Universidade de Aveiro

João
Nuno
Pereira
Silva

Sistema Inteligente para Gestão de Energia em Edifícios com Renováveis e Carregamento de Veículo para Rede

Intelligent Energy Management System for Buildings with Renewables and Vehicle-to-Grid Charging

João Nuno Pereira Silva

Sistema Inteligente para Gestão de Energia em Edifícios com Renováveis e Carregamento de Veículo para Rede

Intelligent Energy Management System for Buildings with Renewables and Vehicle-to-Grid Charging

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Informática, realizada sob a orientação científica de Paulo Jorge de Campos Bartolomeu, Professor Auxiliar do Departamento de Engenharia Electrónica, Telecomunicações e Informática da Universidade de Aveiro

Este trabalho foi realizado no âmbito do projecto EV4ENERGY (CENTRO-01-0247-FEDER-046995), obtendo financiamento do programa FEDER e da FCT/MCTES através de fundos nacionais e, quando aplicável, cofinanciado por fundos comunitários no âmbito do projeto UIDB/50008/2020-UIDP/50008/2020.

o júri / the jury

presidente / president — **Professora Doutora Pétia Georgieva Georgieva, Professora Associada, Universidade de Aveiro**

vogais / examiners committee — **Arguente Principal: Doutor Pedro Nuno da Silva Faria, Investigador Auxiliar, Isep - Instituto Superior de Engenharia do Porto**

Orientador: Doutor Paulo Jorge de Campos Bartolomeu, Professor Auxiliar em Regime Laboral, Universidade de Aveiro

agradecimentos / acknowledgements

Firstly, I would like to thank my parents and grandparents who always supported me throughout my academic journey.

Thanks to the University of Aveiro and its professors for everything that I have learnt and for giving me the tools to overcome this obstacle. I also want to thank the EV4ENERGY project for making this study possible, as well as my supervisor, Professor Paulo Bartolomeu, who was always available to guide me and help me find the answers to my questions.

Finally, i want to thank all the friends, both the ones that i already had and the ones that i met more recently, that were present in this chapter of my life.

Palavras-chave casas inteligentes, energia, EV, residência, previsão de consumo, algoritmo de decisão, sistema de gerência de energia, HEMS, ML

Resumo Nos últimos anos, as energias renováveis têm sido alvo de um forte desenvolvimento. A conscientização sobre a poluição por combustível fósseis tem vindo a aumentar e, com isso, o uso de veículos elétricos (EVs). Neste sentido, tem havido um esforço para manter a distribuição de energia sustentável e encontrar formas de reduzir o seu preço. O objetivo deste estudo é construir um algoritmo de decisão que ajude a minimizar os custos de energia elétrica de uma residência, fazendo uso de carregadores V2H (*Vehicle-to-Home*). Assim, os EVs podem ser usados como uma forma de armazenar energia que pode ser fornecida de volta à casa durante os períodos de maior necessidade. Uma das informações que o algoritmo proposto requer é a previsão do consumo energético da casa. Portanto, um modelo de previsão de consumo de energia doméstica foi também desenvolvido neste trabalho, incluindo uma versão que não requer informação histórica. Este modelo é útil enquanto não há informação histórica suficiente para treinar um modelo mais confiável. O algoritmo de decisão foi testado num ambiente simulado e comparado com um algoritmo de decisão base. Nos vários cenários e casas testadas, a abordagem proposta obteve uma redução média de 19.29% nas despesas energéticas da casa.

Keywords

smart home, energy, EV, household, load demand, consumption forecast, decision algorithm, energy management system, HEMS, ML

Abstract

Renewable energies have recently seen a strong development. The awareness of the masses regarding the pollution due to fossil fuels is rising and with it, the use of electric vehicles (EVs). Hence, there is an increasing effort to keep energy distribution sustainable and to find ways of reducing its price. The aim of this study is to build a decision algorithm that will help minimize the electrical bill of a household, making use of V2H (Vehicle-to-Home) chargers. In this approach EVs can be used to store energy, which can then be supplied to the household during periods of high demand. One of the inputs that the designed algorithm requires is the household's energy consumption forecast. Therefore, a energy consumption predictor was developed in this work altogether with a version that does not require past information of the specific household. This predictor is useful while there is not enough past data to train a more reliable model. The decision algorithm was tested in a simulated environment against a baseline decision algorithm. In the several scenarios and test houses, the proposed approach attained an average of 19.29% decrease in the energy expenses of the household.

Contents

List of Figures

List of Tables

Acronyms

ANN Artificial Neural Network.

CNN Convolutional Neural Network.

EV Electrical Vehicle.

FFNN Feed Forward Neural Network.

HEMS Home Energy Management System.

LSTM Long Short Term Memory.

MAE Mean Absolute Error.

MAPE Mean Absolute Percentage Error.

ML Machine Learning.

MLP Multi Layer Perceptron.

NN Neural Network.

RNN Recurrent Neural Network.

SoC State of Charge.

SVR Support Vector Regression.

V2G Vehicle to Grid.

V2H Vehicle to Home.

WAPE Weighted Absolute Percentage Error.

Chapter 1

Introduction

According to the UN, two thirds of the world's population is projected to live in urban areas by 2050 [3], which would inherently increase the demand for vehicles to provide urban mobility and, in turn, increase fossil fuel consumption and greenhouse emissions. Furthermore, the residential sector accounts for 27% of the global energy consumption [4].

In the most recent years, renewable energies have seen a strong development. The awareness of the masses regarding the pollution due to fossil fuel is rising and with it, the use of electric vehicles (EV). Bloomberg New Energy Finance (BNEF) reports that EV sales are set to increase more than 80% in 2021, in comparison with the last year [5].

According to the study in [6], Electric Vehicles could potentially provide 45% reduction in carbon emissions compared to conventional internal combustion engine vehicles after considering the energy cost of production, assembly, transportation, and usage. To further strengthen this trend, new legislation restricting carbon emissions has been limiting the use of petrol engine vehicles. As an example, the U.K. plans to ban sales of new petrol and diesel cars from 2030 [7].

In the future, however, an excessive number of electrical vehicles can put a lot of pressure in the energy supply grid if the charging of said vehicles is uncoordinated. This is already a problem in our everyday energy use since there are clear peaks of electricity usage, which puts the grid in a lot of pressure [8].

Renewable sources have great uncertainty and their availability can be very volatile. It is possible that the periods of high production and consumption don't occur simultaneously. Storing energy produced by renewable sources is a way of solving this problem. For example, if a smart home is able to produce electricity by utilizing solar panels, or if, at that moment, the electrical grid is supplying energy at low cost, but there is no current demand for electricity, a smart agent could find efficient ways of storing the extra energy, such as EVs or other available batteries, to be used later in the day when demand rises.

V2H (vehicle to home) technologies and bi-directional chargers are a factor that can provide economic value. They allow to provide the energy accumulated in an electrical vehicle back to the household that the vehicle is connected to. In this scenario, the car would act as an energy storage system, although, due to its uncertain availability, must be treated differently. This feature of electrical vehicles can be of great value if we are trying to better utilize energy that is being produced and consumed at different rates throughout the day.

1.1 Objectives

The current work is part of the EV4ENERGY project whose main objective is to put the EV at the center of the building energy ecosystem [9]. The current research aims at building a control algorithm that, by utilizing future household energy demand predictions and electric vehicle availability forecasts, can optimize the electric budget of a smart home.

The decision algorithm is meant to work in a contained environment, only receiving inputs related to the household. Based on those inputs, the algorithm charges or discharges the available batteries, in order to satisfy all energy needs in that environment.

One of the components of great value for this algorithm is a energy consumption predictor. This would allow the model to anticipate the demand in the household. For this purpose, a energy consumption predictor is also studied in this work.

The main objectives of this study are the following:

- Be able to predict household energy consumption in the short term.
- Be able to achieve accurate enough household energy consumption predictions out of the box, when there are no historical records available.
- Build a decision algorithm that can achieve significant cost reduction in the electrical bill of a smart house across different scenarios.

As mentioned, the proposed algorithm utilizes the energy stored in the connected EVs. However, since these vehicles need to be ready for use, the algorithm cannot have complete freedom to discharge them. The decision algorithm can also benefit from information relative to the connected EVs, such as departure time and energy required for the next trip. Ways of achieving this information without relying on the user are studied in the following chapter, however, a solution for this was not implemented.

1.2 Organization

The remainder of this document is structured as follows: Chapter 2 describes the state of the art techniques that are being used to forecast energy demand in the household perspective, as well as possible approaches to forecast Electrical Vehicle (EV) availability and load demand. In Chapter 3 a household load demand forecast system is built. Firstly, different machine learning techniques are studied and compared. Finally, two different models are built: a specific model that is optimized to predict energy consumption values for one specific household, and a generic model that can make good enough out of the box predictions for any household without further training. In Chapter 4 a energy management decision algorithm is designed and implemented. This algorithm utilizes the previous household energy consumption predictors, along other inputs, in order to minimize the electrical bill of a smart house.

Chapter 2

State of the art

In this chapter, a bibliographic research is conducted on the state of the art methods to forecast household energy consumption and EV behavior and energy necessities. The gathered information is relevant in order to implement a predictor that will serve as a input to a energy management decision algorithm.

2.1 Energy Forecast

This section presents the result of the bibliographic research regarding energy consumption forecast in residential buildings. Some of the presented techniques are used in this work to build a energy consumption predictor that is used as an input to a energy management decision algorithm. The achieved prediction can be used as a valuable insight on the future behavior of the household and allow the proposed system to take the necessary actions in advance.

2.1.1 Search

The selected key String was utilized in Scopus to retrieve state of art documents in this subject, although some of them were retrieved from previous research in the EV4ENERGY project. It was given priority to more recent (mostly more recent than 2016) and most cited papers. An analysis of the documents' abstract allowed for a narrower selection.

```
(House* or residen*) W/3 (electric* or energy or load) and (((electric*
or energy) PRE/1 (consumption or usage or demand)) or load) and
(forecast* or predict*) and ("machine learning" or ML or "deep learning")
and ("short-term" or "short term" or hour* or daily) and
(eval* or compar* or analysis)
```

2.1.2 State of the art

It is expected for certain independent factors to have an impact on energy consumption, such as the relationship between outdoor temperature and electricity demand. In Nordic countries, this relationship is easily visible, especially during cold seasons, as the use of space and water heating appliances increases.

In [10] a Kernel Density Estimation (KDE) is used to examine the relationship between temperature and energy consumption, providing the probability distribution of future demand values. Then, a hybrid method combining non parametric model and time series analysis is developed and compared against an auto regressive moving average model with exogenous variables (ARMAX) model that doesn't take into account temperature data.

Utilizing temperature data resulted in an improvement in the forecasting capabilities of the model. On the other hand, without the temperature data, the model fails to trace high spikes of the power curve, despite still achieving good results in the medium power ranges.

The study in [11] compares energy forecasting techniques that allow individual or aggregated prosumers to anticipate their future energy consumption. To this end, a set of models were compared. One of the tested models is a time-series based linear regression model, Seasonal Auto regressive Integrated Moving Average (SARIMA), secondly, a feedforward neural network, fully-connected multi-layer perceptron (MLP), in addition, a LSTM neural network and a support vector regression (SVR) model were also compared. An ensemble model, containing all other modules in addition to one that combines their outputs was also implemented.

Regarding aggregated consumption, results shown that the ensemble model performed the best (14.4% MAPE with weather data), followed by the SVR model (interestingly, SVR performed better without weather data: 15.4% MAPE without weather data). As for the individual households, the SVR model, despite still having a not so good accuracy, performed better than the other models (53.45% MAPE), including the ensemble model (68.51% MAPE) and the LSTM model (100.84%).

The work in [12] compares different ML models that aim at predicting hourly residential consumption. The compared models were: Linear Regression, feedforward neural network (FFNN), suport vector regression (SVR), Least Squares Support Vector Machines (LS-SVM), Hierarchical Mixture of Experts (HME) with Linear Regression Experts, HME with FFNN Experts, and Fuzzy C-Means with FFNN.

The study compared the previous models against several datasets, more importantly, a residential data set containing three residential houses. House 1 is a standard two-story residential home. House 2 is retrofitted with more energy efficient appliances, water heater, and HVAC. House 3 was built using construction techniques and materials designed to help reduce energy consumption. Additionally, it contains set of photovoltaic pannels for generating electricity and solar thermal water heater. A dataset containing commercial building was also utilized.

While neural network based models showed a better performance on commercial buildings, the best results on residential data were achieved by LS-SVM, achieving a MAPE value of 16.11%, 20.47% and 21.33% for house 1, 2 and 3 respectively. It is worth mentioning that SVR achieved similar results. Additionally, Fuzzy C-Means with FFNN achieved the following MAPE values: 17.57%, 21.96%, 24.20% for house 1, 2 and 3 respectively, performing the best in terms of neural networks based models for residential data.

In [13] three machine learning techniques are implemented to predict energy consumption. The study focuses on the prediction of the future energy consumption of an Estonian household and it is based on data recorded minutely for one month.

The machine learning models implemented are linear regression, tree based regression and quadratic or dual SVM regression.

Tree based regression shows better results when compared to linear regression. However, SVM is the best performing model out of the three, its RMSE value is half of the linear regression model and 28% less when compared to tree based regression. In addition, SVM has a lower training time, 1.75 sec, while tree based regression takes 2.63 sec.

The work in [14] aims to compare five ML algorithms in hourly and daily forecast. The five evaluated models are artificial neural network (ANN), support vector regression (SVR), least-square support vector machine (LS-SVM), Gaussian process regression (GPR) and Gaussian mixture model (GMM). It is also presented a hybrid model that integrates LS-SVM with a forward physics-based models, utilizing air conditioner data.

ANN, SVR, LSSVM and the hybrid model perform similar in relation to daily consumption forecast. The hybrid model achieved slight improvements in hourly prediction. Overall, the models achieved a better performance when forecasting the next hour consumption compared to next 24 hour consumption.

The study in [15] analyses the problem of energy consumption prediction at single household level. The proposed model consists on a Support Vector Regression (SVR) (a version of SVM for regression problems) with both daily and hourly data granularity. The model was trained with past hourly electricity consumption (obtained with smart meters), weather conditions, temperature, humidity, hour of the day, day of the week, month, season and electricity TOU rate.

Most of the 15 households used in this research had a good MAPE value for daily forecast, with the best value being 12.78% and the worst 34.95%. The average MAPE was of 22.64%. The accuracy for hourly forecast suffers from high fluctuation, with the best value being 23.31% and the worst 64.38% (average of 37.3%). In addition, results show that five households, whose consumption patterns were more unpredictable, benefited from random sample splitting, in contrary to the other households, with similarities in their consumption patterns over time, that performed better with time-based training and testing subsets splitting.

In the study [16] the authors define a feed forward artificial neural network (ANN) to forecast hourly and daily energy consumption at the household level in Portugal. The inputs used are historical data load, electric appliance, occupancy and area apartment, for a total of 16 inputs, weather data was not used.

The ANN architecture was built using a three-layer feedforward configuration, the input layer contains 16 and 11 neurons for daily and hourly model respectively, 20 neurons in the hidden layer (after trial and error between 3 and 30 neurons), and 1 neuron in the output layer. The network was trained using the Levenberg-Marquardt algorithm that enables pruning, a processes that determines units that are not necessary to the network and removes them, reducing the complexity of the network.

For the daily average consumption, the model scored a MAPE value of 4.2%. The hourly model was tested in two houses for three days, scoring a MAPE value of, for the first house, 16.0%, 10.0% and 12.9% for the first, second and third day respectively, and for the second house, 21.5%, 19.3% and 23.5% for the first, second and third day respectively.

An ensemble based machine learning framework for day-ahead forecasting of household energy consumption is presented in [17]. This work, utilizes ANNs as ensemble members due to their resiliency in generating diverse realizations.

To improve generalization, resamples are performed in the training stage trough a two-stage diversity controlled random sampling procedure. The diverse resamples make it so that the ensemble members have different optimum configuration as the training is carried out with different information. In the ensemble integration phase, where estimates obtained from the ensemble members are combined to produce the final ensemble estimate, Multiple Linear Regression (MLR) is utilized. The used features consist of electricity consumption data from the four previous days and from the same day, but from two previous weeks, as well as temperature data.

The model was tested against a single ANN model and a ANN-based Bagging ensemble model. The proposed model scored a MAPE value of 14.43%, the best value compared to the single ANN (18.3% MAPE) and ANN-based Bagging ensemble model (15.16% MAPE).

The authors of the study in [18] propose a long short-term memory (LSTM) recurrent neural network (RNN) based framework for short-term load forecasting for individual residential households.

LSTM is a RNN architecture, recurrent neural networks are "sequence-based models, which are able to establish the temporal correlations between previous information and the current circumstances". This characteristic makes this model a strong candidate for load forecasting problems since householders' routine have significant impact at later time intervals.

The LSTM model receives energy consumption data from the past K steps as well as day of the week information, the outputs are fed to a conventional feed forward neural network (FFNN) which maps the LSTM output to a single energy consumption forecast value.

The proposed approach was compared against other ML models (backpropagation neural network (BPNN), k-nearest neighbour (KNN) regression, extreme learning machine (ELM) and input selection combined with hybrid forecasting (IS-HF)). In addition to individual household forecasts, the authors also tested the models for aggregated household prediction. The LSTM model was the best model for individual (MAPE: 44.06%) and aggregated forecast (MAPE: 8.58%). The difference in MAPE values shows the difference between individual and aggregated household prediction as the aggregated load smooths the profiles, while individual households are prone to unpredictable peaks.

Electrical consumption is highly related with factors such as temperature, day of the week, etc. However, one cannot disregard the correlation that residents' behaviour has with energy consumption throughout the day, for example, events like shower and laundry will occur at similar times during the several days.

The authors of the study in [19] included appliance measurements in the training data of a LSTM recurrent neural network to tackle this issue. The objective is to better classify the lifestyle of the household, improving the interpretation of volatile peaks. Out of 19 appliances, 6 were selected as the most important ones and used to train the model, they were: clothes dryer, clothes washer, dishwasher, heat pump, television, and wall oven.

The performance of the model was tested against a FFNN and a KNN algorithm, also, the models were trained only with total house measurements and, for comparison, total measurement plus appliances. The proposed LSTM outperformed the FNN and KNN algorithm and in all cases. Models that were trained with appliance data outperformed house measurements only. The best LSTM model achieved a MAPE of 21.99%.

The method proposed in [20] consists on a CNN-LSTM neural network that aims at predicting housing energy consumption. The CNN layers receives input data processed every hour by a sliding window algorithm and extract features from variables affecting energy consumption. The output is then used as the input to the LSTM layer, which is suitable for modeling temporal information of irregular trends in time series components. Lastly, a fully connected layer generates a predicted time series of energy consumption.

The model was compared to conventional ML models: linear regression model (LR), random forest regression (RF), decision tree (DT) regression and multilayer perceptron (MLP), as well as other deep learning based models: LSTM, GRU, Bi-LSTM, and Attention LSTM. The proposed method outperformed all the compared models. The model was tested minutely, hourly, daily and weekly. The performance results of the proposed model resulted in a MAPE value of 32.83% and 31.83% for hourly and daily prediction, respectively.

A similar approach is implemented in [21], where the authors propose another CNN-LSTM model for individual household load forecasting. CNN models are good for extracting valuable patterns and filtering out noise. However, they are not the best model for long temporal dependencies. On the other hand, LSTM works well with temporal dependencies but only utilizes data attributes provided in the training set. The idea of this paper is to build an hybrid model that combines the strengths of both models. The considered features were: hour of the day, day of the week, energy consumption and holidays.

The proposed model was compared with other models: ELM, KNN, back propagation neural network (BPNN), input selection combined with hybrid forecasting (IS-HF) and the deep learning model based on LSTM (without CNN). The proposed model achieved better performance than all the other models, having a MAPE value of 40.38%. Results also show that, for the proposed model, the MAPE value increases less, as the prediction window get bigger, in comparison to the LSTM only model.

The authors also performed a clustering analysis using the proposed method in an attempt to improve performance by grouping similar households. The day was subdivided into four periods (overnight, breakfast, daytime and evening) and households with similar profiles were grouped. While in some cases this approach led to a better MAPE value, on average the MAPE value increased to 44.76%.

The study in [22] proposes a hybrid deep learning framework that combines multiple LSTM neural networks with stationary wavelet transforms (SWT) to perform energy consumption forecasting for individual households. The objective is to use SWT to alleviate the volatility and increase the data dimensions, in order to increase the LSTM forecasting accuracy.

Energy consumption data (or signal) is split into several sub-signals using stationary wavelet transforms. For each sub-signal, a LSTM neural network is used to perform a forecast. The forecasting results of the all the sub-signals are integrated using inverse stationary wavelet transform.

The proposed model achieved a MAPE value of 12.84% for 30 minutes step-sized data. The model was compared against other models: the persistent method (28.65% MAPE), support vector regression (SVR) (23.42% MAPE), single LSTM neural network (28.25% MAPE) and hybrid CNN-LSTM (25.22% MAPE).

2.1.3 Discussion

Forecasting household energy consumption is a difficult subject. Unlike the forecast of the electrical consumption in large buildings or aggregated city loads, where the energy demand curve is smooth due to unpredictable events not having a great impact in the overall picture, single family homes have very volatile peaks and minimums in their energy consumption. Despite this, energy consumption still follows a reasonably predictable behavior and variables like weather, temperature and appliance usage can help a machine learning model to establish relations and achieve better performance.

As presented, SVR is a not so complex machine learning technique based on support vector machines that is used in these problems and presents a realistic way of predicting future energy demand [15]. LS-SVM is a similar approach that also has similar results to SVR [12].

Neural Networks (NNs) are a very common method in machine learning and in this problem it is no exception. A simple MLP can achieve good results as can be seen in [16], especially considering that, in said study, no weather data was utilized. MLP is a type of FFNN where every neuron in one layer is connected to all neurons in the next layer. This is the most common type of NN and is often referred as ANN.

However, one type of neural network has a significant portion of the studies in this area: the LSTM neural network [18]. Some studies show that the predicting performance of the LSTM model can improve. One of these techniques [20] works by adding a CNN layer to reduce noise and improve the number of features. Other example [22] works by adding a SWT filter to decompose the consumption signal into sub signals and use several LSTM models for each subsignal.

Because neural networks are a more complex technique, it would make sense that it would surpass simpler models, however, studies that compare these approaches, in relation to energy consumption prediction [11, 12] claim that SVR achieved a better score than LSTM. However, from a purely theoretical point of view, it does not make sense that a SVR model, which does not take time into consideration, would achieve better results in a time series problem when compared to a LSTM neural network. Although recent studies show complex approaches and are able to improve the accuracy of LSTM based models, the complexity of the implementation is a factor worth considering.

It is worth noting that comparing the results of the model implementations is a hard task since these were tested against different data sets, while also trying to predict results in different time intervals. What proves to be a good solution in one case may not be in a different scenario, depending on the provided features, temporal correlation of the dataset and its time span. An overview of the most relevant studies can be seen in Table 2.1.

Model	Prediction Interval	MAPE (%)
LS-SVM [12]	hourly	19.3
SVR [15]	hourly	37.3
SVR [15]	daily	22.64
ANN [16]	hourly	17.2
ANN [16]	daily	4.2
LSTM [18]	30 minutes	44.06
LSTM w/ appliance [19]	30 minutes	21.99
CNN-LSTM [20]	hourly	32.83
CNN-LSTM [20]	daily	32.83
CNN-LSTM [21]	30 minutes	40.38
LSTM [22]	30 minutes	28.25
SWT-LSTM [22]	30 minutes	12.84

Table 2.1: Household load demand forecast research summary

2.2 Electric Vehicle Trip and Charge Behavior

This section presents the result of the bibliographic research regarding electric vehicle availability and energy necessity forecast in the household perspective. This study was conducted in order to gather information on how to predict EV behavior and necessities (such as departure time and energy requirements for the next trip). This was considered an important input to the proposed energy management decision algorithm, since one of the possible actions this algorithm would have is to charge or discharged an EV connected to a household. A accurate predictor would enable the decision algorithm to act in the best way possible without the risk of harming the daily lives of the EV user. It is worth noting that this field of study is not so advanced and there was a higher difficulty to find valuable studies. Some of the presented works are not exactly in the same area but were still considered relevant within this study. The lack of studies and, most importantly, datasets in this area made impossible to finalize this study with an implementation of said predictor.

