UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE ENGENHARIA GEOGRÁFICA, GEOFÍSICA E ENERGIA

Developing a battery management system for self-consumption systems

Carolina Baptista Crespo

Mestrado em Engenharia da Energia e Ambiente

Dissertação orientada por:
Miguel Centeno Brito

2022

Acknowledgements

I begin by thanking my supervisor, professor Miguel Centeno Brito, for his guidance and availability, as well as his willingness and promptness to supervise my work and suggest an incredibly interesting subject, despite my late request. The chance to work with him was very gratifying, and I look forward to continuing to do so for the coming few years.

I would also like to express my gratitude to Guilherme Luz and Rodrigo Amaro E Silva for their continuing analysis and constructive criticism which certainly helped challenge and guide me throughout this year. The active interest they demonstrated in both my work and my professional development, suggesting resources, scrutinising my work or simply being a willing ear, will not be forgotten and was much appreciated.

A special thank you is also owed to Guilherme Luz once more, but also João Silva and Joana Baptista for welcoming and adopting me as the PhD student that I wish to be but am not yet. The companionship and good humour in room 8.3.33 were instrumental in helping me stay motivated through the hardships, one (silly or serious) conversation at a time.

This work is a part of the Smart Storage Open Platform (SSOP) project, led by Smart Energy Lab, and the data which made it possible was provided by EDP Comercial.

Sustainable Development Goals

Solar energy has enormous potential to ensure widespread access to clean and sustainable energy. Combining solar energy with efficient energy storage solutions in order to overcome its variability will be an essential step to ensure that it becomes an increasingly used energy source.

Thus, this work is aligned with goal 7 of the United Nations Sustainable Development Goals: Affordable and Clean Energy, in particular, with target 7.2, to increase substantially the share of renewable energy in the global energy mix by 2030.[1]

Resumo

A energia transformou-se num recurso essencial para a atividade humana. Atualmente, fontes de energia não renováveis permanecem as principais fontes de energia, mas um esforço global coletivo está a ser realizado de modo a inverter esta situação, aumentando a expressão de fontes de energia renováveis nas nossas redes elétricas. Neste contexto, tornou-se comum a instalação de pequenos sistemas fotovoltaicos em residências privadas para auto-consumo. Inicialmente, estes sistemas eram em larga medida apoiados por subsídios estatais, remuneração elevada por energia solar injetada na rede, etc. Estes incentivos têm vindo a ser reduzidos ou removidos por completo à medida que a tecnologia fotovoltaica se torna cada vez mais acessível. Por este motivo, um melhoramento contínuo dos benefícios garantidos pela instalação destes sistemas é sempre uma contribuição positiva no sentido de manter o incentivo para a sua instalação, e desse modo contribuir para o objetivo mais amplo de um sistema energético tendencialmente renovável. A adição de sistemas de armazenamento de energia, tais como baterias, a sistemas fotovoltaicos residenciais permite uma maior flexibilidade no uso da energia gerada localmente; no entanto, o custo adicional da bateria requer uma estratégia de gestão de energia bem ajustada e afinada de modo a que esta seja uma situação favorável para o utilizador.

Este trabalho divide-se em duas partes: 1) elaboração e teste de modelos de previsão de geração fotovoltaica e de carga residencial e 2) desenvolvimento de uma estratégia de gestão de energia baseada em Reinforcement Learning capaz de resolver o problema de gestão de energia de uma residência com um sistema fotovoltaico e uma bateria para armazenamento de energia, concretamente, a gestão da bateria, incorporando os modelos de previsão previamente gerados, com o intuito de melhorar o planeamento, e o objetivo específico de minimizar a fatura energética do utilizador.

Na parte 1), começou-se pela adaptação de modelos Random Forest (RF) de um trabalho prévio, bem como a elaboração de modelos Artificial Neural Network (ANN) nos mesmos moldes, para a previsão de geração fotovoltaica e de carga para o dia seguinte, comparando ambos um com o outro, e com um modelo de persistência como referência. Enquanto que ambos representaram melhorias face ao modelo de persistência, os modelos RF tiveram um melhor desempenho do que as ANN, tanto para a geração fotovoltaica como para a carga. Em seguida, determinou-se a elaboração de modelos capazes de prever quantidades cumulativas de geração PV e carga, para diferentes horizontes: 1, 3, 6, 12 e 24 horas. O objetivo foi o de ser capaz de providenciar ao sistema de gestão de energia residencial (HEMS) a informação relevante relativa às 24 horas seguintes e necessária para o planeamento, mas de forma compacta, facilitando a interpretação da mesma. Mais uma vez os modelos RF e ANN foram comparados nos mesmos moldes, com as ANN a mostrar melhor desempenho desta vez, para a maioria dos horizontes considerados e de acordo com duas das três métricas.

Na parte 2), estas previsões cumulativas foram incorporadas num modelo de Reinforcement Learning (RL) que foi treinado de modo a realizar o planeamento do armazenamento de energia de uma bateria

residencial. Inicialmente, um modelo de RL puro foi elaborado mas, após este demonstrar resultados pouco promissores, um modelo híbrido RL/baseado em regras foi desenvolvido, capaz de impor ações ótimas em situações em que existe uma ação ótima evidente, deixando que um agente RL escolha a ação nas restantes situações. O modelo desenvolvido foi comparado com dois modelos de referência: maximização de auto-consumo (SCM), um modelo simples baseado em regras que é comummente usado na gestão de baterias, e Mixed-Integer Linear Programming (MILP), um modelo que recebe uma série temporal inteira de dados e retorna a solução ótima absoluta, permitindo assim avaliar o potencial máximo de poupanças possíveis relativamente ao modelo base.

Os modelos foram desenvolvidos e testados com base em dados reais de geração fotovoltaica e carga provenientes de 127 residências localizadas em território português.

De um potencial máximo de 5.9% de poupança para o utilizador mediano, calculado pelo MILP relativamente ao SCM, o modelo híbrido RL/baseado em regras atingiu apenas 1.9% de poupanças utilizando previsões perfeitas (dados reais) e apenas 0.7% usando previsões baseadas numa ANN.

A recomendação final deste trabalho é portanto que o SCM é suficientemente adequado ao contexto português atual a nível energético, e que o sistema híbrido aqui testado não apresenta vantagens suficientes para justificar o seu uso, tendo em conta que se trata de um sistema bastante mais complexo, requerendo custos adicionais. À medida que a transição energética em Portugal progride, é possível que a crescente penetração de geração renovável na rede elétrica leve à alteração de estruturas de preço, etc., sendo que nessa situação é possível que o SCM deixe de ser a estratégia mais adequada, pelo que a situação deverá ser reavaliada no futuro.

Palavras-chave: Sistemas Residenciais de Gestão de Energia, Sistemas Fotovoltaicos Residenciais, Sistemas de Armazenamento de Energia Residenciais, Previsão de Geração PV, Previsão de Carga.

Abstract

Energy has become an essential resource for human activity. Currently, non-renewable energy sources remain the main sources of energy, but a collective global effort is underway to reverse this situation, by increasing the expression of renewable energy sources in our electrical grids. In this context, it has become common to install small photovoltaic systems in private residences for self-consumption. These were initially encouraged largely by state subsidies, high feed-in tariffs, etc., which have begun to decrease or disappear altogether as PV technology becomes more and more affordable. For this reason, continual improvement of the benefits provided by installing PV systems in homes is a positive contribution towards keeping the incentive for their installation, thus contributing to the broader goal of an increasingly renewable energy system. Adding energy storage, such as batteries, to residential PV systems allows for more flexibility in the use of the locally generated energy; however, the added cost of the battery requires a well-tuned energy management strategy in order for this to be a beneficial arrangement for the user.

This work built a hybrid Reinforcement Learning/rule-based energy management strategy aimed at residences with a PV system and a battery as an energy storage system, using Deep Learning to generate cumulative PV and load forecasts for several different horizons, and testing it using real data from 127 households located in Portuguese territory. It also used a Mixed-integer Linear Programming model in order to determine an optimal solution to the energy management problem, thus evaluating the maximum possible potential for cost savings. From a maximum potential of 5.9% cost savings for the median user, the hybrid RL/rule-based system was able to achieve only 1.9% savings using perfect forecasts (i.e, real data), and only 0.7% when using ANN-based forecasts.

Keywords: Home Energy Management System, Residential PV, Residential Energy Storage Systems, PV forecasting, Load forecasting.

Contents

1 Introduction **1**
- 1.1 Context . . . 1
- 1.2 Objectives . . . 2
- 1.3 Contributions of this work . . . 3
- 1.4 Document organisation . . . 3

2 Literature review **5**
- 2.1 Introduction: the Portuguese context for residential PV systems . . . 5
- 2.2 Machine Learning: theoretical fundamentals . . . 7
- 2.3 PV forecasting . . . 11
- 2.4 Load forecasting . . . 16
- 2.5 Home Energy Management Systems . . . 17
- 2.6 Previous work . . . 21
- 2.7 Final remarks . . . 22

3 Methodology **23**
- 3.1 General data description & analysis . . . 23
- 3.2 Forecasting . . . 27
- 3.3 Home Energy Management System . . . 36

4 Results and discussion **49**
- 4.1 Forecasting . . . 49
- 4.2 Home Energy Management System . . . 58

5 Conclusions and future recommendations **69**

References **73**

A Examples of cumulative forecast model performance **79**

B HEMS - additional experiments **83**
- B.1 Using PV forecasts or load forecasts exclusively . . . 83
- B.2 Training with fewer data points . . . 84

List of Figures

2.1.1 Lithium-ion battery price outlook. Source: BloombergNEF[10] 6
2.2.1 Example of a simple neural network with one hidden layer. 9
2.2.2 Agent-environment interaction in a Markov Decision Process. Adapted from Sutton *et al.* (2018) [14] 10

3.1.1 Example data from one dwelling 24
3.1.2 Percentage of NAN data per month (any variable). 24
3.2.1 Dummy example of the scatter plots used later in this work to visualise model performance. 28
3.2.2 Examples of hyperparameter analysis for PV point-forecasting. Each point is one trained model (metrics are calculated for validation data). Outliers have been removed for ease of viewing. 33
3.3.1 Illustration of the quantities considered in each timestep of the MDP, and which time period they refer to. Each point t, $t+1$, ... on the x axis corresponds to a *step* for the MDP, i.e., a moment when the agent chooses an action for the *following* timestep. Note how, at the time of choice, the agent has no information about the following timestep, where the action will take place. This is valid for variables PV_t, L_t, ΔE_t and GI_t. 39
3.3.2 Schematic of the MDP of the proposed solution. 42
3.3.3 Simplified schematic of actor-critic methods. TD error stands for *temporal difference error*, the difference between estimated reward and the actual reward received. Adapted from Zhang *et al.* (2020) [14] 46
3.3.4 Clipped loss in PPO-Clip. r stands for probability ratio between the action under the current policy and the action under the previous policy, and L^{CLIP} is the clipped loss. For positive advantages (left), loss is clipped for large r values, discouraging the agent from making that action much more likely than it was. For negative advantages (right), loss is clipped for small r values, discouraging the agent from making that action much less likely than it was. Image source: Schulman *et al.* (2017) [71] 47
3.3.5 Interaction between modules. 47

4.1.1 Daily profile examples from illustrative dwellings and illustrative days. 49
4.1.2 Model comparison for PV point-forecast 50
4.1.3 Example of the ANN and RF models prediction accuracy on PV point-forecast for one dwelling. The red line marks where perfect predictions would fall. Darker blue represents overlapping points. 51
4.1.4 R^2 of single-point forecast of K_{cs} for all dwellings. 52
4.1.5 Model comparison for load point-forecast 52

4.1.6 Example of the ANN model prediction accuracy on load point-forecast for two dwellings. The red line marks where perfect predictions would fall. Darker blue represents overlapping points. . . . 53
4.1.7 Model comparison for PV cumulative forecast . . . 55
4.1.8 nRMSE comparison for cumulative PV forecasting for individual dwellings, different horizons. On the left, 1, 3, 6 and 24 hour horizons can be seen together; on the right, only the 12 hour horizon. . . . 55
4.1.9 Model comparison for load cumulative forecast . . . 56
4.1.10 Scatter plot for the 24h cumulative load forecast on one dwelling. For this dwelling, nRMSE is 18% and R^2 is 0.1. . . . 57
4.1.11 nRMSE comparison for cumulative load forecasting for individual dwellings, all horizons. 57
4.2.1 Energy cost reductions using a battery . . . 59
4.2.2 One example of two cloudy days managed by SCM and by MILP. Due to scarce PV generation, SCM barely uses the battery. MILP is able to perform pre-charging mostly during the night, when prices are low, for use during higher-price periods. . . . 59
4.2.3 Energy cost comparisons of different tariffs . . . 60
4.2.4 Added cost [%] of using the pure RL model when compared to the RL/rule-based hybrid and SCM. . . . 61
4.2.5 Seed variability (cost difference between min and max costs) for some scenarios, using a tri-hourly tariff. . . . 62
4.2.6 Correlation between performance on the training data and performance on the same data for the same seed. Correlation is computed for each dwelling. . . . 62
4.2.7 Comparison of the results of 1 run and 4 runs through the training data, for the tri-hourly tariff, perfect forecast scenario (RL/rule-based). . . . 63
4.2.8 Seed correlation scatter plots, tested on one dwelling (25 seeds). Each point corresponds to one seed. . . . 64
4.2.9 Cost difference [%] of various HEMS variations when compared to SCM, for bi- and tri-hourly tariffs . . . 65
4.2.10 Cost difference [%] of various HEMS scenarios when compared to using no forecast, for bi- and tri-hourly tariffs . . . 65
4.2.11 Fulfilled potential [%] of various HEMS variations when compared to SCM, defined as the ratio of observed cost savings to *potential* cost savings according to MILP results, for bi- and tri-hourly tariffs . . . 65
4.2.12 Correlation heatmap between the fulfilled potential of several scenarios and forecast nRMSE . . . 67
4.2.13 Examples of the hybrid model performance over two days using perfect forecasts and tri-hourly tariff on different dwellings, in different periods. . . . 68

A.0.1 Example of ANN and RF model prediction accuracy on PV cumulative forecast for all considered horizons, for one dwelling. . . . 80
A.0.2 Example of ANN and RF model prediction accuracy on load cumulative forecast for all considered horizons, for one dwelling. . . . 81

B.1.1 Cost difference [%] of additional scenarios (in yellow) when compared to SCM, for the tri-hourly tariff. Previous results are reproduced here for reference. . . . 83

B.2.1 Cost difference [%] of additional scenarios (in yellow) when compared to SCM, for the tri-hourly tariff. Previous results are reproduced here for reference. 84
B.2.2 Fulfilled potential [%] of additional scenarios (in yellow) when compared to SCM, for the tri-hourly tariff. Previous results are reproduced here for reference. 84
B.2.3 Fulfilled potential [%] of additional scenarios when compared to SCM, for the tri-hourly tariff, vs. number of valid data points available for training. 85

List of Tables

2.6.1 Summary of results obtained on the previous work [3] 22

3.2.1 Features used on each forecast, for point-forecasting [1] 29
3.2.2 Hyperparameters used on RF model (PV and load) 30
3.2.3 Hyperparameters used on ANN models for point-forecasting 33
3.2.4 Features used on each forecast, for cumulative forecasting 35
3.2.5 Hyperparameters used on ANN models for cumulative forecasting 35
3.3.1 Tariff regimes considered, prices from EDP Comercial[59, 60] 37
3.3.2 Summary of state variables on the HEMS environment. 41

4.1.1 PV point-forecast: median of each metric 51
4.1.2 Load point-forecast: median of each metric 53
4.2.1 Median cost difference [%] when compared to SCM 64
4.2.2 Median of fulfilled potential [%] (portion of potential savings that were in fact achieved) 66

Nomenclature

AI	Artificial Intelligence
ANN	Artificial Neural Network
ARIMA	Auto-Regressive Integrated Moving Average
CNN	Convolutional Neural Network
DER	Distributed Energy Resource
DL	Deep Learning
EMS	Energy Management System
ESS	Energy Storage System
EV	Electric Vehicle
GA	Genetic Algorithm
HEMS	Home Energy Management System
kNN	k-Nearest Neighbours
LSTM	Long Short-Term Memory (Neural Network)
MDP	Markov Decision Process
ML	Machine Learning
MLP	Multilayer Perceptron
MPC	Model Predictive Control
NWP	Numerical Weather Prediction
PPO	Proximal Policy Optimisation
PV	Photovoltaic
ReLU	Rectified Linear Unit
RF	Random Forest
RL	Reinforcement Learning

RNN	Recurrent Neural Network
SB3	Stable Baselines 3
SCM	Self-consumption Maximisation
SOC	State of Charge
SVM	Support Vector Machine
ToUA	Time-of-use Arbitrage
V2G	Vehicle to Grid

Chapter 1

Introduction

1.1 Context

In the current global context of climate change, there is an urgent need to reduce CO_2 and other greenhouse gas emissions in order to meet the target defined by the Paris Agreement, signed in 2015, of keeping the global average temperature increase below 2°C compared to pre-industrial levels, or more ideally 1.5°C.

One of the main sectors to decarbonise is the energy sector, where renewable energy is playing a central role. In particular, distributed energy resources such as wind and solar power have contributed not only to decarbonisation but also to the democratisation of the energy system, putting choice in the hands of the consumer and ultimately allowing them to become energetically self-sufficient if they so wish. These new solutions may also allow a reduction of investment needs in the electricity grid, having the potential to reduce, in theory, the guarantee of power required from producers.

However, this decentralised model creates new challenges for the public electricity grid, which must now be able to integrate thousands of *prosumers*, as this integration, if incorrectly handled, is a possible factor of instability for the grid. Additionally, future financial models need to be discussed as well, since, under the current model, the increase in distributed generation may reduce demand from the electricity grid, decreasing the associated revenue but not costs for the grid, and resulting in higher energy prices for traditional consumers.[2]

Another highly relevant sector in decarbonisation is mobility, in which electric vehicles (EVs) will certainly have an important role to play. However, with the increase in the number of EVs in circulation, questions arise about what is the ideal compromise between the needs of users and the investment needs in the electric grid, since these vehicles will represent non-negligible consumption for the grid. Additionally, these vehicles may provide services to the grid itself, through the Vehicle-to-Grid (V2G) concept, where the connection of these vehicles may be bidirectional, allowing the grid to use energy from the vehicles themselves, in return for a fee, in order to balance supply and demand throughout the day.

In the context of distributed energy resources (DERs), the coupling of DERs with energy storage systems (ESS) such as batteries or hydrogen is a possibility. The systems currently in use are, for the most part, managed according to simple rules which are not very flexible. Seeing as these systems have

a relevant degree of variability (the variability inherent to the renewable generation, load variability and variability of energy purchase/sell prices), there is added value in the possibility of introducing flexibility in consumption forecasting. Moreover, as the installed capacity of PV generation increases, this added value provided by flexibility increases as well, as a result of the large amounts of energy produced around the middle of the day by these PV systems — which is also likely to affect price structure.

Given all these factors, energy systems will have increasingly decentralised production, high flows of information and data processing, bidirectional energy flows, mixing the roles of producer and consumer; as well as needs for better and more efficient storage solutions that are able to satisfactorily respond to this growing degree of complexity.

1.2 Objectives

Considering this context, this work then aims to create a Home Energy Management System (HEMS) to manage self-consumption systems with PV coupled to batteries, that is able to respond to current challenges and prioritise the consumers' interest, namely, their financial gain. HEMS systems currently in use aim at maximising self-consumption, a logic that may quickly become outdated, as variable energy prices become the norm, and there is a growing need for flexibility from both the grid and consumers. For example, it may be more beneficial for a consumer to, at a given moment, charge their battery for later use and use energy from the grid now if the price is expected to increase later; or even to use grid energy to charge their battery, at off-peak hours when the energy price is low, in order to use it later when grid prices are higher.

A system suited to this context will need to be able to make sufficiently reliable forecasts of both load and PV generation in a short-term (day-ahead) horizon, as well as take into account the variable electricity prices, in order to be able to maximise financial gains for the consumer.

With this in mind, this work will be guided by the following question:

How to create a HEMS for PV+battery self-consumption solutions that is fit for the current context and prioritises the needs of the user?

More specifically, the following research questions are formulated:

1. Which algorithm and methods are best suited to the task of forecasting load and PV generation for residential systems?

Artificial intelligence (AI) is expected to be a useful tool to perform load and PV generation forecasting, and it is currently a frequently used tool in research on this topic, and with encouraging results. Previous work on this project has already tested using a Random Forest (RF) algorithm for both load and PV generation forecasting[3], and this work now intends to test an Artificial Neural Network (ANN) for the same task, with the objective of choosing the algorithm that shows the best performance for each case.

2. How to best make use of the load and generation forecasts so as to enable an optimised management of the battery system?

Forecasts are important tools to enable the energy storage management system to make decisions based on data that is as reliable as possible, but it will then be necessary to test various management

strategies to see which best meets the desired objectives. For the purposes of this work, Reinforcement Learning (RL) will be used in order to train a model which, ideally, will be able to make optimised decisions aiming at maximising the financial gain of the customer.

This work is part of project Smart Storage Open Platform (SSOP), led by Smart Energy Lab. EDP Comercial has provided the data which will be used on this dissertation, for both the forecasting and the energy management steps, corresponding to PV generation and consumption data from 193 Portuguese households. This data was anonymised and is devoid of additional information on these households, such as location and number of occupants.

1.3 Contributions of this work

The main contributions of this dissertation and its proposed methodologies are considered to be the following:

- Methodologies tested and verified on large amounts of data, both in terms of different dwellings and long periods of time (over one year in most cases), unlike many papers which test their methodologies on one day, one week, etc. This ensures that the performance of the proposed methods is evaluated on a large array of situations, as opposed to specific situations with large room for improvement, which may unintentionally bias the results.
- Integrated approach, performing both forecasting and energy management on the same datasets, which allows for the evaluation of the true impact of forecast error on the end result, whereas most papers perform only either one or the other.
- Proposed structure of cumulative forecasts, not otherwise found in literature, in order to provide the HEMS with the relevant information in a compact format.
- An in-depth analysis using different scenarios in order to evaluate the impact of tariff choice, forecast error and energy strategy selection on the energy bill.

1.4 Document organisation

This dissertation is organised into five different chapters. Chapter 1 introduces the framework establishing the necessity to develop this work, states the goals of this work as well as its main contributions, frames it according to UN's Sustainable Development Goals and outlines document structure.

A brief literature review was conducted in chapter 2, which begins by summarising the current Portuguese context for residential PV systems, and then addresses the state-of-the-art of energy management and forecasting, as well as of the Machine Learning tools used here, namely Artificial Neural Networks (ANNs) and Reinforcement Learning (RL).

Chapter 3 lays out the methodology applied here, focusing on describing model implementation, as well as the details of the different scenarios employed.

Chapter 4 displays the main results of the simulations, drawing meaningful comparisons, as well as discussion and analysis stemming from said results.

Finally, chapter 5 includes a summary of the most relevant results, as well as the main conclusions to be drawn, and additionally some recommendations for future works in this field.

Chapter 2

Literature review

On this chapter, the Portuguese context and prospects for self-consumption and small energy storage systems is briefly analysed. The basics of Machine Learning are laid out, with a special focus on Artificial Neural Networks and Reinforcement Learning, which will be used in this dissertation for forecasting and energy management, respectively. Next, a state-of-the-art analysis is done on PV forecasting, load forecasting and the management of energy storage systems, which will help guide decisions made aiming at constructing better models. Finally, the key aspects of the previous work on this project, a master's thesis by Emanuel Parracho, are summarised.

2.1 Introduction: the Portuguese context for residential PV systems

In Portugal, feed-in tariffs for small PV systems (smaller than 3.68 kW) were introduced in 2008 at 0.65€/kWh[4]. From 2015, feed-in tariffs were gradually cut, since it was considered that the technology had reached a greater level of maturity, and the new tariff for PV energy from small generation injected into the grid is based on the monthly average energy price on the Iberian energy market (specifically, 90% of that average).[5] This is a much lower price than before, encouraging self-consumption and discouraging oversized systems, since exporting energy into the grid yields a significantly lower tariff than the average purchase electricity price.

Considering this regime, S. Rodrigues *et al.* (2015) [6] performed an economic analysis of PV self-consumption in Portugal, and found that, as of 2015, 100% self-consumption was generally more profitable than exporting part of the energy to the grid. It also found that a PV system without a battery bank seemed to be more profitable than a PV system connected to one. Two years later, in 2017, Camilo *et al.* (2017) [7] did a similar study and found that battery storage was not economically viable at all in Portugal for all considered system configurations (it yielded a negative NPV).

Even more recently, Foles *et al.* (2020) [8] compared different scenarios for residential PV systems for self consumption, with and without storage, for three different locations in Portugal. For the storage scenarios, the energy management system considered was the simplest one, self-consumption maximisation, where priority is given to immediate self-consumption, followed by battery storage, followed by grid injection. The study found that, as of 2020, regardless of location, a grid-connected standalone PV system exporting surplus generation to the grid was the most profitable scenario. It also found that PV + battery systems were only profitable for the largest PV installed capacity. However, the authors conclude

the study saying that a) it is expected that battery system costs will decrease in coming years, increasing the profitability of the PV + battery scenario, and b) the use of an intelligent energy management system is also likely to improve results for this scenario. Regarding point a), in fact, a 2017 report by the International Renewable Energy Agency (IRENA) concluded that the rapid deployment of new battery storage technologies have led to considerable cost reductions, particularly with the case of lithium-ion batteries, and estimated that by 2030 the cost of battery storage could fall by 50 to 60%, when compared to 2017.[9] 2018 Battery Price Survey by BloombergNEF [10] reached a similar conclusion, as seen on Figure 2.1.1. This report predicted prices under $100/kWh by 2024. A more recent report by BloombergNEF, Electric Vehicle Outlook 2022 [11], mentions how the timing for this achievement has now become somewhat more uncertain, due to the rise in raw material prices, which could delay the $100/kWh milestone by a couple of years.

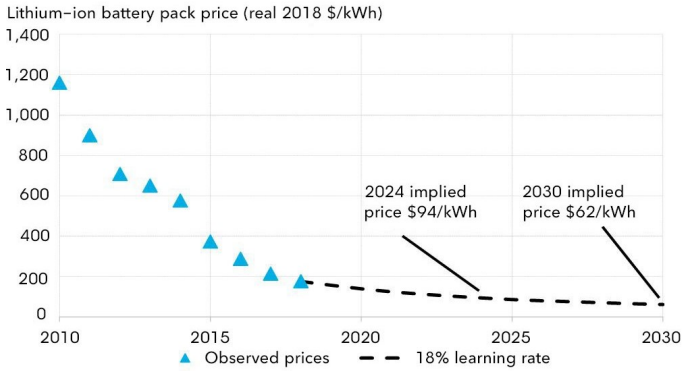


Figure 2.1.1: Lithium-ion battery price outlook. Source: BloombergNEF[10]

Further, C. Villar *et al.* (2017) [12] studied PV self consumption in the Portuguese context. It concluded that, in Portugal, in order to maximise profitability, PV systems will tend to be undersized. This is, once again, because the remuneration for surplus energy injected into the grid is quite low, leading to a strong correlation between profitability and the share of self-consumption, and discouraging grid injection. The study further concluded that the accommodation and retail sectors are the ones most suitable for PV self-consumption. The suitability of the residential sector is varied, as demand profiles vary a lot between residences. Since profitability depends on the share of self-consumption, PV is most suited for demand profiles more congruent with PV generation profiles. The authors note that storage and demand response should be encouraged as solutions for the mismatch between demand and generation profiles.

This work will consist of two essential tasks: a forecasting task, and an energy management task. Machine Learning is increasingly used to solve these types of tasks, and will be used as well in this work. In the next section, the theoretical fundamentals of the Machine Learning aspects relevant to this work are laid out.

2.2 Machine Learning: theoretical fundamentals

Machine learning (ML) is an umbrella term for the ways a computer may learn to do a task without being explicitly programmed to do so. This is done through the analysis of large amounts of data: a ML model trains on data and learns to recognise patterns in that data, and in doing so, it becomes able to apply that knowledge to different data in order to solve tasks.[13]

There are three main types of ML: supervised, unsupervised and reinforcement learning.

Supervised learning consists of classification or regression tasks. In classification, we have a set of labelled data, containing different samples for which a series of characteristics, or features, are known, as well as the class to which the sample belongs, also called target. The model learns the relationships between the features, and learns how to predict the class (for classification) or a numeric value (for regression) of unlabelled samples based on those features.[13]

Unsupervised learning uses unlabelled data. The model does not learn to assign a sample to a predetermined class — instead, it learns how to divide samples into groups, or clusters, based on the degree of similarity between them.[13]

In *reinforcement learning*, the model learns how to perform a task by trial and error. Each outcome of one or more actions is associated with a certain cost or reward, and the model learns which actions to take by seeking to minimise cost or maximise reward. These are defined through a cost function or a reward function.[14]

One of the most important concepts in ML is *generalisation*. A good ML model must be able to generalise, that is, to apply what it has learned to new data, and be able to achieve good performance in doing so. For this to be possible, the data used for training must be representative of the data the model will encounter in the future. A possible cause for poor model performance may be *overfitting*. This is when a model learns the training data too well, including the noise, and becomes unable to generalise to future data. In order to be able to detect overfitting, model performance must be tested on a separate *testing set*, which consists of samples the model has never seen before, and is separate from the *training set*, on which the model was trained. Alternatively, *underfitting* could also occur, if the model stops training too soon, before being able to learn the underlying relationships in the data. Finding a balance between the two is one of the challenges in using ML, and is often called the *bias-variance dilemma*: a low-complexity model is less subject to variability due to noise in the training data, but is more likely to introduce systematic error, or bias, by failing to account for a set of factors. On the other hand, a high-complexity model is not prone to systematic error, but may have non-systematic errors due to variance in the training data.[13]

2.2.1 Random Forest

Random Forest (RF) is based on decision trees. These can be used for both regression and classification tasks. Decision trees work through a top-down approach by dividing the predictor space (that is, the set of possible values for all predictors $X_1, X_2, ..., X_p$) into distinct non-overlapping regions $R_1, R_2, ..., R_j$, and making the same prediction for all observations that fall in a given region. The splits are made based on training data with the aim of minimising a chosen cost function.[15]

One problem with decision trees is high variance. This means if we train decision trees on different subsets of the same dataset, it is likely that the decision trees obtained will be quite different from one another. RF attempts to solve this problem by building multiple decision trees. Each decision tree is trained on a different bootstrapped sample out of the dataset. For each each tree, only a random subset of m out of the p predictors is available for each split. This prevents very strong predictors from completely overshadowing moderately strong predictors. This process is repeated N times, yielding N trees. Then, each tree makes a prediction for each sample, and, for regression tasks, the mean of the predictions is taken as the final prediction.[15]

2.2.2 Artificial Neural Networks

Artificial Neural Networks (ANNs) are subset of machine learning models, often called Deep Learning (DL). They are inspired by the way human brains operate. An ANN consists of a system of several layers of *neurons*, or nodes, interconnected with each other. These layers include the input and output layers, as well as one or more hidden layers. Each neuron in a layer, with the exception of the output layer, is connected to every neuron on the following layer. Each of these connections has an assigned weight, ω_{ij} (where i is a neuron on the nth layer and j a neuron on the $(n+1)$th layer), and it is these weights which are (automatically) adjusted while training the model in order to produce the desired output. Each neuron also has a bias, b, which is simply another name for a scalar which is added to the weighted sum, and is also adjusted during training. Next, each weighted sum is fed into an *activation function*, which is a nonlinear function (some commonly used activation functions are the *sigmoid* function, *ReLU*[1] and hyperbolic tangent, or *tanh*). This introduces a nonlinearity — this function is what allows neural networks to produce the impressive results they are able to produce, since without them, the whole network could be represented by a matrix and would just amount to a linear transformation.[15]

[1] ReLU stands for Rectified Linear Unit, and is the function $f(x) = x^+$. Its use has become more and more widespread in ANNs in recent years, as it solves a number of problems other activation functions were known to have, such as the *vanishing gradient problem*.

