Forecasting Electricity Consumption in Industrial Kitchens

Jorge Ruiz Amantegui

Thesis to obtain the Master of Science Degree in

Energy Engineering and Management

Supervisors: Prof. Amáncio Lucas de Sousa Pereira

Prof. Hugo Gabriel Valente Morais

Examination Committee

Chairperson: Prof. Duarte de Mesquita e Sousa

Supervisor: Prof. Hugo Gabriel Valente Morais

Member of the Committee: Prof. Tiago Manuel Ferreira Pinto

September 2022

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

To Magalí and Tomás,

for always being there with me.

Acknowledgements

I would like to express my deep gratitude to Professor Lucas Pereira and Professor Hugo Morais for their time, support, insight, and patience that made this thesis possible.

I would also like to show my gratitude to all my friends, from whom I have learned so much and have helped me become the person I am today.

Last but not least, to my parents, for their never-ending support.

Thank you.

Abstract

As the global demand for electricity keeps increasing, electricity generation from coal and gas hit record levels and as a result the global electricity sectors annual carbon dioxide emissions reach the highest values ever recorded. To fight the energy crisis and global warming, penetration of renewable energy sources (RES) and energy efficiency are of upmost importance.

With this idea in mind, this thesis looks at the energy consumption and consumption forecast in a very intensive and little researched energy consuming business such as industrial kitchens (IKs).

The main goal of this research is to contribute to the study of data load forecasting, in this case restaurant kitchens data load. Specifically for scenarios with low amounts of data. Two scenarios are compared to each other: data consisting of a virtual aggregate and a real aggregate. Different machine learning (ML) models are built for different time horizon testing. The models are tested for each individual appliance and for a virtual aggregate (the sum of the loads of individual appliances). The goal is to test if there is an improvement in the accuracy of the forecast when forecasting for each individual appliance instead of the total load of the IK. Three prediction algorithms are used to forecast, these are: Random Forest, Long Short-Term Memory and Facebook Prophet.

The performance of the different algorithms are compared against each other making use of two different performance metrics, and so will the performance of the forecast of the virtual aggregate against sum of the forecast of the individual appliances.

The results obtained in the elaboration of this thesis don’t show an improved accuracy in the forecast of the individual appliances against the total load. However, there is room for improvement and this methodology already presents merits and benefits.

Keywords

Industrial Kitchens, Load forecasting, Long Short-Term Memory, Machine Learning, Prophet, Random Forest, Restaurants

Resumo

Como a procura global de eletricidade continua a aumentar, a produção de eletricidade a partir do carvão e do gás atingiu níveis recorde e, como resultado, as emissões anuais de dióxido de carbono dos sectores globais de eletricidade atingem os valores mais elevados alguma vez registados. Para combater a crise energética e o aquecimento global, a penetração das fontes de energia renováveis (FER) e a eficiência energética são da maior importância.

Com esta ideia em mente, esta tese pretende analisas o consumo de energia e comparar com previsão do consumo em cozinhas industriais (IK) que sô caracterizadas por serem consumidores intensivo de energia.

O principal objetivo desta investigação é contribuir para o estudo da previsão da carga de dados, neste caso, a carga. Especificamente, as previsões são feitas para cenários cenários com quantidades baixas de dados. Dois cenários são comparados um com o outro: dados compostos por um agregado virtual e um agregado real. Diferentes modelos de machine learning (ML) são construídos para diferentes horizontes temporais. Os modelos são testados para cada aparelho individual e para um agregado virtual (a soma das cargas de aparelhos individuais). O objectivo é testar se há uma melhoria na precisão da previsão ao fazer previsões para cada aparelho individual em vez da carga total da IK. Três algoritmos de previsão são utilizados para prever, nomeadamente: Random Fotest, Long Short-Term Memory e Prophet propost o pela Facebook.

O desempenho dos diferentes algoritmos são comparados, assim como o desempenho da previsão do agregado virtual em comparação com a soma da previsão dos aparelhos individuais.

Os resultados obtidos na elaboração desta tese não mostram uma maior precisão na previsão dos aparelhos individuais em relação à carga total. No entanto, há margem para melhorias e esta metodologia já apresenta méritos e benefícios.

Palavras-chave

Cozinhas Industriais, Previsão de carga, Long Short-Term Memory, Machine Learning, Prophet Software, Random Forest, Restaurantes

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ANN	Artificial Neural Network
AR	Autoregressive
ARIMA	Autoregressive Integrated Moving Average
ARIMAX	Autoregressive Integrated Moving Average including exogenous variables
CNN	Convolutional Neural Network
DL	Deep Learning
DT	Decision Tree
EU	European Union
FFNN	Feed Forward Neural Network
IK	Industrial Kitchen
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
ML	Machine Learning
MLP	Multi-Layer Perceptron
PV	Photo Voltaic
RES	Renewable Energy Sources
RF	Random Forest
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SARIMA	Seasonal Autoregressive Integrated Moving Average
SARIMAX	Seasonal Autoregressive Integrated Moving Average including exogenous variables
SDG	Sustainable Development Goals
SVM	Support Vector Machine
SVR	Support Vector Regressor
WEFN	Water-Energy-Food Nexus

List of Symbols

φ	Coefficient of autoregressive model
θ	Coefficient of the moving average
δ	Vector of rate of adjustment
ε	Forecasting Error
ψ	Adjusting parameter. Prophet constant growth model
σ	Collection of random variables in a decision tree

Chapter 1

Introduction

This chapter gives a brief overview of the whole work. The structure of the document will be presented here as well as the scope, motivations and objectives of the thesis.

1.1 Motivation and Context

Since the turn of the twenty-first century, the number of people with access to energy has boomed. In two decades 1400 million people have gained access to electricity just in Africa and Asia [1]. The global demand for electricity keeps increasing, and strong economic growth, combined with more extreme weather conditions than in 2020 boosted the demand by more than 6% in 2021 [2]. This is the largest ever annual increase in absolute terms and the largest percentage rise since 2010 [2]. Figure 1. 1 shows these changes in electricity demand and a prediction for the near future.

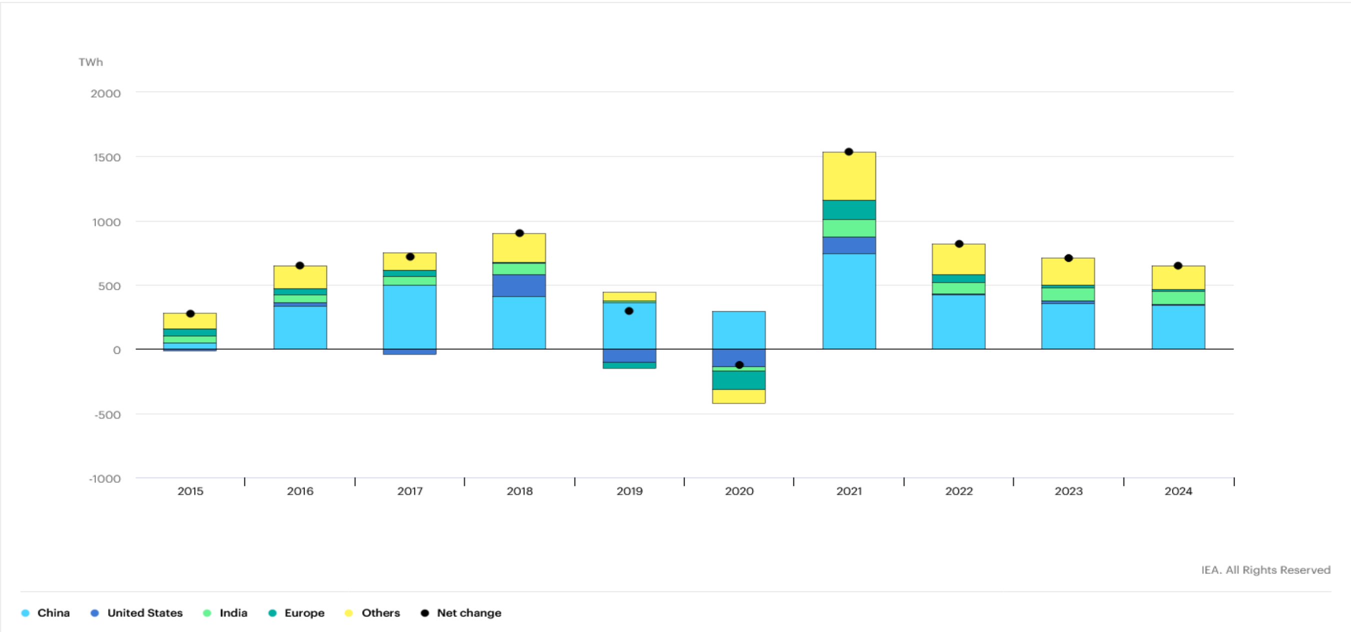

Figure 1. 1 Global changes in electricity consumption [2]

Despite the efforts made in recent years regarding the development and implementation of renewable energy sources (RES), electricity generation from coal and gas hit record levels and as a result, the global electricity sector's annual carbon dioxide emissions reached the highest value ever recorded [2]

To fight climate change and the increase in energy demand the United Nations sets seventeen goals in the Sustainable Development Goals (SDG). Out of these 17 goals, the seventh is affordable and clean energy, this seventh goal is divided into five sub-categories, two of which are described in targets 7.2 and 7.3. Target 7.2 establishes the need to increase the share of RES in the global mix. Despite the growth seen in these kinds of energies in recent times, the total share of them over the total energy consumption only amounts to 17%. The deployment levels of RES are still far away from meeting those established in SDG7 to achieve decarbonization of the energy sector.

Increasing the penetration of RES in the grid is not enough to achieve the goals set, that is why target 7.3 refers to energy efficiency. The goal is to increase the energy efficiency by 2030 at a rate of 2.6% annually (doubling the average annual increase in efficiency between 1990 and 2010) [3], a very ambitious goal.

To achieve the goals set by the UN regarding electricity consumption, it is crucial to take a closer look at the consumption of industrial kitchens (IK). According to Mudie et al. [4], electricity consumption in IK is vastly greater than literature estimates, their research carried out in the United Kingdom (UK) showed that the energy used for managed sites was 7.52 TWh while the previous estimations were 0.02 TWh [4]. Kitchen and food-related activities greatly outweigh other energy using activities [5]. IKs use between 5 and 7 times more energy per square meter than other commercial building spaces [6] as a consequence of this, studies show the carbon footprint in the contract catering sector is 80% higher than initially estimated [7].

Despite the high energy usage of IK [4], [5], very little research exists related to their current consumption and energy reduction strategies [7]. Much of the work that has been done on this topic is related to energy efficiency matters, in consonance with target 7.3 from SDG7 and with the European Union (EU) DIRECTIVE (EU) 2018/2002 which aims at increasing energy efficiency by 32.5% by 2030. This directive mostly forgets about IK but does include energy labelling and eco-design as a means to achieve the energy efficiency increase, this can be related to kitchens as many of the appliances must be labelled.

Nonetheless, some research has been carried out to study the impact of IK on the consumption of electricity. Such is the case for nexIK [8], a project with the goal to propose a new holistic methodology to quantify the interactions between Water, Energy and Food (known as Water-Energy-Food Nexus WEFN) in IKs. The project intends to collect data that will then be used to model the WEFN following a data-driven system. This is a challenge of great importance to effectively install RES and energy storage in the IKs. This is of course related to the goals of increasing energy efficiency and the implementation of RES in the energy mix. Forecasting energy consumption helps to increase the penetration of RES thanks to better coordination of generation and electricity consumption This thesis will explore new methodologies of forecasting timeseries to study if there is room to improve the performance of said forecast.

1.2 Objectives

Given the above context, this thesis explores a new methodology to study the forecast of the electricity consumption of IK. Even though timeseries forecasting is a field with many studies done, but very few of them are related to IK.

Three main objectives developed in this thesis will be:

- Develop methods to forecast the virtual aggregated power demand data.
- Develop methods to forecast individual appliances power demand.
- To compare the performance of the forecast of the virtual aggregate against the sum of forecast

of the individual appliances, using different performance metrics, to study the value in forecasting for each individual appliance.

- Compare the performance of different forecasting algorithms.

It is important to define the concept of virtual aggregate. This is the load of the restaurant defined as the sum of the loads of each individual appliance used in the restaurant. It is different from the real aggregate (the measured value for the load) because it does not include certain elements such as lighting or air ventilation.

Since finding suitable forecasting methods for an effective management of energy resources is of great importance to improve the efficiency in energy consumption and to decrease its impact on the environment [9] the objectives of this thesis support the international goals to achieve a more energy efficient industry and to improve the implementation of RES in the energy mix.

1.3 Document Organization

This work will be divided into several chapters. First of all, the state of the art will begin by exploring the high energy consumption in kitchens around the world, followed by an introduction to machine learning (ML) as a means to forecast said energy consumption. The current state of forecasting timeseries using ML will be presented.

The third chapter presents the methodology used to forecast the energy consumption in a high-end restaurant in Portugal and compares the results. Three different forecast algorithms will be used.

The fourth chapter will present how this methodology was applied to the available and data and the results obtained. The results are also discussed in this chapter. Finally, a chapter to involve the final conclusions of the work will be presented.

.

Chapter 2

State of the Art

This chapter will review the current literature on energy consumption in industrial kitchens and the literature on load forecasting of virtual aggregate and real aggregate.

2.1 Electricity Consumption on Industrial Kitchens

Industrial kitchens (IKs) are among the highest electricity consumers, and research shows that the consumption of these kitchens is even higher than originally estimated by the literature[5]. Despite the important energy usage of IK, few work has been done to study the kitchens energy consumption [5].

Research wise, the majority of the efforts have been devoted to developing benchmarking strategies to enable accurate and fair comparisons across sites. These research efforts are motivated by the many different cultures and culinary habits that lead to very different levels of electric energy consumption. For example, *Figure 2. 1* depicts the annual consumption of 772 restaurants in the UK, which as can be observed varies significantly across the site.

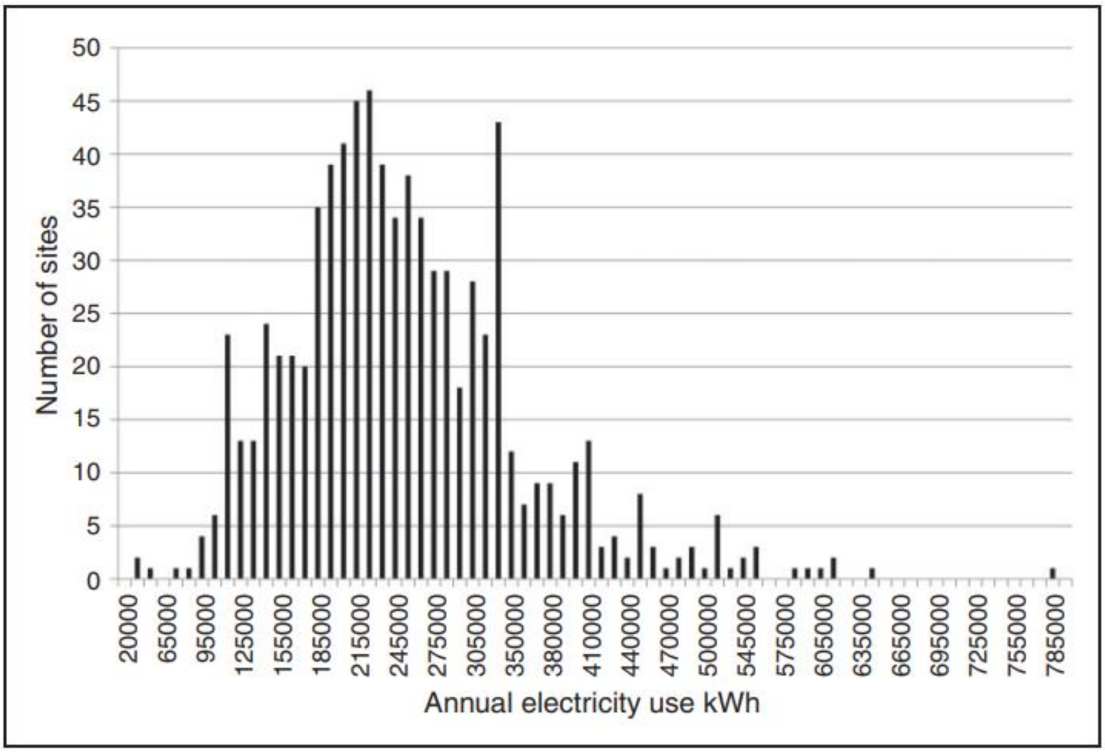

Figure 2. 1 Annual electricity use in restaurants in the UK (Source [4]*)*

A study carried out in the UK, in the spirit of reducing energy consumption from restaurant kitchens identified the most consuming appliances [5]. The appliances contributing mostly to the overall consumption are those in the refrigeration section (41%), followed by fryers (13%), combination ovens (12%) and grills (12%).

In China, on the other hand, it is considered that Chinese barbeque (BBQ) pork and poultry are the most popular menu items in authentic Chinese restaurants, therefore Chan et al. [10], seeing the lack of information regarding this subject, carried a study of the energy usage of Chinese BBQ. This study showed that the cost of the energy consumption for roasting a piglet could amount for up to 10% of the whole cost of serving such a dish (raw materials, wages of the workers, infrastructure, etc.) and that there is room for improvement in said area. Furthermore, from the Hong Kong Polytechnic University, with the understanding of a need to shift from a fossil fuel energy system to a renewable one, a study was carried out to propose a method for establishing energy consumption benchmarks in commercial complexes [11], including those with commercial kitchens. This study understands that IKs are considered intensive electricity users and much of their electric consumption can be explained by the equipment that the kitchen has installed. Other factors affecting the electricity consumption include the number of seatings, number of meals covered, type of dish (fast food, Chinese BBQ, cafeteria, etc.) and the air extraction. This paper draws the attention to the value in recording energy consumption for each individual appliance but accentuate the difficulty of this given that the only reliable source of information for the energy consumption at the time was the energy bill. Now a day smart meters could be installed per appliance.

In the USA, some initial studies were carried out a long time ago to estimate the consumption of the food service industry with the knowledge that energy-management was of great importance [12], regardless of this, the work carried out in more recent times to study this field is scarce even though the restaurant industry represents more than 4% of the gross domestic product of the USA and is responsible for approximately 33% of all the electricity consumed [13] according to a 2013 study, an impressive amount of energy. Some of the work that has been carried out to benchmark the energy consumption on IKs deals with the issue of defining an appropriate metric to compare performances for individual kitchens [14]. This occurs because of the high number of factors that can affect the final consumption of an IK. Hedrick et.al [14] propose a methodology in which each restaurant can their own utility data to create a suitable metric, this methodology includes the prediction of annual energy use as part of the process. Their study showed that the most important factors determining final energy consumption are the number of meals, hours of operations, location (climate) and building type.

Overall, a study developed by Higgins et al. showed that the majority of the literature regarding the sustainability of restaurants only engages with parts of sustainability, particularly ecological aspects, rather than holistic sustainability [15]. This limits the possibility to achieve more sustainable restaurant options. There are, however, some exceptions to this, in 2017 a holistic evaluation of the carbon and energy impact of different food preparation options in delivering restaurant's menu was carried out in [16]. After contemplating seven energy reduction scenarios the study determined that up to 58% of energy could potentially be saved.

Also in Portugal, researchers benchmarked the electricity consumption of three luxury dining restaurants from five-star hotels [17]. Among the different findings, it was concluded that the surveyed restaurants had very similar levels of electricity consumption, in particular when the number of services is the same (e.g., breakfast, lunch and dinner). Furthermore, it was also observable that the operation of these spaces obeys pre-defined routines which ultimately results in stable energy consumption through the days. Another work regarding Portuguese restaurants proved a relationship between attendance and energy consumption (as it would be expected) and identified refrigeration as the most energy consuming activity [18]. More research shows that the food service sector amounted for approximately 1.1% of the total national electricity and gas consumption of the country [18]. This value differs greatly from other studies in the sector shown above, illustrating that more work needs to be done in this sector.

Regarding the introduction of RES in the operation of IKs, research is very scarce. When using “scholar.google.com” as an academic research engine using the keywords “renewable”,” industrial” and “kitchen” almost all of the work that appears is related to developing biomass fuel from industrial kitchen waste or implementing a heat recovery system. In a similar manner, by searching for the exact desired results “implementation of renewable energy in industrial kitchens” there is not any related work in the first four pages. By changing the keyword “industrial” to “commercial” there is more luck, and some research is found.

Among the works that were found, two researches involve buildings composed of a mix of apartments and one commercial kitchen, these buildings are connected to a small-scale microgrid which supplies renewable energy and electric storage. The microgrids are composed of PV arrays and wind turbines as well as additional back-up, by either being connected to the grid or to a backup diesel generator. Microgrid sizing and costs are generally sensitive to load shape [19]. The research showed that, by implementing a thermal storage there was potential to increase renewable penetration by 7% and decrease renewable curtailment (renewable energy resource that cannot be utilized, and is therefore wasted) by 11 % [20] indicating that multiple types of storage can benefit renewable energy penetration. Even though the goal is to have a building with a 100% penetration of renewable energy sources, sometimes this is not the most economically viable option. In the work done by Aldaoubab et al., after doing an economical analysis it demonstrates that lowering the desired penetration from 100% to 80% cuts the energy cost significantly [19] due to the required sizing of the microgrid. The economical aspect is a crucial concept to move towards a 100% renewable energy penetration.