2.2.1 Search

The selected key String was utilized in Scopus to retrieve state of art documents in this subject, although some of them were retrieved from previous research in the EV4ENERGY project. Documents were prioritized based on the year (mostly more recent than 2018). An analysis of the abstracts allowed for a better selection.

```
(EV or "electric vehicle" or "electric vehicles") PRE/1 ((charg* PRE/1
(load or behavior or demand)) or availability or "smart charging") and
("machine learning" or ML or forecast or prediction)
```

2.2.2 State of the art

The methodology presented in [1] is based on driving pattern data in USDOT's National Household Travel Survey and it is used to simulate PHEV charging and gasoline consumption. The information relative to the trips of approximately 170,000 vehicles is used to track EV battery SoC throughout the day and to determine EV charging patterns. Additionally, demographic information is considered to evaluate how the driver characteristics can influence the vehicle consumption habits.

The study's results show that a compact PHEV travels on average 66.7% in Charge depleting (CD) mode, consuming an average of 47.3 kW/h and 7.2 L of gasoline per week. The average utility factor was 63.5% for males and 72.9% for females and, in both cases, it is sensitive to age. The charging peak occurs at 8 PM most days, as can be seen in Fig. 2.1, although drivers in older age groups have earlier charging peaks.

The study in [2] explores the charging behavior of nearly 8000 EVs to investigate where people charge, be it at home, at work, or at public location.

This study shows that EV users tend to do most of the charging at home, Fig. 2.2, further demonstrating the importance of this topic. According to the study, socio-demographic (gender and age) patterns, vehicle characteristics, commute behavior and workplace charging availability influence charging tendencies. EV users with charging facilities at home also use workplace charging infrastructure when the residential electricity rate is high or workplace charging is free, however, charging vehicles at the workplace is significantly reduced during the weekend.

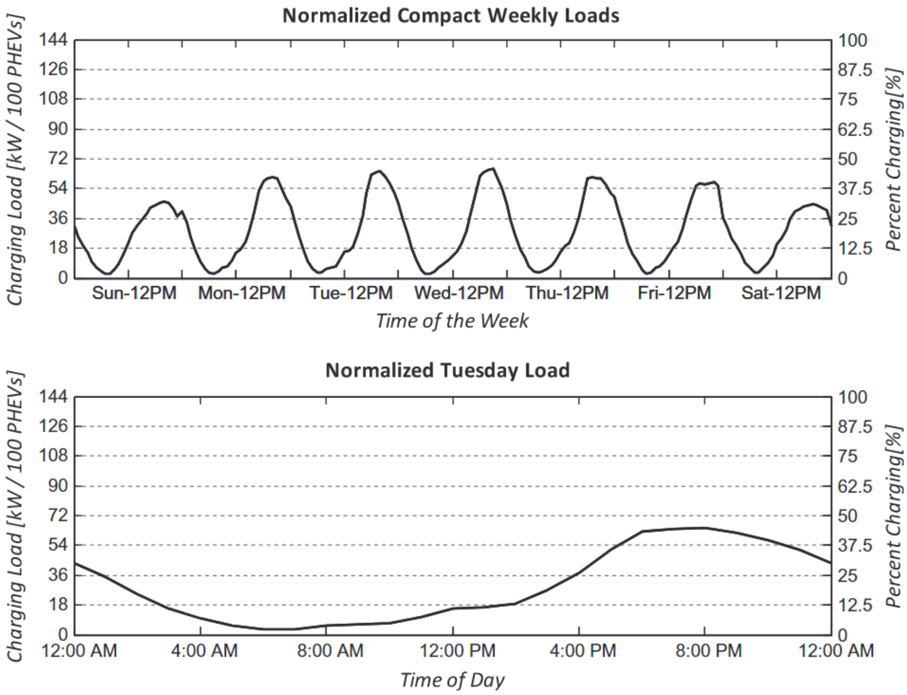


Figure 2.1: Charging behavior [1]

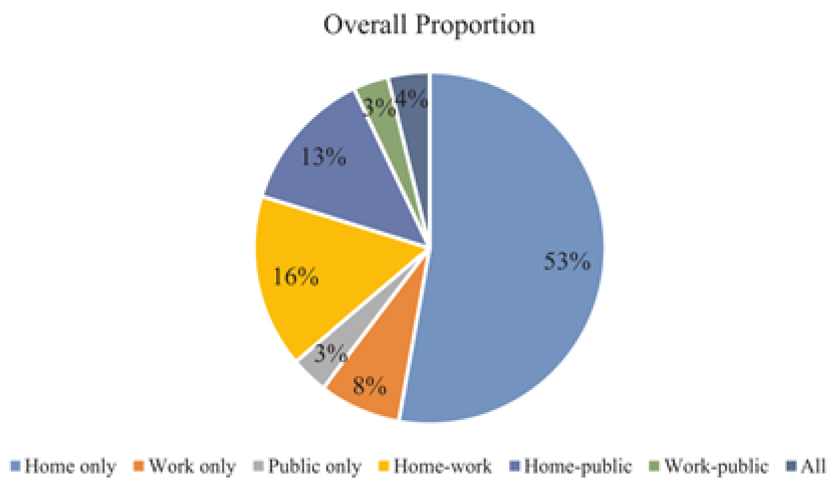


Figure 2.2: Charging proportion [2]

The lack of real world data is a problem in the area of EV user behavior, as a way to solve this issue, the research in [23] proposes a EV charging load simulation method, utilizing the Monte Carlo method, based on probabilistic distribution models of temporal-spacial travel mode, charging preference and energy consumption rate. Furthermore, data indicates that the user demographics impacts his driving behavior. As an example, the departure and arrival time of young adults is much less variant than that of elderly people. With this in mind, user demographics and social characteristics are also taken into account (age, gender and education level).

As already stated, the Monte Carlo method was used to simulate travel behavior. Variables were sampled sequentially with probability distributions. The start time of the trip, starting at the household, was generated first, from there, the destination was sampled, then the driving time, distance and finally parking time, according with the ending time of the parking action, the next trip is simulated.

Based on the simulated traveling behaviors of each user, the charging load profile are also simulated, considering charging preference. Two charging preferences are considered. One of them, is conservative charging, where users charge their EV batteries when the remaining energy is unable to support the next trip. The other is a positive charging preference, where EVs will always be charged.

The authors performed a case study where 100000 EVs were simulated. Results show that the proposed method can simulate realistic profiles based on the user demographics.

The work in [24] introduces a data-driven framework of charging load profile generation for residential plug-in electric vehicles, utilizing real world historical residential charging behavior data.

The analyzed data shows that lots of daily residential parking behavior only has one time of parking at home. However, there are many cases where more than one parking action occurs within one day. Most residential parking behaviors don't include more than five parking actions.

The proposed framework utilizes the number of home parking events for each vehicle within a day, time information (arrival time and departure time) and state of charge (SoC) information (start and end). Utilizing the SoC values, it is possible to know whether the vehicle performed a charging action during the home parking. In this work, the authors assumed that the decision to charge or don't charge an EV, when a parking action occurs, is determined by the parking duration and SoC.

The K-nearest algorithm was utilized to construct a charging decision making model, the goal is to decide whether a charging action is necessary when a new parking action occurs. In case a charging action is necessary, a charging duration model decides what would be the charging time, in order to calculate the amount of energy to be charged.

The relation between charging duration and SoC is explored. However, data shows that charging duration can have randomness with regard to parking duration. To handle uncertainty, a multiple channel method, utilizing a Kernel Density Estimator and a Probability Density Function is considered, producing the probability density function for each given parking duration interval.

No result values were presented in the paper. However, visualizations show that the proposed model achieved a good capability to generate reasonable residential charging load profiles under different charging rates.

Multiple machine learning algorithms are implemented in order forecast the household day-ahead EV charging time in [25].

The paper claims that an accurate prediction of "no charge" is relatively more crucial than other labels, as EV charging is a heavy electrical load and the management system would attain better flexibility on load shifting. Therefore, this paper also aims at forecasting the no-charge days. This paper also wants to build a pre-recognition prediction that provides a hint about the result. The pre-recognition module and the no charge module outputs feed the final predictor that outputs the final results.

The utilized data has a time horizon of 20 days. The utilized algotithms were: RF (configured with 100 decision trees), AdaBoost and Gboost (both with 100 decision trees), gaussian Naive Bayes, KNN (5 nearest-neighbors) and, finally, ANN (configured with 1000 hidden neurons). A naive predictor is also implemented, as a baseline, that predicts the same value as the last observation.

All the algorithms performed better than the naive predictor. The four best models were chosen to build an ensemble model that was then used to be the pre-recognition module, these modules were (in order): RF (accuracy: 0.634), Naive Bayes, AdaBoost and GBoost. It is noted that adding other models to the ensemble algorithm damaged the accuracy of the ensemble model. For the no charge label, AdaBoost achieved an accuracy of 0.996, better than any individual or combined models. The final predictor module, that combines the last two modules outputs, was an ensemble model composed of RF, GBoost, NB and AdaBoost and achieved an accuracy value of 0.724, a 9% increase over the best individual model, RF.

The research in [26] collected EV public charging behaviour data and proposed a model that combines statistical analysis and machine learning techniques, namely, unsupervised clustering algorithm and MLP, to perform day-ahead EV parking and load prediction.

For this study, data from each charging session was captured, including plug-in time, charging start time, charging stop time, charger plug-out, charging current and power consumption values per minute. Due to having users with distinct behaviour, it is very difficult to have a unique distribution to model the different behaviours. The authors utilized K-means clustering to partition the data and to create assumptions of the the users charging behaviour. According to the experiments, users in this study can be categorized in four groups.

Multilayer perceptron was utilized to process user charging data records and perform classification based on both clustering labels from K-means algorithm and hand-labeling. The purpose of this network is to perform future classification on any new users, without having to process the whole dataset again.

The user groups obtained were used to generate the day-ahead predictive energy demand boundary using statistical analysis. Results show that the model was able to predict EV load demand with a MAPE value of about 24%.

The research in [27] studies the prediction of EV charging session duration and energy consumption in an university campus. Besides utilizing a historical charging dataset, this study also uses weather, traffic and local events information. The authors compare four machine learning algorithms: RF, SVM, XGBoost and ANN. Additionally, two ensemble models that utilize the previous models (except the ANN) were implemented, one using voting regressor and the second using stacking regressor.

Interestingly, the most important feature for predicting session duration was maximum traffic after arrival. In relation to session duration, out of the singular models, ANN with 3 hidden layers performs the worse with a mean absolute error (MAE) of 73.7 minutes. SVM, RF and XGBoost perform similarly with 67.4, 68 and 68 minutes of MAE, respectively. The ensemble models perform the best out of all the models in all the evaluation scores. Voting ensemble achieved a MAE of 66.5 minutes and stacking ensemble a score of 67.1 (stacking ensemble was better in other evaluation scores). User predictions for session time were also part of the study and achieved a MAE value of 394 minutes, therefore being really far of.

In relation to energy consumption, RF was the best individual model (3.39 kW/h MAE value). The best model was Stacking ensemble (3.38 kW/h MAE value) in all parameters but the improvement was not significant compared to session time prediction.

Authors concluded that weather, traffic and events data resulted in an improvement in the EV charging behavior predictions, comparing to other studies that utilized the same dataset.

The work in [8] explores smart charging with the goal of preventing uncoordinated electric vehicle charging. To achieve this goal, the forecast of energy requirement (through next trip distance) and parking duration is explored.

The dataset that was utilized in this work contains logs of multiple users that recorded their travel behaviour for one week. As the time series for each user is very short, learning weekly patterns is unlikely. The researchers assumed that the users behaviour followed a weekly pattern and that there are similarities within the user types. In addition, the only parking events considered were home parking. The approach consisted in deriving aggregated features for single parking events. The features utilized were: parking event (index, start time, duration), length of the trip that followed the parking event, user (id, type), previous parking location, previous trip duration and distance, and previous trips (duration, distance and number).

The compared models were: a naive predictor that predicted the last observed value, Quantile regression, Quantile regression with MLP and, finally, Multivariate conditional kernel density estimator. The last model did not show interesting results. Quantile prediction allows accounting for the uncertainty in the forecast. Quantile predictors provide more information in decisions under uncertainty since they provide the interval in which the output is, within a given probability. The authors argue that "in EV charging, it is not only essential to know whether the starting time will be short or long, but also whether the forecaster is confident in its forecast".

Results show that the MLP exhibits the best results on average. Quantile regression also outperforms the naive benchmark in both parking duration and next trip distance. For all predictors utilizing previous user data to train the model outperformed utilizing only user type and current session information (improves the forecasting accuracy by 13.7% for parking duration and 0.56% for trip distance compared to the data generated at the charging stations). However, some predictions were very incorrect and no pattern for these outliers was found. Overall, predictions suffered from large mean absolute percentage error (MAPE) values, with the best model (MLP) achieving a MAPE value of 299.37% for parking duration and 103.35% for trip distance. The authors claim that these values were caused by the occurrence of high values in the distribution (occurrence of very high parking durations, outliers, etc). The median absolute percentage error (MdAPE) has more acceptable values: 29.83% for parking duration and 45.42% for trip distance. The study claims that even with low forecasting accuracy, the predicted information still helps with ranking the flexibility of different charging processes.

In the same work, a case study was performed. Here, a coordinator can control the charging of all EVs in the test set. If no coordination is enforced, the number of charging processes would lead to a grid congestion on distribution level. The coordinator can interrupt charging processes given an heuristic. The utilized heuristics were: random, FIFO (first-come-first-out), EV with the highest remaining parking duration first and, finally, EV with the shortest next trip.

For this study, interrupting charging sessions at random instants impairs 25.1% of mobility events. The FIFO heuristic improves the percentage of impaired mobility events down to 17.5%. The next trip distance heuristic outperforms the random heuristic but is worse than the FIFO. Parking duration heuristics results in a impaired mobility of 16.5% when using lower quantiles (looking for a long parking duration with high certainty).

The work in [28] proposes a SVM model to perform short term load forecasting of electrical vehicles utilizing driving and travel patterns.

This work tackles the lack of real historical data of EV charging events by producing charging event data utilizing national (UK) statistical data such as travel distance and trip duration.

The utilized features in this study were: time data (week, day and half-hour), new EV plug-in connections for every half hour, (normalized and non-normalized) charging demand for each half hour, and historical data (previous day and week normalized load and average of the four previous weeks).

The results were compared with the Monte Carlo method showing that the SVM has a better performance and is more sensitive to hourly fluctuations, achieving a MAPE value of 3.69%.

The study in [29] investigates EV charging behavior in order to optimize the charging schedule. The aim is to predict stay duration and energy consumption based on historical charging records to determine the energy allocation schedule.

The data utilized contains users with at least 100 charging records from a university campus and from residential users in the UK. The objective is to predict the stay duration once a user initiates a charging session, utilizing the usersID, start time and day of the week. Based on the output, the stay duration is then used, in combination with start time and day of the week, to calculate energy consumption.

This paper compares a set of models: Multiple Linear Regression (MLR), Support Vector Regression (SVR), Decision tree (DT) regression, Random Forest (RF) Regression, K-Nearest Neighbor (KNN) Regression, Kernel Density Estimator (KDE) and a statistical model that predicts the average of the past data as a baseline.

Results show that, for duration, SVR performs the best overall (SMAPE value of 10.54%), but Diffusion based KDE (DKDE) is better when the ratio of entropy/sparsity of the user in terms of start time vs stay duration is high. For energy consumption, RF is better overall (SMAPE value of 8.65%), but DKDE is better when the ratio of entropy/sparsity of the user in terms of stay duration vs energy consumption is high. The ratio of entropy/sparsity is greater when the past data of the user has more different values and therefore, is more unpredictable.

Based on the findings, the authors proposed an ensemble algorithm that, based on the previous user data, determines whether SVR or DKDE would be used for predicting stay duration and whether RF or DKDE would be used for predicting energy consumption. This model decreased the prediction error by 11% and 22% for duration and energy consumption, respectively.

The scope of this project was to perform better scheduling of EV charging. Utilizing a simulation with real world data, researcher claim that the achieved model can reduce 27% of peak load, 10% of load variation, and 4% cost reduction, compared to uncoordinated charging.

In [30] an ANN and a LSTM are compared in the task of EV charging load prediction from the charging station perspective.

The selected LSTM neural network was trained with the mean squared error (MSE) loss function and Adam optimizer. The data utilized has the range of one year and was grouped in intervals of 15 minutes and 30 minutes.

Results show that ANN achieved a RMSE value of 36.6 and 76.4 and a MAE value of 21.7 and 43.1 for 15 and 30 minutes respectively. The LSTM achieved a RMSE value of 12.3 and 28.2 and a MAE value of 5.5 and 16.8 for 15 and 30 minutes respectively. It is possible to see that LSTM has a lower prediction error than traditional artificial neural networks, as expected since this is a time series problem. It is also worth noting that with bigger time intervals, the prediction error increases.

2.2.3 Discussion

Electrical Vehicles can be have a big impact when trying to optimize the energy budget for a smart home. Since V2H technologies allow the vehicle to provide electricity back to the household, acting as a energy storage system, they can provide economic value in periods of high energy demand. Energy flexibility is the difference between the minimal required state of charge (SoC) and a fully charged battery. However, EVs are, first and foremost, the main means of transport of its user so a way of forecasting the EV availability and energy demand is needed.

Forecasting the behavior and load necessities of an EV in the household perspective is a difficult problem to solve. The predictability of user behavior is dependent of the EV user demographics, as was noted in [1, 2, 23]. Some EV users have travel patterns that are predictable, for example the departure time of and arrival time of a person that has a fixed schedule job.

The lack of real word datasets in this area is one of the biggest constraints when trying to solve the current problem. To try to overcome this issue, the work in [23] proposed a load profile simulation where users and their daily trips were simulated based on real world data and demographics. The correspondent charging demand was also simulated. The work in [28] also tackled this issue by producing charging event data based on statistical data, this time, to evaluate a SVM model.

The area of machine learning applied to electric vehicle behavior and charging load is very unexplored. However, some related papers were analyzed in this chapter.

A major part of the research in this area is not on the perspective of the household, even though most of the EV charging is performed at home, as pointed by [2]. Some of the studies that were collected perform EV charging behavior forecast in public charging stations, where the specific user past history is not really considered. However some of these works mainly utilize past load demand to forecast future load demand [27, 30, 29], which can also be use full in the current problem.

The work in [8] performed a case study where certain heuristics were used to decide if a vehicle should be charged or not, in a scenario where several EVs needed to be charged, but couldn't charge at the same time. The best results were achieved by prioritizing EVs with a short predicted parking duration.

The results of the investigated in the area of EV load demand and availability are summed up in Table 2.2.

However, the user behavior can be volatile. The user may decide to make an unexpected trip and it would be a major inconvenience if the EV did not have the energy required for the trip. This means that it is very important to account for errors. It is unlikely that drivers would accept smart charging that could impair their mobility. As a matter of fact, operators mostly rely on the users' agreement to enable charge flexibility. Another way of dealing with this problem would be to account for user input. If drivers are confident that they don't need a full SoC for their next trip, they can provide that information to the system in order to allow better energy flexibility. Alternatively, users can define profiles that fit their driving habit.

2.3 Conclusion

As mentioned, the current work had as two of its goals to forecast household electricity demand and EV availability/load necessity. To reach that goal, a state of the art research was conducted to find the available options for reaching this objective.

These two inputs would be utilized together to feed a smart agent that optimizes the electrical budget of a smart house in a way that makes use of V2H technologies, that allow electricity, that is being generated by renewable sources or during periods of low demand, to be stored temporarily in a EV and supplied to the household in critical periods.

While household electricity demand forecast is a well researched field and some options are well established to solve this problem, it is not clear what the best approach might be for the specific dataset that was utilized in this study. Therefore, a practical experiment of the previously discussed models can be helpful is deciding which model to use.

On the other hand, EV availability and load necessity forecast is not a field where much research was conducted, specially with regard to the household perspective in this matter. Additionally, there is a serious lack of public datasets that contain the necessary data to conduct a complete experiment on this subject.

The combination of these two modules to optimize energy consumption is, to the author's knowledge, something new, relative to the existent studies in the area of electric consumption in the household perspective.

Model	Objective	Results
Ensemble model (Random Forest, GBoost, Naive Bayes and AdaBoost) [25]	Forecast the household day-ahead EV charging time	Accuracy : 0.724
RF [25]	Forecast the household day-ahead EV charging time	Accuracy : 0.709
K-means [26]	Day-ahead EV public parking and load prediction	MAPE: 24%
Ensemble model (SVM, RF, XGBoost) [27]	EV public charging session duration	MAE: 66.5 mins RMSE: 97.7 (Individual models were close)
Ensemble model (SVM, RF, XGBoost) [27]	EV public charging session energy consumption	MAE: 3.38 KWh RMSE: 5.5 (RF was close)
MLP [8]	Forecast of energy requirement (trough next trip distance) and parking duration	Parking duration MAPE: 299.37%, MdAPE: 29.83% Trip distance MAPE: 103.35%, MdAPE: 45.42%
SVM [28]	Short term load forecasting of EVs	MAPE: 3.69%
SVR [29]	EV Predict stay duration (public charging)	SMAPE: 10.54%
RF [29]	EV Predict energy consumption (public charging)	SMAPE: 8.65%
LSTM [30]	EV charging load prediction from the charging station perspective	RMSE: 28.2 MAE: 16.8 KWh for 30 minutes intervals

Table 2.2: Ev availability forecast research summary

Chapter 3

Household Load Demand Forecast

Energy consumption within a household will frequently follow a pattern. A family can have repeated behaviors and, probably, somewhat constant lows and peaks of energy usage. Historical energy usage records and Machine Learning (ML) can be utilized as a way of building knowledge on the consumption habits of a household and based on the most recent consumption, predict the near future energy demand.

The objective of this study is to utilize machine learning techniques to build a model that can learn the consumption habits and patterns of a household in order to forecast the future electrical load of that particular household. This is called short term time series forecasting: a technique in machine learning, which analyzes data and the sequence of time to predict future events. It is a short term forecast due to small time step that is being predicted (one hour). This means that the records of the train data were collected on hourly intervals and the model predicted the consumption of the next hour.

Machine Learning models usually need to be trained with historical data, which in some cases might not be available and needs to be collected first. Additionally, if the amount of data is small, the machine learning model cannot be expected to preform well. This might not be a problem for research purposes. However, when trying to build a product that can be sold, the user will expect results right away. To overcome this issue, this study also proposes to build a generic model that can predict the consumption of any household with good enough precision and no training. This generic model was trained using several house's historical data. On a real world scenario, when prediction future house consumption of a household, this model is meant to be used in the early stages of its deployment, while not enough data has been collected to train a machine learning model specifically for that household. Once the performance of that specific model surpasses the generic model, the former replaces the later in forecasting household energy consumption.

This chapter is divided into two parts. Firstly, a preliminary study comparing different machine learning techniques is employed. The preliminary study utilizes two datasets, both encompassing information about the consumption of one household. The objective is to select the best suited machine learning model for this purpose as well as the optimal parameters. Secondly, utilizing a dataset with multiple households, a generic model is built. The generic model was utilized in the early stages of the final system, while not enough information about the specific house was collected to train the specific model.

3.1 ML Techniques

According to the previous exploration on the techniques used to forecast electrical loads that was carried out in the previous chapter, Support Vector Regression (SVR) and Long Short Term Memory (LSTM) are the most commonly used techniques to solve this problem.

Support Vector Machines (SVM) [31] is a supervised machine learning technique used for classification problems. SVM utilizes a kernel and an optimizer algorithm. The kernel transforms the data, increasing its dimensionality. This aims at transforming non-linear data into a high dimensional space where data can be linearly separable. The optimizer finds the hyper plane (decision boundary) that separates data. SVR is version of SVM that is utilized for regression problems. Here, the model tries to find the best fit line by reducing the error between observed and predicted values. The model was implemented with the scikit-learn python library [32]. The chosen parameters were set to default, which utilizes the Radial-Basis Function (RBF) kernel, which is used for non-linear problems.

Neural Networks (NN) or Artificial Neural Networks (ANN) are a machine learning technique that mimics the behavior of the human brain [33]. This model is used to model complex patterns by passing the data through layers of neurons. Neurons receive features from the training set and apply an activation function, the result is outputted to another neuron and so on. The output of the neural network represents the combined input of all neurons.

Recurrent Neural Networks (RNNs) implement feedback loops where information from prior inputs is used to influence the current output. Basically, the information a layer had in the previous step is remembered by a memory function. This method is widely used with sequential data.

