the best result and from a comparison with Table 2.1, the obtained average MAPE suggests that this new type of exogenous input does not add information.
Lastly, the same analysis is performed for maximum, minimum and average load values of the previous same-type day. In this occasion, the new input values are divided into working days and holidays and the obtained vector is fed into the LSTM network. Table 2.3 shows the obtained result. Even though form Figure 2.5 (see Section 2.3.2) is possible to observe an improvement of the correlation when clustering the data into day type, the final prediction model does not obtain better forecasting error.

5.3.2 ARX model for load

The ARX model for load prediction is taken from [39]. In this work the ARX model best performs with the autoregressive order $n_{A,y}$ and the input order $n_{A,u}$ equal to 3 and to 1, respectively. The resulting MAPEs and fitting coefficients R^2 are shown in column four and five of Table 5.2 for the proposed model with weather prediction input. Despite its simple structure, ARX model is able to catch the periodicity of the time series. It shows the best results in in 2 cases out of 9.

5.3.3 ARX model for residual

A slight different configuration is used for the residual prediction with ARX model. From LSTM prediction the first $n_{L,y}$ data of the day is used to predict the remaining $(96 - n_{L,y})$. Therefore, the first $n_{L,y}$ load measures are supposed to be known which means that the first $n_{L,y}$ residuals are zero. From this consideration, it is not possible to catch any residual dynamic using the first $n_{L,y}$ samples of the day. To solve this

problem, the following $n_{A,y}$ samples are engaged. To sum up, when referring to ARX residual prediction, the ARX model identification process does not consider the first $n_{L,y}$ samples of the day and the first supposed unknown value results in the $n_{L,y} + n_{A,y} + 1$ sample. For our simulation $n_{A,y} = 3$ and $n_{A,u} = 1$ is used. Furthermore, the prediction performances improves by increasing the autoregressive order while, increasing input order does not lead to better forecasting but only to a more computational demanding routine.

5.3.4 Combined models

To better appreciate the novel method some numerical example are plotted. Two examples with real weather input are shown in Figure 5.1 and Figure 5.2 and two examples with "First Samples" weather model are depicted Figure 5.3 and Figure 5.4. In each plot the observed load is depicted in blue and the predicted one in red. A green plot showing the obtained MAPE values day by day shows how effectively the proposed model improves the forecasting accuracy of the time series prediction. From top to bottom: plot of the prediction with LSTM only, plot of the prediction with ARX on the residuals coming from LSTM prediction, plot of the proposed hybrid model, plot of the prediction performed with ARX model only and plots of temperature and radiation evolution in time.