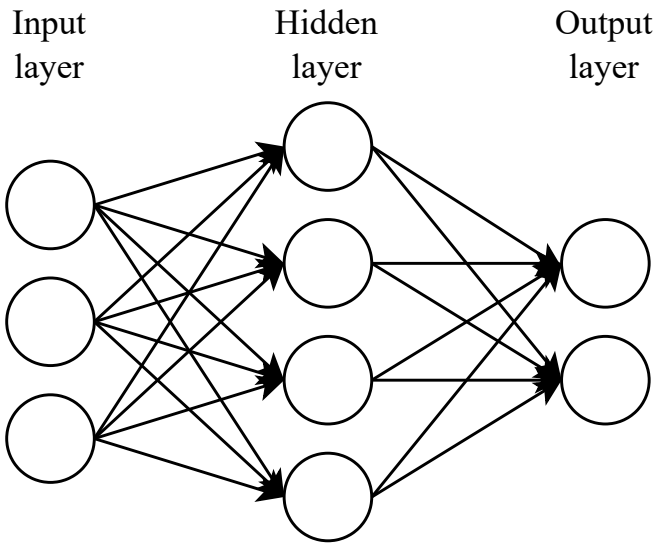


Figure 2.2.1: Example of a simple neural network with one hidden layer.

The value of the kth neuron on layer m is given by Eq. 2.2.1, where n is the previous layer with K_n neurons, $A_i^{(n)}$ the value of each neuron on layer n and g is an activation function.[15]

$$A_k^{(m)} = g\left(b_k^{(m)} + \sum_{i=1}^{K_n} \omega_{ik}^{(m)} A_i^{(n)}\right) \tag{2.2.1}$$

The goal of training an ANN is to find the weights and biases which enable the network to find a function $\hat{y}(x)$ that is a good approximation an unknown function $y(x)$.[16] In order to aid this process, a *loss function* is defined. There are several types of loss functions, but in general, a loss function is a function of the distance between real values y and predicted values $\hat{y}$. Loss functions allow us to ascertain whether or not training is improving the ANN — if the loss function is not decreasing, the ANN is not improving (however, the opposite is not necessarily true). For this we must further split the training dataset — used on all supervised learning tasks — into separate training and validation sets. While the training set is used to update the network, the validation set is used to check whether the model is in fact improving. If training loss is decreasing but validation loss is not, then the model is overfitting on the training set. Commonly used loss functions for regression tasks include mean squared error and mean absolute error.

2.2.3 Reinforcement learning

Reinforcement learning (RL) is a type of ML where an *agent* trains by trial and error in order to learn how to make decisions. The goal is for the agent to make the best possible decisions in order to maximise a numerical reward signal or, more precisely, *expected future reward*.

RL problems are most often framed as Markov Decision Processes (MDPs). In these, there is a set of *states*, s, in which the system may be. In each state, the agent may take an action, a, which may lead it to another state s', or cause it to remain in the same state. Each state/action pair is awarded a certain reward based on the outcome, which depends not only on the state and action but also on the environment. These rewards inform future decisions taken by the agent, as it will seek to maximise accumulated reward over

time.[14]

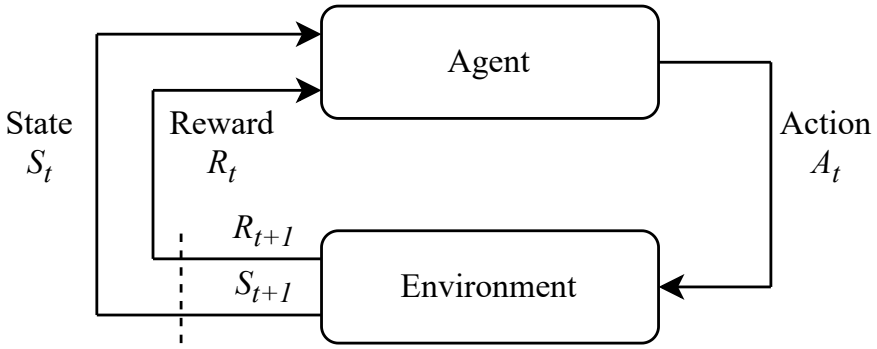


Figure 2.2.2: Agent-environment interaction in a Markov Decision Process. Adapted from Sutton *et al.* (2018) [14]

Aside from the agent and the environment, there are four key elements of a RL system: *policy*, *reward signal*, *value function* and, optionally, a *model*.

A *policy*, π is essentially a mapping from states to actions to be taken when in those states, or, more accurately, to probabilities of selecting each possible action. It is the most immediate factor in determining system behaviour. If the agent is following policy π at time t, then $\pi(a|s)$ is the probability that a is the action taken if s is the current state. Optimal policies are denoted π^*. It is generally not possible to find the optimal policy due to the high computational cost and memory requirements this would demand (as well as, when applicable, data availability — such as in this work), along with uncertainty associated with the environment, but there are means to approximate one and arrive at a well-performing policy.[14]

On each state transition, the agent receives a signal called a *reward signal*, r, whose value changes. The reward system's purpose is to inform the agent on whether the policy needs to be altered: if an action selected by the policy has low reward, the policy may be altered for similar situations in the future in order to improve reward.[14]

The agent's sole objective is to maximise reward over time. The *value function* is what allows it to do this — while rewards are more immediate, the value $v(s)$ of a state s is the total amount of reward which the agent can expect to accumulate over the future, starting from s. This is what allows the agent to be able to choose an action which will lead to a low-reward state, as long as in the following timesteps this is expected to grant the system a higher reward.[14]

A *model* of the environment is something that mimics the behaviour of the environment (i.e, describes the dynamics of state transitions). For example, given a state and action, a model might predict the most likely next state of the system. Methods for solving RL problems which use models are called model-based, and they allow the system to perform *planning* and move past the most basic trial and error approach.[14]

We say a task is *continuing* if the agent-environment interaction continues indefinitely, as opposed to breaking into identifiable episodes.[14] The task at hand for this work is a continuing task, as it will always seek to maximise the economic gain of the user throughout an indefinite future time period.

The value function must seek to maximise expected return. Return G_t is defined as a function of the reward sequence, and for continuing tasks it is defined by Eq. 2.2.2, where γ is the discount rate, which

has a similar purpose to its counterpart in economics: to attribute higher value to sequentially closer rewards.[14]

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... = R_{t+1} + \gamma G_{t+1} \tag{2.2.2}$$

2.3 PV forecasting

Conventional power sources, such as those based on fossil fuels, are what is called dispatchable — that is, power generation is adjustable as needed. On the other hand, renewable sources such as wind and PV are variable, that is, their production is dependent on factors outside of our control, and is difficult to predict. PV variability has two sources: the apparent movement of the Sun, and cloud-induced changes. While the former is slow and obeys well-known mathematical equations (thus being easy to predict with precision), the latter are highly stochastic in nature and can cause significant changes to output in a matter of seconds.[17, 18] From the point of view of grid operators, this variability is a concern, as the balance between generation and consumption must be maintained at every moment, and unanticipated changes in renewable generation may strain the grid.[18]

With the increase in grid penetration of PV, this variability becomes an increasingly relevant issue, in order to ensure grid stability and optimal unit dispatch from an economic point of view. Electrical generation units must be committed and scheduled in advance by the grid operator. Due to the variability of renewables, a number of units with fast ramp rates (whose power output can greatly increase within a short timespan) are required to be on "standby" at any given moment. These and other concerns have motivated a high volume of research seeking to build better PV forecasting models since, at a grid level, a good PV forecast is able to reduce the uncertainty surrounding the power output of PV power plants, and thus allows for the reduction of the number of units on "standby", consequently reducing operation costs.[19] Due to these concerns, much of the research on PV forecasting is focused on large PV power plants, as opposed to smaller rooftop systems.

For large-scale PV power plants, their geographical dispersion tends to have a smoothing effect on variability, as not all panels of the power plant are subject to the same conditions simultaneously. This means that factors such as smaller, isolated clouds will not have a very large effect on overall generation.[18] However, for smaller systems, such as self-consumption systems, variability may be much greater and sharper, as even a small cloud may have a large effect — the system may go from full production to null, and back to full in a couple of seconds.[17] This means that the impact of stochastic changes is larger in smaller systems, making the forecasting of these systems more challenging and prone to higher uncertainty. This is not as large of a problem for the grid, which perceives these systems in aggregated form (with this aggregation generating a smoothing effect due to different orientations, geographical areas, etc.), but it is a problem for a HEMS, which manages a single small PV system.

There are many different types of forecasting models, in a large range of complexity levels. Some of these are elaborated upon in the following sections.

A forecast can be deterministic or probabilistic in nature. In the first case, the forecast outputs a single value; in the second, it outputs a probability distribution. Most PV forecasts, traditionally, are deterministic, as deterministic models are simpler to produce and their performance simpler to assess. However, an increasing number of studies is producing probabilistic forecasts, which enables better risk

assessment and decision making.[19]

2.3.0.1 Forecast horizon

Forecast horizon is one of the main ways in which a forecast may be classified. It is the length of time into the future for which the forecast makes its predictions. PV forecasts with different horizons serve different purposes, and may need to use different types of input variables, as well as different types of models. Intra-hour forecasting, or nowcasting, ranges from a few seconds to 1 hour. Its main function is to detect ramps due to the passing of clouds. It is most relevant for large power plants and grids with high PV penetration, and its purpose is to enable the scheduling of spinning reserves so as to ensure grid stability. The research frontier in nowcasting at the moment lies on sky imaging to detect cloud movement. Intra-day forecasting ranges from 1 to 6 hours and is mostly important for power system operators who manage multiple load zones.[19]

Forecasts from 6 hours to day-ahead are mostly used, on a grid level, for planning and unit commitment. Antonanzas *et al*. (2016) [19] found that the forecast horizon where most research has been done is day-ahead. This is because energy is currently, for the most part, traded on day-ahead markets, meaning planning and unit commitment on a grid-wide scale is done one day in advance.

Horizons of two or more days ahead are used for unit commitment, planning and asset optimisation, as well as for scheduling of power plant maintenance (generally scheduled for when generation is expected to be low). Compared to other horizons, less research is found for these longer horizons.[19]

As the previous work in this project used day-ahead forecasts in order to test model performance[3], this work will begin by using that same horizon in order to draw a first comparison between Random Forest (used by [3]) and Artificial Neural Network (developed in this work) models. The next step, however, will develop cumulative forecasts (predicting the total generation between the present moment and a point which lies a set amount of hours in the future), for five different horizons, spanning from 1 to 24 hours.

While historical data has proven to have a higher importance in shorter-horizon forecasting, Numerical Weather Prediction (NWP) data, which is not available in this work, has a higher importance for longer horizons.[19, 20] Therefore, under these circumstances, a better performance is expected of shorter horizons rather than longer ones.

2.3.1 Persistence models

Persistence models are the simplest and, for this reason, are commonly used as a benchmark for other models. These models assume that certain variables (such as irradiance, power output, clear sky conditions, etc.) remain the same between present time t and $t+1$.[19]

The simplest persistence model, naive or dull persistence, simply assumes that a given variable will remain the same between consecutive timesteps (e.g., the power output 1 hour from now will be the same as the current power output). However, this approach is only adequate for a stationary time series — one whose mean and variance do not change over time — which is not the case with PV generation.[21] A more adequate variation of this would be to assume that the power output at any given time is the same as the power output of the previous day at the same time, as the sun is expected to be in (approximately)

the same position at those two points.[19]

Another approach (stochastic component persistence) is to decompose PV generation in a trend component and a random component, where the trend component is usually the clear-sky production, and the stochastic component is the cloud-induced changes.[21] We then assume the stochastic component remains the same, while the trend component can be computed according to Eq. 2.3.1.

$$\hat{y}(t+T_H) = y_{cs}(t+T_H) + y_{st}(t) \tag{2.3.1}$$

Yet another approach to persistence, clear-sky index persistence, is to take the clear-sky index (Eq. 2.3.2) and apply the concept of persistence to this index — that is, assume the clear-sky index will remain the same between timesteps (Eq. 2.3.3).[21]

$$k_y(t) = \frac{y(t)}{y_{cs}(t)} \tag{2.3.2}$$

$$\hat{y}(t+T_H) = \begin{cases} k_y(t) y_{cs}(t+T_H), \text{ if } y_{cs}(t) \neq 0 \\ y_{cs}(t+T_H), \text{ otherwise (at night)} \end{cases} \tag{2.3.3}$$

Persistence works well for very short horizons, but its performance drops sharply for longer horizons, e.g. for a day-ahead horizon, in broken cloud conditions, it will predict the cloud's effects (drop in generation) at the exact same time, which is unlikely to happen in reality, meaning it will generally fail twice: it will predict a drop in generation when there is none, and fail to predict one when it does occur.

2.3.2 PV performance models

PV performance models, or physical models, rely on numerical weather predictions (NWP) and physical equations predicting the behaviour of PV plants in order to forecast PV generation. The main advantage of these methods is that they do not require historical data, meaning the forecast may be computed prior to plant/module installation. However, its main disadvantage lies in the lack of sufficient temporal and spacial resolution in NWP, meaning the resulting forecast can be much less accurate than statistical methods.[19]

2.3.3 Statistical models

Statistical, or data-driven, PV forecasting models rely on extracting relationships between historical data points in order to predict future generation. Statistical models consistently outperform persistence and physical models in PV forecasting.[19]

Many studies incorporate hybrid models, combining two or more techniques, either all statistical or statistical and physical. These models tend to show better results than stand-alone techniques, but are also by definition more complex. An example of a physical statistical hybrid model is Dolara *et al.* (2015) [22], where a physical clear sky radiation model computes the maximum available solar radiation, which is then fed into an ANN, along with other features. The ANN then predicts PV generation. As for

an example of a hybrid statistical model, VanDeventer *et al.* (2019) [23] combines a Genetic Algorithm (GA) with a Support Vector Machine (SVM) model, managing to greatly improve prediction accuracy when compared to the simple SVM model.

Variables used in a data-driven approach can be exogenous or endogenous in nature. Exogenous variables are independent and exterior to the system. These include irradiation, temperature, etc. Endogenous variables depend on the system itself, and in this context usually consist of historical generation data. Antonanzas *et al.* (2016) [19] observed the following trend: the larger the forecast horizon, the more studies tend to use exogenous as well as endogenous variables. In particular, for day-ahead forecasts, 79% of analysed studies used exogenous variables. As examples: Agga *et al.* (2021) [24] uses variables such as windspeed, temperature, humidity and cloud cover, and Long *et al.* (2014) [25] uses dew temperature, humidity, insolation time, wind speed, precipitation and radiation.

Nespoli *et al.* (2019) [26] compared two models: one (1) is a simple ANN, while the other (2) is a hybrid model combining an ANN with a physical model (which uses endogenous variables). Case 1 took as inputs only data from the previous day (average irradiance, average temperature and average power output), while case 2 (the hybrid model) took as inputs the theoretical solar radiation from a clear sky model and a weather forecast. In both cases, the data was divided into sunny or cloudy days. The results showed that both models performed significantly better for sunny days, which was expected, due to the high variability inherent to cloudy conditions. Additionally, model 2 outperformed model 1 on cloudy days, but on sunny days model 1 outperformed model 2. The authors speculate that the more inconsistent performance of model 2 could be due to the quality of the weather forecast, and the fact that the PV site location and the weather forecast location are not exactly the same. This shows that, depending on the situation, the added complexity of a hybrid physical and statistical model may not always pay off.

Long *et al.* (2014) [25] performed a similar comparison study with four of the most common forecasting techniques: ANN, SVM, k-Nearest Neighbours (kNN) and multivariate linear regression (MLR), for two scenarios: in one, only time series data was available; in the other, exogenous variables (meteorological parameters) were also available. The forecast horizons spanned from intra-day to up to 3 days ahead. While on the former, MLR outperformed the other techniques in most time horizons, on the latter, ANN generally outperformed the others, most notably for day-ahead and two days ahead predictions.

These comparison studies are useful in order to understand which model might fit each situation best, as it is not trivial to draw comparisons between models designed by different authors and tested on different datasets and under different conditions. Although it compares only two forecast models, the present work aims to do something similar.

In the present work, only endogenous variables are available (and exogenous variables cannot be obtained, as the locations are anonymous), which will certainly impair model performance when compared to studies which use exogenous variables. On the other hand, even if we did know the locations of the dwellings, there is no guarantee that exogenous variables such as NWP variables would improve the model, as it is likely that on many dwellings there would not be a meteorological station with sufficient geographical proximity for its data to be sufficiently accurate for the precise location of the dwelling.

Regression models are some of the simplest statistical models. The very simplest would be a basic linear regression, but there are many variations. Regression models can be classified into linear or non-linear, and stationary or non-stationary. Stationary models are those whose mean and variance do not change over time.[21] This is not well suited for PV forecasting, as seen in 2.3.1. As described there,

the stochastic component of PV generation, cloud-induced variability, may be seen as a stationary time-series, and so a stationary model can be applied to PV forecasting. Besides simple linear regression, auto-regressive (AR) models, moving average (MA) models, variations of these such as auto-regressive integrated moving average (ARIMA) and many others can be used.

Kudo *et al.* (2009) [27] compared a direct and an indirect regression methods for PV forecasting. The direct method uses historical generation data as well as NWP variables to predict the power index (generated power divided by extraterrestrial solar radiation) using regression, and then calculates output power; the indirect method uses NWP variables to predict irradiation index (irradiation divided by extraterrestrial solar radiation) and then uses a physical PV performance model to predict generated power. Results from testing at a PV power plant in Japan showed that the direct method performed better; however, the indirect method does not need historical generation data and therefore holds the advantage that it can be used from the initial stages of power plant operation.

Machine Learning (ML) models combine mathematical and statistical models. The use of machine learning in both PV and load forecasting is quite widespread. Some examples of ML models used in PV forecasting include Artificial Neural Networks[22, 24, 26, 28–31], which were covered in 2.2.2, Random Forest (which was used on the previous work on this project[3] — see section 2.6)[31], k-Nearest Neighbours (kNN)[28], and Support Vector Machines (SVM)[23, 32, 33], but there are many more. Antonanzas *et al.* (2016) [19] reviewed a large number of forecasting studies and found that machine learning methods are used in more than half of the papers analysed. ANNs, specifically, are the most common method, making up 24% of the total.

Pedro and Coimbra (2012) [28] compared four different frequently used models: Auto-regressive Integrated Moving Average (ARIMA), kNN, ANN, and GA/ANN. The models used only endogenous inputs. For both forecast horizons considered (1h and 2h), GA/ANN was the best-performing model, followed by the simple ANN. Only kNN performed worse than the benchmark persistence model. Based on the models' different performance within different seasons, the authors comment that developing different models for different seasons could improve the assessment of forecasting skill, by allowing researchers to compare model performance under more "ideal" conditions of low variability and under more adverse conditions of high variability. In this study, the GA was used to optimise parameters such as the number of layers in the network, number of neurons per layer, input variables, and distribution of data between the training and validation sets.

Several forecasting studies that use ANNs take the approach of first classifying days into different day types, and forecasting accordingly. Nespoli *et al.* (2019) [26] splits the days into two sets: those with mean irradiance larger than 150 W/m^2, and those smaller. Mellit *et al.* (2014)[29] classifies days into sunny, partly cloudy and overcast, and uses a different ANN for each day type. Chen *et al.* (2011) [30] classifies days into sunny, cloudy and rainy, also using a different ANN for each day type.

Agga *et al.* (2021) [24] used CNN-LSTM and ConvLSTM neural networks (both models which incorporate convolution operations into a LSTM) to forecast PV power plant production for varying forecast horizons, ranging from one day to one week. Separate tests were run using only endogenous data (historical production time series) as inputs, and then using both endogenous and exogenous data. Results showed that using endogenous data only generally yielded better performance than including exogenous data, with the single exception of the CNN-LSTM model for a 7-day forecast horizon. Additionally, for a forecast horizon of 1 day using endogenous data only, the three models' performance was very similar,

although for longer horizons the convolution models did generally outperform the simple LSTM. This shows that, for short horizons, more complexity does not necessarily yield better results.

2.4 Load forecasting

There are two main approaches to load forecasting in buildings: the forward modelling approach, and the data-driven approach. Forward modelling is often used in new buildings, involves specialised software and requires the input of a large number of parameters. This makes it so that it is generally a cumbersome process. The data-driven method can be applied to existing buildings for which load time series are available, and uses past data in order to build a load forecasting model.[34] This method is able to achieve a high precision and its implementation is much simpler, which is why this approach will be taken in this dissertation.

Many of the aspects mentioned in the previous section regarding PV forecasting are also true for load forecasting, which is why this section will be shorter. Persistence models may also be used as a benchmark, but with some differences: load is not decomposable into a trend and a stochastic component. We may use naive persistence, and so assume that the load at time $t+1$ equals the load at time t, as long as $t+1$ is one day ahead of t, since load profiles, like PV generation profiles, tend to present a distinctive daily shape. A more elaborate version, N-day persistence, would take the load at time t on a given day to be the average of the loads at time t on the N previous days. Furthermore, we could consider only days of the same type as the forecasted day, e.g., consider only the N previous Mondays when forecasting a Monday.[35]

The use of ANNs for load forecasting dates back to the 1990s [36, 37], and nowadays ANNs are one of the most commonly used methods for load forecasting, because they generally tend to perform well at this task.[38] In the following paragraphs, a brief literature review on the topic of load forecasting is conducted. All of the following reviewed articles used ANNs. These were not handpicked: it is because, while ANNs are indeed very prevalent in PV forecasting, they are even more so in load forecasting, which is likely a testament to their good performance, but may also leave opportunity for further experimentation using other methods.

Zhao *et al.* (2018) [39] used a Radial Basis Function (RBF) Neural Network. RBF NNs use radial basis functions as activation functions. These are functions whose output depends on the distance between the input and one fixed point. In this paper, those fixed points, or cluster centres, were determined by applying a k-means clustering algorithm to the training data, which, according to the authors, allows for more accuracy than otherwise artificial or random selection of cluster centres, by ensuring that cluster centres are relatively dispersed and cover all the data. When tested on historical data of a court, this method achieved slightly better performance than a simple ANN.

Recurrent Neural Networks (RNNs) are ANNs which are said to have *memory*, that is, they are able to use the network's previous predictions as input for subsequent predictions, making them suitable for processing sequential data, namely for tasks like speech recognition, but also time series analysis[16]. Cai *et al.* (2021) [40] used a long short-term memory (LSTM) ANN for short-term load forecasting (24 h ahead). It was tested on a load time series from a substation, which means it includes many different loads and of various types, giving it a smoothing effect when compared to load from a single residence. Three was found to be the optimal number of layers for the LSTM, and it was also found to

outperform a simple Multilayer Perceptron (MLP, or a simple ANN). The authors mention, however, the long computation times required by this method.

Cai *et al*. (2020) [41] used a sample weights method coupled with an ANN, where the M sample days most similar to the forecasted day (smallest Euclidean distance) are used to train the network, and samples most similar to the forecasted day are assigned larger weights. Compared to a model trained simply on the M adjacent days to the forecasted day, the sample weights method yielded better results. It also yielded better results when compared to a model trained on the M most similar days, but without the assignment of sample weights. The authors argue that this is because this traditional approach tends to ignore data which may be more similar to the forecasted day, despite being more distant in time. In addition, the sample weights model proved better than other methods used in load forecasting, such as random forest, support vector machine, LSTM networks and linear regression. In particular, it was found that the LSTM had trouble dealing with holiday forecasting, whereas the sample weights method had a better ability to find days similar to the forecast day. The model was tested on data of Guangdong Power Grid referring to a 1-month period in January 2017.

No instances of cumulative forecasting, of either load or PV, as will be applied on this work, were found in the literature.

2.5 Home Energy Management Systems

Home Energy Management Systems (HEMS) are used in residential settings to monitor and control Distributed Energy Resources (DER) as well as Energy Storage Systems (ESS), and schedule electric vehicle (EV) charging, if these are present. These systems may also facilitate demand response, either by informing the users of current prices, renewable generation, etc. so that they may make changes themselves, or by performing appliance scheduling, when smart appliances are present. HEMS may or may not be coupled with sensors such as occupancy or movement sensors.[42] The goal of a HEMS is essentially to help optimise energy consumption with one or more goals in mind, be it energy efficiency, cost reduction, grid reliability or others.[43]

In the current context, wherein, in several countries, feed-in tariffs have been decreasing at the same time that electricity retail prices have risen, the most commonly used energy management strategy is a simple rule-based strategy, self-consumption maximisation (SCM). An alternative strategy to this is time-of-use arbitrage (ToUA), in which the battery is pre-charged with grid power during off-peak periods, for use during peak periods, when the electricity retail prices are high. Both of these are heuristic approaches and do not explicitly seek to minimise cost.[44]

According to J. Solano *et al*. (2018) [45], for cases where the energy bill has high variable charges (energy cost) and low fixed charges (power cost), which is the case for Portugal, SCM tends to be the best rule-based energy management strategy. However, a third rule-based strategy, peak-shaving, may be useful in cases with high fixed charges and lower variable charges. This is the case for countries such as Spain, and may tend to become more common in power grids of the future, as a way to avoid the so-called "utility death spiral", as distributed energy resources become more and more prevalent and utilities get decreasing income from energy sales, but maintain the same costs from power transmission, necessary installed capacity, etc. In that context, the peak-shaving strategy may allow for the reduction of the user's contracted power, for a larger impact on the energy bill than the SCM strategy.[45]

Apart from rule-based strategies, other commonly used energy management strategies (which may sometimes overlap with one another) include:[42]

- Mathematical optimisation — deterministic optimisation-based methodologies, which include Mixed Integer Linear Programming (MILP), used in this work for benchmarking. These methods are able to find the optimal solution, as long as they are fed perfect predictions (i.e, deterministic data), and therein lies their problem: most of these methods have no built-in method for dealing with uncertainty, meaning high uncertainty (such as is true for small residential settings, due to high variability) can lead to poor performance. Methods such as stochastic optimisation can account for modelled uncertainty, computing the objective function for all possible realisations and taking its expected value. However, this increases the computational burden, as well as requires the additional step of explicitly modelling for uncertainty.
- Heuristics and metaheuristics — lighter computational burden, but do not guarantee an optimal solution.
- Game Theory — typically for multi-agent systems, which is not the case here.
- Machine Learning, in particular, Reinforcement Learning.

Some criteria/objectives commonly used for scheduling include minimising costs (primarily energy consumption costs, but may also include device start-up costs, battery deterioration and the cost of a carbon tax), load profiling (e.g., decreasing peak load, reducing grid dependence), ensuring user well-being (e.g., thermal comfort) and reducing emissions.[42, 46]

Across a large number of references on the subject, [46] found that HEMS could reduce energy costs by up to 23% on average — this considering a very broad definition of HEMS, encompassing the control of ESS, HVAC systems, EVs, and other shiftable loads (it is worth noting that the larger the amount of controlled devices, the larger, in principle, the potential for cost reduction), and encompassing many different conditions (pricing, ESS capacity, renewable generation). It also found that HEMS models in the literature use most frequently a 1h time resolution, 24h planning horizon and 1h rescheduling interval.

Azuatalam *et al*. (2019) [44] performed a systematic review of seven different energy management strategies for small-scale PV battery systems. The authors argue that most studies on the matter focus on one specific aspect — be it economic viability, data uncertainty or quality of the strategy employed — which is why they aimed to fill this gap. They compared three rule-based strategies (SCM, ToUA and SCM + ToUA), two optimisation strategies and two machine learning strategies.

This study used persistence models for load and PV forecasting. Electricity tariffs used were based on the Australian context. Under these conditions, it was found that for imperfect forecasts of PV and demand, using a complex strategy may not present benefits when compared to a simple rule-based strategy. However, the authors compare the different strategies qualitatively, by means of a rank, and not quantitatively, and attributed a rank to each aspect (speed, economics, modelling complexity, accuracy, etc.) and not to each strategy as a whole. This hinders in part the drawing of meaningful comparisons between strategies.

Wang *et al.* (2015) [47] uses Mixed Integer Linear Programming (MILP) to model HEMS energy management of a smart home under real-time pricing (RTP) and day-ahead pricing (DAP). Specifically,

it tackles the scheduling problem of the air conditioning system, as well as an ESS coupled with a PV system.

Control timesteps span 15 minutes, and two case studies were considered: one with day-ahead pricing and day-ahead scheduling; another with real-time pricing, a 24h rolling horizon and recalculation every 15 minutes. The algorithm and the two case studies were simulated on a single smart house and a single day, a typical Sydney summer day. It was assumed that the day-ahead forecasts would have a "relatively high degree of accuracy".

Results show that while day-ahead planning seemed to result in less costs for the user, the comparison is not straightforward. This is because the day-ahead forecasts seemed to underestimate ambient temperature, therefore implicitly underestimating cooling needs, which means that the day-ahead scenario is not realistic. This means that the two scenarios are not directly comparable and, more than anything else, shows the importance of forecast accuracy for optimisation algorithms, or, alternatively, of taking into account forecast uncertainty, or risk unintended consequences in terms of either energy consumption or user comfort/convenience. The study also did not provide a baseline against which to compare both methods, showing their potential benefits.

Rasouli *et al.* (2019) [48] compares two different methodological approaches to be used on a HEMS: MILP and a Genetic Algorithm (GA). The goal was to perform scheduling of shiftable, interruptible and thermostatically-controlled loads, as well as control local generation and storage, under dynamic tariffs, minimising total energy costs while taking into account different users' requirements for user comfort, by simulating two different types of consumers.

The proposed methods were simulated on a single case study for a single winter day in Coimbra, Portugal. The discretisation is of 1 minute. No mention is made of forecasting or simulating forecast uncertainty. Results show that MILP provided solutions close to optimality, but at a relatively high computational cost, while GA provided only slightly worse solutions at a much smaller computational cost: for consumer type 1 (willing to sacrifice comfort for lower cost), MILP proved 8% better than GA, while for consumer type 2 (willing to sacrifice cost for better comfort), MILP was 10% better than GA.