However, it is difficult to train standard RNNs to solve problems that require learning long-term temporal dependencies. This occurs because the gradient of the loss function decays exponentially with time (vanishing gradient problem) [34]. Long Short Term Memory (LSTM) networks are a type of RNN that use special units which include a memory cell that can maintain information in memory for long periods of time, solving the vanishing gradient problem.

In this work, the selected architecture contained two LSTM layers. The first, is called the input layer and contains neurons equal to the amount of inputs (number of features * past window, where past window is the amount of historical records the model looks at in order to predict the next value). This is necessary since a different number of inputs was fed to the model during the different experiments. However, a maximum of 512 neurons was set, since the training time increases exponentially with the complexity of the network. The second layer is called a hidden layer. The number of units in this layer is usually between the size of the input and size of the output layers [35]. Therefore, the number of neurons was set to two thirds of those of the input layer. An output layer is also part of the architecture. This last layer provides the output of the model and, since it is only one value (energy consumption), only one neuron is required. Every layer neurons' are fully connected to the previous neurons. A visual representation is present in Fig. 3.1. The LSTM neural network was implemented using keras library [36].

The exploration of state of the art articles confirmed that LSTM is widely utilized to make predictions in time series problems such as the forecast of household consumption. However, some papers also suggested that SVR is a very competent technique that can event outperform LSTM [11, 12]. Studies also shown that models gained from additional features like weather and individual appliance consumption [10, 19].

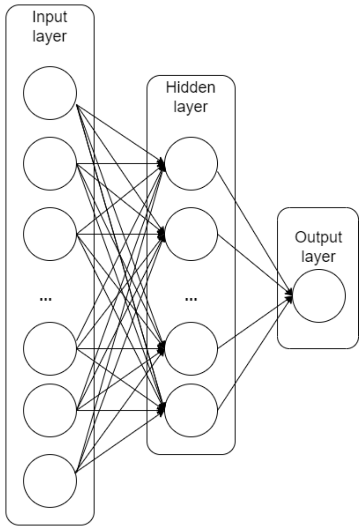


Figure 3.1: LSTM Neural Network Architecture

3.2 Preliminary Study on different ML Techniques

In this section, an early exploration of machine learning techniques was performed. The objective is to select the best ML technique to forecast energy consumption in households, as well as the best features to use. This experiment did not yet utilize the dataset that was used in the final solution, as one of the objectives of this particular experiment is to better understand how the different features influence the performance of the models (what with be impossible in the final dataset that contains several households but not many features).

3.2.1 Methodology

At this stage, two different datasets were analyzed, each containing the historic of energy usage of a household. The objective is to evaluate the presented models and find the best one to perform household load forecast. However, there are some parameters that require fine tuning to achieve the best performance from these models. Since the current problem is a time series problem, a past window needs to be defined. A past window is the amount of historical reads the model looks at in order to predict the next value. For this purpose, past window values of 1 hour, 24 hours (1 day) and 168 hours (1 week) were experimented and compared.

A process of feature selection utilizing XGBoost ranked the features regarding their importance (correlation with the target - consumption). This is done by utilizing boosting trees that fit to the data and evaluate how many times certain features are used to find the target value (the features that are used more times are the most important ones). The objective of feature selection is to remove redundant and irrelevant data. By reducing the number of features, overfitting becomes less likely since there is less opportunity for the model to make decisions based on noise. Since some features have a higher correlation with the target value and, therefore, are more valuable than others, this can improve the accuracy of the prediction by removing misleading data. Additionally, removing features reduces the training time. For this particular study, the 10 features with highest importance were selected to train the models and to evaluate if the algorithms gained from having a more refined set of features. For this reason, the several models are compared using all features, features from a feature selection process, and only the past consumption feature.

The experiment was initiated by splitting the dataset in two parts, separating the last month from the rest of the data. The first part was utilized for training of the machine learning model, and second was utilized for testing (one month is equivalent to about 10% of the whole dataset) . Both splits of the data were then scaled. Feature scaling is a method used to normalize the range of independent variables or features of data. Since features can have very different ranges of values, it is important to normalize this range so that each feature contributes approximately proportionately for the final result.

The training data was then converted into a time series format. This means that the dataset was organized in a way were the model would receive readings from the previous n time steps (with n being the past window) and learn to predict the next consumption value.

3.2.2 Metrics

In order to evaluate the performance of the resulting models several metrics were used, as described below.

MAPE: Mean Absolute Percentage Error represents accuracy as a ratio.

$$MAPE = \frac{100}{n} \sum_{j=1}^{n} \left| \frac{Actual_j - Forecasted_j}{Actual_j} \right| \tag{3.1}$$

where *Actual* represents the actual consumption value, *Forecasted* represents the predicted value and n corresponds to the number of observations.

MAE: Mean Absolute Error expresses the absolute error between observations and predictions.

$$MAE = \frac{\sum_{j=1}^{n} |Forecasted_j - Actual_j|}{n} \tag{3.2}$$

where *Actual* represents the actual consumption value, *Forecasted* represents the predicted value and n corresponds to the number of observations.

MaxAE: Maximum Absolute Error represents the maximum error between one observation and one prediction across all observations.

$$MaxAE = max(|Forecasted_j - Actual_j|) \tag{3.3}$$

where *Actual* represents the actual consumption value, *Forecasted* represents the predicted value and n corresponds to the number of observations.

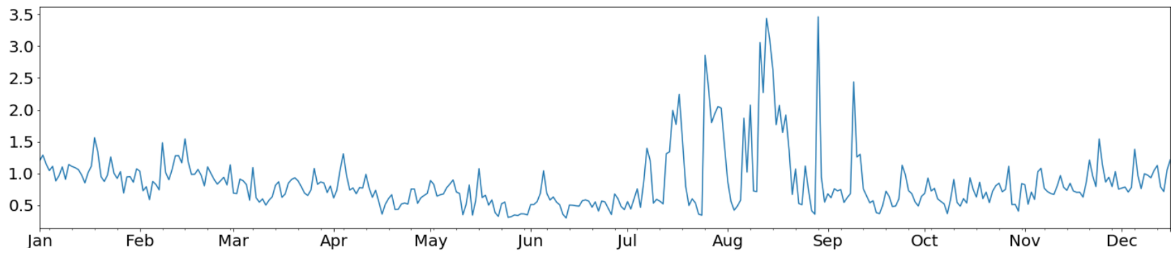


Figure 3.2: Energy Consumption - DataSet 1

3.2.3 DataSet 1

Dataset 1 [37] contains consumption readings of house appliances in kW with a time step of 1 minute across 350 days, starting on January 2016 and including weather conditions of that particular region. The specific region was not specified. The dataset contains a total of 25 raw features that encompass time, weather and electrical consumption (total and per appliance). The consumption behavior over time is depicted in Fig. 3.2. As it can be seen, the represented line is somewhat unpredictable, exhibiting many peaks and lows. The dataset feature importance ranking can be seen in Fig. 3.3.

Results

The test results for the LSTM neural network are presented in Table 3.1. The best performing LSTMs in terms of MAPE benefited from using only 1 past window (Fig. 3.4). This means that only the last value of each feature is used to predict the next result.

Past Window	Features	MAPE	MAE	MaxAE
1	All Features	0.27	0.22	2.14
1	Feature Selection	0.28	0.22	2.1
168	Only Consumption	0.28	0.22	1.99
1	Only Consumption	0.29	0.22	1.89
24	Feature Selection	0.29	0.23	2.49
168	All Features	0.29	0.25	2.53
24	Only Consumption	0.3	0.22	1.9
168	Feature Selection	0.32	0.23	2.59
24	All Features	0.45	0.3	2.13

Table 3.1: Dataset 1, LSTM results

The test results for SVR are presented in Table 3.2, showing that the model scored the best accuracy, achieving the lowest MAPE value (24%). Strangely, the best performing model did not improve from additional features in SVR. This may be due to the unpredictability of the energy consumption line in this case. The prediction line for the best model can be seen in Fig. 3.5.

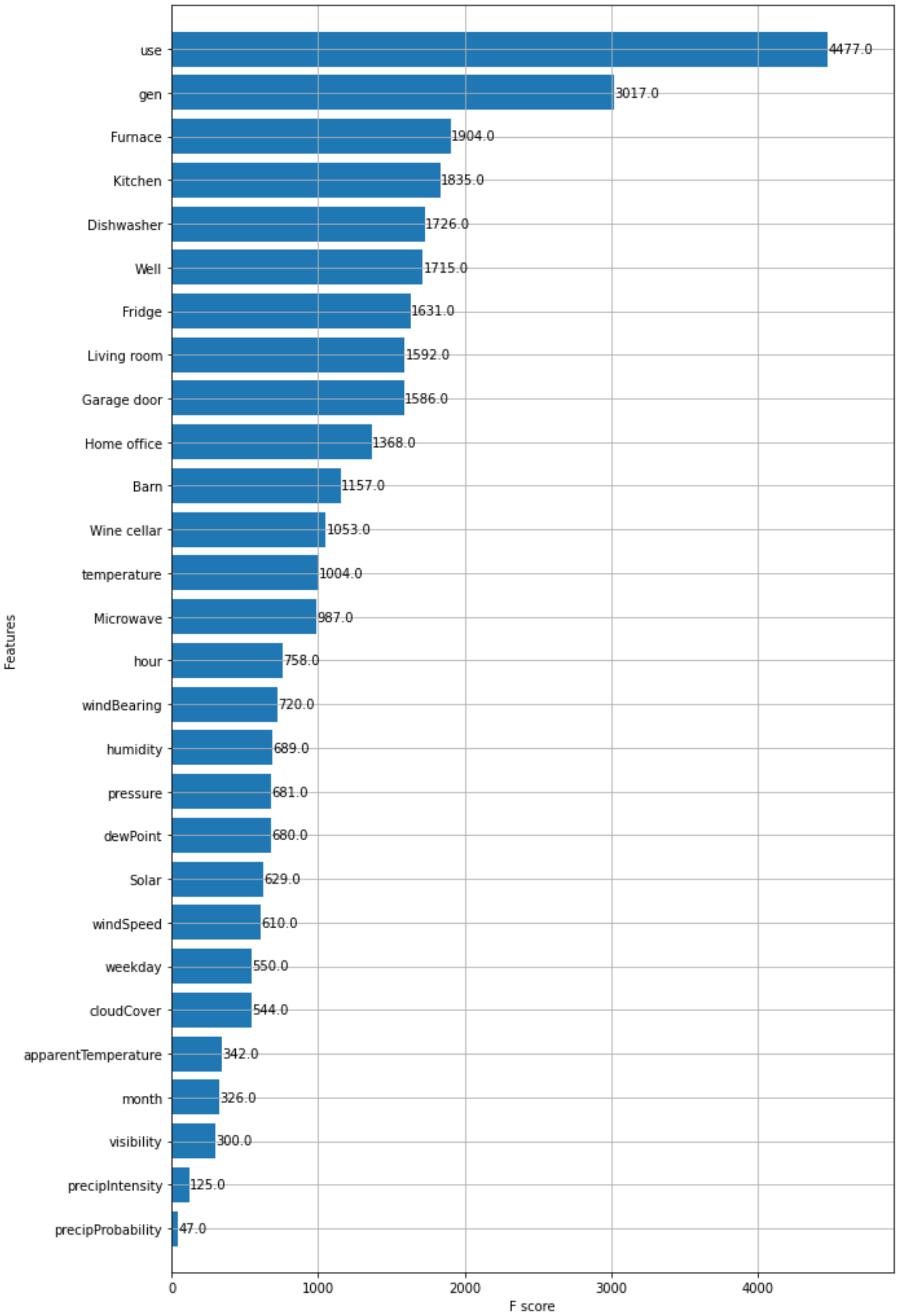


Figure 3.3: Feature Importance - DataSet 1

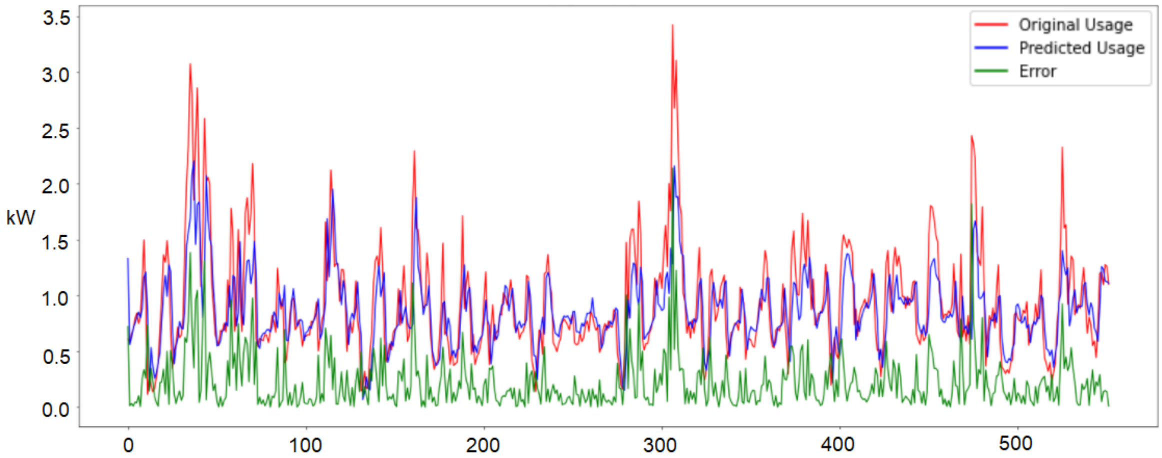


Figure 3.4: DataSet 1, LSTM 1 Past Window, All Features

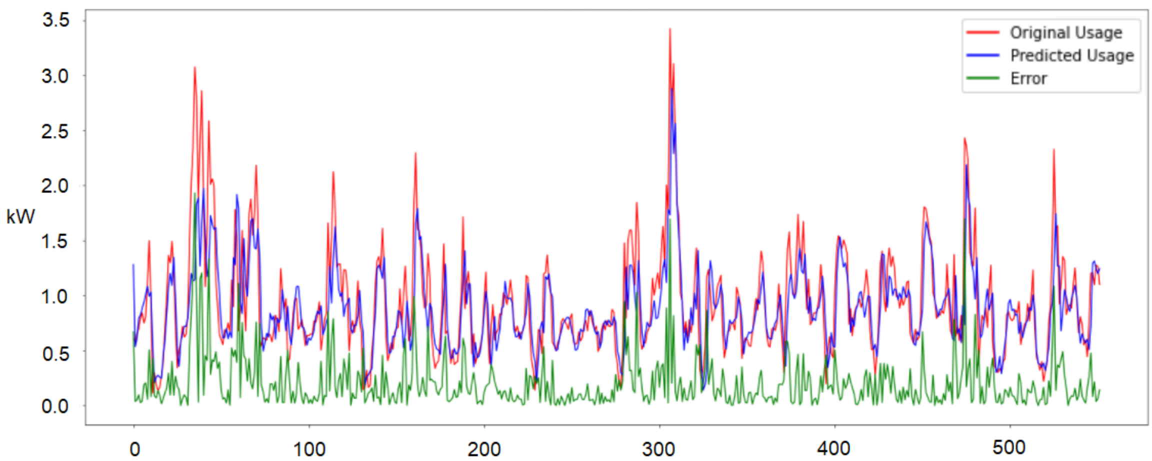


Figure 3.5: DataSet 1, SVR 24 Past Window, Only Consumption

Past Window	Features	MAPE	MAE	MaxAE
24	Only Consumption	0.24	0.21	1.93
1	Feature Selection	0.25	0.21	1.93
168	Only Consumption	0.26	0.21	2.18
24	Feature Selection	0.26	0.21	1.99
1	Only Consumption	0.27	0.22	1.96
1	All Features	0.29	0.24	2.05
168	Feature Selection	0.31	0.24	2.47
24	All Features	0.31	0.27	2.26
168	All Features	0.37	0.27	2.26

Table 3.2: Dataset 1, SVR results

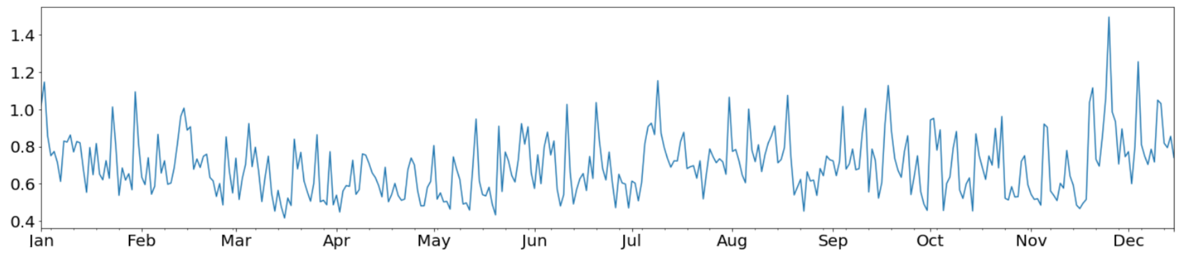


Figure 3.6: Energy Consumption - DataSet 2

3.2.4 DataSet 2

Dataset 2 [38] contains data relative to household energy consumption. The dataset contains reading with 30 minute intervals for appliance consumption and hourly interval for weather conditions. The dataset contains one year of data starting from January 2016. The dataset contains a total of 37 raw features that encompass time, weather and electrical consumption (total and per appliance). The consumption behavior of the house present in the dataset is available in Fig. 3.6. This dataset is not as volatile as the later but it is still difficult to trace a pattern. The ranking of feature importance can be seen in Fig. 3.7.

Results

The test results for the already defined LSTM neural network are presented in Table 3.3.

The best LSTM model in terms of MAPE utilized the last 24 hours of historical data to predict the next result, Fig. 3.8. This model also utilized the features from the feature selection process. The training time for this model was 160 seconds, while the more complex models took almost 30 minutes. The graph shows that the model could not follow the consumption peaks very well. Fig. 3.9, represents the predictions of the same model using only 1 hour of past window and all features, we can see that it follows the consumption curve in the high peaks, achieving the smallest maximum error. However, this model is much more volatile and ended up with the highest MAPE value.

The parameters: 168 past window and all features achieved the second best MAPE value and a decent maximum absolute error compared with the other models, Fig. 3.10. Even though this model is using a lot of features, the LSTM model managed to attribute importance

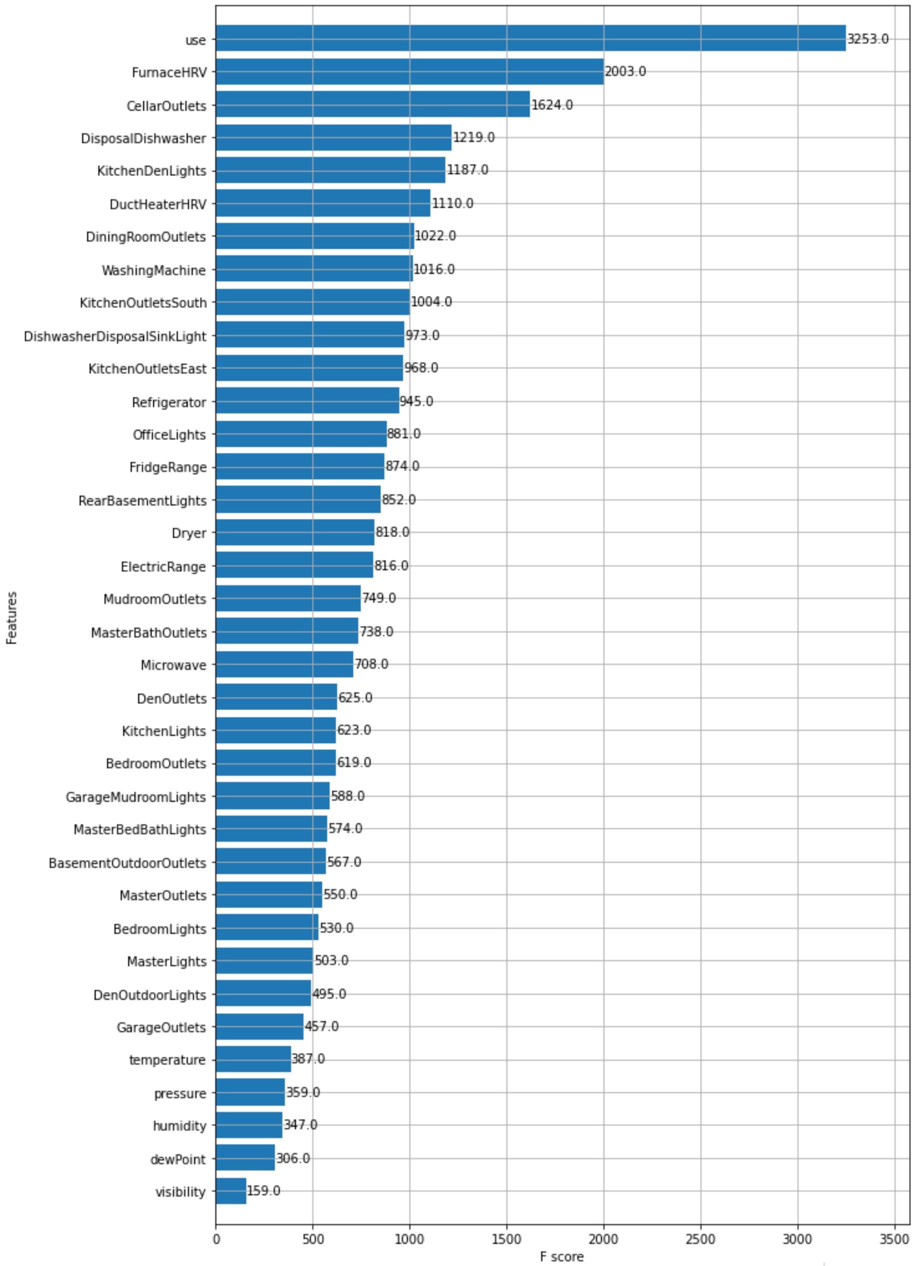


Figure 3.7: Feature Importance - DataSet 2

Past Window	Features	MAPE	MAE	MaxAE
24	Feature Selection	0.285	0.25	2.29
168	All Features	0.294	0.248	2
168	Feature Selection	0.295	0.27	2.41
1	Feature Selection	0.31	0.27	2.44
24	All Features	0.31	0.25	2.05
1	Only Consumption	0.32	0.27	2.41
24	Only Consumption	0.33	0.27	2.49
168	Only Consumption	0.33	0.28	2.45
1	All Features	0.34	0.27	1.76

Table 3.3: Dataset 2, LSTM results

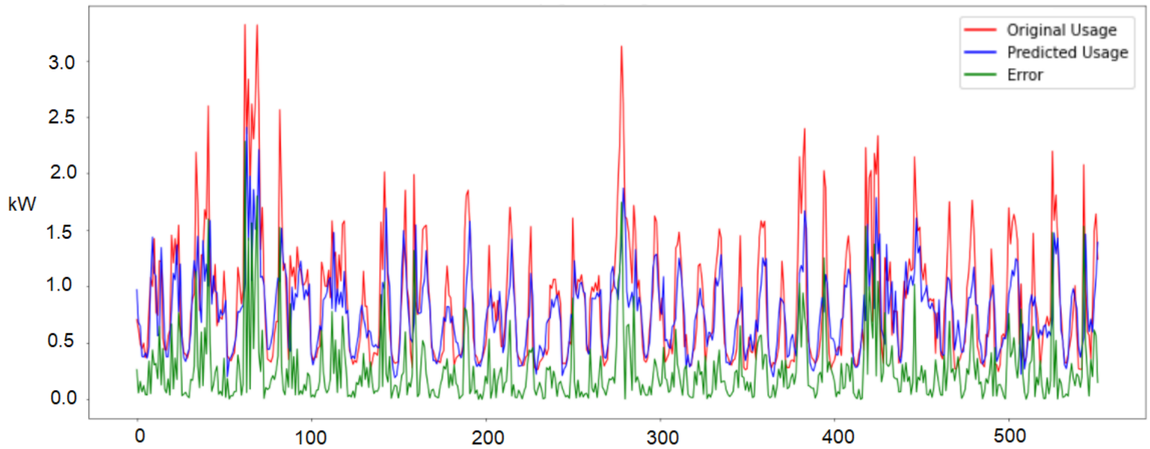


Figure 3.8: DataSet 2, LSTM 24 Past Window, Feature Selection

---

to the right features and still follow the consumption curve. Using many features brings some disadvantages. One of them is the amount of time needed to train the model. In this case the model took almost 15 minutes through train. Other models, however, can see its accuracy severely damaged, as it can be seen in the next SVR model.

