TÉCNICO LISBOA

Forecast of electrical consumption in Industrial Kitchens

Vasco Manuel de Mendonça Andrade

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisors: Prof. Hugo Gabriel Valente Morais
Prof. Amâncio Lucas de Sousa Pereira

Examination Committee

Chairperson: Celia Maria Santos Cardoso de Jesus
Supervisor: Prof. Hugo Gabriel Valente Morais
Member of the Committee: Prof. Sérgio Filipe Carvalho Ramos

June 2023

This work was created using LaTeX typesetting language
in the Overleaf environment (www.overleaf.com).

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

First of all I would like to express my gratitude to professors Hugo Morais and Lucas Pereira, for inviting me into *nexIK* project and trusting my capabilities. Their advices, support and motivation throughout the last months made me learn so much and were decisive in the realization of this work.

To my parents, a special thanks for all the love, unconditional support and sacrifices during this years. Without you I wouldn't be where I am today. Your belief in my capabilities made me endure the hardest times in this journey. I will never know how to express my gratitude for having you. A special word also for my sister that provided me an escape during the stressful times of this months.

To Isabel, a word of gratitude for all the encouragement and support given. Without you this journey would have been much harder. Your emotional support was crucial to overcome myself and to pursue my dream.

To all my friends, from my childhood to the ones I made in Técnico, thank you for all the help and support. Your motivation and cheeriness during all these years is something that I will never forget. To the friends I made in Munich during my exchange, thank you for all the friendship and kindness.

To each and every one of you - Thank you.

Abstract

Forecasting methods have been increasingly used in the power sector, for various purposes. Anticipating network disturbances increases the security of the power system by allowing preventive measures to be taken, to keep the system running. The forecast of active power consumption, combined with load flexibility management, allows for greater energy efficiency, especially at the building's level. Although this topic has already been researched in the literature with regard to the forecast of aggregate loads, it has not yet been substantially addressed in relation to the forecast of individual loads of electrical appliances.

The main contribution of this thesis is the prediction of active power at the appliance level in industrial kitchens, both in the case of day-ahead and hour-ahead predictions. With this purpose, methodologies based on decision trees, gradient boosting and neural networks were applied. Post-processing algorithms were also proposed to reduce prediction errors.

The results obtained were satisfactory, mainly because this topic still presents very few advancements in the literature. A proposed post-processing technique also showed good performance, with an error reduction of up to 29% in some appliances, when compared to the reference techniques, and up to 50% when compared with the results without post-processing.

Keywords

Load Forecast; Appliance; Weighted Prediction; Machine Learning; Day-ahead; Hour-ahead

Resumo

Cada vez mais o setor energético utiliza métodos de previsão, para diversos fins. A antecipação de perturbações na rede aumenta a segurança do sistema elétrico, ao possibilitar que sejam tomadas medidas preventivas para manter o sistema operacional. A previsão de potência ativa consumida, combinada com a gestão de flexibilidade de cargas, permite uma maior eficiência energética, sobretudo ao nível de edifícios. Apesar deste tema já ter sido explorado na literatura no que toca à previsão de cargas agregadas, não foi ainda substancialmente abordado em relação à previsão de cargas individuais de aparelhos elétricos.

A principal contribuição desta tese é a previsão de potência ativa ao nível de eletrodomésticos em cozinhas industriais, tanto no caso de previsão para o dia seguinte como para a hora seguinte. Para tal, foram aplicadas metodologias baseadas em árvores de decisão, *gradient boosting* e redes neuronais. Foram também propostos algoritmos de pós-processamento para diminuir os erros de previsão.

Os resultados obtidos foram satisfatórios, sobretudo devido a este tópico ainda apresentar bastante poucos desenvolvimentos na literatura. A técnica de pós-processamento proposta apresenta também boa performance, com redução do erro até 29% em alguns eletrodomésticos, quando comparado com as técnicas de referência, e de até 50% quando comparado com os resultados sem pós-processamento.

Palavras Chave

Previsão de cargas; Eletrodoméstico; Previsão Ponderada; Aprendizagem Automática; Dia Seguinte; Hora Seguinte

Contents

1 Introduction **1**

1.1 Motivation . . . 3

1.2 Contribution . . . 4

1.3 Outline . . . 4

2 Background and Related Works **7**

2.1 Forecasting Horizons . . . 9

2.2 Forecasting Models . . . 10

2.2.1 Traditional Models . . . 10

2.2.2 Artificial Intelligence/Machine Learning based Models . . . 11

2.2.3 Deep-Learning based Models . . . 12

2.3 Load forecasting - Aggregate Level . . . 12

2.4 Load Forecasting - Appliance Level . . . 14

2.5 Load Forecasting - Application in Industrial Kitchens . . . 14

3 Appliance Level Load Forecasting - Proposed Methodology **17**

3.1 Data Pre-Processing . . . 19

3.1.1 Missing Data and Filtering . . . 20

3.1.2 Sliding Window and Training Length . . . 21

3.2 Forecasting Algorithms . . . 22

3.2.1 XGBoost . . . 22

3.2.2 CatBoost . . . 23

3.2.3 LSTM . . . 23

3.3 Model optimization . . . 24

3.3.1 Window Size Analysis . . . 25

3.3.2 Training Set Length . . . 25

3.3.3 Hyperparameter Optimization . . . 26

3.4 Data Post-processing . . . 27

3.4.1 Most Recent Prediction . . . 28

3.4.2 Persistence Based Prediction . . . 28
3.4.3 Average Value Prediction . . . 29
3.4.4 Weighted Prediction . . . 29
3.5 Performance evaluation . . . 31

4 Appliance Level Load Forecasting - Performance Evaluation **33**
4.1 Dataset Description . . . 35
4.1.1 Results from pre-processing . . . 36
4.2 Window Size Analysis . . . 39
4.3 Best Training Length . . . 40
4.4 Hyperparameter Optimization Results . . . 42
4.5 Day-ahead Results . . . 42
4.5.1 Kitchen 1 . . . 43
4.5.2 Kitchen 2 . . . 43
4.6 Results - Hour-ahead . . . 45
4.6.1 Kitchen 1 . . . 45
4.6.2 Kitchen 2 . . . 46
4.7 Training Times . . . 47
4.8 Post-processing Results . . . 48
4.9 Discussion . . . 52

5 Integration with flexibilities **55**
5.1 Demand-Side Flexibility . . . 57
5.2 Methodology . . . 58
5.3 Results . . . 60

6 Conclusion **63**
6.1 Conclusions . . . 65
6.2 System Limitations and Future Work . . . 66

Bibliography **67**

A Hyperparameter Optimization Results **75**

List of Figures

2.1 Forecasting Horizons . . . 10

3.1 Proposed methodology for forecasting . . . 19
3.2 Example of feature and target sequences - day-ahead . . . 21
3.3 LSTM block architecture (Source: [53]) . . . 24
3.4 Expanding window cross-validation (Source: [60]) . . . 26
3.5 Example for Most Recent Prediction . . . 28
3.6 Example for Persistence-based Prediction . . . 29
3.7 Flowchart for weighted predictions . . . 31

4.1 Original load data of Induction Plate . . . 36
4.2 Original load data of one appliance - Exhaust . . . 38
4.3 Processed load data of one appliance - Exhaust . . . 38
4.4 Original load data of one appliance - Salamander 1 . . . 39
4.5 Processed load data of one appliance - Salamander 1 . . . 39
4.6 Post-processing techniques for day-ahead analysis - Infrared Shelf . . . 49
4.7 Post-processing techniques for day-ahead analysis - Exhaust . . . 49
4.8 Post-processing techniques for hour-ahead analysis - Convection Oven 2 . . . 51
4.9 Post-processing techniques for hour-ahead analysis - Heating Lamp 4 . . . 52
4.10 Post-processing techniques for hour-ahead analysis - Refrigerator 1 . . . 52

5.1 Methodology for integration with flexibilities . . . 58
5.2 Sample of Dish Washer original load . . . 59
5.3 ON state duration forecast - Drinks Refrigerator . . . 61
5.4 ON state duration forecast - Refrigerator 1 . . . 62
5.5 ON state duration forecast - Garde Manger 2 . . . 62
5.6 ON state duration forecast - Refrigerator 4 . . . 62

List of Tables

3.1 Historic window size values . . . 25
3.2 Hyperparameters for optimization . . . 27
3.3 Set of weights for weighted predictions . . . 31

4.1 Description of each kitchen . . . 35
4.2 Null values generated due to re-sampling to 30 minutes . . . 35
4.3 Null values generated due to re-sampling to 5 minutes . . . 36
4.4 Data characteristics after pre-processing . . . 37
4.5 Historic window size analysis - day-ahead . . . 40
4.6 Historic window size analysis - hour-ahead . . . 40
4.7 Day-ahead analysis - Best training length for K1 . . . 41
4.8 Day-ahead analysis - Best training length for K2 . . . 41
4.9 Hour-ahead analysis - Best training length for K1 . . . 42
4.10 Hour-ahead analysis - Best training length for K2 . . . 42
4.11 Day-ahead RMSE [W] - Kitchen 1 . . . 43
4.12 Day-ahead RMSE [W] - Kitchen 2 . . . 44
4.13 Hour-ahead RMSE [W] - Kitchen 1 . . . 45
4.14 Hour-ahead RMSE [W] - Kitchen 2 . . . 46
4.15 Average training times [s] for day-ahead predictions . . . 47
4.16 RMSE [W] of post-processing methods for K1 - day-ahead analysis . . . 48
4.17 RMSE [W] of post-processing methods for K2 - day-ahead analysis . . . 49
4.18 RMSE [W] of post-processing methods for K1 - hour-ahead analysis . . . 50
4.19 RMSE [W] of post-processing methods for K2 - hour-ahead analysis . . . 50

5.1 Classification of IK appliances according to their flexibility potential (Source: [7]) . . . 58
5.2 RMSE [s] of post-processing methods for K1 . . . 60
5.3 RMSE [s] of post-processing methods for K2 . . . 60

A.1 Hyperparameters for XGBoost - day-ahead analysis 76
A.2 Hyperparameters for XGBoost - hour-ahead analysis 76
A.3 Hyperparameters for CatBoost - day-ahead analysis 77
A.4 Hyperparameters for CatBoost - hour-ahead analysis 77
A.5 Hyperparameters for LSTM - day-ahead analysis 78
A.6 Hyperparameters for LSTM - hour-ahead analysis 78

List of Algorithms

3.1 Average predicted value 30

Acronyms

AI	Artificial Intelligence
AR	Autoregression
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive Moving Average
ANN	Artificial Neural Network
CatBoost	Categorical Boosting
CNN	Convolutional Neural Networks
DL	Deep Learning
DER	Distributed Energy Resources
GBDT	Gradient Boosting Decision Trees
GHG	Greenhouse Gas Emissions
I	Intregration
IK	Industrial Kitchens
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MA	Moving Average
MLPNN	Multi-Layer Perceptron Neural Network
ML	Machine Learning
MAE	Mean Absolute Error
PV	Photovoltaic
RF	Random Forests
RES	Renewable Energy Sources
RNN	Recurrent Neural Network

SVM Support Vector Machine

XGBoost Extreme Gradient Boosting

1

Introduction

Contents

1.1 Motivation 3
1.2 Contribution 4
1.3 Outline 4

1.1 Motivation

In recent years, climate change became a trending topic and many countries are starting to implement measures to reduce Greenhouse Gas Emissions (GHG) to minimize the increase in global temperature[1].

In Portugal, the government presented in 2019 the "Road map to Carbon Neutrality 2050" with the main goal of reducing the GHG emissions between 85% and 90% of the 2005 values[2]. Another plan was presented by the government, called the "National Energy and Climate Plan 2021-2030" that pretends to take a large step towards carbon neutrality[3].

With the increasing energy demand in the world and the objectives assumed by different governments to reduce carbon emissions, energy efficiency has become an intensively discussed topic. In the European Union only, energy production and use is responsible for 80% of all GHG emissions, where buildings account for 40% of energy consumption and 36% of carbon emissions [1], indicating that it is possible to increase energy saving potential in this situations. Also in the UK, the catering industry alone is estimated to be responsible for an excess in consumption of 21 600 million kWh per year [2]. UK's industry represents about 20% of the total energy consumption in the UK and generates 32% of the UK's heat-related carbon emissions, mostly from fossil fuels [3].

Forecasting of power consumption is one of the most widely researched areas in the past decades. In recent years, the increased market competition and renewable energy integration increased the complexity of the power system and the difficulty to obtain quality forecasts, meaning that forecasting has become increasingly important to energy systems planning and operation [4].

In Industrial Kitchens (IK) load forecasting becomes an important task. The activity of this type of industry requires the use of various types of equipment with high energy consumption rates for meal preparation, such as cooking, refrigeration, lighting and hot water supplying [5], with refrigeration often representing the largest category of energy usage [2]. Energy efficiency in this type of building has not been intensively researched yet, especially when combined with forecasting techniques, and it presents an opportunity to try and develop power forecast approaches for this type of industry. Having accurate forecasts combined with load flexibilities and renewable energy systems can result in a considerable increase in the energy efficiency of these buildings. This was shown in [6], where researchers formulated seven possible reduction scenarios, and five of them produced financial payback in one year or less. The most favorable scenario, with the replacement of fryers and reduction of freezing demand, resulted in an energy saving of 58%.

This work is integrated in the project *nexIK: Exploring the Human-Water-Energy Nexus in Industrial Kitchens* (`https://nexik.tecnico.ulisboa.pt/`), which presents as research framework the following objectives [7]:

[1] https://unfccc.int/process/the-paris-agreement/long-term-strategies
[2] https://apambiente.pt/clima/roteiro-para-neutralidade-carbonica-2050
[3] https://www.portugalenergia.pt/setor-energetico/bloco-3/

- Collect year-long water and electricity consumption data from IKs in actual operational conditions;
- Engage with the major IK stakeholders, and understand their perceptions and motivations towards the Water-Energy-Food Nexus, energy efficiency and flexibility opportunities;
- Develop data-driven methods to model and understand resource consumption in IKs;
- Assess the potential of IKs to contribute to increasing the penetration of Renewable Energy Sources (RES) in the energy mix by participating in demand-side flexibility markets;
- Explore potential opportunities to scale up the project findings to other regions of Portugal, Europe, and the World.

1.2 Contribution

The main contribution of this work is to develop and evaluate methodologies to perform forecasts of active power in IK at the appliance level, to effectively understand resource consumption in IK. It is also proposed a different approach to perform forecasts, to relate the forecasts with the flexibility of the appliances to increase energy use efficiency and optimize appliance usage for providing demand-side flexibility considering scenarios with and without the presence of Distributed Energy Resources (DER). The previously stated objectives lead to the following goals of this thesis:

- Implement forecast methodologies to forecast active power in IK at the appliance level, for day-ahead and hour-ahead scenarios;
- Develop a methodology to consider the appliances time in the ON state, by forecasting the duration of the ON state instead of the active power;
- Implement and develop post-processing techniques based on target selection to increase the forecasted data accuracy.

The forecasting horizons in this analysis were chosen as hour-ahead and day-ahead to provide two different time ranges for understanding resource consumption, with the day-ahead analysis performed to obtain information on the overall daily load patterns and the hour-ahead focused on the detailed behavior of each load.

1.3 Outline

This report is structured as follows:

- In Chapter 2 it is presented a review in detail on the state of the art in power forecasting. Several forecasting horizons and models are presented. Some related work in the field of power forecasting is also presented, together with recent advances and developments in power forecasting techniques;
- Chapter 3 presents the methodologies implemented in this work. The pre-processing techniques used are explained, as well as the forecast methods to use. Post-processing methods applied to increase prediction accuracy are also explained;
- The results of methodologies explained in Chapter 3 are presented in Chapter 4. The influence of the pre-processing and post-processing methods is exemplified and the results are analyzed and discussed;
- In Chapter 5 it is developed a different methodology to combine the forecast of power with the flexibilities of the appliances;
- In Chapter 6 the main conclusions of this thesis and future work are presented.

2

Background and Related Works

Contents

2.1 Forecasting Horizons 9
2.2 Forecasting Models 10
2.3 Load forecasting - Aggregate Level 12
2.4 Load Forecasting - Appliance Level 14
2.5 Load Forecasting - Application in Industrial Kitchens 14

This chapter provides insight on the state of the art in power forecasting, detailing aspects such as forecasting horizons and methods. To clarify and support the methodology to be presented, it is presented a review of the state of the art on these topics. Section 2.1 summarizes the types of forecasting horizons used, while Section 2.2 presents a review of the different types of forecast models. Finally, Sections 2.3 to 2.5 present some related work about the horizons and methods indicated in the previous sections, for aggregate, appliance, and IK level respectively.

2.1 Forecasting Horizons

The performance and accuracy of forecast methods depend on several factors, such as the input data, the number of features, or the correlation between input and output [8, 9]. However, the time horizon of the forecast is probably one of the most important aspects that should be considered in the analysis of the methods. The main categories to define the forecasting horizon are (1) very short-term forecasting (also known in the literature as *nowcasting*), (2) short-term forecasting, (3) medium-term forecasting, and (4) long-term forecasting [8]. These categories are relevant to choose a forecasting model since different models may fit better into different categories. An overview of the different horizons is presented next, with a visual representation presented in Figure 2.1.

- **Very short-term forecasting**

  As the name indicates, this time horizon comprises a very short time window, from a few seconds to 30 min. It is sometimes called also *nowcasting* or ultra short-term forecasting [8].

- **Short-term forecasting**

  Short-term forecasting usually has a temporal horizon from 30 min to 6 hours. However, some authors consider a horizon of one day or up to seven days as a short-term horizon [9]. This forecasting horizon is useful in designing energy management systems for Photovoltaic (PV) panels and enhancing the security of the grid operation [9].

- **Medium-term forecasting**

  Medium-term forecasting has a window from 6h to 24h, although some authors have considered one-week or even one-month time horizons in this forecasting period. [9]. This allows the operational planning of power systems, by predicting the future operating conditions and possible technical violations of the electric grid. The 6h to 24h window was chosen to provide two different time horizons for the hour-ahead and day-ahead analysis of this work. This way, the hour-ahead scenario is inserted in the short-term and the day-ahead in the medium-term.

- **Long-term forecasting**

Long-term forecasting usually covers more than 24h ahead scenarios. Depending on the application, long-term forecasts can achieve months or years of forecasting [9]. These time horizons are usually useful for long-term power generation, transmission, and planning.

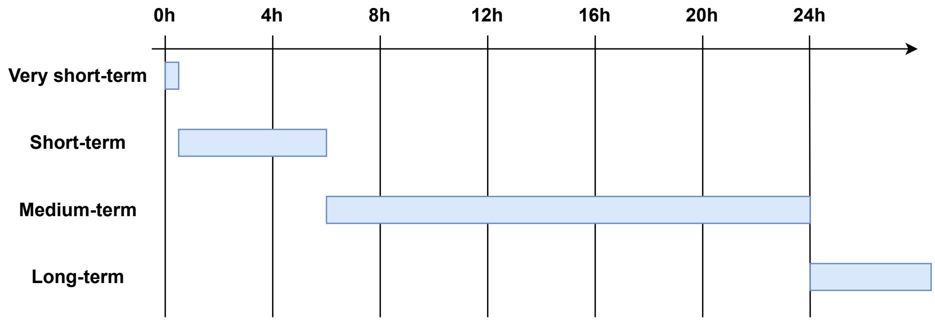

Figure 2.1: Forecasting Horizons

Other classifications can be adopted to describe the forecast concerning the time horizon. For example, other authors [10] present a different approach, using a terminology based on intra-hour, intra-day, six hours to day-ahead, and two days ahead or longer. These categories often overlap with the very short, short, medium, and long-term categories mentioned.