Hou *et al.* (2019) [49] analysed different HEMS methodologies and strategies. With a baseline of PV generation with load scheduling, each further addition of first an ESS, then an EV (with vehicle-to-home), then a specialised strategy focused on managing the two available energy storage options, was able to generate relevant cost reductions. It also compared 4 different MILP-based methods, of which the best was the one using CPLEX (a solver for optimisation algorithms), both in terms of cost reductions and of calculation time. The planing horizon considered was 24h, and the granularity 15 minutes.

It is worth noting that this work's authors did not perform forecasting, and assumed the necessary correct forecasts are available. Furthermore, the simulation used wide-scale PV generation data (MW-scale) and simply scaled it down to a reasonable scale for a residence. This may generate bias, as larger systems have much smoother generation curves than smaller ones.

Wu *et al.* (2015) [50] tested and compared three different approaches for tackling HEMS: MILP, Continuous Relaxation (CR) and Fuzzy Logic Control (FLC). The simulated case study included PV, domestic hot water usage (with an electric water heater), an ESS, and other appliances. Data was collected from a home in Illinois, USA, pertaining to two days during October 2012. Day-ahead forecasts were generated for energy prices (with real-time pricing), hot water usage, outdoor temperature, PV and critical loads (non-shiftable), with large errors in order to simulate real operating conditions. Results showed

that all three strategies performed well, even with inaccurate forecasting information. MILP performed only slightly better than the others, but at a greater computational cost, especially when compared to FLC, which has the added advantage of not needing forecasts. Authors therefore highlight FLC as the most promising method.

Dinh *et al.* (2022) [51] Compared three strategies aiming at total energy cost reduction through a HEMS managing a renewable energy system and an ESS. These strategies were simulated using data from February 2014, from three homes in London, UK, classified based on residents' behaviour into stable, fluctuating and chaotic. The three strategies were:

- A MILP-based supervised learning strategy, where MILP provides the optimal solutions to historical data, which are then used to train a supervised learning model which will approximate the MILP solver. This strategy requires forecasts, and for this purpose an RNN is used for hour-ahead load prediction.
- A Reinforcement Learning strategy, using the Deep Deterministic Policy Gradient (DDPG) algorithm, using only 5 state variables: real-time energy consumption, real-time irradiation, current state-of-charge, real-time price, and time t (therefore not including forecasts).
- Forecast-based MILP, where an RNN generates forecast values at the start of the day, which are then fed into a MILP algorithm, outputting the sequence of control variables for the ESS and renewable energy system for the entire day.

The first two strategies are implemented on an hourly basis, whereas the third is implemented on a daily basis. Results showed that the MILP-based supervised learning strategy performed the best on all three scenarios, providing the highest cost reduction, followed closely by the DDPG strategy. Forecast-based MILP performed the worst out of the three, which was expected due to its coarser time resolution.

Lissa *et al.* (2021) [52] developed a DRL algorithm, specifically Q-learning, for indoor and domestic hot water temperature control, in the presence of a PV system, aiming to reduce energy consumption. The model was simulated on a residential building in Ireland. Without compromising user comfort (i.e., ensuring temperatures remain within the setpoints), the algorithm achieved 8% energy savings when compared to a rule-based model, and up to 16% energy savings during the summer period, which showed the highest potential for improvement. Additionally, local renewable energy consumption was higher using the model, meaning less energy is consumed from the grid. Another relevant observation is that the Q-learning algorithm became stable after only one month of simulation.

Ji *et al.* (2019) [53] used a Deep Reinforcement Learning (DRL), in particular, a Deep Q-Network (DQN), in order to build an EMS for a microgrid. The problem was formulated as a MDP, whose reward was the negative of the rescaled operating cost of the microgrid at each timestep, and took as state variables the energy prices and net load of the previous 24h, as well as the current state-of-charge (SOC) of the ESS. The MDP model then attempts to find an optimal scheduling policy to maximise total expected rewards. The proposed model performed better than the uncontrolled approach, and somewhat better than other Q-learning based approaches.

Appino *et al.* (2017) [54] used probabilistic forecasts to compute dispatch schedules for what they named a *dispatchable feeder*, consisting of a residential load, small PV generation and an ESS. The model is able to perform online adjustments to the dispatch schedule using MPC. It is also able to

implement a security level, that is, a given probability of avoiding imbalances. For the forecasts, quantile regression was used, which enables the implicit modelling of the correlation of forecast errors at adjacent timesteps. This model was tested on a single Australian household with rooftop PV generation and a battery. The results showed that a) the security level was indeed met and b) the proposed model could present economic benefits when compared to the most common alternative approaches for the forecasting step, deterministic forecasts and worst-case scenario approaches. Additionally, the authors denoted cost of security as the difference of cost between the deterministic and probabilistic case, and found that setting higher security levels (thus reducing the number of imbalances) leads to an increased cost of security.

2.5.1 Forecast uncertainty

Beaudin *et al.* (2015) [46] found that most works on HEMS focus solely on scheduling, and not on forecasting. For this reason, these works either a) assume the forecasts are accurate, or b) insert errors into the data in order to simulate forecast errors. Alternatively, many works do not use forecasts at all, instead making decisions based only on past and present data, which removes the need to account for uncertainty. Another alternative is to consider the worst case scenario (e.g. highest possible load, lowest possible renewable generation), such as in robust optimisation.[42] A variety of other approaches also exist.

For dynamic pricing situations, prices can only be estimated, leading to some uncertainty as well.

Unlike conventional model-based approaches to HEMS, Deep Reinforcement Learning-based approaches do not require an explicit model of the uncertainty.[53] Instead, they may implicitly learn to detect higher uncertainty-prone situations, and act accordingly, since the agent may be penalised during training for being overly optimistic.

Ji *et al.* (2019) [53] uses DRL for microgrid scheduling without explicit uncertainty modelling, and results show that the model is able to perform satisfactorily and implicitly foresee high-uncertainty situations, finding a solution that is close to optimal, achieving a 21% cost reduction when compared to the baseline strategy.

2.6 Previous work

This dissertation builds on work done on a previous dissertation by Emanuel Parracho.[3] Both dissertations use similar datasets, but not the exact same, since the present work includes more recent data, as well as not the exact same dwellings. The previous work focused only on forecasting for both PV generation and load, using a Random Forest (RF) algorithm.

Parracho (2021) [3] trained three different RF models: one for PV generation, one for total load, and one for net load. Each of these was compared with a persistence model for benchmarking. The results were as shown on table 2.6.1.

Table 2.6.1: Summary of results obtained on the previous work [3]

Model	Persistence nRMSE [%]		RF nRMSE [%]	
	Mean	Median	Mean	Median
PV generation	67.71	62.36	68.80	51.72
Total load	121.30	119.38	91.87	89.89
Net load	150.74	153.18	97.51	95.57

In repeating previous work and instead applying it to ANNs, the aim is to find the best algorithm for the forecasting of each of the variables, in order to improve the data fed into the HEMS, and thus improve its chances of success.

Compared to RF, ANNs are significantly less interpretable, due to the high number of connections within the network. However, their complexity also tends to make them able to yield good results when applied to a greater variety of tasks. ANNs are the most used machine learning techniques in solar power forecasting, being used in about 24% of PV forecasting papers, while RF is used in only about 5%.[19] However, Benali *et al*. (2019) [31] compared a RF and an ANN model for solar PV forecasting with a forecast horizon spanning from 1 to 6 hours and a different model for each season, and found that RF performed better for all horizons and all 4 seasons.

2.7 Final remarks

Across all reviewed articles, the author noticed that both forecasting and energy management studies tend to use relatively small sample sizes, namely, one month, a select few weeks out of the year, etc. The datasets used are also carefully selected so as to be robust, without missing or erroneous data, and outliers are sometimes removed. Additionally, usually only one sample, or one building, is used, or, alternatively, data from an entire power plant or grid is used.

In this dissertation, the opposite is true. Large amounts of data are available, both in terms of number of different dwellings, and long periods of time. For most dwellings, nearly two years of data are available. However, this data often presents many gaps and some inconsistencies. It is undeniable that this will certainly both hinder model performance and influence results. However, it can be more realistic in the sense that real-world systems often show faults, and may need to be trained on similarly imperfect data.

Data is available for 193 dwellings, of which 127 were used. This allows for a more in-depth analysis and removes the possibility of accidentally arriving at a solution which works only for a specific case: on the contrary, it perhaps allows for a more accurate representation of how the model is actually expected to perform after large-scale implementation.

With these in mind, it is expected that the results show that the model created here may underperform when compared to the literature, but on the other hand, that performance may be more realistic for practical applications.

Chapter 3

Methodology

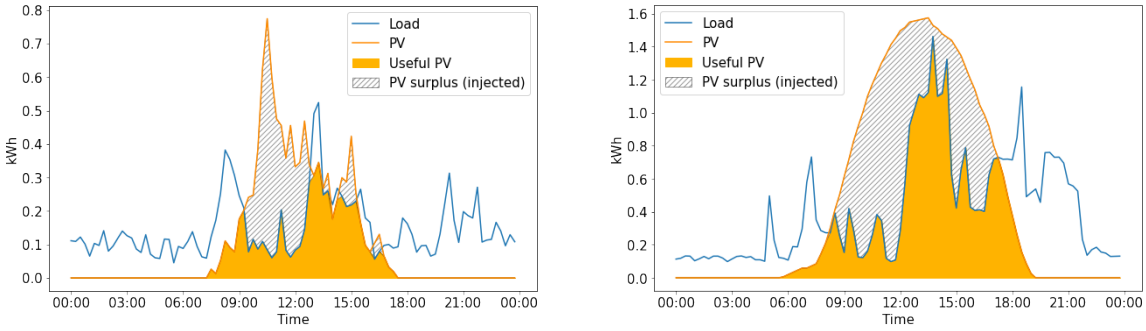
3.1 General data description & analysis

The data used in this dissertation for both forecasting and simulating the Home Energy Management System corresponds to PV generation, grid consumption and grid injection data, with 15-minute granularity, for 193 anonymous dwellings of unknown location within Portugal. The data collection period varies but is generally comprised between July 2020 and April 2022.

For both forecasting and HEMS simulation, PV generation data was used as is, whereas for consumption we are interested in total load, and not net load. This is obtained indirectly according to Eq. 3.1.1. It is important to note that the previous work has not defined load the same way, and thus the results are not directly comparable.

$$\text{total load} = \text{grid consumption} + \text{PV generation} - \text{grid injection} \tag{3.1.1}$$

Figure 3.1.1 shows two example days from one dwelling: one winter day and one summer day. Useful and surplus PV is illustrated assuming a system without energy storage. Winter days have typically highly variable PV generation, as illustrated, while summer days tend to show a smooth profile more often. Regarding load, note that a specific range of relatively low values occurs often, likely corresponding to the base load of the dwelling, such as the refrigerator and other constant loads, with occasional high values, which are often hard to predict. This illustrates the challenges of forecasting: for PV generation, smooth profiles are easy to predict, but those are the exception rather than the rule. Without resorting to data external to the system, such as NWP data, predicting the effect of cloud cover on PV generation is generally difficult, often impossible. For load, predicting the exact time at which a sudden load peak will occur is not simple, as it depends on human behaviour. Additionally, the fact that a small range of smaller values is more common than larger values may lead some ML models to learn to predict values in that range only, as it is "safer" than risking predicting a larger value and getting it wrong.



a Winter day (21-01-2021)

b Summer day (18-08-2021)

Figure 3.1.1: Example data from one dwelling

3.1.1 Treatment of missing and erroneous data

Missing data (denoted NAN) can be a problem for the tasks at hand, particularly for HEMS simulation, where having continuous trustworthy data is important, but also in forecasting, seeing as missing data will affect sums, means and standard deviations that are used as input features for the models.

For forecasting, NAN data has a larger impact on load than PV generation, seeing as the load variable depends on three variables and a load data point is rendered as NAN if any of the three variables are NAN, whereas PV generation is one variable only.

The percentage of NANs per month was analysed in order to detect possible bias sources for the subsequent work. As can be seen on Figure 3.1.2, monthly variation exists, but it was not considered significant. It is worth noting that May and June, the months with lowest NAN percentage, are also the months with fewer total data points, due to the data collection period (July 2020 to April 2022 for most dwellings).

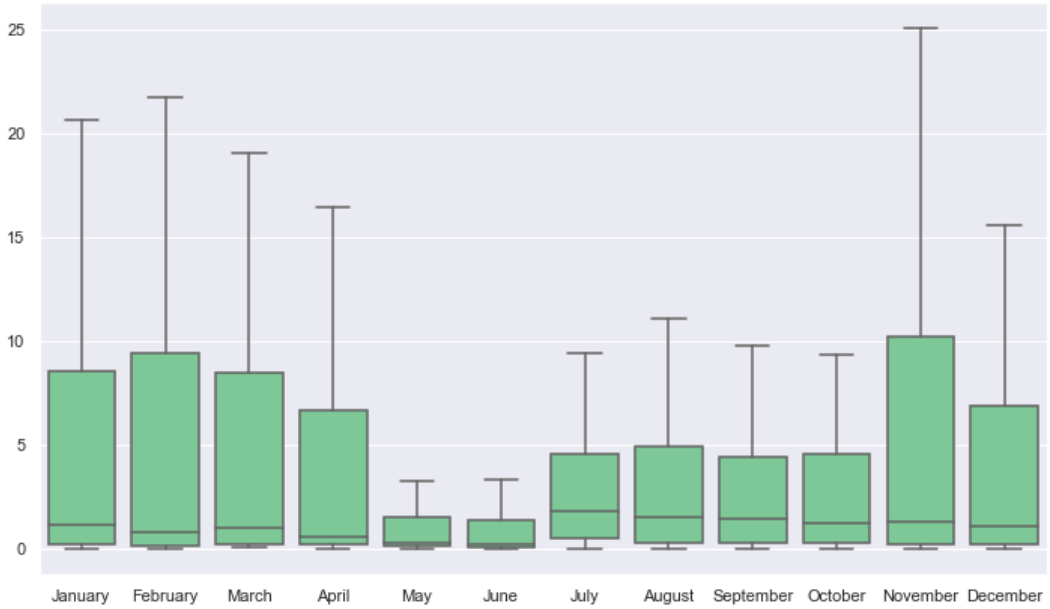


Figure 3.1.2: Percentage of NAN data per month (any variable).

Data quality varied significantly from dwelling to dwelling. To try to minimise the impact of poor data quality, some of the dwellings were excluded from consideration. The conditions for exclusion were:

- Over 10% of NAN values in any column (total 56 dwellings excluded)
- Over 15 continuous days of zero PV generation (total 10 dwellings excluded, 5 of which had no PV generation data whatsoever)

In total, 66 dwellings were excluded, leaving a remaining 127 dwellings which were used to develop and simulate the algorithms in this work.

Most of the remaining dwellings still had small amounts of missing data, which needed handling, since all time series must be continuous in order to allow for uninterrupted episodes in the HEMS stage of this work (and by extension, for the cumulative forecasts). For this reason, NAN data points needed to be replaced with real numeric values. Interpolation would not be appropriate — in operating mode, data faults are expected to occur as well, and interpolation assumes knowledge of the future, which would not be possible. Interpolation also has a smoothing effect on the data, which may artificially make forecasting seem easier than it is in reality — creating artificially optimistic results. Instead, a "pessimistic" approach was taken here — one that can handle faulty data in real-time operation: NAN values in the PV time series were replaced with zeros, while NAN values in load were filled by propagating the last valid value.

In addition to missing data, some time series may include erroneous data originating from measuring problems. Part of this is impossible to detect consistently; however, the data points with potential to cause larger complications (particularly in the HEMS stage of this work) are those with abnormally large values (e.g., a value 100 times larger than the mean of the time series). There was an attempt to remove these values (avoiding case-by-base analysis) by removing all values larger than twice the 99th quantile of the time series.

3.1.2 Training, validation and test datasets

In order to train and evaluate the models, available data must be split into training and testing datasets. Training data is sometimes further split into training and validation. In that case, training data is used to directly train the model, while validation data is auxiliary to training: the training algorithms regularly check the model's performance on validation data during training, in order to ensure the model is generalising and learning the relevant relationships between data points, as opposed to simply overfitting on the training data. Finally, after the model is trained, its performance is evaluated on test data.

It is vitally important to ensure a clean separation between the train/validation set and the test set, in particular. Since the test set is used to assess model performance, the data within it should be data never before seen by the model; otherwise, the model will have information it should not have access to. Kapoor *et al.* (2022) [55] looked into a phenomenon it dubbed the "reproducibility crisis in machine learning-based science", by analysing several papers in diverse fields, finding that many papers which claim drastically good results have, in fact, committed one of several types of lapses, leading the results to be overly optimistic. Most of the mistakes listed involve the lack of a clean separation between

the training and test sets. Some papers lack a test set whatsoever, simply testing on the training data, which will evidently lead to overfitting and consequently overly optimistic results; others employ pre-processing based on both datasets (such as feature selection, meaning features are chosen based on information about the test set which the model should not have access to). Others yet exhibit what the authors call *temporal leakage*: when training a model to make predictions about the future, the test set should contain only data points occurring entirely *after* all data points on the training set: the training and testing sets should never overlap.

Regrettably, Parracho (2021) [3] is at fault for the latter: its train/test split uses every fourth week as test data and the remainder as training data, which means the model is tested on data which occurs *before* some of the data which it is trained on, constituting temporal leakage, as the model has information about the future *a priori* — e.g, a user may purchase a new appliance during the analysis period and temporal leakage means the model will have information on this beforehand, if trained on future data. Or, if the model trains on two non-consecutive weeks separated by one week that is then used for testing, it will more easily predict PV generation for that test week, since it has information about the future.

Instead, in this work, a simpler approach is taken: the full range of available data for each dwelling is split into two, with the first half being used for training and validation, and the second for testing. The main consequence of this is obvious: unlike in the previous work, the model now has no information about the future, which is more realistic. This also means that performance is expected to be poorer than before.

Unfortunately, in avoiding temporal leakage, there is another issue mentioned in [55] which cannot be completely avoided with the available data: sampling bias. This is when the test set does not cover the full extent of the domain in which the model must perform (e.g., spatial bias, where a model is evaluated on a restricted geographic location but claims to be valid for a wider area). In this work, we may call what is present *temporal sampling bias*. This bias exists only because, for most of the dwellings, over one year but under two years of data is available. This means that the latter half of the full dataset, used for testing, usually does not have data pertaining to the four seasons simultaneously. For this reason, performance on the test set may not represent average model performance throughout the year. Ideally, this issue would be fixed by having at least two full years of data available for each dwelling, therefore having proportional representation of each season in both the training and test sets.

Aside from temporal sampling bias, this train/test split may also impair model performance, since, while training, the model may not witness the full range of yearly variation, therefore having a more limited ability to make predictions on seasons it may not have observed during training.

Ultimately, the decision to tolerate temporal sampling bias as opposed to temporal leakage rests on the fact that this is more aligned with realistic conditions for real-life model operation, as well as the fact that, since not all dwellings have datasets corresponding to the same period, this effect may be somewhat lessened by looking at the results across all dwellings at once. Nonetheless, temporal sampling bias should always be kept in mind when analysing the results.

For this work, it should be noted that the data split is the same across all models. All models, including persistence, are tested on the latter half of the available data.

Random forest models use only one set of training data, and since no hyperparameter tuning is performed here for RF, a validation set is not necessary for these models, meaning they use the full first half of the complete dataset for training. ANNs, on the other hand, require validation data for a) early

stopping (stopping training early if performance on validation data does not improve, in order to avoid overfitting on the training data) and b) hyperparameter tuning. The split was therefore moving one in four samples (chronologically) from the training set to the validation set **only for ANN models**. This has one possible downside: ANN models will have 25% fewer samples to train with than RF models.

3.2 Forecasting

The goal of the forecasting study in this dissertation is to find the best performing model for specific conditions in order to integrate it with a HEMS. These conditions are: small residential PV systems (higher variability and unpredictability than larger systems), no access to NWP data, and anonymity (no access to location/occupant/etc. data). Input data available for forecasting is then historical generation/load data and general calendar/hourly information.

This section begins by describing how RF and ANN models were used to build point-forecasting models as a first approach to comparing the two models. Next, it describes how both methods were used once more to build cumulative forecasting models with the goal of summing up the most relevant information for the HEMS in few data points, then choosing the best out of the two to be used in the HEMS stage of this work.

3.2.0.1 Metrics

The main metrics used here to assess model performance were nRMSE (normalised root mean square error), nMBE (normalised mean bias error), R^2 (coefficient of determination) and, to a lesser extent, nMAE (normalised mean absolute error) (Eqs. 3.2.1-3.2.4, where y is the real value and $\hat{y}$ the predicted value).

$$\text{nRMSE} = \frac{\sqrt{\sum(y-\hat{y})^2}}{\sum y} \tag{3.2.1}$$

$$\text{nMBE} = \frac{\sum(y-\hat{y})}{\sum y} \tag{3.2.2}$$

$$R^2 = 1 - \frac{\sum(y-\hat{y})^2}{y-\bar{y}^2} \tag{3.2.3}$$

$$\text{nMAE} = \frac{\sum|y-\hat{y}|}{\sum y} \tag{3.2.4}$$

nMAE and nRMSE are often used in forecast evaluation. Both measure the average magnitude of the error without considering direction, meaning that only positive values or zero are possible. nMAE is linear, which means all instances are weighted equally. nRMSE, on the other hand, squares the prediction differences before taking the mean, which means this metric is more deeply affected by large prediction differences. This is in line with the goal of these forecasts: smaller errors are less important than larger ones.

nMBE takes the mean of the difference between real and forecasted values. The direction of the error is not disregarded in this metric. It therefore allows one to verify whether the forecast is biased, that is, whether it tends to make predictions consistently below or consistently above the real values. Ideally, nMBE would be zero.

R^2 measures how good the fit is. It measures the extent to which variability in the forecasted values correlates to variability in the real values. It is the metric that best directly describes how close to the red line each prediction falls in plots such as the one in Figure 3.2.1, which is an illustration, with synthetic data, of the plots used later in this work to visualise model performance. This metric is important to differentiate a model which has, for example, low nRMSE because it accurately predicts variability, from a model which has low nRMSE because the predicted variable itself has low variance. For a variable with low variance, a model which consistently predicts values very close to the mean will automatically have a low nRMSE, but not necessarily a high R^2. R^2 typically ranges from 0 to 1 (0 meaning no correlation and 1 meaning perfect correlation) but can sometimes take negative values, if the model performs worse than simply predicting the mean, or if there exists an inverse correlation.

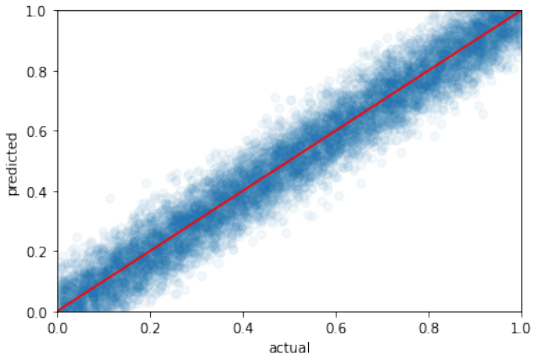


Figure 3.2.1: Dummy example of the scatter plots used later in this work to visualise model performance.

All metrics are normalised in order to allow for the drawing of meaningful conclusions across all dwellings, as well as a more accurate sense of the magnitude of the error without the need for additional information, such as the time series mean.

3.2.1 Persistence models

As in Parracho (2021) [3], persistence models were used as benchmarks for the first stage of forecast model testing. This choice derives from these models' simplicity and how frequently they are used as benchmarks in the literature. Forecast skill (FS) was then calculated based on the benchmark, as the improvement in nRMSE of the model when compared to persistence, as in Eq. 3.2.5.

$$FS = \frac{\text{nRMSE}_{persistence} - \text{nRMSE}_{model}}{\text{nRMSE}_{persistence}} \tag{3.2.5}$$

Clear sky persistence is normally applied in a PV forecasting context. However, for a 24-hour horizon, clear sky persistence and naive persistence are equivalent, with naive persistence being simpler. For this reason, naive persistence was applied here to the 24-hour point-forecasting, for both PV generation and load, meaning the persistence forecast always predicts that a given PV generation or load value will

be equal to the value on the same time of day, the day before.

3.2.2 Machine learning models

Parracho (2021) [3] built Random Forest models for load and PV generation forecasting, which predict, based on historical data only, the load and PV generation values on each timestep, 24h in advance.

The first step in this work was to adapt and tweak those models, as well as build equivalent models using ANNs, in order to compare the two methods' performance and make a first assessment of model quality. Therefore, to allow for a fair comparison, both models used the same input features — those shown on table 3.2.1.

Table 3.2.1: Features used on each forecast, for point-forecasting [1]

Load	Day of year
	Time of day
	Load at time of forecasting
	Load one week before forecasted time
	Whether it is a weekday or weekend
	Whether it is a holiday
	Sum of total load on the 24h before forecasting
	Mean of load on the 24h before forecasting
	Standard deviation of load on the 24h before forecasting
	Sum of total load on the week before forecasting
	Mean of load on the week before forecasting
	Standard deviation of load on the week before forecasting
PV	Solar elevation
	Solar azimuth
	Day of year
	PV generation at time of forecasting
	Sum of total PV generation on the 24h before forecasting
	Mean of PV generation on the 24h before forecasting
	Mean of K_{cs} on the 24h before forecasting
	Standard deviation of K_{cs} on the 24h before forecasting

Two of the input features for the PV model include the clear sky index, K_{cs}. This is not calculated according to its original definition, the ratio between ground-level irradiation and irradiation above the atmosphere, as these quantities are not known (irradiation data is not available). Instead, an adapted version of K_{cs} was defined here as in Eq. 3.2.6, where $PV(t)$ is the generation on a given time of day t on a particular day, and $\max(PV_{15d}(t))$ is the maximum generation on time of day t found in the 15 days prior to time of forecast. This essentially removes the variation due to sun movement from the variable (as long as one considers that, in a period of two weeks, the daily movement profile of the sun does not vary significantly), which in effect means that the variable still represents cloud cover.

$$K_{cs} = \frac{PV(t)}{\max(PV_{15d}(t))} \tag{3.2.6}$$

[1] Day of year and time of day variables were made cyclical for ease of handling by the models (in order to prevent, e.g., December 31st from being interpreted as the polar opposite of January 1st). This was done by creating two variables for each variable to be made cyclical: $\sin\left(\frac{2\pi\, x}{\max(x)}\right)$, $\cos\left(\frac{2\pi\, x}{\max(x)}\right)$.

Feature importance analysis was not done in this dissertation due to the fact that it is not trivial to obtain feature importance in a meaningful way from ANNs. However, this analysis was done by Parracho (2021) [3]. In that work,

- for PV forecasting, the two largely most important features were the mean K_{cs} the day before, and the K_{cs} 24h before the forecasted timeslot. Other somewhat important features were the standard deviation of K_{cs} on the day before, the sum of total PV generation the day before, the day of the year, and the standard deviation of PV generation the day before. Other considered features had a low importance. The features deemed most important on the previous work were also used in the present work.
- for load forecasting, the two broadly most important features were the load 24h before the forecasted timeslot and the time of day. Other variables, namely sun movement variables, were found to have high importance in the previous work's model but were not considered for load in this work, due to how the load variable was defined differently — it was considered that load should be, for the most part, independent from sun movement. The obvious exception is possible HVAC loads, but for these cases the model should be able to infer suitable correlations from time of day and day of year variables. In addition, the previous work considered variables such as whether mandatory quarantine was enforced, considering how some of the data collection took place during the start of the COVID-19 pandemic, but these features were found to have a low importance and are therefore not used here.

3.2.2.1 RF model

Due to both a) load being defined differently by Parracho (2021) [3] and b) available data being somewhat different for this work than it was for the previous one, the decision was taken to re-run the RF forecasts on the new data. Hyperparameter optimisation was not executed here however, and the hyperparameters used were those that showed the best result in [3] (see table 3.2.2). In addition, the train/test split was changed in this work for the reasons mentioned on section 3.1.2, as well as to keep a consistent train/test split across the two phases of this work (forecasting and HEMS).

The Sci-kit Learn Python library [56] was used for RF models, specifically its RandomForestRegressor class.

Unlike ANNs, RF does not require feature normalisation, since it is a tree-based algorithm, making its preprocessing stage simpler and quicker.

Table 3.2.2: Hyperparameters used on RF model (PV and load)

Number of estimators (number of individual decision trees which make up the Random Forest)	50
Minimum samples per leaf	80
Maximum depth (maximum number of splits)	20

3.2.2.2 ANN

Unlike the RF model which was adapted from an existing model, two ANN models were developed from scratch for the task of PV and load forecasting, using the same features as those used by the RF model. The framework used to develop this model was Tensorflow 2.0 [57], an open source library for ML model development. Despite the fact that Sci-kit Learn also supports ANNs, Tensorflow was chosen due to its modularity and ease at composing a model out of select individual components, as well as compatibility with Keras[58], a deep learning API, and Tensorboard, which provides built-in visualisation of model training, metrics, hyperparameters, etc. — being very useful for hyperparameter tuning.

For network stability, and so that all input variables are on equal footing, inputs are normalised (using Sci-kit Learn's StandardScaler). Although the activation function may change according to hyperparameter tuning, the output layer's activation function here is always ReLU. Typically, all ANN models' output layers use one of two activation functions: linear (y=x) or ReLU. The advantage of ReLU here is that it leaves positive values unchanged, while transforming any negative values that may occur into zeros, thus preventing negative outputs. This is evidently useful here, as there are no negative values on load or PV generation.

An *early stopping* callback was used to prevent overfitting: this tests the model on validation data at the end of every epoch, and stops training if the validation loss stops improving for over a certain number of epochs.

The selected loss function was Huber loss (Eq. 3.2.7). Huber loss essentially combines mean squared error (MSE) and mean absolute error (MAE), producing a more robust loss function by seeking to combine each one's strengths and abate each one's weaknesses. MSE has a tendency of being dominated by outliers, in this case instances of very large errors, due to squaring the values. Conversely, MAE's linearity is a problem for small values, since its gradient will be the same no matter what, as long as the error's magnitude is larger than zero. This means that a very small error will generate the same gradient as a large one, which can lead to convergence problems. Huber loss essentially uses MSE for small errors, and MAE for large errors, thus aiming to avoid both these weaknesses.

$$Huber = \begin{cases} \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}(y-\hat{y})^2, & |y-\hat{y}| \leq \delta \\ \frac{1}{n}\sum_{i=1}^{n} \delta(|y-\hat{y}| - \frac{1}{2}\delta), & |y-\hat{y}| > \delta \end{cases} \tag{3.2.7}$$

Hyperparameter tuning

Hyperparameter tuning consists of an exploration of different hyperparameter combinations for a given learning algorithm, in order to select the optimal one.