The test results for the SVR model are presented in Table 3.4. The best SVR model in terms of MAPE, surpassing even the LSTM model, uses 24 hours of past window and the features from feature selection just like the best LSTM model. Even though this model scored the best MAPE, it has higher MAE and a high maximum absolute error, higher than all the LSTM models. The prediction curve can be seen in Fig. 3.11. The training time of this model was 6.3 seconds, while the maximum training time of all the SVRs was 103 seconds (168 past window, all features). As a side note, an example of what can happen when to many features are utilized we have the SVR model that used an entire week of past data and all of the available features, Fig. 3.12. The prediction curve achieved in this model is almost an average of the overall consumption.

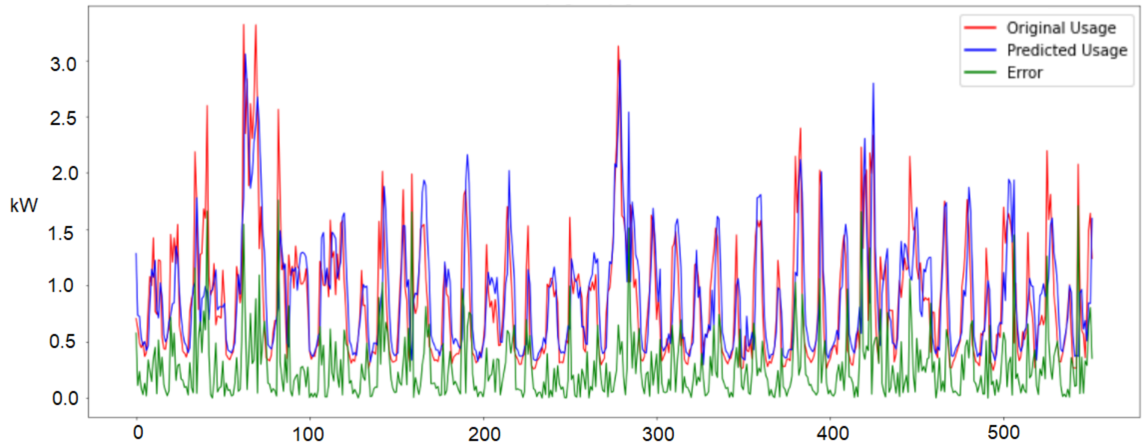


Figure 3.9: DataSet 2, LSTM 1 Past Window, All Features

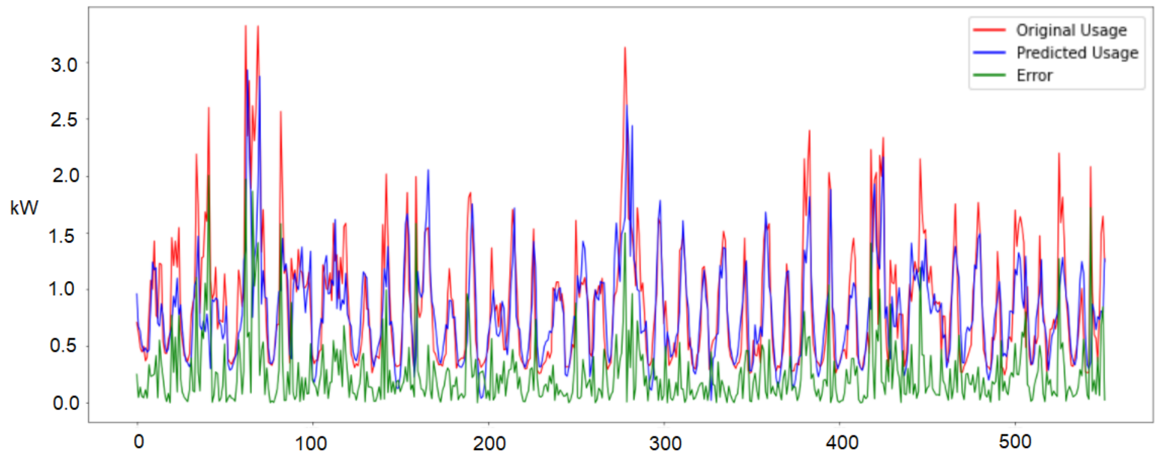


Figure 3.10: DataSet 2, LSTM 168 Past Window, All Features

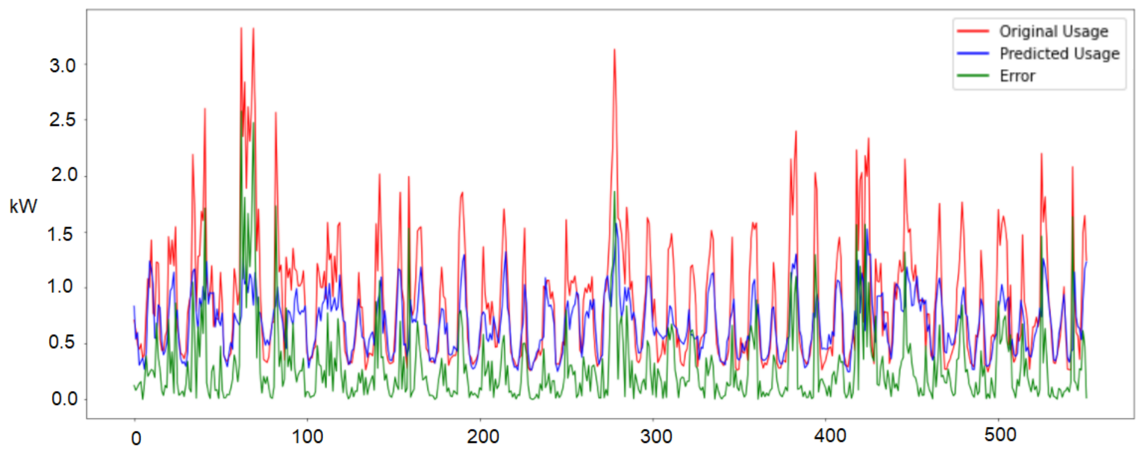


Figure 3.11: DataSet 2, SVR 24 Past Window, Feature Selection

Past Window	Features	MAPE	MAE	MaxAE
24	Feature Selection	0.279	0.26	2.55
1	Only Consumption	0.3	0.28	2.43
168	Feature Selection	0.32	0.309	2.65
1	Feature Selection	0.32	0.28	2.48
1	All Features	0.34	0.3	2.42
24	Only Consumption	0.34	0.3	2.61
168	Only Consumption	0.37	0.34	2.65
168	All Features	0.49	0.42	2.73
24	All Features	0.53	0.39	2.59

Table 3.4: Dataset 2, SVR results

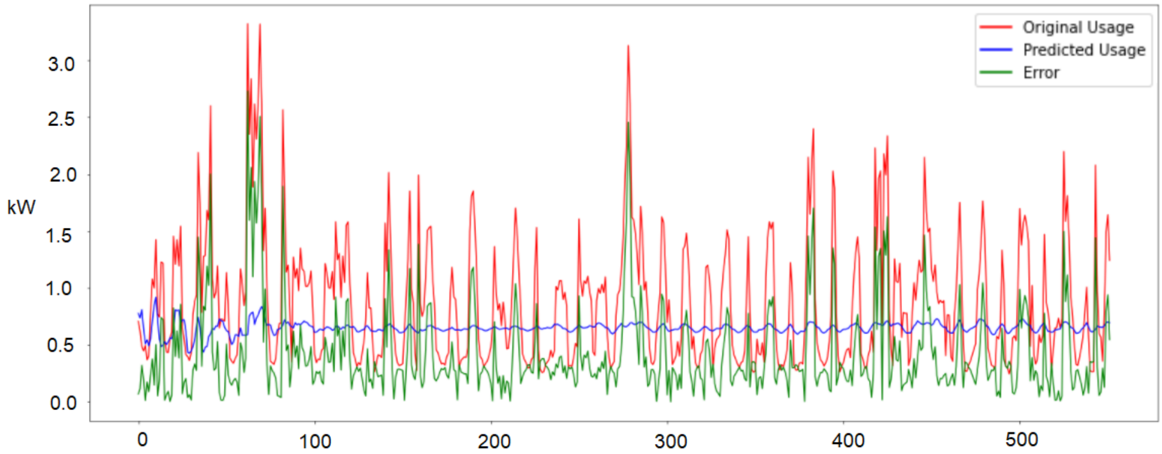


Figure 3.12: DataSet 2, SVR 168 Past Window, All Features

3.2.5 Conclusion

In both datasets both models were very close. SVR managed to get better results than the LSTM model in the first dataset, but on the second one, only the MAPE value was superior. The best model appears to be SVR with a past window of 24 hours, but the decision is not clear. Further research might be necessary, as the results appear to change with the dataset. The optimal past window appears to be 24 hours, also, in the second dataset, the model benefited from a feature selection process.

3.3 Generic Model

So far, models have been tested by splitting the data related to a single house into train and test sub-datasets. This means that the test results obtained so far were obtained when the model already had many samples, e.g., at least a few months of historical data. On a real world scenario, the user will expect results from a given machine learning model right away. Therefore, it cannot afford to wait for the model to have sufficient historical data before start providing adequate forecasts. At the same time, it should not be expected from a model with little to no training data to achieve acceptable results.

In this section, the option of having a generic model, which does not require the knowledge of the prior behavior of a specific house to predict future electrical consumption values, is presented. This model was trained with the load demand data of several houses, so that it can represent the consumption behavior of typical households. This model is utilized in the early stage of deployment, when not enough data has been collected to generate a specific model. Once the specific model outperforms the generic one in terms of its accuracy, the later is no longer used. The specific model was trained solely with the specific data pertaining to the household. Since a new dataset is being used in this experiment, a new test was performed comparing the LSTM and SVR models, to see if the past results are valid here, when it comes to training exclusively with one house and making predictions for the same house.

3.3.1 DataSet

The dataset [39] contains energy consumption readings for a sample of 5567 London Households that took part in the UK Power Networks led Low Carbon London project between November 2011 and February 2014. The dataset contains only information about the date and power readings. Firstly, the data was separated into multiple files, one for each house. This was necessary since several users were mixed in the same file and, often times, the same user had reading across multiple files. Also, since the original dataset contained reading with the time steps of half an hour, all the records were aggregated into one hour intervals. The dataset was also subject to a data treatment process, where we made sure the households with empty records would not be used. Additionally, this process also ensured that all the records were continuous and spaced with the same interval.

3.3.2 Metrics

The metrics used in this section were the same metrics used in the preliminary study, defined in Section 3.2.2 . However, this dataset contains a particularity which are very low, occasional, consumption values. MAPE calculates how different the prediction is, in percentage, relative to the real value. This means that a very small real value outcomes in a very large MAPE error, even if the predicted value is also small, but still proportionally large. With near zero values, the forecast can easily be several times greater than the actual value. This causes the MAPE value to increase to infinity, or to be undefined (in case the actual value is zero). To solve this problem a new metric is introduced [40].

WAPE: Weighted absolute percentage error equally penalizes for under-forecasting or over-forecasting, and doesn't favor either scenario.

$$WAPE = \frac{\sum_{j=1}^{n} |Actual_j - Forecasted_j|}{\sum_{j=1}^{n} |Actual_j|} \tag{3.4}$$

where *Actual* represents the actual consumption value, *Forecasted* represents the predicted value and n corresponds to the number of observations.

3.3.3 Specific Model

The specific model was trained only with the load demand data of a single household. The model was then used to forecast the power demand of that particular household. Since a new dataset is being used in this experiment, a new test was performed comparing the LSTM and SVR models, when it comes to training exclusively with from one house and making predictions for the same house. This time however, since a lot less features are available, the number of inputs fed to the model did not change. Also the assumed window of past values is 24, since it was found to be the best value in the previous study. Hence, the models received the data from the last 24 hours to make a prediction. The models were also compared with a baseline model. The baseline prediction is the same as the last observed value (current value).

Results in Table 3.5 show that the SVR algorithm achieved a better performance in terms of MAPE , WAPE and MAE. Interestingly, the LSTM model achieved better maximum absolute error in almost all the test houses. Since in the previous tests, performed with different datasets, SVR achieved better performance and this test does not indicate different results, SVR was used has the specific model. This time, the baseline, which predicts the last observed value, scored far worst results, the only exception being the maximum absolute error of the first test house Table 3.6. This means that the ML models are more valuable in this dataset, comparing with the datasets from the last section.

House	SVR				LSTM			
	MAPE	WAPE	MAE	MaxAE	MAPE	WAPE	MAE	MaxAE
MAC000205	0.36	0.38	0.19	3.18	0.51	0.44	0.23	3.02
MAC002086	0.20	0.21	0.04	0.81	0.24	0.24	0.05	0.77
MAC000387	0.30	0.35	0.17	2.38	0.36	0.36	0.17	2.47
MAC000276	0.24	0.23	0.15	1.17	0.27	0.24	0.16	0.96
MAC001251	0.4	0.4	0.28	2.58	0.5	0.42	0.3	2.36

Table 3.5: SVR vs LSTM performance in 5 test Houses

House	BaseLine			
	MAPE	WAPE	MAE	MaxAE
MAC000205	0.48	0.5	0.24	2.44
MAC002086	0.31	0.34	0.07	1.04
MAC000387	0.46	0.43	0.22	2.5
MAC000276	0.27	0.27	0.18	1.24
MAC001251	0.5	0.47	0.32	2.81

Table 3.6: Baseline performance in 5 test Houses

3.3.4 Generic Model

As discussed, the objective of the generic model is to predict the energy consumption while historical data is still being collected to feed a ML model trained with data from that particular household.

Previous research has shown that SVR is the best model when it comes predicting household energy usage on one specific house. The SVR model in question is, however, implemented with the *sklearn* library and does not support incremental learning. This means that all the training must be performed in one go, with all the training data stored in memory. Hence, the model is not be able to continue training incrementally. Additionally, the SVR model is quite slow for larger amounts of data, making it impractical to build this generic model. Since the previously tested LSTM model, implemented using the keras library, performed decently and does support incremental learning, it was chosen to be the base for the generic model, which is to be trained with several households.

Initially, 500 users where selected at random to be the training pool for the generic model. However, the training process was shown to be very time consuming, with the model taking about 10 minutes per house. Therefore, only 50 households were used to train this generic model. This decision does not seriously harm the model since in further testing it was found that more train data beyond a certain point does not necessarily improve the performance of the model. However, since the model supports incremental learning, it is possible to keep training the model. This feature was also used to keep training an instance of the generic model with the same data that was used for the specific model. This way, while the specific model is being trained, the generic model can also improve its results for a specific household. The generic model used the last 24 hours of consumption, hour and weekday values as features, similarly to the specific model.

The training process requires feature scaling. This process aims at centering at value zero and helps the machine learning model achieving better results. For this purpose the *sklearn* library was used to instantiate a standard scaler. This scaler works with the mean and standard deviation of each feature and all the training dataset should be considered to retrieve these values. Therefore all the 500 houses were used to build this scaler even though some of the houses that contributed to the scaler did not actually train the model.

3.3.5 Generic Model and Specific Model Simulation

To emulate the behavior of both models, a simulation was performed utilizing 5 randomly selected households. This process iterated over the records of each house being tested, emulating the passage of time, and evaluating the performance of both models as more and more historical data is available.

Since the specific model was only trained with data from one specific user, and since that, at the beginning of the experiment there was no prior data from this user, the specific model is bound to not achieving good results at the start. In a real world scenario, it would be possible to retrain the specific model every hour, when a new record is created. However, for time reasons, the model was only retrained every week. This means that on the first week, only the generic model was available. On theory, with the passage of time, the specific model should outperform the generic model, since it is being trained for the specific user that is being tested.

For each hour, both specific and generic models made their predictions on the next energy consumption value. Each record was stored in a dataset that contains the observed values in the simulation. This dataset was used to train the specific model and to provide the latest records for the models to make predictions (since the models take into account the last 24 steps of data to predict the next value). Every week the specific model was retrained with the observed readings of that week. Since the generic model supports incremental learning, an instance of this model was also retrained with the same data. The objective is to understand if the generic model is good enough to be a substitute model in the early stages and, also, how long does it take for the specific model to start providing valuable predictions.

A product based on this approach harnesses the all-encompassing nature of the generic model and use its prediction in the early stages of the deployment. This model is, then, replaced by the specific model, that was trained especially for that household. This replacement is performed as soon as the system recognizes that the specific model is making less errors in its predictions than its generic counterpart. To attain this, the MAE metric was chosen, comparing the last 72 hours of predictions from both models. As soon as the specific model has a smaller MAE over the last three days, the generic model is replaced and is no longer used.

3.3.6 Results

House	Generic Model				Specific Model			
	MAPE	WAPE	MAE	MaxAE	MAPE	WAPE	MAE	MaxAE
MAC000032	1.95	0.44	0.25	4.69	1.83	0.39	0.22	4.64
MAC000091	0.37	0.37	0.1	2.38	0.24	0.28	0.08	2.56
MAC000112	0.45	0.44	0.25	3.59	0.3	0.25	0.2	3.45
MAC000258	0.46	0.36	0.12	2.43	0.39	0.32	0.11	2.3
MAC000283	0.42	0.37	0.11	2.11	0.3	0.3	0.09	2.07

Table 3.7: Generic model vs Specific model in 5 test Houses

House	BaseLine			
	MAPE	WAPE	MAE	MaxAE
MAC000032	2.32	0.68	0.36	6.24
MAC000091	0.28	0.33	0.09	2.08
MAC000112	0.4	0.45	0.24	4.19
MAC000258	0.71	0.71	0.24	3.04
MAC000283	0.49	0.47	0.14	2.96

Table 3.8: Baseline performance in 5 test Houses

The results presented in the Table 3.7 show that the specific model achieved better results in all the metrics, except in the second test house where the specific model ended up with higher maximum absolute error. Despite this, it is safe to say that the specific model has a better performance than the generic model, as expected.

It is worth mentioning that these metrics encompass the whole house dataset, which means that the first week (where all the specific model's predictions were zero) and the following days (where the specific model did not have much training) also count for these results.

To remove this factor, another test was performed with the same scenario, but this time only the predictions after 1 year were evaluated (about 6 months of data left). The purpose of this is to see the evolution of both models as more data from the specific household is fed to the models. The results show, for the first test house MAC000032, for the generic model, the WAPE value is 0.35 and for the specific model 0.28. This shows that with time, both models can improve their performances with the passage of time.

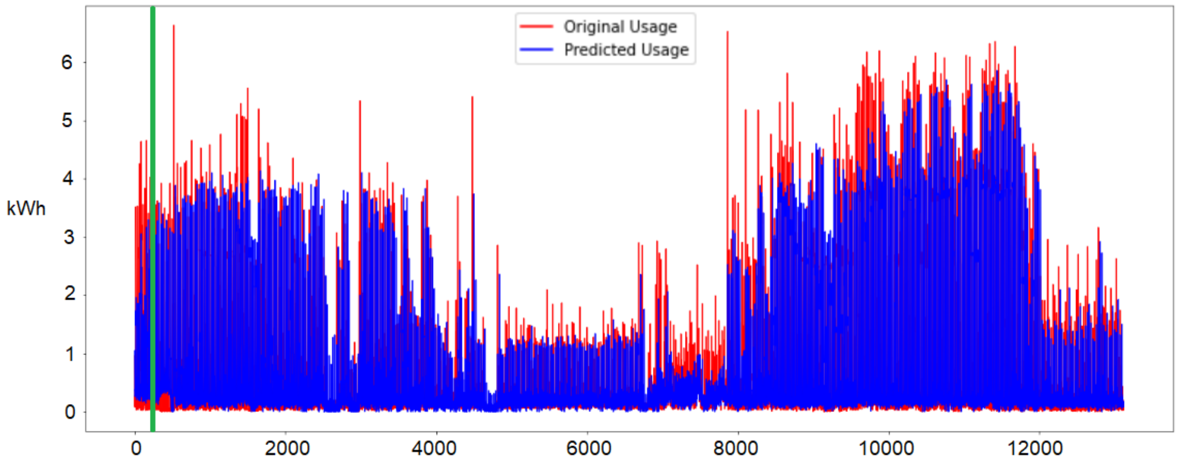
Additionally, the Table 3.8 shows the results obtained using the baseline approach. We can see that these results are, in their majority, worst than both the generic and specific models.

The Figures [3.13a, 3.13b, 3.13c, 3.13d, 3.13e] document the final model consumption curve for the five test houses. These graphs show how the final load forecasting mechanism behaved: utilizing the generic model in the beginning of the deployment and, later, after the green vertical line, replacing it with the specific model. This substitution is performed comparing the MAE values of both models on the last 72 hours. It is possible that the specific model outperforms the generic model temporally. However, it does not seem likely, unless in a extreme scenario, since all the tests indicate that the specific model will always eventually achieve better results. Therefore, the replacement of the forecasting model is permanent. On some test houses, this replacement occurs as soon as possible. For example, in the first test house [3.13a, the green line appears at hour 240, meaning that the generic model was used for 10 days (7 days while the specific model was only gathering data + 3 days of the system gathering error metrics for comparison). However in other houses, such as MAC000112 3.13c, this replacement only occurred after 533 hours (more than 22 days). This occurs because the specific model required more time to provide better predictions than the generic model.

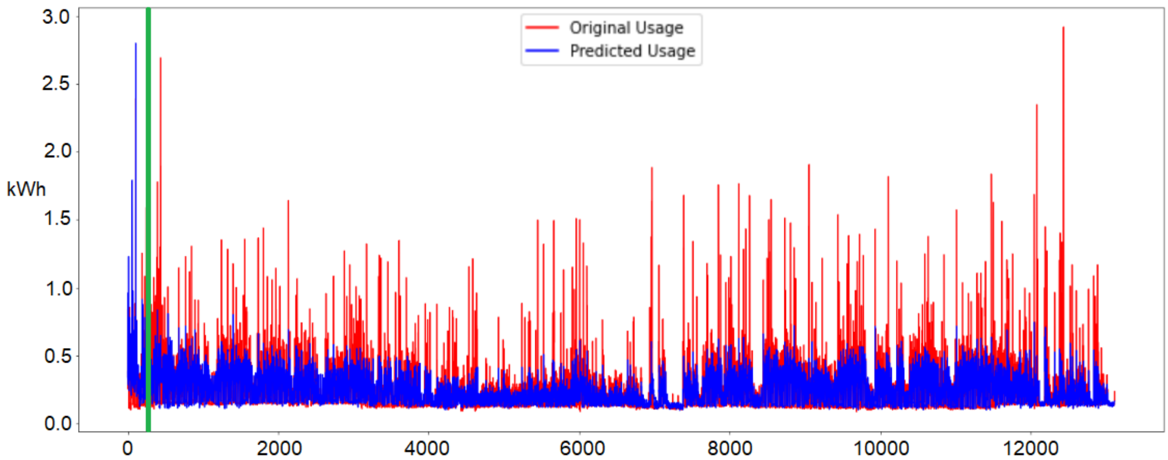
3.4 Conclusion

In this study, a household load demand forecast system is built. Initially, two different techniques (LSTM and SVR) were compared on a test house. In this initial study, the input parameters were tweaked to understand how they influenced the model behavior. It was found that the consumption values in the previous 24 hours were the most important to predict future load demand. Additionally, the models benefited from a process of feature selection. Utilizing a dataset with several houses, both techniques were briefly compared again. Results show that SVR achieved better results than LSTM except in terms of maximum absolute error, where in some cases LSTM was better.

