PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

REINFORCEMENT LEARNING WITH NEURAL NETWORK-BASED VALUE FUNCTION APPROXIMATION TO MANAGE FLEXIBLE ELECTRIC LOADS AND DISTRIBUTED ENERGY RESOURCES

FELIPE ANDRÉS HAASE VARGAS

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Advisor:
ÁLVARO HUGO LORCA GÁLVEZ

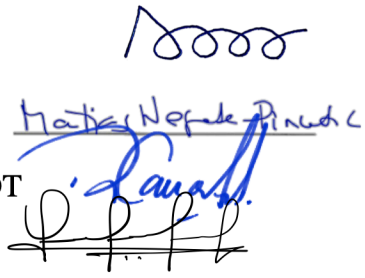
Santiago de Chile, June 2021

PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

REINFORCEMENT LEARNING WITH NEURAL NETWORK-BASED VALUE FUNCTION APPROXIMATION TO MANAGE FLEXIBLE ELECTRIC LOADS AND DISTRIBUTED ENERGY RESOURCES

FELIPE ANDRÉS HAASE VARGAS

Members of the Committee:
ÁLVARO HUGO LORCA GÁLVEZ
MATÍAS ALEJANDRO NEGRETE PINCETIC
RODRIGO ARNALDO CARRASCO SCHMIDT
CARLOS ALBERTO BONILLA MELENDEZ



Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Santiago de Chile, June 2021

Gratefully to my parents, sister and friends.

AGRADECIMIENTOS

Nadie dijo que realizar una investigación con un estallido social y una pandemia sería fácil, pero todo lo que comienza debe terminar y los resultados son satisfactorios.

Primero que nada agradecer a mi familia: papá, mamá y hermana, que han dado todo siempre por mí. Gracias al esfuerzo diario que han dado durante mi vida para formarme como persona y profesional. A mis familias que hicieron posible mi estadía en Santiago durante mi vida universitaria: Sandra, Mario, Roxana, Jorge, Marcelo, Roxana, Andrés y Yenny y a mi segunda madre Violeta, que me recibió durante los últimos años. A mi tío Hugo un abrazo al cielo, por enseñarme que las cosas siempre se pueden hacer mejor.

A mi profesor guía Álvaro, por todo el apoyo durante lo meses más difíciles, donde la motivación a veces se perdía y especialmente por haberme motivado desde mi primera investigación de pregrado, a seguir investigando ahora en el Magister. Al equipo de OCM 2019 con quienes tuve el gran agrado de compartir oficina por medio semestre.

A mis amigos que hice en la universidad y que me han acompañado en este bello camino. Especialmente a La Parrukería: Flo, Roa y Checho, tremendos amigos con quienes compartimos incluso seminarios de investigación para motivarnos con nuestras tesis en los meses de encierro de esta pandemia. A Tamy, que junto a Flo fueron un gran apoyo para mi en los últimos días escribiendo la tesis, con largas tardes de videollamada de escritura de tesis. A *Oh Canada*: Tisco, Javier, Seba y Barri, que desde la carretera austral que no nos separamos, con los que puedo contar siempre tanto en lo personal como académico.

A genomawork y el gran equipo humano que la compone, la gran startup de la que estoy feliz de ser parte y me ha apoyado tanto en este proceso. A mis amigos con los que fui parte de Educalabs/Neolítica, Hielo, Benja, Nacho y Emilio. A todos quienes han sido parte de mi vida en estos últimos dos años. Gracias.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Motivated by the challenge of power generation through renewable resources and the imminent massive adoption of electric cars in homes, this paper proposes an online optimization scheme based on Reinforced Learning for decisions associated with two distributed energy resources at homes: a solar panel with a battery and an electric car capable of delivering energy to the home. The model states are represented by the energy in both the battery and the car and other exogenous factors such as the home's energy consumption, temperature, and humidity. As there is an infinite space of states, discretization is performed on the energy stored in the battery and the car. The model is approximated by Value Function Approximation, using a Neural Network as an approximation, which serves as a regression function. The neural network is trained with state-action vectors and expected values of the future cost of the actions taken. Two experiments are carried out to test the model's effectiveness: an adjustment of the hyperparameters of the neural network in search of the model that best approximates the data; and a simulation of decisions in a home with real data. The results obtained from making day-to-day decisions are compared with three simpler policies designed based on the data's nature. The designed model obtains an average 13% of cost advantage over a year as compared to the benchmark policies.