2.2 Forecasting Models

Researchers have studied numerous different methods and models to perform power forecasts. These methods can be identified as **traditional** models, regarding models based on physical and statistical approaches, and as **artificial intelligence/machine learning** [11]. In the following subsections, an overview of those methods is presented.

2.2.1 Traditional Models

The most basic model proposed in the literature is the persistence model, which assumes that the next day's load will be the same as the previous day. It is a simple model and highly depends on the load pattern's stability and weather factors. However, its often used as a benchmark for other forecasting models [9]. Weather conditions such as temperature, cloud cover, and air pressure are often the features used in physics-based models and are more accurate when designed specifically for one location [8]. The accuracy of the physical forecasting models is higher when the weather conditions are stable [12], and thus the performance is largely affected due to the sharp changes in meteorological variables.

Statistical models use historical time series and mathematical equations to extract patterns and correlations from past input data. The most commonly used statistical models for power forecasting are the Autoregressive Moving Average (ARMA) and Autoregressive Integrated Moving Average (ARIMA). The ARMA model consists of the combination of a Autoregression (AR), used to specify the previous values from the time series which are used to predict the current values [13], with a Moving Average (MA), that specifies how deviations from the time series mean for previous values are used to predict the current values. It is considered a good prediction model for stationary time series data [14]. An improvement to this model is ARIMA, which is an extension of the ARMA model. It incorporates an Intregration (I) factor that is important in some problems to remove the non-stationarity from the data. Also, the existence of a backward shift operator allows this model to extract some inherent features from the time series data [15].

2.2.2 Artificial Intelligence/Machine Learning based Models

Artificial Intelligence (AI)/Machine Learning (ML) algorithms are designed to study the behavior in data and are often used to implement certain learning abilities and logical reasoning without using explicit instructions [16]. They rely on the ability of the AI to learn from historical data to predict the output through further training runs. In contrast to the traditional models, methods based on AI and ML can learn complex relationships and correlations present in data [17]. This type of model includes, among others, the concept of Support Vector Machine (SVM), Decision Tree, Gradient Boosting, and Artificial Neural Network (ANN).

SVM is a widely used algorithm in pattern recognition and prediction. The core principle of SVM is to find the support vector used to construct the optimal classification hyperplane in the training set [18]. It is known for preventing the overfitting of training data and not requiring the tuning of model parameters during iterations. It is also a model suited to perform in high-dimensional data.

Extreme Gradient Boosting (XGBoost) is an ensemble method based on regression trees, especially on Gradient Boosting Decision Trees (GBDT) [19]. This algorithm first builds a certain number of weak learners, usually classification regression trees, to train weak learners. To obtain the final regression model it also performs weighted summation after training [20]. Another model based on GBDT is the Categorical Boosting (CatBoost). It is designed to handle high-dimensional data with categorical features. Some of the model features include native handling of categorical features [21], which saves time and effort in feature engineering, automatic handling of missing values, which eliminates the need for manual imputation and built-in feature importance and permutation importance calculations, as well as visualization tools for monitoring model performance [22].

ANN consists of a network of artificial neurons, connected in different layers such as input, hidden and output layers [15]. Several models were developed based on this architecture, such as Multi-Layer Perceptron Neural Network (MLPNN), consisting of a neural network with one or more hidden layers between the input and output layers. MLPNN can map the input-output relationship using proper training of the network and can be used not only for load forecasting but also for the forecast of PV generation [23]. MLPNN can solve problems with complex input and output relationships, which usually cannot be solved using a single-layer neural network.

Another model based on neural networks is the Recurrent Neural Network (RNN), which relies on a feedback system to inherit the previous time step values. It can learn and process complex and compound relationships and computational structures [8]. Different researchers proposed different RNN architectures, with the feedback loops going from the hidden layer to the input layer or from the output layer to the input layer [24, 25].

2.2.3 Deep-Learning based Models

Also called deep neural networks, these models are an evolution from the conventional ML techniques and are currently state-of-the-art in ML [8]. These techniques can learn from voluminous input data and predict the output by perceiving the patterns in the data. Compared with the other ANN-based methods described previously, Deep Learning (DL) networks present the addition of several hidden layers, for example, to learn representations of data [26].

Models such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks are among the most used and developed models. CNN uses several blocks, such as convolution, activation, and pooling, all functioning together for feature extraction and transformation [27]. LSTM networks represent an improvement of a traditional RNN, with a gated memory cell instead of the hidden unit. Apart from this memory cell, three units control the flow of information in the block. These units include an input gate, an output gate, and a forget gate [28]. This allows the LSTM to extract important information from historical data and avoid overfitting and fading [29].

Initially designed for machine translation tasks, Transformer models have also been successfully applied to various other domains, including forecasting [30]. The model architecture uses self-attention mechanisms, which enables the model to capture dependencies between different elements in a sequence without the need for recurrent connections of RNN [31].

2.3 Load forecasting - Aggregate Level

Recognizing future electricity consumption patterns is an important part of the planning, operation, and exploration of electrical power systems [32]. This way, authors in [33] implement an ANN-based model to perform day-ahead forecasts at the distribution level, considering residential, commercial, and industrial load profiles.

Load forecasting tends to present a more time-correlated behavior at the aggregate level [34] since the electricity demand in this scenario exhibits significant changes during the month but slow variations from day to day. For that reason, authors [34] consider a simple persistence model for load forecasting in a shopping mall. Despite using a simple model, the persistence exhibits good accuracy.

Other authors [35] researched weather forecasting and peak load uncertainty to improve probabilistic forecasts. To this end, the authors propose a model that consists in applying ANN for normal load forecasting and using probabilistic models of occurrence and magnitude to predict peak abnormal differential loads. The ANN-based model uses different weather and temperature forecasts, such as hourly deterministic and probabilistic forecasts, as well as calendar information. The peak load forecasting is only made with the calendar information. Then, the final load is the sum of the two predicted loads.

ML methods such as SVM present good results in household short-term load forecasting [36], especially when compared with statistical methods such as ARIMA. Statistical models cannot perceive non-linear patterns in data, while on the other hand, methods ML based are more suited to learn such non-linear relationships present in data. Despite that, the authors refer that SVM has some difficulty detecting peaks in load consumption.

In another research [37] the authors compare an SVM with ANN in load forecast. By using datasets with a different number of business days and two computation software (*R* and *Matlab*), the authors conclude that SVM can outperform ANN in terms of forecasting accuracy, despite requiring approximately the same computational time. The authors also concluded that energy consumption has considerable variations over time.

Authors in [38] suggest a short-term load forecasting method based on XGBoost that selects similar days to construct a feature map for load forecasting. A case study at the distribution level results in a lower error (9.3%) of the proposed method when compared with the traditional implementation of XGBoost (13.14%) and LSTM (14.31%).

To perform predictions at the country level, authors in [39] compare linear regressions with ML-based methods, such as Random Forests (RF) and XGBoost, with both the ML methods presenting similar results in terms of absolute and squared errors and outperforming the simple linear regression model.

A combination of LSTM and XGBoost models is presented in [40] to perform day-ahead forecasts with XGBoost presented as a tool to evaluate feature importance, with similar day detection. The key idea is to determine attribute weights using the XGBoost algorithm and calculate the distance between the selected day and the day that relies on the different measured attributes. This hybrid approach in forecasting outperformed the standard approaches of LSTM, ARIMA, and SVM.

In [41], authors implement CatBoost for medium and long-term power forecasting, proposing a two-layer model, with the first layer being the factor correlation analysis and the second layer the load forecasting model. This resulted in a model with better performance than other models, such as XGBoost and MLPNN.

To improve forecasting accuracy, authors in [42] consider a combination of Bayesian neural networks, with clustering and a linear regression model as post-processing techniques, which are subsequently applied to the output of the model to account for systematic biases and to improve the forecasting accuracy of the model. These analyses were related to the distribution feeder level and proved to be effective not only in reducing errors but also in reducing error deviation.

2.4 Load Forecasting - Appliance Level

In comparison to the forecast at the aggregate level, appliance-level forecasting was not intensively researched yet. However, some studies have been published in the last few years, some of which are mentioned next.

Device-level forecast demand at a particular time highly depends on factors such as user availability, preference, weather condition, and device settings. This way, forecasting models suffer from stochastic user behaviors and external factors that are hard to capture, resulting in a higher forecast error. This may cause an imbalance in the market due to the scheduling of false and unplanned energy demand. Therefore, authors in [43] refer that a market player is always more interested in knowing the maximum forecast error limit that can be handled in the flexibility market without further financial losses. By obtaining cost savings of 42% using forecasts with a low precision score (0.3), the authors conclude that for the device-level demand forecast, financial gain for a market is much better than implied by the error metrics.

To make day-ahead predictions on the load profile of refrigerators, authors in [44] use clustering techniques alongside Fuzzy-Rule-based and SVM models. Here, researchers use historical data relative to several months to obtain and generate the daily profiles of 20000 refrigerators. The SVM model presents a Mean Absolute Error (MAE) higher than the Fuzzy-Rule-based model.

As Section 2.2 mentions, ML and DL-based models are the most reliable methods for accurate load forecasts. This way, authors in [45] apply networks based in LSTM and compare them with other models, such as feed-forward neural networks and RF, to perform day-ahead forecasts with 10 and 15-minute intervals. This analysis considered different household appliances such as TV, fridge, laptop, electric heater, cookers, and microwaves. The LSTM outperformed the other two models, especially when considering features such as the last time the appliance was running and the last time the appliance was off.

2.5 Load Forecasting - Application in Industrial Kitchens

There are very few research works on load forecast in IK. In [46] the author uses models such as Prophet, RF, and LSTM to perform hour-ahead forecasts for appliances in one Kitchen, as a way to evaluate if the sum of the errors of the individual forecasts of the appliances presents a lower error than forecasting the sum of the appliances' consumption (described as a *Virtual Aggregate*). It is concluded that the forecast of the *Virtual Aggregate* presents a lower error for every model except Prophet, with the RF model presenting the better accuracy to perform both the individual forecasts and the *Virtual Aggregate* forecast.

3

Appliance Level Load Forecasting - Proposed Methodology

Contents

3.1 Data Pre-Processing 19
3.2 Forecasting Algorithms 22
3.3 Model optimization 24
3.4 Data Post-processing 27
3.5 Performance evaluation 31

This chapter explains the methodology used to achieve accurate day-ahead and hour-ahead forecasts, from the input data to the final results. It consists of five topics: pre-processing of data, implementation of forecasting models, model optimization, post-processing of data and evaluation metrics for validation. Figure 3.1 presents a visual description of the methodology considered for this chapter.

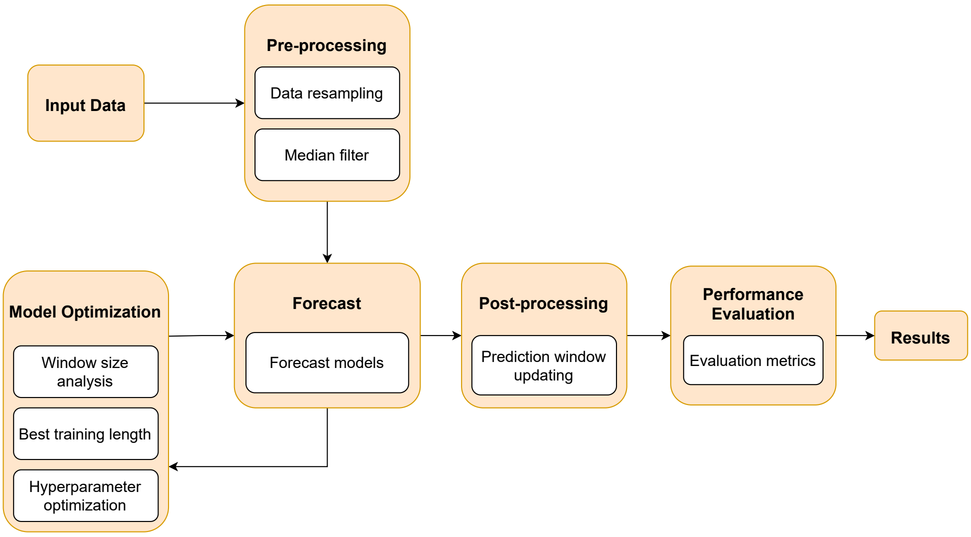

Figure 3.1: Proposed methodology for forecasting

The pre-processing stage includes data analysis and treatment tools such as re-sampling and filtering, to present data at an adequate sampling rate for the required forecasts and to remove noise, outliers, and negative values. The main purpose of the application of these tools is to provide quality data as input for the forecast models.

Model performance is dependent on different factors such as hyperparameter optimization, best historic size, and best training length, which require different analyses to achieve better results. The resulting predictions will be post-processed to improve the quality of the forecasts, and lastly, a set of evaluation metrics will be used to validate the performance of the proposed methodology.

The following sections detail the different topics of the presented methodology. In Section 3.1, the pre-processing tools and methods are implemented. Section 3.2 it is presented an overview on the forecast models used in this work, and the respective optimization is explained in Section 3.3. Finally, the post-processing of predictions and the validation metrics are presented in Sections 3.4 and 3.5 respectively.

3.1 Data Pre-Processing

Data pre-processing consists in applying techniques to convert the raw data into data with a consistent and understandable shape [47]. In this work, the input data has a very high sampling rate, in the order of seconds. For hour-ahead but especially for day-ahead forecasting, this adds another challenge since it becomes clear that it implies the forecast of a large number of samples. Ultimately, this represents a challenge for both forecasting accuracy and computational resources, mainly in the time to train the learning models. To minimize this problem, data is re-sampled to a higher period accordingly to the type of prediction, namely thirty minutes for day-ahead forecasts and five minutes for hour-ahead forecasts implying a forecast of 48 samples for day-ahead predictions and 12 samples for hour-ahead, to fulfill the objectives of the project set in Chapter 1. This is done considering the mean value of the samples in the original data, for each re-sampling period.

3.1.1 Missing Data and Filtering

Missing data is always a matter of great importance when using forecast models. From the load forecasting perspective, the ideal situation is to have a smooth stream of data values feeding into the forecast model. Records with missing, irregular, or incomplete data can have a significant impact on the accuracy of the prediction. Hence, abnormal records must be completed, corrected, or eliminated so that the ultimate predictive model is as accurate as possible [48]. The data used in this work was already cleansed to be used in other works in the scope of the *nexIK* project [49]. Nonetheless, re-sampling data to a constant frequency can cause some missing data to appear. This way, and to avoid considerable changes to the patterns of the original data, the missing data points caused by resampling were dropped.

Smoothing filters are used for noise reduction, and there are various methods that can be applied. For this study, a median filter from the *scipy* library was used to remove some noise created by changing the sampling rate of data and to smooth the signal. Median filters have excellent noise-reduction capabilities for certain types of noise, give less blurring than linear filters, and are particularly effective for impulse noise reduction [50]. This particular technique applies a filter with a window size k that represents the number of adjacent data points used to compute the median in the filtered signal. In this work, a window size of 7 was used, corresponding to the median being computed with three points backward, the point itself, and three points forward. In other words, one hour and a half for each direction for day-ahead scenarios, and fifteen minutes in each direction for hour-ahead scenarios.

After inspecting the original datasets to be used, it was noticed that some had negative load values, a characteristic that does not correspond with a device's behavior since an electric appliance cannot supply energy to the grid. This way, a simple filter of negative loads was applied to replace negative load values with zero values.

3.1.2 Sliding Window and Training Length

To perform the forecasts, the models require the data to be separated into features and targets. Authors in [51] use the method of sliding windows to reconstruct the data into a multidimensional matrix containing features and targets to smooth the non-linearity and non-stationary behavior of load time series.

For this work, the features considered are the historical values of power and the targets, the load values to be predicted, but separated into two different matrices. Using data re-sampled, it results in a feature dataset comprising sequences of the previous W_{hist} load values, while the targets comprise sequences of the next W_{pred} values. This implies that the resulting sets of data will consist in 2D matrices, where the number of rows corresponds to the number of periods and the number of columns consists in the number of features (for the feature set) or the number of targets (for the target set). An example of how these datasets were generated for day-ahead forecasting with two days of historical measurements can be seen in Figure 3.2, with W_{hist} = 96 and W_{pred} = 48.

Features

	1	2	3	4				95	96
Sequence x	00:00 15/02/2019	00:30 15/02/2019	01:00 15/02/2019	01:30 15/02/2019	...	...	...	23:00 16/02/2019	23:30 16/02/2019
Sequence x+1	00:30 15/02/2019	01:00 15/02/2019	01:30 15/02/2019	02:00 15/02/2019	...	...	...	23:30 16/02/2019	00:00 17/02/2019
Sequence x+2	01:00 15/02/2019	01:30 15/02/2019	02:00 15/02/2019	02:30 15/02/2019	...	...	...	00:00 17/02/2019	00:30 17/02/2019

Targets

	1	2	3	4				47	48
Sequence x	00:00 17/02/2019	00:30 17/02/2019	01:00 17/02/2019	01:30 17/02/2019	...	...	...	23:00 17/02/2019	23:30 17/02/2019
Sequence x+1	00:30 17/02/2019	01:00 17/02/2019	01:30 17/02/2019	02:00 17/02/2019	...	...	...	23:30 17/02/2019	00:00 18/02/2019
Sequence x+2	01:00 17/02/2019	01:30 17/02/2019	02:00 17/02/2019	02:30 17/02/2019	...	...	...	00:00 18/02/2019	00:30 18/02/2019

Figure 3.2: Example of feature and target sequences - day-ahead

Straight after, the data is divided into training and test data. However, to avoid train targets being included in the test set as features, a buffer zone in data was created between the end of the training set and the beginning of the test set, with the size of one historic window. This way, there is no risk that some information from the training set ends up in the test set, with the possibility of biasing the model. Then, this data is used as input for the forecasting models.

In this work, the load values were re-scaled in the interval [0,1] for the LSTM network, allowing a faster convergence for the model, to avoid numeric instability and to better calculate the weights of the network. This was not done for the GBDT models because these algorithms are not sensitive to the variance in the data [19].

3.2 Forecasting Algorithms

In this work, the forecast algorithms used are **XGBoost**, **CatBoost**, and **LSTM**, which based on the good results mentioned in Chapter 2 for other forecasting studies, present the best chances of performing good quality forecasts. A detailed explanation of the algorithms is provided next.

3.2.1 XGBoost

Extreme Gradient Boosting, as the name suggests, uses gradient boosting algorithms combined with decision trees to achieve high accuracy and efficiency in a variety of tasks, such as classification and regression, and is particularly effective for high-dimensional datasets, where there are many features. In this algorithm, decision trees (also called weak learners) are created in sequential form, perform a prediction and sequentially add them to the model [19]. By iteratively adding decision trees to the model, the final result is a strong, upgraded predictor. The mathematical formulation for XGBoost is detailed in Equation (3.1)

$$\hat{y_i} = \sum_{k=1}^{K} f_k(x_i), f_k \in F \tag{3.1}$$

where K is the number of trees, f is the functional space of F, being F the set of possible decision trees.

In XGBoost, each new learner is trained to minimize the residual error of the previous model. This is done by optimizing an objective function, which measures the difference between the predicted and actual values of the target variable [19]. One key feature of XGBoost to optimize the objective function is the use of regularization to prevent overfitting. It can be applied to the parameters of the model, as well as to the weights of the individual decision trees. This helps to ensure that the model generalizes well to new data. The formulation of the objective function with regularization terms is presented in Equation (3.2) [52].

$$obj(\theta) = \sum_{i}^{n} l(y_i, \hat{y_i}) + \sum_{k=1}^{K} \Omega(f_k) \tag{3.2}$$

where the first term is the loss function and the second is the regularization parameter. To perform forecasts with XGBoost, the *xgboost* library in *Python* is used.