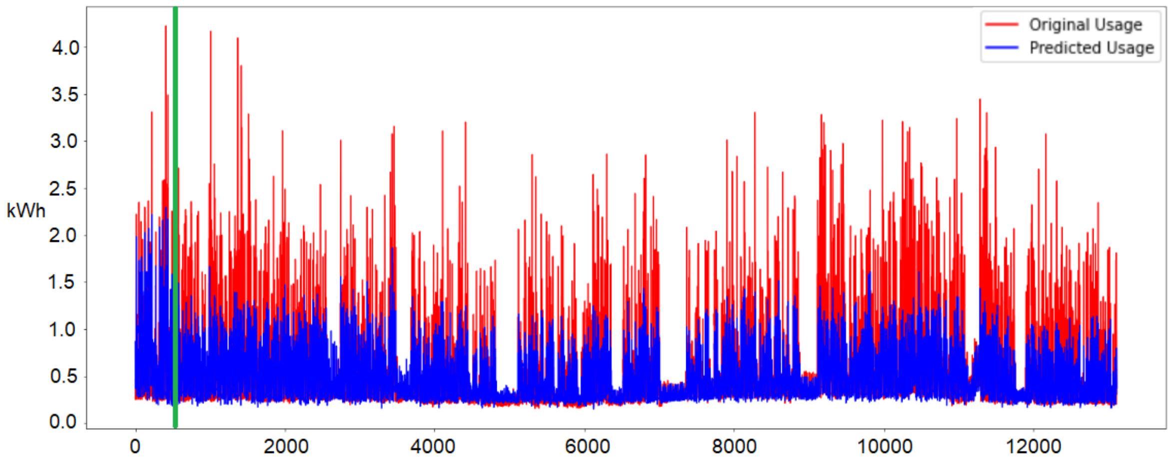
The final system makes use of both discussed ML techniques. One of them, LSTM, is used to build a generic model, trained with several houses in order to have a general concept of the average house behavior. Since the model is trained beforehand, it is used while not much information is known about a test house. The second technique, SVR, shown better results for training with information relative to only one house and predicting consumption records for the same house. Therefore, this model was used as the specific model, which is trained with data from a single test house. In a initial stage, as there is not much training data available, it is expected that this model to perform poorly. However, as time goes on, this model outperforms the generic model and thus, replaces it. The generic model predicted future house consumption values with acceptable precision, achieving better results than the baseline approach. Eventually, in all cases, the specific model outperformed the generic model. Once the specific model gets better than the generic model, the former replaces the later. In some cases this substitution was carried out very early (10 days) and in another case more than 22 days were necessary for the specific model to achieve better performance than the generic model.



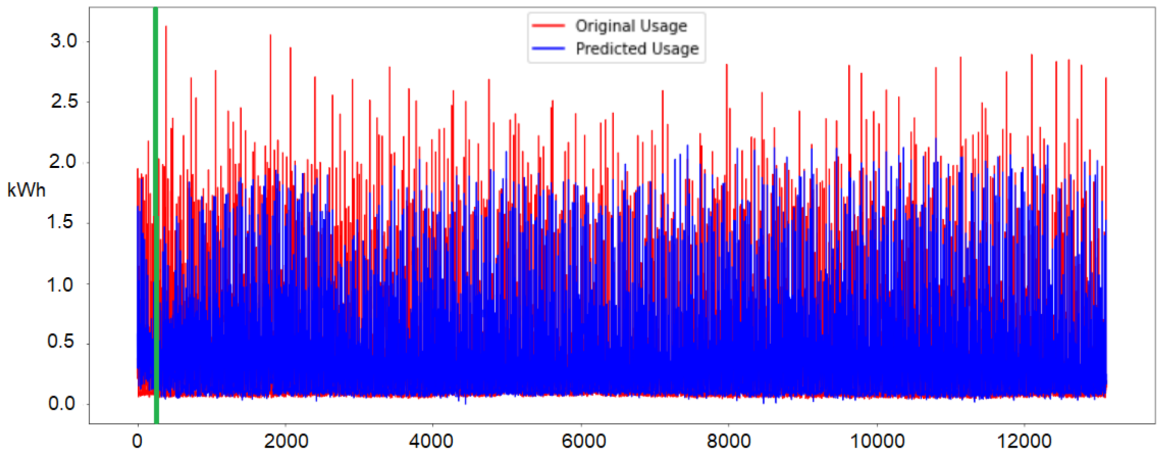
(a) MAC000032 Model Prediction



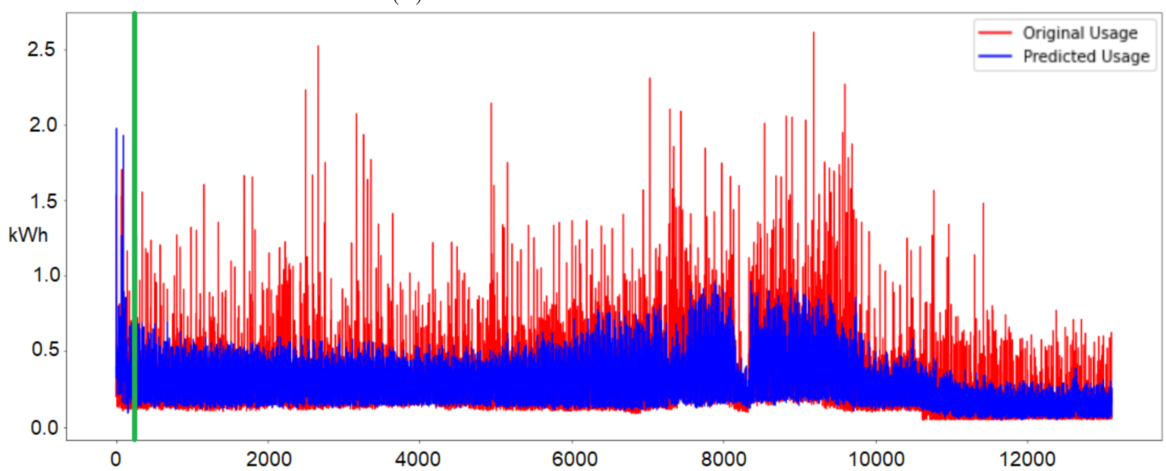
(b) MAC000091 Model Prediction



(c) MAC000112 Model Prediction



(d) MAC000258 Model Prediction



(e) MAC000283 Model Prediction

Figure 3.13: Final Prediction graphs of 5 Households

Chapter 4

Energy Management Decision Algorithm

In this chapter, a decision algorithm is built. The goal of the decision algorithm is to use household energy demand forecast, EV information and other inputs in order to minimize the electrical bill of a household. This algorithm is able to charge and discharge EVs and stationary batteries in the household, to suppress possible energy needs. This can be done using V2G chargers, a technology which enables bidirectional charging. This algorithm was tested in a simulated environment.

Since this chapter deals with energy quantities and charge rates, it is important to have a clear understanding of the kW and kWh metrics. Kilowatt (kW) is a measure of power. It reflects the rate of energy usage. Appliances normally have a fixed rate that they need in order to function. Kilowatt-hour (kWh), on the other hand, is a measure of quantity. It reflects an amount of energy used. This measures can be a bit difficult to understand since the the nomenclature is not direct. As an example, if an 1kW appliance is working for one hour, it will consume 1kWh. If the same appliance is working for two hours, it will consume 2 kWh.

In this chapter a bibliographic analysis focused on previous studies that tackle a similar problem to the current one is presented. Then, a description of the context and environment in which the study takes place is shown. A detailed analysis on the proposed algorithm behavior is presented in the next section. Additionally, the behavior of the simulation in which the algorithm will be tested is also looked at. Finally, the results obtained for the designed algorithm are presented and discussed.

4.1 Bibliographic research

In this sub-section a short state of the art research is presented regarding decision algorithms and home energy management systems.

In [41] a machine learning approach is utilized to forecast vehicle availability in order to provide vehicle to home (V2H) service. Since the objective of the study was to reduce the user electrical bill, the charging strategy was optimized based on the vehicle availability forecast, by determining the optimal charging–discharging schedule based on the electricity cost at each period. The EV availability forecast was obtained using Light Gradient Boosted Machine (LightGBM). The authors claim the total electricity cost decreased to approximately half. However, this study only dealt with EVs and not with other variables such as energy production.

The work in [42] proposes a multi-agent approach for residential demand response, utilizing load forecasting. The objective is to have the electrical devices controlled by a reinforcement learning (RL) agent that, using information on the current electrical load and the load forecast for the next 24 hours, can meet the device electrical needs while staying within the transformer limits. In that study, only 9 devices are considered, more specifically, 9 EVs covered by a single transformer. Each EV implements three policies. One of them ensures that a vehicle achieves the desired minimum battery charge, the second ensures that the overall load of the transformer stays within the predefined limits, the last uses the load prediction for the next 24 hours and makes it so that the vehicles are charged during the lowest load periods. Results show that RL is suitable for residential demand response. Agents mainly waited for low load periods to charge the EV but still utilized small periods of high load when necessary. The study main objective was to utilize coordinated charging to avoid overloading the grid and not to minimize the electrical costs of the user. Additionally, the study only dealt with electrical vehicles, ignoring other variables in the household.

The study in [43] presents a more conventional HEMS (Home Energy Management System) architecture. However, the existing systems mainly acts at the household appliance level, unlike this work, that intends to deal with energy transactions between the household and other elements, such as EVs. One other example is the work in [44] which performs appliance scheduling based on priority levels. Once again, that work deals only with solar power and grid power, leaving the EVs behind. In [45] proposes two electric energy management systems in the context of a grid-connected residential neighborhood with electric vehicles, battery storage, and solar photo- voltaic (PV) generation. On the other hand, the system being developed in this study is meant to work in a more contained environment (a single household), which differentiates this study from other similar studies that have deal with similar variables but encompass several houses of a neighborhood as the environment.

4.2 Context

The decision algorithm is meant to minimize the electrical bill of a smart house. The scenario in which this algorithm would be used requires a domestic building that is capable of producing energy. The inhabitants of said household would also possess EVs that would eventually be connected to the smart house. As already mentioned, V2G chargers enable the algorithm to view these EVs as a battery, since they can be charged later discharged, for example to give back energy for the house in periods of high demand. The only particularity is the unreliable availability of these vehicles. The house in question would also possess one or more stationary batteries, which can store and provide energy, with the advantage of being always available. This household, like any another house, would be in constant need of energy. Energy consumed by artificial lights, home appliances, computers and so on, is viewed generically as home energy needs / energy consumption. This electrical system is also connected to the distribution grid, that provides payed energy, or receives excesses of energy whenever required. A sketch of this environment can be viewed in Fig. 4.1.

The goal is to build a smart agent that receives several inputs such as the connected EVs, household energy demand forecast, energy production and so on, and take action in order to reduce the electrical bill of the house. The actions this agent can perform are basically

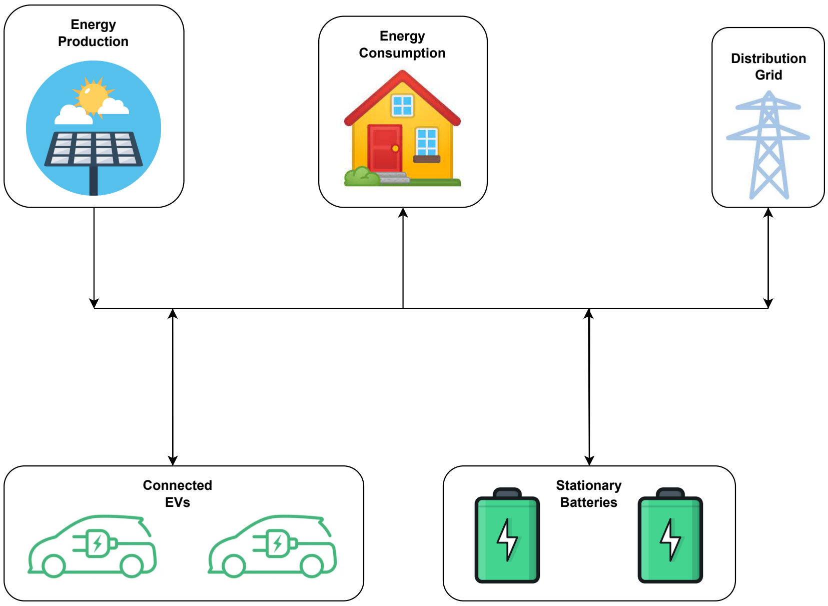


Figure 4.1: Energy System

exchanges of energy between the available elements. The algorithm would be run periodically and, each time, decide what batteries to charge or discharge in order to satisfy the needs of the house as well of the EVs.

4.2.1 Components and Algorithm Inputs

In this sub-section a more detailed description of what are the components that make up this environment and that the proposed algorithm has to deal with.

Household Energy Consumption Meter and Prediction

One of the inputs the decision algorithm receives is a prediction of the energy consumption of the household for the next hour. This prediction is useful since it can establish a concrete value of the household needs as well as give an insight on the ratio consumption versus estimated production. This way, the agent can have an idea whether the household needs energy or have an excess of it in the next hour. In this study, house consumption is viewed as a whole, so, it is only relevant to know that energy is being consumed and not which appliances are consuming it.

The model that was utilized for this is the same that was built in this study and its description is available in the chapter of Household Energy Consumption. Basically, a generic model is used in the early stages of the experiment to predict household consumption, while another model is being trained with the specific household consumption records. Since the model needs the consumption values in order to train the ML model and predict the next values, a consumption meter is also necessary. This meter would provide the total energy spent on the household in each time step (in kWh).

Energy Production Meter

Energy production can be performed in several ways. For simplicity reasons, in this work, the only form of energy production the household is capable of is through solar panels.

Production readings can be obtained in a number of ways. Other than a physical meter, production values in kWh can be obtained utilizing a mathematical formula from the solar irradiance value. Solar irradiance is the power per unit area received from the sun. The formula is the following [46]:

$$E = A * r * I * p \tag{4.1}$$

where **E** is the energy produced in KWh, A is the area of the solar panel, r is the yield of the solar panel given by the ratio: electrical power (in kWp) of one solar panel divided by the area of one panel (most solar panels are between 15 percent and 20 percent efficient [47]), I is the solar irradiance, and p is the overall solar panel performance including all losses such as orientation and aging (75 percent is a good value).

The irradiance values can be obtained from a meter, or from an API such as [48]. This is a free API that provides DNI (Direct Normal Irradiance) readings in real time and also forecasts these values in the near future. This is available in the endpoint: `https://api.solcast.com.au/world_radiation/forecasts`.

Stationary Batteries

One of the goals of this study is to successfully take advantage of mobile batteries such as EVs. However, that is not the only kind of battery the algorithm is going to have to deal with. Stationary batteries are batteries that are connected to the house all the time and can be used whenever the decision algorithm deems it. This kind of battery is actually much more useful than the battery of an EV, since its availability is certain. Batteries have information that is relevant for the algorithm to make its decisions:

- Maximum Capacity. The total amount of energy the battery can hold in kWh.
- State of Charge (SoC). The current amount of energy the battery is holding, between 0 and 1 (0 means empty and 1 means full).
- Charge rate. The amount of energy that can be charged/discharged per unit of time, in kW.
- Charge cost. Batteries have battery cycles, this is basically a way of measuring how much a battery can be utilized before its performance starts to deteriorate.

Electric Vehicles

One for the fundamental elements in this system are electric vehicles. V2G chargers allow bidirectional charging which enables the proposed energy management system to utilize energy stored in the vehicles to suppress energy needs of the household.

EVs are, non the less, still vehicles. This means that the availability of their batteries is uncertain. Also, unlike stationary batteries, the intelligent system cannot simply remove all the energy in the EV since it would impair the daily schedule of the user. The variables that were found to be of use for the system are:

- Battery related variables (such as battery capacity, SoC, charge rate, etc). It is essential to know how much energy the EV is currently holding, as well as how fast can it charge.
- Energy needed for the next trip. This would be equivalent to how many litters of gas a normal car would need for the next trip. This would work as a minimum threshold for the EV Soc. This variable was expressed in a value between 0 and 1 (as was the SoC), with 1 meaning: the user needs the EV completely filled for the next trip. Basically, when the EV is leaving, it must contain more than the value of energy that was predicted to be needed for the next trip.
- EV departure time. This is relative to the time at which the EV is going to leave. Basically, it is a deadline to fulfill the requirements in the previous item.

It is worth mentioning that, initially, the EV arrival time was also considered important since it could be used to predict if there would be a new battery needing or dispensing energy. However, this wasn't found to be of much use since it would not be possible to know if the vehicle might have charged while it was away, and if it would need energy or have an excess of it, making it pointless to know when an EV would arrive.

An attempt to make a model that could predict the energy needed for the next trip and the departure time (as well as arrival time) was done. Unfortunately, the model could not get good results, in part due to the lack of correct data there was to train a proper model. Datasets containing the information relative to vehicle departure times and energy usage per trip were very rare, furthermore, the ones that were found contained a big number of obvious outliers and records that did not match (such as vehicles leaving at random times, EVs spending the entirety of their fuel on a 30 minutes trip, etc). The responsibility of dealing with the battery of an EV should not fall upon a ML model that cannot predict the EV behavior with an acceptable error. The EV behavior can be unpredictable or inconsistent, emergency trips might be necessary and the amount of energy the EV is left with should be decided by the user. To solve this, on a real world scenario, both departure time and necessary energy could be asked to the user through a mobile application.

Grid

Every house is connected to the distribution grid. This grid is what provides energy to the common houses. Unlike the energy being produced in the solar panels, the energy obtained from the grid has a price. In this project, the decision algorithm tries to minimize the electricity bill and, to do that, it needs to minimize the energy obtained from the grid. However, since even when the energy production is low or when the available batteries are empty, the energy demand in the household and in the connected EVs does not stop. In this scenario the grid acts as an infinite source of energy that, obviously, has a price.

4.3 Decision Algorithm

In reality, taking the energy from one place and moving it to another is not what happens in a electric system. The only actions available are charging or discharging batteries, either stationary or from EVs. When a battery is charging, for example at 2 KW, it will charge 2 KW regardless from where that energy might have originated from. However, by discharging a battery with a certain rate and charging another battery with the same rate, given that the two batteries are the only batteries charging or discharging, the energy flows from one battery to another. The way the algorithm works is by finding the amount of energy that will be consumed in the next time step, and making sure that energy is going to be available from somewhere, preferably without using the grid. So in reality, the problem can be viewed as taking energy from one place and taking it to another.

For each module that can receive or give energy, the decision algorithm decides how much energy it should give or receive. Every time the decision algorithm is called to make decisions, it creates two lists. One of them contains the elements that can receive energy and the other contains the elements that can dispense energy, both lists are organized by a priority value.

Priority levels represent how much something "wants" to charge or discharge. For example, the work in [44] addresses an approximate subject and utilizes a similar principle. There are five levels of priority for charging and discharging, and for the two lists the priority levels are inverted. Basically, 0 is the level with more priority and 4 the level with less priority for discharging something. However, for charging priority, 4 is the level with more priority and 0 the level with less priority. An item with D discharge priority only gives energy to an item with C charge priority if D <= C, as can be seen in Fig. 4.2. The only purpose of having the priority values reversed is to facilitate the decision rule.

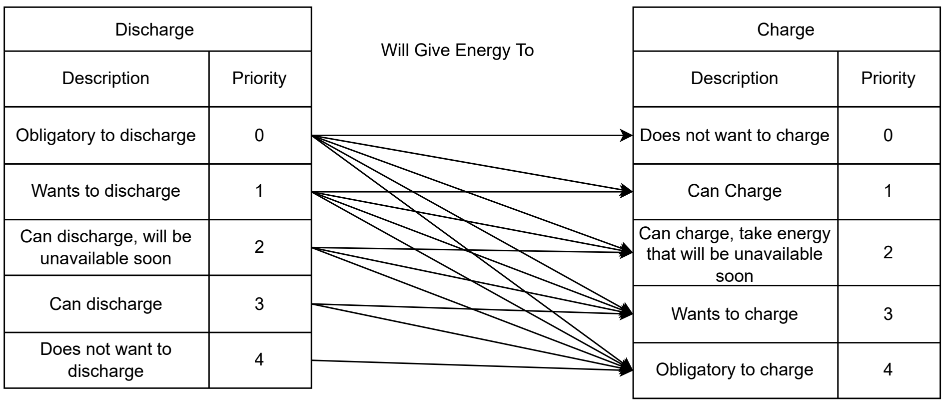
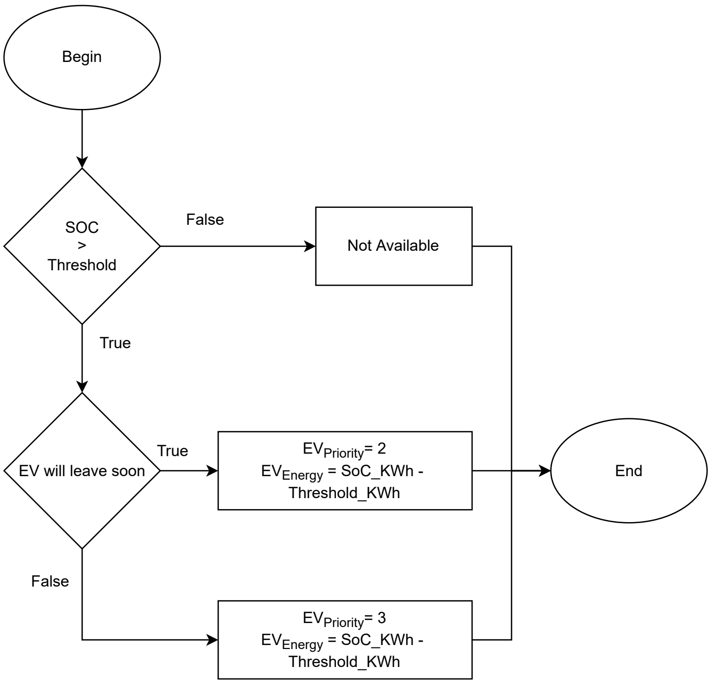


Figure 4.2: Decision algorithm Priorities

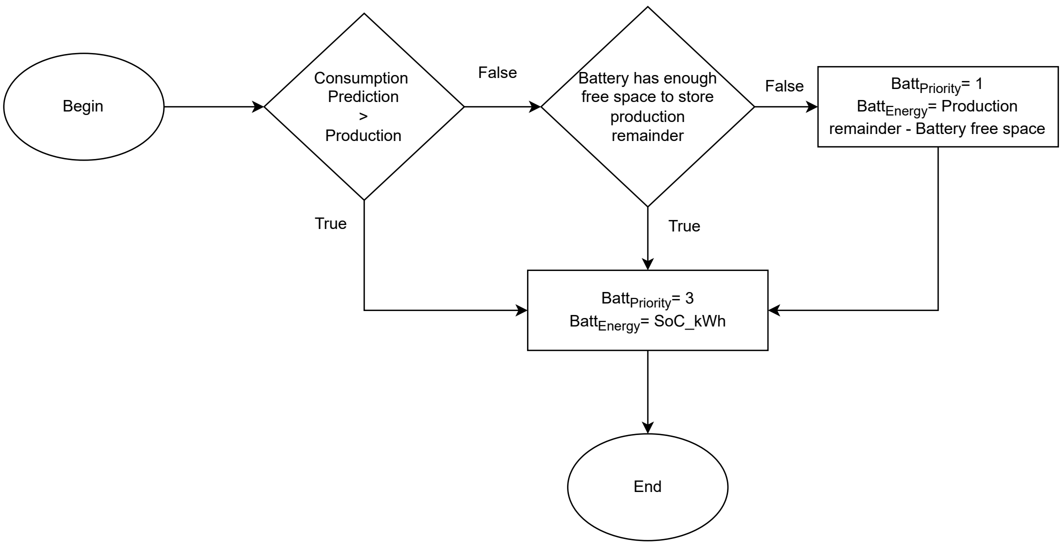
As already mentioned the decision algorithm creates a list of the elements that can provide energy and the elements that can receive energy in the next time step. The elements in these two lists are only eligible to charge or discharge. Being in the list does not mean that the object was necessarily charge or discharge. In fact, the same EV or battery can likely appear in both lists, since they are eligible to charge or discharge (a battery that is not full can charge or discharge). Whether they were chosen to do one thing or the other depends on the rest of the environment. Additionally, one object can appear in the same list several times, as it may require to receive different amounts of energy with different priorities, or be able to provide energy with different priorities (an EV that has a very low SoC wants to charge with high priority until it meets the minimum requirements but can continue charging after that). It is also insured that the components are treated by order of priority, for example: the available item with highest priority for discharging (in this case the smaller number) is always exhausted before the next item. The algorithm also decides the amount of energy that is charged or discharged, according to the situation, taking the necessary precautions and ensuring that the maximum charge rate and capacity of the batteries is not ignored and that possible battery loss during the charging and discharging actions are accounted for.

The decision making algorithm was implemented in a Python script.