Not part of a building, but definitely a commercial kitchen, the food truck business has recently become very popular [21] and has several ways to reduce their environmental impact. The first and most obvious way is to reduce the impact caused by the fact that the food truck is a motorized vehicle, by shifting from a classic gasoline or diesel engine to an electric or hybrid. This will reduce the environmental impact at an economic cost. Now, to power the kitchen, most of the food trucks use a generator that typically runs on gasoline, diesel, propane or natural gas [21]. To improve the renewable energy penetration, a generator can be integrated into a solar system. Food trucks have a very limited space, and this must be taken into account when sizing the solar system that could be installed in them. The work done by W. Young shows that a small, limited kitchen with minor loads can be powered fairly easily with enough roof space [21] and food trucks with bigger loads can still greatly benefit from installing a PV system.

2.2 Forecasting Electricity Demand

Energy forecasting is a crucial aspect of energy planning, management, and optimization with the goal of increasing efficiency and decreasing consumption.

As has been mentioned above, the amount of work done regarding energy consumption in IK is not abundant, still, load forecasting is a very explored topic (especially for household energy demand) and many different approaches have been taken to predict a power demand curve. The three main approaches to building energy consumption and modelling forecasting can be classified as physics-based, data-driven and hybrid models [22].

A physical model provides a theoretical input to characterize the underlying physical mechanism of the object of study. It tries to interpret and summarize the effects of a real phenomenon into a few equations. Physical models can, for example, be used to find out the amount of heat lost through the walls of a building (knowing inside and outside temperature, the materials of the wall, the area of the wall, etc.) or the amount of energy a photovoltaic system will generate. Physical models can provide valuable insight into the general physical mechanism and potential knowledge but are often limited to simple systems [23].

An example of a physical model used to forecast energy consumption in a building is presented by X. Lü et al. [23]. In that research, the model is based on an energy balance for a whole building divided into the following key components: indoor air, the envelope of the building, internal heat gain, heat loss from ventilation and the heating load. For each of these components, the equations that govern their behaviour must be developed.

On the other hand, data driven models disregard explicit knowledge of the systems physical behaviour and learn from the input and output data. Data driven models are often referred to black box models. This is because the relationships between the input and the output data are not easily described and can hardly be transposed to a physic-based analysis [22]. They propose a modelling and forecasting frameworks based on data analysis schemes and often include algorithms that benefit from the recent development in the field of machine learning [22]. These models have become very popular for the task of energy forecasting in buildings, providing new case studies, algorithms and results while technical challenges remain [24].

Data driven models can be divided into statistical or conventional models and machine learning (ML) models. The latter can be separated into shallow ML and the most recent deep learning (DL) techniques. The work in this thesis follows a data driven approach for electricity load forecasting. Next, a survey of some of the existing literature in this topic is presented.

2.2.1 Statistical Based Load Forecasting

Two very popular conventional methods of building energy consumption model forecasting are autoregressive models and statistical regression. They provide a good balance between implementation simplicity and forecasting accuracy. They have limitations with respect to the forecasting horizon (forecasting for far away time periods) and to model nonlinear data patterns [22].

Autoregressive models (AR) are one of the most classical modelling and forecasting techniques based on statistical analysis of timeseries. It only requires the training set to be stationary: this means that the statistical properties of the timeseries should be time-invariant. This does not mean that the timeseries do not change over time, just that the way it changes does not change over time. AR models can take several forms, such as an autoregressive integrated moving average (ARIMA). ARIMA models forecast on a combination of the past values in the series (called lags) and the past errors in the series (error lags), they are long memory models because the forecasted value at a moment *(t)* is dependent on the forecasted value on *(t-1)*. This can be seen in equation 2.1, where t is the timestep, $\hat{Y}$ the predicted value and X the timeseries values. φ is the coefficient of autoregressive model, θ is the coefficient of the moving average, C is a constant and ε is the forecasting error.

$$\hat{Y}_t = C + \sum_{i=1}^{p} \varphi X_{t-1} - \sum_{j=1}^{q} \theta \varepsilon_{t-j} \tag{2.1}$$

Basic autoregressive models only consider the recent past historical load demand data points to predict their future states. Therefore, they can only provide short-term forecasting, which limits their application and accuracy [22]. To deal with this, several modifications have been made to ARIMA models to overcome these problems. Such is the case for seasonal ARIMA (SARIMA) models (they consider seasonality of the data) or ARIMAX models (including exogenous variables). These two combined form SARIMAX model. Models presented in [25] and [26] are both examples of ARIMA or its modified versions being used with success in forecasting the energy consumption of a building.

Another very popular form of statistical forecasting is statistical regression, which aims to model a relationship between an output and contributing inputs in the form of an equation [22]. Many different forms of statistical regression have been used before for the goal of forecasting energy consumption in a building: multiple linear regression [27], ordinary least square regressions [28], piecewise linear regression [29], general linear regression [30], elastic net regressions [31], Bayesian regressions [32] and gaussian process regression [33].

Their popularity is mostly related to their simple implementation and relatively explicit formulation linking input contributing variables to output energy consumption, still forecasting the performance of these models are reasonably good for most applications [22]. On the downside, most regression techniques are unable to deal with non-linearity phenomena and a large amount of data is also required to capture all possible scenarios [22].

A more recently developed regression model is Prophet, by Facebook [34]. Prophet is a modular regression model with interpretable parameters. This algorithm has been used to predict energy consumption in various works and in different ways. In [35] and [36] implementing a hybrid prophet-LSTM (Long-Short Term Memory) model to forecast electric load is presented. In [37] the implementation of prophet to use real-time day ahead forecasts, while [38] implements a long-term forecast (one year or beyond) and performed outstandingly in metrics such as accuracy, generalization and robustness.

In [39] a review and comparison of how well Prophet, ARIMA and five different neural network architectures perform on both short and longer forecasting horizons are presented. This research is particular in the sense that its forecast is focused on two communities that experience frequent power outages due to heavy loads on the existing electricity grid. Therefore, the periods with low consumption (such as weekends) were removed as they were of no interest (the goal is to predict possible power outages). Prophet did not outperform the other models, but ARIMA did for 24 hours forecasting horizons. However, the ARIMA model is limited in that it must be trained on the same timeseries that are being predicted. This means the model cannot be transferred effectively.

2.2.2 Machine Learning Based Load Forecasting

ML is a field of AI that gives computers the ability to learn without being explicitly programmed [40]. Generally, ML means that a computer's performance according to some metric for a given task improves with experience [41]. It can occur that DL and ML are terms that often seem interchangeable by how they are described sometimes. A reason for this is that all DL is ML, but not all ML is DL. DL is a sub-category of ML, Figure 2. 2 illustrates this. Table 2. 1 gives a brief overview on ML types of learning.

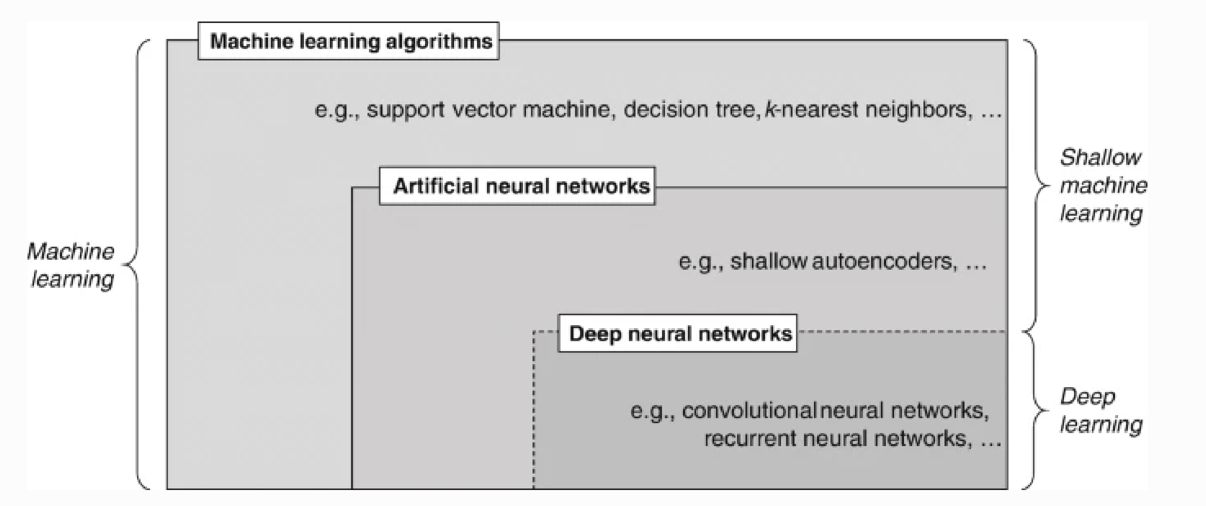

Figure 2. 2 Venn diagram of Machine Learning concepts and classes (Source: [41])

Type	Description
Supervised Learning	Supervised learning requires a training dataset that covers examples for the input *(x)* as well as labelled answers or target values *(y)* for the output. The pairs of input and output data in the training set are used to calibrate the open parameters of the ML model. Once the model has been trained, it can be used to predict the target variable *(y)* for unseen input data *(x).* This type of learning can then be sub-categorized into regression (prediction of a numeric value) and classification (the result is categorical class).
Unsupervised learning	Unsupervised learning takes place when the learning system is supposed to detect patterns without any pre-existing labels or specifications. Training data consists of variables *(x)* with the goal of finding structural information of interest, such as clustering.
Reinforcement Learning	In a reinforced learning system, it is necessary to describe the current state of the system, a specific goal, a list of allowable actions and their environmental constrains for their outcomes. Once this is defined, the ML model experiences the process of achieving the goal by itself using the principle of trial and error to maximize a reward. These models are many times applied at games such as chess.

Table 2. 1 Overview of types of machine learning [41]

Classic Machine Learning Approaches

Even though ML techniques have seen an increase in popularity in recent times, mainly due to the increase in data available and computational power, some of the techniques have been around for a long time. K-nearest neighbours is a technique for pattern clustering and classification introduced in 1951 [42]. Here, one unknown element will be classified into an already defined category depending on what category most of the K (decided by the data analyst) nearest elements (neighbours) are classified as. It can be applied to timeseries and relies on the idea that similar patterns can be identified and classified according to their properties: for example, energy demand could be linked to weather, location, occupancy, squared meters, etc. So, given a set of observations, clusters are created based on a user-defined feature.

In [43], researchers developed a *k*-NN model for 24 hours ahead electricity load forecasting of mixed used building. The research concluded that a higher number of end-consumers meant a less accurate forecast. Wahid and Kim [44] developed a similar research doing a 24 hour ahead forecast, this time on residential buildings. Their research showed the best training to test data ratio was 60-40%. A comparison over different time horizons and different models (*k*-NN, Bayesian regression, Support Vector Machine, and ARMA) was conducted in [45]. Results showed that *k*-NN performed better for appliances and home levels with a one week forecasting horizon but was outclassed for shorter periods.

Another popular technique used are support vector machines (SVM). They are efficient for non-linear problem solving and offer accurate results even with smaller datasets [22]. When used for regression this technique can be called support vector regression (SVR). The goal is to find the best fitting function.

A challenge in successfully implementing an SVM is the calibration of their parameters, this is a difficult but decisive process [22].

In [46], a research to forecast cooling and heating energy demand with exogenous input variables is carried out using SVM. The research showed that appropriate input data selection could improve the forecast accuracy. Similar conclusions were drawn from the research carried out in [47], here a short-term forecast was carried out for a university building in Spain, comparing the results of three different methods: SVR, multilayer perceptron and multilinear regression. SVR model showed the higher accuracy and again the accuracy of the model increased with an appropriate selection of variables.

Another shallow machine learning method covered in this subsection of the thesis is decision trees (DT). DT follow the idea of a tree, from the roots to the leaves. There is an initial root node where depending on the conditions established it diverges into other nodes, that can again diverge into more nodes until it reaches a leaf (a valid classification or forecast value). This is illustrated in Figure 2.3. Many different types of DT have been developed, each with their own virtues and utilities.

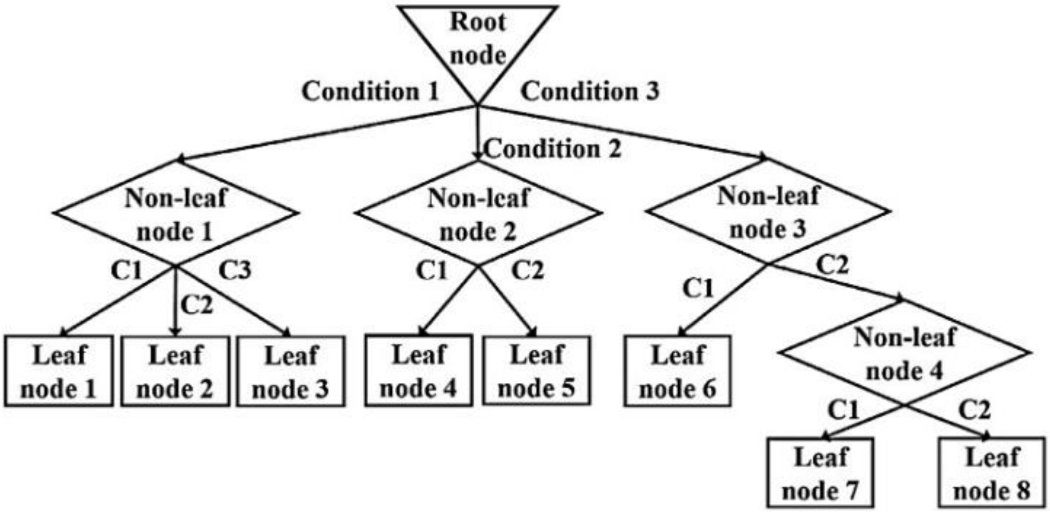

Figure 2.3 Schematic of a DT (decision tree) with input-variable based conditional separating into non-leaf nodes until final leaf nodes are reached (Source: [22]).

In [48], a comparison of stepwise regression, multi-layer-perceptron and DT models to forecast residential household electricity load is carried out. Here DT proved to be the most accurate during the summer period, although it was outperformed in the winter. A different approach was followed in [49], here data from 80 residential buildings from Japan was used to classify the building into high or low electricity consumers. Results showed a 92 % success in classification.

Another methodology used in forecasting energy consumption in buildings are ensembled models. These are still considered shallow learning. Ensemble models are data-driven algorithms designed for forecasting application. They use a specific framework focusing on the improvement of prediction performance and on the trade-off of the strengths and weaknesses of predictive algorithms [22]. Ensemble models include two main steps [50]: (1) a number of sub-models called base learners (heterogeneous models) or base models (homogenous models) are obtained, (2) comparison of their respective forecasting results is preformed, results are weighted and combined to generate and optimal output. Homogenous modelling creates a sub-samples from the original dataset which are then processed through one singe data-driven technique [22]. Heterogenous modelling on the other hand uses a number of different forecasting algorithms trained on the same dataset, the forecasting results from each model are then weighted to give a final output.

Similar to a DT, but with its own characteristics is random forest (RF). RF is considered an ensemble homogenous model. An advantage over conventional DT is that it corrects the tendency to overfit their training set. It does so by generating a group of several DTs whose results are aggregated into one [51]. They usually implement randomization on two different levels, by having each DT trained with a random subset of observations and then having each tree node divided by considering a random subset of variables [22]. In [52], a study comparing the iterative reweighted least square regression method to a RF to predict heating and cooling load is carried out. In this study, RF outperformed the other model. In a similar fashion, [53] conducts a research comparing RF, regression tree and SVR for the forecasting of electricity consumption in two institutional buildings. RF outperformed both models for both buildings.

Deep Learning Approaches

Deep learning techniques have seen their popularity increase in recent years [54], this is mainly because of: their ability to handle large amounts of data, their improved feature extraction abilities, and their improves model performances [55]. Runge and Zmeureanu [55] carried out a review of deep learning techniques used in forecasting energy use in buildings. This review showed that the most popular techniques used are: autoencoders, recurrent neural networks and deep neural networks.

M Bourdeau et al. [22] also carries out a review on methodologies used to forecast consumption in a building, including artificial neural networks (ANN). ANN are one of the most applied building energy consumption forecast methods, but there are not considered DL per se, it is dependent on the number of hidden layers that an ANN can be considered DL. Shallow architectures refer to one to three levels of non-linear operations, whereas, deep architectures contain four or more levels [55]. This subsection of the thesis intends to give an understanding of deep learning algorithms and to explore the state of the art of said methodologies.

According to [55], there have been three main ways in which DL models have been applied to building energy forecasting:

i) Increasing the number of hidden layers in a feed forward neural network / multilayer perceptron
ii) Applying some recurrent neural network (RNN) (RNN, long short-term memory (LSTM), etc.)
iii) Sequentially coupling of different types of algorithms.

The use of DL offers some potential advantages and disadvantages over more traditional methods. Firstly, in DL methods feature extraction can become automatically learnt [55], a skill that otherwise requires expertise in the field. Also, DL methods can easily incorporate large amounts of data. On the other hand, a disadvantage of DL methods is they are typically difficult to train and contain large number of hyperparameters.

There are many different DL techniques that have been applied to energy building forecasting, such as: autoencoder, recurrent neural network, convolutional neural networks (CNN) or deep belief networks. This work will focus on LSTM which is a form of a RNN. A RNN can be differentiated from a feedforward neural network (FFNN) by the presence of a feedback loop. This loop means that values calculated for time *(t)* are influenced by those calculated in *(t-1)*. This could be described as a sort of "memory". With timeseries data, RNN can learn and model the temporal behaviours present in the timeseries data and use the feedback connections in order to recall calculations from previous steps [56]. LSTM, unlike RNN, includes gates, these gates are applied to control information flow and help decrease short-term memory loss. An illustration of a RNN and a LSTM network can be seen in Figure 2. 4. Where *t* is the timestep, x_t is the input vector, h_t is the hidden layer and/or output vector, tanh and σ are activation functions, and c_t is the cell state.

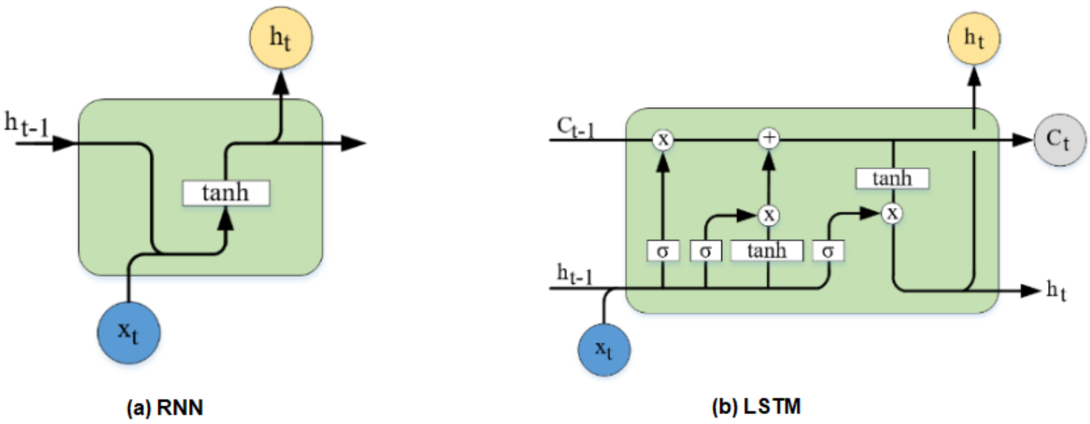

Figure 2. 4 Representation of a: (a) RNN and a (b) LSTM network (Source: [55])

The following discussed research papers refer to models used to forecast energy consumption at a district level (multiple whole buildings energy loads aggregated). In [9], a hybrid system involving LSTM and multi-layer perceptron (MLP) is used to forecast natural gas consumption. Clustering was used to partition the data into consumption profiles after which the MLP would decide on which LSTM model to handle the data forecast. This methodology was compared against benchmark models and showed improved accuracy. Furthermore, research conducted in 2020 [57] compared a variety of different algorithms in order to forecast the heating demand of a district system. Their research showed that the LSTM models were among the highest performing algorithms. Regarding commercial-residential buildings, forecast using LSTM models has also been used in [58]. In this research LSTM models were compared to other prominent data-driven models and their experiment showed the LSTM model obtained the lowest forecasting error.

Now, the following papers discuss energy forecasting at building level. In [59], twelve different shallow and deep learning algorithms are used to forecast a building cooling load, the study showed that LSTM and extreme gradient boosting were amongst the most accurate models. More research involving LSTM forecast is done in [60], where a forecast energy consumption for an educational building is presented, showing different LSTM based models outperformed SVR, deep belief networks and ARIMA models

Overall, it is safe to state that forecasting energy consumption, in different scenarios (city level, district, commercial-residential, etc.) using DL, and LSTM models is a very researched topic, and the performance of these models is proven.