3.2.2 CatBoost

Categorical boosting has many similarities with XGBoost, but also some significant differences. One key difference between XGBoost and CatBoost, as already mentioned in Chapter 2, is the way both handle categorical variables. XGBoost requires that categorical variables be pre-processed into numerical features before training, while CatBoost can handle categorical features directly without the need for pre-processing [21].

Despite having only numerical features in the data, it might be useful to interpret them as categorical for CatBoost, as it uses a new schema for calculating leaf values when selecting the tree structure, which helps to reduce overfitting [21]. To handle these features, CatBoost introduces two additional terms: the categorical feature index j, and the categorical feature value v. These terms are used to compute the split gain for each node of the decision tree, as explained below.

$$gain = (loss_{left} + loss_{right} - loss_{parent}) - regularization \tag{3.3}$$

In Equation (3.3), the terms $loss_{left}$, $loss_{right}$ and $loss_{parent}$ represent the losses for the left and right child nodes and parent node, respectively. The split gain is calculated for each categorical feature, and the feature with the highest gain is selected for the split [22]. This process is repeated until a stopping criterion is met, such as maximum tree depth, for example. For the implementation in the *python* environment, the *catboost* library is used.

3.2.3 LSTM

As mentioned in Chapter 2, LSTM is one of the most used models in deep learning applications, including load forecasting. LSTMs were introduced to overcome the problem of vanishing gradient in RNNs by incorporating a special memory cell that can selectively remember or forget information over time. The memory cell is composed of three gates [28]:

- Input gate: determines which new information to add to the cell state based on the current input and the previous hidden state
- Forget gate: determines which information to discard from the cell state based on the current input and the previous hidden state
- Output gate: determines which information to output from the current cell state to the current hidden state based on the current input and the previous hidden state.

The input to the LSTM is a sequence of vectors, such as a time series, and the output is also a sequence of vectors, the predicted values. The LSTM layers can be stacked to increase the depth of the network and improve its ability to capture complex dependencies in the input sequence. Still, it also increases the processing time of the network.

The schematic of an LSTM cell is presented in Figure 3.3, and a mathematical formulation of the nodes present in a cell is analytically described in Equations (3.4) to (3.8).

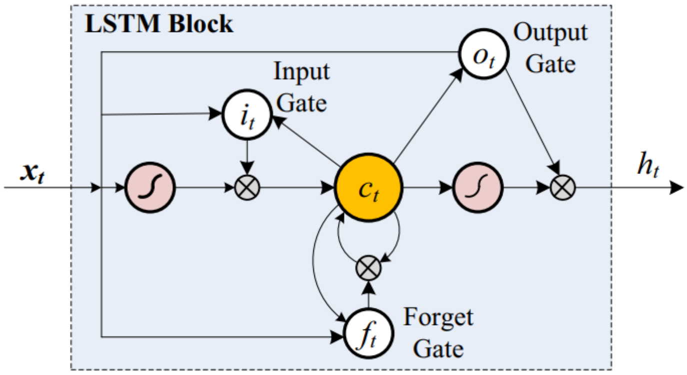

Figure 3.3: LSTM block architecture (Source: [53])

$$i_t = \sigma(W_{xi}X_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{3.4}$$

$$f_t = \sigma(W_{xf}X_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{3.5}$$

$$c_t = f_t c_{t-1} + i_t\, tanh(W_{xc}X_t + W_{hc}h_{t-1} + b_c) \tag{3.6}$$

$$o_t = \sigma(W_{xo}X_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{3.7}$$

$$h_t = o_t\, tanh(c_t) \tag{3.8}$$

where c_t, i_t, f_t, and o_t are respectively the state cell, input gate, forget gate, and output gate at the time instant t. Also, the final output from the LSTM unit is represented as h_t. x_t is the current sample and the activation function σ is the sigmoid function.

During training, the LSTM parameters are updated using backpropagation [54], which involves computing the gradients of the loss function with respect to the network parameters, over multiple time steps. This algorithm can be computationally expensive, especially for long sequences. *Python*'s library *keras* was used in this work to perform the training and forecasting with LSTM.

3.3 Model optimization

In this section, different optimization techniques are applied to the forecasting models, to achieve the best possible performance. Section 3.3.3 provides an overview of the hyperparameter optimization of both types of models used in this work. The horizon size and training length are also important when it is desired to obtain the best performance possible, so both topics are explored in Sections 3.3.1 and 3.3.2 respectively.

3.3.1 Window Size Analysis

The historical window size is crucial to feed the model data from which it can learn the patterns of load consumption. Authors in [55] perform a sensitivity analysis ranging from 1 to 30 days of historic window size and identify the optimal length as 5 days for day-ahead forecasts. Generally, the accuracy of the prediction model improves as the size of the training window increases, i.e. it is better to have sizeable historical data for training the prediction model so that it covers all possible patterns spanning time series [56]. However, a significant drawback of having large historical data is if the behavior of the data changes quickly, the trained model may not track or ignore those changes, resulting in inaccurate forecasts. It is also expected that due to the different types of appliances studied in this work, some may require a larger historical window to properly identify patterns, while other appliances might perform better considering small windows.

In this work, the maximum window size for the day-ahead scenario was set to three days, mainly because of the short dimensions of the available datasets. However, for hour-ahead forecasts, there is the possibility of implementing the models with a wider range of historic periods since the prediction window is much smaller. This way, the values chosen to analyze the historic window size influence in the forecast performance are presented in Table 3.1.

	Day-ahead	**Hour-ahead**
Horizon size (hours)	24, 48, 72	2, 4, 6, 8, 12, 24
Number of samples	48, 96, 144	24, 48, 72, 96, 144, 288

Table 3.1: Historic window size values

3.3.2 Training Set Length

The main goal of this topic is to find the training length that better suits the forecast models. To evaluate the model performance for different training lengths, cross-validation-based techniques were applied in this work. Cross-validation is a widely used forecast evaluation technique to avoid the potential overfitting issues [57].

A widely used approach is based on expanding window cross-validation [58]. This technique performs cross-validation by fixing the length of the test set and expanding the length of the training set in each iteration [58]. A simplified representation of this technique is presented in Figure 3.4, with the training set in blue and the test set in red.

Another approach is the V-fold cross-validation, where it first partitions the data into V equally sized segments [59]. One of the V segments is used as validation data, and the rest V-1 segments are used as training data. This process is repeated V times without replication of the validation dataset. The performance of the model is usually evaluated based on its average performance across the V segments.

Since the order of data points is crucial for the forecasts, the training sets can only correspond to observations that occurred prior to the observations that form the test set. This way, it was chosen the expanding window technique with the training set expanding in 4 iterations instead of the V-fold, to find the size of the training subset that results in a lower error for the forecasts.

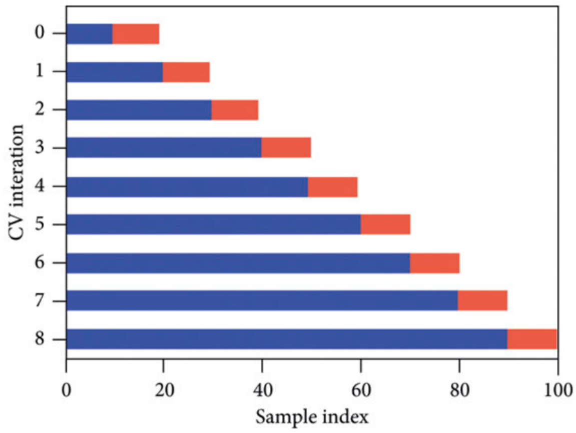

Figure 3.4: Expanding window cross-validation (Source: [60])

3.3.3 Hyperparameter Optimization

ML and DL algorithms rely on hyperparameters to perform the forecasts that are not learned during the training process but defined beforehand, which can significantly affect the performance. Thus the values of these hyperparameters must be selected carefully [61]. For decision tree-based and neural network-based algorithms, the parameters in analysis and the respective ranges of values are presented in Table 3.2.

Hyperparameter optimization aims to find the combination of hyperparameters that yields the best performance on the given dataset. This is typically done by performing a search over a range of possible values, using techniques such as grid search, random search, or Bayesian optimization [62]. The model's performance is evaluated using a validation set, and the hyperparameter values that yield the best performance are selected.

In this work, Bayesian optimization is used to tune the ML and DL models. This type of optimization involves constructing a probabilistic model of the objective function, which is updated as more evalua-

Algorithms	Parameters	Range of Values
Decision Tree based	Learning rate	0.005, 0.01, 0.02
	Maximum depth of each tree	2, 3, 4, 5
	Column sample by tree	[0.2, 0.8]
	Subsample	[0.2, 0.8]
	Regularization parameters λ, α	1, 5, 10, 50
LSTM	Number of neurons in each layer	16, 32, 64
	Number of hidden layers	1, 2, 3
	Activation function	tanh, linear, sigmoid
	Dropout in each neuron	0, 0.1, 0.2

Table 3.2: Hyperparameters for optimization

tions are performed. The model is then used to select the next set of hyperparameters to be evaluated based on the expected improvement in performance. To perform this optimization, the *Optuna* [1] framework is used. *Optuna* provides a high-level interface for defining and running search experiments and a wide range of search algorithms for finding the best set of hyperparameters [63].

Apart from the parameters tuned using the *Optuna* library, for the GBDT-based algorithms, the number of trees to be built was defined as *n_estimators* = 3000. For the LSTM model, two deeply connected layers were also added, with the first layer presenting 32 neurons and the last layer presenting a number of neurons corresponding to the number of targets of the analysis. Also, the number of epochs for the LSTM to run was defined as 50. Parameters such as *early stopping rounds = 50* for GBDT algorithms and checkpoint callbacks for LSTM were also used to reduce the training time of the models.

The hyperparameter optimization of the algorithms was computed separately from the main algorithm, mainly because it is a very computationally expensive and time-consuming task [64]. After optimizing each model and dataset, the hyperparameters providing the best performance were selected and placed on the main algorithm models.

3.4 Data Post-processing

After getting the predicted values, some methods were applied to perform corrections and enhance the obtained results. Negative power values should not exist, so the first step was to apply a filter to eliminate negative values from the predictions, as was done with the input data.

Since the forecast algorithms are trained for multiple targets at once, the output of the forecast models consists also of multiple targets, forming a 2D matrix, with each line containing the forecast of the next W_{pred} values. This means that the predicted windows overlap and consequently, there is more than one prediction for each timestamp. With this, it is possible to update the predicted values at each timestamp, based on techniques that consider the different values available for each t. The methods considered in

[1] https://optuna.org/

this work are:

- Most recent prediction for timestamp t
- Persistence-based prediction for timestamp t
- Average value of predictions for timestamp t
- Weighted predictions for timestamp t

3.4.1 Most Recent Prediction

Using the most recent prediction for a timestamp is straightforward assuming that the prediction for a point in time t is given by the last value predicted for time instant t. In other words, it is the prediction made at the time instant $t-1$, which in theory, represents the prediction with the most up-to-date information. An example on how the predicted value is selected with this approach is presented in Figure 3.5.

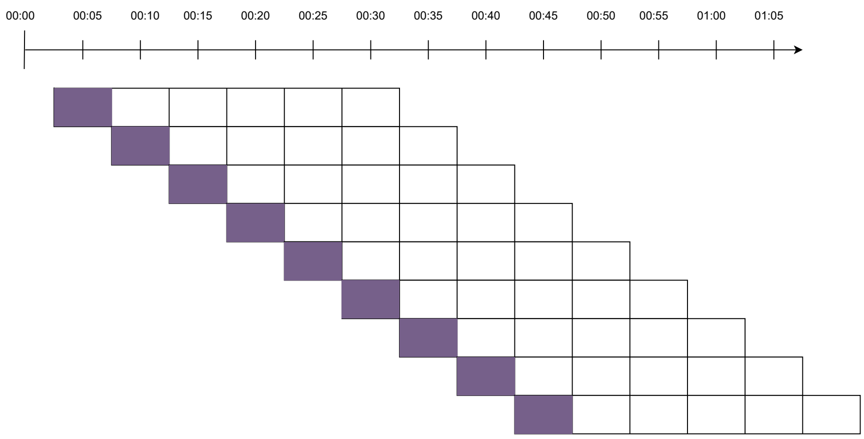

Figure 3.5: Example for Most Recent Prediction

3.4.2 Persistence Based Prediction

The optimization using a persistence model assumes that for each prediction window, the results of the forecasting model present a trend in the results and that the same iteration of prediction presents a similar error for the timestamps in the different windows. In other words, it assumes that, for example, the first prediction for the time instant t presents a similar error in relation to the first prediction for the time instant $t + W_{pred}$. Hence, this method analyzes the first W_{pred} timestamps and finds which prediction presents the lowest error for each t. For $W_{pred} + t$, it is assumed that the most accurate prediction is at the same relative distance in time as it was for the first W_{pred} points analyzed. An example on how the predicted value is selected with this approach is presented in Figure 3.6, considering a simple scenario with W_{pred} = 6.

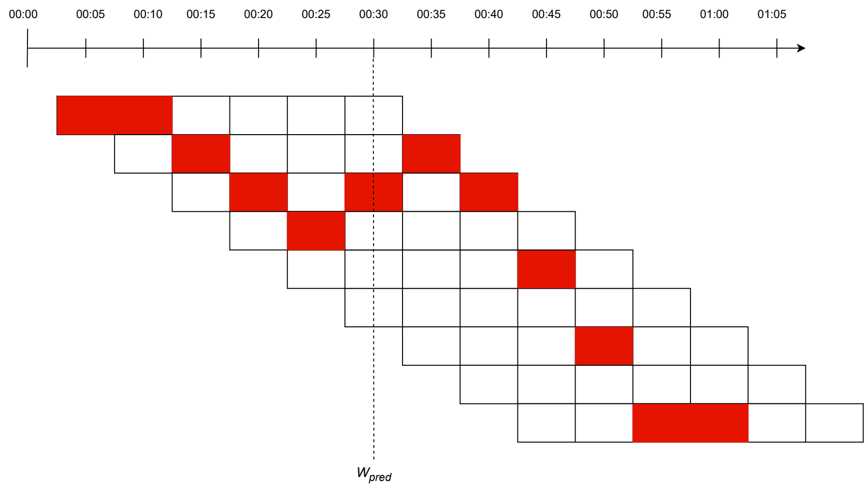

Figure 3.6: Example for Persistence-based Prediction

Here it is possible to understand the pattern selected in the predictions. With the first two timestamps being obtained by selecting the oldest values available (for 00:05 and 00:10), the first two timestamps after W_{pred} (00:35 and 00:40) are also obtained by selecting the oldest prediction available.

3.4.3 Average Value Prediction

The algorithm to compute the average predicted value for each timestamp considers how the sequences of predicted values are presented. The predicted values shape presents the same dimensions as the target values for the test set. By re-inspection of Figure 3.2, it is visible that not only the first and last timestamps are the only points that present just one value of predictions, but also that it is necessary to compute the average value on the diagonals of the matrix to obtain the average value for each timestep. The algorithm developed for this effect is presented in Algorithm 3.1.

3.4.4 Weighted Prediction

A method based on a weighted average was also developed to simulate a memory function that can retain past error values for better accuracy of predictions. This method is based on the Roth-Erev algorithm, presented in Equation (3.9), that considers two features: the previous coefficient value and the error made in the last prediction [65]. The algorithm uses a weight factor W that works as an inertia constant with a value between 0 and 1. It is regulated based on the dependence of each appliance on the past values to predict the future load accurately. The closer the weight value is to 1, the larger

Algorithm 3.1: Average predicted value

begin

  Input: $predictions$

  Define avg_val

  for $\underline{line}$ **in** $\underline{range(1, predictions.shape[0] + predictions.shape[1])}$ **do**

    $start\_col$ = $max(0, line - predictions.shape[0]$

    $count$ = $min(line, (predictions.shape[1] - start\_col), predictions.shape[0])$

    $aux\_iteration$ = []

    for $\underline{j}$ **in** $\underline{range(0, count)}$ **do**

      $aux\_iteration.append(predictions[min(predictions.shape[0], line) - j - 1][start\_col + j])$

  $avg = mean(aux\_iteration)$

  $avg\_val.append(avg)$

  Return: avg_val

the dependency on the past values. This methodology was already explored in [66], where a reinforcement learning mechanism was developed to choose the most appropriate strategy for each moment in Electricity Markets. Three methods were presented to predict the best strategy for the next day in the Electricity Market, with one of those methods being the Roth-Erev algorithm.

$$C_{t+1}^{n} = C_{t}^{n} \times W - |(R_t - P_t^n)| \times (1 - W) \tag{3.9}$$

In that work, the Roth-Erev algorithm obtains the most accurate prediction for a given instant t. It uses the coefficient as a memory for the errors between predictions and the real value. The flowchart representative of this methodology is presented in Figure 3.7. In Equation (3.9), coefficients C_{t+1}^{n} and C_t^n are one-dimensional vectors, with C_t^n representing the mean errors between each prediction position n for the historical values until t and test values. This way, C_{t+1}^{n} is also a 1-D vector, where its positions n are determined individually with the historical and actual error values. Each position represents the prediction number of the respective time instant: if position n yields the lowest error for t, the n_{th} prediction made for t is the most correct one. Then, this index is saved and used to set the n_{th} prediction as the most accurate value for $t+1$. This implies that the final prediction for time instant t can never present an error lower than the lowest error prediction for the same instant. The error values for $|(R - P_t^n)|$ are then added to the historical component, while the errors between the set of predictions and the test value for $t+1$ represent the actual error to get the prediction for $t+2$. This process is carried out until the end of the predictions' data is reached.

With the final prediction being a function of two components, the sum of the weight values that multiply by these components must be enforced to 1. However, the appliances are expected to have different dependencies of the past and actual error values. This way, a sensitivity analysis was performed about the weight parameter to understand which values better suit each appliance. The values chosen for the sensitivity analysis are presented in Table 3.3.

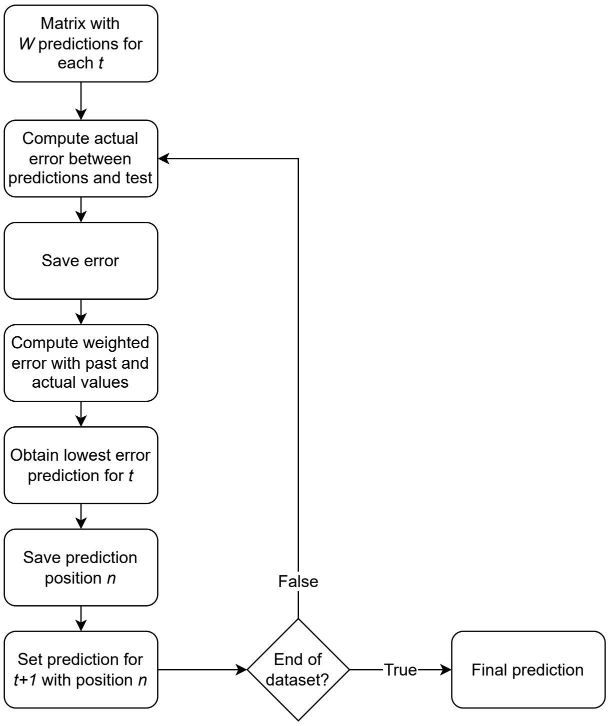

Figure 3.7: Flowchart for weighted predictions

	W
Values	0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95

Table 3.3: Set of weights for weighted predictions

3.5 Performance evaluation

This section presents techniques used to evaluate the performance of the proposed methodology. The predictions were evaluated in different stages: at the output of the forecast models, to provide an overview of the forecast quality, during the model optimization stage to obtain the best hyperparameters, historic window size, and training length, and finally, after the post-processing techniques, to provide a more conclusive result about the methodology. The metrics were computed with the average error for all prediction windows, instead of being computed only for each point in time.