Parracho (2021) [3] implemented hyperparameter tuning by performing a case-by-case analysis, using a random search algorithm which, for each dwelling, tries several different combinations of hyperparameters, in the end picking the combination which performs better for each case. Here, a more holistic and less computationally-heavy approach was taken: 1) choose 40 dwellings at random to perform hyperparameter analysis on, 2) use a random search algorithm to try different combinations of hyperparameters for each dwelling, and 3) analyse existing trends in the metrics as a function of hy-

perparameters. This way, one can see which hyperparameters truly affect model performance and which ones are inconsequential, analyse trends and avoid producing artefacts (e.g. avoid producing a very badly performing model simply because the random search algorithm may not have tried any favourable combination of hyperparameters for one dwelling), and essentially produce a set of hyperparameters which work well under the majority of circumstances, and can therefore be applied to any case under the expectation that they will produce results sufficiently close to optimal, instead of performing hyperparameter tuning case-by-case; therefore reducing computational cost.

The hyperparameters analysed were, with respective ranges/options in brackets:

- Number of hidden layers (2, 3 or 4);
- Learning rate (1×10^{-6} to 1×10^{-2});
- Hidden layer size, or number of neurons per hidden layer (32 to 512);
- Activation function (ReLU or tanh);
- Whether to use dropout[2] (with a dropout rate of 20%, meaning one fifth of neurons are "turned off" at any time during training).

For the PV forecasting models, after performing hyperparameter analysis, the findings, and corresponding conclusions, were: (see figure 3.2.2 for an example of the visualisation of this analysis)

- Very high learning rates (in the range of 1×10^{-3} to 1×10^{-2}) generated many instances of a 100% RMSE and -100% NMBE, suggesting high learning rates tend to generate models which always output zeros. Below this range, there seems to be a trend of increasing R^2 for larger learning rates, and this trend stagnates in the range of 1×10^{-4} to 1×10^{-3}. NMBE is smaller above 1×10^{-4} for training data but larger for test data, suggesting signs of possible overfitting. A learning rate of 1×10^{-4} will be used, as this seems to strike a perfect balance.
- Using 3 and 4 hidden layers was roughly equivalent and better than 2 regarding NMBE and R^2, with 3 generally yielding a lower RMSE, so 3 hidden layers will be used.
- 200 to 300 neurons per layer seems to be the optimal range for all metrics, despite the fact that the differences are not very prominent. 200 neurons per layer will be used.
- Regarding activation function, ReLU yielded smaller RMSE and higher R^2 than *tanh*, with no noticeable effect on NMBE. ReLU will be used.
- Using dropout improved all three metrics versus not using dropout, although the differences were not very marked. Nonetheless, dropout will be used.

[2]Dropout "deactivates" a set percentage of neurons on each layer during training, by multiplying their value by zero. The specific deactivated neurons change with each iteration. This, in theory, means that the ANN will not grow excessively dependent on one single feature or set of features, making it more likely to generalise and to still be able to yield a decent prediction on an instance with erroneous data on a particular feature.[16]

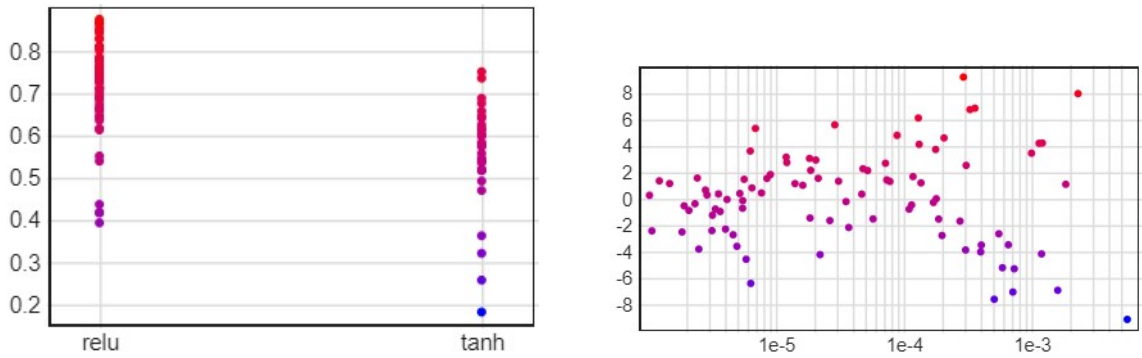
For the load forecasting models, the analysis is as follows:

- The same pattern of high learning rates (>1×10^{-3}) generating poor results that was found for PV was also found for load. Outside this range, no trend was found, therefore 1×10^{-4} will be used.
- Using 4 hidden layers proved better than 2 or 3 when measured via RMSE or NMBE, with no significant difference registered in R^2. 4 hidden layers will be used.
- Hidden layer size had no appreciable effect on model performance, so the previous value of 200 neurons will be kept.
- Activation function showed an impact only on R^2, where ReLU yielded better results, so ReLU will be used.
- Using dropout showed slight but observable improvements in all three metrics, which is why dropout will be used.

Table 3.2.3 is a summary of the hyperparameter combinations chosen for each model. It should be highlighted that some of the correlations which were found in this analysis were not very strong. Notwithstanding, while the chosen hyperparameters do not constitute a perfect recipe, they do constitute a set of hyperparameters that is not expected to negatively affect the model.

Table 3.2.3: Hyperparameters used on ANN models for point-forecasting

	PV	Load
Learning rate	1×10^{-4}	1×10^{-4}
Number of hidden layers	3	4
Hidden layer size	200	200
Activation function	ReLU	ReLU
Dropout	Yes	Yes



a R^2 vs. activation

b NMBE vs. learning rate

Figure 3.2.2: Examples of hyperparameter analysis for PV point-forecasting. Each point is one trained model (metrics are calculated for validation data). Outliers have been removed for ease of viewing.

3.2.3 Clear sky index (K_{cs}) forecast

One more test was done on point-forecasts: predicting K_{cs} as opposed to PV production. The goal was to inspect whether the models are able to predict anything beyond daily sun movement: PV generation is highly dependant on sun position (thus varying predictably according to the time of day and day of the year), and this variability is easy to learn and predict; in fact, if that were all the ML model is able to do, it could easily be done instead by a physical model which would predict it mathematically and deterministically, without requiring large amounts of data for training.

Since K_{cs} does not account for sun position, only for cloud cover, a decent performance on this forecast would mean that these models have some ability to predict a highly stochastic variable that is cloud cover. Instead, poor performance on this forecast would suggest that the proposed ML models are only able to predict sun movement, which in turn would mean that they may not pose an advantage, and a physical model may be more adequate.

Since this was meant as a simple and quick experiment, no hyperparameter tuning was performed here, instead using the same hyperparameters as the PV production forecasting models, as well as the same features.

R^2, in particular, will be analysed for this forecast, since that is a metric of correlation between real and predicted values. Any value reasonably above zero would show that the model is able to predict K_{cs} to some degree — simply predicting the mean value on all instances might yield a decent nRMSE but would yield an R^2 of 0.

3.2.4 Cumulative forecasts

For reasons that will be described on section 3.3.3, it was determined that the forecasts used to feed the HEMS model should be cumulative forecasts. These forecast a cumulative quantity Q, defined as the total sum of a quantity q (load or PV generation) between present time t and a future time $t+n$, n timesteps into the future (Eq. 3.2.8).

$$Q = \sum_{t=t}^{t+n} q \tag{3.2.8}$$

Both RF and ANN models were compared once again, for both load and PV generation. All five forecast horizons (1, 3, 6, 12 and 24 hours) are predicted by the same ANN or RF model, that is, these models can by nature be multi-output models, therefore not requiring one separate model for each variable. An additional horizon of 15 minutes was considered at first, but was abandoned since it showed poor results, making it unhelpful and unnecessary — applying persistence to this horizon would likely produce reasonable performance, but adds no new, useful information for the HEMS.

ANNs with multiple outputs calculate the average loss of all the outputs and try to minimise that average. For this reason, it is important to standardise outputs as well as inputs; otherwise, learning will prioritise the minimisation of the error on the output variable which has the largest values, and therefore the largest errors — in this case, the 24-hour horizon — while neglecting variables with smaller values. Sci-kit learn's StandardScaler is unhelpful here, since it creates a zero-mean distribution, generating negative results. This makes it so that some of the network's outputs remain negative, even after applying

the inverse transformation (e.g., if the test set registers a value smaller than all values on the training set). Sci-kit learn's MinMaxScaler would seemingly produce better results, since it scales inputs into a [0,1] range; however, using this scaler produced very poor results (worse than not applying any scaling whatsoever). This might be because it takes the absolute maximum and minimum values in order to define the value range, and some of those, particularly the maximum values, might be outliers (measuring errors, etc.) which got past the filters designed to catch them. A different approach was tested, and ultimately adopted, since it showed better performance — simply dividing all values by each time series' mean (always using the mean of the training dataset).

Hyperparameter analysis for the ANNs was repeated, and will not be discussed in detail since it is similar to the previous sections, but the hyperparameters used are visible on table 3.2.5.

When compared to the previous models, two features were added on the PV models: K_{cs} and clear sky production (i.e., the largest production value of the past two weeks for a certain time of day). On the load models, features pertaining to load values on the previous week (sum, mean, standard deviation) were removed, leaving only those pertaining to the previous 24 hours. The full list is available on table 3.2.4.

Table 3.2.4: Features used on each forecast, for cumulative forecasting

Load	Day of year
	Time of day
	Load at time of forecasting
	Load one week before forecasted time
	Whether it is a weekday or weekend
	Whether it is a holiday
	Sum of total load on the 24h before forecasting
	Mean of load on the 24h before forecasting
	Standard deviation of load on the 24h before forecasting
PV	Solar elevation
	Solar azimuth
	Day of year
	PV generation at time of forecasting
	Clear sky production
	Clear sky index (K_{cs})
	Sum of total PV generation on the 24h before forecasting
	Mean of PV generation on the 24h before forecasting
	Mean of K_{cs} on the 24h before forecasting
	Standard deviation of K_{cs} on the 24h before forecasting

Table 3.2.5: Hyperparameters used on ANN models for cumulative forecasting

	PV	Load
Learning rate	1×10^{-4}	1×10^{-4}
Number of hidden layers	2	2
Hidden layer size	200	200
Activation function	tanh	relu
Dropout	Yes	Yes

3.3 Home Energy Management System

Many works on HEMS use forecasts as model inputs but focus solely on the HEMS model itself as a scheduling problem, and either a) assume forecasts are accurate or b) inject synthetic errors into the forecasts in order to simulate real forecast error[46]. By taking a holistic approach and including both steps, this work aims at providing a more realistic simulation of what performance would look like in real-world operation, as well as comparing several different scenarios in order to assess the true impact of forecast use and forecast accuracy.

3.3.1 Case study definition and system specifications

Many HEMSs aim to schedule and control several household appliances and/or use devices such as sensors in order to detect motion or room occupancy.[42] That is not the case for this work — here, a simpler, non-invasive approach is taken, where occupant behaviour is not interfered with, nor is there an attempt to change the residence's load profile. The target of the scheduling exercise is solely the energy storage system, and load forecasting is based only on past data.

Regarding system specifications, the battery charging and discharging efficiency were both set at 95% (90.25% round-trip efficiency). The battery capacity for each dwelling was defined as 1 kWh/kWp of installed PV capacity, while the maximum charging and discharging power was, in kW, half of its capacity in kWh. This was done in order to streamline the process and in some way attempt to standardise its results, considering the high number of diverse residences to analyse. It is worth noting however that, in reality, batteries with these exact specifications may not exist for some of the dwellings.

In addition, the PV time series available from raw data were scaled so that $\sum PV = \sum load$ before being used for HEMS simulation purposes (for all cases: MILP, RL and SCM). This is because the existing systems which generated this data are not coupled to batteries, and for this reason, and as a result of the relatively low FiT in Portugal, the PV capacity was chosen so as not to be oversized — therefore, in most cases, peak PV generation practically never exceeds load, which means there is no excess generation that can be stored in an ESS. The presence of an ESS would enable a larger installed PV capacity to be profitable, so this re-scaling (which would generally increase total PV generation) was deemed acceptable. The reason for scaling so that $\sum PV = \sum load$ as opposed to using a fixed scaling factor is, once again, so that the results and analysis become somewhat standardised.

In all cases, the optimisation goal was the same: to minimise cost. This was done using time-of-use (bi- and tri-hourly) tariffs, since, as will be shown in later sections, SCM is optimal for a fixed tariff, eliminating the need for a better system (additionally, fixed tariffs usually are not the most well-suited to home PV systems, with or without energy storage). For both cases, the feed-in tariff (FiT) was considered constant, at 0.05€/kWh, for simplification purposes. In reality, this value varies monthly, is known only by the end of each month, and is calculated as 90% of the mean closing price on the Iberian Energy Market for that month.[5]

It is important to note that the costs considered here account only for energy costs, excluding power costs and any other types of costs. Excluding power costs means that performing peak shaving is not explicitly incentivised, nor is its effect measured. Despite this, peak shaving may happen indirectly when avoiding consumption from the grid during peak hours, which have the highest energy prices.

Tariff regimes used here are described on table 3.3.1, and were taken from EDP Comercial's website.[59, 60] It is worth noting that the tri-hourly tariff has a winter and a summer variation. The winter variation was used at all times in this work for simplification purposes.

Table 3.3.1: Tariff regimes considered, prices from EDP Comercial[59, 60]

Prices [€/kWh]		
Fixed tariff	Bi-hourly tariff	Tri-hourly tariff
0.2282 (constant)	0.2022 (off-peak, 10 pm to 8 am) 0.2561 (peak, 8 am to 10 pm)	0.1936 (void, 10 pm to 8 am) 0.3279 (peak, 9 am to 10:30 am, 6 pm to 8:30 pm) 0.2232 (full, all other times)

3.3.2 Benchmarking

This work is aimed at elaborating and evaluating a RL model for HEMS. In order to enable that evaluation, two other commonly-used methods were used for benchmarking purposes: self-consumption maximisation (SCM) and a Mixed Integer Linear Programming (MILP) optimisation model.

3.3.2.1 SCM as a baseline

Self-consumption maximisation (SCM), a rule-based algorithm, was used here as a baseline against which to compare the improvements granted by the use of the RL algorithm. SCM is common amidst simple battery management systems, and its use is widespread, which is the reason it was chosen as a baseline. SCM behaves as described by Algorithm 1.

For a fixed tariff, SCM is by definition optimal, as there is never any added value in delaying energy consumption from the battery (seeing as it is always used for self-consumption, and always replaces buying from the grid at the same price). This is backed by the fact that MILP and SCM are shown to be equivalent for this tariff further ahead in this work (see Figure 4.2.1b).

Algorithm 1 Self-consumption maximisation

if PV > load **then**
  use PV generation to fully cover load
  if $SOC < SOC_{max}$ **then**
    use surplus PV to charge battery (injecting excess into the grid if SOC_{max} is reached)
  else if $SOC = SOC_{max}$ **then**
    inject excess PV into grid
  end if
else if PV $\leq$ load **then**
  use full PV generation to partly cover load
  if $SOC > 0$ **then**
    use battery to cover remaining load (purchasing from the grid if $SOC = 0$ is reached)
  else if $SOC = 0$ **then**
    purchase from grid to cover remaining load
  end if
end if

3.3.2.2 MILP as the best case scenario

Mixed-integer Programming, or Mixed-integer Linear Programming (MILP), is a class of optimisation models which seek to explicitly maximise or minimise a given quantity, while subject to some constraints. These are usually employed on scheduling tasks which require the efficient use of limited resources.[61]

This work adapted an open-source MILP model by Edward Barbour on GitHub[62] built for battery scheduling, which uses Python Optimization Modeling Objects (Pyomo)[63, 64]. The chosen solver was Gurobi[65].

This model analyses a full time series of load and PV generation data and finds an optimal solution considering the given constraints and objective function. For this reason, this model was used as a benchmark of the best possible solution, by running it with perfect forecasts, i.e., real data.[3]

The constraints provided were essentially the physical limits of the battery, while the objective function was simply to minimise total cost. For a mathematical formulation of the model please refer to the implementation on GitHub[62].

It is highly unlikely that any RL model will be able to achieve the optimal solution given by MILP. The goal of this comparison is simply to assess *how close* the RL model is able to get to the optimal solution, as well as establish the maximum possible savings that could be achieved, therefore establishing the *potential* for improvement — thereby assessing whether it is even useful to study alternatives to the baseline model SCM.

3.3.3 Markov Decision Process definition

An MDP is defined by (S, A, P, R, γ), where S is the set of possible states, or state space, A is the set of possible actions, or action space, P is the transition function, which gives the probability of the system transitioning from one state to another state depending on the action taken, R is the reward function, and γ is the discounting factor.

Summarily, on each timestep, the system is in a given state s_t, the agent takes an action a_t, leading the system to transition into another state s_{t+1} and receive a reward r_t accordingly.

For this problem, the *action* $a_t \in A$ taken by the agent on a given timestep t is one single continuous value representing the battery's charging/discharging power for the $\Delta t = 15$ minutes before time t. a_t is positive when charging, negative when discharging, and zero when the battery is not used.

The continuous action space is defined as:

$$A = [-P_{max}, P_{max}] \tag{3.3.1}$$

where P_{max} is the battery's max charging/discharging power, in kW.

[3] This model was also briefly considered as a possible option for real-time implementation, but having to correct for imperfect forecasts was found to be impractical, and small experiments seemed to show that running a frequently-updated rolling window version of the model would have a greater computational cost than RL.

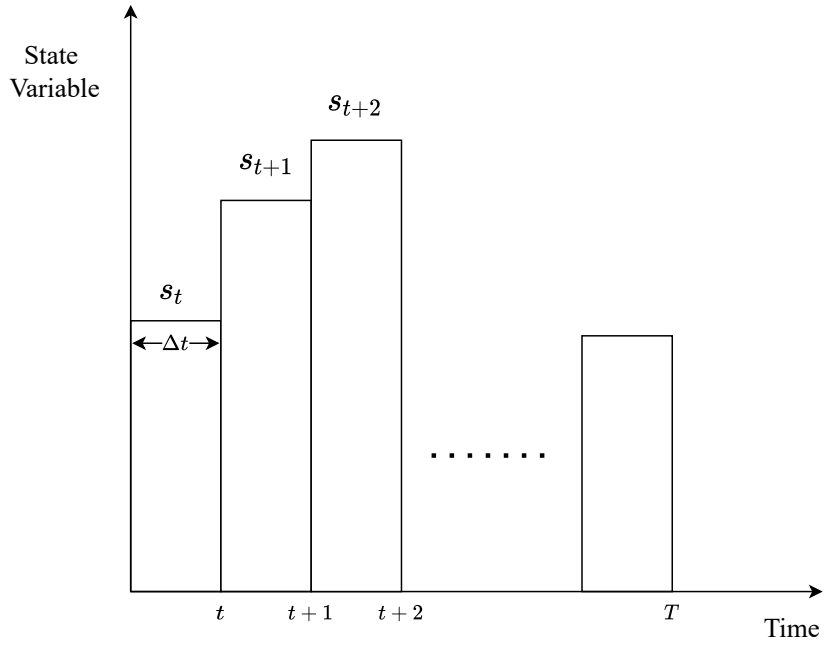


Figure 3.3.1: Illustration of the quantities considered in each timestep of the MDP, and which time period they refer to. Each point t, $t+1$, ... on the x axis corresponds to a *step* for the MDP, i.e., a moment when the agent chooses an action for the *following* timestep. Note how, at the time of choice, the agent has no information about the following timestep, where the action will take place. This is valid for variables PV_t, L_t, ΔE_t and GI_t.

The *state* $s_t \in S$ observed by the agent on each timestep is defined by quantities relative to the 15 minutes *previous* to moment t, as seen on Figure 3.3.1, as well as the cumulative forecasts for the following 1, 3, 6, 12 and 24 hours, for both PV and load. The exception is the state-of-charge, which corresponds to a point observation. s_t is then defined as the vector:

$$s_t = (SOC_t, PV_t, L_t, \Delta E_t, GI_t, EP_t, \sin(h), \cos(h), \hat{F}_t) \tag{3.3.2}$$

where:

- $SOC_t \in [0, SOC_{max}]$ is the battery state-of-charge at time t.
- PV_t is the observed photovoltaic generation, in kWh, within the 15 minutes before time t.
- L_t is the observed load, in kWh, within the 15 minutes before time t.
- ΔE_t is the energy, in kWh, added to, or extracted from, the battery. ΔE_t depends exclusively on the action a_t, and is positive when charging, and negative when discharging. It is computed as $\frac{a_t}{4}$, where 4 is the number of timesteps per hour.
- GI_t is the grid-injected energy (or negative net load) within the 15 minutes before time t, calculated as $GI_t = PV_t - L_t - \Delta E_t$. This value is positive when energy is injected into the grid, and negative when it is extracted from the grid.
- EP_t is the energy price at time t,

- $\sin(h)$ and $\cos(h)$ are the sine and cosine of the time of day. The sine and cosine are used so that the hour variable is made smooth and cyclical, avoiding a discontinuity in the variable values. This way, the model correctly interprets e.g. 23h59 and 00h as similar, and not as opposites.
- $\hat{F}_t$ is itself a vector containing cumulative load and PV generation forecasts for the following timesteps, corresponding to total PV or load predictions within the next 1, 3, 6, 12 and 24 hours.

From these, SOC_t and ΔE_t are endogenous variables, depending only on the actions taken by the agent, while all others are exogenous variables, external to the agent and depending directly or indirectly on the environment.

The advantage of the cumulative forecasts is twofold: a) it allows for relatively long-term planning without including excessive granularity and risking the *curse of dimensionality*[4], by providing the agent with long-term information in a more compact format, b) whereas normally with point forecasts uncertainty would significantly increase for longer horizons, cumulative forecasts by definition will have lower uncertainty, due to the smoothing effect (e.g., in the case of intermittent clouds, it is not relevant — as long as far enough into the future — the exact time at which a dip in output will occur, only its effect on total cumulative generation).

It is important to note that ΔE_t is the energy variation as seen by the HEMS and not by the battery itself. Δb_t, the variation of the battery's state-of-charge on timestep t, is a different quantity, as it is affected by the charging/discharging efficiency η, and as such is given by Eq. 3.3.3.

$$\Delta b_t = \begin{cases} \eta \Delta E_t & , \Delta E_t \geq 0 \text{ (charging)} \\ \frac{\Delta E_t}{\eta} & , \Delta E_t < 0 \text{ (discharging)} \end{cases} \tag{3.3.3}$$

State variables SOC and ΔE are updated according to Eqs. 3.3.4 and 3.3.5, respectively.

$$SOC_{t+1} = SOC_t + \Delta b_t \tag{3.3.4}$$

$$\Delta E_{t+1} = \frac{a_t}{4} \tag{3.3.5}$$

Table 3.3.2 holds a summary of all state variables for ease of viewing.

[4]The phenomenon known as curse of dimensionality states that the cardinality (number of unique points of a state space) grows exponentially with the dimensionality (number of state variables). However, the number of observations typically does not, meaning that the problem's state space becomes sparsely populated with observations, leading to possible difficulties in learning.[66]

Table 3.3.2: Summary of state variables on the HEMS environment.

Variable	Definition
SOC_t	State-of-charge
PV_t	Observed 15-minute PV generation (kWh)
L_t	Observed 15-minute load (kWh)
ΔE_t	15-minute state-of-charge variation (kWh)
GI_t	15-minute grid injection (kWh)
EP_t	Energy price (€)
$\sin(h), \cos(h)$	Time-of-day
$\hat{F}_t$	Forecasted values (cumulative) — length-10 vector

Not every action is possible in every state. Namely, one cannot charge a full battery, or discharge an empty one. In a classic optimisation problem (i.e., MILP), constraints regarding physical limits are explicitly modelled in the problem formulation. Unfortunately, RL lacks a straightforward way to do this (aside from action masking[67], which may be used to impose constraints on actions taken by the agent, but is only possible for discrete action spaces, and is therefore not applicable in this case).

For this reason, an *ad hoc* workaround was implemented for this purpose. When an illegal action is proposed by the agent, it is automatically substituted for the closest legal action (e.g., if the chosen action would take the state of charge beyond its maximum value, the action that would take it to its maximum value is chosen instead). After computing a theoretical temporary state-of-charge SOC_{temp} based on the proposed action only, the legal action taken is determined by Algorithm 2.

Algorithm 2 Determining a legal action

if $SOC_{temp} \geq SOC_{max}$ **then**
  $\Delta b \leftarrow SOC_{max} - SOC$
  $\Delta E \leftarrow \frac{\Delta b}{\eta}$
  $a_t \leftarrow 4\frac{\Delta b}{\eta}$
  $SOC \leftarrow SOC_{max}$
else if $SOC_{temp} \leq SOC_{min}$ **then**
  $\Delta b \leftarrow SOC_{min} - SOC$
  $\Delta E \leftarrow \eta \Delta b$
  $a_t \leftarrow 4\eta \Delta b$
  $SOC \leftarrow SOC_{max}$
else
  $SOC \leftarrow SOC_{temp}$
end if

A schematic of this specific Markov Decision Process can be seen on Figure 3.3.2.

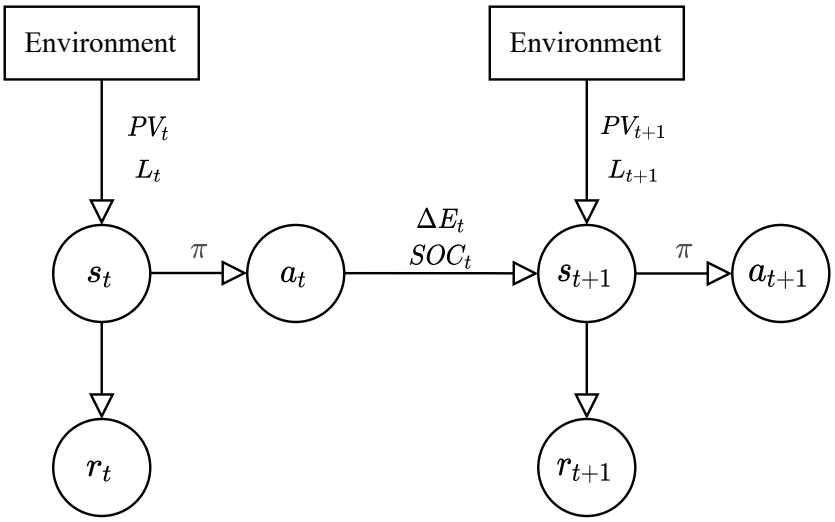


Figure 3.3.2: Schematic of the MDP of the proposed solution.

3.3.3.1 Reward design

The *reward* is a function of the system's state and the action taken by the agent, $r_t(s_t, a_t)$, and was built as depending primarily on the monetary value received/paid for the energy on each timestep, mo_t. This, in turn, depends on the energy injected into or extracted from the grid (GI_t), and the energy price EP_t.

Additional components were added in order to improve learning:

1. Penalisation for illegal actions (physically impossible actions) and forbidden actions (unquestionably non-optimal actions — see section 3.3.4);
2. Additional reward for charging the battery from the grid when grid energy is at its lowest price, *if the 24-hour load forecast is greater than the 24-hour PV generation forecast*. This was added because this behaviour is one that may lead to large monetary savings, and was not being observed as taken by the agent otherwise. Of course, the benefits of this depend largely on the quality of the 24-hour forecasts.

r_t is then given by Eq. 3.3.6, where BR_t (Eq. 3.3.7) is the *base reward* based on monetary value, applicable in all timesteps, and AR_t is the *added reward* (Eq. 3.3.8) for specific situations.

$$r_t = BR_t + AR_t \tag{3.3.6}$$

$$BR_t = 20 \times GI_t \times EP_t \tag{3.3.7}$$

$$AR_t = \begin{cases} -SOC_{max} \times 0.1, & \text{if illegal action is attempted} \\ -SOC_{max} \times 0.05, & \text{if undesirable action is attempted} \\ +\Delta E_t \times SOC_{max}, & \text{if battery is charged when price is lowest and } \hat{PV}_{24h} < \hat{L}_{24h} \\ 0, & \text{otherwise} \end{cases} \tag{3.3.8}$$

The method found to keep reward impact consistent across different dwellings was to scale the reward according to the battery capacity for each dwelling — i.e., penalisation equal to 10% of the value of battery capacity for each dwelling when the agent attempts to take an illegal action, 5% of battery capacity when a forbidden action is attempted (see section 3.3.4), and $\Delta E_t \times SOC_{max}$ under the conditions of item 2 on the above list.

3.3.4 Proposed RL scenarios

Seven different scenarios were considered and run for the RL models. These included different tariffs (bi- and tri-hourly only since, as was previously mentioned, SCM already represents the optimal scenario for a fixed tariff); forecast variations (*perfect forecast* — i.e., cumulative quantities directly calculated from real data —, ANN forecast and no forecast at all); and whether or not to enforce a number of *optimal actions* — i.e., whether to use a pure RL model or a hybrid RL/rule-based model.

While the RL agent *chooses* an action in advance for a period of 15 minutes, without knowing the real production and load values during that time, one could imagine a simple controller, external to the agent, which enforces a number of rules in real time, depending on energy balances at every moment. This is feasible, allows for a better performance, and does not require the system to be able to know the future with zero uncertainty, and so is worth exploring.

The optimal actions considered here are as follows:

1. **During maximum price periods, if there is a production deficit, use energy from the battery, if available, to cover the deficit.** This action is optimal since the best possible use for stored energy is to avoid consumption from the grid when the price is maximum.
2. **Always charge excess PV production to the battery if the battery is not full.** This action is optimal since the FiT here is considered constant — therefore, selling excess production now or selling it later on yields the same profit.
3. **Prevent charging from the grid, except when the energy price is minimum.** The true optimal action here would in fact be to prevent charging from the grid only when the price is maximum (this distinction only matters for a tri-hourly tariff), but indeed the selected version of this action performed better.
4. **Prevent the agent from selling battery-stored energy to the grid.** Selling battery-stored energy to the grid (considering a constant FiT) effectively means that energy, and therefore profit, is being lost due to η in the charging and discharging. Selling to the grid directly at time of production would be preferred. If a non-constant FiT is employed, the agent should be prevented only from discharging the battery into the grid when the FiT is at its lowest, since it could become beneficial to store energy in order to sell later at a higher price.

5. **Prevent charging from the grid followed by discharging, during a minimum price period.** As a result of optimal action 3, the battery may only be charged from the grid when the energy price is minimum. Initially, the agents showed a tendency to charge the battery using grid energy at the beginning of the minimum price period, only to discharge it *before that minimum price period was done*. This effectively led to wasted energy (due to η), with no upsides — directly using energy from the grid to cover consumption would be preferred. Therefore, during a minimum price period, if the agent charges using energy from the grid *even once*, it is prevented from discharging the battery at all for the remaining period until prices rise.

The agent also received a penalty (5% of battery capacity, as mentioned in 3.3.3.1) to its reward while attempting any of the forbidden actions (3-5).