2.3 Summary

The above work shows that little work has been done to study the electric energy consumption of IK even though they are very intensive energy consumer. Their energy consumption is vastly underestimated as the literature shows. Furthermore, most of the work done in this area is related to benchmarking energy consumption one way or another so IK can compare their metrics to others and have an understanding on where they stand energy efficiency wise. Unfortunately, little work has been done to study the integration of RES in IK. On the few work that was presented on this note, it showed there is room to improve RES penetration into IK consumption.

Finding suitable forecasting methods for an effective management of energy resources is of paramount importance for improving the efficiency in energy consumption and decreasing its impact on the environment [9], and because of this, the topic of energy forecasting is a widely researched one. Unfortunately, there is no work related to IK energy consumption forecasting, contrary to building energy forecast modelling. This state of the art reviews some of the very different techniques used to forecast building energy consumption and classifies the methods into three categories:

i) Statistical based forecasting or classical methods.
ii) Shallow or classical machine learning algorithms.
iii) Deep learning algorithms.

Autoregressive models, statistical regression models and Prophet are amongst the statistical methods used and discussed to forecast energy building consumption. Regarding classical ML approaches the use *k*-NN, SVR, DT and ensemble models such as RF is a widely extended practice. Finally, several DL approaches have been used to forecast energy consumption, one of which is LSTM, this method has shown very robust results in this matter.

With this information in hand, this thesis will try to fill in the existing gap in the field of IK energy consumption forecast. To do so, it will use real data collected from a restaurant in Portugal and apply the existing knowledge of forecasting in building to IK. Furthermore, this thesis will explore the potential improvement in the accuracy of the forecast by forecasting for each individual appliance and summing these forecasts as opposed to forecasting for the whole load.

Chapter 3

Methodology

This chapter presents the methodology used to turn the initial raw data into useful data via data cleaning and pre-processing. An introduction to the methods selected to forecast (Prophet, RF and LSTM) is presented. Finally, the processes the data undergoes to train and test are presented and the performance metrics selected to evaluate the models are also discussed.

3.1 Overview

With the goal to study the value of forecasting for each individual appliance and summing those forecast instead of forecasting for the virtual aggregate (this concept is defined in chapter 1.2) this chapter will present the methodology developed to obtain the final conclusions. An illustration of the overall methodology can be seen in Figure 3. 1.

The first step in this methodology is to carry out a pre-processing of the data. This includes methods to handle null values and outliers, data resolution resampling and normalization of the data. In the following step, data is further processed to create the virtual aggregate, which consists of the sum of the consumption timeseries of each individual appliance loads.

The data is also processed to turn it into a supervised learning problem by the process of windowing. Next, three distinct forecasting algorithms are evaluated against this data (both the virtual aggregate and the load of the individual appliances). To do this, 85% of the data is used to train the models while the 15% remaining is used to test and obtain metrics of their performances.

The performance of the models will be evaluated by measuring the error they present in their forecast, to do this, the root mean squared error (RMSE) metric was selected. Due to the high variance in the load some of the appliances present, mean absolute error (MAE) was also selected to obtain a fairer estimate of the precision of the models when forecasting for the appliances.

Finally, the load forecasted for the induvial appliances will be summed and compared against the forecast of the virtual aggregate, this process will be done for the three algorithms employed to forecast. The goal is to establish if there is an improvement in the performance by forecasting for the individual appliances as opposed to forecasting for the virtual aggregate.

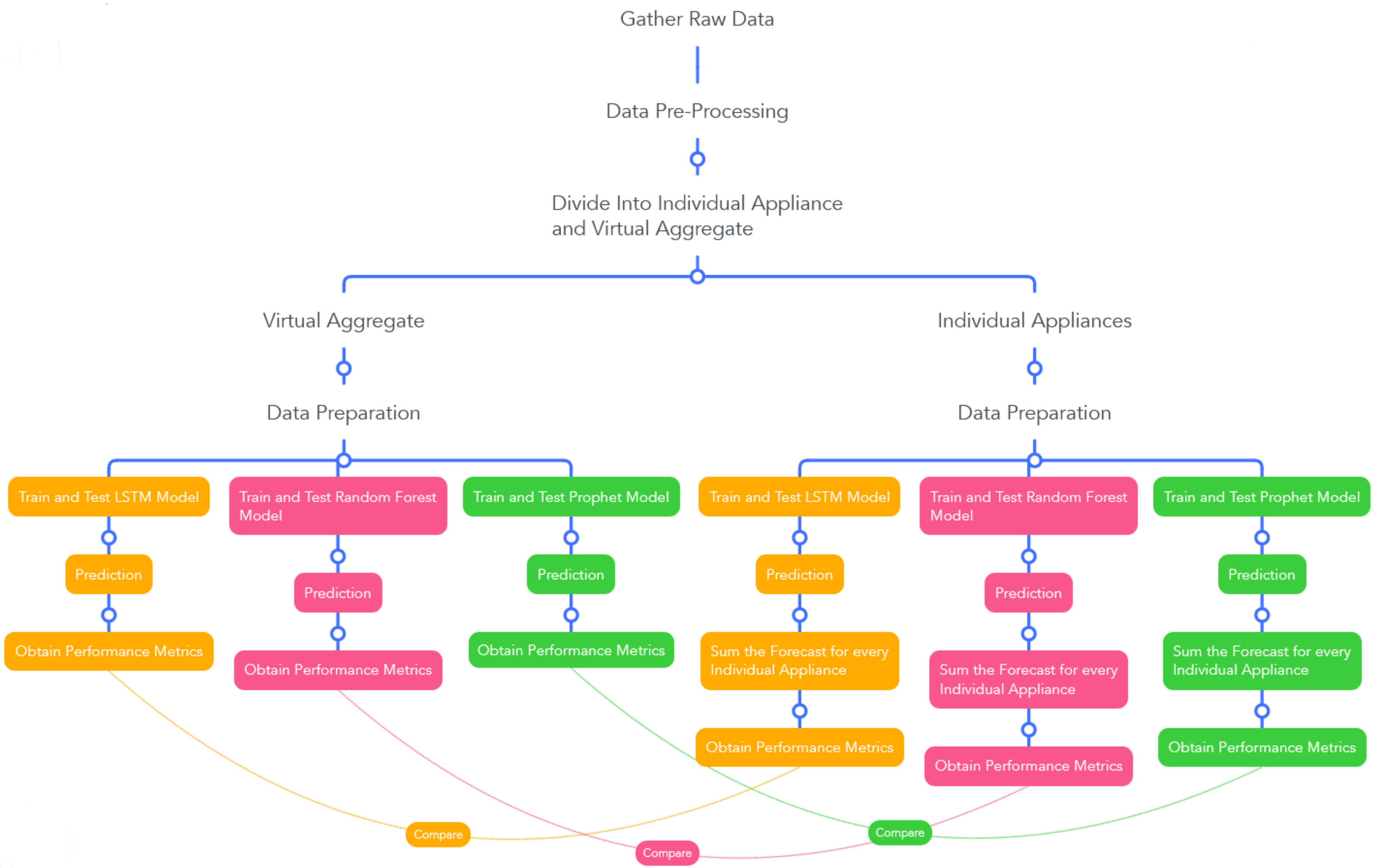

Figure 3. 1 Workflow methodology

3.2 Data Cleaning and Pre-processing

The notion of data pre-processing affecting the outcome of a ML process is widely accepted [61] and is therefore a fundamental aspect of any ML project. This is mostly since raw data is many times unstructured, and contains missing value and noise.

A crucial aspect of pre-processing is dealing with null values and outliers in the dataset. A null value is an instance where there is no value present, which is not the same as a value of zero. For instance, the power demand of an appliance at a certain timestamp could be zero watts, with this information it is known that said appliance is not working or not consuming power at that instant, but if the value is null this means that the power demand is unknown. This phenomenon can occur when the data provides from a sensor recording and the sensor has failed at some point, so is the case in this thesis.

An outlier is a value that is far from the average data and could also possibly be a reading error. There are several ways to deal with outliers and null values. In this thesis, given that it is a possibility for an appliance to not be working at a certain moment, its consumption would be equal to zero, this means that zero values are not considered outliers or sensor errors. Furthermore, appliances have a high range of values (because zero values are acceptable) making it harder to determine whether a seemingly anomalous value is indeed an outlier or not. Because of this, a visual inspection of the timeseries is carried out and any value which is clearly an error is removed. No errors in this matter were found in the thesis.

On the other hand, null values are very easy to spot. Once spotted, the null value must be replaced by a valid value that is interpretable. An efficient manner to do this in a timeseries is by either a forward or backwards fill function, these are illustrated at Figure 3. 2. In this thesis a forward fill function was used, this means that the null values at an instant *(t)* obtain their new value from the value at an instant *(t-1)*. Given the high frequency of data collection (every second for the real aggregate and every 5 seconds for the appliances) it should not be an issue to replace the null values in the manner explained.

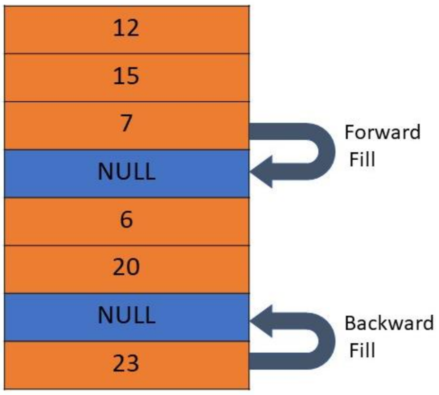

Figure 3. 2 Forward and Backwards Fill

After handling null values, the dataset is resampled to a five-minute interval. This means that the dataset now is represented by data points given each five minutes, these values are obtained by the average of all datapoints held inside that five-minute interval. Now it can be said that the data has a five-minute resolution. This eliminates possible remaining noise and makes for a smoother representation of the timeseries. A five-minute resolution smoothed the data while maintaining enough datapoint to work with.

Unfortunately, this process can create new null values. This is because the method will create a new value for every five minutes of data, ranging from the first data point to the last one, but if there is an interval of five minute where no datapoints were sampled, it will create a null value. Even though the sampling for the appliances was every five seconds, due to possible sensor errors or log in errors in the timestamps there is a period of time with no recordings (not null values). For every five minutes with no data a null value is created. This issue is dealt with before the training process. The data will go through a process described here as “windowing” and explained in section 3.4.1 of this same chapter. After this windowing process and before training, the algorithm will drop any window that includes a null value and will not be used for training the model.

After this point is when the virtual aggregate is created. This is easily done by summing the loads of all the appliances into a unique load.

At some moment it will be necessary to compare performance metrics, how well does one model forecast compare against another one, or how does the forecast of one appliance compare against another one. Doing this can be hard since not all appliances consume the same amount of power. To tackle this the data is normalized before training and testing. To normalize is to eliminate the unit of measurement. Data normalization is also a technique to improve data quality and the performance of machine learning algorithms and is a practice widely extended and researched [62]. There are several ways of normalizing data, in this thesis a max normalization approach is followed. This means that the data will undergo a process to have all the values rescaled between 0 and 1. To do this, in a dataset (x), each datapoint (x_i) is divided by the maximum value of the dataset, as can be seen in Equation 3.1. This method was selected for various reasons: easy interpretability of both the timeseries after normalizing and the performance metrics derived from the forecast of this timeseries and ease to rescale later to obtain the real values of the forecast which will be later used. An example of data normalization can be seen in Figure 3. 3, here it can be seen how the timeseries maintains the exact distribution, but the values are now rescaled between 0 and 1.

$$x_{i,normalized} = \frac{x_i}{\max(x)} \tag{3.1}$$

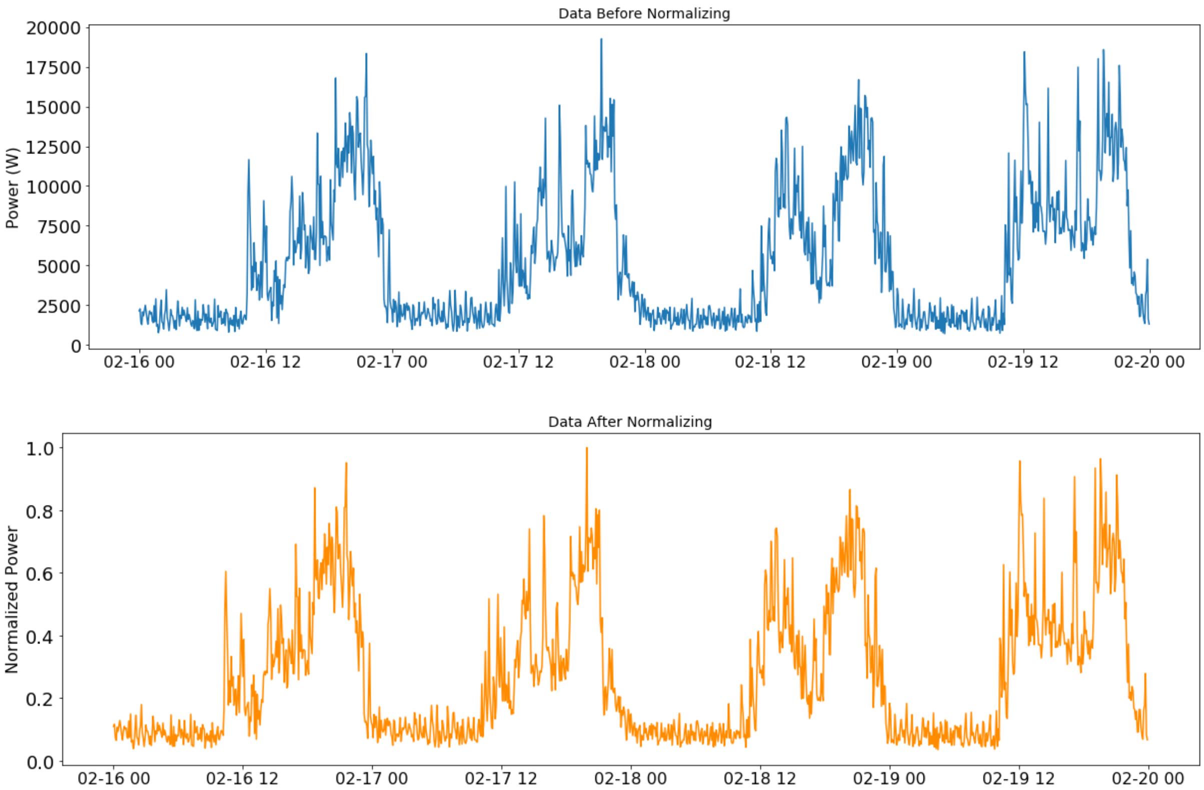

Figure 3. 3 Data Normalization Example

3.3 Load Forecasting Algorithms

This section provides a general overview of the forecasting algorithms that were evaluated in this thesis.

3.3.1 Algorithms

As it has been introduced in the state of the art, many different algorithms and approaches have been carried out to forecast the energy consumption in buildings. Only the data driven ones (or black box) will be taken into account in this thesis. These can be divided into statistical, classical ML and DL. One model will be selected from each of these to do the forecast and evaluate their performances. These are: Prophet, RF and LSTM.

The first was selected for its ease of implementation and the interpretability of its parameters. The second was selected because, as a classical ML algorithm, its ensemble nature makes it less prone to overfitting, and the amount of data used in this thesis is scarce. The latter was selected because RNN can learn and model the temporal behaviours present in the timeseries data and use the feedback connections in order to recall calculations from previous steps [56], also the inclusion of gates in the LSTM help reduce short term memory loss. Furthermore, the research has shown many good results in implementing LSTM models to forecast energy consumption. The three algorithms will now be presented.

3.3.2 Prophet

Prophet is a modular regression model which is designed to have intuitive parameters that can be adjusted without knowing the details of the underlying model [34]. It uses a decomposable timeseries model with three main components: trend, seasonality and holidays. Equation 3.2 shows how these elements are combined.

$$y(t) = g(t) + s(t) + h(t) + \epsilon(t) \tag{3.2}$$

Where $g(t)$ is the trend function, it models non periodic changes in the value of the timeseries. Here $s(t)$ represents periodic changes (such as weekly or yearly seasonality) and $h(t)$ represents the effect of holidays which occur irregularly on the schedule. The error is represented by $\epsilon(t)$.

This specification is similar to a generalized additive model (GAM). This has the advantage that it decomposes easily and accommodates new components as necessary, for instance when a new source of seasonality is identified. GAM is also very quick to fit [34].

This model approaches the forecasting problem as a curve fitting exercise, which is inherently different from timeseries models that explicitly account for the temporal dependence on structure in the data, this formulation provides a number of practical advantages [34]:

- Flexibility: Seasonality is easily accommodated with multiple periods and allows the analyst to make different assumption about trends.
- Measurements do not need to be regularly spaced and there is no need to interpolate missing values.
- Fitting is very fast, allowing the analyst to interactively explore model specifications.
- The model has easily interpretable parameters that can be changed by the analyst.

Trend

Prophet implements two models to determine the trend: a saturating growth model and a piecewise linear model.

The saturated growth model can be expressed as seen in Equation 3.3. Here C is the carrying capacity, k the growth rate and m an offset parameter. The carrying capacity represents the maximum for the system. If modelling for users of a restaurant at a given time, this carrying capacity could be the number of seating available.

The complete equation with all parameters depending on time is not shown here to maintain simplicity.

$$g(t) = \frac{C}{1+\exp(-k(t-m))} \tag{3.3}$$

Equation 3.3 is not an exact representation of the equation used by the Prophet model, this is because they have adjusted so that the carrying capacity and the growth factor are both dependant on time. Prophet incorporates changes in the growth model by explicitly defining changepoints where the growth rate is allowed to change. These change points S occur s_j, j = 1, …, S times. A vector of rate of adjustment δ is defined, where δ_j is the change that occurs. The rate at any time t is then the base rate *k*, plus all of the adjustments up to that point. This is defined as equation 3.4.

$$a_j(t) = k + a(t)^T \delta \tag{3.4}$$

When rate k is adjusted, the offset parameter *m* must also be adjusted. The correct adjustment to *m* at each changepoint is represented with the letter ψ and has its own equation to compute its value. Neither the equation for the correct adjustment or the final equation for the trend are shown here as they are not necessary to understand the ongoing process.

For forecasting problems that do not exhibit saturating growth, a piece-wise constant rate of growth provides a useful model, defined by equation 3.5.

$$g(t) = (\, k + a(t)^T \delta)t + (m + \, a(t)^T \psi) \tag{3.5}$$

Prophet incorporates an automatic changepoint feature, of course the data analyst can select the changepoints if he has knowledge of this. In a restaurant, this could be an expansion of the local or a change in chef as an example.

Seasonality

Prophet relies on the Fourier series to provide a flexible model of periodic effects. This is expressed in equation 3.6. Where P stands for the expected seasonality (seven for a weekly period, 365.25 if yearly, etc.). Fitting seasonality requires to adjust the 2N parameters, how this is done is out of the scope of this thesis.

$$s(t) = \sum_{n=1}^{N} [a_n \cos(\frac{2\pi nt}{P}) + b_n \sin(\frac{2\pi nt}{P})] \tag{3.6}$$

The final component to examine is how Prophet handles holidays and events, but this will not be discussed in this thesis as it is irrelevant to the data used here.

This methodology was implemented in this thesis using Python and importing the Prophet library[1]. The

3.3.3 Random Forest

As discussed in section 2.2.2, RF is an ensemble model which groups several DT and the final outcome takes into account the outcome of each DT in the RF. Each DT is fitted to a sub-sample sample of the dataset, by averaging the output of these DT the final outcome is obtained. These is a good method to avoid over-fitting.

RF offers various computational and statistical advantages such as [63]:

- Naturally handle both regression and classification
- Relatively fast to train and predict
- Depend on only a few tuning parameters (according to [63], there are three parameters that can be tuned to improve the accuracy of the model; *m*, the number of randomly selected predictor variables; *J,* number of trees; and *tree size.* RF only being somewhat sensitive to *m*).
- Have a built-in estimate of generalization error
- Can be used directly for high-dimensional problems
- Measures of variable importance
- Differential class weighting
- Missing value imputation
- Visualization
- Outlier Detection
- Unsupervised learning

Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest [51].