To quantify the errors, three different metrics were used. The simplest metric consists in computing the MAE of each prediction compared to the test value. Another metric that is often used in load forecasting is the RMSE [67], which emphasizes peaks in error, and the relation between RMSE and MAE provides insight into the distribution of the error along the predictions. A normalized version of RMSE concerning the maximum value on the respective dataset is also presented as NRMSE. The corresponding expressions of the metrics are presented in Equations (3.10) to (3.12).

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y(t) - \hat{y}(t)| \tag{3.10}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y(t) - \hat{y}(t))^2} \tag{3.11}$$

$$NRMSE = \frac{1}{y_{max}}\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y(t) - \hat{y}(t))^2} \tag{3.12}$$

In these equations, $y(t)$ stands for the truth value of power at the instant t, while $\hat{y}(t)$ is the predicted value for the same time point, y_{max} is the maximum value on the dataset (determined after the pre-processing of data), and N represents the number of values.

4

Appliance Level Load Forecasting - Performance Evaluation

Contents

4.1 Dataset Description 35
4.2 Window Size Analysis 39
4.3 Best Training Length 40
4.4 Hyperparameter Optimization Results 42
4.5 Day-ahead Results 42
4.6 Results - Hour-ahead 45
4.7 Training Times 47
4.8 Post-processing Results 48
4.9 Discussion 52

This chapter provides the results obtained in the case study, used to evaluate the proposed methodology presented in Chapter 3. In Section 4.1 a detailed description of the datasets used is given. In Sections 4.2 and 4.3 respectively the results for the historic window size analysis and training set length are presented. Sections 4.5 and 4.6 show the final results of the methodology for both day-ahead and hour-ahead scenarios. In Section 4.7 it is analyzed the training time of the models mentioned previously, and in the last section the results will be discussed and compared.

4.1 Dataset Description

In this work, the datasets used are from two different kitchens in Funchal, Portugal, monitored during the Future Industrial Kitchen project [68]. These kitchens have several appliances with different characteristics. The electricity consumption was measured using the e-Gauge at a sampling rate of 1/5 Hz. The period of measurement for the first kitchen ranges from the sixth of February of 2019 (6^{th} of February 2019 00:00:00) to the fourth of March at 08:47 (4^{th} of March 2019 08:47:17). Since the last six hours of these datasets have zero values, the period considered goes only to the 03:00:00 of the 4^{th} of March. For the second Kitchen, the period of measurement starts on the eleventh of March 2019 at 11:20 (11^{th} of March 2019 11:20:10) and ends on the third of April of 2019 at 08:45 (3^{rd} of April 2019 08:45:59). The description of each kitchen is presented in Table 4.1.

Kitchen	Area [m^2]	Seat Capacity	Operation Type
Kitchen 1	58.15	50	7 days a week for dinner service Some occasional breakfast services
Kitchen 2	25.52	50	7 days a week for dinner service

Table 4.1: Description of each kitchen

The datasets contain information about electric measurements such as voltage, current, reactive, active, and apparent power. In this work, only active power is used, mainly because it is pretended to forecast real power values for the appliances. In both kitchens, the active power is given in Watts (W). As an illustration of the data that will be used, Figure 4.1 shows a plot with the active power demand from the Induction Plate, one of the appliances in K2.

Re-sampling data can originate some NaN values, and so the number of NaN values originated was measured, with the information on these values being presented in Table 4.2 and Table 4.3. This analysis is important to understand if it is necessary to apply interpolation techniques.

Kitchen	Number of samples	Null values
Kitchen 1	1254	28
Kitchen 2	1074	13

Table 4.2: Null values generated due to re-sampling to 30 minutes

Kitchen	Number of samples	Null values
Kitchen 1	7524	196
Kitchen 2	6442	105

Table 4.3: Null values generated due to re-sampling to 5 minutes

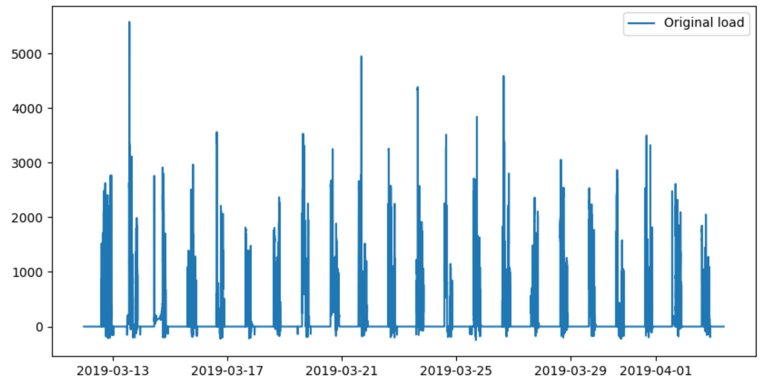

Figure 4.1: Original load data of Induction Plate

The missing data points represent only around 2.5% of the total data in kitchen 1 (K1) and 1.5% in kitchen 2 (K2). More important than this is that, for K1, the missing data consists of the majority of one specific day (12-02), so removing approximately one day of data is not expected to affect the overall data quality. For K2, the missing data is spread out evenly on the dataset's days; thus, removing these points is not expected to affect the data patterns on each day. Also, by this analysis, it is possible to notice that the NaN values originated by re-sampling are just a small fraction of the dataset. This way, the missing values were dropped.

4.1.1 Results from pre-processing

This subsection presents the results of implementing pre-processing techniques mentioned in Section 3.1. To provide details on the data after pre-processing, characteristics as the maximum value and the maximum value after re-sampling after filtering the data are presented in Table 4.4. This second characteristic shows how re-sampling data to different sampling periods and filtering affects the maximum power values used to compute the NRMSE metric. Obtaining an approximated value of each appliance's "nominal" power is also useful since the original load can be affected by phenomena such as transient states.

Kitchen	Device	Max Power [W]	Max Power (30 minutes) [W]	Max Power (5 minutes) [W]
Kitchen 1	Blast Chiller	5316	925	1202
	Convection Oven 1	8797	1551	4946
	Convection Oven 2	8337	1984	7092
	Dish Washer	4451	1406	3484
	Drinks Refrigerator	623	416	430
	Dual Fryer	5191	496	3116
	Fish Refrigerator	3709	260	439
	Freezer	1319	713	724
	Garde Manger 1	1034	162	296
	Garde Manger 2	418	66	114
	Ice Machine	421	315	366
	Infrared Shelf	1132	885	889
	Meat Refrigerator	1274	213	492
	Mise en Place	854	392	729
	Salamander 1	3148	2969	3003
	Salamander 2	3180	2976	2997
	Vegetables Refrigerator	897	158	397
Kitchen 2	Blast Chiller	1614	351	407
	Convection Oven	6218	1714	5808
	Exhaust	609	357	360
	Heating Lamp 1	4451	1406	71
	Heating Lamp 4	247	234	239
	Hothouse	2541	1464	2455
	Induction Plate	5575	1013	2322
	Microwave	3323	28	105
	Refrigerator 1	758	200	390
	Refrigerator 2	743	95	217
	Refrigerator 3	724	128	407
	Refrigerator 4	827	147	318
	Sous Vide	2203	1029	2104

Table 4.4: Data characteristics after pre-processing

For some appliances, and especially for the larger sampling period, the maximum power obtained from re-sampling and filtering techniques presents a value much lower than in the original data. One factor that has a large influence is the period of operation of the appliances: while some present high and constant periods of operations (such as Salamanders, Freezers, and Exhausts), others present very small periods while switched ON (in the order of a few minutes), and for a day-ahead analysis with a sampling rate of 30 minutes these periods of operation are absorbed by the mean value used to define each re-sampled point of data.

Figure 4.2 shows the original load values of one appliance (Exhaust - K2), at the original sampling rate, with negative values and visible noise. After the pre-processing methodology, the re-sampled and filtered data that will be the input of the forecast models are presented in Figure 4.3.

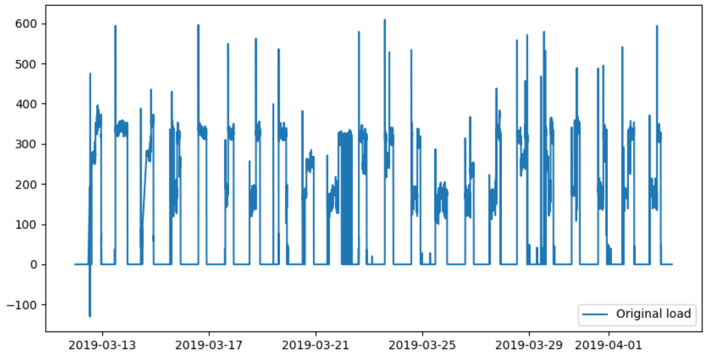

Figure 4.2: Original load data of one appliance - Exhaust

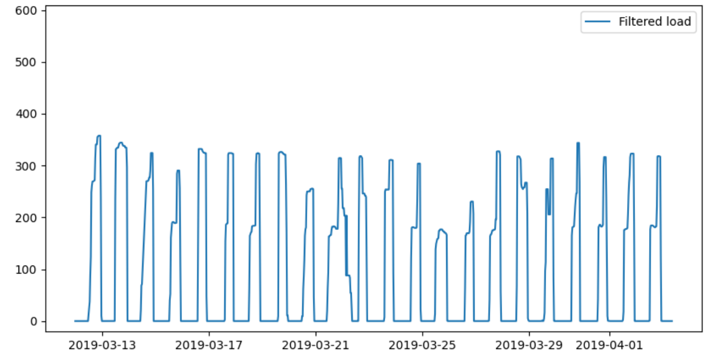

Figure 4.3: Processed load data of one appliance - Exhaust

The processed load shows no negative values and much lower noise due to the re-sampling to a lower frequency and the filtering with the median values. This results in data with recognizable patterns for each day, which allows the forecast models to perform more accurate forecasts for a day-ahead analysis. Also, the original load presents peaks that are most likely caused by transient states in the electric components of the appliances. Load filtering removes these values, which are not the main target for the forecasts, but instead the load's overall behavior.

For the 5 minutes sampling period, there are appliances with very considerable differences between the maximum power before and after pre-processing. However, for this analysis, the number of appliances in which this situation is verified is fewer than for the day-ahead scenario. This happens because with the 5-minute resolution used in this situation only the appliances with very short periods of operation are affected by the sampling rate. This is the case with Garde Mangers for example, which have an operating period of around 3 minutes. With this, it is possible to conclude that some appliances are very sensitive to the sampling rate used and that it directly affects the forecasting quality. Figure 4.4

and Figure 4.5 present the influence of the pre-processing techniques for the Salamander 1 - K1, in the 5-minute scenario.

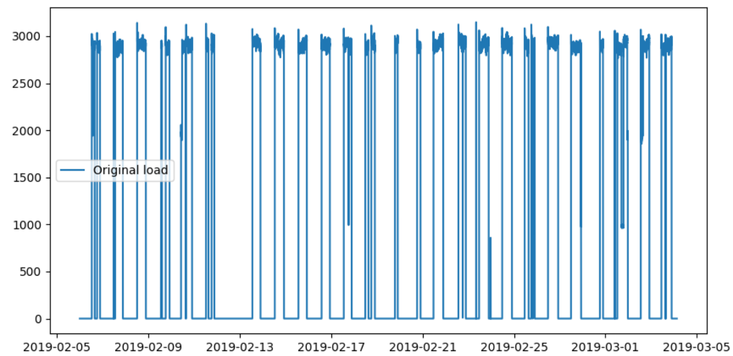

Figure 4.4: Original load data of one appliance - Salamander 1

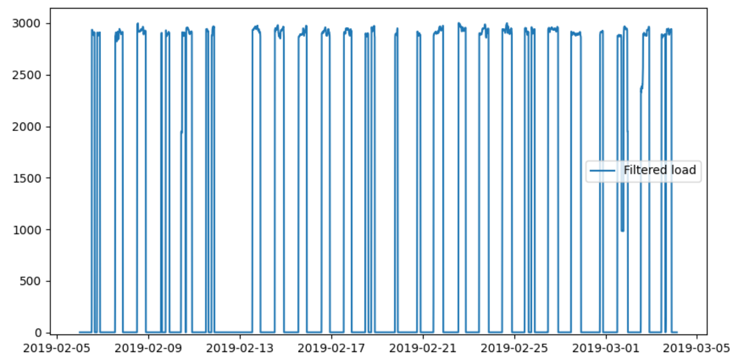

Figure 4.5: Processed load data of one appliance - Salamander 1

4.2 Window Size Analysis

This section presents the results of the methodology described in Section 3.3.1. For this analysis, the models' hyperparameters were kept constant and unchanged for all analyses, while the training size was set as the first 19 days for K1 datasets and the first 18 days for K2. These values represent the maximum training size to fulfill the requirements set for the test set length mentioned in Section 3.1.2. Tables 4.5 and 4.6 present the results obtained for the different historic window sizes, for the day-ahead and hour-ahead scenarios, respectively.

Kitchen	Window size	XGBoost		CatBoost		LSTM	
		NRMSE	StD	NRMSE	StD	NRMSE	StD
Kitchen 1	24h	16.05%	17.81%	15.45%	17.53%	**25.94%**	10.84%
	48h	**15.52%**	17.59%	**15.27%**	17.47%	26.26%	10.92%
	72h	16.07%	17.94%	15.56%	17.75%	27.88%	10.58%
Kitchen 2	24h	**13.95%**	21.02%	**13.83%**	20.88%	**20.67%**	18.45%
	48h	14.98%	20.17%	14.93%	20.25%	20.80%	18.06%
	72h	15.03%	14.35%	14.99%	13.57%	21.27%	12.62%

Table 4.5: Historic window size analysis - day-ahead

Kitchen	Window size	XGBoost		CatBoost		LSTM	
		NRMSE	StD	NRMSE	StD	NRMSE	StD
Kitchen 1	2h	10.15%	14.79%	9.80%	14.59%	13.75%	15.90%
	4h	10.10%	14.94%	9.77%	14.75%	14.35%	15.69%
	6h	10.19%	14.85%	9.82%	14.68%	14.06%	15.95%
	8h	10.32%	14.88%	9.91%	14.70%	12.56%	15.69%
	12h	10.15%	15.43%	9.74%	15.35%	**11.42%**	16.30%
	24h	**9.65%**	15.84%	**9.19%**	15.70%	14.97%	16.60%
Kitchen 2	2h	4.79%	12.63%	4.64%	12.41%	5.26%	13.80%
	4h	4.69%	12.67%	4.54%	12.43%	**4.90%**	13.57%
	6h	4.71%	12.76%	4.6%	12.67%	5.10%	14.17%
	8h	4.77%	12.92%	4.66%	12.75%	5.75%	14.66%
	12h	4.83%	13.11%	4.71%	13.08%	5.51%	15.07%
	24h	**4.07%**	13.50%	**3.99%**	13.15%	6.44%	14.64%

Table 4.6: Historic window size analysis - hour-ahead

From the day-ahead analysis is possible to conclude that there are two different horizon lengths that better suit the forecast models, the 48-hour horizon for K1 and the 24-hour horizon for K2. For both kitchens LSTM model is showing worse performance and the GBDT models are performing with a similar accuracy. This way, the results of these models are taken more into account than the LSTM results, and the horizon windows are set as 48 hours for K1 and 24 hours for K2.

As for the day-ahead scenario, hour-ahead results for the GBDT models present similar results throughout the different horizons tested, with the 24-hour historic window size having the lowest error. LSTM model presents different optimal window sizes than the other models but with larger errors for both kitchens and for every window size. With this analysis, the historic window size was set as 24 hours for every model in K1 and K2 in the hour-ahead scenario.

4.3 Best Training Length

For kitchen 1 the input datasets consist of approximately 26 days of data, with 5 days set for the test period. For kitchen 2, the input data consists of 22 days of measurements, with 3 days set for testing. The kitchens were set with different test periods to verify the robustness of the methodology.

To obtain the best training length for the forecast models, the datasets of both kitchens were partitioned into four equal subsets, with the test sizes fixed. Splitting the datasets resulted in training subsets with a size of approximately 4 days and 18 hours for kitchen 1 and 4 days and 12 hours for kitchen 2.

This analysis uses the most accurate historic window size for each scenario as determined in the previous section, namely 48 hours for kitchen 1 and 24 hours for kitchen 2 in day-ahead, and 24 hours for both kitchens in hour-ahead.

In Table 4.7 and Table 4.8 it is presented the average NRMSE of the appliances for each kitchen in the day-ahead scenario.

Training subsets	XGBoost		CatBoost		LSTM	
	NRMSE	StD	NRMSE	StD	NRMSE	StD
1	20.05%	21.49%	19.32%	18.24%	23.97%	14.96%
2	16.02%	16.91%	16.10%	16.73%	**23.41%**	13.30%
3	18.13%	19.52%	17.69%	18.61%	25.82%	14.06%
4	**15.52%**	17.59%	**15.27%**	17.47%	25.71%	10.97%

Table 4.7: Day-ahead analysis - Best training length for K1

Training subsets	XGBoost		CatBoost		LSTM	
	NRMSE	StD	NRMSE	StD	NRMSE	StD
1	**12.41%**	20.19%	**11.91%**	20.65%	**18.52%**	17.40%
2	13.76%	21.63%	13.86%	21.10%	21.22%	18.93%
3	14.06%	22.24%	13.18%	19.77%	20.70%	18.95%
4	13.95%	21.02%	13.83%	20.88%	20.84%	18.51%

Table 4.8: Day-ahead analysis - Best training length for K2

For K1, the results obtained were expected. With small datasets, the increase in the number of subsets used to train models improves the accuracy of predictions. Despite the LSTM model presenting better results for fewer subsets used, there is a considerable difference in performance relative to XGBoost and CatBoost. This way, the training length for K1 - day-ahead scenario was set with the 4 subsets, approximately 19 days. K2 results however are not the expected, with the smallest training length presenting consistent and better results for every model. This is probably caused by more constant patterns presented in data from K2, in relation to K1 appliances. Nevertheless, since the two kitchens are independent from each other: have different appliances and working periods, the training length for K2 - day-ahead was set with the first subset. Since each subset has a length of approximately 4 and a half days and the results show that K2 appliances benefit from smaller training times, the training length was set to 4 days.

In Table 4.9 and Table 4.10 it is presented the average NRMSE of the appliances for each kitchen in the hour-ahead scenario.

Training subsets	XGBoost		CatBoost		LSTM	
	NRMSE	StD	NRMSE	StD	NRMSE	StD
1	12.56%	21.57%	12.54%	21.99%	13.53%	22.17%
2	9.75%	16.11%	9.34%	15.96%	**13.34%**	17.71%
3	10.02%	15.85%	9.56%	15.62%	15.43%	17.47%
4	**9.65%**	15.84%	**9.19%**	15.70%	15.85%	16.90%

Table 4.9: Hour-ahead analysis - Best training length for K1

Training subsets	XGBoost		CatBoost		LSTM	
	NRMSE	StD	NRMSE	StD	NRMSE	StD
1	**4.02%**	15.10%	4.10%	14.62%	**5.40%**	14.92%
2	4.49%	13.32%	4.52%	13.12%	6.04%	13.91%
3	4.52%	14.36%	4.49%	14.17%	5.70%	15.25%
4	4.07%	13.50%	**3.99%**	13.15%	5.70%	14.43%

Table 4.10: Hour-ahead analysis - Best training length for K2

In this analysis, results for K1 present a similar behavior as for day-ahead, with the longer training length presenting better results for XGBoost and CatBoost, with a difference of around 15% in relation to LSTM. For K2 there is also a big difference in the results, with the best results of XGBoost and CatBoost being 1 and 4 subsets respectively. Since the lowest error in this analysis was obtained for 4 subsets and the difference in the results of XGBoost is not significant, the training length for both kitchens was set as the same, with the 4 subsets used, approximately 19 days for K1 datasets and 18 days for K2.