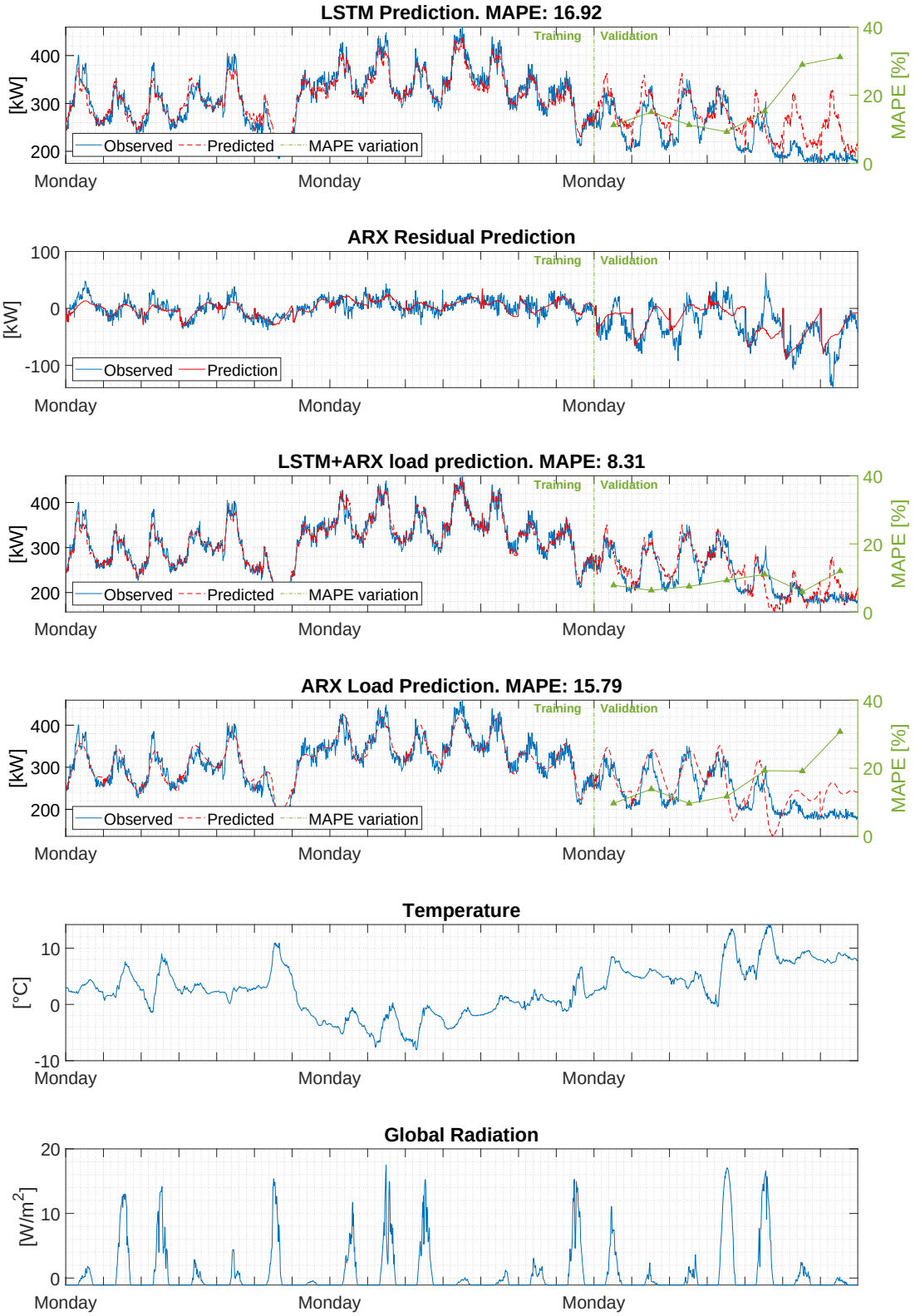


Figure 5.1. Prediction result for week no. 2 with real weather data input. In blue observed data and in red prediction data. From top to bottom: prediction with LSTM model, residual prediction with ARX model, prediction with the combination of LSTM and ARX model, load prediction with ARX model and temperature and radiation. In green is highlighted the validation set and the green plot shows how MAPE varies day by day