For a random vector *X* representing real data input, and a variable *Y* representing real value response, the goal is to find a prediction function *f(X)* to predict *Y*. The prediction function is determined by a loss function *L(Y,f(X))* and defined to minimize the expected value of the loss [63]. Logically, the error is a measurement of how far away *f(X)* is from the actual value *Y*. This loss function must be selected accordingly to the problem being handled, regression or classification. A typical loss function for regression is squared error (equation 3.7), while a typical loss function for classification is zero-one (equation 3.8). Of course, in this thesis, forecasting energy consumption will be a regression task.

$$L(Y, f(X)) = (Y - f(X))^2 \tag{3.7}$$

$$L(Y, f(X)) = I(Y \neq f(X)) = \begin{cases} 0 \; if \; Y = f(X) \\ 1 \; otherwise \end{cases} \tag{3.8}$$

[1] Prophept, https://facebook.github.io/prophet/docs/quick_start.html#python-api

Ensembles construct *f* in terms of a collection of so-called "base learners" $h_1(x), \ldots, h_J(x)$. These base learners are combined to give the ensemble predictor *f(x)* For regression, the final output *f(x)* comes of the average of all the base learners (equation 3.9), while for classification it is the value that is most frequently predicted by the base learners (equation 3.10). In a random forest the j^{th} base learner is a tree denoted by $h_j(X, \sigma_j)$ where *X* represents real input data and σ_j is a collection of random variables. It is important to note that the σ_j's are independent through *j =1* to *j = J* [63], making each tree unique. This creates randomness in each tree and making each tree different to the other ones ensures that not every tree will obtain the same results. This is how RF deals with overfitting.

$$f(x) = \frac{1}{J}\sum_{j=1}^{J} h_j(x) \tag{3.9}$$

$$f(x) = argmax \sum_{j=1}^{J} I(y = h_j(x)) \tag{3.10}$$

Trees used in RF are based on a binary partition of individual variables, depending on the value of a certain variable it will split to one side or another [63]. This process carries on until a terminal node or "leaf" is reached. For a continuous predictor variable, a split is determined by a split-point; it will split one way or another depending on if the value is higher or lower than the split-point. This is illustrated in Figure 3. 4.

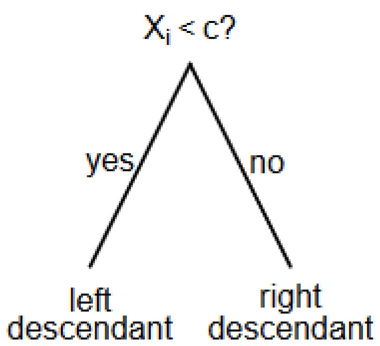

Figure 3. 4 Splitting on a continuous predictor variable X_i, using split point c [63].

The particular split a tree uses to partition a node into two descendants is chosen by considering every possible split on every predictor variable and choosing the "best" according to some criterion. In the regression context, a typical splitting criterion is the mean squared residual at the node [63].

This forecasting methodology was implemented in Python by using the "sklearn.ensemble" import of

"RandomForestRegressor"[2].

3.3.4 Long-Short Term Memory Deep Neural Network

The LSTM model is a RNN specially designed to overcome the exploding/vanishing gradient problems that typically arise when learning long-term dependencies [64]. A basic LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The latter allows the network to reset its state, this gate was not initially part of the LSTM but was later introduced [64]. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information associated with the cell.

In short, the LSTM architecture consists of a set of recurrently connected sub-networks, known as memory blocks. The idea behind the memory block is to maintain its state over time and regulate the information flow through non-linear gating units. The output of the block is recurrently connected to the input block and all the gates [64].The visual representation can be seen in previous chapter 2.2.2, Figure 2. 4. It is not on the scope of this thesis to give a deep understanding of the mathematics behind the functioning of the LSTM, however, some basics will be covered.

The input to the LSTM block will be a combination of the current input x_t and the previous output of the LSTM y_{t-1}. Each component will have a weight associated with it. Now, the input gate will combine the current input x_t, the previous output y_{t-1} and the cell value in the last iterations c_{t-1}. Again, each value has weights associated with them [64].

The forget gate determines which information should be removed from its previous cell states c_{t-1}. The activation values of the forget gate at time t depend on the current input x_t, the previous output y_{t-1} and the cell value in the last iterations c_{t-1}. The output gate is also dependent on these values; x_t, y_{t-1} and c_{t-1}.

Finally, the output block combines the current cell value c_t and the current output gate value o_t. It is important to note that the weights associated with the input, cell and output value mentioned earlier, do not have to be the same at each block or gate. These weights are adjusted by implementing a full gradient training.

This forecasting methodology was implemented in Python by using the LSTM library from "keras"[3].

[2] Random Forest Regressor, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

[3] Keras LSTM, https://keras.io/api/layers/recurrent_layers/lstm/

3.4 Performance Evaluation

The goal of this thesis is to test if forecasting consumption for the individual appliances and summing these forecasts can improve the forecast vs forecasting for the virtual aggregate. With that goal in mind, the workflow followed to compare those forecasts is the one shown in Figure 3. 1. Models for each appliance will be evaluated individually, then a model for the virtual aggregate will be created and evaluated. Finally, the outputs of the individual appliances must be de-normalized, summed, normalized again (to the maximum value of the virtual aggregate) and compared to the performance of the virtual aggregate model.

To be able to compare and draw conclusions from the forecasts it is necessary to obtain performance metrics of these. This section will discuss the preparation the data must go through, the process of training and testing, and conclude with the selection of an appropriate performance metric that allows to understand how well the models are behaving.

3.4.1 Data Preparation

Before training and testing, it is important to turn the data fit into a supervised learning problem to train the LSTM and RF models. This is done via the process of windowing. To begin with, it is important to define what a timeseries is. Picture a datapoint *y* in an instant *t*, this could be anything, in this thesis, it is power demand and is measured in Watts. The main characteristic is that *y* is dependent on *t*, which means in an instant (*t*+1) *y* can change. Timeseries can easily be plotted in an *x* and *y* plot, as were seen for example in Figure 3. 3. The issue is, the relationship between *y* and *t* isn't always straightforward, that is why we develop algorithms and models that can predict *y* values for the future.

To make these models work, it is important to turn these timeseries into something called supervised learning. Table 3. 1 and Table 3. 2 have been included here to better illustrate this concept. The first one shows a timeseries where *y* value increases by one in each *t* increment meaning *t* = *y*. The latter illustrates how these windows would look like for the mentioned timeseries and input window size of 5 and output window size of 1, the size of the windows can vary to better adjust to the dataset.

This process turns a vector of values (a timeseries) into an array consisting of a list of vectors. Where each vector has values for the input window (blue) and values for the output window (orange). The goal of this process is to feed this data to the model, where it will train to recognize patterns relating input windows to output windows as these windows slide to cover the entirety of the data. This process is used to adapt the data to train the RF and LSTM models. A characteristic of this process is that the bigger the size of these windows the smaller the amount of them there will be, this means there will be less windows to train with. The output window size will be selected according to the time horizon that it is desired to forecast. For example, to perform a forecast with one hour time horizon, using a data with a 5-minute resolution, the output window will be equal to 12 data points. For LSTM and RF, it was selected an input window of 288 and an output window of 12. This means that the model will learn to predict the next hour of consumption by looking at the previous 24 hours. The model has a one hour forecast horizon. This process is unnecessary to train the Prophet model, in this case the model learns the patterns from the training data and makes a prediction for the remaining of the data (the test data).

Table 3. 1 Timeseries *t=y*

(t)	*y*
0	0
1	1
2	2
3	3
…	…
n	n

Window number	Input Window Values					Output Window Values
1	0	1	2	3	4	5
2	1	2	3	4	5	6
3	2	3	4	5	6	7
4	3	4	5	6	7	8
5	4	5	6	7	8	9
.	.	.	.	.	.	.
96	95	96	97	98	99	100

Table 3. 2 Window Example. Input n = 5; Output n = 1

This process turns a vector of values (a timeseries) into an array consisting of a list of vectors. Where each vector has values for the input window (blue) and values for the output window (orange). The goal of this process is to feed this data to the model, where it will train to recognize patterns relating input windows to output windows as these windows slide to cover the entirety of the data. This process is used to adapt the data to train the RF and LSTM models. A characteristic of this process is that the bigger the size of these windows the smaller the amount of them there will be, this means there will be less windows to train with. The output window size will be selected according to the time horizon that it is desired to forecast. For example, to perform a forecast with one hour time horizon, using a data with a 5-minute resolution, the output window will be equal to 12 data points. For LSTM and RF, it was selected an input window of 288 and an output window of 12. This means that the model will learn to predict the next hour of consumption by looking at the previous 24 hours. The model has a one hour forecast horizon. This process is unnecessary to train the Prophet model, in this case the model learns the patterns from the training data and makes a prediction for the remaining of the data (the test data).

To train the LSTM algorithm in Python it is necessary to handle the data to turn it into a tensor. A tensor is a container of data of N dimensions, being N bigger than 3. Like a vector has 1 dimension or a matrix has 2 dimensions, a tensor might have more than 2. The LSTM algorithm will take a 3-dimensional tensor (batch size, time step, input) as input data to work with.

3.4.2 Training and Testing Procedure

An essential part of forecasting is training the model. To do this, one must divide the available data into training data and testing data. This is because if one were to use the same data to train and test the model one will obviously obtain a biased result. Training and testing data must always be separate datasets.

Training data is a set of examples the model uses to learn and fit the parameters of the model with the goal to produce a trained model that can generalize what has learnt and perform with unseen data. In this case, by perform, we refer to forecast energy consumption seeing past consumption. Now, test and validation set are concepts that are sometimes referred to as the same in the literature, however, Russel & Norvig state in their book [65] "If the test set is locked away, but you still want to measure performance on unseen data as a way of selecting a good hypothesis, then divide the available data (without the test set) into a training and validation set". It can be said that the validation set is the sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters. Finally, the test dataset is a sample of data used to provide an unbiased evaluation of a final model fit on the training dataset.

There is no one answer for all as to how much of the data should be used for training and how much for testing, as this will depend on the problem at hand, the method to be used, the amount of data available, etc. For this particular case, it was decided the training would be composed of blocks of full consecutive days. The fact that only full days are used to train ensures that the model has had the same amount of data to learn the consumption patterns at different times of the day, and it is important that these are consecutive days since the consumption at the end of one day is always followed by the beginning of the next day and there still could be relationships here, even though this time of the day is of less interest than when the restaurant is operating. Finally, it was decided that the testing data would compose roughly 15% of the available data while the 85% remaining will be used for training or validation. It was decided that a validation set would be used in the training of the LSTM model only. This is because LSTM models have more hyperparameters to train and can benefit more from the implementation of a validation set (at the cost of shortening the training set) and because of the ease to implement this using the keras LSTM model in python. This means that 80% of the data was used to train, 5% (data belonging to one day) was used to validate and the remaining 15% to test. This means that if we consider the validation set as a part of the training procedure, all three models will use the same amount of data for training and testing. This is 5179 datapoints for training (remember each datapoint is the value of the average consumption over the last five minutes) and 970 datapoints for testing.

As it was discussed in the previous chapter (3.4.1), both the LSTM and RF forest undergo the process of windowing to turn it into a supervised learning problem. This means that the model will learn the relationships between 288 datapoints (one day data) and the next twelve datapoints (one hour) throughout the whole training dataset moving the window twelve datapoints at a time. When testing, the model will read 288 datapoints and predict the next twelve datapoints according to the patterns it learned when the model was being fit. This means that to not fit the same datapoint more than once, the window of test data presented to the model to forecast will slide twelve datapoints at a time. It is essentially a one hour forecast horizon being done on the data, as new data comes in new forecasts are being made until the entirety of the test data has been forecasted. Finally, all these forecasts are chained together to create unique forecasted consumption timeseries that overlaps with the test data.

As for the Prophet model, this methodology is not necessary, after the model has been fit to the training data, it can forecast a value for a given timestamp, therefore the windowing concept does not contribute to a better forecast. Because of this, two different approaches were followed to execute the Prophet forecast. In a first approach, after fitting the model to the training data, a forecast was elaborated for the entire test set. In this methodology, the prophet method sees the same amount of training data as the other two methods but is required to make a forecast with a three-day horizon while the other methods only do a one-hour horizon forecast. The second approach takes advantage of the small time it takes the prophet model to fit, this time, making one-hour time horizon forecast (as in the other methods) and retraining the model after each forecast with the new incoming data (the real consumption for the forecasted hour).

3.4.3 Performance Metrics

The next step once the model has effectuated their forecasts is to see how well they have performed. This is done by comparing the forecast to the actual target value and obtaining a value for the error. An illustration of a forecast against the test set can be seen in Figure 3. 5. There are several ways to obtain the error value, two popular techniques are the root mean squared error (RMSE) and mean absolute error (MAE). These two can be expressed as seen in Equation 3.11 and 3.12. Where *n* is the number of samples, y is the target value and $\hat{y}$ is the forecasted value.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{3.11}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (3.12)$$

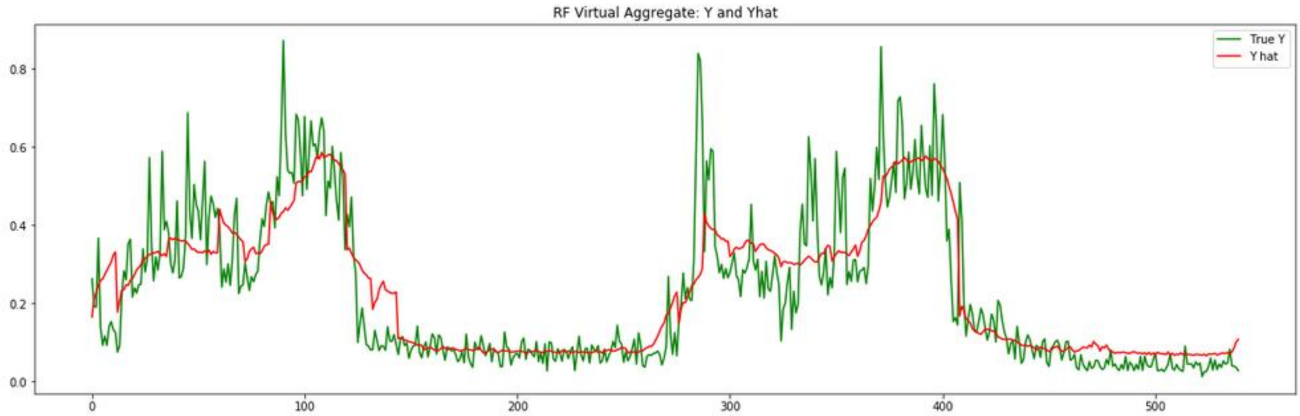

Figure 3. 5 Forecast (red) against Test Set (green)

While they have both been used to assess model performance for many years, there is no consensus on the most appropriate metric for model errors [66]. While the MAE gives the same weight to all errors, the RMSE penalizes variance as it gives errors with larger absolute values more weight than errors with smaller absolute values. When both metrics are calculated the RMSE will never be smaller than the MAE [66], the ratio of RMSE to MAE ranges from 1.63 to 2.29 [66]. One major concern to use the RMSE as a performance metric is its sensitivity to the outliers, but as discussed in previous chapters, outliers are not a big concern in the dataset being handled. Some researchers affirm that by giving a higher weighting to unfavourable conditions, the RMSE usually is better at revealing model performance differences, and another advantage is that RMSE avoids the use of absolute values which are undesirable in many mathematical calculations [66]. The underlying assumption when presenting RMSE is that errors are unbiased and follow a normal distribution [66]. Since our dataset is bound to be very diverse (individual appliances are going to have distinct consumption patterns from one another) it is hard to select a performance metric that will satisfy all needs. With all this taken into account, it was decided to measure the error in the forecast by using both RMSE and MAE to obtain more knowledge of the performance of the algorithms.

Chapter 4

Case Study & Discussion

This section will review the results obtained following the methodology previously described. First, the aggregated demand will be shown, this is the consumption for the whole restaurant. This includes consumption from sources other than the appliances, such as lightning and air ventilation.

With the thesis goals in mind: to compare the load forecast of the virtual aggregate against the sum of the forecast of individual appliances; the results for the individual appliances will first be presented, their RMSE and MAE values. Then their un-normalized forecasts will be presented, this will help identify which appliance contributes more to the error in the forecast. Then, the forecast for the virtual aggregate will be presented. The virtual aggregate will then be compared to the sum of the forecasts of the individual appliances. The models will also be compared against each other.

4.1 Exploratory Data Analysis and Pre-Processing

This subsection explores the initial data that has been used for the elaboration of the thesis, its origin, and its method of collection.

The data for this master thesis was collected by using a set of sensors in one gourmet restaurant in Portugal, during a period of four consecutive weeks. Several sensors were installed but the ones of interest for this master thesis are only the ones regarding electric energy consumption. The sensors measured both the total consumption of the IK and of several individual appliances. The process of collecting the data was not carried out in this thesis as this work had already been carried out. The data used in this thesis is real data that has to undergo pre-processing to be useful.

4.1.1 Aggregated Demand

This dataset collects the data of power demand over time for the whole restaurant, not just the appliances. This data was collected using a Fluke smart-meter connected to the main breaker box of the industrial kitchen. The data was collected at a restaurant named UVA in Funchal, the data ranges from the 2nd of February 2019 to the 15th of the same month. Only 13 days of data, but with a frequency of data collection of 1 Hertz (Hz). The magnitudes measured are active power and reactive power. A total of 1.165.190 data points compose the dataset. The dataset did not contain any null values.

Figure 4.1 illustrates the active power demand of the restaurant over time for the mentioned period. The lines being really thick is due to the high frequency of data collection and therefore density of data. This figure shows there are no outliers and no noticeable missing data entries.

From Figure 4.2, the daily consumption pattern is more discernible, consumption rises considerably at lunch and dinner time, as would be expected from a restaurant.

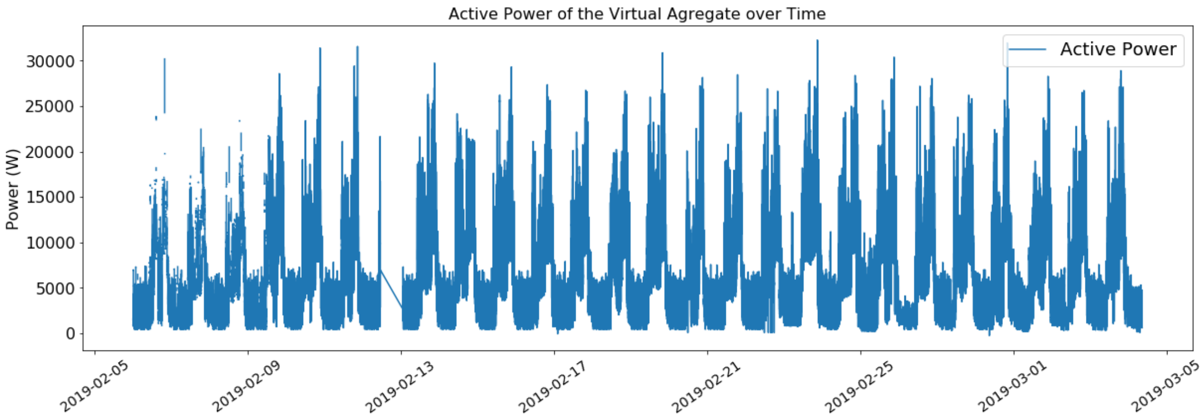

Figure 4.1 Aggregated Load

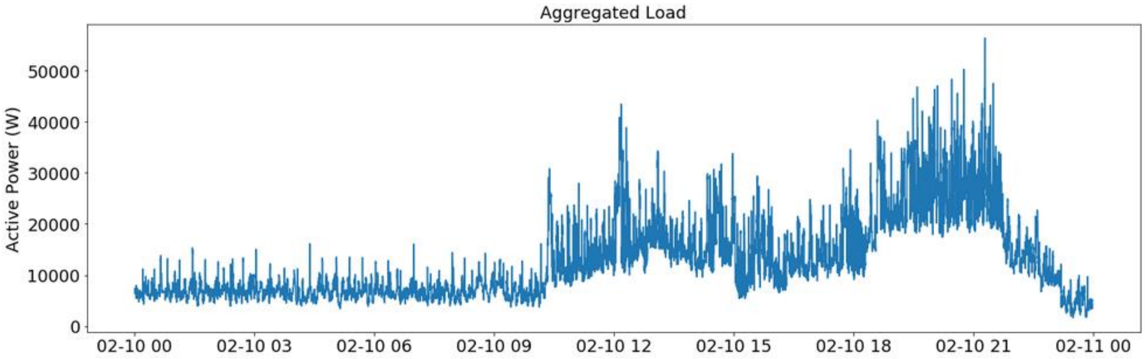

Figure 4.2 Aggregated Load demand for one day

4.1.2 Individual Appliance Demand

The demand of individual appliances was collected using the Egauge circuit-level smart-meter. A total of 18 appliances were individually monitored for up to 30 days at UVA. For each appliance active power and reactive power (in each line if it is a triphasic appliance) are measured. For the power calculations, a medium filter was applied to reduce the inevitable noise due to the measurement process.

The data collected in this dataset ranges from the sixth of February of 2019 at 00:00 (2019-02-06 00:00:00) to the fourth of March at 08:47 (2019-03-04 08:47:17). The data is collected for a period of five seconds and the value represents the average over those five seconds.

Table 4. 1 Appliances monitored at UVA restaurant

Appliance Name	Installation Type
Glasswasher	Single-phase
Blast Chiller	Single Phase
Freezer	Single Phase
Drink refrigerator	Single Phase
Fish refrigerator	Single Phase
Garde Manager 10	Single Phase
Garde Manager 13	Single Phase
Ice Machine	Single Phase
Infrared Shelf	Single Phase
Meat Refrigerator	Single Phase
Mise en Place 14	Single Phase
Vegetable refrigerator	Single Phase
Dual Fryer	Tri-Phase
Convection Oven 01	Tri-Phase
Convection oven 02	Tri-Phase
Dishwasher	Tri-Phase
Salamander 01	Tri-Phase
Salamander 02	Tri-Phase

A few figures will now be shown to illustrate the behaviour of some of the appliances, not all will be displayed as just a few are enough to exemplify their heterogenous and sometimes erratic behaviour.