4.3.1 Elements Eligible to Discharge

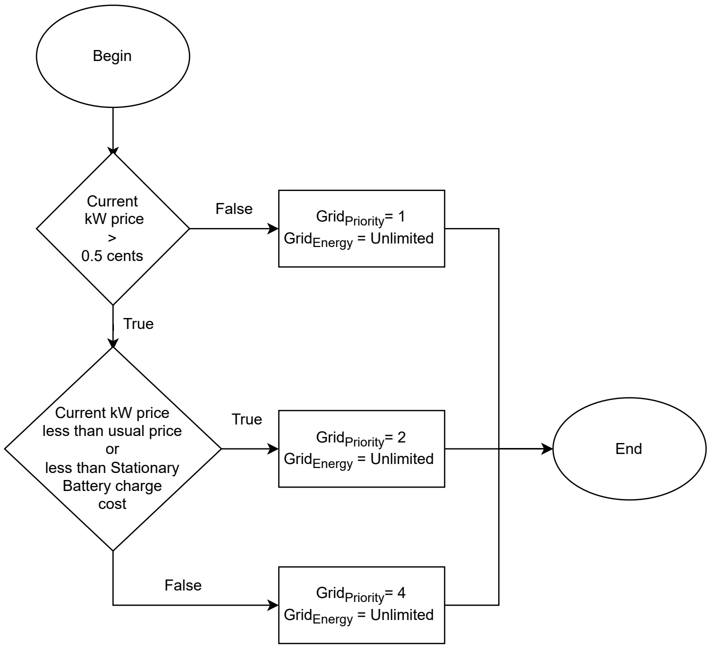


(a) Decision algorithm discharge priority flow chart - EV



(b) Decision algorithm discharge priority flow chart - Stationary Battery

In terms of elements eligible to perform discharge action, the level 0 is the level with higher priority. Elements with this level always provide energy. The only element that receives this score are the solar panels, as the energy being produced needs to flow somewhere immediately, even if it means throwing it away.



(c) Decision algorithm discharge priority flow chart - Grid

Figure 4.3: Decision Algorithm flow charts on deciding the priority of the elements to perform a discharging action

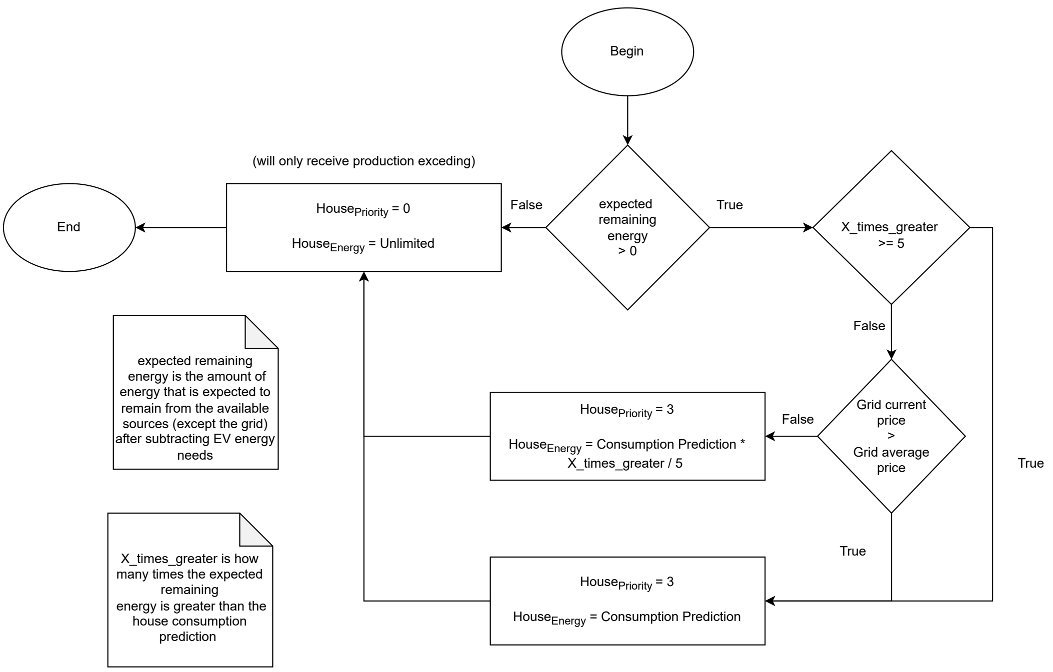
---

As already mentioned, electrical vehicles have a user defined parameter that defines the minimum value of charge the EV must hold. If the current SoC is below this level, the EV is considered unavailable to provide energy. On the other hand, if the EV contains more energy, it is available to provide energy to the house or to other batteries. In case the EV would be parked for a long time, it receives a priority of 3, meaning that it can provide energy if necessary (to suppress household needs, or to charge an EV that has very low SoC, as will be explained next). If the EV would leave soon, it receives a priority of 2. This means that EVs that were leaving soon, provided energy before EVs that were staying parked. This is because the EV is seen as a source of energy that would soon be unavailable. Also, priority level 2 specifically allows the stationary batteries to receive energy from the EV in question, in case the algorithm senses that the household will have a deficit of energy in the short term. Additionally, in both scenarios, the amount of energy the EV is willing to discharge is the difference between the current SoC and the user defined energy needs for the next trip, given that the SoC is higher than the threshold. This ensures that the EV never goes below this boundary. The flow chart can be seen in Fig. 4.3a.

Stationary batteries, most of the times, have a priority level of 3, meaning that the energy they hold is only spent on items that highly require it. In this case, all the energy contained in the battery is ready to be provided. However, if the production of energy is currently higher than the household consumption prediction and the available space in the connected batteries and EVs is not enough to store the predicted remainder of energy, the battery's priority is set to 1, in order to empty the battery by making its energy more accessible. This is done to create room for the energy being produced. In this case, the battery wants to discharge an amount of energy that is enough to make room for the expected production remainder (if there are multiple batteries, not all of them perform this action, only the ones needed to create enough room for the exceeding production). If the battery charge rate and maximum capacity can still hold additional energy in the next time step, the stationary battery will have another entry on the list with the normal priority. The flow chart can be seen in Fig. 4.3b.

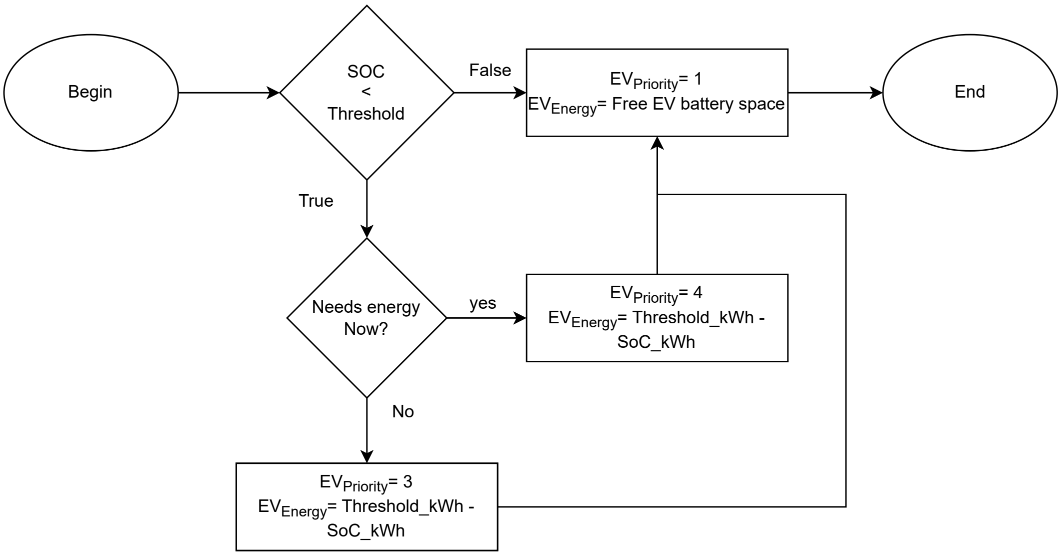
The grid has an unlimited supply of payed energy and is able to supply whatever energy is required. The priority of the grid to provide energy is decided based on its current price, Fig. 4.3c. This price is compared with the grid's average price as well as the cost of charging the available batteries. Batteries have expected life spans that depend on how much they are used, since they deteriorate with use over time. From these variables, it is possible to define a price to charge a battery. The algorithm considers this price as the price of discharging the battery, since when the battery is discharge it has to be charged later (if an EV gives an amount of energy, this energy needs to be restored later). Basically, the algorithm takes advantage of periods when the grid price is lower than usual. It is worth noting that with a more complex system to predict the grid price behavior, the algorithm could perform more complex decisions. At the moment, the decision making is overly simplified, since interactions with the grid are not in the scope of this research.

4.3.2 Elements Eligible to Charge

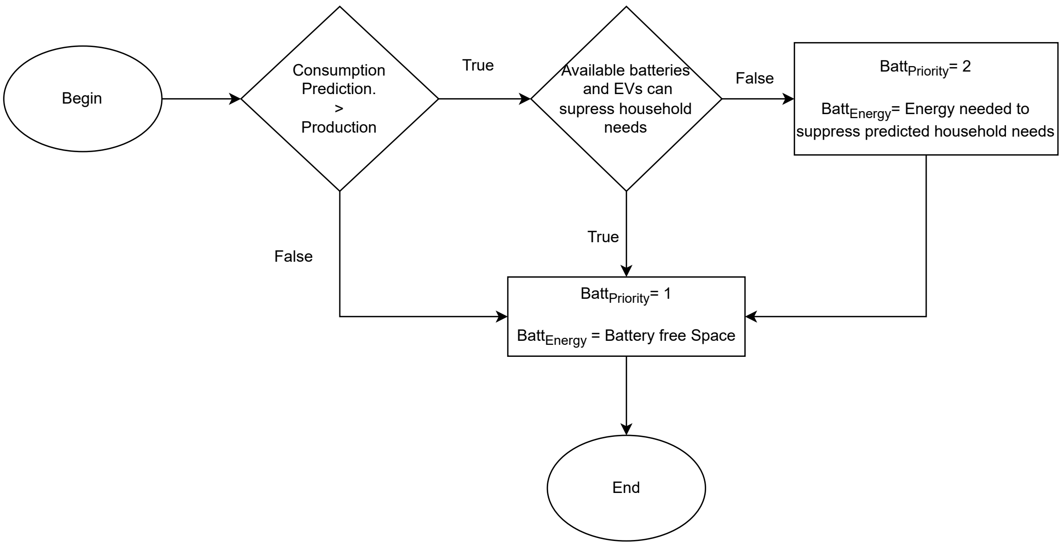


(a) Decision algorithm charge priority flow chart - House Consumption

All energy required by the household has to be provided since the system cannot impair the daily life of its users. One way of providing energy to the household is discharging a battery at a rate that satisfies the household needs. The amount of energy that the house needs is predicted by a machine learning model. Therefore, if the energy provided by the grid is to be avoided, this prediction can be used to find out the rate to discharge an EV or stationary battery. However, if the prediction is to high or to low, it can pose a problem. If



(b) Decision algorithm charge priority flow chart - EV



(c) Decision algorithm charge priority flow chart - Stationary Battery

Figure 4.4: Decision Algorithm flow charts on deciding the priority of the elements to perform a charging action

---

the prediction is too high, some of the energy being discharged does not get used and flows to the grid, being lost. This is a really bad scenario, which should be avoided, particularly, if the available energy is not much to begin with. If the guess is too low, the remaining of the household needs are automatically, suppressed by the distribution grid. This can be used to the algorithm advantage since in some scenarios it might be better to buy energy from the grid than overestimating the amount of energy the household would need, and discharge an EV too much (for example, when the grid price is low and there is not much energy available outside of the grid). Note that this is not as much of a problem in a real world scenario, in which the algorithm would run with small time steps, since the errors in energy prediction would be corrected in a very small window. However, since the current work was tested in a simulated environment with one hour time steps, this problem can impair the results of the algorithm and needs to be dealt with. The algorithm constructs an idea of how much energy is available (not counting with the grid obviously), how much energy is needed in the next time step and how much energy produced in the solar panels could go to waste in the next time step (energy being produced needs to flow somewhere, if all batteries are full or already using the maximum charge rate, the energy flows to the grid and is “wasted”). With these parameters, as well as grid energy price, it decides how much energy to reserve for the household. If the algorithm realizes there is no available energy in the batteries, it decides to set household receive energy priority to 0 (only receives from the solar panels), just so that, if there is an unexpected high production value in the next time step, the house can benefit. However, most of the times this just means that the distribution grid will supply the household. In a scenario where an excess of energy is available (being produced or in storage) or the grid price is too high, the algorithm uses the energy prediction value with the highest priority, not afraid of wasting energy. However, if there is not much energy available and the grid price is acceptable, the algorithm considers not trusting the prediction as much and reserves only a fraction of it, as it will have an error and could translate in energy loss. The algorithm also considers that there might be an error in the energy production prediction, and makes sure the energy being produced is always directed to the household before going to the grid. The flow chart is available in Fig. 4.4a.

All EVs must have a SoC value higher than the user defined threshold. If this is not the case, the EV is given a high priority for charging. Utilizing the estimated departure time, energy needed and charge rate, the model can decide if the charging action needs to start as soon as possible or if it can wait. If the charging action needs to start right away, the EV receives a priority value of 4 (mandatory), if not, a priority value of 3 is given, which is still a high priority. The amount of energy that the EV will charge is the amount needed to reach the threshold. If or after the minimum energy requirements are satisfied, the EV receives a priority of 1, only receiving energy if there is an exceeding amount somewhere. The amount of energy in this case is whatever value the EV can still hold, Fig. 4.4b.

Stationary batteries, most of the times, have a priority of 1. This means that they store energy from the solar panels that exceeded the current household and EV needs.The amount of energy in this case is whatever value the EV can still hold. Batteries can also store energy from the grid, if it is currently with a very low price. If the consumption prediction is higher than the production, the algorithm looks at the available energy in stationary batteries and vehicles. If the energy stored is not enough to satisfy the house needs in the next time steps, the battery’s priority is increased to receive energy. This way, the battery is eligible to receive energy from EVs that are leaving soon (and thus are not available in the next time steps to help satisfy the household consumption). Additionally, it increases the battery chances to receive energy from the grid, in case the grid price is lower than average. If the battery charge rate or maximum capacity can still hold more energy, the battery is reinserted on the list with normal priority. The flow chart can be seen in Fig. 4.4c.

The grid is also capable of receiving unlimited exceeding energy. This is not optimal since this energy is basically being given away. This is a last resort, in case there is no other way of storing it. Therefore the grid is given a priority level of 0 for receiving energy. There are plenty of ways of using the exceeding energy. For example, in a more realistic scenario, this energy could be used to turn on a washing machine or other appliances. Additionally, instead of giving this energy to the grid, it can be stored in the form of temperature, by turning on the heating or cooling system.

4.3.3 Baseline Decision Algorithm

To evaluate the performance of the proposed algorithm, a term of comparison in needed. For this, a baseline decision algorithm was also built, privileging the energy produced in solar panels to be used in the household, charge EVs or stationary batteries. The baseline algorithm always charges the EVs when they are connected, until they are filled, even if that requires utilizing energy from the grid. Additionally, EVs do not engage in discharging actions. This means that this algorithm does not make use of bi-directional chargers and that energy charged onto the EV is used only by that EV. On the other hand, the stationary Battery is only used to store energy from the solar panels, no energy from the grid is requested, however, it can still provide energy to the house or EVs. To decide how much energy to reserve for the household consumption, the consumption prediction value is the last observed consumption value.

4.4 Simulation

The decision algorithm described was tested in a simulated environment, since there was not enough time to install the desired system on a real household and collect results, nor does this study possess the required means to make that happen. Additionally, simulation enables the rapid test of multiple scenarios and facilitates possible modification that the algorithm might need.

The components of this system were simulated utilizing the Python programming language.

4.4.1 Components

The components that make part of the environment needed to be implemented in Python classes. This section gives a brief overview on how those modules work.

Consumption Meter

In the simulated environment, the consumption readings were be obtained from a csv file. The dataset is the same as the one used in Section 3.3.1 [39]. It contains household consumption reading for London houses. The simulation iterates through the file and for each time step it reads the next consumption value.

Production Meter

As introduced, irradiance values can be obtained from an API, such as [48]. Unfortunately, this API, like all others that were found, has a strict limit of API calls per day which means that it cannot be used to test the presented algorithm. This is because in the simulation, several days were simulated in a short time span, thus implying several API calls in a reduced period. To overcome this problem, the irradiance values were obtained from a CSV file, provided by the same website. This file contains irradiance readings from London (same location as the consumption dataset) and it is relative to the same periods of time. The simulation iterated through the file and, for each time step, read the next irradiance value.

This value is used to compute the total energy production in that time step, according to Equation 4.1.

Batteries

Batteries were implemented in a simple class. This class was used to represent both stationary batteries and EV batteries. Batteries have a maximum capacity, current capacity and charge rate. Additionally, price and battery cycles were also implemented as variables to calculate a price of charging the battery. Losses in the charging and discharging process are also considered.

EVs

Both departure time and necessary energy would be asked to the user through a mobile application. On the proposed simulation environment to test the algorithm, several trips were simulated in a short amount of time. Therefore, this information was generated randomly.

Several EVs were initialized, equal to the maximum EVs defined in the simulation initialization. This number can be viewed as the number of EVs owned by the family, or the number of chargers available. Every hour, for every free charger that is not occupied, there is a probability that an EV would arrive and take that space. The probability of an EV arriving is 10% every hour except in the hours: 11, 12, 13, 18, 19, 20, 22, 23. These hours were defined as the most common hours for an EV to arrive and the probability of an EV arriving is boosted to 25%.

The SoC that the vehicle is holding, when it arrives, is defined based on the SoC with which the vehicle left, the previously user defined threshold (energy needed for the trip) and a random value that represents how much of the threshold value was actually used, according to Equation 4.2:

$$SoC = Prev.SoC - Prev.Threshold * used \tag{4.2}$$

where *used* is a value between 0.2 and 0.8. This value represents what percentage of the value that was predicted to be needed was actually used. The EV never has a SoC greater than one or smaller than zero. If this is the first time the EV is being introduced, the SoC value is 0.

The minimum battery threshold represents how much energy the EV will need in the next trip (this value is meant to be an estimation from 0 to 1, where 1 represents full capacity). It was defined with triangular probability, as depicted in Fig. 4.7. This distribution allows to define a minimum, maximum and most common value or mode. On this instance, the minimum value is 0.1, maximum is 0.8 and mode is 0.3. These values were sometimes reduced in the eventuality that the EV was schedule to leave soon. This was necessary to avoid creating a scenario in which it was not possible to charge the EV on time.

The real departure time was chosen right away. The hours that were considered more common to leave the house were 8, 14 and 20. The simulation chooses one and generates the actual departure time with a Gaussian distribution, Fig. 4.6, with mean on that value and deviation of 2 hours. If the the resulting hour is lower than the current hour the EV only leaves in the next day. This means that the EV was never more than 24 hours parked. The departure guess, which was the user's guess to when he was going to leave, was created based on the previously generated real departure time. The guess was generated with the Gaussian

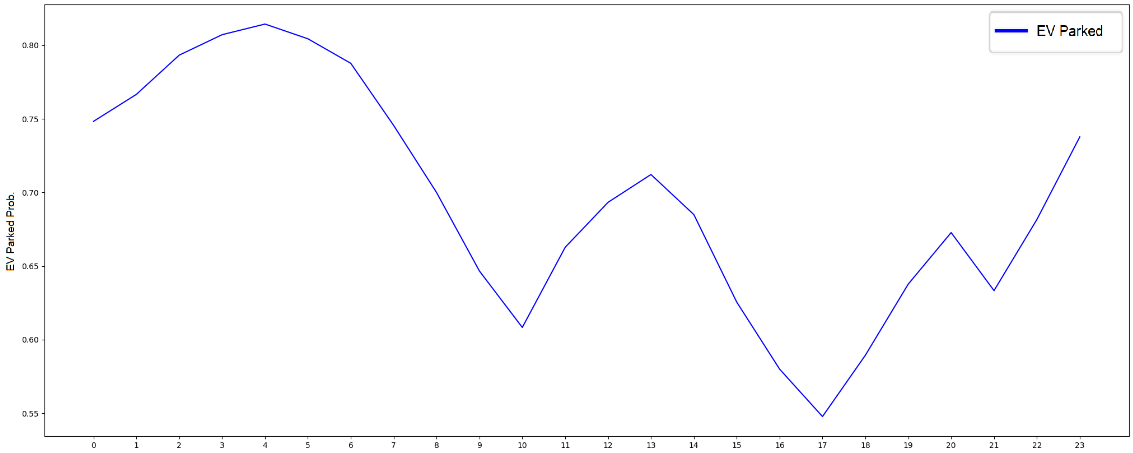


Figure 4.5: Probability of an EV being parked per hour of the day

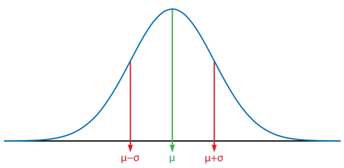


Figure 4.6: Gaussian Probability

distribution once again, with mean equal to the real value and a maximum standard deviation of 2 hours (if the EV was leaving soon, the deviation is smaller as the user is more sure that he needs to leave). The real departure time was only used by the simulation to keep track of when the vehicles should leave. The decision algorithm only looks at the departure guess that has some level of error.

The departure and arrival time behavior of an EV according to the simulation can be seen in Fig. 4.5. At each hour the line depicts the probability of an EV being parked at home. As it can be seen, the probability of the vehicle being at home decreases in the morning, afternoon and around dinner time. This probability is higher at nigh, lunch time and evening.

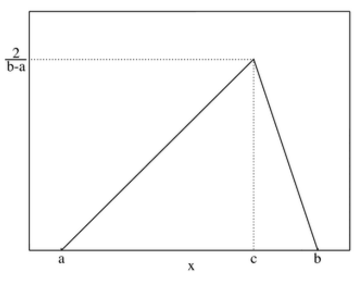


Figure 4.7: Triangular Probability

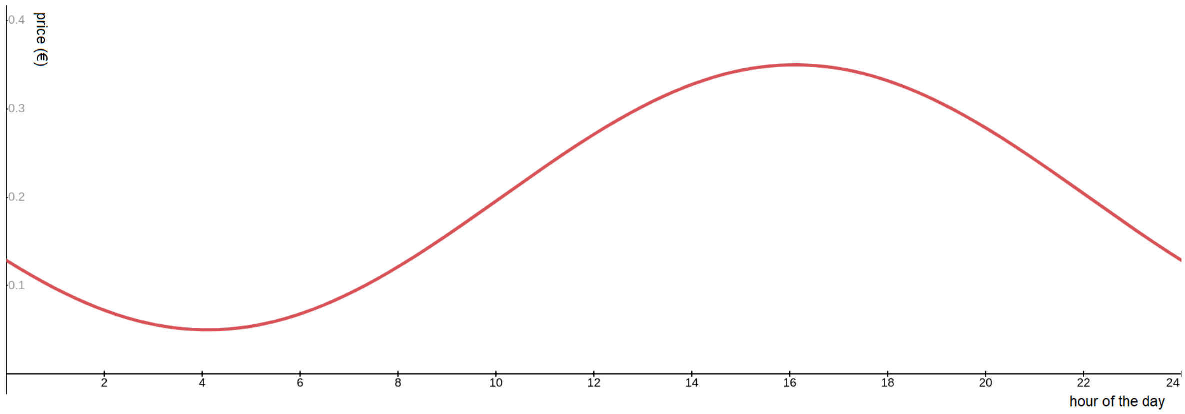


Figure 4.8: Dynamic Grid kWh cost per hour of the day

Grid

The grid is not the main focus of this experiment. However, it was still considered to achieve more realistic results. Two different grid pricings were analyzed. One with a linear price of 20 cents per kWh, and another with a variable price that fluctuates between 5 cents and 35 cents, with average price of 20 cents per kWh. The period with the lowest price is around 4 to 5 a.m., when the energy usage is the lowest. The period with highest price is around 5 p.m., when demand is higher. This pattern has a daily periodicity, the visual representation is available in Fig. 4.8, that corresponds to Equation 4.3. Needless to say that this representation is a simplified approximation. However, it follows a similar behavior to a real world scenario, as can be seen from the behavior presented in [49].

$$price = 0.15 * (cos((hour/3.82) - 10.5) + 1) + 0.05 \tag{4.3}$$

where price is the price of the kWh in euros and the current hour is the hour of the day.

4.4.2 Simulation Behavior

To test the decision algorithm, the previously mentioned components were simulated as Python classes. A main simulation loop that collects these inputs and provides them to the algorithm is also necessary.