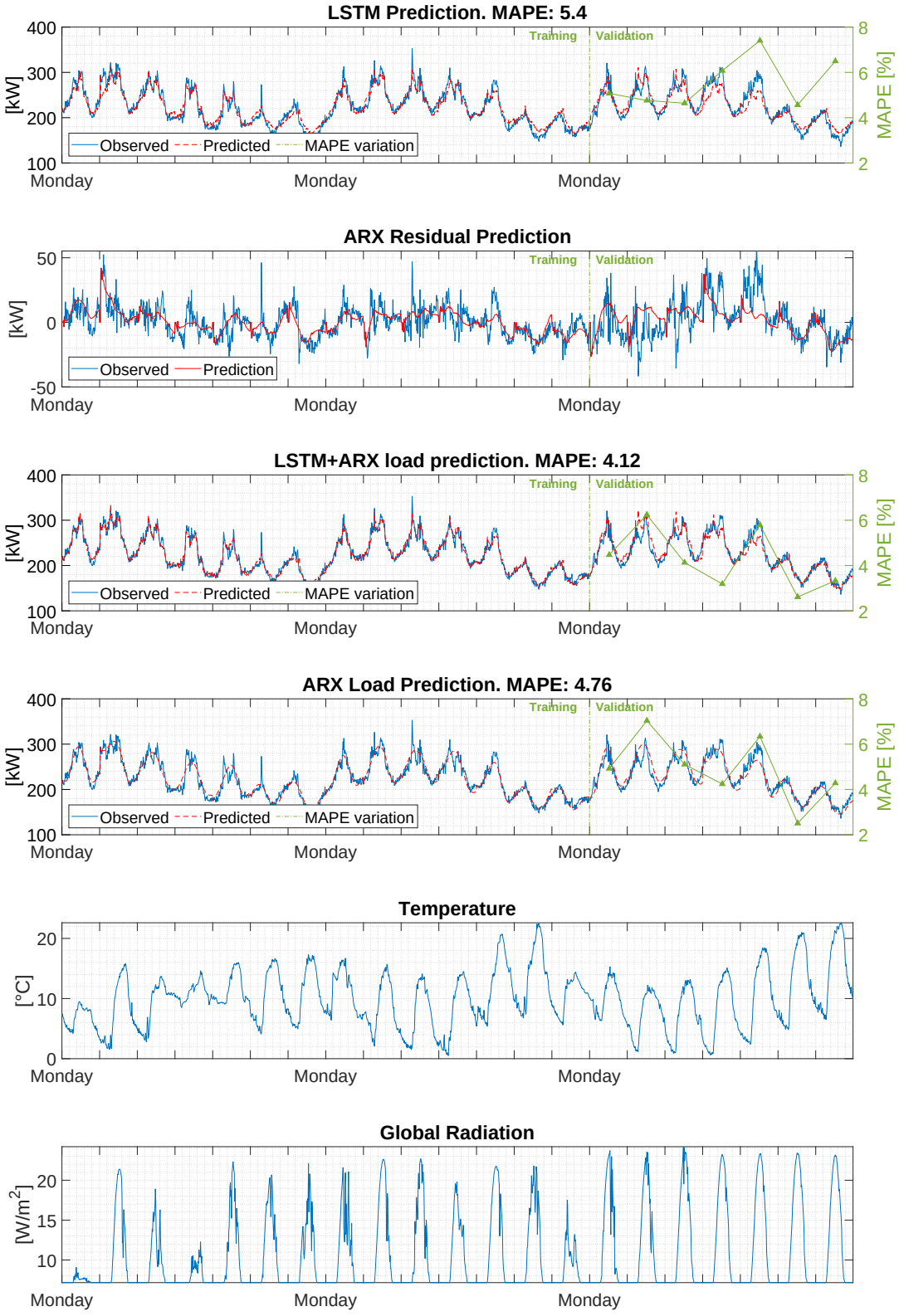


Figure 5.2. Prediction result for week no. 9 with real weather data input. In blue observed data and in red prediction data. From top to bottom: prediction with LSTM model, residual prediction with ARX model, prediction with the combination of LSTM and ARX model, load prediction with ARX model and temperature and radiation. In green is highlighted the validation set and the green plot shows how MAPE varies day by day