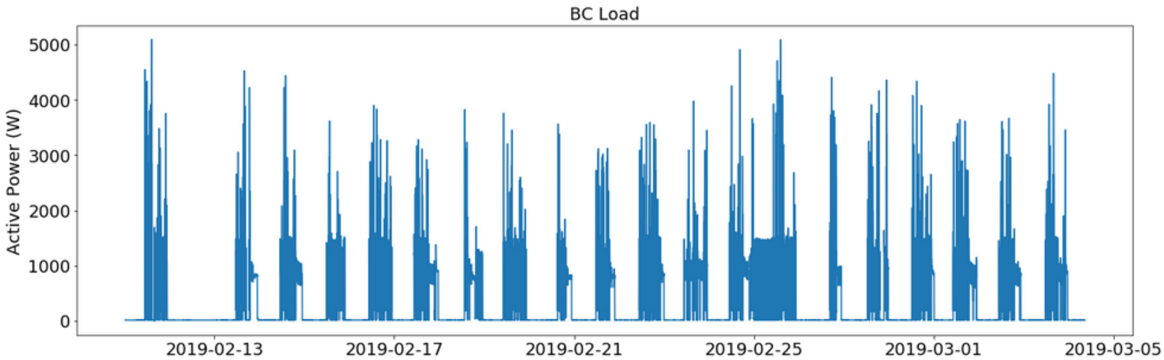

Figure 4. 3 Load of the blast chiller

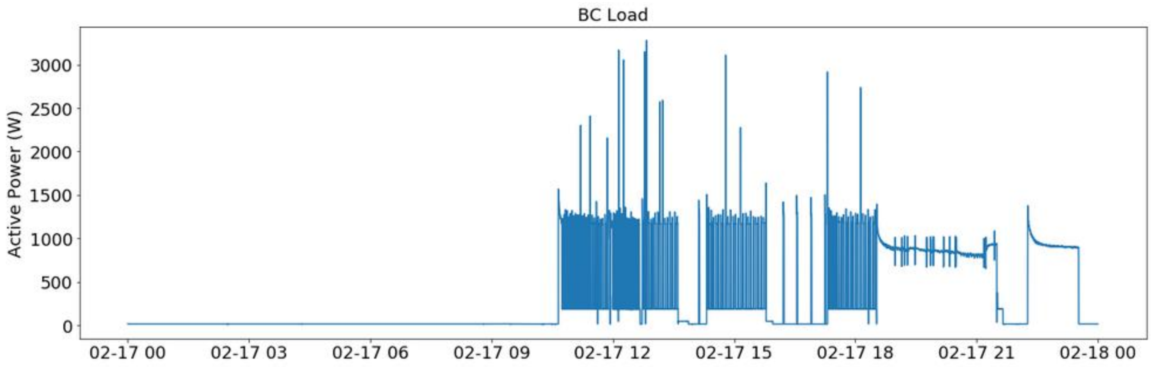

Figure 4. 4 Blast chiller power demand on the 17th of February

The blast chiller shows an erratic behaviour in the power demand. There are visible peaks in consumption followed by zero values, there does not seem to be clear a pattern to the spacing between the peaks. This is however expected of an appliance which must be turned on demand of the necessities. Figure 4. 4 shows the power demand of the blast chiller over one day, as expected its consumption is directly related to lunch and dinner time, being zero out of these hours (contrary to the freezer for example). The spikes mentioned before are more clearly seen here, and although they might seem like outliers they were not considered as such in this thesis as their values are recurrent throughout the whole dataset, implying it is not a measurement error.

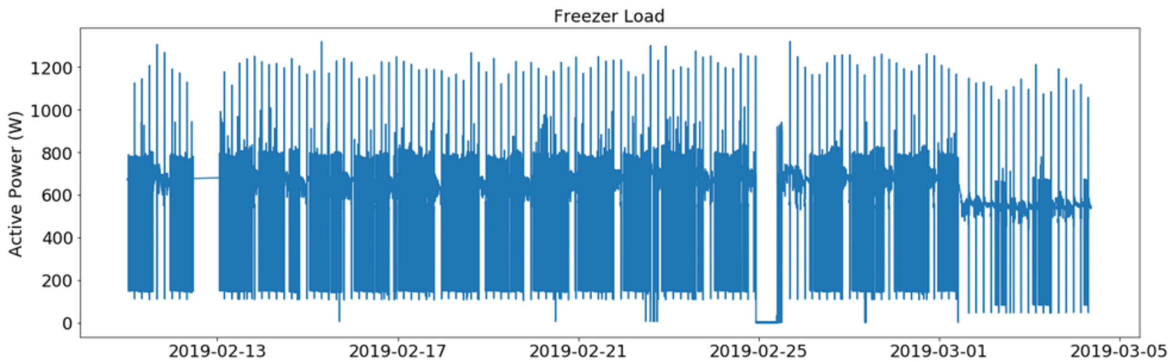

Figure 4. 5 Freezer load demand

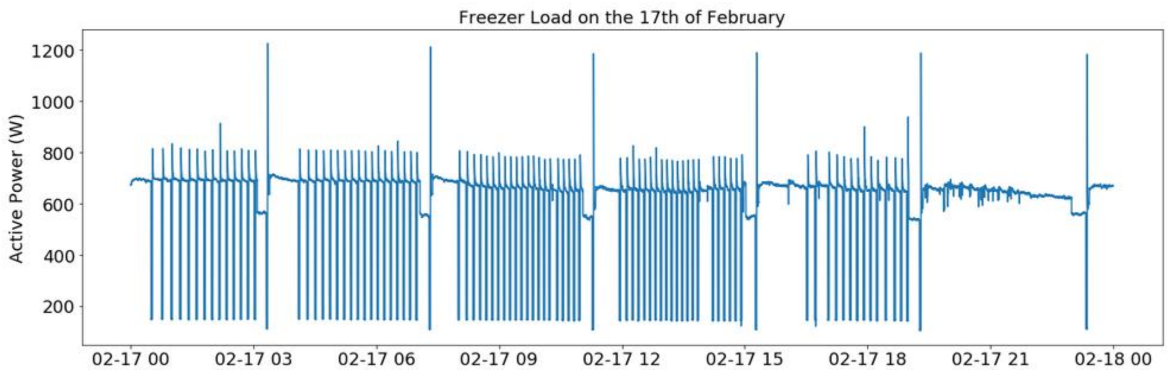

Figure 4. 6 Freezer load on the 17th of February

On the other hand, the freezer shows a more homogenous behaviour. A power demand varying from

200 to 800 and spikes that go up to 1200 (presumably because of the increased power demand after the door has been left opened). It is noticeable two areas where the consumption is missing. This issue will be addressed during the pre-processing of the data. Figure 4. 6 better illustrates how the consumption of the freezer is more constant throughout the day, even though it ranges from 200 to 800 watts it is even throughout the whole day.

As expected from a restaurant, the power demand of the appliances can follow one of two trends, either it is always on (such is the case for the freezer) or they have a higher demand at lunch and dinner time.

4.1.3 Virtual Aggregate

In order to understand how much of the total demand of the restaurant (i.e., aggregated), can be explained by the individual appliances, the notion of virtual aggregate was introduced. In this context, virtual aggregate corresponds to the sum of the consumption of each of the individual appliances and is given by equation 4.1.

$$VA_X(t) = \sum_{a=1}^{A} X_a(t) \tag{4.1}$$

Where $X_a(t)$ corresponds to the measured quantity (e.g., active power) in appliance a at time instant t. For illustration purposes, the virtual aggregate for the duration of the deployment is depicted in Figure 4. 7. As it can be observed, there are some gaps in the data, right before the 13th of February and some anomalies in the form of scattered points before the 9th. In Figure 4. 8 a one-day load has been isolated to better exemplify the load of the restaurant and the trend, as expected this load follows the same pattern as the aggregated demand (higher consumption at lunch and dinner, and valleys at night and morning). The data here still shows a high variance, due to the many appliances being studied and their different behaviour. Figure 4. 9 shows the load represented with a five-minute resolution, this smoothens the dataset and makes the patterns more obvious. This is the resolution that will be used to train the models.

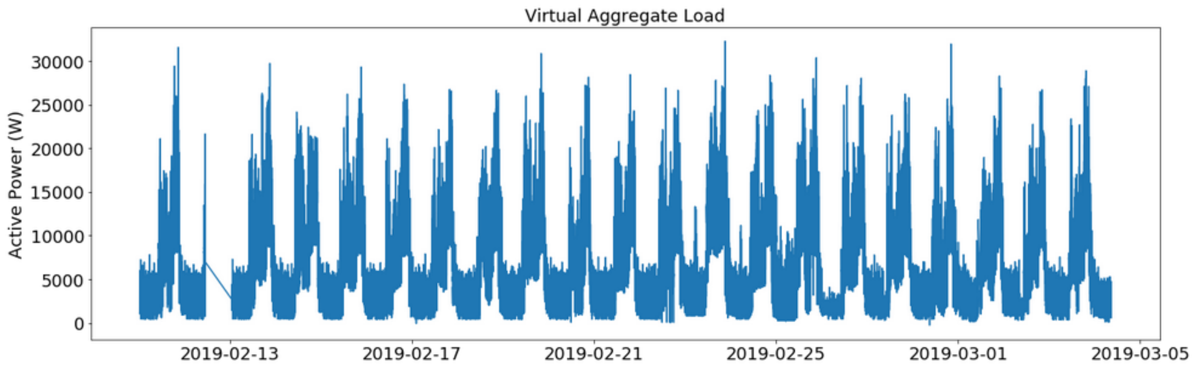

Figure 4. 7 Virtual aggregate load

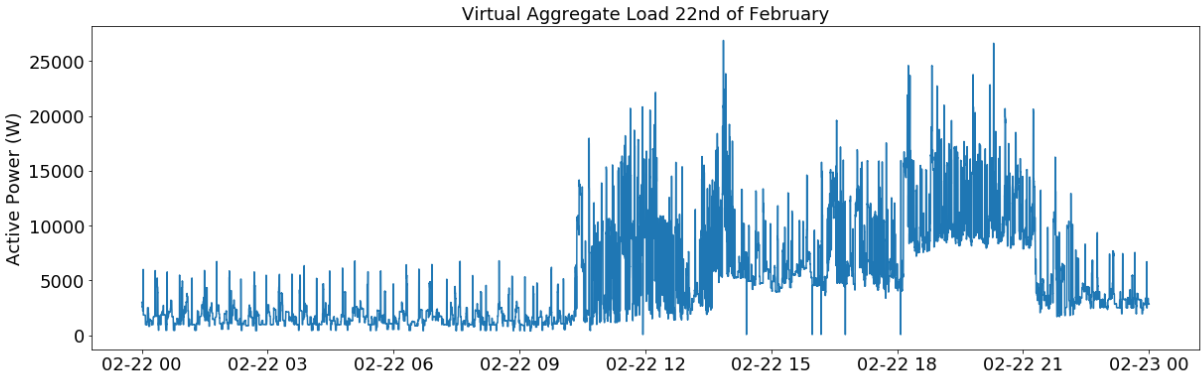

Figure 4. 8 Virtual aggregate load 22nd of February

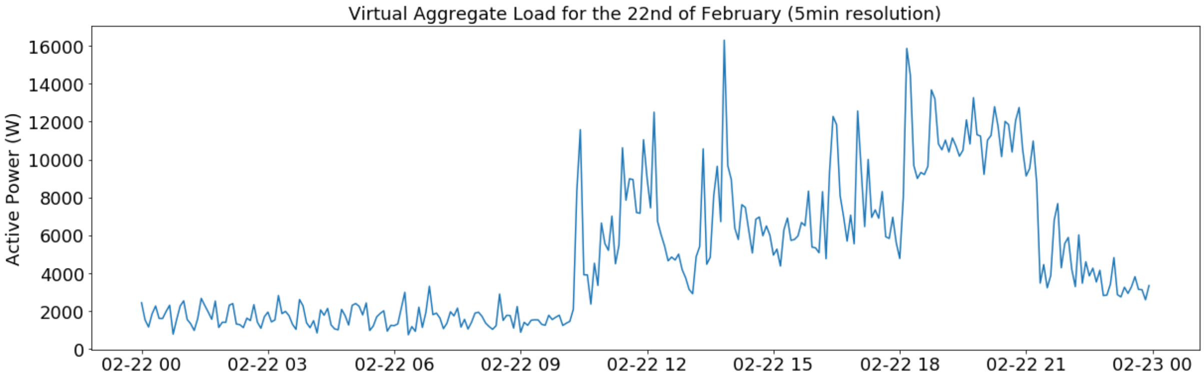

Figure 4. 9 Virtual aggregate load 22nd of February (5min resolutions)

Now that both the real and the virtual aggregate have been presented, Table 4. 2 shows a brief summary of each appliance, the virtual aggregate and the real aggregate before any data pre-processing has been done to them.

Table 4. 2 Summary of the datasets

Appliance	Start Date	End Date	Max. Power	Avg. Power	Nº Sample	Nº Null Samples
Blast Chiller	2019-02-11	2019-03-04	5331	214,6	429162	8056
Freezer	2019-02-11	2019-03-04	1319	576,5	429162	7936
Drink Refrigerator	2019-02-11	2019-03-04	623	144,7	429162	17
Fish Refrigerator	2019-02-11	2019-03-04	3709	95,8	429162	107
Garde Manager 10	2019-02-11	2019-03-04	1278	180,0	429162	8232
Garde Manager 13	2019-02-11	2019-03-04	418	70,5	429162	8075
Glass Washer	2019-02-11	2019-03-04	1885	43,2	429162	4
Ice Machine	2019-02-11	2019-03-04	421	143,4	429162	7433
Infrared	2019-02-11	2019-03-04	1133	210,5	429162	147
Meat Refrigerator	2019-02-11	2019-03-04	1274	131,7	429162	54
Mise en Palace	2019-02-11	2019-03-04	854	239,5	429162	8073
Vegetable Refrigerator	2019-02-11	2019-03-04	897	99,4	429162	146
Dual Fryer	2019-02-11	2019-03-04	5222	219,3	429162	2619
Convection Oven 1	2019-02-11	2019-03-04	8797	386,7	429162	7044
Convection Oven 2	2019-02-11	2019-03-04	8337	382,2	429162	7344
Dish Washer	2019-02-11	2019-03-04	4456	480,3	429162	1949
Salamander 1	2019-02-11	2019-03-04	3148	943,9	429162	5995
Salamander 2	2019-02-11	2019-03-04	3182	526,4	429162	6819
Virtual Aggregate	2019-02-11	2019-0-04	32261	5088,7	429162	8238
Real Aggregate	2019-02-02	2019-02-15	58110	11398	1165190	0

By comparing the virtual aggregate and the real aggregate rows it can be seen that the percentage of energy explained due to the appliances is considerably low. Despite 18 appliances being monitored, it was only possible to explain roughly 40% of the kitchen demand. According to the original project report, this occurs because the main breaker box that was monitored includes many other loads besides the kitchen appliances. For example, all the lighting of the dining room and washrooms are also connected to this breaker box.

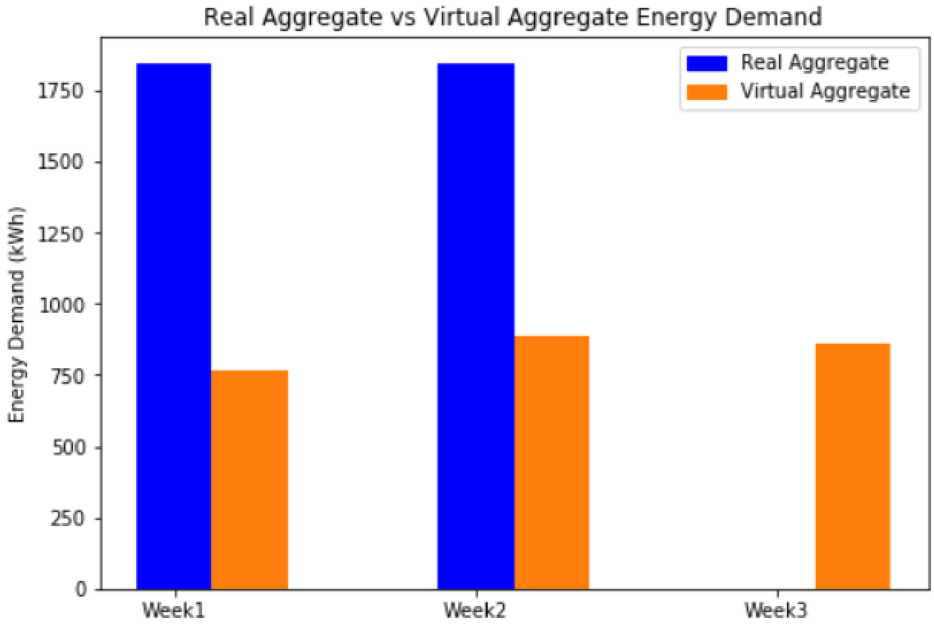

Figure 4. 10 Real aggregate vs virtual aggregate electricity demand

Bearing in mind the goal of this thesis, the amount of electricity consumption that remains unrelated to the appliance's electricity consumption, and the smaller dataset for the real aggregate, from now on only the virtual aggregate and the appliances will be used to train and test the models.

4.2 Forecasting Results

4.2.1 Virtual Aggregate

The virtual aggregate, as expressed in Figure 4. 11 is the sum of the load of all the individual appliances. As mentioned in the introduction to this chapter, the forecast for the virtual aggregate has been done using three different methods: Prophet (with two different time horizons), RF and LSTM.

Figure 4. 11 shows the virtual aggregate divided into training (blue) and test set (green) as used by the Prophet model, the forecast made by the model can be seen in red, over the green test set it is trying to forecast. The missing values at the beginning are due to the change of data resolution from five seconds

to five minutes. Figure 4. 12 is a zoom in on the test and forecast, showing both the forecast done with the three days horizon model (blue) and the one-hour horizon model (red). It is hard to tell from these plots which is performing better, a table with the performance metrics will be introduced.

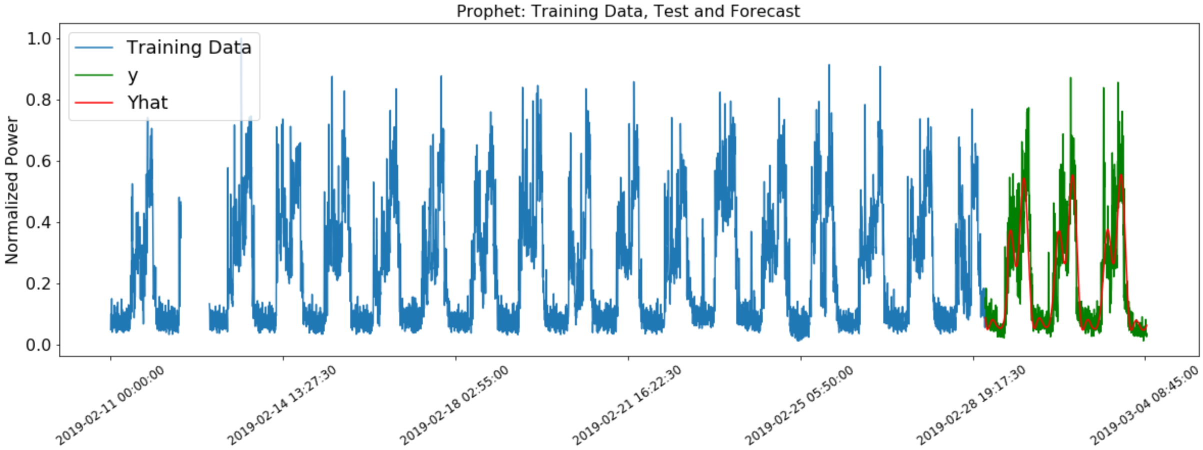

Figure 4. 11 Virtual Aggregate, Prophet training data, Y and Yhat

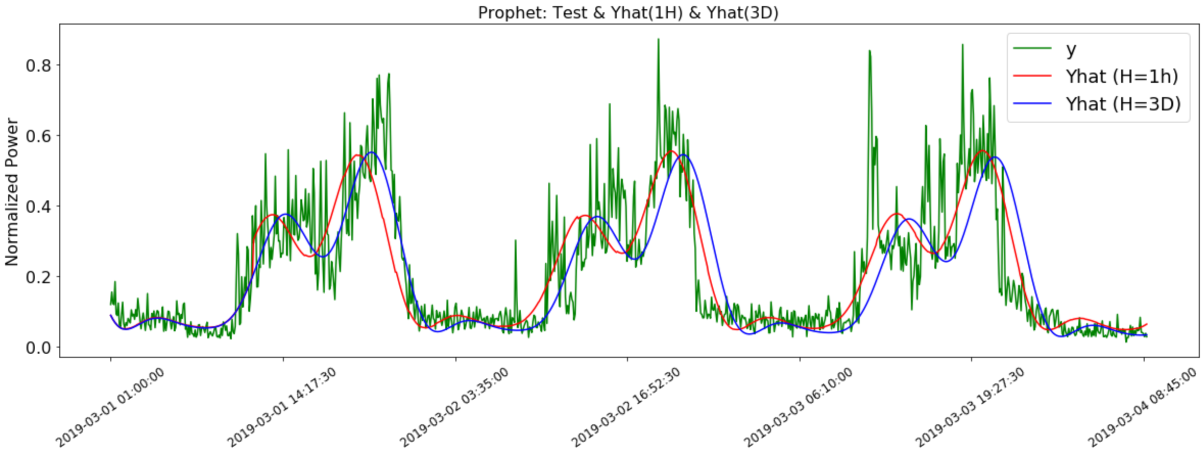

Figure 4. 12 Prophet: Test & Yhat(1H) & Yhat(3D)

Figure 4. 13 shows the training (blue) and test (green) data used to train and test the RF model. Furthermore, the forecast (red) for the test data is shown overlapping the test data. Figure 4. 13 is a zoom-in on the test and forecast that makes the comparison easier. Again, it is hard to tell how well the forecast is doing, but it is clear that the model is following the consumption pattern.