Keywords: machine learning, reinforcement learning, energy, consumption forecasting, neural networks, value function approximation, VFA

RESUMEN

Ante los desafíos que conllevan tanto la implementación de generación de energía mediante recursos renovables, como el inminente aumento de los autos eléctricos en los hogares en el corto plazo, se propone un modelo de optimización mediante un esquema de Aprendizaje Reforzado para la decisión de acciones que tomar sobre dos recursos energéticos distribuídos: un panel solar con batería y un auto eléctrico con capacidad de entregar energía al hogar. Los estados del modelo son representados por la energía tanto en la batería como en el auto, y otros factores exógenos como el consumo energético del momento del hogar, temperatura y humedad. Al existir un espacio infinito de estados, se realiza una discretización sobre los estados de batería de ambos artefactos. El modelo es aproximado mediante Aproximación de Función de Valor, utilizando como aproximador una Red Neuronal, que sirve como función de regresión. La red neuronal es entrenada con vectores de estado-acción (X) y valores esperados del costo futuro de la acción tomada (Y). Se realizan dos experimentos para probar la efectividad del modelo: un ajuste de los hiperparámetros de la red neuronal, en busca del modelo que mejor logre aproximar los datos; y una simulación de decisiones en un hogar con datos reales otorgados por Pecan Street. Se comparan los resultados obtenidos de tomar decisiones día a día con tres políticas diseñadas a partir de la naturaleza de los datos. El modelo diseñado obtiene un 13% de ventaja por sobre la mejor política diseñada, en el promedio anual.

Keywords: machine learning, reinforcement learning, energy, consumption forecasting, neural networks, value function approximation, VFA

1. INTRODUCTION

Climate change is one of the most important challenges of the modern world. The increase of carbon dioxide concentration in the atmosphere has been an alert for most countries to move from carbon-intensive generation to renewable energy sources. For example, the Chilean government has recently announced that by 2040, the generation side of the Chilean electric power system will have completely removed coal units. In fact, Chile has already been transforming the electrical generation to a cleaner one, achieving by 2018 that 45.5% of the electrical generation in Chile came from renewable sources. On this line, it is expected by 2030 that 80% of electrical generation will come from renewable sources (de Chile, 2018; Simsek et al., 2020). Considering this, it is also expected that renewable energy sources will also be used in households, complementing the traditional energy supply from large and far-away generators. For example, in Chile, from 2020, there is a support grant for solar panel acquisition for domestic houses to help families reduce their energy cost at home and increase the sustainability of the power supply (Chile, 2020). This will generate new challenges in managing these resources. Variable renewable energy generates uncertainty about the electric generation, which depends on exogenous conditions such as solar radiation, temperature, and humidity. On the other hand, electric power requirements increase due to electric cars' incorporation and their need to quickly charge significant amounts of energy. There is a high amount of energy needed for a car to be driven and last for enough travel time.

Further, battery technologies are improving fast, becoming cheaper and lasting longer (Rahman, Oni, Gemechu, & Kumar, 2020). This makes them more accessible to store energy in houses, especially when it comes from renewable sources.

The generation uncertainty of variable renewable energy, added to the increasing number of electric cars and many other high-powered devices, will considerably increase electric consumption and the power required for connecting many high-powered devices at the same time. One of the city's challenges is to distribute the electrical resources intelligently.

This is done by controlling and scheduling the electrical loads efficiently, improving the system's behavior on the demand side.