4.4 Hyperparameter Optimization Results

The results of this analysis were aggregated in six tables, one for each model and scenario. Since each table contains the results for all appliances in both kitchens, the Tables' extent might confuse the reader. This way, the results are presented in Appendix A. The configurations obtained with this optimization are used to obtain the forecasting results presented in the following sections.

4.5 Day-ahead Results

This section presents the results obtained for the day-ahead analysis, with Sections 4.5.1 and 4.5.2 presenting the results of the forecast models for K1 and K2 respectively, using the three forecast models mentioned previously.

4.5.1 Kitchen 1

This section presents the results of the forecast models for the appliances in K1. Table 4.11 presents the average RMSE value and the standard deviation between every prediction and target window, for the different methods and appliances.

From this table, it is possible to conclude that the GBDT algorithms present similar results, with neither model presenting consistent and better results. Thus, it is impossible to identify whether XGBoost outperforms CatBoost or vice-versa clearly. The biggest improvement of XGBoost on accuracy was on the Dual Fryer, with a reduction of 10.6% in error in relation to CatBoost, while this model presented the biggest improvement in performance in the Freezer, with a reduction of 10.2% of the error in relation to XGBoost. On the other hand, the LSTM model presents a poor performance for the majority of the appliances, when compared with the two previous methods.

Device	XGBoost		CatBoost		LSTM	
	RMSE	StD	RMSE	StD	RMSE	StD
Blast Chiller	193.4	43.1	186.3	45.7	207.5	41.4
Convection Oven 1	246.6	53.2	248.5	39.8	281.0	41.0
Convection Oven 2	288.4	103.2	283.0	117.6	302.4	98.9
Dish Washer	160.3	32.1	162.9	43.2	203	43.6
Drinks Refrigerator	62.1	17.8	61.4	17.0	65.6	13.8
Dual Fryer	31.3	7.3	35.0	5.4	44.0	10.5
Fish Refrigerator	26.2	6.4	25.7	6.7	52.5	4.3
Freezer	54.1	15.9	48.6	13.2	203.6	17.6
Garde Manger 1	17.3	11.6	17.4	12.8	54.4	4.0
Garde Manger 2	11	5.0	10.7	5.3	26.6	1.3
Ice Machine	78.3	11.4	76.4	10.8	89.7	15.3
Infrared Shelf	192.1	35.8	202.4	24.3	216.1	34.4
Meat Refrigerator	26.3	11.8	25.3	12.1	52.3	8.0
Mise en Place	42.1	20.3	42.4	20.4	134.9	7.1
Salamander 1	932.9	146.9	908.4	150.9	919.2	210.1
Salamander 2	605.1	71.1	645.4	47.4	672.0	92.2
Vegetables Refrigerator	19.4	7.0	19.5	7.3	41.4	4.2

Table 4.11: Day-ahead RMSE [W] - Kitchen 1

4.5.2 Kitchen 2

In Table 4.12 it is presented the RMSE value and the standard deviation of the different models, for each appliance. As for the results of K1, the forecast results in K2 are similar for the GBDT models. For most appliances, there is only a slight difference in the errors of the GBDT models, with only the Hothouse registering a considerable reduction of 17.8% of RMSE with XGBoost, in relation to CatBoost. The LSTM model continues to present a poor performance compared to the other two models. Despite presenting similar values of error for some appliances as the Blast Chiller or the Convection Oven, it presents errors two to four times bigger for other appliances, as is the case with Refrigerators. The standard deviation of the error also presents similar values between the two GBDT models, indicating that both present the best choices to perform the forecasts.

Device	XGBoost		CatBoost		LSTM	
	RMSE	StD	RMSE	StD	RMSE	StD
Blast Chiller	26.2	12.0	28.2	10.5	29.6	10.6
Convection Oven	110.4	177.5	112.7	176.2	118.5	178.5
Exhaust	59.0	15.3	59.4	19.1	86.6	14.6
Heating Lamp 1	1.1	0.8	1.0	0.8	1.0	0.9
Heating Lamp 4	0.2	0.0	0.6	0.1	0.3	0.0
Hothouse	152.2	25.5	185.1	31.3	255.4	63.9
Induction Plate	124.0	46.5	119.2	44.0	120.7	36.6
Microwave	0.5	0.0	0.6	0.0	0.0	0.0
Refrigerator 1	25.8	4.4	26.1	3.8	33.7	3.6
Refrigerator 2	20.1	2.3	18.7	3.3	41.1	1.8
Refrigerator 3	26.8	2.9	26.6	2.7	41.9	4.1
Refrigerator 4	15.0	1.3	14.1	1.5	46.2	2.6
Sous Vide	145.2	26.4	142.5	30.0	161.6	35.1

Table 4.12: Day-ahead RMSE [W] - Kitchen 2

The main conclusion obtained in the day-ahead analysis is that the type of appliance is crucial for the quality of the forecast. For this reason, in some appliances was not possible to obtain significantly accurate forecasts, especially for those that present irregular usage patterns that, even with pre-processing techniques, cannot be removed. This is mainly the case in devices such as refrigerators and freezers.

Appliances that present low utilization rates are not the best suited to be filtered using median filters since, in this situation, the median values are mostly around zero. This is the case with appliances such as microwaves and heating lamps. This results in data that is mainly zero, with the values related to the activations of the appliances being considered outliers by the models, and thus not taken into account in the forecasts.

One factor that considerably influences these results is related to the nature of this type of forecast. In IK some appliances have very short operating times when compared with the length of a day, which makes detecting consumption patterns in short periods much harder since the forecast models will only detect the overall daily changes in the load. It is the case for Garde Mangers and some Refrigerators that require sampling periods deeply under 30 minutes, which causes a considerable increase in the time and computational costs to perform the forecasts, especially for a day-ahead analysis. Appliances' load forecast in a day-ahead scenario is still a topic with very little research in the literature, presenting another constraint for this analysis.

Having such a small dataset to perform day-ahead analysis also constrains the ability to perform this type of forecast. While usually authors one year of data or more [69], [40] to properly train ML and DL models in load forecast, in this work the available amount of data is less than one month, insufficient for the forecast models to capture data characteristics, especially in the appliances that present more complex usage patterns. Also, complex models based on neural networks require large amounts of data to properly calculate the model parameters. This is mainly the reason why in day-ahead analysis the LSTM-based model has a very poor performance.

4.6 Results - Hour-ahead

This section presents the results for the hour-ahead scenario. Following the same report structure as for the previous analysis, Sections 4.6.1 and 4.6.2 present the forecast models' results for K1 and K2 respectively, using the three forecast models mentioned previously.

4.6.1 Kitchen 1

The results of the forecast models for K1 appliances, namely the average RMSE for each prediction window and respective standard deviation, are presented in Table 4.13.

Device	XGBoost		CatBoost		LSTM	
	RMSE	StD	RMSE	StD	RMSE	StD
Blast Chiller	116.1	144.3	118.9	143.5	148.0	162.8
Convection Oven 1	309.3	509.3	311.6	509.2	350.0	557.7
Convection Oven 2	309.6	600.0	313.8	593.9	326.6	600.1
Dish Washer	253.5	397.9	259.6	386.1	272.8	420.7
Drinks Refrigerator	128.0	63.6	127.3	59.6	143.6	66.0
Dual Fryer	65.9	145.5	69.0	141.3	76.7	147.8
Fish Refrigerator	44.4	30.1	44.7	30.7	49.8	32.3
Freezer	44.4	25.0	42.8	24.1	340.1	36.6
Garde Manger 1	6.5	13.3	7.0	12.9	5.7	13.8
Garde Manger 2	0.9	3.0	0.9	3.0	0.7	3.1
Ice Machine	77.9	52.9	77.5	50.7	90.5	52.2
Infrared Shelf	108.5	154.7	104.9	154.6	120.9	170.4
Meat Refrigerator	18.5	39.9	18.2	39.6	19.1	42.5
Mise en Place	98.3	61.0	97.5	60.1	105.1	64.3
Salamander 1	307.8	478.4	299.4	456.2	338.3	580.5
Salamander 2	215.9	417.1	230.6	419.7	249.2	502.2
Vegetables Refrigerator	41.7	36.1	42.1	36.6	43.7	40.0

Table 4.13: Hour-ahead RMSE [W] - Kitchen 1

The main difference in relation to the day-ahead analysis is that the LSTM model presents better performance in relation to the day-ahead analysis. Although the GBDT models indicate a higher accuracy for most of the appliances, the fact is that in most cases, the error difference in relation to the LSTM model is not drastic as it was for the day-ahead analysis. Apart from that, both GBDT models present similar error values, as for the day-ahead analysis, with Salamander 2 presenting the biggest reduction in the error, of 6.4% in relation to CatBoost, while this model's best performance is for the Infrared Shelf, with only a reduction of 3.3% in the error in relation to XGBoost.

4.6.2 Kitchen 2

As for K1, the results of the different forecast models for K2 appliances, namely the average RMSE for each prediction window and respective standard deviation, are presented in Table 4.14.

Device	XGBoost		CatBoost		LSTM	
	RMSE	StD	RMSE	StD	RMSE	StD
Blast Chiller	33.5	52.3	36.8	54.3	47.0	68.3
Convection Oven	221.6	306.0	232.0	304.3	272.9	342.5
Exhaust	18.9	33.1	20.7	32.9	23.6	36.9
Heating Lamp 1	0.7	2.9	0.9	2.8	0.6	3.0
Heating Lamp 4	7.4	24.3	8.8	23.0	7.6	29.8
Hothouse	145.6	317.8	146.7	291.4	153.6	333.7
Induction Plate	76.7	180.6	79.9	159.7	88.1	186.7
Microwave	0.5	3.1	0.5	3.1	0.4	3.6
Refrigerator 1	24.6	27.1	27.0	26.5	30.4	28.3
Refrigerator 2	1.8	10.5	2.1	10.5	1.4	10.8
Refrigerator 3	6.7	21.8	4.8	21.8	7.6	28.4
Refrigerator 4	7.7	12.7	9.5	13.2	8.5	6.6
Sous Vide	92.5	230.8	114.1	231.2	95.4	222.5

Table 4.14: Hour-ahead RMSE [W] - Kitchen 2

Here XGBoost clearly outperforms the remaining models. It presents the best performance in the majority of the appliances, despite presenting similar or higher standard deviation values in relation to the remaining models. The biggest decrease in the error in relation to the CatBoost model is obtained for Sous-Vide, where the XGBoost model presents an improvement of 18.9%, while in relation to the LSTM model it presents a decrease of 28.7% for the Blast Chiller.

For hour-ahead analysis, the difference in results between the two types of models is not as expressive as for day-ahead. This situation is verified mainly by the increase in the number of available points in hour-ahead predictions, that favors neural network based models, which require great amounts of data to properly train and fit the network. However, summarizing the results of the two kitchens GBDT models still present better results than the LSTM model, especially XGBoost. Despite having appliances in which LSTM presents better results, the differences in relation to the other models are not very significant, while the opposite is not always verified. There are appliances that achieve much better results with the GBDT models, such as the Convection Oven 1, Freezer and Salamander 1 in K1, or the Blast Chiller and the Convection Oven in K2.

4.7 Training Times

When forecasting models are used, it is important to consider the respective computation times. In this subsection, the training times for each method will be presented. Especially for short-term forecasts, the training time is also relevant for the reliability of the methodology. Thus, a balance between low RMSE and reduced training time of the model should exist.

In ML and DL models, hyperparameter optimization can significantly impact the training time required. For GBDT algorithms, the main hyperparameters to affect the time required for model training are learning rate, maximum depth, and number of trees. Also, data re-sampled for the day-ahead analysis results in a smaller dataset than for hour-ahead due to the re-sampling periods chosen in Section 3.1. The length of the training set and historic window size also affect the training time of the models because data with a smaller historic window will present fewer features, requiring less time to train. Since different appliances use different sets of hyperparameters, Table 4.15 presents the average training time for each model and each analysis. This times were obtained in a computer with a 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz CPU, 16.0 GB RAM, and 64 bits operating system.

Analysis	Kitchen	XGBoost	CatBoost	LSTM
Day-ahead	Kitchen 1	**16.2**	106.4	35.9
	Kitchen 2	8.8	18.1	**6.2**
Hour-ahead	Kitchen 1	**64.8**	68.7	254.2
	Kitchen 2	**61**	79.7	360.5

Table 4.15: Average training times [s] for day-ahead predictions

The main conclusion is that the XGBoost model averages a lower training time in every situation except for the day-ahead analysis in K2, where LSTM is the fastest model. Between the models that presented better results in the previous analysis, XGBoost clearly outperforms CatBoost in terms of training time, presenting an advantage, even when XGBoost had slightly lower accuracy than CatBoost.

Significant differences exist in the training time of the LSTM model for the two different analyses. This is mainly because of the different dataset dimensions for the two analyses and the hyperparameters chosen for model optimization emphasize the influence of the dataset size in the training time of the LSTM-based model more than in GBDT models, where only for low learning rates and high values of tree depth this models will have long training times.

From the several analysis made and from the discussion on the results obtained, it is possible to conclude that XGBoost presents the more adequate choice to perform the forecasts in this work. Despite not presenting the best accuracy for all scenarios, it presents the best trade-off between prediction accuracy and model training time. This way, the post-processing methodologies were analyzed for the results obtained with XGBoost.

4.8 Post-processing Results

In the previous sections, the results obtained directly from the forecast models were analyzed. In this section the methodologies for post-processing of results explained in Section 3.4 are implemented for both kitchens and scenarios (day and hour-ahead).

First, the predictions were adjusted to identify and update values lower than 0. As mentioned beforehand, cleaning the negative values with pre-processing techniques implies that negative values are unlikely to appear, and the consequence in the error metric values is residual. Then, the main goal is to determine which is the best method to update the prediction windows, for each time instant.

Tables 4.16 and 4.17 present the results of the post-processing methodologies for K1 and K2 respectively, for the day-ahead analysis.

Device	Forecast Results	Last Prediction	Persistence Prediction	Average Prediction	Weighted Prediction
Blast Chiller	193.4	144.1	148.5	190.5	**129.2**
Convection Oven 1	246.3	164.3	181.1	244.9	**158.4**
Convection Oven 2	288.4	**191.4**	224.7	283.8	192.0
Dish Washer	160.3	**112.5**	122.6	157.4	113.0
Drinks Refrigerator	62.1	48.3	54.8	61.1	**41.7**
Dual Fryer	31.3	24.0	27.5	30.6	**22.5**
Fish Refrigerator	26.2	16.1	19.1	25.5	**15.2**
Freezer	54.1	34.6	36.5	53.2	**31.8**
Garde Manger 1	17.3	15.2	15.8	17.0	**13.8**
Garde Manger 2	11.0	8.9	10.6	10.9	**8.2**
Ice Machine	78.3	51.6	56.5	76.3	**44.1**
Infrared Shelf	192.1	120.0	132.0	187.4	**98.7**
Meat Refrigerator	26.3	19.5	22.6	25.8	**18.0**
Mise en Place	42.1	30.4	33.9	41.2	**27.4**
Salamander 1	932.9	577.0	606.5	919.5	**536.6**
Salamander 2	605.1	453.5	472.1	600.5	**451.4**
Vegetables Refrigerator	19.4	16.4	18.2	19.0	**14.7**
Total Error	2986.8	2027.7	2183.3	2944.8	**1916.6**

Table 4.16: RMSE [W] of post-processing methods for K1 - day-ahead analysis

These results present a clear improvement in the performance of the methodology when compared with the error in the output of the forecast models. For both scenarios, the weighted prediction method presents the best results. For K1 results, it presents a reduction of 35.8% in relation to the forecast results and 5.5% in relation to the last prediction. For K2 results it presents an improvement of 24.3% in relation to the forecast results and 9.9% in relation to the last prediction method.

Figures 4.6 and 4.7 present the results of the post-processing methodologies in the day-ahead scenario, for two different appliances - Infrared Shelf of K1 and Exhaust of K2. These figures present the first point from each prediction window of the respective methods. From these figures, it is possible to visualize the influence of the weighted prediction algorithm, especially when the device is switched ON.

Device	Forecast Results	Last Prediction	Persistence Prediction	Average Prediction	Weighted Prediction
Blast Chiller	26.2	19.7	23.1	26.2	**18.8**
Convection Oven	110.4	103.8	102.7	104.9	**100.6**
Exhaust	59.0	48.2	46.9	57.5	**40.6**
Heating Lamp 1	**1.1**	**1.1**	**1.1**	**1.1**	**1.1**
Heating Lamp 4	0.2	**0.1**	**0.1**	0.2	0.2
Hothouse	152.2	130.1	127.7	148.3	**111.6**
Induction Plate	124.0	109.1	117.7	121.1	**100.8**
Microwave	**0.5**	**0.5**	**0.5**	**0.5**	**0.5**
Refrigerator 1	25.8	28.7	24.0	24.4	**21.2**
Refrigerator 2	20.1	18.2	17.5	19.2	**12.9**
Refrigerator 3	26.8	27.5	26.0	26.6	**24.0**
Refrigerator 4	15.0	14.6	12.8	13.0	**7.9**
Sous Vide	145.2	**91.6**	98.8	140.9	95.5
Total Error	706.4	593.3	592.9	683.9	**534.7**

Table 4.17: RMSE [W] of post-processing methods for K2 - day-ahead analysis

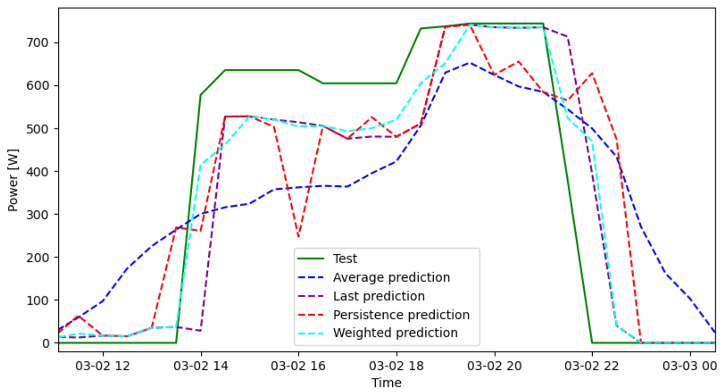

Figure 4.6: Post-processing techniques for day-ahead analysis - Infrared Shelf

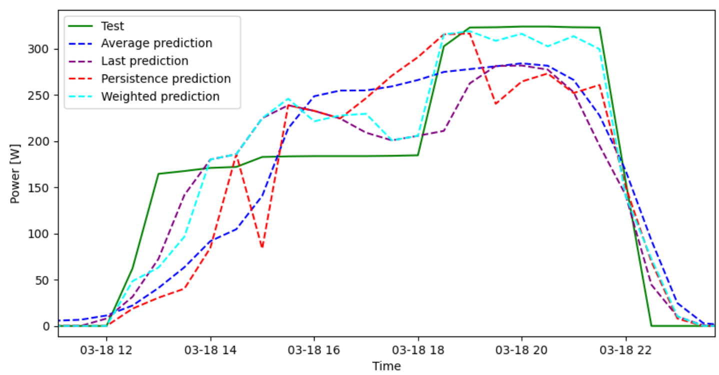

Figure 4.7: Post-processing techniques for day-ahead analysis - Exhaust

Tables 4.18 and 4.19 present the results obtained with the different post-processing techniques, for the appliances of both kitchens in the hour-ahead scenario.