The main simulation is responsible for initializing all components. The simulation keeps a date variable as if it was the current time. The production values are obtained from a CSV file, which contains the irradiance readings from the same location as the test houses. The consumption and production readings are read for the same timestamp to avoid inconsistencies (both production and consumption datasets contain the same date periods).

For every time step, the main simulation gathers all the information from the inputs and feeds it to the decision algorithm. In the algorithm, these inputs are processed and organized in a list of priorities, as already explained. After this, the decision algorithm returns a list of decisions. Each decision contains an object that it is related to (a specific EV for example), a charge or discharge option and the amount of kW the charging or discharging action should have. Then, the main simulation makes sure these decisions are reflected on the simulated objects. The mentioned architecture can be better observed in Fig. 4.9.

The simulation updates the inputs and runs the decision algorithm in one hour time steps. On a real world scenario, the algorithm would be called several times per hour, possibly every few minutes. This would make possible to keep correcting deviations and adapting to the situation. For example, if the house consumption prediction value was very far off, it could be rapidly corrected, or if one EV left or arrived in the middle of the time step, it could be quickly dealt with. However, since the consumption and production datasets only contain information in one hour time steps, the simulation needs to run in the same basis, as more granular information is not available. The study in [44] opted for the same solution to a similar problem.

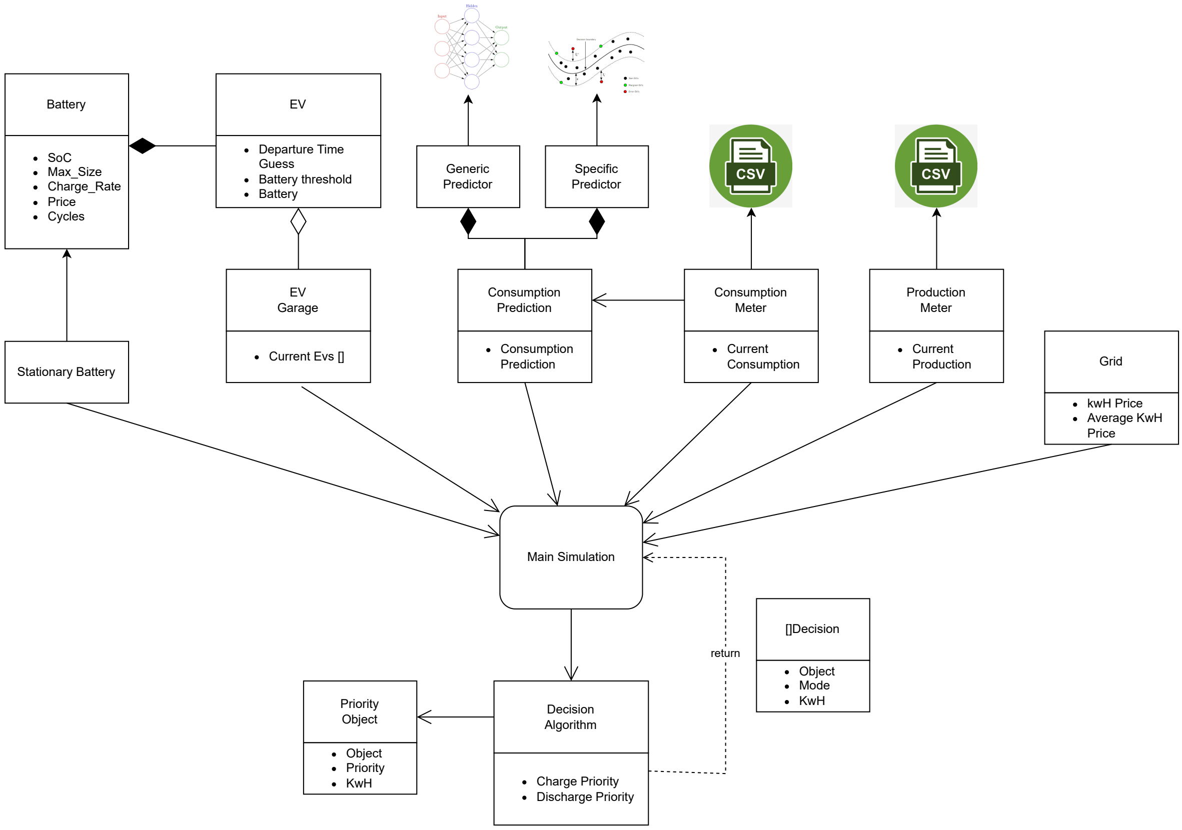


Figure 4.9: Architecture of the simulated environment

4.5 Test and Results

In this section, the built decision algorithm is tested using the defined simulation. Several parameters in the environment are tweaked to test the algorithm under different conditions. A baseline decision algorithm is also tested for comparison, as defined in Section 4.3.3.

4.5.1 Scenarios

In order to test the presented algorithm, several scenarios were planned out. These scenarios are meant to explore variations in the testing environment and highlight the strengths and weaknesses of the algorithm.

For the sake of simplicity certain values were kept constant across all tests and are documented in Table 4.1.

Constants	Value
Battery charge/discharge energy loss	2%
EV battery max capacity	24 kWh
EV battery charge rate	3 kW
EV battery price	4500 €
EV battery price per kWh	0.038 €
Stationary battery max capacity	50 kWh
Stationary battery charge rate	5 kW
Stationary battery price	10000 €
Stationary battery price per kWh	0.04 €

Table 4.1: Values kept Constant through the experiment

The three base scenarios vary in energy produced versus energy consumed ratio. In the first base scenario, the average monthly energy produced is lesser than the household energy consumption. In the second, the average monthly value of production and household consumption are on the same level. On the third, the production is greater than the consumption. The production value is changed by defining a larger or smaller solar panel area. On these base scenarios the rest of the experiment components are as follows:

- 2 EVs
- 1 Stationary Battery
- Dynamic Grid behavior

Inside each scenario, four sub-scenarios take place, each varying one of the previous parameters:

- Sub-scenario 1 - Only 1 EV
- Sub-scenario 2 - 3 Evs
- Sub-scenario 3 - No stationary batteries
- Sub-scenario 4 - Grid price with linear behavior

4.5.2 Metrics

To evaluate the performance of the decision algorithm the following metrics are used.

Electric Bill Cost: The electric bill is what this decision algorithm is trying to minimize in the first place. Basically, is how much the user would pay for the house electric bill for all the energy bought from the distribution grid.

Battery Depreciation Cost: This cost relates to how much the value of the used batteries is going to decrease due to using them. As mentioned, from the battery price and it's life expectancy, a cost is attributed to charging the battery. Note that this value is probably not a good approximation of reality since the way battery cycles work is not as simple. In reality, other parameters affect the battery life in ways that are not accounted for in this simulation, however, for the sake of simplicity, a constant cost per kWh charged is how this metric was defined 4.1. This metric was only used as a means of comparing the decision algorithm with the baseline, to understand which algorithm shortens the batteries' life faster.

Total Cost: This metric is simply the Electric Bill Cost and Battery Depreciation Cost summed up.

Emitted CO2: Due to this being a project that deals with green energy being produced in solar panels and electric vehicles, the emitted CO2 was also taken into account. To simulated this, a constant value of grams of CO2 emitted per kWh of consumed energy that was imported from the grid was defined. This value is 233 grams, according to [50, 51] which is a study done in the UK, where the used datasets are from. The value of this metric increased when energy that was imported from the grid was consumed by the household or by one of the EV (not when it is charged, only when used). However if the energy is green (from the solar panels) it did not emit any CO2.

4.5.3 Results and Discussion

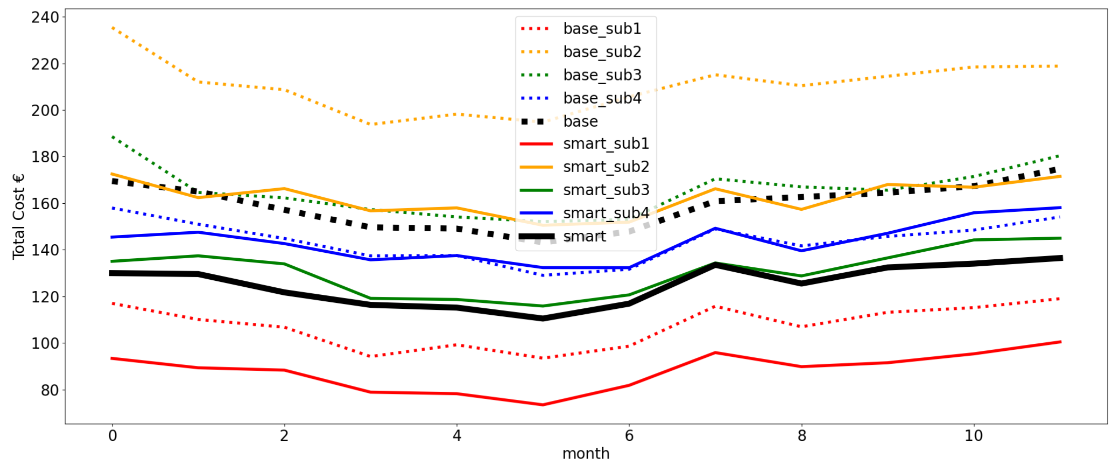
This section presents the results obtained for the previously defined experiment. A total of 10 test households, with different house consumption behaviors, were picked at random, from the consumption meter dataset [39], and were subject to all the scenarios. The average results for all the houses are discussed, as well as a more detailed look into one of the households. The duration of all tests is of 1 year.

Average Total Cost

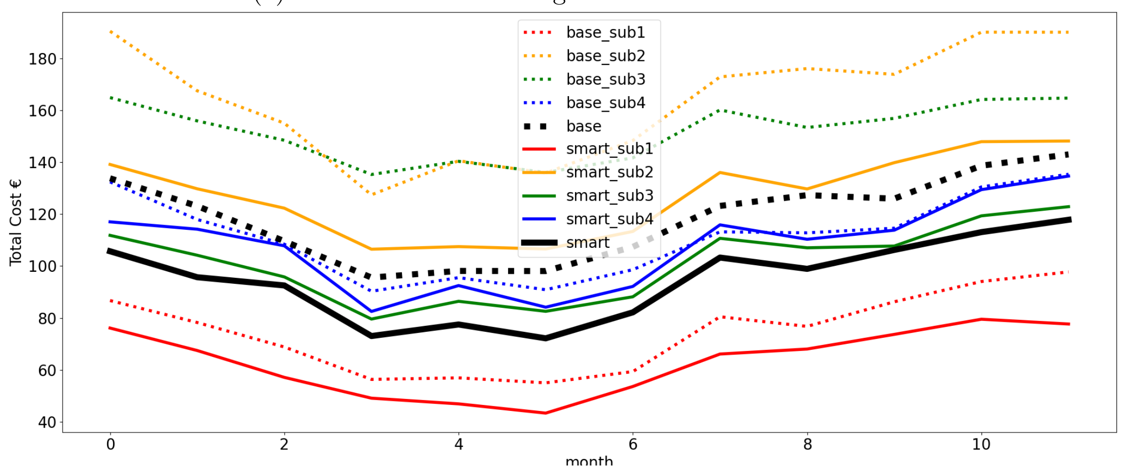
As defined, 10 households were subject to three test scenarios, each encompassing 4 more sub-scenarios. The factor that varies within scenario 1, 2 and 3 is the level of energy production. In scenario 1, this production is much lower than the consumption, in scenario 2, they are similar, and in scenario 3 the production is greater than the consumption.

In Fig.4.10 a plot of the average total electrical cost in these 10 houses by month can be seen. There are three figures, each relative to one scenario and containing all sub-scenarios for the designed decision algorithm as well as for the baseline decision algorithm. As can be seen in the legend, the continuous lines are relative to the proposed algorithm and the dotted lines represent the baseline decision algorithm. The black and thicker lines are relative to the base scenarios. The thinner lines are relative to the other sub-scenarios. Red for sub-scenario 1, orange for 2, green for 3 and blue for 4.

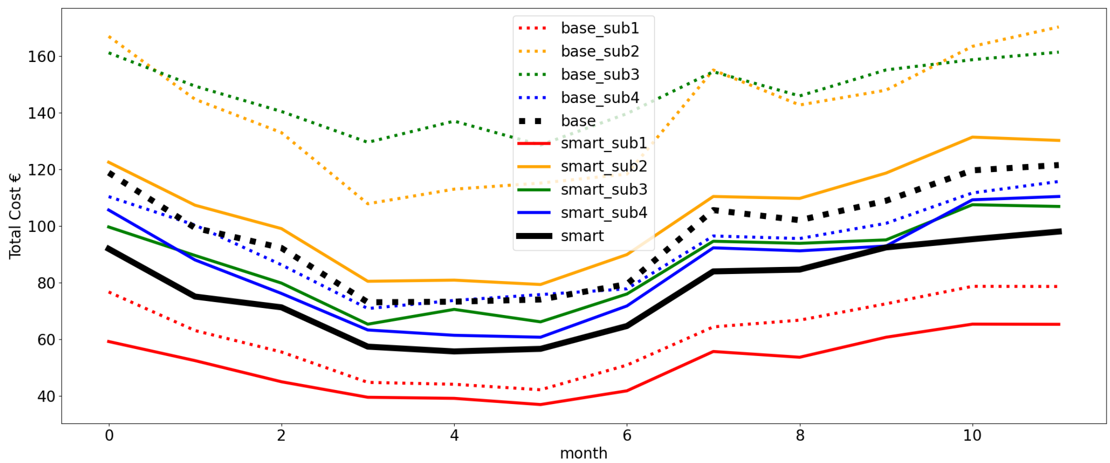
The first impression, is that the dotted lines are, on average, above the continuous lines, especially, when they are the same color (relative to the same sub-scenario). This means that the designed decision algorithm is in fact better than the baseline algorithm. Looking at all the figures, it is possible to see that the y-axis values decrease from image a) to b) to c). This



(a) 10 Test Houses Average Total Cost for Scenario 1



(b) 10 Test Houses Average Total Cost for Scenario 2



(c) 10 Test Houses Average Total Cost for Scenario 3

Figure 4.10: 10 Test Houses Average Total Cost

is because, in each scenario there is more and more energy being produced in the solar panels, meaning that less energy is bought from the grid, thus reducing the total cost.

On sub-scenario 1 (red), one less EV relies on the system to charge, therefore, it is normal that the total cost is less than the base scenario (black). Sub-scenario 2 (orange), on the other hand, increased the number of EV to three. Naturally, this increases the total cost. The total cost achieved by both algorithms decreases as more energy production is available, however, the savings achieved by the proposed model stay somewhat constant, as can be seen in Table 4.2.

Much more interesting is the sub-scenario 3 (green). In this sub-scenario the stationary battery is removed. Basically, the always available battery does not exist for the algorithm to store energy in. This resulted in the baseline algorithm's total cost to stay somewhat constant, especially from scenario 2 to 3, unlike the other sub-scenarios that had their total cost significantly decrease, since more energy is being produced (green dotted line in Fig.4.10c).

The designed algorithm is immune to this because it is able to make use of V2G chargers and treat the EVs as batteries and take energy from them for supplying the household, which the baseline algorithm can't. This is another strong suit for the designed algorithm - don't depend on stationary batteries. In Table 4.2, it is possible to see that the baseline algorithm practically did not reduce the total cost from Scenario 2 sub-scenario 3 (1821.73€), to Scenario 3 sub-scenario 3 (1762.20€), even though almost double the energy is being produced. Because of that, this is the sub-scenario where the proposed model achieved the biggest savings, up to 40%.

On sub-scenario 4 (blue), the grid price exhibits a linear behavior, unlike the high and low behavior exhibited on the previous sub-scenarios. One thing to notice is that on the proposed algorithm (continuous blue and black lines), the blue line is higher than the black line. This means that, even though the grid price has the same average price over the day, the proposed model can make use of the low price periods in order to produce savings. However, the baseline algorithm (dotted blue and black lines), has the black line higher than the blue line, which means that it prefers the flat price. This probably happens because the time the grid energy is higher coincides with a period of high house consumption (5 p.m.), since the grid is more saturated at that time. So the baseline algorithm is buying energy in periods of high price and not benefiting from the low energy price (around 4 a.m.) to charge batteries.

One other thing to note on sub-scenario 4, is that it is, by far, the worst case scenario. In Scenario 1, the proposed approach didn't practically achieve any benefits in cost reduction. It is important to understand what having a distribution grid with linear behavior implicates. Given that the environment energy needs are x kWh over the month, and the energy available to suppress those needs through production are y kWh. If x > y, meaning the total production is not enough for the total energy required, eventually, the algorithm would need to rely on the distribution grid to face the energy demand. The proposed algorithm does not create more energy than the baseline, it simply is better at postponing to buy energy, waiting for a better deal. In other words, the proposed algorithm has to buy a similar amount of energy to the grid, it simply waits for the best time to do it. If the grid price is forever the same, there would be no better deal, meaning that the proposed algorithm made more use of the available batteries (wearing them and increasing the battery cost metric), for no advantage at all, since that, at the end of the month it would need to buy the same amount of energy from the grid. Why is it that the proposed algorithm still has a better overall performance in sub-scenario 4? Likely, because it makes use of a energy prediction model, unlike the baseline which uses the last observed value as a prediction. The prediction with less error and the

		Total Cost		
Scenario	Sub-scenario	Baseline Alg.(€)	Proposed Alg.(€)	Saving(%)
Scenario 1	Base Scenario	1911.22	1502.12	21.4
	Sub1	1289.44	1056.55	18.06
	Sub2	2525.72	1948.05	22.87
	Sub3	1986.71	1569.53	20.99
	Sub4	1728.24	1723.28	0.29
Scenario 2	Base Scenario	1423.70	1138.05	20.06
	Sub1	896.69	758.74	15.38
	Sub2	1968.00	1526.64	22.42
	Sub3	1821.73	1216.15	33.24
	Sub4	1340.43	1294.37	3.44
Scenario 3	Base Scenario	1168.44	927.43	20.62
	Sub1	738.49	614.97	16.72
	Sub2	1679.24	1260.49	24.93
	Sub3	1762.20	1045.51	40.67
	Sub4	1115.99	1023.52	8.29

Table 4.2: House Average Annual Total Cost by scenario and Decision Algorithm

caution to not overestimate house consumption, discharging more energy than needed from a battery in the process are, most likely, what is causing the proposed approach to still achieve better results in this sub-scenario.

Individual House Analysis

The overall results in terms of total cost were already discussed. Here, a more detailed summary of the results of one of the 10 households is looked into, more specifically, house 0. There was no rule to chose which house to present in this section. It is believed that all test houses are somewhat similar and representative of the results. However the output for each individual household can be obtained in [52].

The household consumption versus production can be seen in Fig.4.11, as already mentioned, the production value varies by scenario. In Fig.4.12 the total cost evolution across all scenarios can be seen. Tables 4.3 and 4.4 document all results for the proposed and baseline algorithm, only for base scenario 1 (for sake of simplicity).

As introduced, the total cost metric is the sum of the electric bill cost and the battery cost (expenses of charging a battery due to wear and tear). As can be seen in Table 4.3, the total value of battery cost is actually higher for the proposed approach, comparing with Table 4.4. This means that the proposed decision algorithm is actually wearing these batteries faster than the baseline, due to using them a lot more. However, the gains that the decision algorithm can obtain in the reduction of payment is enough to cover this extra expense. Furthermore, with the improvement of battery technology the wear and tear costs will possibly be reduced.

The “Imp. Evs” column has to do with the number of cases where an EV was impaired by the algorithm, meaning, the algorithm prevented it from its normal daily trips. This is basically relative to an EV that left with a lower SoC than the user defined minimum charge threshold needed for the next trip. As can be seen, this never happened in the tests and it highly unlikely to happen,. However, it can occur when the guess of the date in which the

Month	Total Cost(€)	Bill Cost(€)	Battery Cost(€)	Imp. Evs	CO2(Kg)	House Consumption (kWh)	Prod. (kWh)
1	139.84	112.15	27.69	0	191.81	490.28	31.31
2	128.61	102.69	25.92	0	209.93	606.17	60.59
3	118.92	94.28	24.65	0	181.92	520.41	65.73
4	98.0	70.7	27.3	0	140.69	312.8	146.15
5	112.88	87.07	25.81	0	156.87	382.64	74.91
6	95.15	68.97	26.18	0	127.66	260.65	117.87
7	96.9	71.02	25.88	0	120.42	175.47	89.72
8	91.28	66.45	24.83	0	110.63	193.69	114.57
9	94.42	71.31	23.11	0	107.48	189.8	93.22
10	90.85	65.33	25.52	0	119.94	222.8	127.26
11	106.77	82.07	24.7	0	142.26	287.05	56.14
12	129.4	101.23	28.17	0	197.84	504.98	49.77
Total	1303.02	993.27	309.76	0	1807.45	4146.74	1027.24

Table 4.3: Proposed Decision Algorithm Results for House0, Base Scenario 1

EV would depart was very far off, or when the EV charge requirements were too high for the parking time (EV was not able to completely charge in the time that was given). These instances were majorly prevented by the test simulation, but, as random numbers are in play, it may still happen.

One final thing to note, the emitted CO2 for the proposed algorithm is a bit greater than the CO2 emitted by the baseline algorithm. This parameter is not constant across all houses and scenarios. High CO2 emission may be caused by the house necessitating more energy in periods of low production, thus requiring energy from the grid. Additionally, the difference between the two algorithms can come from the proposed method taking advantage of the low grid prices to fill the batteries with non-green energy. Despite this, the difference is so small it could just be a coincidence. Actually, in certain scenarios such as scenario 3, sub-scenario 3 (high production, no stationary battery), the emitted CO2 is much greater for the baseline algorithm, as the proposed method supplies the household with green energy stored in the EVs and the baseline doesn't have that option. In that scenario, for house 0, the proposed algorithm emits 1173.39 Kg of CO2 and the baseline 1702.88 Kg. On average, across all houses and sub-scenarios, the proposed algorithm emits 1276 Kg of CO2, while the baseline emits 1346 Kg. This is not a huge difference, but still shows that the proposed algorithm makes better use of the green energy being produced in the solar panels.

In Fig. 4.12 it is possible to see that in some months and sub-scenarios, the baseline algorithm achieved a better performance than the proposed algorithm. These rare exceptions are more likely to be found when looking at one specific house, since the average of all results tends to be smoothed. In both baseline and proposed algorithms, the simulation requires, on average, the same amount of energy from both algorithms. The consumption and production are exactly the same, however the behavior of the EVs is simulated and, therefore, somewhat random. So it is possible that, in a rare occurrence one of the algorithms might have had a harder task at hands.