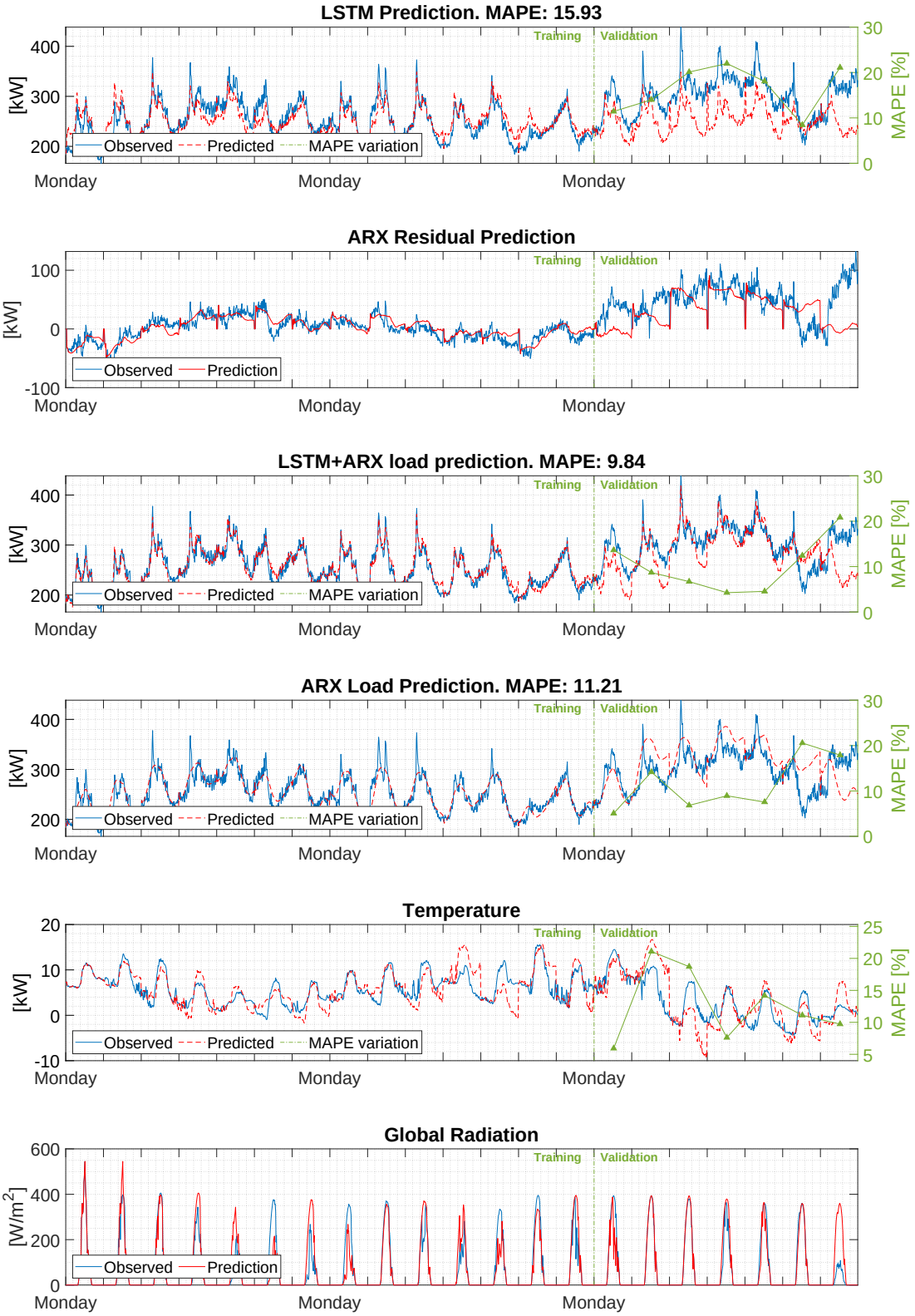


Figure 5.3. Prediction result for week no. 6 with "First Sample" model weather input. In blue observed data and in red prediction data. From top to bottom: prediction with LSTM model, residual prediction with ARX model, prediction with the combination of LSTM and ARX model, load prediction with ARX model and prediction of temperature and radiation. In green is highlighted the validation set and the green plot shows how MAPE varies day by day