Device	**Forecast Results**	**Last Prediction**	**Persistence Prediction**	**Average Prediction**	**Weighted Prediction**
Blast Chiller	116.1	**42.8**	101.9	111.4	43.5
Convection Oven 1	309.3	**160.7**	225.6	299.8	163.6
Convection Oven 2	309.6	171.1	171.1	300.0	**168.4**
Dish Washer	253.5	174.6	174.6	247.5	**172.4**
Drinks Refrigerator	128.0	**65.8**	**65.8**	124.2	70.2
Dual Fryer	65.9	43.6	44.5	63.5	**42.6**
Fish Refrigerator	44.4	33.4	33.4	43.1	**30.8**
Freezer	44.4	31.5	41.1	41.5	**28.8**
Garde Manger 1	6.5	**6.3**	6.4	**6.3**	6.4
Garde Manger 2	0.9	**0.8**	**0.8**	0.9	**0.8**
Ice Machine	77.9	32.3	53.8	73.3	**31.3**
Infrared Shelf	108.5	**33.3**	83.4	102.4	34.7
Meat Refrigerator	18.5	**13.4**	**13.4**	18.1	**13.4**
Mise en Place	98.3	**82.6**	93.4	94.7	85.5
Salamander 1	307.8	**92.9**	**92.9**	294.3	104.4
Salamander 2	215.9	**51.0**	**51.0**	208.4	60.9
Vegetables Refrigerator	41.7	30.2	39.5	40.5	**28.8**
Total Error	2143.0	**1066.3**	1292.6	2070.0	1086.7

Table 4.18: RMSE [W] of post-processing methods for K1 - hour-ahead analysis

Device	**Forecast Results**	**Last Prediction**	**Persistence Prediction**	**Average Prediction**	**Weighted Prediction**
Blast Chiller	33.5	19.6	19.6	31.2	**18.0**
Convection Oven	221.6	100.5	103.2	210.9	**97.9**
Exhaust	18.9	7.5	7.5	17.9	**6.9**
Heating Lamp 1	0.7	**0.6**	0.7	0.7	**0.6**
Heating Lamp 4	7.4	**3.8**	**3.8**	7.0	3.9
Hothouse	145.6	88.8	88.8	142.4	**86.8**
Induction Plate	76.7	**32.1**	**32.1**	72.8	32.4
Microwave	0.5	**0.2**	**0.2**	0.5	0.3
Refrigerator 1	24.6	**17.5**	21.3	24.2	18.5
Refrigerator 2	1.8	**1.6**	**1.6**	1.8	**1.6**
Refrigerator 3	6.7	**6.3**	**6.3**	6.7	6.4
Refrigerator 4	7.7	5.9	5.9	7.6	**5.6**
Sous Vide	92.5	**35.9**	35.9	88.2	38.4
Total Error	638.0	320.4	327.1	611.8	**317.4**

Table 4.19: RMSE [W] of post-processing methods for K2 - hour-ahead analysis

For the hour-ahead analysis, there are several cases where the errors of the last prediction and persistence-based predictions are equal. This happens when, for the first window of predictions in the test set, the most accurate value for each time instant is the last prediction made. In this analysis, the weighted prediction was not as effective, as for the day-ahead analysis. For K1 results, the last prediction method presented a reduction of 50.2% in the error in relation to the forecast results and a reduction of 1.9% in relation to the weighted predictions. Regarding K2 results, the weighted predictions present an improvement of 50.3% in the error of the forecast results and an improvement of 1% in relation to the last prediction method.

Figures 4.8 to 4.10 present the results of the post-processing methodologies in the hour-ahead scenario for different appliances, namely the Convection Oven 2 of K1 and the Heating Lamp 4 and Refrigerator 1 of K2. For the Convection Oven 2 the results present an improvement on the results obtained with the weighted prediction method, in relation to the remaining post-processing methods, with the persistence-based predictions and the most recent predictions being overlapped, as the results from Table 4.18 indicate.

For the results of Refrigerator 1, it is visible that in some of the periods the weighted predictions method does not present the same accuracy as the most recent prediction method, even though it still presents lower error than the original results and the remaining post-processing techniques. Regarding the results of Heating Lamp 4, the results justify the minimal difference in the error metrics, with the most recent and persistence-based predictions being practically equal to the weighted prediction results.

One of the main conclusions of the post-processing analysis is that, from the methods presented in Section 3.4, the ones that present the best performance are the last prediction and weighted prediction methods. While for K2 the persistence-based prediction presented also good results, it never presented the best performance for any scenario. The average prediction method proved to be poorly performing.

Despite presenting a similar performance with the last prediction method in the hour-ahead scenario, the weighted predictions present the best results for a considerable number of appliances, and it also clearly outperforms the rest of the post-processing methods in the day-ahead analysis, presenting a reliable choice in this type of analysis.

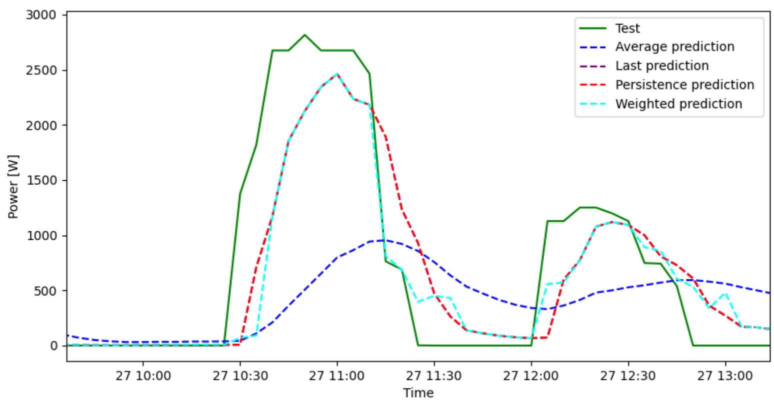

Figure 4.8: Post-processing techniques for hour-ahead analysis - Convection Oven 2

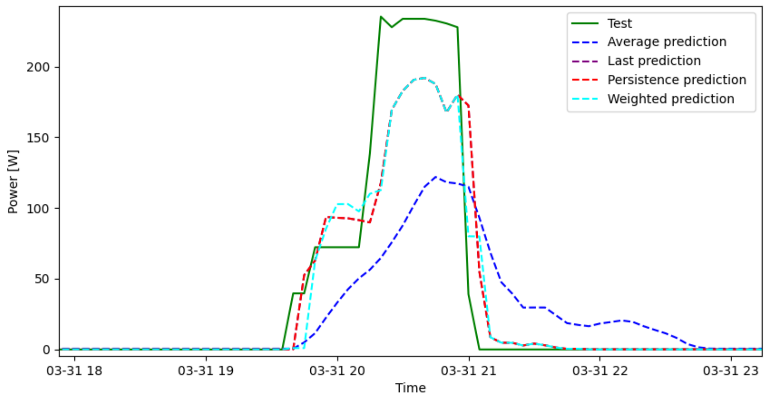

Figure 4.9: Post-processing techniques for hour-ahead analysis - Heating Lamp 4

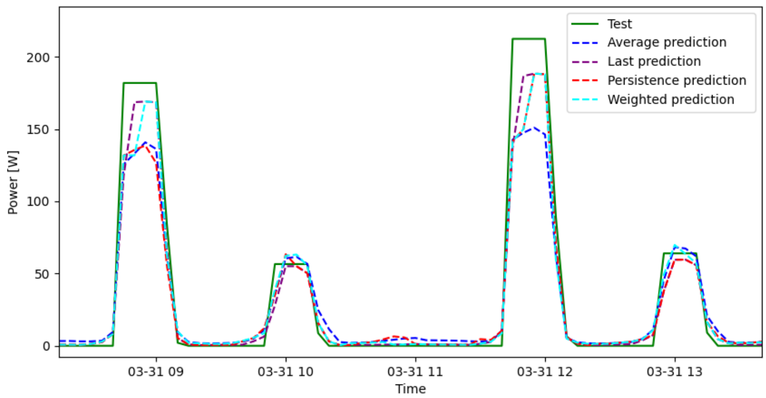

Figure 4.10: Post-processing techniques for hour-ahead analysis - Refrigerator 1

4.9 Discussion

Overall, the GBDT models present better results than the LSTM. For some appliances, interpreting features as categorical results in slightly lower errors. However, CatBoost presents an average training time greater than XGBoost, and the trade-off between the error reduction and training time is, in most situations, not worth the application of a slower method as CatBoost. Despite the relatively low training time of the LSTM model in some analyzes, the results are by far worse than the results presented by XGBoost.

Appliances that present low utilization rates are not the best suited to be filtered using median filters since, in this situation, the median values are mostly around zero. This is the case with appliances such as microwaves and heating lamps. This results in data that is mainly zero, with the values related to the activations of the appliances being considered outliers by the models and, thus not taken into account in the forecasts.

Regarding the post-processing techniques implemented in this work, several conclusions can be registered. Firstly, for the day-ahead analysis, the weighted predictions algorithms reduce the error for almost all appliances when compared with the most recent/last prediction available, which should present the value forecasted with more updated information on the data, thus presenting more accuracy. The effectiveness of the methodology is not homogeneous for all appliances but it results in an improvement of 17.8% in the error for the Infrared Shelf of K1 and 45.9% for Refrigerator 4 of K2. The persistence-based predictions present higher errors in relation to the weighted algorithm, but in relation to the last prediction method, it presents higher errors for the K1 appliances but a lower error for the majority of K2 appliances. From this, it is possible to conclude that the type of appliance is relevant for determining the predicted values to consider.

For the hour-ahead scenario, the post-processing technique proposed are not as effective as for the day-ahead. Nevertheless, it still presents better results than the remaining methods for several appliances, presenting a reduction in error up to 7.1% for the Vegetables Refrigerator of K1 and up to 15.9% for the Heating Lamp 4 of K2. Also, this technique still presents a great improvement in relation to the results without post-processing.

For both scenarios, the average prediction technique presents poor performance. By giving the same importance to every prediction, this method penalizes the most accurate predictions by adding the values of the least accurate predictions, which are often the oldest predicted values, with less information about the behavior of the appliance. Therefore, this method is not suited as a post-processing technique for this type of analysis.

5

Integration with flexibilities

Contents

5.1 Demand-Side Flexibility 57
5.2 Methodology 58
5.3 Results 60

This chapter considers the integration of the forecast models with the appliances' flexibilities, to provide insight into the potential flexibility in the future. Section 5.1 presents a summary of load flexibility state of the art, Section 5.2 presents the proposed methodology to perform this analysis and Section 5.3 presents the results obtained with the proposed methodology.

5.1 Demand-Side Flexibility

Several definitions for flexibility have been proposed in recent years, According to [70], flexibility corresponds to the amount of energy and the duration of time to which the device energy profile (energy flexibility) and/or activation time (time flexibility) can change. In [71] flexibility is defined as the system's ability to accommodate variation in net load, by modulating the feed-in or feed-out of power across the grid over time. Besides central generation, several potential sources of flexibility are demand response, energy storage, and distributed energy generators [72]. Further sources of flexibility can be provided by grid interconnections and combined-cycle units [71].

Load flexibility and forecasting are two topics that appear interconnected in energy management, especially when the energy is provided by RES. With increasing RES penetration in the electric grid, the uncertainty and volatility of this sources often requires the use of load flexibilities to maintain a balance between generation and demand [73]. One extreme example of this problem is the voltage collapse in a system with high penetration of photovoltaic generation, due to partial clouding [74].

Self-sufficiency through independence from energy supply companies is usually a dominant motivation factor of prosumers for installing microgeneration sources [75]. This, together with the environmental concerns of prosumers represents social motives to reduce their greenhouse gas footprint [76], shifting the consumption away from hours where RES production is low and/or the consumption is forecasted to be high becomes important. To provide solutions for these situations, this chapter presents an integration of the forecasts with load flexibilities.

In the *nexIK* project flexibility was defined as the capacity of electric devices to adjust their energy consumption to support the electric grid and keep the user comfort [7]. Three parameters are considered when classifying the flexibility of a particular appliance: load shift, load reduction, and user comfort. Based on this and the appliances commonly found in IK, it was developed the classification presented in Table 5.1.

In this chapter the main focus is on the time flexibility of the loads, to allow load-shifting operations in the appliances. Section 5.2 presents the methodology used to integrate the forecast with the load's time flexibility in this analysis.

Appliance	Not Flexible	Flexibility	
		Time	Power
Dishwasher		X	
Freezer		X	X
Refrigerator		X	X
Microwave	X		
Stove	X		
Oven	X		
HVAC / AC		X	X
Lighting			X

Table 5.1: Classification of IK appliances according to their flexibility potential (Source: [7])

5.2 Methodology

In this section, it is presented the methodology applied in the flexibilities analysis. Figure 5.1 represents the pipeline built and implemented for this analysis.

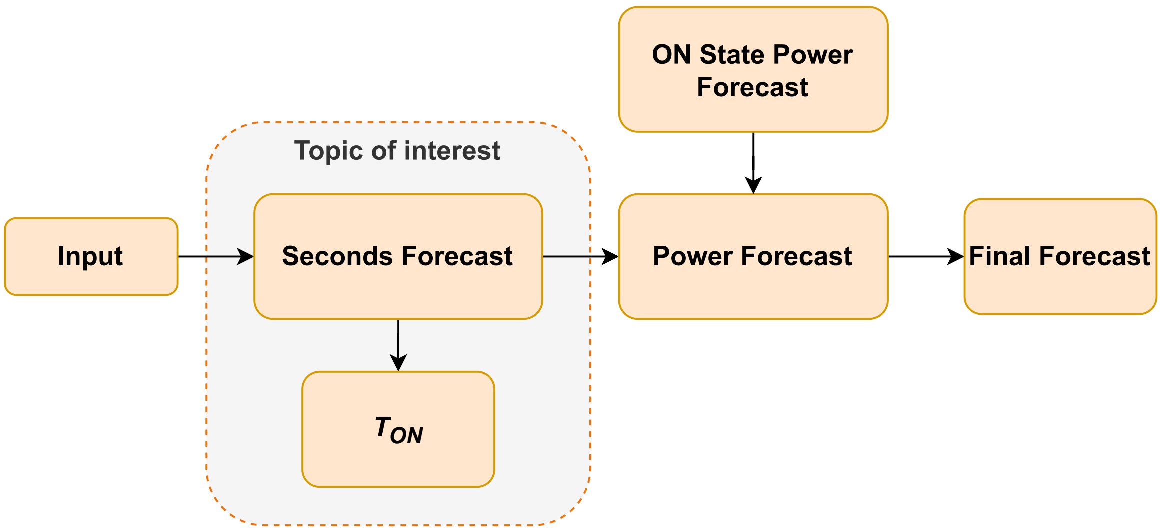

Figure 5.1: Methodology for integration with flexibilities

The main goal of this methodology is to provide a framework for a combination of forecasts and flexibilities, with the results of this analysis focused on the accuracy of the forecasts of the number of seconds in the ON state of each appliance. This way it is possible to perceive future flexibility in time to perform load-shifting operations.

The main feature of this methodology consists in separately forecasting the number of seconds in which an appliance is ON and the power consumption in the ON state, allowing to understand the potential load flexibility in time and how much power is available. This method was developed for appliances that present a clear ON/OFF behavior, as is the case with Refrigerators, Salamanders, and Garde Mangers, for example. Forecasting the duration of the ON states allows to apply variations to the different loads, by providing the user with knowledge about the periods with greater power consumption. From inspection of the datasets, the appliances chosen for this analysis were the Refrigerators, Freezer, Garde Mangers, and Mise en Place from kitchen 1, and the Hothouse, Sous-Vide, and Refrigerators from kitchen 2. The Dish Washer of kitchen 1 was not included since this methodology requires that the appliances present an ON/OFF behavior, which is not the case with the data obtained for this appliance, as can be seen in Figure 5.2.

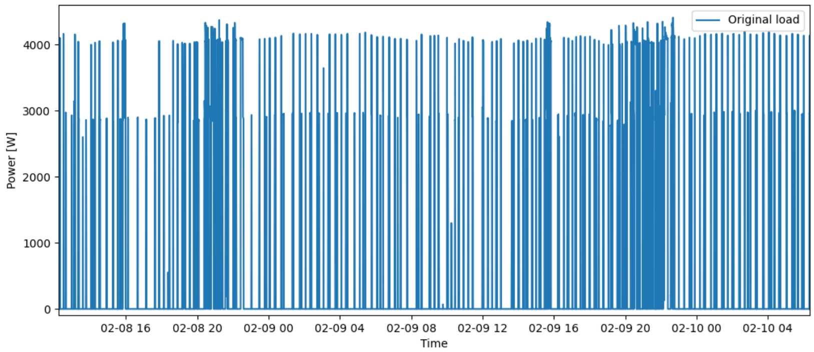

Figure 5.2: Sample of Dish Washer original load

To obtain a clear view of the ON states of the appliances, the sampling rate was changed to one minute. This way it is possible to capture the activation patterns for all devices mentioned. However, this represents a problem in terms of execution time for the day-ahead analysis, where a sampling rate of one minute implies the forecast of 3600 samples. This way, the integration with flexibilities was limited to the hour-ahead scenario, with a change in the historical window size from one day to one hour, in order to reduce the number of samples in each historic window used to train the model.

In this methodology, the input data is used to build a dataset with information on the number of seconds in the ON state per minute. Then, the data is pre-processed as for the traditional forecast and pipelined to the forecast models. These forecast models output predictions for the number of seconds in the ON state for the test period. Simultaneously, the power corresponding to the ON states of the appliances is extracted from the original data and pre-processed. This variable was obtained by the detection of values above a threshold of 80% of the operating power of the device. This operating power was obtained by inspection of the datasets prior to the analysis and can represent a constraint to real-life implementation. This data is then fed to a linear regression model to perform the forecast. For this variable, it was chosen a simpler method since it is expected that the power in the ON state of this type of appliances does not present abrupt oscillations.

Then, the post-processing methodologies mentioned in Section 3.4 are applied to the data in order to update the prediction windows, resulting in the final forecasted values. The two variables forecasted are then combined to obtain a forecast of power and compared with a reconstruction of the original power signal, made from the two built datasets of time and power in the ON state. For error measuring and to evaluate the performance, the RMSE metric was used.

5.3 Results

This section presents the results obtained for the forecasting of seconds in the ON state of the appliances mentioned in the previous section. The forecasts were performed using the same training lengths, with the historical and prediction windows presenting a size of one hour or (60 samples) each. The forecast model used was XGBoost due to the good results presented for the previous analysis.

With a very small sampling period, the training time of the models must be taken into account. To apply the proposed post-processing techniques, the training time has to be lower than the sampling period, to allow the prediction update at each timestep. This way, the training data was considerably shortened, to 4 days in kitchen 1 and 3 days in kitchen 2.