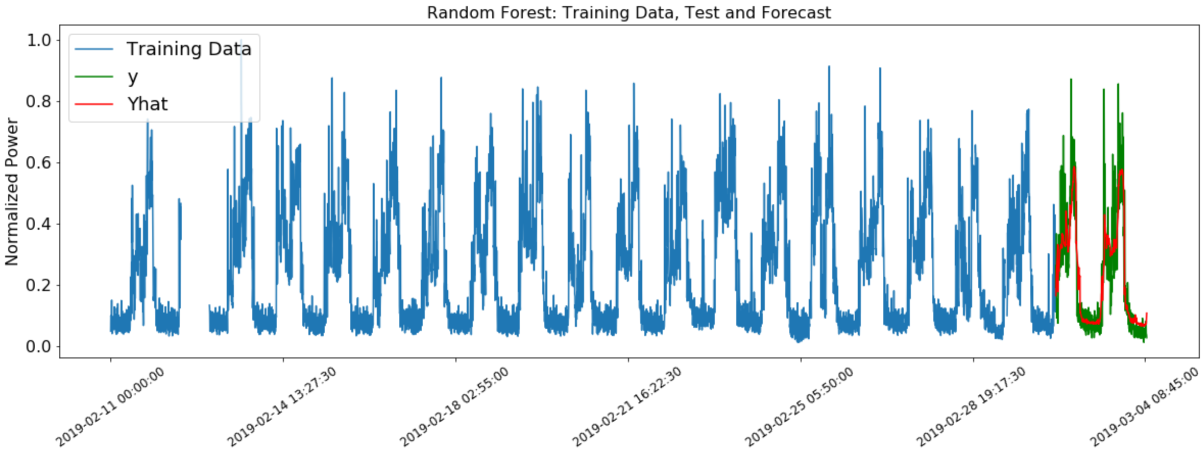

Figure 4. 13 Virtual Aggregate, Random Forest Training, test and forecast

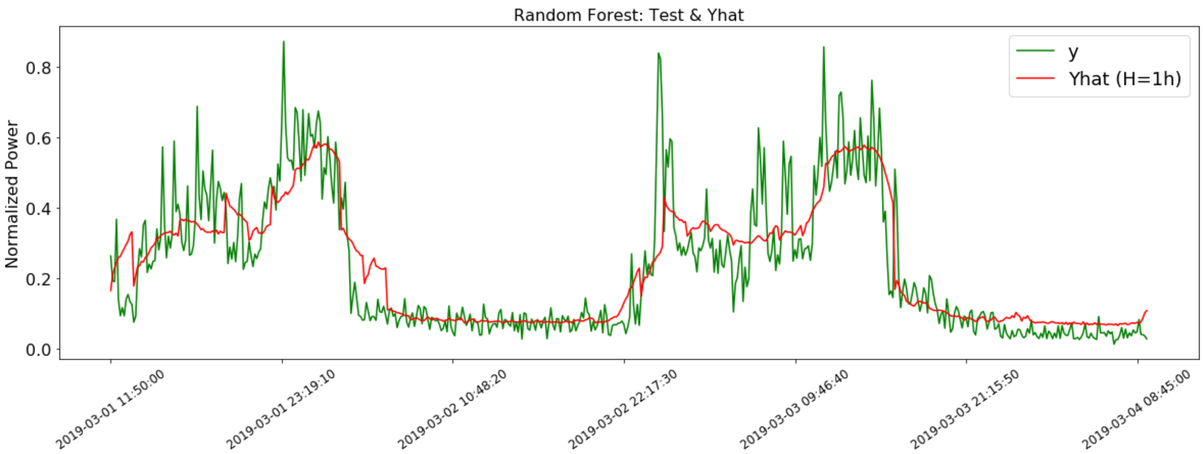

Figure 4. 14 Random Forest: Test and Yhat

Figure 4. 15 shows the training, validation and test data used by the LSTM model to train and test on the virtual aggregate. The training data is shorter than for the other models because one day worth of data is used for validation. Figure 4. 16 is a zoom-in on the test and forecast, for easier appreciation.

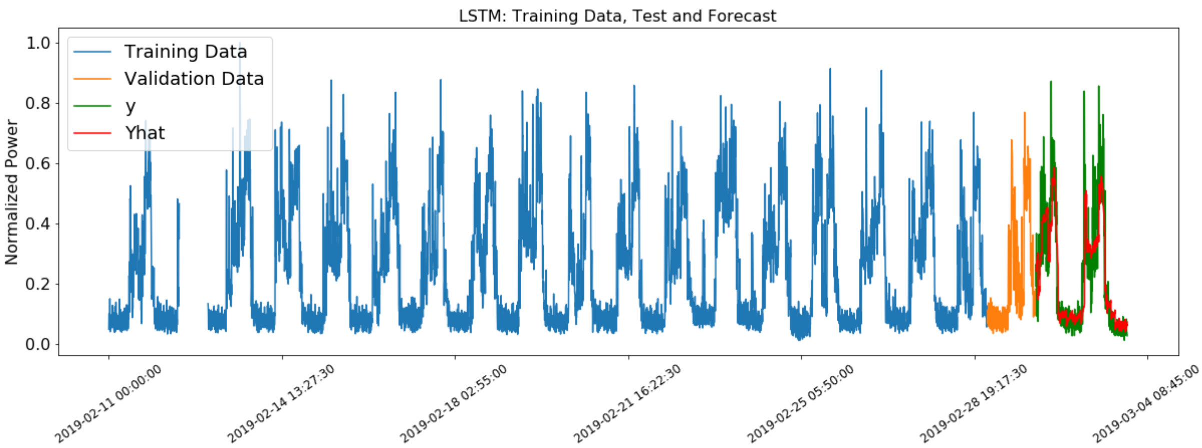

Figure 4. 15 LSTM Virtual Aggregate: Training, Validation, Y and Yhat

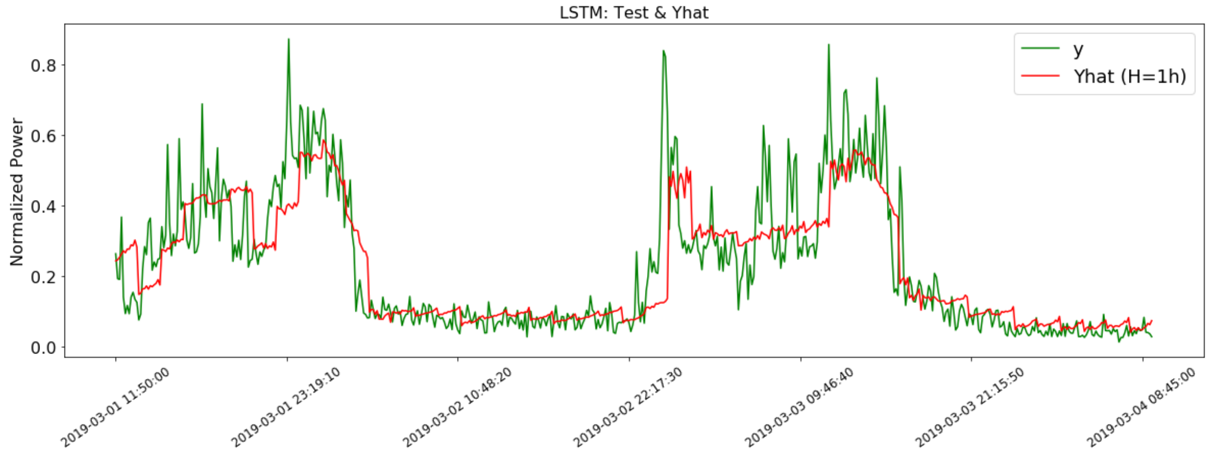

Figure 4. 16 LSTM Virtual Aggregate: True Y vs Yhat

	LSTM	**RF**	**Prophet (Three days horizon)**	**Prophet (One hour horizon)**
Virtual Aggregate RMSE	0.11	0.10	0.12	0.10
Virtual Aggregate MAE	0.066	0.066	0.073	0.068

Table 4. 3 Virtual Aggregate normalized RMSE comparison for different algorithms

From Figure 4. 12, Figure 4. 14 and Figure 4. 16 one can see the forecast of the model against the target value, but it is not easy to establish which model performed better. For this reason, Table 4. 3 is included, it shows the RMSE and MAE values of the error for each prediction algorithm. One can see here RF was the model that performed the best.

4.2.2 Individual Appliances

Table 4. 4 shows the RMSE value for each appliance and each algorithm. Red indicates the algorithm that performed worst and green the one that did best. Every algorithm managed to be the best and the worst for at least one appliance, although the margins are quite small in most of the cases. However accurate this table is, Figure 4. 17 provides this information faster and by taking a fast look one can easily figure out which appliances posed the most problems in their electricity forecast.

	Prophet(H=3D)	Prophet (H=1h)	RF	LSTM
Blast Chiller	0.20	0.17	0.15	0.17
Freezer	0.28	0.12	0.010	0.10
Drink Refrigerator	0.32	0.33	0.27	0.29
Fish Refrigerator	0.21	0.21	0.23	0.22
Garde Manager 10	0.18	0.18	0.16	0.17
Garde Manager 13	0.19	0.19	0.18	0.19
Glass Washer	0.06	0.04	0.0002	0.005
Ice Machine	0.31	0.31	0.29	0.31
Infrared	0.31	0.26	0.24	0.25
Meat Refrigerator	0.21	0.21	0.22	0.22
Mise en Palace	0.34	0.33	0.26	0.28
Vegetable Refrigerator	0.18	0.18	0.17	0.17
Dual Fryer	0.08	0.08	0.08	0.08
Convection Oven 1	0.13	0.13	0.14	0.14
Convection Oven 2	0.13	0.13	0.15	0.13
Dish Washer	0.20	0.19	0.20	0.21
Salamander 1	0.34	0.25	0.25	0.22
Salamander 2	0.24	0.21	0.15	0.17

Table 4. 4 Individual Appliances normalized RMSE comparison for different algorithms

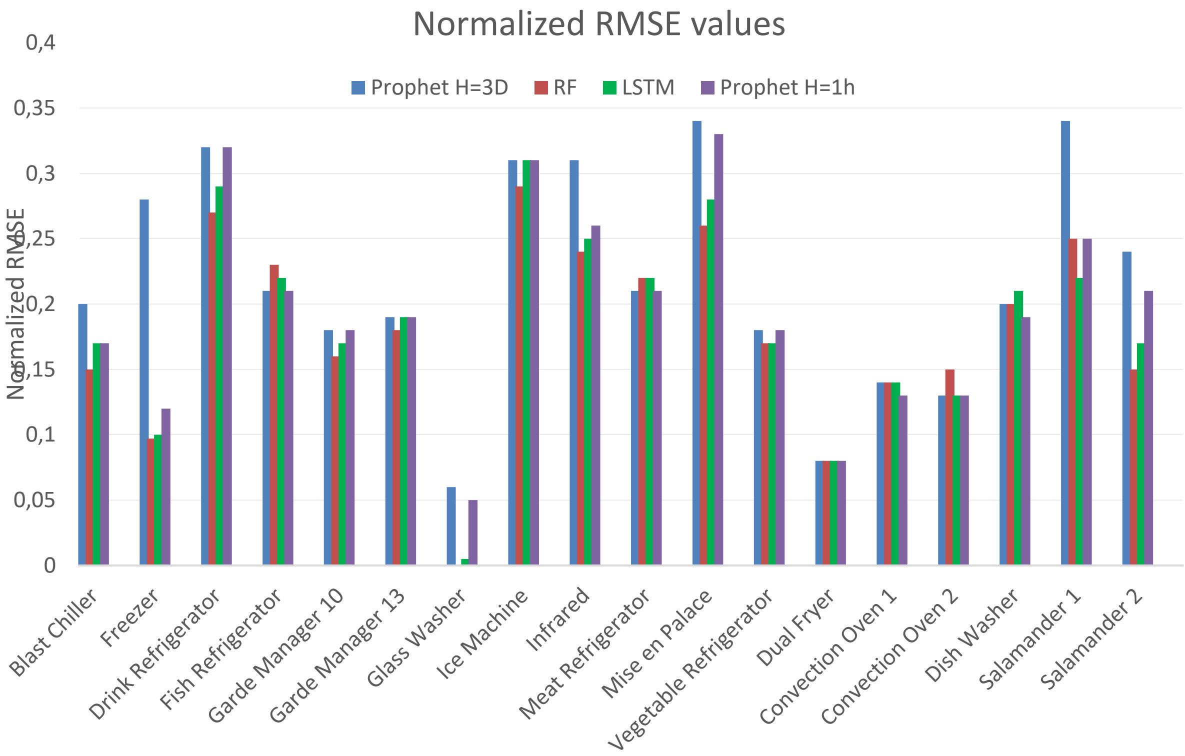

Figure 4. 17 Normalized RMSE values

Extraordinary results seem to have been achieved forecasting the consumption of the glass washer, however, this is not the case. By looking at Figure 4. 18 it can be seen that the appliance had not been working for quite a time now, and therefore the forecast is simply all zero values. RF and LSTM captured this anomaly with success, while Prophet still had some issues as it can be seen its RMSE value for the glass washes is higher than for the other two algorithms.

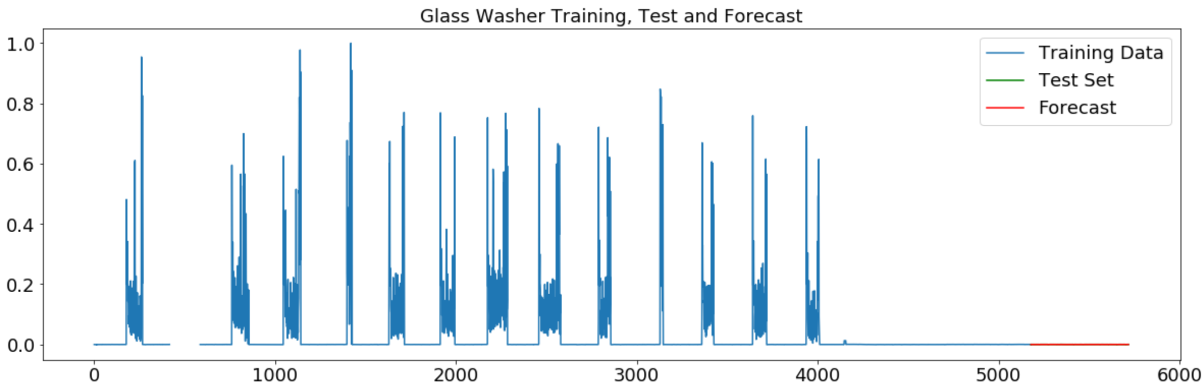

Figure 4. 18 Glass washer training, test, and forecast

While Figure 4. 17 gives a good understanding of which appliances were best forecasted and which were worst, it is still not clear which appliances contribute more to the total error of the prediction. This is because the values are seen in Figure 4. 17 are normalized.

4.2.3 Virtual Aggregate vs Sum of Individual Loads Forecasts

Prophet

This section of the thesis will now compare the results obtained by comparing the sum of the forecast of the appliance against the forecast done over the virtual aggregate. This will be done for the four methodologies employed.

First, the Prophet's results are presented. Figure 4. 19 shows the real load consumption (Y) and both the forecast for the virtual aggregate (red) and the sum of the forecast for the individual appliances (green). Surprising results were achieved here, as it was not expected that both forecasts, even though not identical, followed such a close resemblance. Figure 4. 20 shows the same graph except for the horizon forecast here is one-hour and the model is retrained with new entering data (every hour) benefiting from its low time to fit the models. Again, both forecasts follow almost the same exact pattern.

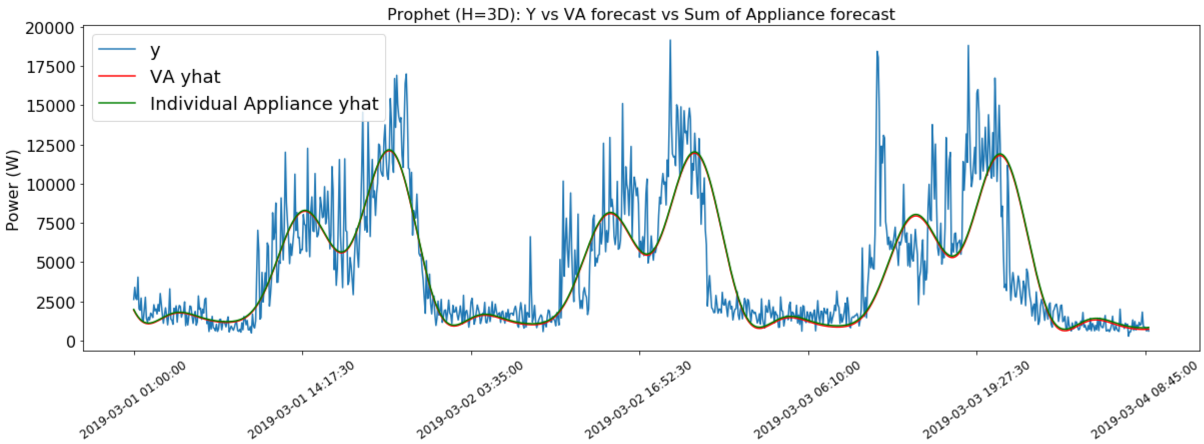

Figure 4. 19 Prophet (H=3D). VA forecast vs Sum of appliance forecast

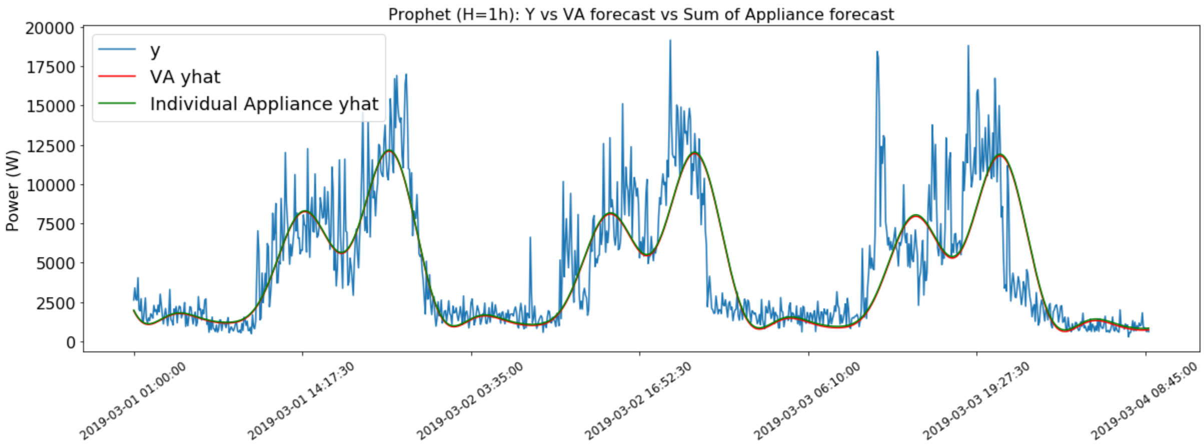

Figure 4. 20 Prophet (H=1h) VA forecast vs sum of appliance forecast

From both these figures, it is difficult to figure out which one is performing better, therefore, a comparison of both forecast was shown previously on Figure 4. 12, here both forecasts (Prophet forecast for two different horizons) are plotted against the load consumption. Only the virtual aggregate forecast is included since in this case virtual aggregate and individual appliance forecast are almost identical. Table 4. 5 shows the performance metrics for both methodologies. As was expected, the results obtained with a shorter horizon and more training data outperformed the three-day horizon forecast. The discussion of the results will now focus on the best performing Prophet Model.