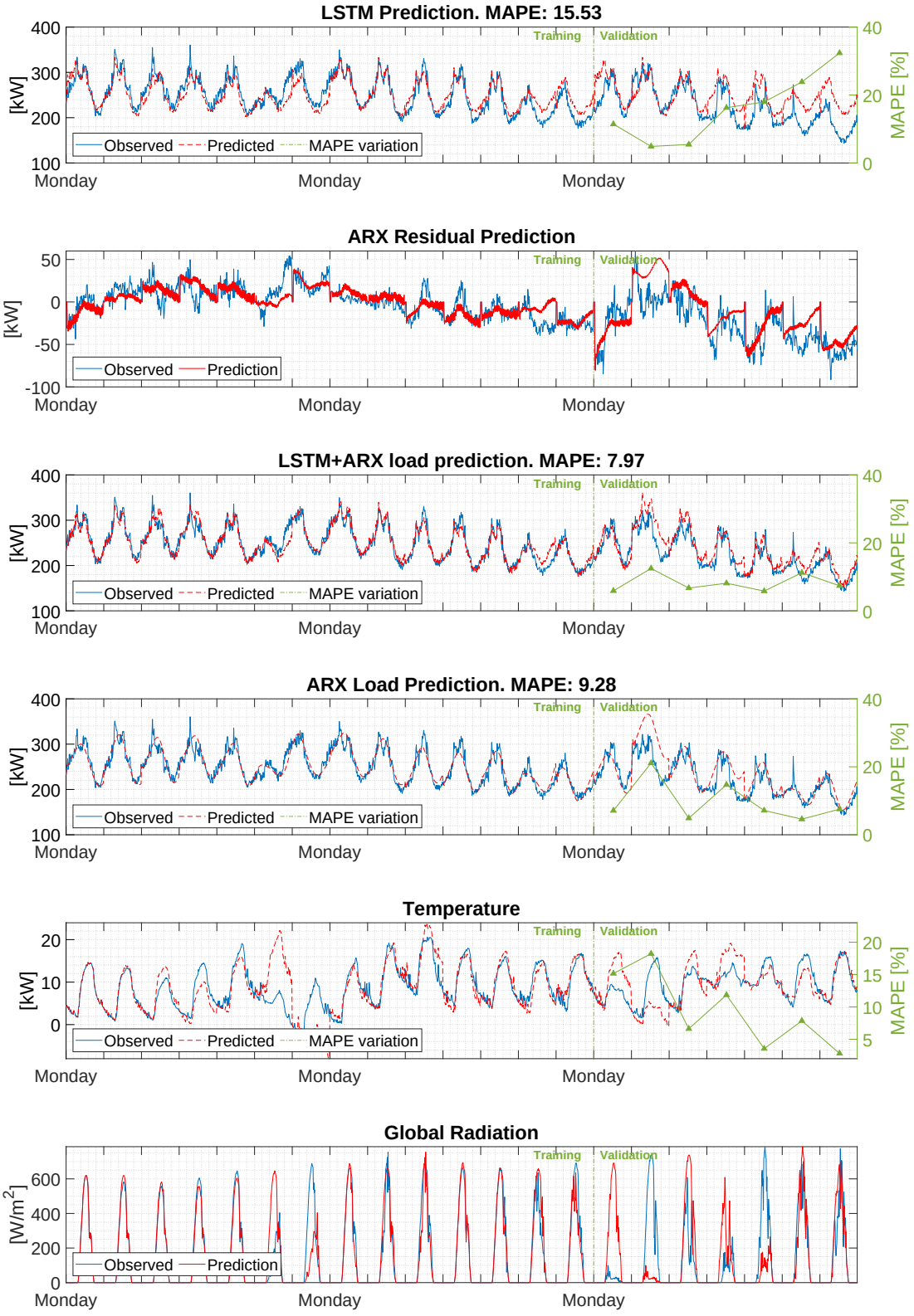


Figure 5.4. Prediction result for week no. 8 with "First Sample" model weather input. In blue observed data and in red prediction data. From top to bottom: prediction with LSTM model, residual prediction with ARX model, prediction with the combination of LSTM and ARX model, load prediction with ARX model and prediction of temperature and radiation. In green is highlighted the validation set and the green plot shows how MAPE varies day by day

Conclusions and Future Developments

In this thesis a novel approach for non-residential building electricity consumption forecasting is presented. The proposed model consists of a combination of non-linear and linear predictions to catch all the possible dynamics and information contained in the training time series. Preliminary the weather data are fed into weather model to perform a prediction of these variables. The first step of the proposed prediction model consists of the training and prediction of the load consumption through LSTM Neural Network. Then, the obtained residual is used to identify the parameters of an ARX model used to forecast the error made by LSTM model. In the end, the final prediction of the time series is the sum of the two predictions. The most influencing weather input are temperature and global radiation and, more importantly, a good forecasting algorithm for those weather variables are a key point for a good forecasting performance. For weather prediction, the usage of complex non-linear Neural Network based algorithm does not lead to better forecasting performances compared to more intuitive methods and the much higher computational request for the identification of the model parameters confirms their usage has a negative result. Moreover, the Fictitious Input is embedded in the model. This input models the evolution of time and thus improves the performances of the overall prediction by intrinsically performing a kind of frequency decomposition of the signal. The reduced amount of data available is a key point. The configuration of the LSTM Neural Network has a good exploitation of the provided data but further researches is recommended in order to exploit better the information. Further analyses could be done also on data pre-processing and on how techniques like clustering and more advanced data augmentation should be conducted. A fundamental consideration prior new analyses is to keep using a linear model with the opportunity to compute guaranteed error bounds on the error through the Set Membership (SM) approach. Additional tests should be conducted on new open-source dataset in order to cross validate the obtained results considering the framework, the type of area the factory is located and additional correlated input that could help in improving the performance.