It is a fact that these actions severely restrict the RL agent's freedom to experiment and explore, so it is important to note that the model developed here does not claim to be a pure RL model. It is, in fact, a RL/rule-based hybrid, and it is this hybrid quality which allows it to show improved results when compared to SCM, as will be demonstrated.

The reasoning behind the three different forecast variations is that comparing using the ANN forecast with no forecast allows one to evaluate the benefit added by the forecasts as they are, while comparing using the perfect forecast with ANN forecast allows one to gauge the potential benefit of investing in improved forecasts.

The seven scenarios considered were:

1. Perfect forecast + tri-hourly tariff + do not enforce optimal actions (pure RL model)
2. Perfect forecast + tri-hourly tariff + enforce optimal actions (hybrid RL/rule-based model)

From here on out, all scenarios are based on the hybrid model, i.e., include enforcing optimal actions. The reason for this is that, for the first two scenarios, this was shown to significantly improve performance for all dwellings, while having no downsides, as will be made clear in the results chapter.

3. No forecast + tri-hourly tariff
4. ANN forecast + tri-hourly tariff
5. Perfect forecast + bi-hourly tariff
6. No forecast + bi-hourly tariff
7. ANN forecast + bi-hourly tariff

3.3.4.1 State transition

Each state transition is governed by the structure shown in Algorithm 3.

Algorithm 3 State transition function

Get h_t, PV_t, L_t and a_t
if hybrid model **then**
Enforce optimal actions: replace a_t with optimal action if applicable (numbers 1 and 2 of the optimal actions list)
end if
Apply efficiency
Check if action is possible (enforce physical limits of the battery)
if hybrid model **then**
Check if action is allowed (numbers 3-5 of the optimal actions list)
end if
$SOC \leftarrow SOC_{temp}$
Calculate GI_t
Update state
Get $r(t)$

3.3.5 Choice of RL algorithm

Reinforcement learning can be model-based or model-free. While the first includes a predictive model of the environment and can perform *planning*, the latter bypasses planning, instead learning a policy, or state-value function (Q-function) directly by trial and error. Most currently available algorithms, particularly all available on Stable Baselines 3, are model-free.[68] Model-free methods are simpler and avoid issues such as model-based's compound errors[69], but have their own drawbacks: they are very *sample inefficient*, meaning that they require a large number of samples (i.e., observations), sometimes millions, to learn satisfactorily.[68]

Within model-free methods, there is a multitude of algorithms to choose from, as more and more algorithms are developed. The first restriction on choice of algorithm is whether the action space is discrete or continuous. In this work, the action space was defined as continuous, which rules out DQN (Deep Q-Network) based algorithms, built only for discrete actions[5].

Proximal Policy Optimisation (PPO) [71] was ultimately the RL algorithm of choice for this work.

PPO is a **policy gradient method**, which means it directly performs learning on the policy function π, as opposed to focusing on learning the Q-value function.[72] It is also an **actor-critic** method. This means it is composed of two separate neural networks: one learns the **policy** function $\pi(.|s)$ and generates an action (the **actor**), while the other learns the **value** function $V^{\pi}(s)$ and evaluates the result (the **critic**).

PPO is essentially an improved version of another policy gradient method, called Trust Region Policy Optimization (TRPO). TRPO was created with the goal of preventing policy updates that are too large and may result in performance collapse. Policy updates are such that they attempt to maximise the advantage, *while* being subject to a hard KL constraint[6] that prevents the policy update from being too

[5]DQN uses an ANN which takes the state as input and outputs a Q-value for each of the N discrete actions. Each output node corresponds to an action, which is why it is not suitable for a continuous action space. [66, 70]

[6]Kullback–Leibler divergence, or KL divergence, also called relative entropy, is a measure of how different a given prob-

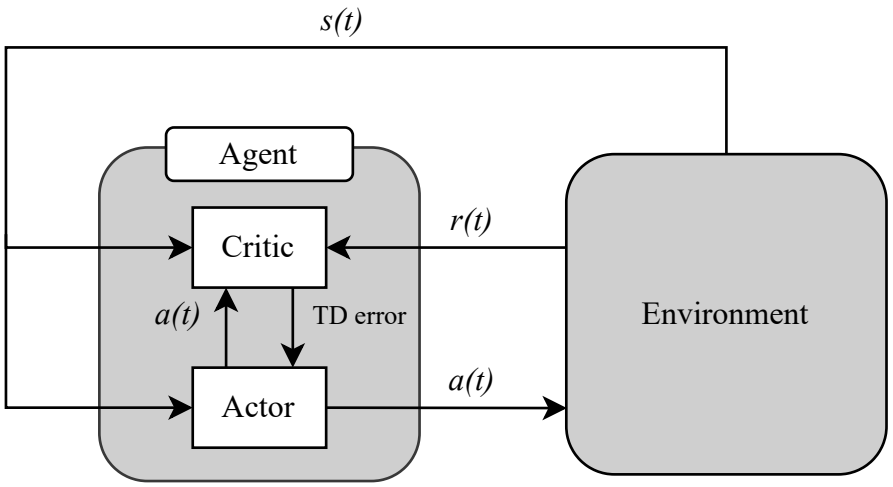


Figure 3.3.3: Simplified schematic of actor-critic methods. TD error stands for *temporal difference error*, the difference between estimated reward and the actual reward received. Adapted from Zhang *et al.* (2020) [14]

large.[72] However, this hard constraint makes it quite computationally heavy.

PPO is a simpler, more efficient version of TRPO. There are two distinct versions, both of which avoid using KL divergence as a hard constraint. PPO-Penalty calculates an approximation of KL divergence, but uses it as a penalty in the objective function instead of a hard constraint. PPO-Clip does not rely on KL divergence in either the objective function or as a hard constraint, instead clipping the objective function, thereby removing incentives for large policy updates.[74]

This simplifies the algorithm and increases efficiency, so that, according to its original paper, PPO was shown to outperform previous methods on almost all the continuous control environments it was tested on.[71]

PPO is more stable than other algorithms, as the policy update restriction helps prevent the sudden large drops in performance which are known to afflict other algorithms.[68]

PPO-Clip is the version available on Stable Baselines 3, and is the one which will be used in this work.

ability distribution is from a second probability distribution.[73] Applying a KL constraint in this context means that a given action cannot become much more likely or much less likely than before in a single policy update.

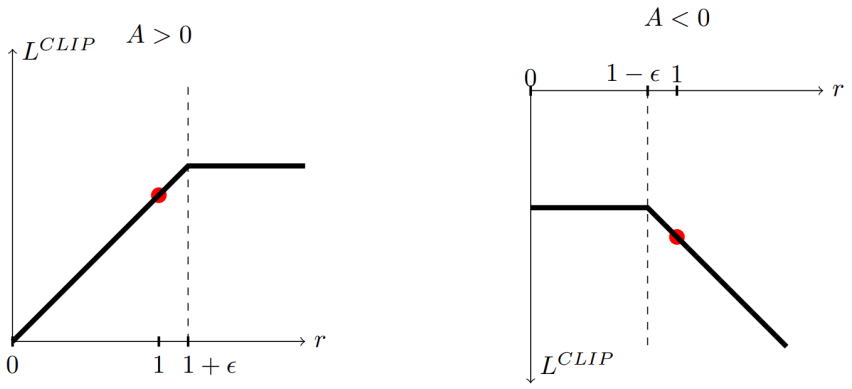


Figure 3.3.4: Clipped loss in PPO-Clip. r stands for probability ratio between the action under the current policy and the action under the previous policy, and L^{CLIP} is the clipped loss. For positive advantages (left), loss is clipped for large r values, discouraging the agent from making that action much more likely than it was. For negative advantages (right), loss is clipped for small r values, discouraging the agent from making that action much less likely than it was. Image source: Schulman *et al.* (2017) [71]

3.3.6 Model framework and architecture

The environment was modelled using OpenAI Gym [75], an open source Python library which provides a framework for building environments for RL agent testing, with built-in functions and architecture for agent-environment communication. Every aspect of the HEMS' setting is modelled inside this environment, namely, battery specifications, battery/PV/home/grid interactions, etc.

The RL agent is built using the Stable Baselines3 module [76], also open source, which provides a set of reliable implementations of RL algorithms. This module was chosen due to its simple, stable and fast implementations.

The interaction between these modules is illustrated in Figure 3.3.5.

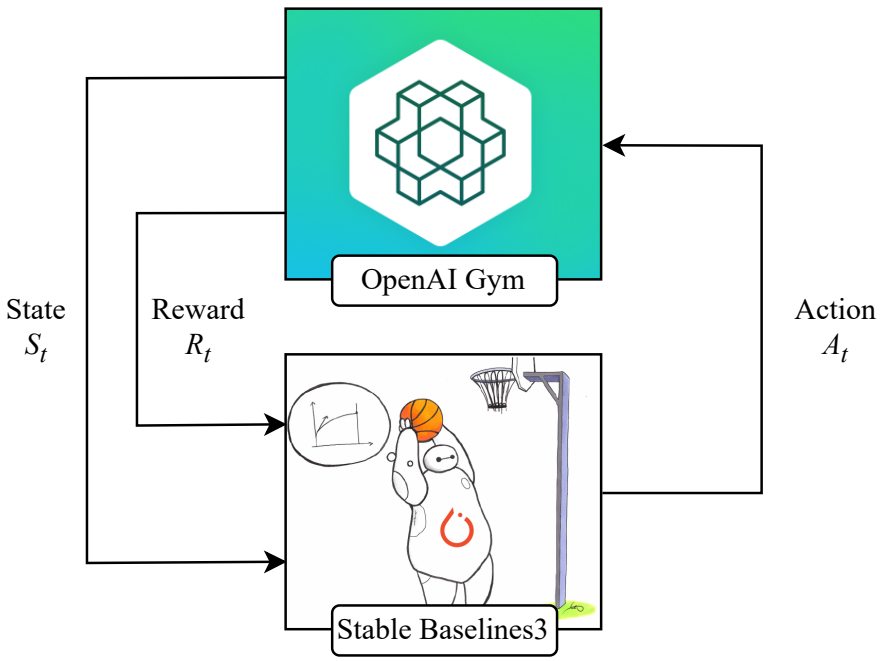


Figure 3.3.5: Interaction between modules.

Model-free methods' sample inefficiency means that obtaining a better performance generally means training for more timesteps and with more data. In this specific case, with limited data points, greatly increasing the number of timesteps would mean reusing the same data points several times. This, in turn, would bring a risk of overfitting on the training data, which is why it was generally avoided. In fact, only one passage was done through the training data, and the justification for this will be presented later, in section 4.2.2.2.

Initial experiments were done which included a 15-minute forecast, but the ANN results for this horizon were not good. Despite this, the RL algorithm seemed to perform well even with the bad quality 15-minute forecasts, suggesting it was able to ignore those forecasts and perform well using only the rest. In any case, due to the high uncertainty for small systems, the best forecast method for such a short horizon is likely to be simply persistence. The persistence forecast, however, would merely produce redundant values. For these reasons, the decision was to remove the 15-minute forecast entirely.

3.3.6.1 Random seed variability

During model exploration, it was observed that the random seed used for each trial (which affects model initialisation) may have an unexpectedly large effect on the final result. As will be shown ahead, this variability is not negligible, considering that the margins for maximum possible cost reduction are quite small.

It was further observed empirically that seeds which yielded better results on training data also performed better on test data, which would later be confirmed to be true for the general case.

For this reason, and since this work aims at building a well-performing model as opposed to testing or developing a new algorithm, the following methodology was adopted: 1) for each dwelling, train 5 agents on different random seeds, 2) check which agent performs better on the training data, 3) use that as the final agent, and evaluate its performance on the testing data. Further reasoning for adopting this methodology is that it would be a feasible methodology for real-life implementation (i.e., train several agents on different random seeds, implement the best one for day-to-day operation), and therefore a reasonable assessment of the expected implementation results.

Further ahead on section 4.2.2.2, the results of applying this methodology will themselves allow us to substantiate its choice.

3.3.6.2 Hyperparameter optimisation

Small experiments testing different hyperparameters generally led to either no appreciable difference, or worse-performing or even unstable agents. In fact, the trend in recent years is for new RL algorithms (including PPO) to be built in such a way that requires less and less hyperparameter tuning[68], which is why hyperparameter optimisation was not performed here, therefore simply using the default SB3 hyperparameters for PPO.

Chapter 4

Results and discussion

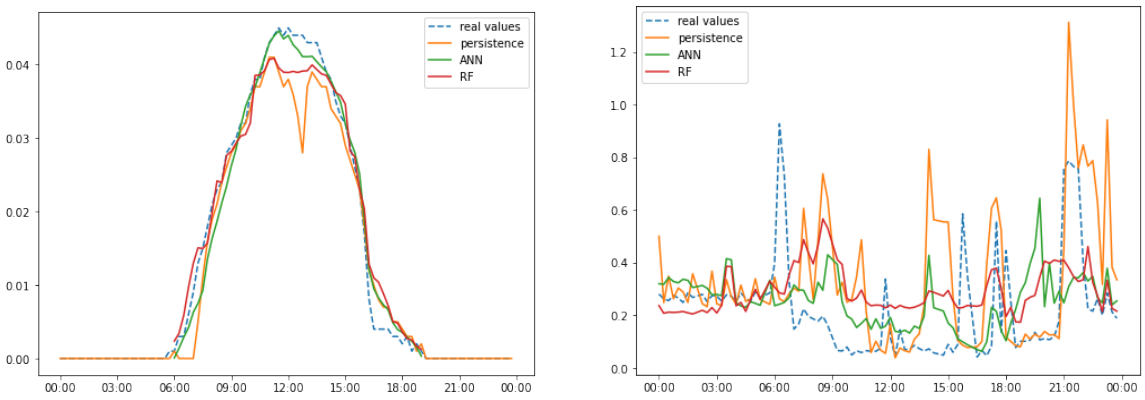
4.1 Forecasting

This section contains the results and comparison of point forecasting models, followed by cumulative forecasting models, for both PV generation and load.

4.1.1 Day-ahead point forecasting

This section presents results for the point-forecasts of PV generation and load, comparing ANN and RF models with each other, and with the baseline persistence model.

Figure 4.1.1 shows examples from exemplificative dwellings and days illustrating the different models' performance, on both PV generation and load forecasting. Figure 4.1.1a illustrates the main flaw of persistence methods: they may carry over anomalies from the previous day onto the next, when it is nonsensical to do so (see the dip in production around noon). Figure 4.1.1b typifies how load is a more stochastic variable and therefore harder to predict, particularly the large peaks. Moreover, notice how ML methods typically do not reach values as high as the peaks on the real data: predicting a peak is high risk, low reward — there is a relatively low probability that this prediction will be correct, paired with a high risk of getting a large error instead.



a PV **b** Load

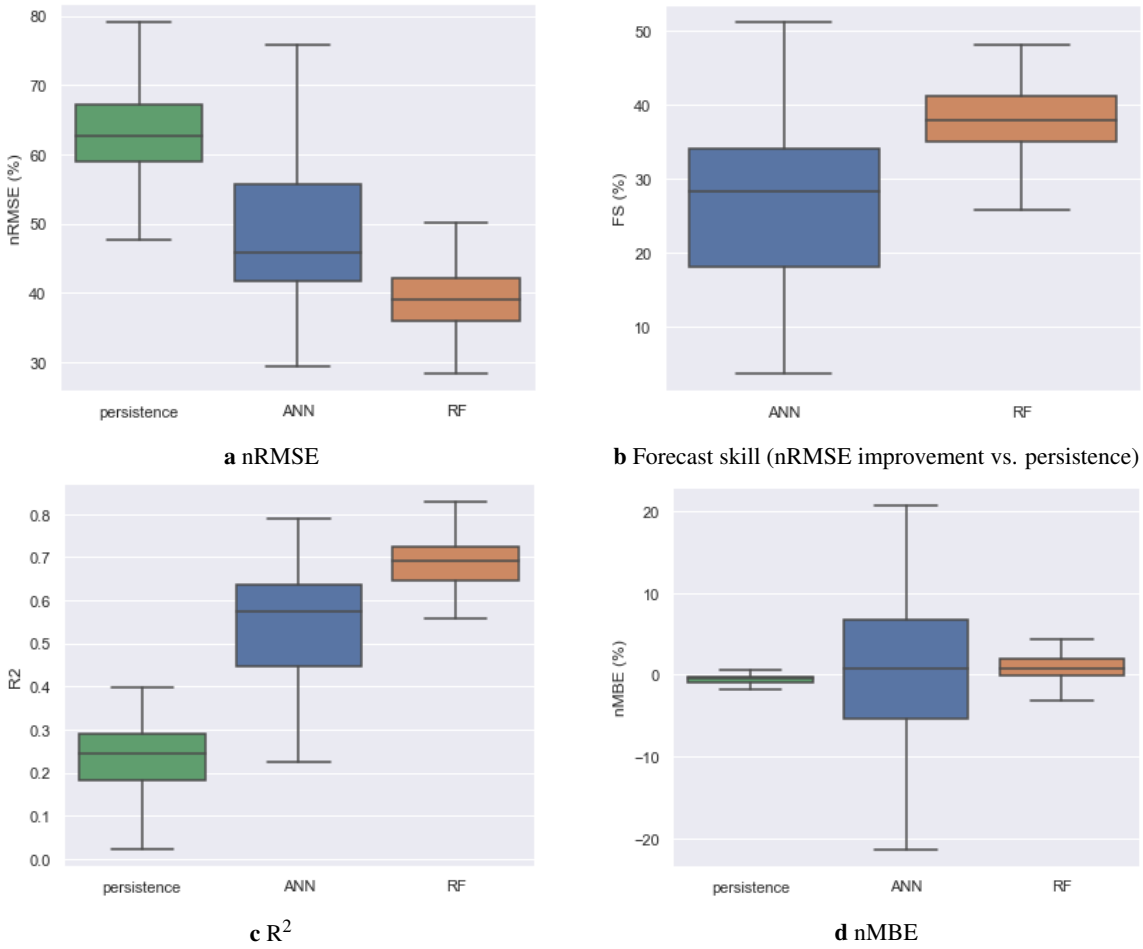
Figure 4.1.1: Daily profile examples from illustrative dwellings and illustrative days.

4.1.1.1 PV generation forecasting

Results for the PV point-forecast models are presented in figure 4.1.2. Both ANN and RF show some improvement over the persistence models, with RF being markedly better: larger median forecast skill, as well as a higher R^2, and fewer instances of very poor performance, pointing to a higher model stability. For all three models, there is a small number of very poorly performing models (most of which are not present on the plots for ease of viewing), characterised by high nRMSE (up to 149%, 353% and 97% for persistence, ANN and RF respectively), low forecast skill (as low as -503% for ANN — some of the models failed to learn and instead predict all zeros) and low R^2 (large negative values).

Persistence has the lowest nMBE, which may be expected since the distribution of predicted values is by default nearly identical to the distribution of real values. Both ANN and RF generate models with significant nMBE, with this being more pronounced for ANN, meaning that ANN may generate more models which consistently underpredict or overpredict PV generation.

Median values for metrics across all dwellings are presented in table 4.1.1.



a nRMSE

b Forecast skill (nRMSE improvement vs. persistence)

c R^2

d nMBE

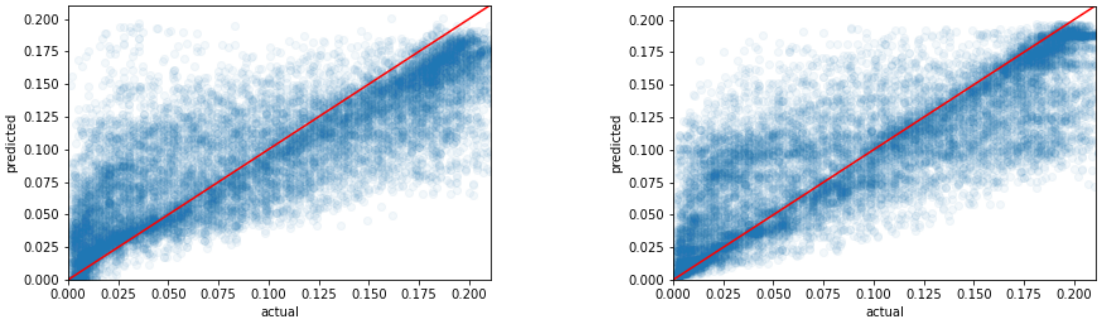
Figure 4.1.2: Model comparison for PV point-forecast

Table 4.1.1: PV point-forecast: median of each metric

	Persistence	ANN	RF
nRMSE (%)	62.6	45.8	39.1
Forecast skill (%)	-	28.2	37.9
R^2	0.24	0.57	0.69

Figure 4.1.3 shows an example of each model's performance for one particular dwelling. In a perfect forecast, predictions would exactly match reality, and every point would therefore fall on the red line. It is visible that most dots do indeed fall close to this red line, and the existing correlation between real and predicted values is clear (which is supported by the R^2 values).

In this example (which is typical, and illustrative of many others), both models tend to sometimes overpredict low values and underpredict high values (see the denser clusters of dots on the top right and bottom left of the plots). This occurs more often and is somewhat more noticeable on ANN than RF models, perhaps justifying the differences registered between the median metrics of the two models, particularly R^2.



a ANN **b** RF

Figure 4.1.3: Example of the ANN and RF models prediction accuracy on PV point-forecast for one dwelling. The red line marks where perfect predictions would fall. Darker blue represents overlapping points.

In addition, it was briefly tested whether the number of available valid data points per dwelling correlated to model performance for that dwelling (measured with forecast skill), i.e., if more data could train better models. The correlation was found to not be very relevant (0.3), suggesting that collecting data from larger periods may not be sufficient to produce better models.

Clear sky index (K_{cs})

Results of the K_{cs} experiment using both ANNs and RF are in Figure 4.1.4. Some outliers with a large negative R^2 were removed for ease of viewing; this occurred particularly using ANNs (7 dwellings displayed an R^2 smaller than -2, up to -42), suggesting once more that RF is more stable. R^2 is significantly lower than in the previous case, which was to be expected, but is still positive for the large majority of dwellings, particularly using RF, where a median of 0.2 is achieved. Knowing that K_{cs} is a highly stochastic variable and often variations may simply be unpredictable (especially without NWP data), this shows that the models are sometimes indeed able to predict variability beyond simply that resulting from sun movement — particularly the RF model.

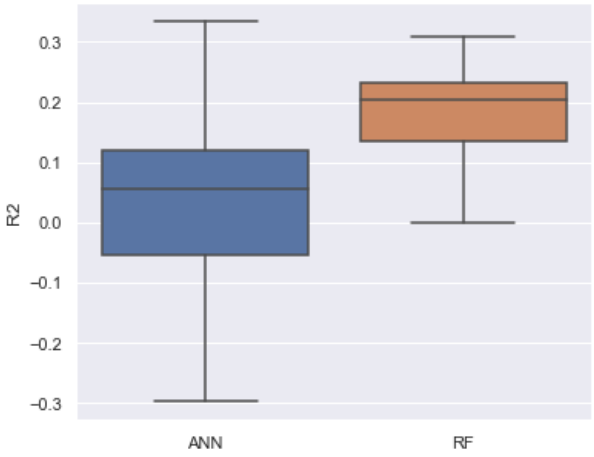
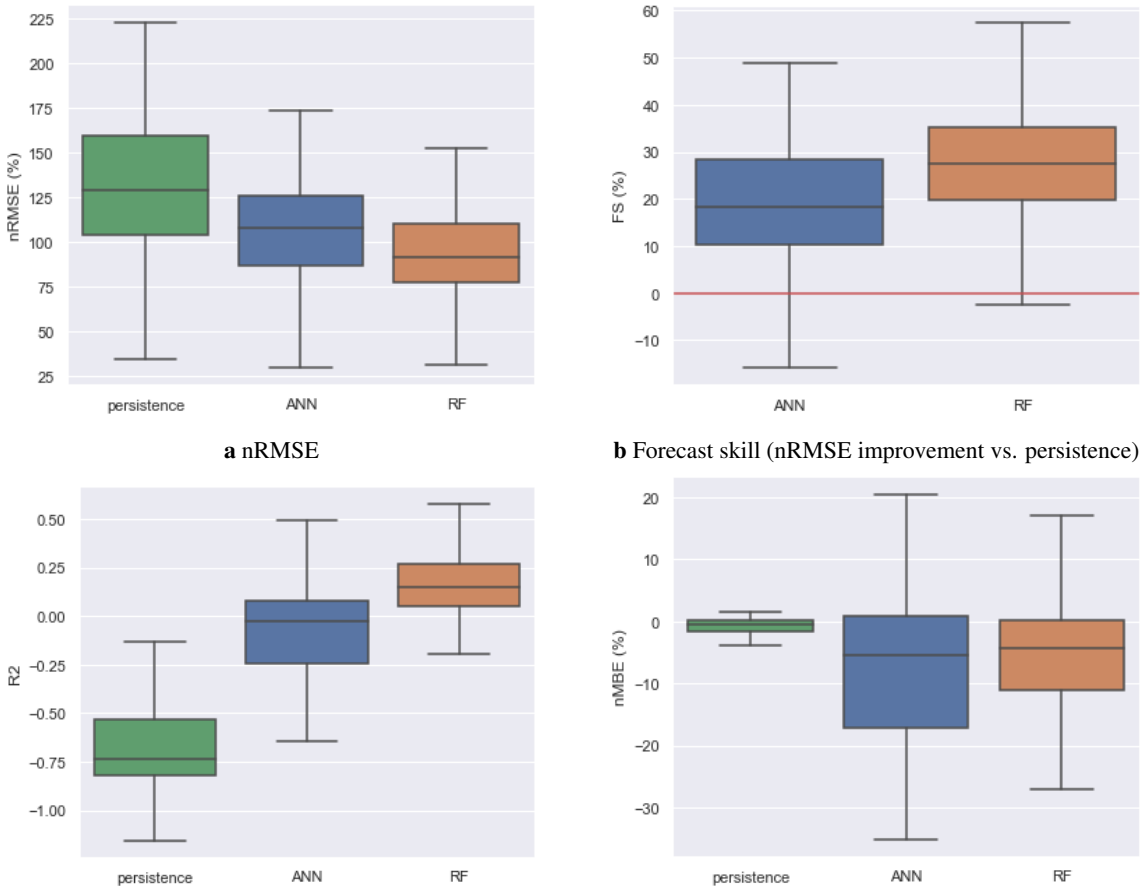


Figure 4.1.4: R^2 of single-point forecast of K_{cs} for all dwellings.

4.1.1.2 Load forecasting

Results on load forecasting (figure 4.1.5) show a similar trend to PV forecasting: both ANN and RF represent an improvement over persistence, with RF showing the best performance, both in terms of nRMSE, R^2 and nMBE. Median values for metrics are shown in table 4.1.2.



a nRMSE

b Forecast skill (nRMSE improvement vs. persistence)

c R^2

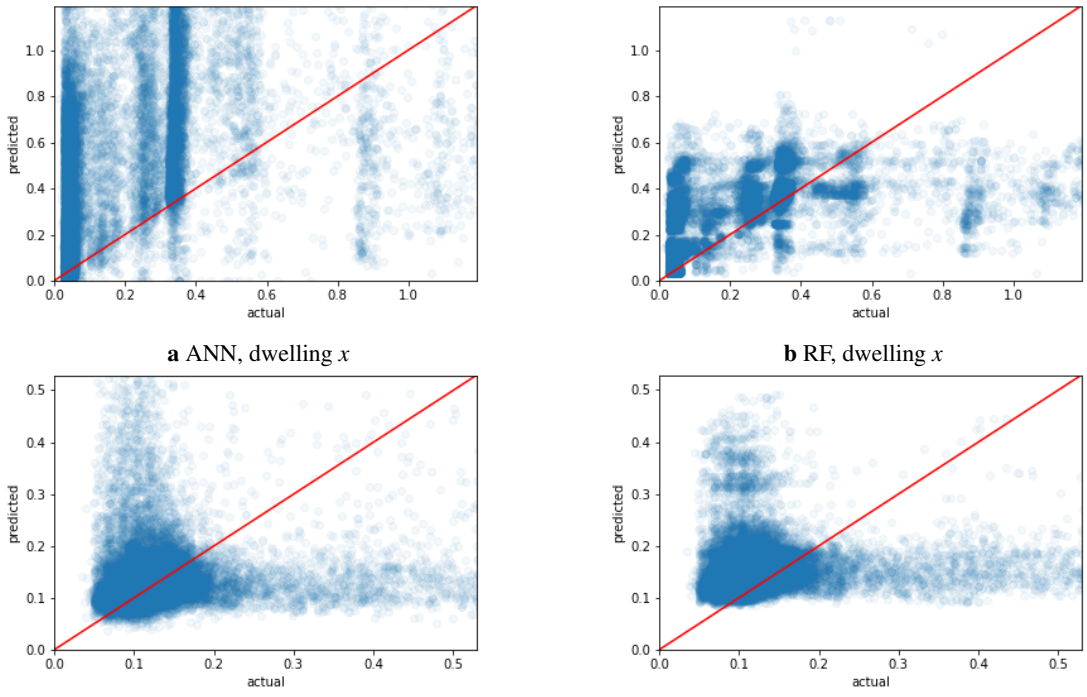
d nMBE

Figure 4.1.5: Model comparison for load point-forecast

Table 4.1.2: Load point-forecast: median of each metric

	Persistence	ANN	RF
nRMSE (%)	128.8	108.0	91.6
Forecast skill (%)	-	18.4	27.4
R^2	-0.74	-0.03	0.15

In figure 4.1.6 we can compare ANN and RF performance, for two dwellings. These display two typical patterns representative of those observed in many dwellings. For dwelling *x*, on most timesteps, load values fall within several almost-discrete ranges of values. RF is able to recognise this and accurately predict which range *some* of the time, while also failing often. Notice as well how the model seldom predicts values higher than 0.7, despite the fact that these exist, although in smaller quantity. Predicting these large values represent the high-risk, low-reward situations mentioned earlier, which is why the model largely ignores them altogether. On dwelling *y*, both models fail to learn any meaningful correlation, as evidenced by the lack of dots near the red line for larger values. Small values are predicted more often and, because they are more common, are often correct, while large values are never accurately predicted. This is a typical example of a model which would have a relatively low nRMSE, but also a very low R^2.



a ANN, dwelling *x*

b RF, dwelling *x*

c ANN, dwelling *y*

d RF, dwelling *y*

Figure 4.1.6: Example of the ANN model prediction accuracy on load point-forecast for two dwellings. The red line marks where perfect predictions would fall. Darker blue represents overlapping points.

Other typical patterns include always predicting values in a single narrow range, such as close to the mean, or otherwise close to the dwelling's base load. The common thread amongst most cases is a weak or otherwise nonexistent correlation between real and predicted values.

As we will see ahead, the main reason for this is essentially the high granularity. Human behaviour, which reflects on domestic load, is somewhat consistent and may lead to particular load patterns for each

dwelling, but this consistency may not be at such a high granularity as that which is considered here as timestep duration: 15 minutes. Results from the cumulative forecast (section 4.1.2) show that reducing the granularity by considering one-hour timesteps leads to significant improvements.