Month	Total Cost(€)	Bill Cost(€)	Battery Cost(€)	Imp. Evs	CO2(Kg)	House Consumption (kWh)	Prod. (kWh)
1	163.39	145.79	17.6	0	188.46	490.28	31.31
2	161.91	145.3	16.62	0	213.23	606.17	60.59
3	147.31	131.32	15.99	0	185.04	520.41	65.73
4	121.49	101.77	19.72	0	130.31	312.8	146.15
5	140.2	123.54	16.66	0	152.59	382.64	74.91
6	121.54	103.39	18.15	0	120.64	260.65	117.87
7	124.29	105.46	18.83	0	111.67	175.47	89.72
8	118.25	99.33	18.92	0	109.96	193.69	114.57
9	131.25	112.0	19.25	0	118.67	189.8	93.22
10	102.66	86.76	15.9	0	98.2	222.8	127.26
11	147.49	130.73	16.77	0	139.35	287.05	56.14
12	158.25	141.23	17.02	0	193.07	504.98	49.77
Total	1638.03	1426.62	211.43	0	1761.19	4146.74	1027.24

Table 4.4: Baseline Decision Algorithm Results for House0, Base Scenario 1

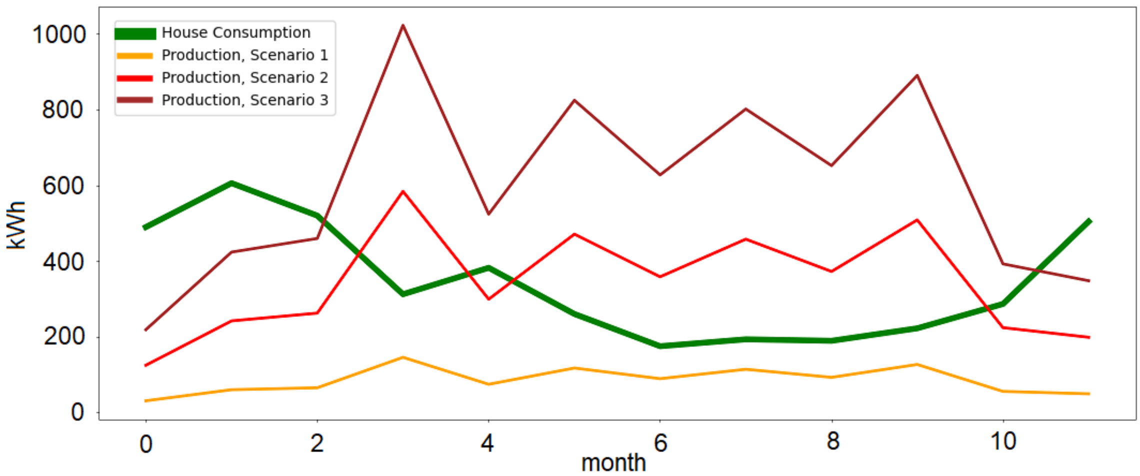


Figure 4.11: House 0 - Consumption vs Production

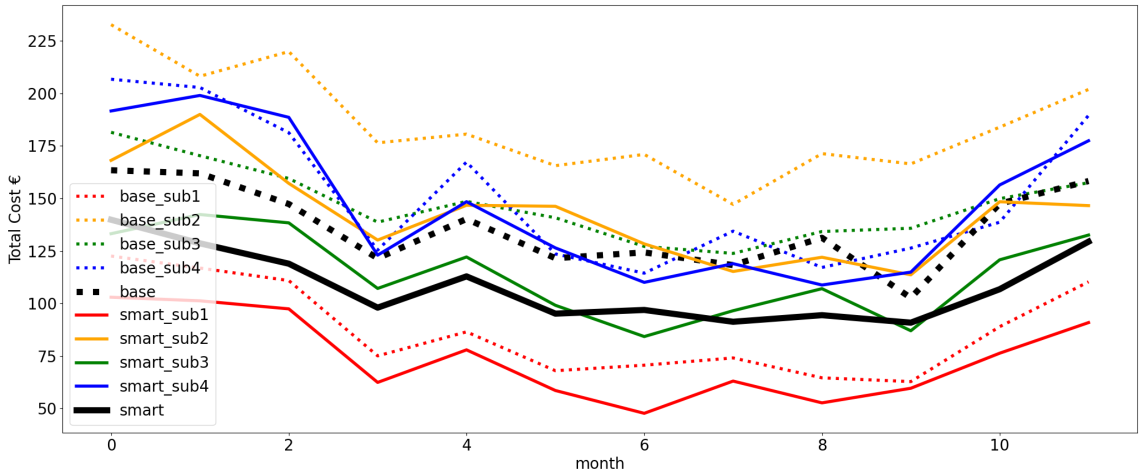
4.6 Discussion

In this chapter a decision algorithm was proposed. The algorithm's purpose is to minimize the electricity costs of the household it is inserted in. The algorithm was designed to deal with the behavior of one particular house and works only with variables from that house. The algorithm is meant to work in a environment where there is constant need of energy (a house or residential building), EVs owned by the user of the system, stationary batteries, some sort of energy production module (in this particular study - solar panels) and finally, a distribution grid.

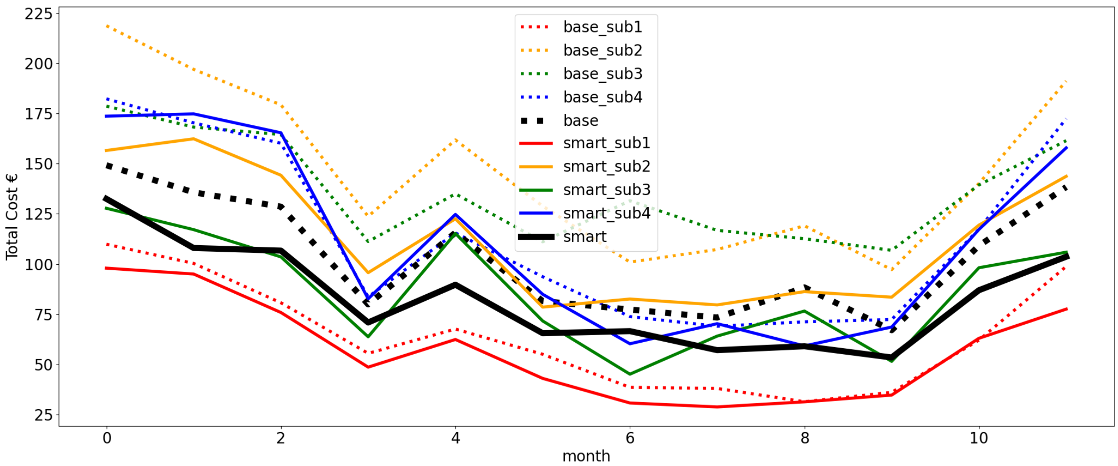
The proposed method actions are only to charge or discharge the available batteries, despite of the inputs containing more information. The algorithm works by organizing the elements in two lists, a list with elements that are eligible to charge and another with elements that are eligible to discharge. These list are organized by priority values. The algorithm then iterates through the lists making sure that every kW that is being charged is also being discharged from another place.

To test the defined algorithm a simulation environment is built encompassing all required components. The simulation mimics how the environment would play in a real world scenario. Off course, a simulation is never as good as a test in a real world scenario, but an effort was made to allow this algorithm to be tested in a representative environment. Variables such as consumption and production were actually collected from real life datasets to further decrease the amount of error. The simulation was coded using the Python programming language.

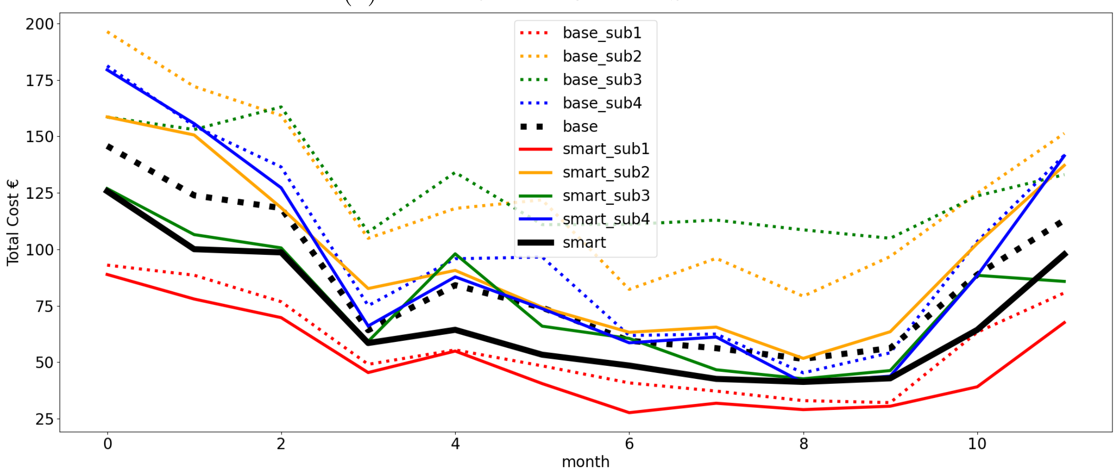
The proposed approach was shown to perform better than a baseline algorithm, that was built for comparison purposes. The designed method and the baseline were compared in a series of scenarios and in different test houses. For the different scenarios, the worst amount of savings the proposed approach achieved was a price reduction of 0.29%, showing that the algorithm cannot achieve any savings in a situation where the grid has a linear price behavior and the production module does not provide much energy. The best result was a price reduction of 40.67% and were achieved when the household did not have a stationary battery to rely on, and therefore, V2G chargers had a greater importance. The average result of the proposed algorithm, across all sub-scenarios and test houses, was a 19.29% cost reduction, comparing to the baseline.



(a) House 0 Total Cost for Scenario 1



(b) House 0 Total Cost for Scenario 2



(c) House 0 Total Cost for Scenario 3

Figure 4.12: House 0 Total Cost

Chapter 5

Conclusion

This work was conducted within the scope of the EV4ENERGY project, whose main objective is to put the EV at the center of the building energy ecosystem [9]. This study proposes a decision algorithm capable of reducing the energy costs of a residential building without interfering with the daily lives of its inhabitants. Initially, two stepping-stones were meant to be developed to aid the decision algorithm, a household load demand predictor and a model to forecast EV availability and energy requirements.

With these goals in mind, a bibliographic research was carried out. Two main techniques were found to be commonly used to predict household consumption: SVR and LSTM. Both models were compared and the achieved results favored SVR. This technique was chosen to implement a specific model, that trains with data relative to a single house and only makes prediction for that same house. Since the LSTM model supports incremental learning, unlike SVR, it was chosen to implement a generic model, trained with several houses and whose objective is to be able to predict consumption values for any household without training with previous records from that house.

The state of the art in forecasting EV availability and energy was shown to be undeveloped and there was a serious lack of datasets to train ML models with. The lack of datasets and information in this area made it impossible to create a complete study and develop a usable prediction system.

The final result of this work is a decision algorithm capable of reducing the electric costs of a household. For this purpose, the algorithm utilizes the household energy consumption prediction, renewable sources of energy and the distribution grid, as well as EVs and stationary batteries to store energy. The algorithm was tested in a simulated environment against a baseline, for multiple households and across different scenarios. Results indicate that the proposed algorithm serves its purpose, the worst amount of savings the proposed approach achieved was a price reduction of 0.29% and the best 40.67%. The average result across all houses and sub-scenarios is a 19.29% saving in the electrical expenses of the household, comparing with the baseline algorithm.

5.1 Future Work

One of the inputs of great value to the decision algorithm is the information relative to the connected EVs, namely departure time and energy needed for the next trip. Currently, these inputs are designed to be collected directly from the user, through a mobile application.

This is not how this component was initially though. There was an effort to build a machine learning model that could accurately predict these values. However, due to the lack of proper datasets, and time to carry out the study, this approach did not provide good results and was abandoned. The lack of datasets in this area was mentioned in almost all articles studied in the state of the art section, which proves that this is a real problem in this matter. However, being able to achieve the mentioned values without the need for user input is very important, as it is not expected for the user to be constantly bothered to give a guess for these values. A more in depth study on this area would be valuable to enhance the decision algorithm as a system. A good starting point for a future work in this area is to actually create a dataset with information related to EV travel behavior, since there is any available.

In this work, a household load demand forecaster was also built. A LSTM neural network was implemented based on the researched related work. The devised architecture was not subject to more variations seeking further improvements. The SVR ended up outperforming the LSTM, but the potential in NN is much higher. A more in depth study on how many neurons and layers a NN can benefit from, in the energy consumption prediction area, will surely contribute to improve the results.

Despite the SVR algorithm achieving better performance in predicting energy consumption, it was not chosen to implement the generic model. This is because the utilized library did not implement incremental learning, which would have forced the training process to be performed at once. Further research on how to implement a SVR based model, that supports incremental learning, might improve the results.

The utilized dataset to build the generic and specific models contained several houses. However, it did not contain a high number of features. This study could have benefited from a more detailed dataset to build the generic model, something to account when designing the data acquisition system underlying the forecasting algorithm.

The decision algorithm was tested in a simulated environment. Obviously, despite all the effort, a simulated environment is never the same as a real world experiment. To study the actual impact of the proposed approach, further studies would need to be applied in different conditions. One of the main aspects that could influence the results is running the algorithm at a much higher frequency, and not once every hour. Also, one of the main aspects of the algorithm is its interaction with the distribution grid. The behavior of the distribution grid, in terms of price to sell and buy energy, was not one of the main objectives of this study. However, a more advanced system needs to contain a module that can make predictions on the grid's energy price.

Bibliography

[1] Jarod C. Kelly, Jason S. MacDonald, and Gregory A. Keoleian. Time-dependent plug-in hybrid electric vehicle charging based on national driving patterns and demographics. *Applied Energy*, 94:395–405, 2012. doi:10.1016/J.APENERGY.2012.02.001.

[2] Jae Hyun Lee, Debapriya Chakraborty, Scott J. Hardman, and Gil Tal. Exploring electric vehicle charging patterns: Mixed usage of charging infrastructure. *Transportation Research Part D: Transport and Environment*, 79, feb 2020. doi:10.1016/J.TRD.2020.102249.

[3] United Nations Department of Economic and Social Affairs Population Division. 68% of the world population projected to live in urban areas by 2050, says UN | UN DESA | United Nations Department of Economic and Social Affairs, 2018. URL: https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html.

[4] Payam Nejat, Fatemeh Jomehzadeh, Mohammad Mahdi Taheri, Mohammad Gohari, and Muhd Zaimi Muhd. A global review of energy consumption, CO2 emissions and policy in the residential sector (with an overview of the top ten CO2 emitting countries). *Renewable and Sustainable Energy Reviews*, 43:843–862, mar 2015. doi:10.1016/J.RSER.2014.11.066.

[5] Colin McKerracher, O'DonovanAleksandra, Nick Albanese, Nikolas Soulopoulos, David Doherty, Milo Boers, Ryan Fisher, Corey Cantor, James Frith, Siyi Mi, Andrew Grant, Jinghong Lyu, Kwasi Ampofo, and Allen Abraham. EVO Report 2021 | BloombergNEF | Bloomberg Finance LP, 2021. URL: https://about.bnef.com/electric-vehicle-outlook/.

[6] Xudong Zhang, Feng Gao, Xianzheng Gong, Zhihong Wang, and Yu Liu. Comparison of Climate Change Impact Between Power System of Electric Vehicles and Internal Combustion Engine Vehicles. pages 739–747, jul 2017. URL: https://link.springer.com/chapter/10.1007/978-981-13-0158-2_75, doi:10.1007/978-981-13-0158-2_75.

[7] Outcome and response to ending the sale of new petrol, diesel and hybrid cars and vans - GOV.UK. URL: https://www.gov.uk/government/consultations/consulting-on-ending-the-sale-of-new-petrol-diesel-and-hybrid-cars-and-vans/outcome/ending-the-sale-of-new-petrol-diesel-and-hybrid-cars-and-vans-government-response.

[8] Julian Huber, David Dann, and Christof Weinhardt. Probabilistic forecasts of time and energy flexibility in battery electric vehicle charging. *Applied Energy*, 262:114525, mar 2020. doi:10.1016/J.APENERGY.2020.114525.

[9] Projeto Ev4Energy. URL: http://www.ev4energy.com/.

[10] Fatima Amara, Kodjo Agbossou, Yves Dube, Sousso Kelouwani, and Alben Cardenas. Estimation of temperature correlation with household electricity demand for forecasting application. In *IECON Proceedings (Industrial Electronics Conference)*, pages 3960–3965. IEEE Computer Society, dec 2016. doi:10.1109/IECON.2016.7793935.

[11] Teodor Petrican, Andreea Valeria Vesa, Marcel Antal, Claudia Pop, Tudor Cioara, Ionut Anghel, and Ioan Salomie. Evaluating forecasting techniques for integrating household energy prosumers into smart grids. *Proceedings - 2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing, ICCP 2018*, pages 79–85, oct 2018. doi:10.1109/ICCP.2018.8516617.

[12] Richard E. Edwards, Joshua New, and Lynne E. Parker. Predicting future hourly residential electrical consumption: A machine learning case study. *Energy and Buildings*, 49:591–603, jun 2012. doi:10.1016/J.ENBUILD.2012.03.010.

[13] Noman Shabbir, Roya Ahmadiahangar, Lauri Kutt, and Argo Rosin. Comparison of machine learning based methods for residential load forecasting. *2019 Electric Power Quality and Supply Reliability Conference and 2019 Symposium on Electrical Engineering and Mechatronics, PQ and SEEM 2019*, jun 2019. doi:10.1109/PQ.2019.8818267.

[14] Bing Dong, Zhaoxuan Li, S. M.Mahbobur Rahman, and Rolando Vega. A hybrid model approach for forecasting future residential electricity consumption. *Energy and Buildings*, 117:341–351, apr 2016. doi:10.1016/J.ENBUILD.2015.09.033.

[15] Xiaoou Monica Zhang, Katarina Grolinger, Miriam A.M. Capretz, and Luke Seewald. Forecasting Residential Energy Consumption: Single Household Perspective. *Proceedings - 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018*, pages 110–117, jan 2019. doi:10.1109/ICMLA.2018.00024.

[16] Filipe Rodrigues, Carlos Cardeira, and J. M.F. Calado. The Daily and Hourly Energy Consumption and Load Forecasting Using Artificial Neural Network Method: A Case Study Using a Set of 93 Households in Portugal. *Energy Procedia*, 62:220–229, jan 2014. doi:10.1016/J.EGYPRO.2014.12.383.

[17] Mohammad H. Alobaidi, Fateh Chebana, and Mohamed A. Meguid. Robust ensemble learning framework for day-ahead forecasting of household based energy consumption. *Applied Energy*, 212:997–1012, feb 2018. doi:10.1016/J.APENERGY.2017.12.054.

[18] Weicong Kong, Zhao Yang Dong, Youwei Jia, David J. Hill, Yan Xu, and Yuan Zhang. Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network. *IEEE Transactions on Smart Grid*, 10(1):841–851, jan 2019. doi:10.1109/TSG.2017.2753802.

[19] Weicong Kong, Zhao Yang Dong, David J. Hill, Fengji Luo, and Yan Xu. Short-term residential load forecasting based on resident behaviour learning. *IEEE Transactions on Power Systems*, 33(1), 2018. doi:10.1109/TPWRS.2017.2688178.

[20] Tae Young Kim and Sung Bae Cho. Predicting residential energy consumption using CNN-LSTM neural networks. *Energy*, 182:72–81, sep 2019. doi:10.1016/J.ENERGY.2019.05.230.

[21] Musaed Alhussein, Khursheed Aurangzeb, and Syed Irtaza Haider. Hybrid CNN-LSTM Model for Short-Term Individual Household Load Forecasting. *IEEE Access*, 8:180544–180557, 2020. doi:10.1109/ACCESS.2020.3028281.

[22] Ke Yan, Wei Li, Zhiwei Ji, Meng Qi, and Yang Du. A Hybrid LSTM Neural Network for Energy Consumption Forecasting of Individual Households. *IEEE Access*, 7:157633–157642, 2019. doi:10.1109/ACCESS.2019.2949065.

[23] Jing Zhang, Jie Yan, Yongqian Liu, Haoran Zhang, and Guoliang Lv. Daily electric vehicle charging load profiles considering demographics of vehicle users. *Applied Energy*, 274:115063, sep 2020. doi:10.1016/J.APENERGY.2020.115063.

[24] Zonggen Yi and Don Scoffield. A Data-Driven Framework for Residential Electric Vehicle Charging Load Profile Generation. *2018 IEEE Transportation and Electrification Conference and Expo, ITEC 2018*, pages 220–225, aug 2018. doi:10.1109/ITEC.2018.8450228.

[25] Songpu Ai, Antorweep Chakravorty, and Chunming Rong. Household EV charging demand prediction using machine and ensemble learning. *Proceedings - 2nd IEEE International Conference on Energy Internet, ICEI 2018*, pages 163–168, jul 2018. doi:10.1109/ICEI.2018.00037.

[26] Yingqi Xiong, Bin Wang, Chi Cheng Chu, and Rajit Gadh. Electric Vehicle Driver Clustering using Statistical Model and Machine Learning. *IEEE Power and Energy Society General Meeting*, 2018-Augus, dec 2018. arXiv:1802.04193, doi:10.1109/PESGM.2018.8586132.

[27] Sakib Shahriar, A. R. Al-Ali, Ahmed H. Osman, Salam Dhou, and Mais Nijim. Prediction of EV charging behavior using machine learning. *IEEE Access*, 9:111576–111586, 2021. doi:10.1109/ACCESS.2021.3103119.

[28] E. S. Xydas, C. E. Marmaras, L. M. Cipcigan, A. S. Hassan, and N. Jenkins. Forecasting Electric Vehicle charging demand using Support Vector Machines. *Proceedings of the Universities Power Engineering Conference*, 2013. doi:10.1109/UPEC.2013.6714942.

[29] Yu Wei Chung, Behnam Khaki, Tianyi Li, Chicheng Chu, and Rajit Gadh. Ensemble machine learning-based algorithm for electric vehicle user behavior prediction. *Applied Energy*, 254:113732, nov 2019. doi:10.1016/J.APENERGY.2019.113732.

[30] Juncheng Zhu, Zhile Yang, Yan Chang, Yuanjun Guo, Kevin Zhu, and Jianhua Zhang. A novel LSTM based deep learning approach for multi-time scale electric vehicles charging load prediction. *2019 IEEE PES Innovative Smart Grid Technologies Asia, ISGT 2019*, pages 3531–3536, may 2019. doi:10.1109/ISGT-ASIA.2019.8881655.

[31] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing 2004 14:3*, 14(3):199–222, aug 2004. URL: https://link.springer.com/article/10.1023/B:STCO.0000035301.49549.88, doi:10.1023/B:STCO.0000035301.49549.88.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[33] Subana Shanmuganathan. Artificial neural network modelling: An introduction. *Studies in Computational Intelligence*, 628:1–14, feb 2016.

[34] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems*, 6(2):107–116, 1998. `doi:10.1142/S0218488598000094`.

[35] Adam Blum. Neural networks in C++ : an object-oriented framework for building connectionist systems. 1992.

[36] François Chollet et al. Keras. `https://keras.io`, 2015.

[37] Smart Home Dataset with Weather Information. URL: `https://www.kaggle.com/datasets/taranvee/smart-home-dataset-with-weather-information`.

[38] Smart* Data Set for Sustainability. URL: `https://traces.cs.umass.edu/index.php/Smart/Smart`.

[39] Smartmeter Energy Use Data in London Households. URL: `https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households`.

[40] Stephan Kolassa and Wolfgang Schütz. Advantages of the MAD/Mean Ratio over the MAPE. *Foresight: The International Journal of Applied Forecasting*, (6):40–43, Spring 2007.

[41] Donovan Aguilar-Dominguez, Jude Ejeh, Alan D.F. Dunbar, and Solomon F. Brown. Machine learning approach for electric vehicle availability forecast to provide vehicle-to-home services. *Energy Reports*, 7:71–80, may 2021. `doi:10.1016/J.EGYR.2021.02.053`.

[42] Ivana Dusparic, Colin Harris, Andrei Marinescu, Vinny Cahill, and Siobhàn Clarke. Multi-agent residential demand response based on load forecasting. *2013 1st IEEE Conference on Technologies for Sustainability, SusTech 2013*, pages 90–96, 2013. `doi:10.1109/SUSTECH.2013.6617303`.

[43] Bin Zhou, Wentao Li, Ka Wing Chan, Yijia Cao, Yonghong Kuang, Xi Liu, and Xiong Wang. Smart home energy management systems: Concept, configurations, and scheduling strategies. *Renewable and Sustainable Energy Reviews*, 61:30–40, aug 2016. `doi:10.1016/J.RSER.2016.03.047`.

[44] Prakash Pawar, Shantanu Sampath, Trisha Ghosh, and K P Vittal. Load scheduling algorithm design for smart home energy management system. In *2018 IEEE 7th International Conference on Power and Energy (PECon)*, pages 304–309, 2018. `doi:10.1109/PECON.2018.8684141`.

[45] Energy Management Systems for Residential Buildings with Electric Vehicles and Distributed Energy Resources. *IEEE Access*, 9:46997–47007, 2021. doi:10.1109/ACCESS.2021.3067950.

[46] Solar Panel Energy Output. URL: https://photovoltaic-software.com/principle-ressources/how-calculate-solar-energy-power-pv-systems.

[47] Solar Panel Yield. URL: https://news.energysage.com/what-are-the-most-efficient-solar-panels-on-the-market/.

[48] Solcast. URL: https://solcast.com/.

[49] Voltage control in a smart distribution network using demand response. *Conference Proceeding - 2014 IEEE International Conference on Power and Energy, PECon 2014*, pages 319–324, mar 2014. doi:10.1109/PECON.2014.7062464.

[50] Carbon tracker. URL: https://bulb.co.uk/carbon-tracker/.

[51] Greenhouse gas reporting: conversion factors 2020. URL: https://www.gov.uk/government/publications/greenhouse-gas-reporting-conversion-factors-2020.

[52] 10 households complete output attachment. https://drive.google.com/drive/folders/1QlxvFoP9PAUUwO5uB1hPkp3SJV74KKj2?usp=sharing.