Bibliography

[1] R. M. Murray, K. J. Astrom, S. P. Boyd, R. W. Brockett, and G. Stein, "Future directions in control in an information-rich world," *IEEE Control Systems Magazine*, vol. 23, no. 2, pp. 20–33, 2003.

[2] Z. Aung, M. Toukhy, J. Williams, A. Sanchez, and S. Herrero, "Towards accurate electricity load forecasting in smart grids," *Proceedings of DBKDA*, pp. 51–57, 2012.

[3] H. S. Hippert, C. E. Pedreira, and R. C. Souza, "Neural networks for short-term load forecasting: A review and evaluation," *IEEE Transactions on power systems*, vol. 16, no. 1, pp. 44–55, 2001.

[4] P. Lauret, M. David, and D. Calogine, "Nonlinear models for short-time load forecasting," *Energy Procedia*, vol. 14, pp. 1404–1409, 2012.

[5] Y. Liang, D. Niu, and W.-C. Hong, "Short term load forecasting based on feature extraction and improved general regression neural network model," *Energy*, vol. 166, pp. 653–663, 2019.

[6] N. Liu, Q. Tang, J. Zhang, W. Fan, and J. Liu, "A hybrid forecasting model with parameter optimization for short-term load forecasting of micro-grids," *Applied Energy*, vol. 129, pp. 336–345, 2014.

[7] H. Zheng, J. Yuan, and L. Chen, "Short-term load forecasting using emd-lstm neural networks with a xgboost algorithm for feature importance evaluation," *Energies*, vol. 10, no. 8, p. 1168, 2017.

[8] S.-l. Yang, C. Shen *et al.*, "A review of electric load classification in smart grid environment," *Renewable and Sustainable Energy Reviews*, vol. 24, pp. 103–110, 2013.

[9] K. Zor, O. Timur, and A. Teke, "A state-of-the-art review of artificial intelligence techniques for short-term electric load forecasting," in *2017 6th International Youth Conference on Energy (IYCE)*. IEEE, 2017, pp. 1–7.

[10] I. Moghram and S. Rahman, "Analysis and evaluation of five short-term load forecasting techniques," *IEEE Transactions on power systems*, vol. 4, no. 4, pp. 1484–1491, 1989.

[11] Y. Yang, S. Li, W. Li, and M. Qu, "Power load probability density forecasting using gaussian process quantile regression," *Applied Energy*, vol. 213, pp. 499–509, 2018.

[12] D. J. Leith, M. Heidl, and J. V. Ringwood, "Gaussian process prior models for electrical load forecasting," in *2004 International Conference on Probabilistic Methods Applied to Power Systems*. IEEE, 2004, pp. 112–117.

[13] "Previsione del carico a breve termine utilizzando modelli di processo gaussiano," *Proceedings of Coimbra's Institute of Systems and Computer Engineering*.

[14] M. Gilanifar, H. Wang, E. E. Ozguven, Y. Zhou, and R. Arghandeh, "Bayesian spatiotemporal gaussian process for short-term load forecasting using combined transportation and electricity data," *ACM Transactions on Cyber-Physical Systems*, vol. 4, no. 1, pp. 1–25, 2019.

[15] E. Ceperic, V. Ceperic, and A. Baric, "A strategy for short-term load forecasting by support vector regression machines," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4356–4364, 2013.

[16] S. R. Abbas and M. Arif, "Electric load forecasting using support vector machines optimized by genetic algorithm," in *2006 IEEE International Multitopic Conference*. IEEE, 2006, pp. 395–399.

[17] J. Moon, J. Kim, P. Kang, and E. Hwang, "Solving the cold-start problem in short-term load forecasting using tree-based methods," *Energies*, vol. 13, no. 4, p. 886, 2020.

[18] A. Almalaq and J. J. Zhang, "Evolutionary deep learning-based energy consumption prediction for buildings," *IEEE Access*, vol. 7, pp. 1520–1531, 2018.

[19] J. Lu, Q. Zhang, Z. Yang, and M. Tu, "A hybrid model based on convolutional neural network and long short-term memory for short-term load forecasting," in *2019 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 2019, pp. 1–5.