Device	**Forecast Results**	**Last Prediction**	**Weighted Prediction**	**Training Time [s]**
Drinks Refrigerator	22.5	10.3	**8.0**	20.6
Fish Refrigerator	10.5	**5.9**	6.1	19.7
Freezer	10.7	6.1	**5.4**	35.9
Garde Manger 1	4.7	**2.8**	3.1	24.6
Garde Manger 2	13.2	**6.8**	7.3	33.9
Meat Refrigerator	14.8	10.0	**9.7**	21.0
Mise en Place	23.2	8.9	**8.8**	55.8
Vegetables Refrigerator	4.0	**2.4**	2.8	9.8

Table 5.2: RMSE [s] of post-processing methods for K1

Device	**Forecast Results**	**Last Prediction**	**Weighted Prediction**	**Training Time [s]**
Hothouse	6.0	**2.6**	4.3	25.1
Refrigerator 1	18.4	7.0	**6.1**	37.6
Refrigerator 2	17.1	**6.2**	9.1	27.6
Refrigerator 3	20.5	**7**.5	21.1	21.1
Refrigerator 4	17.9	**6.6**	**6.6**	55.6
Sous Vide	3.6	**1.0**	**1.0**	28.0

Table 5.3: RMSE [s] of post-processing methods for K2

This results represent the average error between the test windows and the predicted windows of the duration of the ON state. Regarding the accuracy of post-processing methodologies, the results vary among the different appliances, but the last prediction method slightly outperforms the weighted prediction method. Regarding the training time, every appliance in the analysis presented a model training time lower than one minute. However, for the Mise en Place and Refrigerator 4 this duration is close to the limit, indicating that depending on the appliance, it might be necessary to modify the settings of the forecast, by reducing the complexity of the model or by changing the hyperparameters to values that increase the training speed of the model.

The post-processing techniques however might not be suitable to be applied in this situation, due to the sampling period used in this methodology and the training time of the forecast models. For these techniques to be applied, the training time has to be lower than the sampling period, to allow the prediction update at each timestep.

Figure 5.3 presents the result for the Drinks Refrigerator. Here it is possible to visualize the difference in the post-processing methods' accuracy. Despite that, the forecast is not able to produce completely accurate results, as the ON state duration forecasted does not reach 60 seconds in any timestep. However, it still provides knowledge about the periods in the ON state of the device. In this results it is possible to verify that the activation of this load could be shifted, if necessary, from 07:00 to 07:30, due to the gap in relation to the activation at 08:45.

Figure 5.4 presents the ON state duration forecast for Refrigerator 1, where the results of the post-processing techniques show some differences. While the last prediction method presents lower error for the periods where the appliance is OFF, the weighted prediction method is more accurate on the timesteps when the appliance switches ON of OFF. This favors the error on the weighted prediction since the RMSE is more sensitive to abrupt variations in the error. Still, both approaches present good results on the forecast of the ON state duration. Figures 5.5 and 5.6 present the results for other appliances, namely Garde Manger 2 and Refrigerator 4 respectively.

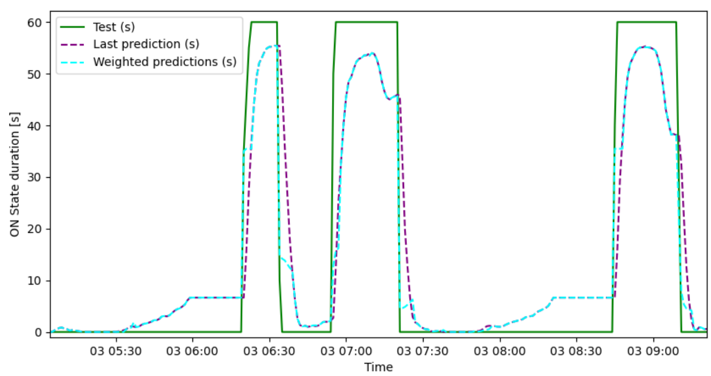

Figure 5.3: ON state duration forecast - Drinks Refrigerator

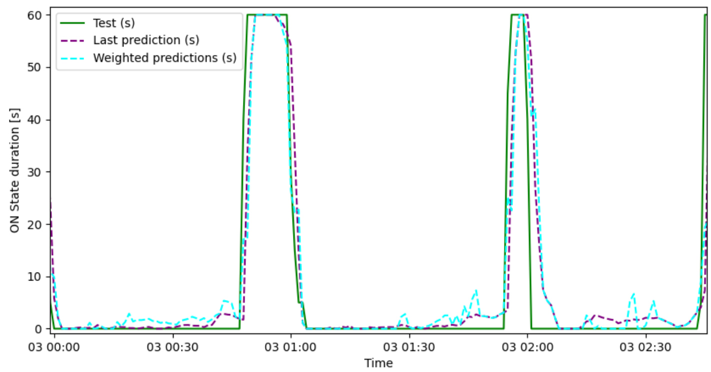

Figure 5.4: ON state duration forecast - Refrigerator 1

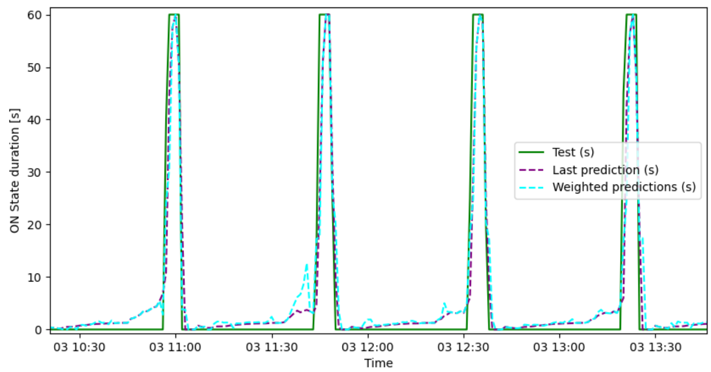

Figure 5.5: ON state duration forecast - Garde Manger 2

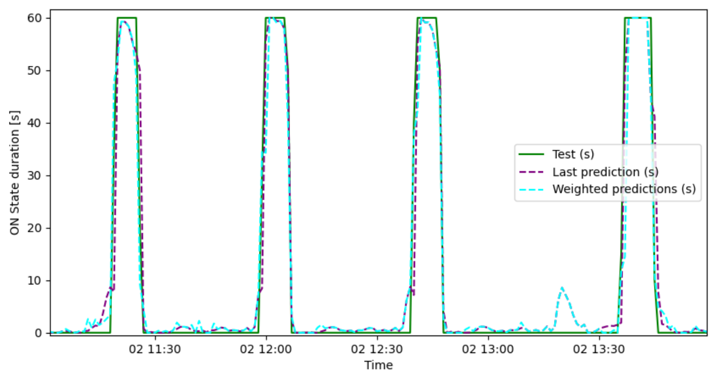

Figure 5.6: ON state duration forecast - Refrigerator 4

6

Conclusion

Contents

6.1 Conclusions 65
6.2 System Limitations and Future Work 66

6.1 Conclusions

Providing accurate load forecasts is useful for decreasing economic costs, increasing security and management of the network, and for the integration with different elements of the electric grid, such as RES. Here, apart from the economic advantages, quality forecasts present a big step for the energy transition.

In this work, a methodology based on ML and DL models to perform day-ahead and hour-ahead forecasts at the appliance level was proposed, implemented, and analyzed. In Chapter 2 the strengths of the forecasting models mentioned in the literature were presented, together with some related works. The proposed methodology also presented a different approach to the post-processing techniques, with a technique based on a reinforcement learning mechanism proven to be effective, mainly on the day-ahead analysis.

The implemented methodology consisted of powerful forecasting models such as XGBoost, CatBoost, and LSTM to detect and forecast load consumption patterns in the datasets provided. For these methods to be completely effective, data was pre-processed with re-sampling and filtering techniques. Also, to provide the data separated into targets and features, the sliding window method was used.

Chapter 4 presents the results obtained for the case study, which included the analysis of two kitchens with 17 and 13 appliances each. With the purpose of increasing the performance of the forecast models to the maximum, different optimization procedures were adopted, such as determining the best historical window size of the features, the best training length, and the tuning of each model's hyperparameters. Sections 4.2 to 4.4 present the results of these steps of the methodology. Here it was verified that the day-ahead and hour-ahead analysis presented different optimal settings for each kitchen, namely in the day-ahead scenario where K1 and K2 had historic window sizes of two days and one day respectively.

Regarding the forecast models implementation and respective post-processing Sections 4.5, 4.6 and 4.8 presented several tables and figures to provide the results in a detailed but visual and appealing way, facilitating the understanding of the performances and findings of the proposed methodology. Also, the training time analysis detailed in Section 4.7 presented crucial results to evaluate the reliability of the forecast models. At this point, it was concluded that despite the different models presenting advantages and disadvantages, XGBoost presented the overall best performance/efficiency ratio. Also, the post-processing technique proposed for weighted predictions showed effectiveness in the day-ahead analysis, presenting an overall error reduction in K1 up to 5.5% in relation to the last prediction method and 35.8% in relation to the results without post-processing. In this analysis, some appliances presented considerable reductions in the error in relation to the last prediction, as is the case of the Blast Chiller (10.3%) and Infrared Shelf (17.8%). In K2 the weighted prediction method reduced the error up to 9.9% in relation to the last prediction method and 24.3% in relation to the results without post-processing, with the Refrigerator 1 presenting an error reduction of 29.3% in relation to the last prediction method. For the hour-ahead scenario, both techniques presented similar results, but still presented an overall error reduction of up to 50% in relation to the results without post-processing.

This work also provided an integration of the forecasts with load flexibilities in Chapter 5, whose results allowed to perceive the potential for load shifting in kitchen appliances that present ON/OFF profiles. This, together with the results presented for load forecasting, fulfills the objectives set, by contributing to the goals of the *nexIK* project and increasing the knowledge about load forecasting at the appliance level in IK, a topic with very few discussion in the literature as of today.

6.2 System Limitations and Future Work

With the dimension of the methodology and tests implemented, some limitations were found. The first is related to the small dataset dimensions for the day-ahead analysis, which represented an issue to improve the forecast quality. Implementing accurate filling of missing data methods after re-sampling might present improved results. Also, due to the number of analyses made for each model, time became a constraint and thus only three models were tested. As future work, different models can be tested, namely the ones mentioned in Chapter 2 providing a more broad analysis of this topic. Adding to this, other forecast horizons can present different results that contribute to increase the knowledge about appliances forecast. Regarding the post-processing methodologies, future work can lead to developing more reliable methodologies or to improve the ones presented in this work.

The algorithm for integration with flexibilities presented some limitations, namely on the detection of the ON state of the appliances, requiring inspection of the data beforehand or knowledge about the nominal power of the appliance to properly detect the seconds in the ON state. To improve this method, developing an algorithm to automatically detect the ON state of appliances may present a significant advancement. Also, the training time of the models in this situation may limit the accuracy of the improvements made with the post-processing methodologies, especially on a production environment.

Bibliography

[1] M. Economidou, V. Todeschi, P. Bertoldi, D. D'Agostino, P. Zangheri, and L. Castellazzi, "Review of 50 years of eu energy efficiency policies for buildings," *Energy and Buildings*, vol. 225, p. 110322, 2020.

[2] S. Mudie, E. Essah, A. Grandison, and R. Felgate, "Electricity use in the commercial kitchen," *International Journal of Low-Carbon Technologies*, vol. 11, no. 1, pp. 66–74, 09 2013.

[3] J. I. Chowdhury, Y. Hu, I. Haltas, N. Balta-Ozkan, G. J. Matthew, and L. Varga, "Reducing industrial energy demand in the uk: A review of energy efficiency technologies and energy saving potential in selected sectors," *Renewable and Sustainable Energy Reviews*, vol. 94, pp. 1153–1178, 2018.

[4] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," *International Journal of Forecasting*, vol. 32, no. 3, pp. 914–938, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169207015001508

[5] T. Nonaka, T. Shimmura, N. Fujii, and H. Mizuyama, "Energy consumption in the food service industry: A conceptual model of energy management considering service properties," in *Advances in Production Management Systems: Innovative Production Management Towards Sustainable Growth*, S. Umeda, M. Nakano, H. Mizuyama, H. Hibino, D. Kiritsis, and G. von Cieminski, Eds. Cham: Springer International Publishing, 2015, pp. 605–611.

[6] S. Mudie and M. Vadhati, "Low energy catering strategy: insights from a novel carbon-energy calculator," *Energy Procedia*, vol. 123, pp. 212–219, 2017, proceedings of 1st International Conference onSustainable Energy and Resource Use in Food ChainsincludingSymposium on Heat Recovery and Efficient Conversion and Utilisation of Waste HeatICSEF 2017, 19-20 April 2017, Windsor UK.

[7] Ana Oliveira, Miriam Ribeiro, Ricardo Rodrigues, Gonçalo de Morais, Hugo Morais, and Lucas Pereira, "On The Role Of Industrial Kitchens In Sustainable Energy Systems: The nexIK Vision," in *CIRED 2023 - The 27th International Conference and Exhibition on Electricity Distribution*, Jun. 2023.

[8] R. Ahmed, V. Sreeram, Y. Mishra, and M. Arif, "A review and evaluation of the state-of-the-art in pv solar power forecasting: Techniques and optimization," *Renewable and Sustainable Energy Reviews*, vol. 124, no. 109792, 2020.

[9] U. Kumar Das, K. Soon Tey, M. Seyedmahmoudian, S. Mekhilef, M. Yamani Idna Idris, W. Van Deventer, B. Horan, and A. Stojcevski, "Forecasting of photovoltaic power generation and model optimization: A review," *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 912–928, 2017.

[10] J. Antonanzas, N. Osorio, R. Escobar, R. Urraca, F. Martinez-de Pison, and F. Antonanzas-Torres, "Review of photovoltaic power forecasting," *Solar Energy*, vol. 136, pp. 78–111, 2016.

[11] I. Moreira, "Secondary load and chp forecast to support short-term distribution network operations," MSc, Técnico Lisboa, University of Lisbon, Lisbon, Portugal, Feb. 2022.

[12] S. S. Soman, H. Zareipour, O. Malik, and P. Mandal, "A review of wind power and wind speed forecasting methods with different time horizons," in *North American power symposium 2010*. IEEE, 2010, pp. 1–8.

[13] B. Nepal, M. Yamaha, A. Yokoe, and T. Yamaji, "Electricity load forecasting using clustering and arima model for energy management in buildings," *JAPAN ARCHITECTURAL REVIEW*, vol. 3, no. 1, pp. 62–76, 2020.

[14] J. D. Hamilton, *Time series analysis*. Princeton university press, 2020.

[15] M. Qamar Raza, M. Nadarajah, and C. Ekanayake, "On recent advances in pv output power forecast," *Solar Energy*, vol. 136, pp. 125–144, 2016.

[16] H. Wang, Y. Liu, B. Zhou, C. Li, G. Cao, N. Voropai, and E. Barakhtenko, "Taxonomy research of artificial intelligence for deterministic solar power forecasting," *Energy Conversion and Management*, vol. 214, p. 112909, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0196890420304477

[17] I. Antonopoulos, V. Robu, B. Couraud, D. Kirli, S. Norbu, A. Kiprakis, D. Flynn, S. Elizondo-Gonzalez, and S. Wattam, "Artificial intelligence and machine learning approaches to energy demand-side response: A systematic review," *Renewable and Sustainable Energy Reviews*, vol. 130, p. 109899, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S136403212030191X

[18] H. Demolli, A. S. Dokuz, A. Ecemis, and M. Gokcek, "Wind power forecasting based on daily wind speed data using machine learning algorithms," *Energy Conversion and Management*, vol. 198, p. 111823, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0196890419308052

[19] G. C. Chen Tianqi, “Xgboost: A scalable tree boosting system,” *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016.

[20] X. Li, L. Ma, P. Chen, H. Xu, Q. Xing, J. Yan, S. Lu, H. Fan, L. Yang, and Y. Cheng, “Probabilistic solar irradiance forecasting based on xgboost,” *Energy Reports*, vol. 8, pp. 1087–1095, 2022.

[21] A. V. Dorogush, V. Ershov, and A. Gulin, “Catboost: gradient boosting with categorical features support,” 2018. [Online]. Available: https://arxiv.org/abs/1810.11363

[22] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, “Catboost: unbiased boosting with categorical features,” in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf

[23] M. B. Tasre, P. P. Bedekar, and V. N. Ghate, “Daily peak load forecasting using ann,” in *2011 Nirma University International Conference on Engineering*, 2011, pp. 1–6.

[24] J. L. Elman, “Finding structure in time,” *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[25] M. I. Jordan, “Chapter 25 - serial order: A parallel distributed processing approach,” in *Neural-Network Models of Cognition*, ser. Advances in Psychology, J. W. Donahoe and V. Packard Dorsel, Eds. North-Holland, 1997, vol. 121, pp. 471–495.

[26] A. Mellit, A. M. Pavan, and V. Lughi, “Deep learning neural networks for short-term photovoltaic power forecasting,” *Renewable Energy*, vol. 172, pp. 276–288, 2021.

[27] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, “Recent advances in convolutional neural networks,” *Pattern Recognition*, vol. 77, pp. 354–377, 2018.

[28] P. Li, K. Zhou, X. Lu, and S. Yang, “A hybrid deep learning model for short-term pv power forecasting,” *Applied Energy*, vol. 259, p. 114216, 2020.

[29] A. Agga, A. Abbou, M. Labbadi, and Y. El Houm, “Short-term self consumption pv plant power production forecasts based on hybrid cnn-lstm, convlstm models,” *Renewable Energy*, vol. 177, pp. 101–112, 2021.

[30] A. L’Heureux, K. Grolinger, and M. A. M. Capretz, “Transformer-based model for electrical load forecasting,” *Energies*, vol. 15, no. 14, 2022. [Online]. Available: https://www.mdpi.com/1996-1073/15/14/4993

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[32] S. Chemetova, P. Santos, and M. Ventim-Neves, “Load forecasting in electrical distribution grid of medium voltage,” in *Technological Innovation for Cyber-Physical Systems*, L. M. Camarinha-Matos, A. J. Falcão, N. Vafaei, and S. Najdi, Eds. Cham: Springer International Publishing, 2016, pp. 340–349.

[33] A. Srivastava, D. Steen, L. Tuan, O. Carlson, I. Bouloumpasis, T. Tran-Quoc, and L. Lemius, “Development of a dso support tool for congestion forecast,” *IET Generation, Transmission & Distribution*, vol. 15, 12 2021.

[34] G. Barchi, M. Pierro, and D. Moser, “Predictive energy control strategy for peak shaving and shifting using bess and pv generation applied to the retail sector,” *Electronics*, vol. 8, no. 5, p. 526, 2019.

[35] L. Xu, S. Wang, and R. Tang, “Probabilistic load forecasting for buildings considering weather forecasting uncertainty and uncertain peak load,” *Applied energy*, vol. 237, pp. 180–195, 2019.

[36] M. A. A. Amin and M. A. Hoque, “Comparison of arima and svm for short-term load forecasting,” in *2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON)*, 2019, pp. 1–6.

[37] E. Vinagre, T. Pinto, S. Ramos, Z. Vale, and J. M. Corchado, “Electrical energy consumption forecast using support vector machines,” in *2016 27th International Workshop on Database and Expert Systems Applications (DEXA)*, 2016, pp. 171–175.

[38] X. Liao, N. Cao, M. Li, and X. Kang, “Research on short-term load forecasting using xgboost based on similar days,” in *2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, 2019, pp. 675–678.

[39] M. M. Gökçe and E. Duman, “Performance comparison of simple regression, random forest and xgboost algorithms for forecasting electricity demand,” in *2022 3rd International Informatics and Software Engineering Conference (IISEC)*, 2022, pp. 1–6.

[40] 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, 2017. [Online]. Available: https://www.mdpi.com/1996-1073/10/8/1168

[41] W. Xiang, P. Xu, J. Fang, Q. Zhao, Z. Gu, and Q. Zhang, "Multi-dimensional data-based medium- and long-term power-load forecasting using double-layer catboost," *Energy Reports*, vol. 8, pp. 8511–8522, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352484722012033

[42] G. Tziolis, C. Spanias, M. Theodoride, S. Theocharides, J. Lopez-Lorente, A. Livera, G. Makrides, and G. E. Georghiou, "Short-term electric net load forecasting for solar-integrated distribution systems based on bayesian neural networks and statistical post-processing," *Energy*, vol. 271, p. 127018, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544223004127

[43] B. Neupane, T. B. Pedersen, and B. Thiesson, "Utilizing device-level demand forecasting for flexibility markets - full version," 2018. [Online]. Available: https://arxiv.org/abs/1805.00702

[44] P. Faria, A. Jozi, and Z. Vale, "Forecasting refrigerators consumption to support their aggregated participation in demand response," in *2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*, 2020, pp. 1–6.