Table 4. 5 Prophet performance metrics

	RMSE (W)	MAE (W)
H=3D Virtual Aggregate	2537	1605
H=3D Individual Appliance	2530	1596
H=1h Virtual Aggregate	2290	1499
H=1h Individual Appliance	2287	1507

Both the results for the virtual aggregate and the sum of the forecast of the individual appliance obtain similar results, but not identical. Stands out, that individual appliance forecast outperforms virtual aggregate when measured with RMSE but not with MAE. This means that the virtual aggregate forecast made an overall more accurate forecast, but also had more big errors. Figure 4. 21 shows a stacked area plot of the forecast for the individual appliances.

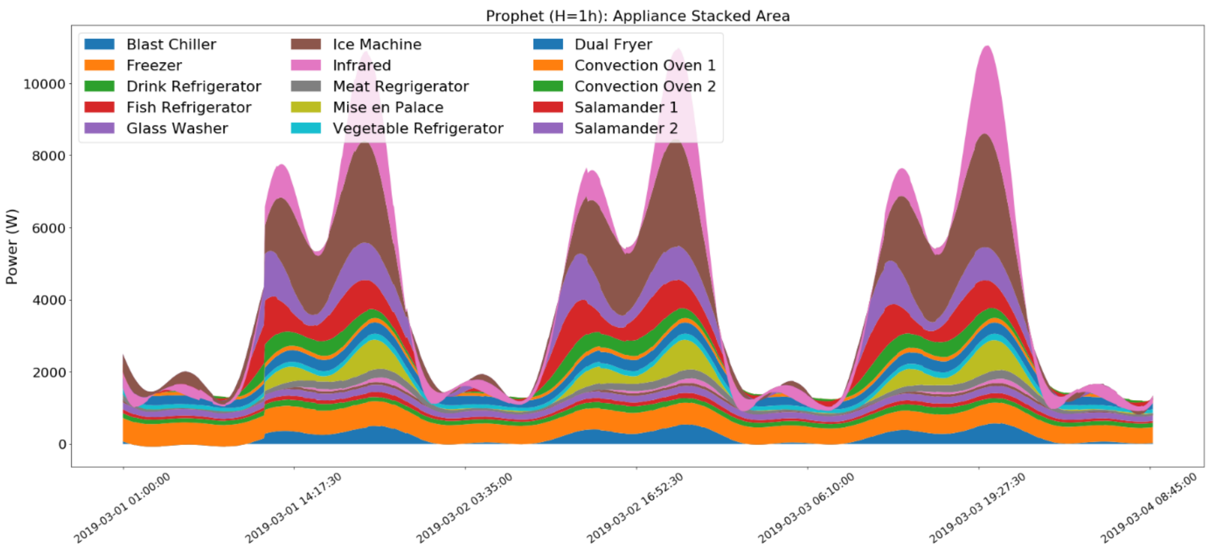

Figure 4. 21 Stacked area plot. Consumption of individual appliances forecasted by Prophet

It is particularly surprising the almost identical results were obtained for the forecast of the virtual aggregate and the sum of the individual appliance when forecasted with Prophet. It is not in the scope of this project to go into detail of the mathematics behind the functioning of the model but seeing these results one can only assume the Prophet model works in a way the sum of the parts is equal to the whole.

Random Forest

The same approach is followed for RF and LSTM. RF results are shown now. Figure 4. 22 shows the

RF forecast for the virtual aggregate vs the sum of individual appliances. Again, it is hard to discern from this figure which method performed better, both curves seem to fit the pattern, and some slight overshooting might be seen for the individual appliances. Table 4. 6 shows the performance metrics for both methods. The results show that a better forecast was done for the virtual aggregate. This may be due to the unpredictability in the behaviour of the appliances. Also, the forecast is clearly better for the virtual aggregate at the stages where the restaurant is not operating since this pattern is very predictable. Figure 4. 23 shows the forecasts for the appliances made by the RF algorithm stacked, it can be seen here how this method is better at following the spikes in consumption of the appliances in contrast to the forecast made by the Prophet model, which follows a smoother pattern.

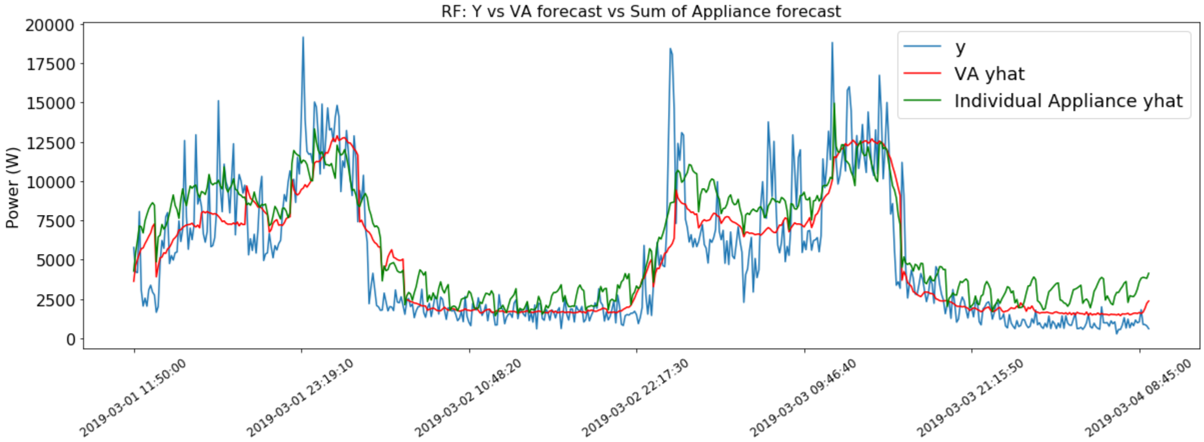

Figure 4. 22 RF virtual aggregate forecast vs sum of individual appliance forecast

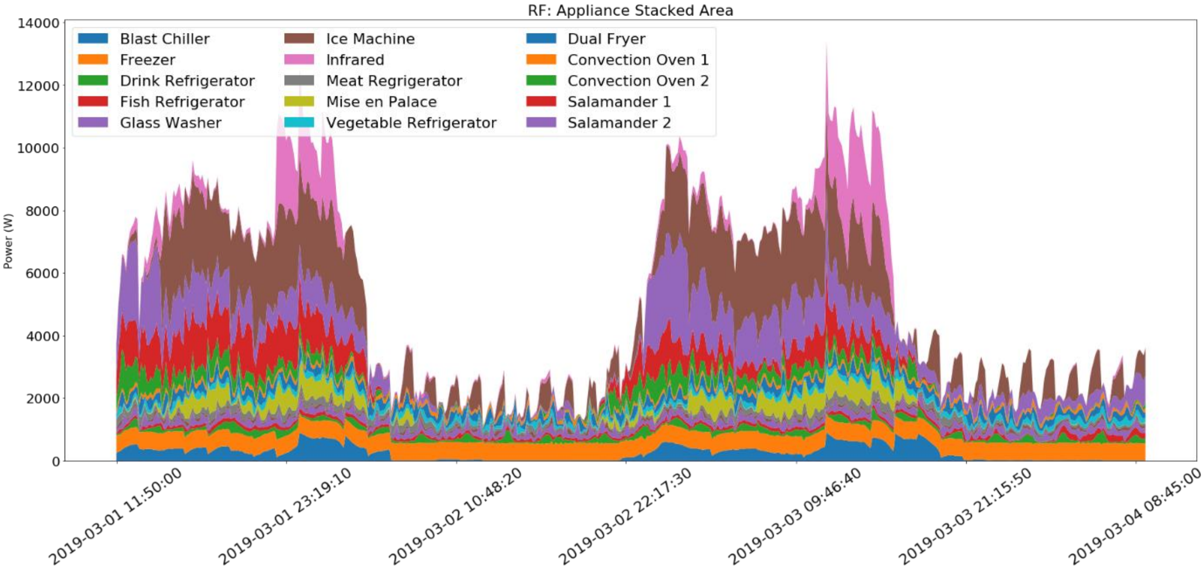

Figure 4. 23 Stacked area plot for the RF appliance forecast

Even though overall forecast is better for RF than for Prophet, stands out the spikes in consumption (saw shaped) predicted at night after the restaurant has closed. While these values in the forecast are a combination of the sum of the forecast of every appliance, most of this error comes from appliances with a “binary behaviour” that the model is incapable of following accurately. This is best exemplified in the forecast of the salamander (a specialized kitchen appliance that one will not commonly find outside of restaurants, it is used to grill, brown and toast faster than a common broiler) seen in Figure 4. 24. Here it can be clearly seen the model is forecasting two different patterns of consumption at the appropriate time (when the appliance is on or off), but those patterns are not correct. While it should be a saturated value or zero it is creating this saw pattern that adds a lot of error when the restaurant is not operating. A possibility for this occurring is the large windows employed for training, where the model is always basing its forecast on the previous 24 hours (and therefore always seeing energy consumption). A possibility to fix this issue (other than adjusting the window sizes) is to establish this problem as a classification problem rather than a regression one. One could force the model to classify the consumption into working or not working and assign values to those classifications (maximum value or zero). This would eliminate the issue where the model is accurately forecasting when an appliance is not working, but not being able to forecast its power consumption.

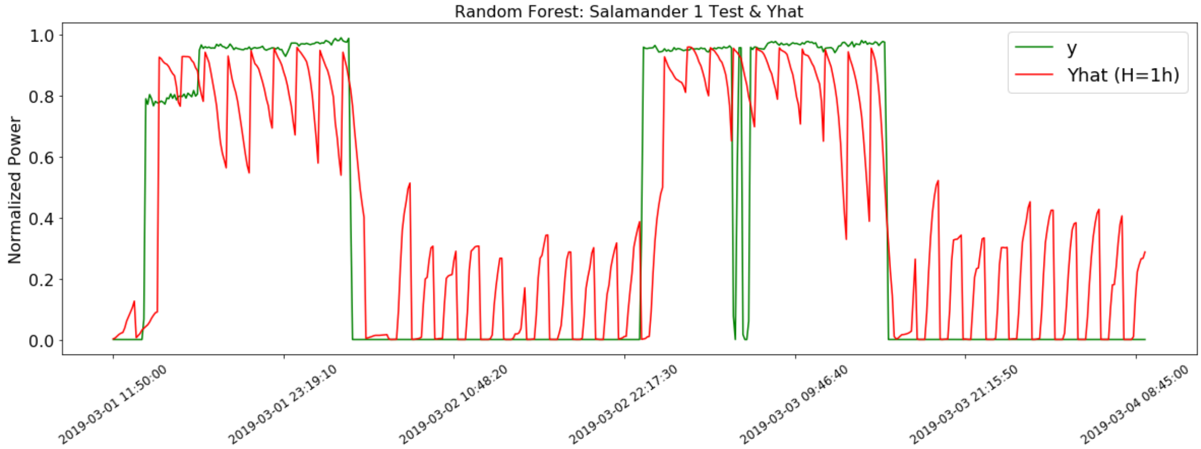

Figure 4. 24 RF Salamander test and forecast

Table 4. 6 RF performance metrics (Virtual aggregate vs sum of individual appliance forecast)

	RMSE	MAE
RF: Virtual Aggregate	2230	1507
RF: Individual Appliance	2520	1998

Table 4. 6 shows the performance metrics for the virtual aggregate forecast and the sum of the individual appliance forecast. Again, as for the Prophet algorithm, the forecast for the virtual aggregate gave a better performance.

LSTM

Finally, the results for LSTM are presented in the same way. Figure 4. 25 shows the curve of the real consumption against the sum of the forecasts of the individual appliances. As it occurs for the other methods, it is hard to estimate from this figure which method is performing better. But the curves do have a similarity with the RF curves. The spikes are more visible here and in RF than in the Prophet forecast. Also, the virtual aggregate seems to perform better in the hours that the restaurant is not operating. Nonetheless, the same saw pattern can be seen in the forecast of the individual appliances, but less accentuated. This pattern can be seen in Figure 4. 26, where the same saw shape as in the forecast of the RF for the salamander can be seen for the forecast of the LSTM, however, the forecast here is closer to the actual value and the spikes are less sharp.

The performance metrics are shown in Table 4. 7, again, the best performance came from the forecast of the virtual aggregate. One possible explanation for this is the unpredictable behaviour of some of the appliances and the little amount of available data since LSTM is a method that thrives with large datasets.

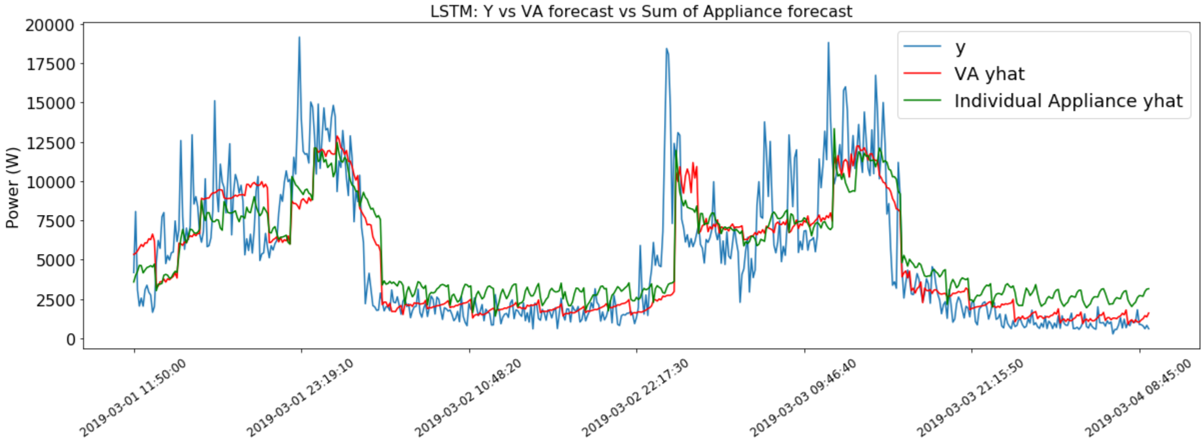

Figure 4. 25 LSTM: Virtual aggregate forecast vs sum of appliance forecast

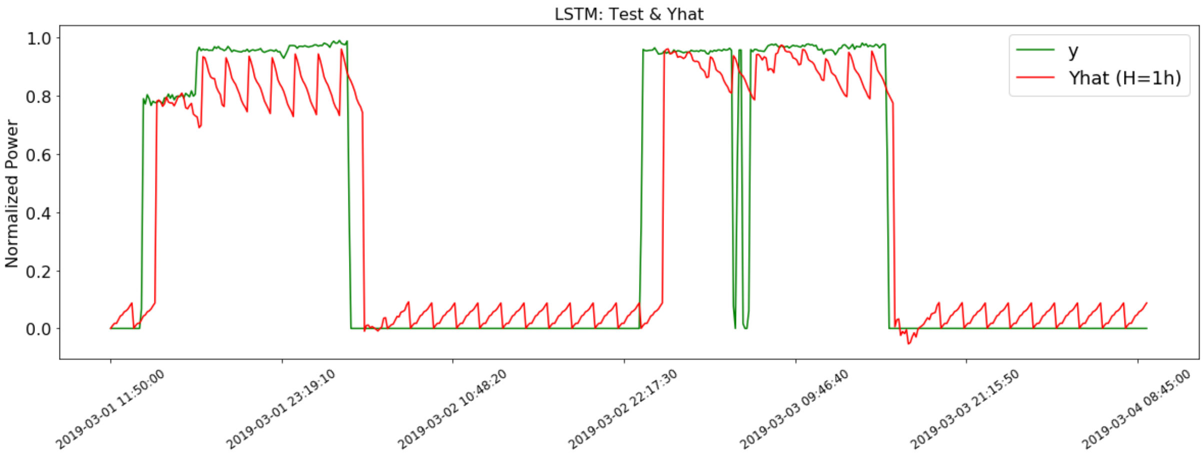

Figure 4. 26 LSTM Salamander 1: Test & Yhat

Table 4. 7 LSTM performance metrics: virtual aggregate vs sum of individual forecast

	RMSE	**MAE**
LSTM: Virtual Aggregate	2461	1564
LSTM: Individual Appliances	2630	1961

Finally, Table 4. 8 shows a summary of all the performance metrics for every algorithm and both methods. The error of the forecast for all the algorithms was smaller when forecasting for the virtual aggregate rather than summing the forecast of the individual appliances.

Table 4. 8 Performance metrics for all algorithms in both methods

	RMSE	MAE
H=1h Virtual Aggregate	2290	1499
H=1h Individual Appliance	2287	1507
RF: Virtual Aggregate	2230	1507
RF: Individual Appliance	2520	1998
LSTM: Virtual Aggregate	2461	1564
LSTM: Individual Appliances	2630	1961

There could be several reasons as to why the forecast performed better for the real load as opposed to the sum of the appliances, and ways to improve the performance.

Firstly, the amount of data available for this thesis was not very big (roughly one month data available). This affects especially the LSTM model which thrives on large datasets. While this issue affects at both the forecast of the virtual aggregate and the individual appliances the load of the IK is much more stable over time than the load most of the appliances. This load always follows the same pattern: peak values at lunch and dinner time followed by a valley after the next lunch time. While the consumption of many of the appliances are irregular in their behaviour (as they are tied to the specific situations that might arise in an IK).

This leads us to the second issue; appliances can sometimes behave like a binary system (on or off) as it was stated above. This can be seen in the forecast of the salamander, which is one of the appliances where the forecast had a higher error. This is perfectly visible in Figure 4. 27, where the forecast of the three algorithms are shown. This is an appliance with an almost binary behaviour that the algorithms fail to predict. All three models seem to predict when this appliance is on or off but do not correctly model the value. A possible solution to this would be, to first identify which appliances have this sort of binary behaviour and then treat the forecast as a classification problem and not a regression one. Other combinations of window sizes could also improve the performance of the forecast, and this possibility should be studied. Furthermore, one must take into account when looking at these results that there was no specific hyperparameter tuning done to the models, the forecast of the appliances should be improved when this technique is applied.

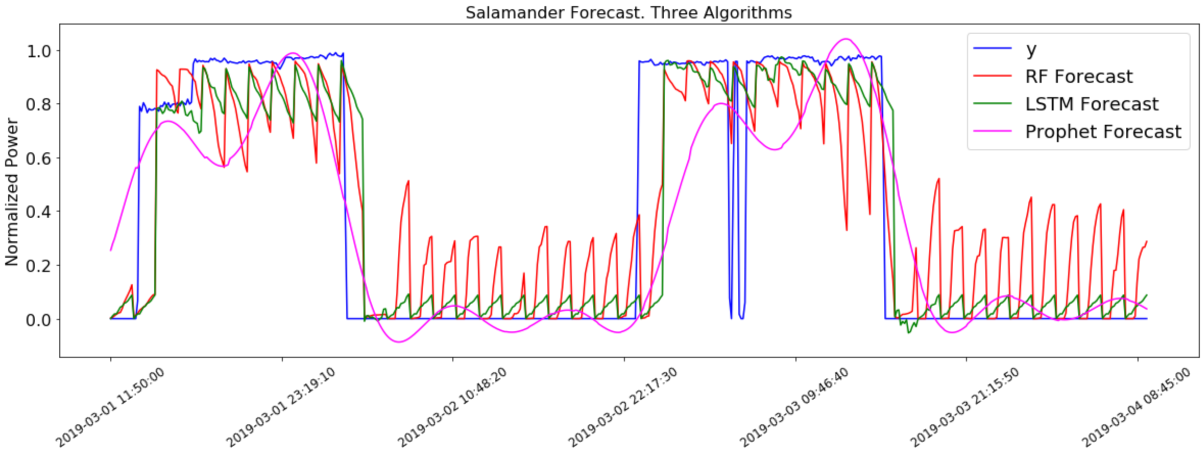

Figure 4. 27 Salamander Forecast: Three algorithms

Chapter 5

Conclusions

This chapter will present a brief overview of the whole work done. Followed by some insight into the main contributions of this thesis and the advantages of the proposed methodology. Next, limitations of the work and future improvements are presented. Finally, the challenges faced during this work and final thoughts on the best way to proceed when implementing this methodology.

5.1 Summary and Implications

This work starts by presenting the current energy and environmental crisis. Energy consumption keeps increasing as does the cost of energy. The effect of burning fossil fuels as an energy source are well known to the scientific community, and integration of RES in the energy mix and energy efficiency is of upmost importance to help against the energy and environmental crisis. With this in mind, this work shifts its focus towards the energy consumption in IKs.

More recent research shows IKs are very energy intensive consumers, and their consumption is much higher than initially estimated. This, paired with the fact that there is very little research done on IKs energy consumption outside of benchmarking suggests there is a topic to explore in forecasting the energy consumption of IKs. This work intends to study a new methodology to forecast the energy consumption in IKs by forecasting for each individual appliance as opposed to forecasting for the virtual aggregate, and this is the main contribution of this work. This was tested out using three different algorithms: Prophet, RF and LSTM.