It may also be that the features used here are insufficient — features such as temperature and occupation are unavailable, and could possibly contribute to a more accurate forecast.

4.1.2 Cumulative forecasts

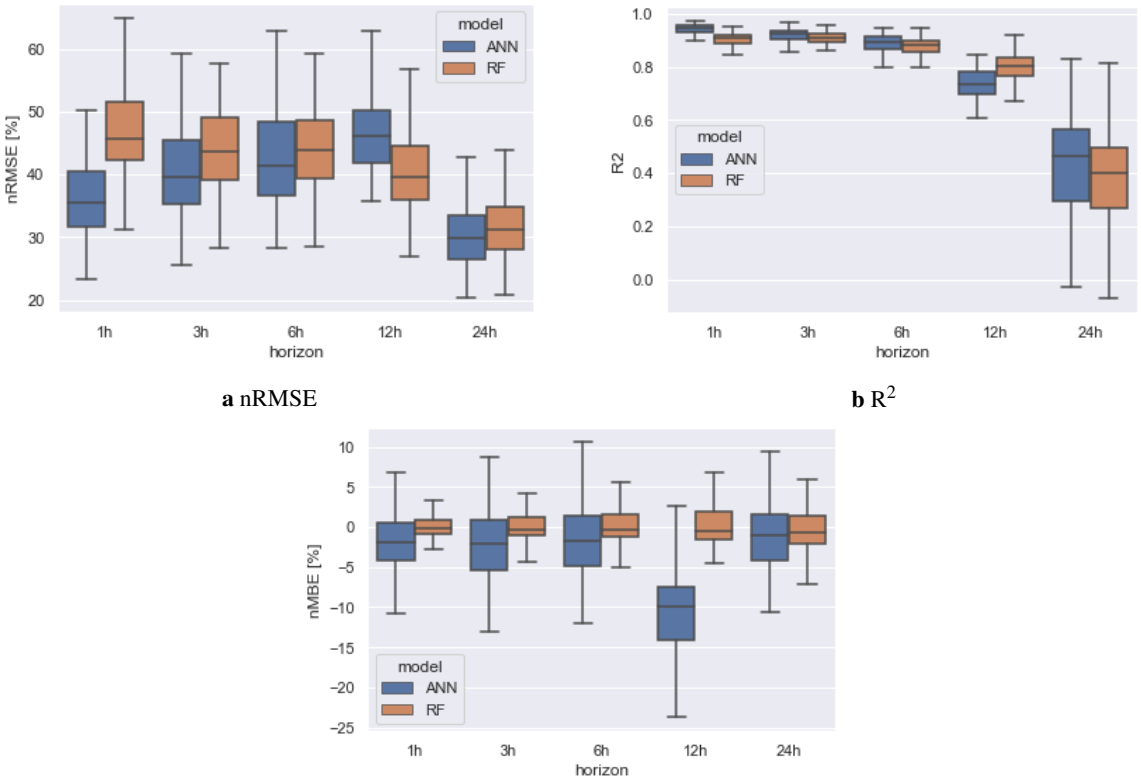
As stated previously, cumulative forecasts are used in this work to present comprehensive information to the RL agent about the following 24 hours in a compact way, thus avoiding the curse of dimensionality. This section presents performance metrics of the models developed here for cumulative forecasting of PV generation and load.

4.1.2.1 PV generation cumulative forecasting

Figure 4.1.7 shows the comparison of metrics for cumulative PV forecasting between the ANN and RF models for the five different horizons. ANN generally outperforms RF in terms of nRMSE and R^2 on all horizons, except the 12 hour horizon. Moreover, noteworthy analysis to be drawn from the metrics includes:

- R^2 decreases as horizon length increases, with a sharp decrease for the 24h horizon. Analysing the scatter plots of real vs predicted values for this horizon shows that indeed predictions become more dispersed, in some cases with the range of predicted values being narrower than the range of real values, suggesting the network has some trouble learning meaningful correlations between features, and may learn to predict values close to the mean instead, in order to minimise loss (similar to, but not as pronounced as, what happened on the load point-forecast).
- For both models, the 24h horizon has the lowest nRMSE, despite also having the lowest R^2. This is likely because this value is less variable than the others, i.e., its mean is not dependant on time of day, whereas it is for other values. This makes sense when coupled with the previous explanation: always predicting values close to the mean has decent likelihood of being successful if that mean changes very little throughout time.
- RF typically has a smaller nMBE than ANN, which is particularly striking for the 12 hour horizon, but is present in all others.
- Whereas for ANN the error increases for larger horizons, it decreases for RF.

Appendix A contains scatter plots showing illustrative examples of ANN and RF model performance on cumulative forecasting for all horizons. The motive for ANN's poor performance on the 12-hour horizon is also well illustrated there: this model makes a number of zero predictions for non-zero real values. This explains the large negative bias visible on Figure 4.1.7c.



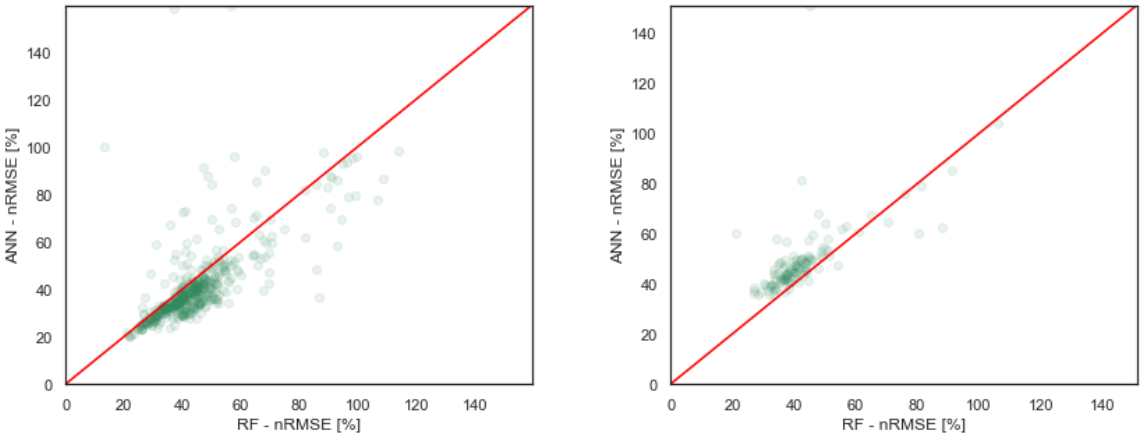
a nRMSE

b R^2

c nMBE

Figure 4.1.7: Model comparison for PV cumulative forecast

Moreover, figure 4.1.8 shows individual dwellings as dots, comparing the performance, via nRMSE, of ANN and RF models on each dwelling. This further illustrates how ANNs typically perform better than RF on most dwellings for all horizons (most dots are below the identity line) except the 12 hour horizon, where the opposite is true. Furthermore, it shows that the performance of one method on each specific dwelling correlates with the performance of the other method for the same dwelling. This suggests some dwellings are inherently more difficult to forecast than others, which could be due to climate or other factors.



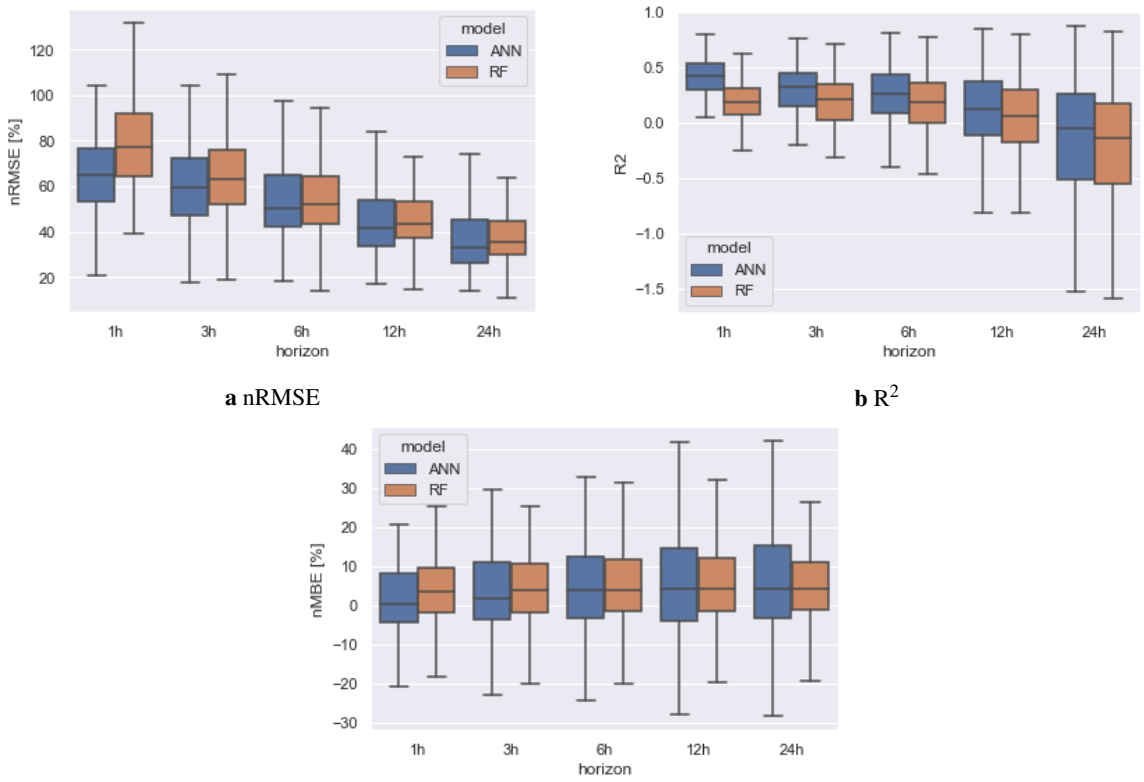
a 1, 3, 6 and 24 hours

b 12 hours

Figure 4.1.8: nRMSE comparison for cumulative PV forecasting for individual dwellings, different horizons. On the left, 1, 3, 6 and 24 hour horizons can be seen together; on the right, only the 12 hour horizon.

4.1.2.2 Load cumulative forecasting

Figure 4.1.9 compares metrics for cumulative PV forecasting between the ANN and RF models for the five different horizons. ANN now outperforms RF on all horizons, when considering both nRMSE and R^2. In this case, the difference in nMBE, while still favouring RF, is less striking than before.



a nRMSE

b R^2

c nMBE

Figure 4.1.9: Model comparison for load cumulative forecast

Moreover, it is visible that, in this case, for both models, nRMSE decreases for longer horizons. Simultaneously, R^2 decreases as well. this is explained by the fact that larger timesteps generate smoother values, with smaller *relative* variability: e.g., for the 24h horizon, on most dwellings, there are no small values. Instead, all values are concentrated on a certain range of relatively high values and, within that range, the relative variability is smaller when compared to the mean value — generating a low nRMSE, even if the correlation between real and predicted values is quite low. This is illustrated by the example shown on figure 4.1.10. In other words, nRMSE decreasing for larger horizons does not mean that the forecasts are better at predicting these horizons, but that these horizons are easier to predict.

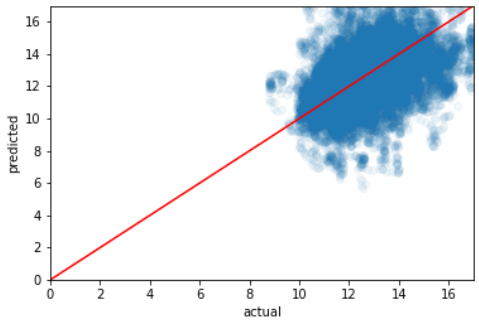


Figure 4.1.10: Scatter plot for the 24h cumulative load forecast on one dwelling. For this dwelling, nRMSE is 18% and R^2 is 0.1.

As in PV forecasting, the performance of both models on each dwelling is correlated with one another, with the difference that ANN is now consistently better on most dwellings, regardless of horizon (see figure 4.1.11). The correlation here also suggests that some dwellings are more difficult to forecast than others, in this case likely due to more erratic behaviour on the occupants' part.

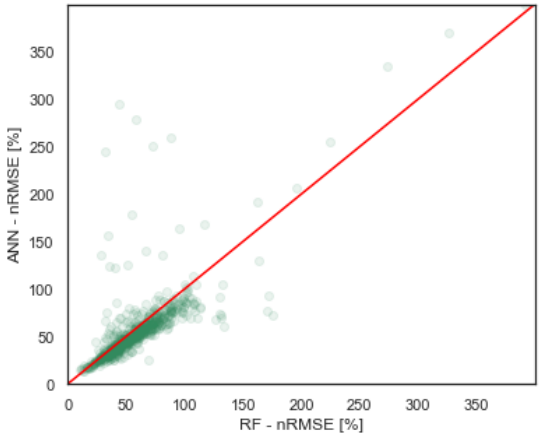


Figure 4.1.11: nRMSE comparison for cumulative load forecasting for individual dwellings, all horizons.

4.1.3 Forecasting summary

To summarise, point-forecasts generally favour RF models, and these also have the advantage of being lighter, requiring less preprocessing and running faster than ANNs. RF generally performs better on all metrics than ANN for these forecasts, for both PV generation and load. Additionally, for PV forecasting, the small K_{cs} forecast experiment shows some ability for these forecasts to predict variability other than simply that caused by sun movement, which emphasises the advantage of using a ML model.

Contrary to this, on cumulative forecasts, ANN models generally outperform RF when considering both nRMSE and R^2. At large, their most prominent issue when compared to RF seems to be their higher bias (nMBE). However, the first two were considered to be more important when going into the following phase: since bias tends to occur in a particular direction, the reasoning was that the RL agent for the HEMS may be able to learn to account for it. Hence, ANNs were chosen to perform the forecasts which will be fed into the HEMS RL agent.

4.2 Home Energy Management System

This section presents the results for the RL-based HEMS strategy developed in this work. It begins by establishing benchmarks with the purpose of calculating maximum potential savings, as well as comparing different tariff structures under the various benchmark methods. For the RL agent, previously stated choices are substantiated and justified, and finally the results of the seven formulated scenarios are presented.

4.2.1 Benchmarking

This section establishes benchmarks through the traditionally-used rule-based method SCM and the optimisation method MILP (providing the optimal scenario), thereby establishing the maximum potential savings. Additionally, a brief comparison of tariffs under the benchmark methods is performed, in an attempt to find trends in which tariffs may be more beneficial for users under each situation.

4.2.1.1 MILP vs. SCM: potential for improvement

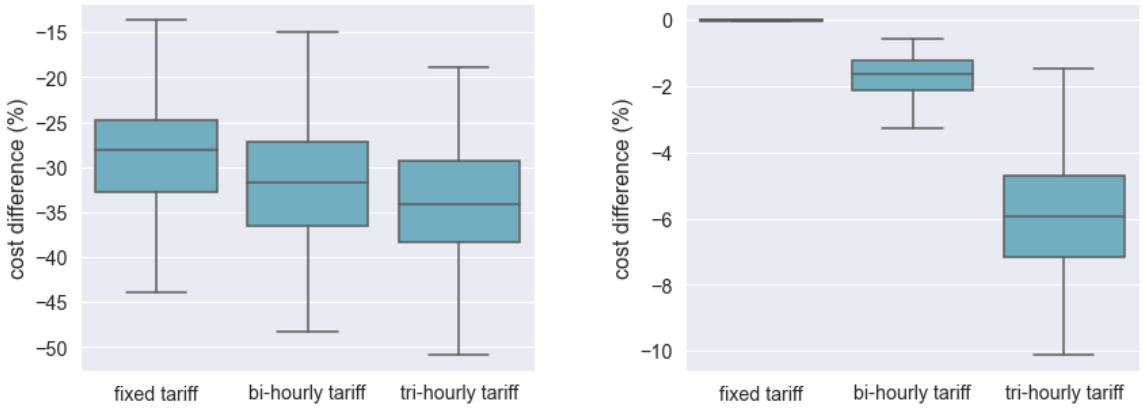
MILP and SCM were used for benchmarking purposes, with the main goal of assessing the maximum potential cost savings, but in the process they allow us to draw additional conclusions.

Figure 4.2.1a shows the cost savings from using a battery with an SCM strategy, when compared to using no energy storage. For all three analysed tariffs, it shows significant cost savings (28 to 34% for the median user), showing that a battery provides significant value even if using a simple rule-based strategy.

Figure 4.2.1b shows the cost savings from the MILP solution when compared to the SCM solution. These confirm how, for the fixed tariff, SCM yields the optimal solution, as evidenced by the fact that MILP is shown to be equivalent to SCM for that case.

The same figure also shows that there is potential for a median cost reduction of 5.9%, when compared to SCM, if using the tri-hourly tariff. This seemingly points, a priori, to a greater potential for energy savings when using RL with the tri-hourly tariff, as long as the RL agent is able to take advantage of that potential. Moreover, it shows that **all** dwellings show **some** potential for energy savings, that is, SCM is never the optimal solution when using the bi- and tri-hourly tariffs.

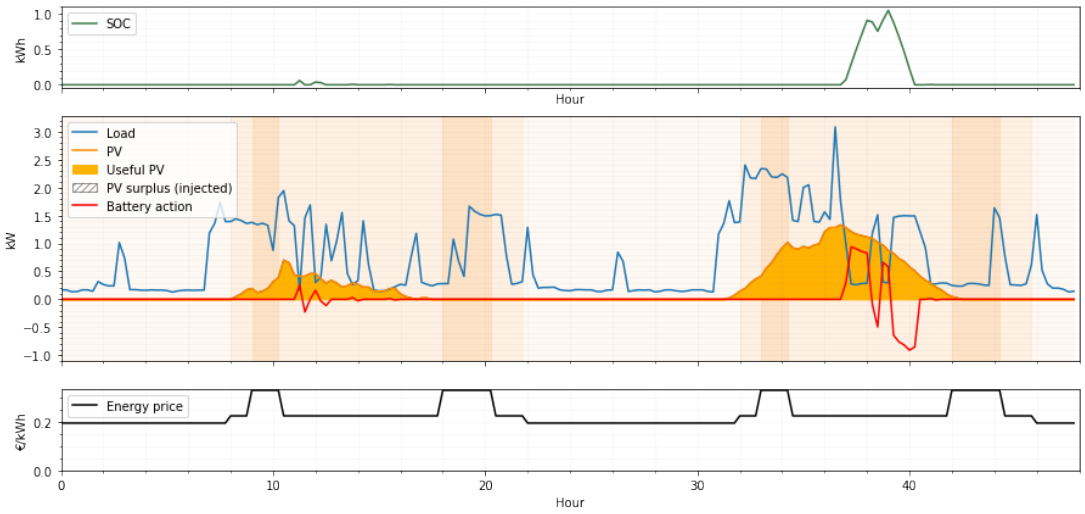
For most cases, the potential savings, illustrated by MILP, are quite small, suggesting that SCM is a quite well-adjusted strategy to present circumstances. In fact, the largest potential for cost savings when comparing SCM to the MILP optimal solution for bi- and tri-hourly tariffs seems to be in pre-charging the battery from the grid during periods with low energy prices **if** not enough PV energy will be generated in the near future to cover load. By pre-charging the battery in these situations, one avoids the need to purchase energy from the grid when the energy prices are at their highest, thus saving money. An example of this can be observed in Figure 4.2.2.



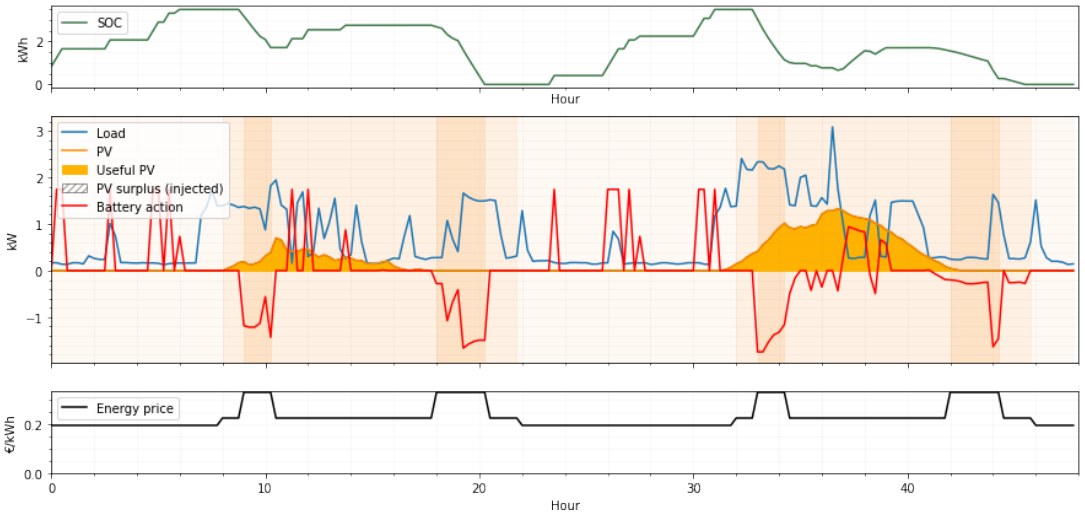
a Cost reduction from using a battery with SCM for energy management vs. no battery

b Cost reduction from using MILP (optimal case) vs. SCM

Figure 4.2.1: Energy cost reductions using a battery



a SCM — no pre-charging

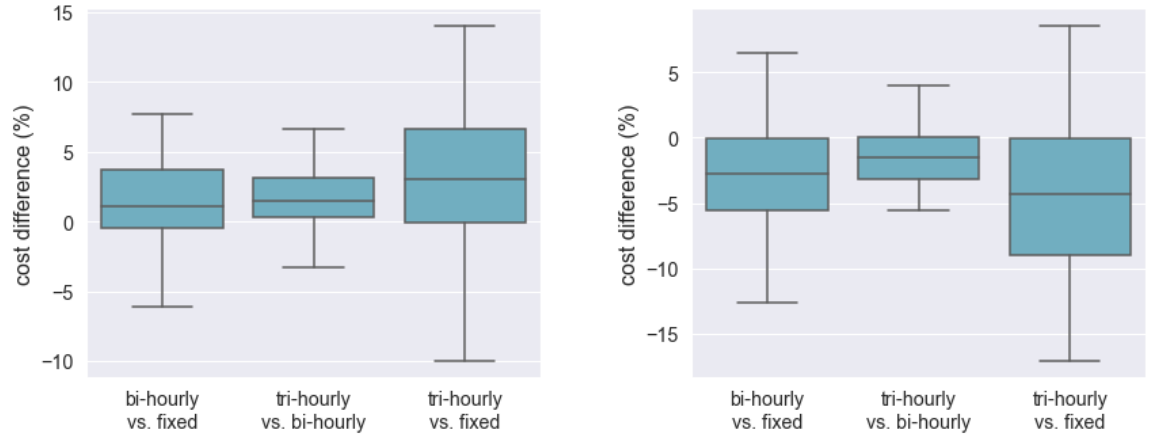


b MILP — pre-charging

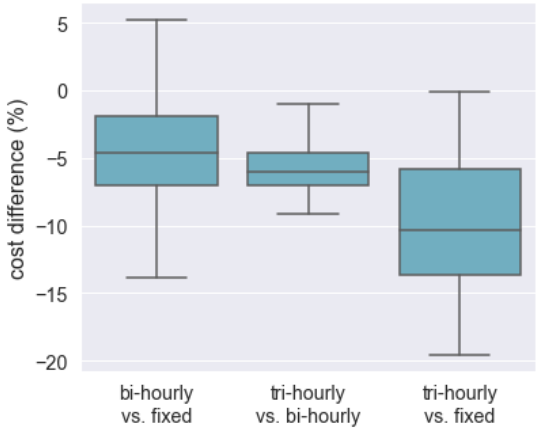
Figure 4.2.2: One example of two cloudy days managed by SCM and by MILP. Due to scarce PV generation, SCM barely uses the battery. MILP is able to perform pre-charging mostly during the night, when prices are low, for use during higher-price periods.

4.2.1.2 Tariff comparison

Further analysis can be done in order to determine the benefits of using each tariff with each method. For this, the costs resulting from each tariff were computed using the various methods, and results are compared in Figure 4.2.3.



a Cost comparison of different tariffs without using a battery **b** Cost comparison of different tariffs using a battery with SCM



c Cost comparison of different tariffs using a battery with MILP

Figure 4.2.3: Energy cost comparisons of different tariffs

Figure 4.2.3a compares energy costs resulting from different tariffs *without* using a battery. Curiously, it shows that, for the median user who does not own a battery, a fixed tariff is slightly better than both the bi- or tri-hourly tariffs. In addition, without a battery, the tri-hourly tariff is, for the median user, slightly worse than the bi-hourly tariff. This may be because, while the fixed tariff is more expensive than the off-peak tariff for bi- and tri-hourly tariff schemes, without a battery, users will more frequently need to use grid energy during peak hours (during which there may be insufficient or nonexistent PV generation), with peak hours having much higher energy prices. We can further speculate that this may be affected by the fact that the data collection period includes the COVID-19 pandemic, and therefore many people spent more time at home during the day than usual, leading to increased load during the day — including periods with higher prices for bi- and tri-hourly tariffs.

Figure 4.2.3b compares tariffs for users with a battery employing SCM as an energy management strategy. In this case, the tri-hourly tariff offers the most savings for the median user, but can be the worst option for some users (around 22%).

Finally, Figure 4.2.3c compares tariffs using MILP. With MILP, the tri-hourly tariff offers the most savings for the median user, being only somewhat worse than others for some users. In fact, generally, the more complex a tariff structure is, the more potential it will present for exploit by an omniscient algorithm such as MILP providing the optimal solution. We will show further ahead (section 4.2.2.3) that this does not mean that this full potential can be harnessed by regular algorithms performing in real time.

4.2.2 Reinforcement Learning

This section first justifies and substantiates previously-made choices regarding the creation of the hybrid RL/rule-based model and the use of different random seeds for each house. Finally, it presents the main results of this dissertation, revealing the hybrid model's performance under six different scenarios, and comparing it with the benchmark methods, to ascertain the real benefits the hybrid model provides.

4.2.2.1 Pure RL vs. RL/rule-based hybrid

The pure RL model was used only on one scenario due to poor results. As a reminder, this model used perfect forecasts and a tri-hourly tariff — the situation with largest potential for improvement. These results are displayed in Figure 4.2.4, showing box plots of the added cost of using the pure RL model when compared to the RL/rule-based hybrid and to SCM. The pure RL model performed worse (got a higher total cost) than both others **on every single dwelling** (added cost is positive in every instance), in most cases performing significantly worse, hence why it was abandoned.

These large differences observed are sure to be in part due to flawed RL models, but also to the inherent uncertainty in future quantities (15 minutes-ahead load and PV generation), which is why rule-based decisions **in real time** significantly improve performance, when compared to the pure RL model, which makes decisions for the following 15 minutes without access to accurate information for that time period.

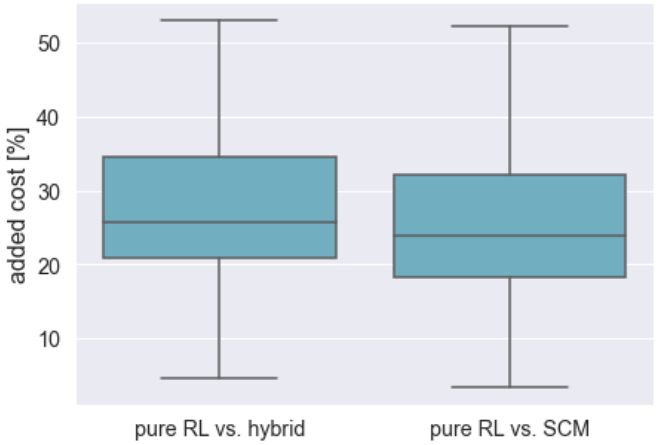
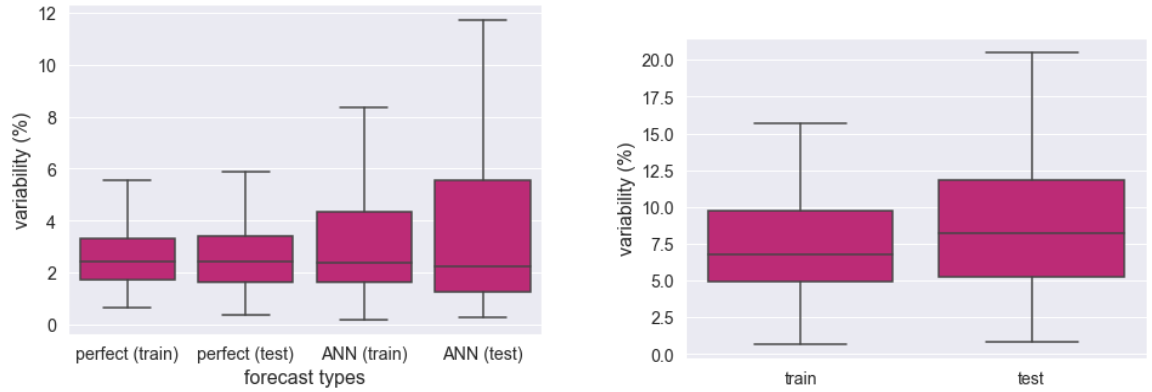


Figure 4.2.4: Added cost [%] of using the pure RL model when compared to the RL/rule-based hybrid and SCM.

4.2.2.2 Random seeds, variability and reproducibility

Having applied the methodology described in section 3.3.6.1, it was possible to verify that the aforementioned observations detected in preliminary experiments were indeed true for the general case. Figure 4.2.5 shows seed variability per dwelling (defined as the cost difference between min and max costs resulting from the different seeds) for different scenarios. Note how variability is largest for the pure RL scenario (without enforcing optimal actions). This was expected, considering that the action is determined solely by the RL agent. For RL/rule-based hybrid scenarios, variability is lower, but its median still lies in the [2, 2.5]% range, which is very relevant considering the median potential cost reduction of 5.9% obtained using MILP (Figure 4.2.1b). Note also how variability reaches larger values using the ANN forecast (despite similar medians), which may be explained by the added entropy from inaccurate forecasts.



a Hybrid RL/rule-based scenarios

b Pure RL scenario (without enforcing optimal actions)

Figure 4.2.5: Seed variability (cost difference between min and max costs) for some scenarios, using a tri-hourly tariff.

Figure 4.2.6 shows, for different scenarios, the correlation between seed performance on the training data and on the test data, computed for each dwelling. It shows that this correlation is strongest for the pure RL scenario, but is quite marked for all three scenarios, thus supporting the chosen methodology (i.e., one can reasonably expect that an agent which performs well on the training data will also perform well on the test data).