[20] M. Alamaniotis, "Synergism of deep neural network and elm for smart very-short-term load forecasting," in *2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*. IEEE, 2019, pp. 1–5.

[21] D.-C. Li, C. Wu, and F. M. Chang, "Using data-fuzzification technology in small data set learning to improve fms scheduling accuracy," *The International Journal of Advanced Manufacturing Technology*, vol. 27, no. 3-4, pp. 321–328, 2005.

[22] D.-C. Li, C.-J. Chang, C.-C. Chen, and W.-C. Chen, "A grey-based fitting coefficient to build a hybrid forecasting model for small data sets," *Applied Mathematical Modelling*, vol. 36, no. 10, pp. 5101–5108, 2012.

[23] H. H. Aly, "A proposed intelligent short-term load forecasting hybrid models of ann, wnn and kf based on clustering techniques for smart grid," *Electric Power Systems Research*, vol. 182, p. 106191, 2020.

[24] M. Talaat, M. Farahat, N. Mansour, and A. Hatata, "Load forecasting based on grasshopper optimization and a multilayer feed-forward neural network using regressive approach," *Energy*, vol. 196, p. 117087, 2020.

[25] P. Zhou, B. Ang, and K. L. Poh, "A trigonometric grey prediction approach to forecasting electricity demand," *Energy*, vol. 31, no. 14, pp. 2839–2847, 2006.

[26] M. Fan, Y. Hu, X. Zhang, H. Yin, Q. Yang, and L. Fan, "Short-term load forecasting for distribution network using decomposition with ensemble prediction," in *2019 Chinese Automation Congress (CAC)*. IEEE, 2019, pp. 152–157.

[27] A. Bracale, G. Carpinelli, P. De Falco, and T. Hong, "Short-term industrial load forecasting: A case study in an italian factory," in *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. IEEE, 2017, pp. 1–6.

[28] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "Stl: A seasonal-trend decomposition," *Journal of official statistics*, vol. 6, no. 1, pp. 3–73, 1990.

[29] L. Kegel, M. Hahmann, and W. Lehner, "Feature-based comparison and generation of time series," in *Proceedings of the 30th International Conference on Scientific and Statistical Database Management*, 2018, pp. 1–12.

[30] Q. Wen, L. Sun, X. Song, J. Gao, X. Wang, and H. Xu, "Time series data augmentation for deep learning: A survey," *arXiv preprint arXiv:2002.12478*, 2020.

[31] A. Hooshmand and R. Sharma, "Energy predictive models with limited data using transfer learning," in *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, 2019, pp. 12–16.

[32] T. Lin, T. Guo, and K. Aberer, "Hybrid neural networks for learning the trend in time series," in *Proceedings of the twenty-sixth international joint conference on artificial intelligence*, no. CONF, 2017, pp. 2273–2279.

[33] C. Huang and C. Moraga, "A diffusion-neural-network for learning from small samples," *International Journal of Approximate Reasoning*, vol. 35, no. 2, pp. 137–161, 2004.

[34] J.-L. Yuan and T. L. Fine, "Neural-network design for small training sets of high dimension," *IEEE Transactions on neural networks*, vol. 9, no. 2, pp. 266–280, 1998.

[35] C. Li, Z. Ding, D. Zhao, J. Yi, and G. Zhang, "Building energy consumption prediction: An extreme deep learning approach," *Energies*, vol. 10, no. 10, p. 1525, 2017.

[36] M. Lauricella and L. Fagiano, "On the identification of linear time invariant systems with guaranteed simulation error bounds," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 1439–1444.

[37] L. Fagiano, "Lecture note of constrained numerical optimization for estimation and control," 2018-19.

[38] S. Bittant, "Lecture note on model identification and data analysis," 2017-18.

[39] L. F. Marco Lauricella, Zhongtian Cai, "Day-ahead building load forecasting with a samll data-set," 2017.

[40] H. Hippert and C. Pedreira, "Estimating temperature profiles for short-term load forecasting: neural networks compared to linear models," *IEE Proceedings-Generation, Transmission and Distribution*, vol. 151, no. 4, pp. 543–547, 2004.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.