[45] M. Razghandi and D. Turgut, "Residential appliance-level load forecasting with deep learning," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–6.

[46] J. Amantegui, H. Morais, and L. Pereira, "Benchmark of electricity consumption forecasting methodologies applied to industrial kitchens," *Buildings*, vol. 12, no. 12, 2022. [Online]. Available: https://www.mdpi.com/2075-5309/12/12/2231

[47] S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, "Big data preprocessing: methods and prospects," *Big Data Analytics*, vol. 1, no. 1, p. 9, Nov. 2016. [Online]. Available: https://doi.org/10.1186/s41044-016-0014-0

[48] F. L. Quilumba, W.-J. Lee, H. Huang, D. Y. Wang, and R. Szabados, "An overview of ami data preprocessing to enhance the performance of load forecasting," in *2014 IEEE Industry Application Society Annual Meeting*, 2014, pp. 1–7.

[49] R. Martins, "Data-driven modeling of energy consumption in industrial kitchens - detection of activations and unsupervised classification," MSc, Técnico Lisboa, University of Lisbon, Lisbon, Portugal, Nov. 2022.

[50] K.-J. Park and S.-Y. Son, "A novel load image profile-based electricity load clustering methodology," *IEEE Access*, vol. 7, pp. 59 048–59 058, 2019.

[51] Q. Jiang, Y. Cheng, H. Le, C. Li, and P. X. Liu, "A stacking learning model based on multiple similar days for short-term load forecasting," *Mathematics*, vol. 10, no. 14, 2022. [Online]. Available: https://www.mdpi.com/2227-7390/10/14/2446

[52] X. Yao, X. Fu, and C. Zong, "Short-term load forecasting method based on feature preference strategy and lightgbm-xgboost," *IEEE Access*, vol. 10, pp. 75 257–75 268, 2022.

[53] M. Abdel-Nasser and K. Mahmoud, "Accurate photovoltaic power forecasting models using deep lstm-rnn," *Neural Computing and Applications*, vol. 31, no. 7, pp. 2727–2740, 2019.

[54] K. Wu, J. Wu, L. Feng, B. Yang, R. Liang, S. Yang, and R. Zhao, "An attention-based cnn-lstm-bilstm model for short-term electric load forecasting in integrated energy system," *International Transactions on Electrical Energy Systems*, vol. 31, no. 1, p. e12637, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/2050-7038.12637

[55] J. Foster, X. Liu, and S. McLoone, "Adaptive sliding window load forecasting," in *2017 28th Irish Signals and Systems Conference (ISSC)*, 2017, pp. 1–6.

[56] I. A. Khan, A. Akber, and Y. Xu, "Sliding window regression based short-term load forecasting of a multi-area power system," in *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, 2019, pp. 1–5.

[57] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Statistics Surveys*, vol. 4, no. none, pp. 40 – 79, 2010. [Online]. Available: https://doi.org/10.1214/09-SS054

[58] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.

[59] J. Xie and T. Hong, "Load forecasting using 24 solar terms," *Journal of Modern Power Systems and Clean Energy*, vol. 6, no. 2, pp. 208–214, 2018.

[60] A. Hasan, O. Kalipsiz, and S. Akyokus, "Modeling traders' behavior with deep learning and machine learning methods: Evidence from bist 100 index," *Complexity*, vol. 2020, pp. 1–16, 06 2020.

[61] R. Khalid and N. Javaid, "A survey on hyperparameters optimization algorithms of forecasting models in smart grid," *Sustainable Cities and Society*, vol. 61, p. 102275, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210670720304960

[62] M. Zahid, F. Ahmed, N. Javaid, R. A. Abbasi, H. S. Zainab Kazmi, A. Javaid, M. Bilal, M. Akbar, and M. Ilahi, "Electricity price and load forecasting using enhanced convolutional neural network and enhanced support vector regression in smart grids," *Electronics*, vol. 8, no. 2, 2019. [Online]. Available: https://www.mdpi.com/2079-9292/8/2/122

[63] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A next-generation hyperparameter optimization framework,” ser. KDD ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2623–2631. [Online]. Available: https://doi.org/10.1145/3292500.3330701

[64] M. Massaoudi, S. S. Refaat, I. Chihi, M. Trabelsi, F. S. Oueslati, and H. Abu-Rub, “A novel stacked generalization ensemble-based hybrid lgbm-xgb-mlp model for short-term load forecasting,” *Energy*, vol. 214, p. 118874, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544220319812

[65] Z. Jing, H. Ngan, Y. Wang, Y. Zhang, and J. Wang, “Study on the convergence property of re learning model in electricity market simulation,” in *8th International Conference on Advances in Power System Control, Operation and Management (APSCOM 2009)*, 2009, pp. 1–5.

[66] T. Pinto, Z. Vale, T. Sousa, I. Praça, G. Santos, and H. Morais, “Adaptive learning in agents behaviour: A framework for electricity markets simulation,” *Integrated Computer-Aided Engineering*, vol. 21, pp. 399–415, 05 2014.

[67] I. K. Nti, M. Teimeh, O. Nyarko-Boateng, and A. F. Adekoya, “Electricity load forecasting: a systematic review,” *Journal of Electrical Systems and Information Technology*, vol. 7, no. 1, p. 13, Sep. 2020. [Online]. Available: https://doi.org/10.1186/s43067-020-00021-8

[68] L. Pereira, V. Aguiar, and F. Vasconcelos, “Future industrial kitchen: Challenges and opportunities,” in *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, ser. BuildSys ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 163–164. [Online]. Available: https://doi.org/10.1145/3360322.3360872

[69] A. Nespoli, E. Ogliari, S. Pretto, M. Gavazzeni, S. Vigani, and F. Paccanelli, “Electrical load forecast by means of lstm: The impact of data quality,” *Forecasting*, vol. 3, no. 1, pp. 91–101, 2021. [Online]. Available: https://www.mdpi.com/2571-9394/3/1/6

[70] B. Neupane, T. B. Pedersen, and B. Thiesson, “Towards flexibility detection in device-level energy consumption,” in *Data Analytics for Renewable Energy Integration*, W. L. Woon, Z. Aung, and S. Madnick, Eds. Cham: Springer International Publishing, 2014, pp. 1–16.

[71] E. Ela, M. Milligan, A. Bloom, A. Botterud, A. Townsend, and T. Levin, “Evolution of wholesale electricity market design with increasing levels of renewable generation,” 9 2014. [Online]. Available: https://www.osti.gov/biblio/1159375

[72] P. Denholm, J. Novacheck, J. Jorgenson, and M. O’Connell, “Impact of flexibility options on grid economic carrying capacity of solar and wind: Three case studies,” 12 2016. [Online]. Available: https://www.osti.gov/biblio/1339345

[73] B. Mohandes, M. S. E. Moursi, N. Hatziargyriou, and S. E. Khatib, “A review of power system flexibility with high penetration of renewables,” *IEEE Transactions on Power Systems*, vol. 34, no. 4, pp. 3140–3155, 2019.

[74] M. Fazeli, J. B. Ekanayake, P. M. Holland, and P. Igic, “Exploiting pv inverters to support local voltage—a small-signal model,” *IEEE Transactions on Energy Conversion*, vol. 29, no. 2, pp. 453–462, 2014.

[75] P. Balcombe, D. Rigby, and A. Azapagic, “Motivations and barriers associated with adopting microgeneration energy technologies in the uk,” *Renewable and Sustainable Energy Reviews*, vol. 22, pp. 655–666, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032113001123

[76] A. Mashlakov, E. Pournaras, P. H. Nardelli, and S. Honkapuro, “Decentralized cooperative scheduling of prosumer flexibility under forecast uncertainties,” *Applied Energy*, vol. 290, p. 116706, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261921002270

A

Hyperparameter Optimization Results

This appendix contains the results obtained in the hyperparameter optimization, using the *Optuna* environment. Tables A.1 and A.2 contain the sets of hyperparameters obtained for each appliance's data, for XGBoost, in both day-ahead and hour-ahead analysis.Tables A.3 and A.4 present the hyperparameters obtained for CatBoost, for both kitchens and scenarios, while the results obtained for the LSTM model are presented in Tables A.5 and A.6.

Kitchen	Device	Subsample	Column Sample by Tree	Tree Depth	Learning Rate	λ	α
Kitchen 1	Blast Chiller	0.8	0.2	2	0.005	1	1
	Convection Oven 1	0.8	0.7	3	0.01	1	10
	Convection Oven 2	0.8	0.8	4	0.02	50	5
	Dish Washer	0.4	0.5	2	0.01	10	5
	Drinks Refrigerator	0.2	0.2	4	0.01	50	5
	Dual Fryer	0.4	0.5	2	0.01	50	10
	Fish Refrigerator	0.8	0.5	4	0.02	1	5
	Freezer	0.8	0.2	2	0.005	10	50
	Garde Manger 1	0.4	0.7	3	0.01	1	1
	Garde Manger 2	0.6	0.4	2	0.01	1	1
	Ice Machine	0.2	0.2	2	0.005	50	5
	Infrared Shelf	0.8	0.7	3	0.005	50	10
	Meat Refrigerator	0.8	0.2	3	0.01	1	5
	Mise en Place	0.2	0.6	2	0.005	1	1
	Salamander 1	0.2	0.4	2	0.02	50	5
	Salamander 2	0.2	0.2	4	0.01	10	10
	Vegetables Refrigerator	0.4	0.7	3	0.02	1	10
Kitchen 2	Blast Chiller	0.2	0.6	4	0.01	50	10
	Convection Oven	0.7	0.2	4	0.01	50	10
	Exhaust	0.7	0.2	3	0.02	1	1
	Heating Lamp 1	0.7	0.8	3	0.02	5	5
	Heating Lamp 2	0.2	0.5	3	0.02	5	5
	Hothouse	0.8	0.5	4	0.02	50	10
	Induction Plate	0.2	0.4	4	0.02	50	1
	Microwave	0.7	0.7	3	0.02	1	1
	Refrigerator 1	0.6	0.8	4	0.02	1	5
	Refrigerator 2	0.8	0.2	4	0.02	50	5
	Refrigerator 3	0.2	0.2	2	0.005	5	10
	Refrigerator 4	0.2	0.4	2	0.02	50	5
	Sous Vide	0.2	0.6	4	0.02	50	50

Table A.1: Hyperparameters for XGBoost - day-ahead analysis

Kitchen	Device	Subsample	Column Sample by Tree	Tree Depth	Learning Rate	λ	α
Kitchen 1	Blast Chiller	0.6	0.7	4	0.01	10	10
	Convection Oven 1	0.8	0.8	3	0.02	50	10
	Convection Oven 2	0.2	0.6	3	0.005	50	5
	Dish Washer	0.2	0.7	3	0.01	50	1
	Drinks Refrigerator	0.8	0.7	4	0.01	50	50
	Dual Fryer	0.2	0.8	4	0.02	5	10
	Fish Refrigerator	0.4	0.5	3	0.005	50	5
	Freezer	0.7	0.4	4	0.005	5	5
	Garde Manger 1	0.8	0.8	4	0.005	5	50
	Garde Manger 2	0.2	0.6	3	0.01	50	50
	Ice Machine	0.2	0.8	4	0.01	50	1
	Infrared Shelf	0.5	0.8	3	0.01	10	10
	Meat Refrigerator	0.7	0.6	3	0.01	1	10
	Mise en Place	0.6	0.7	4	0.02	50	1
	Salamander 1	0.5	0.8	3	0.01	5	50
	Salamander 2	0.2	0.8	3	0.005	1	5
	Vegetables Refrigerator	0.6	0.6	4	0.01	50	10
Kitchen 2	Blast Chiller	0.7	0.5	2	0.01	1	5
	Convection Oven	0.2	0.8	2	0.01	50	50
	Exhaust	0.5	0.8	4	0.01	10	5
	Heating Lamp 1	0.2	0.8	4	0.01	50	1
	Heating Lamp 2	0.2	0.8	4	0.005	5	5
	Hothouse	0.2	0.8	2	0.005	50	50
	Induction Plate	0.4	0.8	4	0.005	10	50
	Microwave	0.8	0.8	4	0.02	5	50
	Refrigerator 1	0.4	0.8	4	0.01	10	5
	Refrigerator 2	0.5	0.6	2	0.02	50	10
	Refrigerator 3	0.2	0.8	4	0.005	50	50
	Refrigerator 4	0.4	0.8	4	0.02	50	1
	Sous Vide	0.7	0.7	4	0.01	10	10

Table A.2: Hyperparameters for XGBoost - hour-ahead analysis

Kitchen	Device	Subsample	Tree Depth	Learning Rate	λ
Kitchen 1	Blast Chiller	0.7	5	0.02	1
	Convection Oven 1	0.7	5	0.01	50
	Convection Oven 2	0.5	3	0.01	10
	Dish Washer	0.6	3	0.02	50
	Drinks Refrigerator	0.6	5	0.01	5
	Dual Fryer	0.2	3	0.02	50
	Fish Refrigerator	0.2	5	0.005	5
	Freezer	0.6	3	0.02	5
	Garde Manger 1	0.7	4	0.02	1
	Garde Manger 2	0.8	3	0.01	5
	Ice Machine	0.5	5	0.02	5
	Infrared Shelf	0.5	5	0.02	5
	Meat Refrigerator	0.7	5	0.02	5
	Mise en Place	0.5	3	0.005	1
	Salamander 1	0.4	5	0.02	10
	Salamander 2	0.8	3	0.005	50
	Vegetables Refrigerator	0.4	3	0.01	1
Kitchen 2	Blast Chiller	0.6	5	0.01	50
	Convection Oven	0.4	5	0.02	1
	Exhaust	0.8	5	0.02	1
	Heating Lamp 1	0.8	5	0.01	1
	Heating Lamp 2	0.2	5	0.01	1
	Hothouse	0.2	5	0.02	50
	Induction Plate	0.2	5	0.02	50
	Microwave	0.7	5	0.01	5
	Refrigerator 1	0.4	5	0.01	1
	Refrigerator 2	0.6	4	0.02	1
	Refrigerator 3	0.7	5	0.01	5
	Refrigerator 4	0.4	5	0.02	1
	Sous Vide	0.2	5	0.02	50

Table A.3: Hyperparameters for CatBoost - day-ahead analysis

Kitchen	Device	Subsample	Tree Depth	Learning Rate	λ
Kitchen 1	Blast Chiller	0.6	5	0.02	10
	Convection Oven 1	0.6	4	0.01	50
	Convection Oven 2	0.6	4	0.01	5
	Dish Washer	0.8	5	0.02	5
	Drinks Refrigerator	0.6	4	0.02	1
	Dual Fryer	0.2	4	0.02	50
	Fish Refrigerator	0.6	4	0.01	1
	Freezer	0.8	4	0.02	10
	Garde Manger 1	0.6	3	0.005	50
	Garde Manger 2	0.7	4	0.01	50
	Ice Machine	0.4	3	0.02	5
	Infrared Shelf	0.8	4	0.02	1
	Meat Refrigerator	0.7	4	0.01	50
	Mise en Place	0.5	5	0.02	50
	Salamander 1	0.6	4	0.01	1
	Salamander 2	0.8	4	0.02	10
	Vegetables Refrigerator	0.8	4	0.005	5
Kitchen 2	Blast Chiller	0.2	3	0.01	50
	Convection Oven	0.4	5	0.005	10
	Exhaust	0.4	5	0.02	1
	Heating Lamp 1	0.5	4	0.02	50
	Heating Lamp 2	0.8	3	0.02	10
	Hothouse	0.5	5	0.02	5
	Induction Plate	0.7	5	0.02	50
	Microwave	0.5	5	0.02	1
	Refrigerator 1	0.6	5	0.02	10
	Refrigerator 2	0.8	5	0.02	50
	Refrigerator 3	0.8	5	0.01	1
	Refrigerator 4	0.5	4	0.02	1
	Sous Vide	0.8	5	0.01	1

Table A.4: Hyperparameters for CatBoost - hour-ahead analysis

Kitchen	Device	Nº of layers	Units	Activation Function	Dropout
Kitchen 1	Blast Chiller	1	64	linear	0.2
	Convection Oven 1	1	32	linear	0
	Convection Oven 2	1	16	linear	0.2
	Dish Washer	2	64	tanh	0.2
	Drinks Refrigerator	3	16	tanh	0.2
	Dual Fryer	1	16	tanh	0.1
	Fish Refrigerator	2	64	linear	0.2
	Freezer	2	64	linear	0.1
	Garde Manger 1	2	16	linear	0.2
	Garde Manger 2	3	32	linear	0.1
	Ice Machine	2	16	tanh	0.1
	Infrared Shelf	3	64	tanh	0
	Meat Refrigerator	1	16	tanh	0
	Mise en Place	1	64	tanh	0.2
	Salamander 1	2	64	linear	0.2
	Salamander 2	3	32	tanh	0.2
	Vegetables Refrigerator	1	16	linear	0.1
Kitchen 2	Blast Chiller	2	16	linear	0.1
	Convection Oven	1	16	linear	0
	Exhaust	2	32	tanh	0.2
	Heating Lamp 1	1	64	tanh	0.2
	Heating Lamp 2	1	64	sigmoid	0
	Hothouse	1	16	tanh	0.1
	Induction Plate	1	64	tanh	0.2
	Microwave	3	64	sigmoid	0
	Refrigerator 1	1	32	tanh	0.1
	Refrigerator 2	1	64	sigmoid	0.2
	Refrigerator 3	1	16	tanh	0.2
	Refrigerator 4	2	32	tanh	0.1
	Sous Vide	2	16	tanh	0.2

Table A.5: Hyperparameters for LSTM - day-ahead analysis

Kitchen	Device	Nº of layers	Units	Activation Function	Dropout
Kitchen 1	Blast Chiller	1	32	tanh	0.1
	Convection Oven 1	2	32	linear	0
	Convection Oven 2	1	32	linear	0
	Dish Washer	1	16	tanh	0.2
	Drinks Refrigerator	2	16	tanh	0.2
	Dual Fryer	3	16	tanh	0.1
	Fish Refrigerator	2	16	linear	0.2
	Freezer	2	64	tanh	0.1
	Garde Manger 1	2	64	linear	0.2
	Garde Manger 2	3	32	linear	0.1
	Ice Machine	2	16	linear	0.1
	Infrared Shelf	3	64	tanh	0.2
	Meat Refrigerator	2	16	tanh	0.1
	Mise en Place	1	16	tanh	0.2
	Salamander 1	3	32	linear	0.1
	Salamander 2	3	16	linear	0.2
	Vegetables Refrigerator	1	16	tanh	0.1
Kitchen 2	Blast Chiller	3	32	tanh	0.1
	Convection Oven	2	16	linear	0.1
	Exhaust	2	32	tanh	0.2
	Heating Lamp 1	1	64	tanh	0.2
	Heating Lamp 2	1	64	sigmoid	0.2
	Hothouse	2	16	tanh	0.1
	Induction Plate	2	64	tanh	0.2
	Microwave	1	32	sigmoid	0.2
	Refrigerator 1	2	32	linear	0
	Refrigerator 2	1	64	tanh	0.2
	Refrigerator 3	1	16	tanh	0
	Refrigerator 4	1	32	tanh	0.1
	Sous Vide	2	16	linear	0.1

Table A.6: Hyperparameters for LSTM - hour-ahead analysis