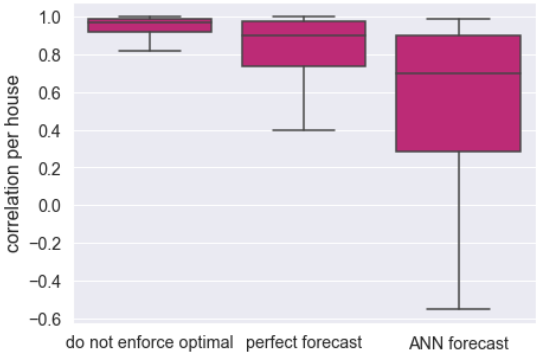
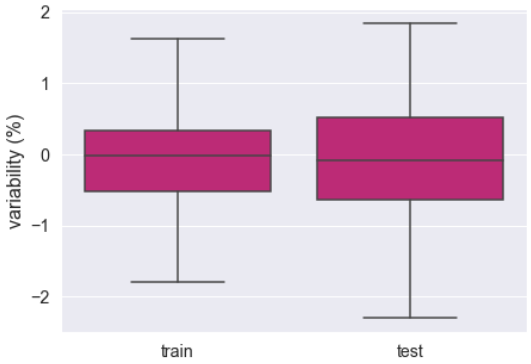
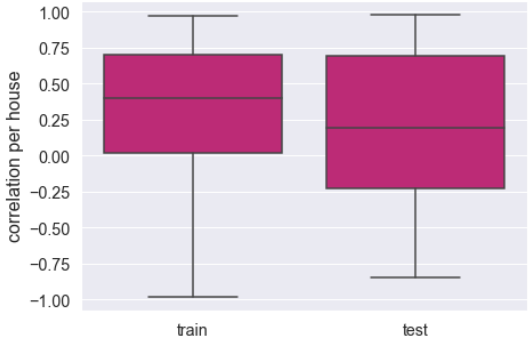


Figure 4.2.6: Correlation between performance on the training data and performance on the same data for the same seed. Correlation is computed for each dwelling.

Section 3.3.5 mentions how model-free methods usually require many timesteps to learn satisfactorily, but also that only one passage through the training data was used on this work. One experiment was in fact done with more training timesteps (specifically, 4 runs through the training data) to observe whether there would be an appreciable difference. Figure 4.2.7 shows the results. Figure 4.2.7a shows the variability (i.e., cost difference) between the best-performing seed for each case (a negative variability here would mean that 4 training runs leads to cost savings when compared to 1 training run). Results show the lack of an significant difference: variability can be both positive or negative, and the median is very close to zero (0.1% for test data) meaning that in close to half the cases, 4-run training actually leads to *worse* results than 1-run training. Furthermore, another interesting observation was made from Figure 4.2.7b: performance on 1-run and 4-run training *for the same seed* shows a very low correlation, meaning that seeds which perform better on 1 training run did not necessarily perform better when trained on 4 runs. Therefore, while Figure 4.2.6 suggests that variability in performance is not due to randomness but indeed due to the fact that some agents become better-trained than others, Figure 4.2.7b suggests that, contrary to what was postulated, this is not due to good solutions becoming inaccessible with certain seeds due to a "bad" initialisation, since a well-performing agent can become worse after further training, or vice-versa. Instead, this is likely due to random fluctuations during training, which are common in RL due to the inherent stochastic nature of training.



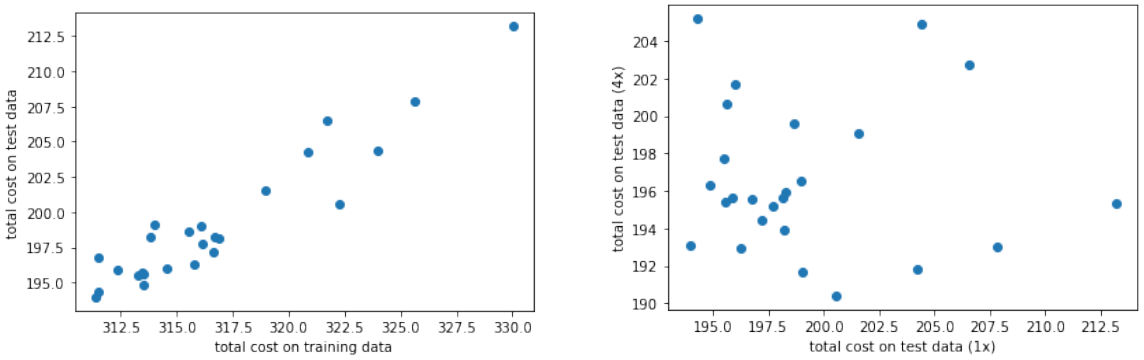
a Cost difference [%] between the best seed between the 4-runs training and 1-run training.



b Correlation between cost for the 4-runs training and 1-run training, for the same seeds.

Figure 4.2.7: Comparison of the results of 1 run and 4 runs through the training data, for the tri-hourly tariff, perfect forecast scenario (RL/rule-based).

This point is perhaps better illustrated by observing results for one specific dwelling, for which 25 different seeds were used as an experiment (Figure 4.2.8). The clear correlation can be seen on Figure 4.2.8a, while a clear lack of correlation is also observable on Figure 4.2.8b.



a Performance on training data vs. performance on test data **b** Performance after training on one run of the training data vs. after training on four runs

Figure 4.2.8: Seed correlation scatter plots, tested on one dwelling (25 seeds). Each point corresponds to one seed.

4.2.2.3 Model performance and forecast impact

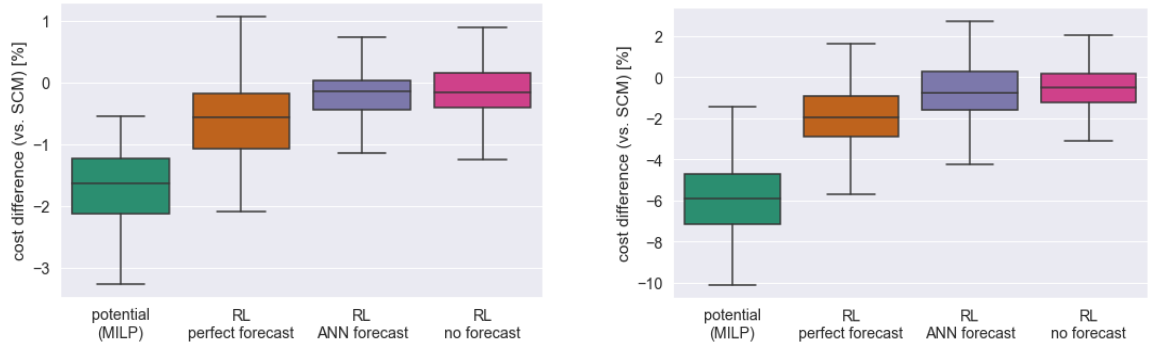
This section contains the final results of this work, separated by tariff type. Figure 4.2.9 shows the various scenarios when compared with SCM, with MILP juxtaposed again for easy comparison. The respective medians can be consulted on table 4.2.1. Notice how the RL/rule-based hybrid method, even when using perfect forecasts, does not come very close to the potential savings, for both tariffs. Notice also how using the ANN forecasts presents little to no advantage over using no forecasts at all, which suggests that a high degree of accuracy is needed for the forecasts to be useful, and this may simply not be attained in most dwellings.

Figure 4.2.10 shows the cost difference of using a perfect forecast or an ANN forecast versus using no forecast at all. In both tariffs, the perfect forecast is able to show improvements in most cases, but for the ANN forecast these improvements are small or non-existent: in fact, the median is close to zero for both tariffs, meaning that using ANN forecasts is not consistently better than not using forecasts.

This point is further illustrated by Figure 4.2.11. We previously defined potential cost savings as the difference between SCM and MILP costs, considering MILP gives the optimal solution. The figure shows the fulfilled potential [%], i.e., the percentage of potential cost savings which are indeed attained by the hybrid models. Interestingly, these results are similar for both tariffs. It is visible once more how there is little difference between using the ANN forecast and no forecast. Even for perfect forecasts, the median fulfilled potential is only 33% for the bi-hourly tariff and 29% using the tri-hourly tariff. It is, however, lower for the other two (see Table 4.2.2), which show more frequent cases of results *worse* than SCM, which happens very little when using perfect forecasts.

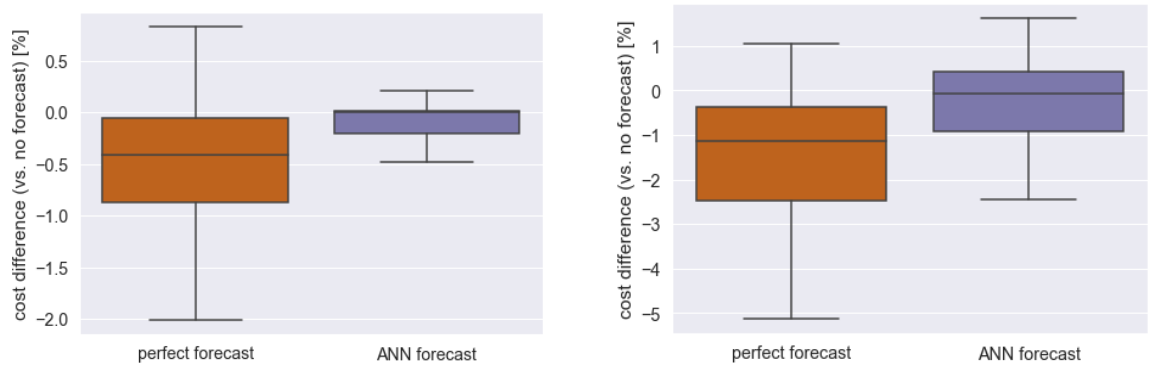
Table 4.2.1: Median cost difference [%] when compared to SCM

	Bi-hourly tariff	Tri-hourly tariff
MILP	-1.6	-5.9
RL - Perfect forecast	-0.5	-1.9
RL - ANN forecast	-0.1	-0.7
RL - No forecast	-0.2	-0.5



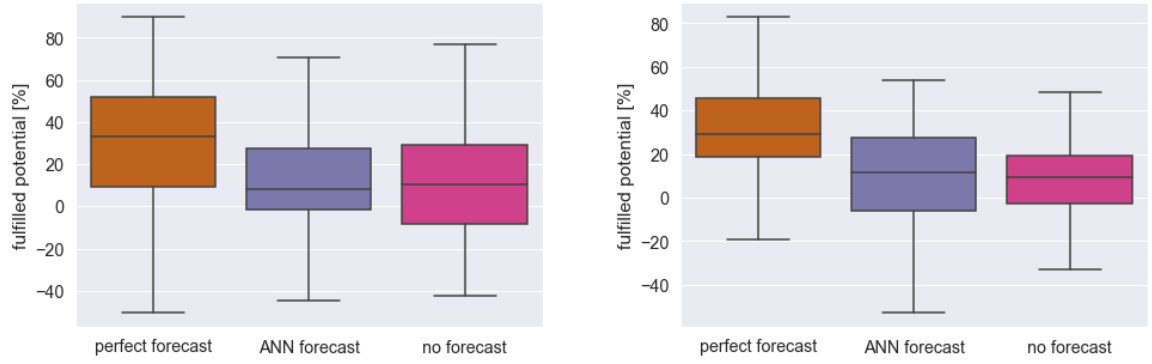
a Bi-hourly tariff **b** Tri-hourly tariff

Figure 4.2.9: Cost difference [%] of various HEMS variations when compared to SCM, for bi- and tri-hourly tariffs



a Bi-hourly tariff **b** Tri-hourly tariff

Figure 4.2.10: Cost difference [%] of various HEMS scenarios when compared to using no forecast, for bi- and tri-hourly tariffs



a Bi-hourly tariff **b** Tri-hourly tariff

Figure 4.2.11: Fulfilled potential [%] of various HEMS variations when compared to SCM, defined as the ratio of observed cost savings to *potential* cost savings according to MILP results, for bi- and tri-hourly tariffs

Table 4.2.2: Median of fulfilled potential [%] (portion of potential savings that were in fact achieved)

	Bi-hourly tariff	Tri-hourly tariff
Perfect forecast	33	29
ANN forecast	8	11
No forecast	11	9

This shows, then, that forecast quality is important, but can only do so much. An additional observation was made which may deepen this point. For the tri-hourly tariff, the correlation was computed between forecast errors and the fulfilled potential of the hybrid model using the ANN forecast. This correlation was near-zero for the PV generation forecasts, suggesting PV forecast quality does not significantly affect model performance. Conversely, for load forecasts, a significant **negative** correlation was observed between forecast error and fulfilled potential — particularly for the 24-hour horizon ($r = -0.48$) and the 12-hour horizon ($r = -0.32$).

As this could, at first glance, simply be a result of some dwellings having a more consistent load profile and therefore being easier to predict, while simultaneously being easier to manage. For this reason, in order to further evaluate how data variability could impact this correlation (perhaps leading to misleading results), a further experiment was performed: computing the correlation between fulfilled potential and the forecast errors for the no forecast and perfect forecast scenarios. With the perfect forecast, the negative correlation was still present for the same variables, but was significantly lower ($r = -0.28$ for 24h and $r = -0.26$ for 12h). With no forecast, no correlation was present.

This strongly suggests that load forecast quality, particularly for the 12-hour and 24-hour horizons, is the most important for model quality. The author speculates here that this may be due to the more long-term planning involved in pre-charging the battery from the grid with low prices if necessary. Furthermore, these results may also suggest that PV generation forecasts are somewhat unimportant for model performance — and, therefore, using NWP data to improve PV generation forecasts may not present an advantage. These points are further explored in Appendix B.1.

A further interesting discovery was made as a result of this: the correlation between the fulfilled potentials for these three different scenarios is relatively low — i.e., one dwelling obtaining a good performance on one scenario does not guarantee that it will obtain an equally good performance for another.

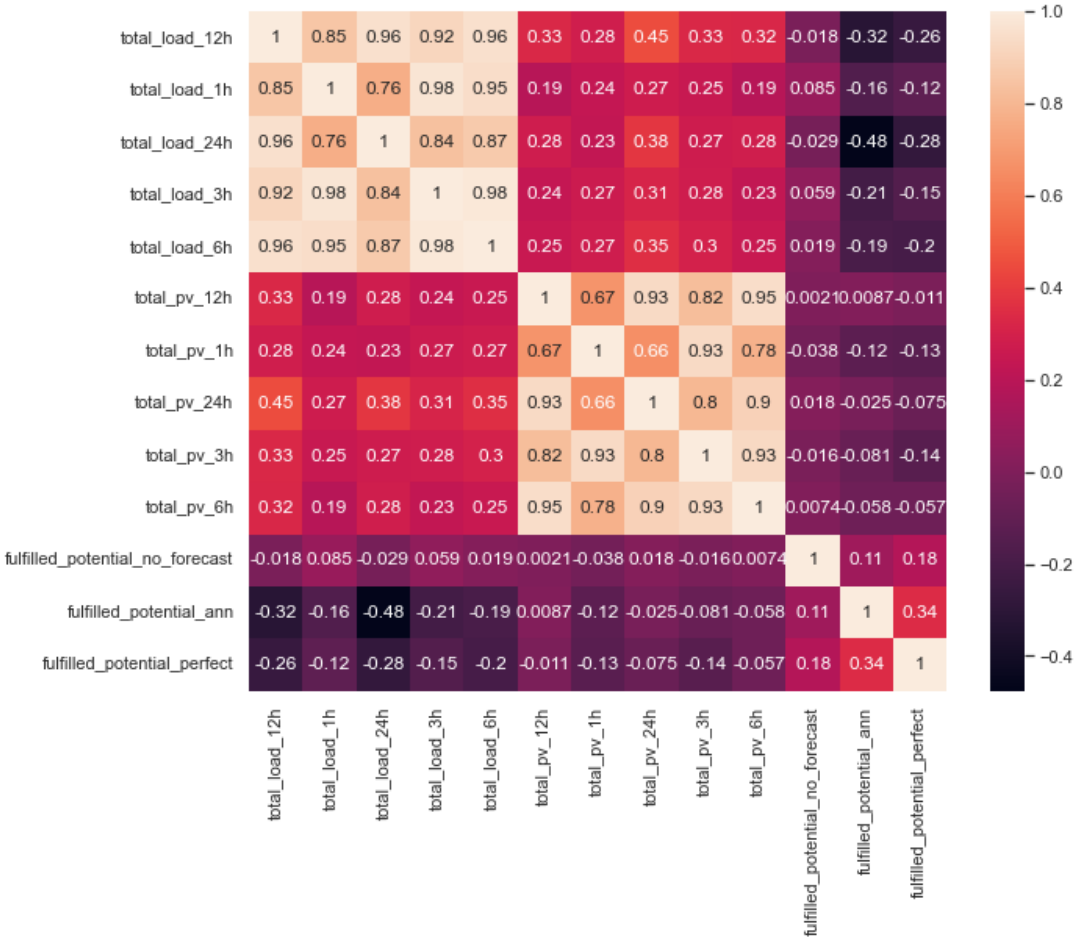
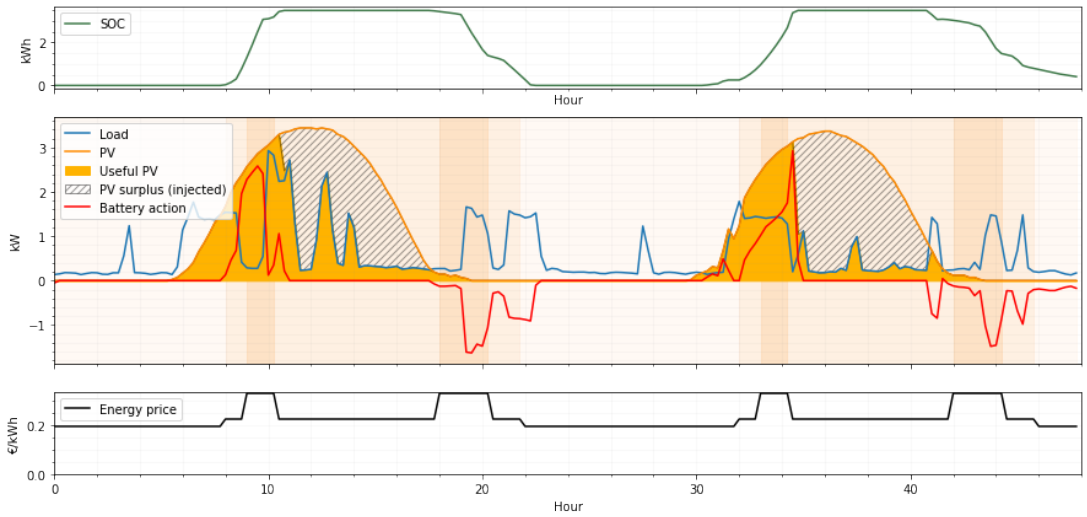


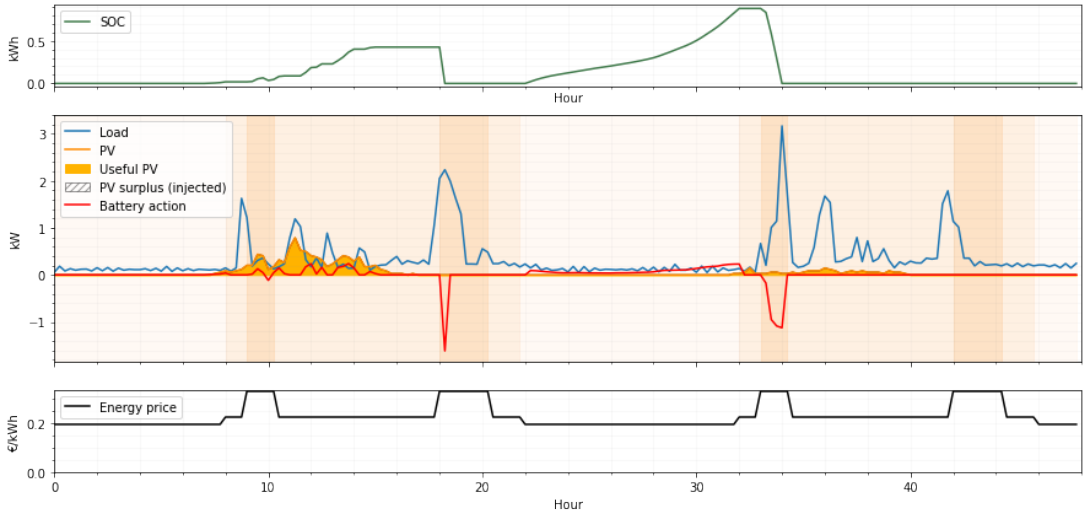
Figure 4.2.12: Correlation heatmap between the fulfilled potential of several scenarios and forecast nRMSE

Figure 4.2.13 shows two examples of model performance, each over a period of two days, one sunny and one cloudy. Figure 4.2.13a shows a sunny period with a smooth PV generation curve. The battery is charged in the mornings and is full before noon, meaning that excess PV generation for the remainder of the day is injected into the grid. Figure 4.2.13b shows cloudy days, with scant PV generation, where multiple instances of pre-charging using lower prices can be seen. On the second day, with lower PV generation than the first, the model is able to perform pre-charging during nighttime to help cover the load peak which is visible in the morning. On the first day, it is also able to hold off on spending battery-stored energy until the evening price peak, in which there is also a large load peak. However, notice how this solution is not optimal — pre-charging would have been useful before the first day as well, and it was not performed; additionally, pre-charging on both days, during medium-price periods, would have benefits in preparation for the evening price peak.

On sunny days, the hybrid model behaves largely how one would expect SCM to behave. Its benefit lies primarily on cloudy days, where PV generation is insufficient and pre-charging (which is not performed by SCM) presents more benefits.



a Sunny days



b Cloudy days

Figure 4.2.13: Examples of the hybrid model performance over two days using perfect forecasts and tri-hourly tariff on different dwellings, in different periods.

The choice of division of data between training and testing was laid out and justified earlier in section 3.1.2 and is subject to constraints posed by the available datasets, but nonetheless has an influence on model performance and results. This is further explored in Appendix B.2, proving that the extent of data available for training the model has a clear impact on performance, at least under one year of training data, since not enough data is available to allow for the use of over one year for training.

Chapter 5

Conclusions and future recommendations

The aim of this work was to build and test a Reinforcement Learning-based Home Energy Management System strategy for a residential battery coupled to a PV generation system, and to test its efficacy when compared to the commonly-used rule-based Self-Consumption Maximisation, which is widely used for this situation. For this purpose, PV generation and load forecasts were also developed, based on two ML models: Random Forests and Artificial Neural Networks.

The forecasting models were first tested on 24-hour ahead point forecasts. For this, RF models had previously been developed by another work, and were here adapted and re-tested along with newly-built ANN models. Care was taken to ensure both models stood on equal footing for comparison. In this step, RF showed better performance than ANN on all metrics.

The next step determined the development of multiple cumulative forecasts with different horizons, namely 1, 3, 6, 12 and 24 hours. This was done as it was considered that this would provide the HEMS with the necessary information, but in condensed form, in order to avoid the curse of dimensionality, i.e., scarcity of sampling information over a large state space. In these cumulative forecasts, ANN outperformed RF, with the notable exception of the 12-hour forecast for PV forecasting. ANNs also generated a higher bias (nMBE) than RF, but this metric was deemed less important than the other two (nRMSE and R^2), as it was expected that the RL model could more easily learn to account for bias, as it generally occurs in the same direction. ANN were therefore chosen to be used in the next step, the development of the RL-based HEMS model.

The HEMS model was initially idealised as a pure RL-based model, but as this proved to yield poor results, a hybrid RL/rule-based model was developed, in which *optimal actions* can be enforced in real time in rule-based fashion by a controller external to the RL agent, with the RL agent making decisions for the remaining situations, in which there is not a clear optimal action to be taken. This addition did allow for improved results, in many cases achieving cost savings when compared to SCM.

Prior to testing the hybrid model, a MILP model was used in order to determine the optimal solution to the battery management of each dwelling for the full test period, thus determining the maximum possible potential for improvement over the SCM solution. This showed that the potential exists for around 5.9% cost savings (using a tri-hourly tariff) and 1.6% (using a bi-hourly tariff) for the median home when compared to the cost using SCM as an energy management strategy.

Six different scenarios were tested for this hybrid model: using no forecasts, ANN forecasts and

perfect forecasts; for both the bi-hourly and tri-hourly energy tariffs. For both tariffs, the hybrid RL/rule-based model is able to fulfil only around 30% of those potential savings when using perfect forecasts, and around 10% using ANN forecasts, for the median user. This translates to only 1.9% cost savings with perfect forecasts and 0.7% with ANN forecasts using the tri-hourly tariff, and even lower values using the bi-hourly tariff. Additionally, using ANN forecasts does not lead to a significant improvement over using no forecasts at all.

For these reasons, the final recommendation of this work is that the commonly-used SCM strategy is adequate to current circumstances in Portugal (current tariffs and energy system composition), and the hybrid RL/rule-based strategy does not show improvements significant enough to justify its employment, considering the added costs in terms of model development and maintenance, data collection and storage, and particularly the computational costs of training a RL model. In fact, it is worth reminding at this point that the calculated costs take into account only energy cost savings, which would play an even smaller relative role in the final energy bill when added to other costs such as power costs.

The main reason why SCM performs well under the current energy system with bi-hourly and tri-hourly tariffs is that it leads to most often only needing to buy energy from the grid during nighttime, when prices are at their lowest.

For a future energy system, which may include a higher renewable penetration, in particular a higher PV penetration, and therefore different variable price structures (such as a lower price in daytime, during peak PV production hours), it may be that SCM will no longer be the most adequate strategy. Future works could test the adequacy of SCM to such a system and whether RL could represent a more appealing option under those circumstances, or whether different rule-based strategies could address the new challenges. That study, however, should take greater care with the load data used — occupant behaviour and load usage would certainly display different patterns under circumstances differing from the current ones, and as such, using data from current load patterns could certainly lead to misleading results.

A future energy system could also include variable real-time energy prices due to variable renewable generation, which could justify the addition of an energy price forecast.

Under the current system, a simpler option is to facilitate each client's analysis of whether a bi-hourly or tri-hourly tariff is more adequate for them, as simply the choice of tariff can produce cost savings comparable to those provided by the hybrid RL/rule-based model for some clients.

Additionally, other rule-based strategies, such as improved versions of SCM, could be tested. In particular, MILP solutions showed some potential for improvement comes from the ability to pre-charge the battery from the grid during low price periods in order to help cover load during days with insufficient PV generation. Therefore, rule-based strategies could be developed with this purpose in mind, along with improved 12-hour and 24-hour cumulative forecasts.

Although this was not tested here, there are likely to exist seasonal fluctuations in SCM's performance: it is well-adjusted for summer conditions, with large PV generation, while winter, with scarcer PV generation, may be the most favourable season for the use of a pre-charging strategy.

Different types of clients could also have different needs, perhaps creating potential for clustering clients according to the shape of their load profile, and developing targeted rule-based strategies according to client type.

Variations of the model developed here could be tested for improvements, such as using different

combinations of cumulative forecasts as input information for the model.

Future additions could also look into using RL for load scheduling and EV-charging scheduling (possibly integrating vehicle-to-home), already performed by some HEMS in the literature with good results, and wherein lies possibly a greater potential for additional cost savings under the current energy systems.

The greatest contribution of this work is perhaps the integration of real PV and load forecasts and HEMS in the same work, thus studying their impact and providing a more realistic and holistic assessment of model performance. This is illustrated by the relevant difference between HEMS performance using perfect forecasts and using ANN forecasts. Additionally, while not tackled by this work specifically, the project in which this work is included will deal with all aspects of building a HEMS, including electronics, databases, security, etc., providing a truly comprehensive study and holistic approach.

It should be reminded once more that the battery capacity scaling in this work was done for standardisation purposes but is not realistic, as it does not match battery capacities available in the market. Future works could study the impact of this simplification, as well as develop more realistic simulations, perhaps even studying the impact of battery capacity on the performance of each energy management strategy.

One last thing that needs to be stressed once more is how, in most cases, the models are not trained with data pertaining to a full year, due to insufficient data. Therefore, the model is not exposed to all seasons during training, meaning it will then be particularly ill-equipped to handle seasons which it has not previously witnessed (and is then tested on). It would therefore be interesting to repeat the simulations performed here using two full years of data for each dwelling.

References

[1] United Nations, "Goal 7 | Department of Economic and Social Affairs," accessed 10 November, 2021.

[2] P. Fox-Penner, "Power after Carbon: Building a Clean, Resilient Grid." Harvard University Press, 2020, pp. 76–81.

[3] E. R. Parracho, "Previsão de consumo de eletricidade para sistemas de autoconsumo solar," M.S. thesis, Faculdade de Ciências da Universidade de Lisboa, Lisbon, Dec. 2021.

[4] "Decreto-Lei n.º 363/2007, Estabelece o regime jurídico aplicável à produção de electricidade por intermédio de unidades de micro-produção," Diário da República nº 211/2007, Nov. 2007.

[5] "Decreto-Lei n.º 153/2014, Cria os regimes jurídicos aplicáveis à produção de eletricidade destinada ao autoconsumo e ao da venda à rede elétrica de serviço público a partir de recursos renováveis, por intermédio de Unidades de Pequena Produção," Diário da República nº 202/2014, Oct. 2014.

[6] S. Rodrigues, F. Faria, N. Cafôfo, X. Chen, and F. Morgado-Dias, "Self-Consumption and Battery Bank PV Systems in the Residential Sector in Portugal," Jun. 2015.

[7] F. Camilo, R. Castro, M. Almeida, and V. Pires, "Economic assessment of residential PV systems with self-consumption and storage in Portugal," *Solar Energy*, vol. 150, pp. 353–362, Jul. 2017.

[8] A. Foles, L. Fialho, and M. Collares-Perreira, "Techno-economic evaluation of the Portuguese PV and energy storage residential applications," *Sustainable Energy Technologies and Assessments*, vol. 39, p. 100 686, Jun. 2020.

[9] IRENA, "Electricity storage and renewables: Costs and markets to 2030," International Renewable Energy Agency, Abu Dhabi, Tech. Rep. 978-92-9260-038-9, Oct. 2017.

[10] BloombergNEF, "A behind the scenes take on lithium-ion battery prices," Mar. 2019.

[11] "Electric Vehicle Outlook 2022," Tech. Rep., Jun. 2022.

[12] C. Villar, D. Neves, and C. Silva, "Solar PV self-consumption: An analysis of influencing indicators in the Portuguese context," *Energy Strategy Reviews*, vol. 18, pp. 224–234, Dec. 2017.

[13] P. A. Flach, "Machine learning: The art and science of algorithms that make sense of data." Cambridge University Press, 2017.

[14] R. S. Sutton, F. Bach, and A. G. Barto, "Reinforcement learning: An introduction," 2nd ed. MIT Press Ltd, 2018.

[15] G. James, "An introduction to statistical learning: With applications in R." Springer, 2021.

[16] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning." MIT Press, 2016.

[17] R. Perez, P. Lauret, M. Perez, M. David, T. Hoff, and S. Kivalov, "Solar resource variability," in *Solar Energy Forecasting and Resource Assessment*. Elsevier, Apr. 2018, ch. 6.