The research carried out in this thesis has also been submitted to review in the scientific journal "Sustainable Energy, Grids and Networks"[4]

5.1.1 Prophet, Random Forest, and LSTM

All three algorithms used to implement this methodology present their own merits. Prophets' performance is peculiar, it is the easiest algorithm to implement, and its performance is almost identical when forecasting the sum of individual appliances or the virtual aggregate. This makes it so that making the comparison of forecasts of the individual appliances using the three different methods Prophet outperforms the other two, measured both in RMSE (2287 (W) against 2520 (W) in RF) and MAE (1507 (W) against 1998 (W) in RF). However, comparing the accuracy of the forecast of the virtual aggregate is a tighter call. It presents the smallest MAE (1499 against 1507 in RF) but the second best RMSE (2290 against 2230 in RF).

RF shines when forecasting for the appliances. It has the best accuracy for 10 out of 17 of the appliances, more than the other algorithms combined (Prophet is the worst in 10 out of 17). This makes it the best option when one would want to forecast a concrete set of appliances. Also, Its performance on the virtual aggregate was one of the best.

LSTM was one of the weakest performing algorithms. It has the largest RMSE for both the virtual aggregate and the sum of individual appliances. It does however score better than RF when measured with MAE for the sum of individual appliances. This is mainly because it does outperform RF when forecasting some of the appliances with binary behaviour as can be seen in Figure 4. 27 of the previous chapter. When forecasting for the individual appliance its performance was average, worst for three

[4] Sustainable Energy, Grids and Networks; https://www.sciencedirect.com/journal/sustainable-energy-grids-and-networks

appliances and best for two. The low accuracy of this model can surely be related to the low amount of available data, on which LSTM thrives.
With the current results, Prophet and RF are the more viable options. Due to its good results and ease of implementation, Prophet seems the better choice and is certainly a good one. But RF has the best performance forecasting on the virtual aggregate and for single appliances. Having the best forecast accuracy for individual appliances can have some benefits that will be expressed in the following section. With a larger dataset, it is possible that LSTM outperformed the other algorithms. Another possibility this methodology allows us to select for each appliance the algorithm that best performs. This might even improve the accuracy of the forecast of the individual appliances over the forecast of the virtual aggregate. This could be done in an automatic way if an algorithm was developed to do so, to select the optimal algorithm for each appliance based on historical data.

5.1.2 Virtual Aggregate vs Individual Appliances

From the results obtained it is not possible to conclude that the forecast was improved by forecasting for the individual appliances. All three algorithms performed better forecasting for the virtual aggregate. However, this does not mean there are no advantages to this new method of forecasting. To forecast for individual appliances allows the user to manually incorporate new appliances to the forecast (as a restaurant is a changing business) and only need to train a new model for the new appliance, while the data of the already existing appliances remains relevant. If the forecast were done for the virtual aggregate, the inclusion of a new appliance could potentially disrupt the new consumption pattern. This could make the older data and models more harm than good in future forecasts. This advantage is maintained when removing an appliance from the restaurant (for maintenance reasons for example), in this case, one can simply remove the forecast for the said appliance from the mix.

Another advantage of this kind of forecasting is that one can create forecasts for different consuming sectors of the restaurant. This means that one can agglomerate the consumption of the freezers, or the ovens for example and study consumption in this manner.

Finally, this methodology allows for the forecast of each appliance to be fine-tuned. Each appliance has its own characteristics and distinctive consumption pattern. This means each appliance can benefit from being forecasted with a different algorithm and even using different window sizes to train the models.

5.2 Limitations and future work

Certain limitations were met during the elaboration of this work. Predominantly the lack of data. The available data was collected in a Portuguese restaurant for roughly three weeks. This posed some problems at the moment of splitting the data into the training set, validation set and testing set. A compromise had to be made and the percentage of data destined to test was smaller than usually recommended in the literature, but this allowed the models to train with at least two weeks’ worth of data. The lack of data certainly hindered the performance of the LSTM model, which thrives on large datasets. To make this data viable lots of cleaning and resampling had to be done.

However, not everything is negative from the lack of data scenario. This is a more accurate representation of the amount of data available by most business. Collecting, and processing data are processes that come with an economic cost, and most businesses do not have the large datasets and computing power that large tech companies work with. This means the methods and results obtained in this work could be more easily transferred to a real-world scenario where the amount of data is scarce.

Nevertheless, this work could be expanded and most surely benefit by working on a **larger dataset** of IK consumption, in this case, the forecast accuracy should improve for both the virtual aggregate and the individual appliances. There were available two extra datasets from two different restaurants, both with a similar amount of data.

Another way to improve the accuracy of the forecast that this work did not cover is **hyperparameter tuning**. As discussed in the previous chapter, this work could also be improved and expanded by treating the forecast of some appliances with binary behaviour as a **classification** problem rather than regression. Also, different **window sizes** to train and test the data should be explored.

References

[1] U. (United N. D. P. IEO (Independant Evaluation Office), «UNDP support to energy access and transition», *UNDP IEO*, diciembre de 2021. http://web.undp.org/evaluation/evaluations/thematic/energy.shtml (accessed 31st of May 2022).

[2] «Electricity Market Report - January 2022 – Analysis», *International Energy Agency*, enero de 2022. https://www.iea.org/reports/electricity-market-report-january-2022 (accessed 20th of May 2022).

[3] Energy Sector Management Assistance Program, «Regulatory Indicators for Sustainable Energy (RISE) 2020: Sustaining the Momentum | ESMAP», December 2020. https://www.esmap.org/rise_2020_report (accedido 31 de mayo de 2022).

[4] S. Mudie, «Energy benchmarking in UK commercial kitchens», *Building Services Engineering Research and Technology*, vol. 37, n.º 2, pp. 205-219, mar. 2016, doi: 10.1177/0143624415623067.

[5] S. Mudie, E. A. Essah, A. Grandison, y R. Felgate, «Electricity use in the commercial kitchen», *Int. J. Low-Carbon Tech.*, p. ctt068, sep. 2013, doi: 10.1093/ijlct/ctt068.

[6] Lucas Pereira, Catarina Rolim, Hugo Morais, Rui Castro, «nexIK: Exploring the Human-Water-Energy Nexus in Industrial Kitchens | Team», *nexIK*. https://nexik.tecnico.ulisboa.pt/team/ (accessed 31st of May 2022).

[7] *Sector Guide Industrial Energy Efficiency Accelerator Contract Catering Sector*. Accedido: 26 de diciembre de 2019. [En línea]. Disponible en: https://www.carbontrust.com/media/163491/contract-catering-sector-guide-industrial-energy-efficiency-accelerator.pdf

[8] «nexIK: Exploring the Human-Water-Energy Nexus in Industrial Kitchens». https://nexik.tecnico.ulisboa.pt/ (accessed 5th of July 2022).

[9] O. Laib, M. T. Khadir, y L. Mihaylova, «Toward efficient energy systems based on natural gas consumption prediction with LSTM Recurrent Neural Networks», *Energy*, vol. 177, pp. 530-542, jun. 2019, doi: 10.1016/j.energy.2019.04.075.

[10] W. W. Chan y P. S. Tse, «A study of the energy usage and future design of roasters in Cantonese kitchens», *International Journal of Hospitality Management*, vol. 20, n.º 1, pp. 45-60, mar. 2001, doi: 10.1016/S0278-4319(00)00034-7.

[11] F. W. H. Y. Bs. Ms. P. Ce. F. Mim. MHKIE, K. T. C. Bs. M. P. Ce. M. MHKIE, W. L. L. M. P. Ce. M. MHKIE, y P. S. K. S. BEng, «Establishing Energy Consumption Benchmarks for Commercial Complexes in Hong Kong», *HKIE Transactions*, vol. 8, n.º 2, pp. 40-47, 2001, doi: 10.1080/1023697X.2001.10667847.

[12] F. Romanelli, «Energy Management in Food Service Systems», *Cornell Hotel and Restaurant Administration Quarterly*, ago. 1978, doi: 10.1177/001088047801900203.

[13] R. B. DiPietro, S. Gregory, y A. Jackson, «Going Green in Quick-Service Restaurants: Customer Perceptions and Intentions», *International Journal of Hospitality & Tourism Administration*, vol. 14, n.º 2, pp. 139-156, abr. 2013, doi: 10.1080/15256480.2013.782217.

[14] R. Hedrick, V. Smith, y K. Field, «Restaurant Energy Use Benchmarking Guideline», NREL/SR-5500-50547, 1019165, jul. 2011. Accessed: 9th of November 2018. [Online]. Available at: http://www.osti.gov/servlets/purl/1019165-J8VWR3/

[15] F. Higgins-Desbiolles, E. Moskwa, y G. Wijesinghe, «How sustainable is sustainable hospitality research? A review of sustainable restaurant literature from 1991 to 2015», *Current Issues in Tourism*, vol. 22, n.º 13, pp. 1551-1580, ago. 2019, doi: 10.1080/13683500.2017.1383368.

[16] S. Mudie y M. Vadhati, «Low energy catering strategy: insights from a novel carbon-energy calculator», *Energy Procedia*, vol. 123, pp. 212-219, sep. 2017, doi: 10.1016/j.egypro.2017.07.244.

[17] L. Pereira, V. Aguiar, y F. Vasconcelos, «Future Industrial Kitchen: Challenges and Opportunities», en *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, New York NY USA, nov. 2019, pp. 163-164. doi: 10.1145/3360322.3360872.

[18] M. C. Almeida, «Analysis of energy consumption in the food service sector», p. 110.

[19] I. Aldaouab, M. Daniels, y K. Hallinan, «Microgrid cost optimization for a mixed-use building», en *2017 IEEE Texas Power and Energy Conference (TPEC)*, feb. 2017, pp. 1-5. doi: 10.1109/TPEC.2017.7868271.

[20] I. Aldaouab y M. Daniels, «Microgrid battery and thermal storage for improved renewable penetration and curtailment», en *2017 International Energy and Sustainability Conference (IESC)*, oct. 2017, pp. 1-5. doi: 10.1109/IESC.2017.8167472.

[21] W. Young, «Applying Solar Energy to Food Trucks», en *Proceedings of the SOLAR 2017 Conference*, Denver, 2017, pp. 1-8. doi: 10.18086/solar.2017.05.01.

[22] M. Bourdeau, X. qiang Zhai, E. Nefzaoui, X. Guo, y P. Chatellier, «Modeling and forecasting building energy consumption: A review of data-driven techniques», *Sustainable Cities and Society*, vol. 48, p. 101533, jul. 2019, doi: 10.1016/j.scs.2019.101533.

[23] X. Lü, T. Lu, C. J. Kibert, y M. Viljanen, «Modeling and forecasting energy consumption for heterogeneous buildings using a physical–statistical approach», *Applied Energy*, vol. 144, pp. 261-275, abr. 2015, doi: 10.1016/j.apenergy.2014.12.019.

[24] M. Bourdeau, X. Guo, y E. Nefzaoui, «Buildings energy consumption generation gap: A post-occupancy assessment in a case study of three higher education buildings», *Energy and Buildings*, vol. 159, pp. 600-611, ene. 2018, doi: 10.1016/j.enbuild.2017.11.062.

[25] G. R. Newsham y B. J. Birt, «Building-level occupancy data to improve ARIMA-based electricity use forecasts», en *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building - BuildSys '10*, Zurich, Switzerland, 2010, p. 13. doi: 10.1145/1878431.1878435.

[26] K. Yun, R. Luck, P. J. Mago, y H. Cho, «Building hourly thermal load prediction using an indexed ARX model», *Energy and Buildings*, vol. 54, pp. 225-233, nov. 2012, doi: 10.1016/j.enbuild.2012.08.007.

[27] M. Parti y C. Parti, «The Total and Appliance-Specific Conditional Demand for Electricity in the Household Sector», *The Bell Journal of Economics*, vol. 11, n.º 1, p. 309, 1980, doi: 10.2307/3003415.

[28] «Dagnely: Predicting hourly energy consumption. Can... - Google Académico». https://scholar.google.com/scholar_lookup?title=Predicting%20hourly%20energy%20consumption.%20Can%20you%20beat%20an%20autoregressive%20model%3F&author=P.%20Dagnely&publication_year=2015 (accessed 14th of June 2022).

[29] Z. Zheng, Z. Zhuang, Z. Lian, y Y. Yu, «Study on Building Energy Load Prediction Based on Monitoring Data», *Procedia Engineering*, vol. 205, pp. 716-723, 2017, doi: 10.1016/j.proeng.2017.09.894.

[30] J.-S. Chou y D.-K. Bui, «Modeling heating and cooling loads by artificial intelligence for energy-efficient building design», *Energy and Buildings*, vol. 82, pp. 437-446, oct. 2014, doi: 10.1016/j.enbuild.2014.07.036.

[31] C. Fan, F. Xiao, y Y. Zhao, «A short-term building cooling load prediction method using deep learning algorithms», *Applied Energy*, vol. 195, pp. 222-233, jun. 2017, doi: 10.1016/j.apenergy.2017.03.064.

[32] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, y D. B. Rubin, *Bayesian Data Analysis*, 0 ed. Chapman and Hall/CRC, 2013. doi: 10.1201/b16018.

[33] C. E. Rasmussen, «Gaussian Processes in Machine Learning», en *Advanced Lectures on Machine Learning*, vol. 3176, O. Bousquet, U. von Luxburg, y G. Rätsch, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 63-71. doi: 10.1007/978-3-540-28650-9_4.

[34] S. J. Taylor y B. Letham, «Forecasting at scale», PeerJ Preprints, preprint, sep. 2017. doi: 10.7287/peerj.preprints.3190v2.

[35] M. J. A. Shohan, M. O. Faruque, y S. Y. Foo, «Forecasting of Electric Load Using a Hybrid LSTM-Neural Prophet Model», *Energies*, vol. 15, n.º 6, p. 2158, mar. 2022, doi: 10.3390/en15062158.

[36] T. Bashir, C. Haoyong, M. F. Tahir, y Z. Liqiang, «Short term electricity load forecasting using hybrid prophet-LSTM model optimized by BPNN», *Energy Reports*, vol. 8, pp. 1678-1686, nov. 2022, doi: 10.1016/j.egyr.2021.12.067.

[37] N. Vasudevan, V. Venkatraman, A. Ramkumar, y A. Sheela, «Real-time day ahead energy management for smart home using machine learning algorithm», *IFS*, vol. 41, n.º 5, pp. 5665-5676, nov. 2021, doi: 10.3233/JIFS-189886.

[38] A. I. Almazrouee, A. M. Almeshal, A. S. Almutairi, M. R. Alenezi, y S. N. Alhajeri, «Long-Term Forecasting of Electrical Loads in Kuwait Using Prophet and Holt–Winters Models», *Applied Sciences*, vol. 10, n.º 16, p. 5627, ago. 2020, doi: 10.3390/app10165627.

[39] O. Foldvik Eikeland, F. M. Bianchi, H. Apostoleris, M. Hansen, Y.-C. Chiou, y M. Chiesa, «Predicting Energy Demand in Semi-Remote Arctic Locations», *Energies*, vol. 14, n.º 4, p. 798, feb. 2021, doi: 10.3390/en14040798.

[40] P. Ongsulee, «Artificial Intelligence, Machine Learning and Deep Learning», p. 6.

[41] E. A. Feinberg y Dora Genethliou, «Load Forecasting», en *Applied Mathematics for Power Systems*,

[42] E. Fix y J. L. Hodges, «Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties», *International Statistical Review / Revue Internationale de Statistique*, vol. 57, n.º 3, p. 238, dic. 1989, doi: 10.2307/1403797.

[43] O. Valgaev y F. Kupzog, «Building power demand forecasting using K-nearest neighbors model - initial approach», en *2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, Xi'an, China, oct. 2016, pp. 1055-1060. doi: 10.1109/APPEEC.2016.7779700.

[44] F. Wahid y D. Kim, «A Prediction Approach for Demand Analysis of Energy Consumption Using K-Nearest Neighbor in Residential Buildings», *IJSH*, vol. 10, n.º 2, pp. 97-108, feb. 2016, doi: 10.14257/ijsh.2016.10.2.10.

[45] D. Lachut, N. Banerjee, y S. Rollins, «Predictability of energy use in homes», en *International Green Computing Conference*, DALLAS, TX, USA, nov. 2014, pp. 1-10. doi: 10.1109/IGCC.2014.7039146.

[46] S. Paudel *et al.*, «A relevant data selection method for energy consumption prediction of low energy building based on support vector machine», *Energy and Buildings*, vol. 138, pp. 240-256, mar. 2017, doi: 10.1016/j.enbuild.2016.11.009.

[47] J. Massana, C. Pous, L. Burgas, J. Melendez, y J. Colomer, «Short-term load forecasting in a non-residential building contrasting models and attributes», *Energy and Buildings*, vol. 92, pp. 322-330, abr. 2015, doi: 10.1016/j.enbuild.2015.02.007.

[48] G. K. F. Tso y K. K. W. Yau, «Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks», *Energy*, vol. 32, n.º 9, pp. 1761-1768, sep. 2007, doi: 10.1016/j.energy.2006.11.010.

[49] Z. Yu, F. Haghighat, B. C. M. Fung, y H. Yoshino, «A decision tree method for building energy demand modeling», *Energy and Buildings*, vol. 42, n.º 10, pp. 1637-1646, oct. 2010, doi: 10.1016/j.enbuild.2010.04.006.

[50] C. Fan, F. Xiao, y S. Wang, «Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques», *Applied Energy*, vol. 127, pp. 1-10, ago. 2014, doi: 10.1016/j.apenergy.2014.04.016.

[51] L. Breiman, «Random Forests», *Machine Learning*, vol. 45, n.º 1, pp. 5-32, 2001, doi: 10.1023/A:1010933404324.

[52] A. Tsanas y A. Xifara, «Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools», *Energy and Buildings*, vol. 49, pp. 560-567, jun. 2012, doi: 10.1016/j.enbuild.2012.03.003.

[53] Z. Wang, Y. Wang, R. Zeng, R. S. Srinivasan, y S. Ahrentzen, «Random Forest based hourly

building energy prediction», *Energy and Buildings*, vol. 171, pp. 11-25, jul. 2018, doi: 10.1016/j.enbuild.2018.04.008.

[54] Y. LeCun, Y. Bengio, y G. Hinton, «Deep learning», *Nature*, vol. 521, n.º 7553, pp. 436-444, may 2015, doi: 10.1038/nature14539.

[55] J. Runge y R. Zmeureanu, «A Review of Deep Learning Techniques for Forecasting Energy Use in Buildings», *Energies*, vol. 14, n.º 3, Art. n.º 3, ene. 2021, doi: 10.3390/en14030608.

[56] Y. Ji, E. Buechler, y R. Rajagopal, «Data-Driven Load Modeling and Forecasting of Residential Appliances», *IEEE Transactions on Smart Grid*, vol. 11, n.º 3, pp. 2652-2661, may 2020, doi: 10.1109/TSG.2019.2959770.

[57] G. Xue, C. Qi, H. Li, X. Kong, y J. Song, «Heating load prediction based on attention long short term memory: A case study of Xingtai», *Energy*, vol. 203, p. 117846, jul. 2020, doi: 10.1016/j.energy.2020.117846.

[58] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, y Y. Zhang, «Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network», *IEEE Trans. Smart Grid*, vol. 10, n.º 1, pp. 841-851, ene. 2019, doi: 10.1109/TSG.2017.2753802.

[59] Z. Wang, T. Hong, y M. A. Piette, «Building thermal load prediction through shallow machine learning and deep learning», *Applied Energy*, vol. 263, p. 114683, abr. 2020, doi: 10.1016/j.apenergy.2020.114683.

[60] Z. Yong, Y. Xiu, F. Chen, C. Pengfei, C. Binchao, y L. Taijie, «Short-term building load forecasting based on similar day selection and LSTM network», en *2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2)*, Beijing, oct. 2018, pp. 1-5. doi: 10.1109/EI2.2018.8582673.

[61] C. V. Gonzalez Zelaya, «Towards Explaining the Effects of Data Preprocessing on Machine Learning», en *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, Macao, Macao, abr. 2019, pp. 2086-2090. doi: 10.1109/ICDE.2019.00245.

[62] D. Singh y B. Singh, «Investigating the impact of data normalization on classification performance», *Applied Soft Computing*, vol. 97, p. 105524, dic. 2020, doi: 10.1016/j.asoc.2019.105524.

[63] A. Cutler, D. R. Cutler, y J. R. Stevens, «Random Forests», en *Ensemble Machine Learning*, C. Zhang y Y. Ma, Eds. Boston, MA: Springer US, 2012, pp. 157-175. doi: 10.1007/978-1-4419-9326-7_5.

[64] G. Van Houdt, C. Mosquera, y G. Nápoles, «A review on the long short-term memory model», *Artif Intell Rev*, vol. 53, n.º 8, pp. 5929-5955, dic. 2020, doi: 10.1007/s10462-020-09838-1.

[65] S. J. Russell, P. Norvig, y E. Davis, *Artificial intelligence: a modern approach*, 3rd ed. Upper Saddle River: Prentice Hall, 2010.

[66] T. Chai y R. R. Draxler, «Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature», *Geosci. Model Dev.*, vol. 7, n.º 3, pp. 1247-1250, jun. 2014, doi: 10.5194/gmd-7-1247-2014.