[18] M. Lave, J. Kleissl, and J. Stein, "Quantifying and Simulating Solar-Plant Variability Using Irradiance Data," in *Solar Energy Forecasting and Resource Assessment*. Elsevier, 2013, ch. 7.

[19] J. Antonanzas, N. Osorio, R. Escobar, R. Urraca, F. Martinez-de-Pison, and F. Antonanzas-Torres, "Review of Photovoltaic Power Forecasting," *Solar Energy*, vol. 136, pp. 78–111, 2016.

[20] P. Bacher, H. Madsen, and H. A. Nielsen, "Online short-term solar power forecasting," *Solar Energy*, vol. 83, no. 10, pp. 1772–1783, 2009.

[21] C. F. Coimbra and H. T. Pedro, "Stochastic-Learning Methods," in *Solar Energy Forecasting and Resource Assessment*. Elsevier, 2013, ch. 15.

[22] A. Dolara, F. Grimaccia, S. Leva, M. Mussetta, and E. Ogliari, "A physical hybrid artificial neural network for short term forecasting of PV plant power output," *Energies*, vol. 8, no. 2, pp. 1138–1153, 2015.

[23] W. VanDeventer *et al.*, "Short-term PV power forecasting using hybrid GASVM technique," *Renewable Energy*, vol. 140, pp. 367–379, 2019.

[24] A. Agga, A. Abbou, M. Labbadi, and Y. El Houm, "Short-term self consumption PV plant power production forecasts based on hybrid CNN-LSTM, ConvLSTM models," *Renewable Energy*, vol. 177, pp. 101–112, 2021.

[25] H. Long, Z. Zhang, and Y. Su, "Analysis of daily solar power prediction with data-driven approaches," *Applied Energy*, vol. 126, pp. 29–37, 2014.

[26] A. Nespoli *et al.*, "Day-Ahead Photovoltaic Forecasting: A comparison of the most effective techniques," *Energies*, vol. 12, no. 9, p. 1621, 2019.

[27] M. Kudo, A. Takeuchi, Y. Nozaki, H. Endo, and J. Sumita, "Forecasting electric power generation in a photovoltaic power system for an Energy Network," *Electrical Engineering in Japan*, vol. 167, no. 4, pp. 16–23, 2009.

[28] H. T. Pedro and C. F. Coimbra, "Assessment of forecasting techniques for solar power production with no exogenous inputs," *Solar Energy*, vol. 86, no. 7, pp. 2017–2028, 2012.

[29] A. Mellit, A. Massi Pavan, and V. Lughi, "Short-term forecasting of power production in a large-scale photovoltaic plant," *Solar Energy*, vol. 105, pp. 401–413, 2014.

[30] C. Chen, S. Duan, T. Cai, and B. Liu, "Online 24-H solar power forecasting based on weather type classification using artificial neural network," *Solar Energy*, vol. 85, no. 11, pp. 2856–2870, 2011.

[31] L. Benali, G. Notton, A. Fouilloy, C. Voyant, and R. Dizene, "Solar radiation forecasting using artificial neural network and random forest methods: Application to normal beam, horizontal diffuse and global components," *Renewable Energy*, vol. 132, pp. 871–884, 2019.

[32] M. Rana, I. Koprinska, and V. G. Agelidis, "2D-interval forecasts for solar power production," *Solar Energy*, vol. 122, pp. 191–203, 2015.

[33] M. De Giorgi, M. Malvoni, and P. Congedo, "Comparison of strategies for multi-step ahead photovoltaic power forecasting models based on hybrid group method of data handling networks and least square support vector machine," *Energy*, vol. 107, pp. 360–373, 2016.

[34] Z. Wang *et al.*, "A Review of Load Forecasting of the Distributed Energy System," *IOP Conference Series: Earth and Environmental Science*, vol. 237, p. 042 019, Mar. 2019.

[35] A. V. Kychkin and G. C. Chasparis, "Feature Extraction for Day-ahead Electricity-Load Forecasting in Residential Buildings," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 13 094–13 100, 2020, 21st IFAC World Congress.

[36] T. Czernichow *et al.*, "Short term electrical load forecasting with artificial neural networks," *Engineering Intelligent Systems for Electrical Engineering and Communications*, vol. 4, no. 2, pp. 85–99, 1996.

[37] D. Srinivasan, "Evolving Artificial Neural Networks for short term load forecasting," *Neurocomputing*, vol. 23, no. 1-3, pp. 265–276, 1998.

[38] B.-S. Kwon, R.-J. Park, and K.-B. Song, "Short-Term Load Forecasting Based on Deep Neural Networks Using LSTM Layer," *Journal of Electrical Engineering Technology*, vol. 15, May 2020.

[39] B. Zhao, Y. Liang, X. Gao, and X. Liu, "Short-Term Load Forecasting Based on RBF Neural Network," *Journal of Physics: Conference Series*, vol. 1069, p. 012 091, Aug. 2018.

[40] C. Cai, Y. Tao, T. Zhu, and Z. Deng, "Short-Term Load Forecasting Based on Deep Learning Bidirectional LSTM Neural Network," *Applied Sciences*, vol. 11, no. 17, 2021.

[41] Q. Cai *et al.*, "Short-term load forecasting method based on deep neural network with sample weights," *International Transactions on Electrical Energy Systems*, vol. 30, no. 5, e12340, 2020. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/2050-7038.12340`.

[42] J. Leitão, P. Gil, B. Ribeiro, and A. Cardoso, "A Survey on Home Energy Management," *IEEE Access*, vol. 8, pp. 5699–5722, 2020, Conference Name: IEEE Access.

[43] U. Zafar, S. Bayhan, and A. Sanfilippo, "Home Energy Management System Concepts, Configurations, and Technologies for the Smart Grid," *IEEE Access*, vol. 8, pp. 119 271–119 286, 2020, Conference Name: IEEE Access.

[44] D. Azuatalam, K. Paridari, Y. Ma, M. Förstl, A. C. Chapman, and G. Verbič, "Energy management of small-scale PV-battery systems: A systematic review considering practical implementation, computational requirements, quality of input data and battery degradation," *Renewable and Sustainable Energy Reviews*, vol. 112, pp. 555–570, 2019.

[45] J. Solano, M. C. Brito, and E. Caamaño-Martín, "Impact of fixed charges on the viability of self-consumption photovoltaics," *Energy Policy*, vol. 122, p. 322, Nov. 2018.

[46] M. Beaudin and H. Zareipour, "Home energy management systems: A review of modelling and complexity," *Renewable and Sustainable Energy Reviews*, vol. 45, pp. 318–335, 2015.

[47] H. Wang, K. Meng, Z. Y. Dong, Z. Xu, F. Luo, and K. P. Wong, "Efficient real-time residential energy management through MILP based rolling horizon optimization," in *2015 IEEE Power Energy Society General Meeting*, 2015, pp. 1–6.

[48] V. Rasouli, I. Gonçalves, C. H. Antunes, and Á. Gomes, "A Comparison of MILP and Metaheuristic Approaches for Implementation of a Home Energy Management System under Dynamic Tariffs," in *2019 International Conference on Smart Energy Systems and Technologies (SEST)*, 2019, pp. 1–6.

[49] X. Hou, J. Wang, T. Huang, T. Wang, and P. Wang, "Smart Home Energy Management Optimization Method Considering Energy Storage and Electric Vehicle," *IEEE Access*, vol. 7, pp. 144 010–144 020, 2019.

[50] Z. Wu, X.-P. Zhang, J. Brandt, S.-Y. Zhou, and J.-N. LI, "Three Control Approaches for Optimized Energy Flow With Home Energy Management System," *IEEE Power and Energy Technology Systems Journal*, vol. 2, no. 1, pp. 21–31, 2015.

[51] H. T. Dinh, K.-h. Lee, and D. Kim, "Supervised-learning-based hour-ahead demand response for a behavior-based home energy management system approximating MILP optimization," *Applied Energy*, vol. 321, p. 119 382, 2022.

[52] P. Lissa, C. Deane, M. Schukat, F. Seri, M. Keane, and E. Barrett, "Deep reinforcement learning for home energy management system control," *Energy and AI*, vol. 3, p. 100 043, 2021.

[53] Y. Ji, J. Wang, J. Xu, X. Fang, and H. Zhang, "Real-Time Energy Management of a Microgrid Using Deep Reinforcement Learning," *Energies*, vol. 12, no. 12, 2019.

[54] R. Appino, J. González-Ordiano, R. Mikut, T. Faulwasser, and V. Hagenmeyer, "On the use of probabilistic forecasts in scheduling of renewable energy sources coupled to storages," *Applied Energy*, vol. 210, Oct. 2017.

[55] S. Kapoor and A. Narayanan, "Leakage and the Reproducibility Crisis in ML-based Science," 2022.

[56] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[57] Martín Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," Software available from tensorflow.org, 2015.

[58] F. Chollet *et al.* "Keras." (2015), [Online]. Available: `https://github.com/fchollet/keras`.

[59] "Tarifários de eletricidade e gás natural para particulares," EDP Comercial, accessed 13 July, 2022.

[60] "O que é a opção horária e qual a melhor para mim?" EDP Comercial, accessed 13 July, 2022.

[61] N. Pisaruk, "Mixed Integer Programming: Models and Methods." May 2019.

[62] E. Barbour, "MILP_vs_Find_batt_scheduler," accessed 13 July, 2022.

[63] M. L. Bynum *et al.*, "Pyomo–optimization modeling in python," Third. Springer Science & Business Media, 2021, vol. 67.

[64] W. E. Hart, J.-P. Watson, and D. L. Woodruff, "Pyomo: Modeling and solving mathematical programs in python," *Mathematical Programming Computation*, vol. 3, no. 3, pp. 219–260, 2011.

[65] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2022.

[66] A. Plaat, "Deep Reinforcement Learning." Springer, 2022.

[67] S. Huang and S. Ontañón, "A closer look at invalid action masking in policy gradient algorithms," *CoRR*, vol. abs/2006.14171, 2020. arXiv: `2006.14171`.

[68] "Reinforcement learning tips and tricks," Stable Baselines3 1.6.1a2 documentation, accessed 23 August, 2022.

[69] M. Janner, "Model-based reinforcement learning: Theory and practice," The Berkeley Artificial Intelligence Research Blog, accessed 23 August, 2022.

[70] "DQN," Stable Baselines3 1.6.1a2 documentation, accessed 23 August, 2022.

[71] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017. arXiv: `1707.06347`.

[72] S. Zhang, Z. Ding, and H. Dong, "Deep reinforcement learning: Fundamentals, research and applications." Springer, 2020.

[73] C. M. Bishop, "Pattern Recognition and Machine Learning." Springer Verlag, 2006.

[74] "Proximal policy optimization," Spinning Up documentation, accessed 23 August, 2022.

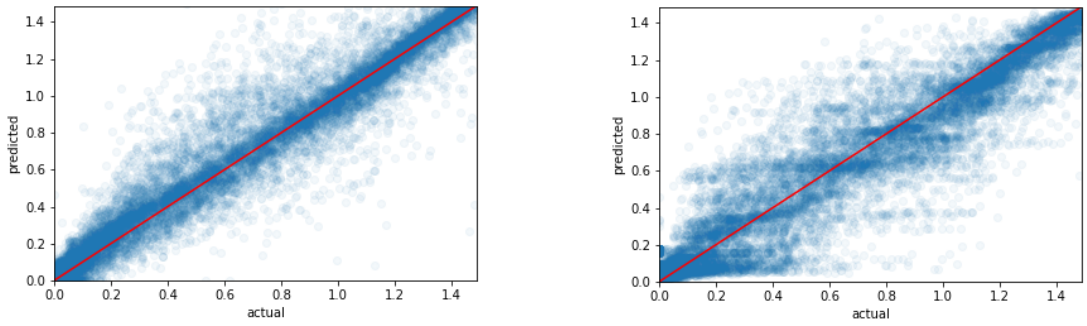
[75] G. Brockman *et al.*, "OpenAI gym," *arXiv preprint arXiv:1606.01540*, 2016.

[76] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.

Appendix A

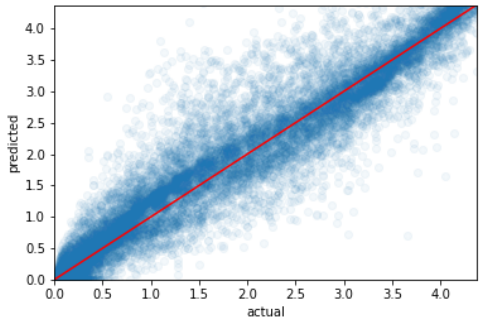
Examples of cumulative forecast model performance

The following pages show PV generation (Figure A.0.1) and load (Figure A.0.2) cumulative forecast model performance for illustrative houses. The superior performance of ANNs is evident in most pictures. Curiously, RF predictions seem to fall into near-discrete values, while this pattern is not observed in real data (nor in RF predictions for point-forecasts). Figure A.0.1g also reveals the source of ANN's poor performance for 12-hour cumulative PV prediction: a number of zero predictions for non-zero values.

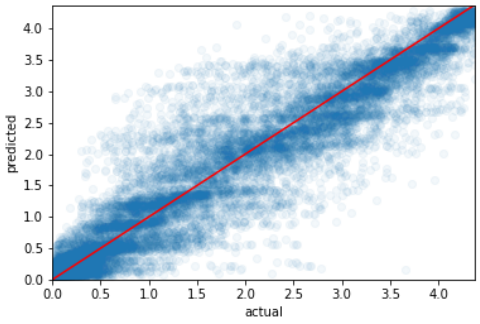


a ANN - 1 hour

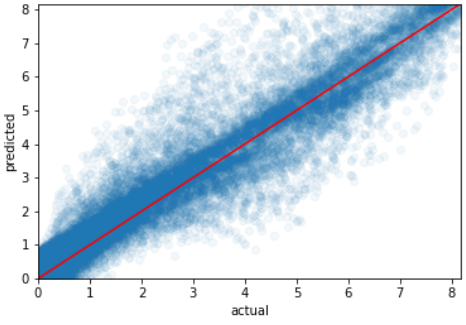
b RF - 1 hour



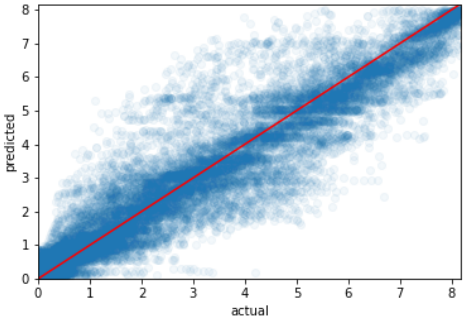
c ANN - 3 hour



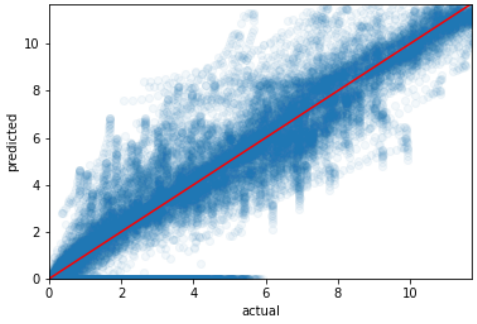
d RF - 3 hour



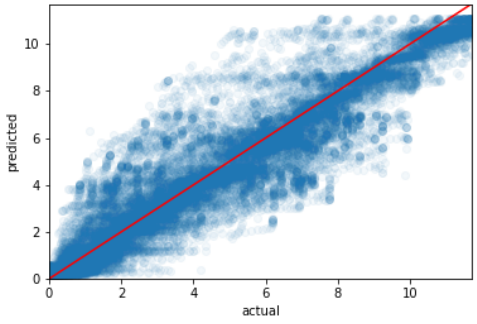
e ANN - 6 hour



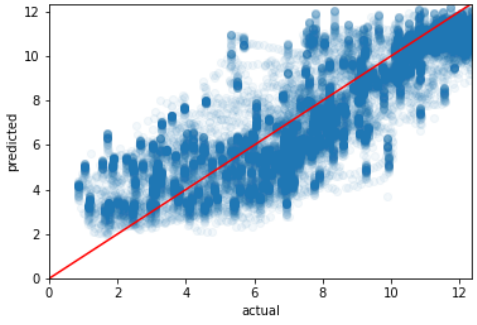
f RF - 6 hour



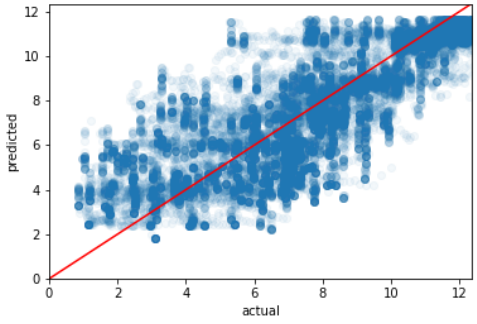
g ANN - 12 hour



h RF - 12 hour

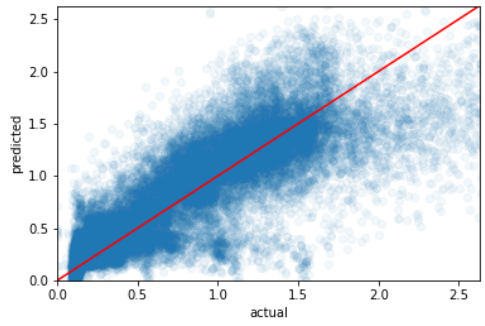


i ANN - 24 hour

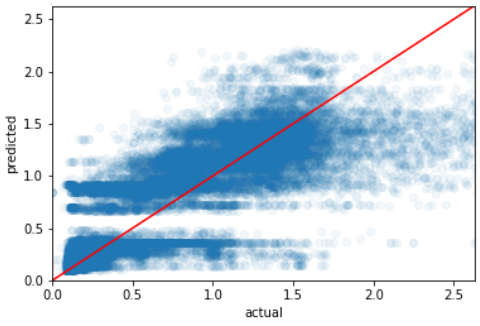


j RF - 24 hour

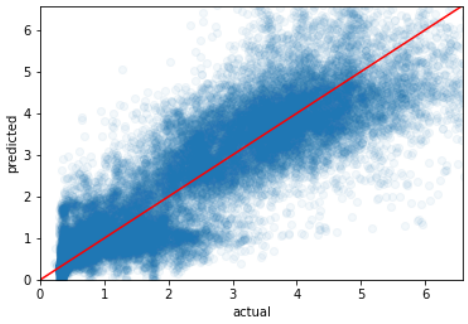
Figure A.0.1: Example of ANN and RF model prediction accuracy on PV cumulative forecast for all considered horizons, for one dwelling.



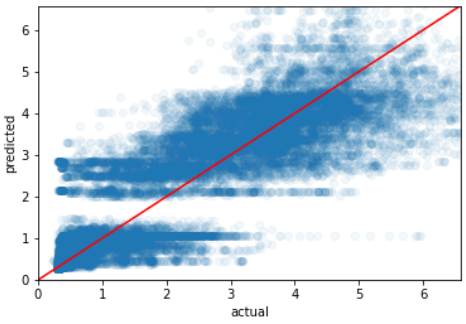
a ANN - 1 hour



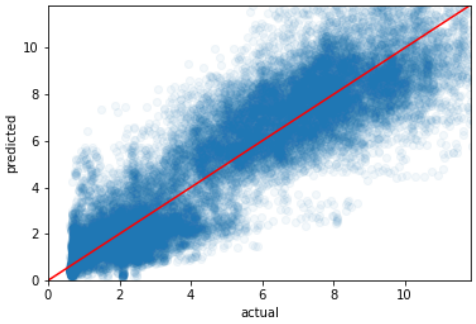
b RF - 1 hour



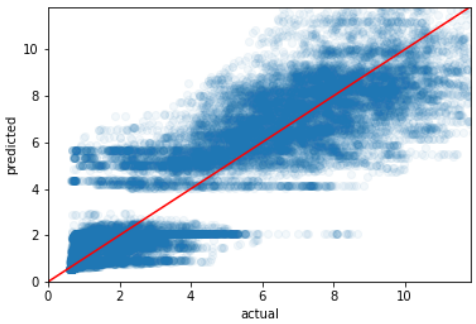
c ANN - 3 hour



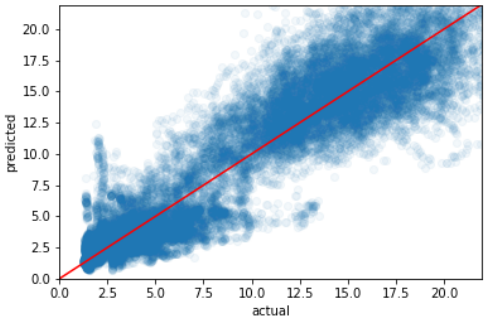
d RF - 3 hour



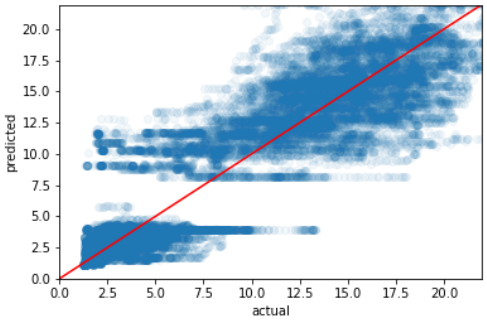
e ANN - 6 hour



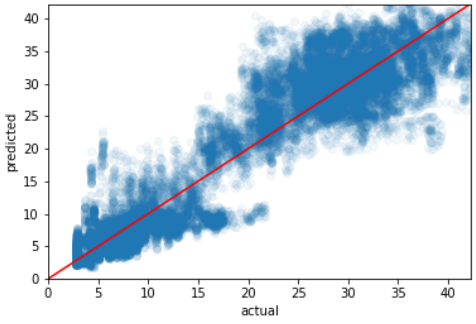
f RF - 6 hour



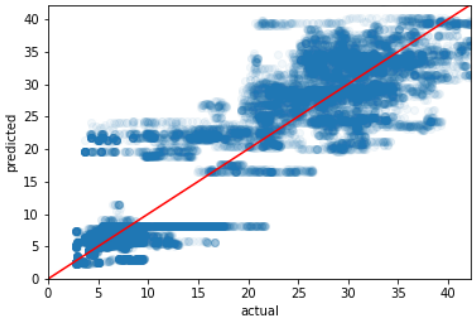
g ANN - 12 hour



h RF - 12 hour



i ANN - 24 hour



j RF - 24 hour

Figure A.0.2: Example of ANN and RF model prediction accuracy on load cumulative forecast for all considered horizons, for one dwelling.

A. EXAMPLES OF CUMULATIVE FORECAST MODEL PERFORMANCE

Appendix B

HEMS - additional experiments

B.1 Using PV forecasts or load forecasts exclusively

In order to expand the point made on page 66 regarding forecast importance (where it is said that load forecasts may be more important than PV forecasts, the latter of which may in fact hold little importance), two additional scenarios were devised: one using only load forecasts (**a**), and the other using only PV forecasts (**b**). These were tested only using perfect forecasts, and the tri-hourly tariff. Results are shown in Figure B.1.1. Both additional scenarios show similar performance, with scenario **a** showing only a slight improvement over scenario **b**, and both are worse than using both forecasts. However, these results are not straightforward. In order to use only one kind of forecast, the additional reward for pre-charging based on 24-hour forecasted values (point 2 on section 3.3.3.1) had to be removed, removing part of the incentive for pre-charging at low prices. Pre-charging, as was previously mentioned, is largely where the potential for improvement over SCM lies. Consequently, having to remove this is likely to negatively affect model performance. That said, this experiment cannot lead to meaningful conclusions.

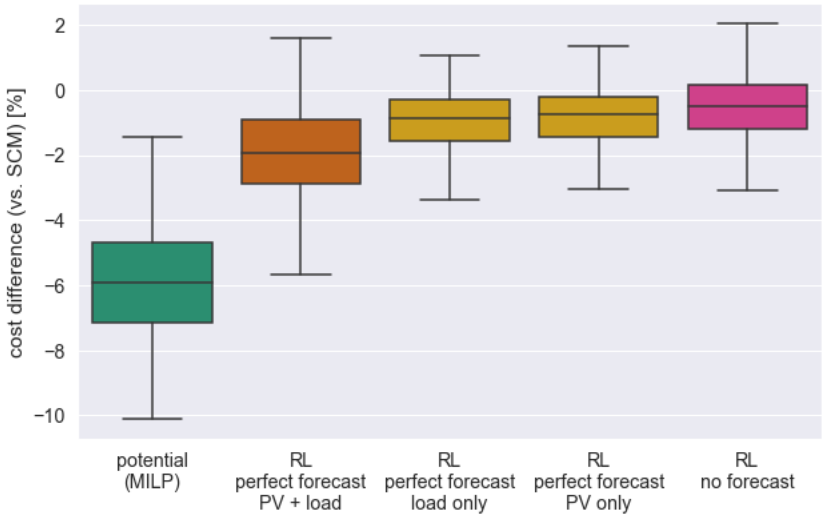


Figure B.1.1: Cost difference [%] of additional scenarios (in yellow) when compared to SCM, for the tri-hourly tariff. Previous results are reproduced here for reference.

B.2 Training with fewer data points

A further experiment was made in order to test the impact of the amount of available data points for training on model performance. This experiment used perfect forecasts and the tri-hourly tariff, and consisted of using differing fractions of the available training data for training the model ($\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$). The piece of training data used is the latter (most recent) corresponding fraction of the full available training data. This choice was made because it mimics the real world, namely a scenario where one would seek to begin HEMS operation with a previous data collection period that is as short as possible. Test data is the same for all scenarios.

Results are shown in Figures B.2.1and B.2.2, where it is clear that using less data leads to a worse performance. Additionally, placing all instances together, a correlation of 0.32 was found between available valid data points for training and fulfilled potential. This can be observed in Figure B.2.3, where the x-axis values were limited for ease of viewing, omitting a few instances of very poor performance, which occurred almost exclusively for the scenarios using fewer data points.

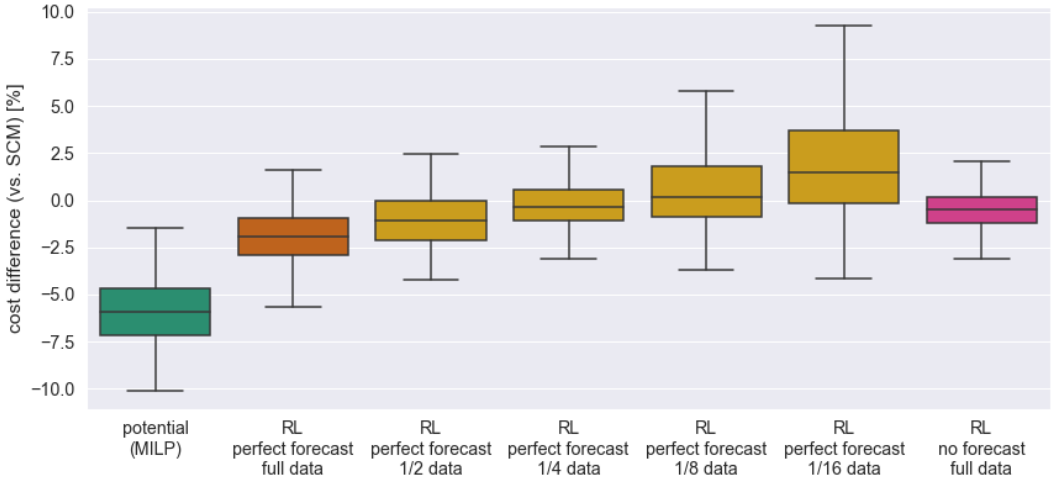


Figure B.2.1: Cost difference [%] of additional scenarios (in yellow) when compared to SCM, for the tri-hourly tariff. Previous results are reproduced here for reference.

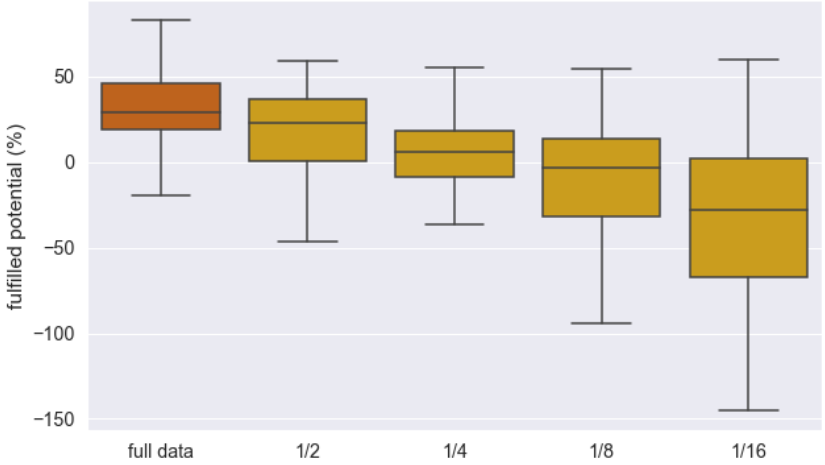


Figure B.2.2: Fulfilled potential [%] of additional scenarios (in yellow) when compared to SCM, for the tri-hourly tariff. Previous results are reproduced here for reference.

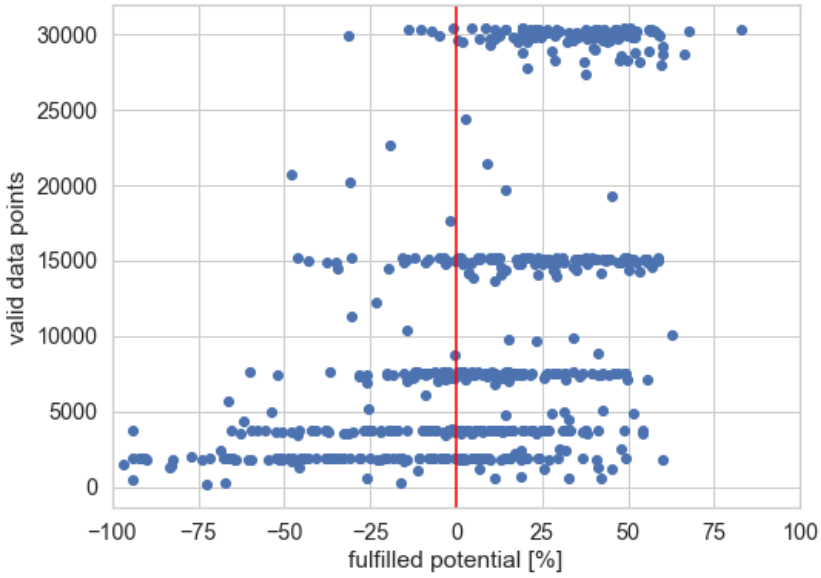


Figure B.2.3: Fulfilled potential [%] of additional scenarios when compared to SCM, for the tri-hourly tariff, vs. number of valid data points available for training.

Since for most houses the HEMS is trained on less than one year's worth of data, it is unclear whether training with data beyond one full year would significantly improve performance. However, with the available data, it is clear that increasing the amount of training data improves HEMS performance, likely due to the model experiencing all seasons during training.