The scientific community has not fully studied how to manage distributed energy resources, dealing with energy consumption uncertainty to make a more efficient use of energy inside homes. However, various previous works provide an important basis for our proposed work. First, some authors have studied reinforcement learning and other machine learning techniques in demand response contexts. In (R. Lu & Hong, 2019) a reinforcement learning scheme is proposed, with a neural network approach for predicting future prices and energy demands. In particular, the model proposed calculates the optimal incentive rate for a house to respond in a demand response incentive-based scheme. A multi-agent approach of reinforcement learning is proposed in (Kazmi, Suykens, Balint, & Driesen, 2019), but it focuses on control of thermostatically controlled and not energy storage or home appliances usage. (Zhang, Li, Sun, & O'Neill, 2016) use a mix between classic optimization and machine learning for learning home heating, ventilation, and air conditioning behaviors, so the model can control them to test how it fits in a demand response environment.

There is also work on Reinforcement Learning with deep learning for deciding on multiple home appliances (R. Lu, Hong, & Yu, 2019). The difference is that deep learning is employed for energy price forecasting and not for the Reinforcement Learning (Q-Learning in this case) function approximation, which is the case of our present work.

More reinforcement learning methods can also be seen in the review paper (Vázquez-Canteli & Nagy, 2019) and references therein. This review describes different reinforcement learning approaches to manage thermal loads, energy loads, and batteries, with different types of approximations such as Q-Learning.

Also, previous work from Google's DeepMind team employed machine learning techniques that resulted in a 40% reduction in energy consumption from cooling data centers (Gao, 2014). Further, Google's recent publication on how they are moving toward 24x7

carbon-free energy in every data center of the world specifically provides another piece of great inspiration for our proposed work, as the implementation is also based on reinforcement learning using deep learning as support for consumption forecasts (Google, 2018).

Further, robust optimization for power system operations was presented in (Mena, Escobar, Lorca, Negrete-Pincetic, & Olivares, 2019), where new optimization and stochastic modeling tools were developed to allow smarter real-time decision-making schemes for power systems with a massive integration of wind and solar power, supporting the hypothesis on how exogenous factors affect energy consumption prediction. One configuration of an energy community is introduced in (X. Lu et al., 2020) where they present a robust optimization model for demand response that also includes the concept of electric vehicles participating in the grid (also known as vehicle-to-grid, or V2G, schemes).

The vehicle to grid problem is also treated in this paper, and it also has been widely studied, aiming to minimize costs for the electrical vehicle users. Nevertheless, in recent studies, references (He, Venkatesh, & Guan, 2012; Di Giorgio, Liberati, & Pietrabissa, 2013) do not use real-time information or focus only on the vehicle-to-grid decisions without taking decisions in other electrical artifacts.

Moreover, information available these days opens a world for solving difficult problems. This line (Ning & You, 2019) shows how data, deep learning, and other data-driven mathematical programming frameworks are growing fast to deal with uncertainty on stochastic problems. Problems also related to vehicle-to-grid prediction are developed (Ebrahimi & Rastegar, 2020). The authors propose a clustering method from electric vehicle charging behaviors by selecting a subset of the electric vehicles as representatives. Then they estimate a total load of a charging station based on the representative charging profiles. The research does not apply any learning methods for predictions. Instead, they apply data mining. On the other hand (Avendano, Ruyssinck, Vandekerckhove,

Van Hoecke, & Deschrijver, 2018) applies different data-driven machine learning methods, being one of them a reinforcement learning scheme with neural networks as an approximation. The paper shows that machine learning algorithms for control can be as good as rule-based policies, but the machine learning-supported methods are more scalable and require less knowledge.

Real-world data from the same data provider as our work (Pecan Street) was used to solve a similar problem in (Chung, Maharjan, Zhang, & Eliassen, 2020). They apply game theory to solve a collaborative framework where houses work together to minimize electricity dynamic prices, as they are affected by the house consumption behaviors. However, the article does not include any control over renewable energy sources or electric cars and house batteries as it schedules home appliance usage.

As mentioned above, real data is used in this research. The source, Pecan Street, contributes to academic research by making data publicly available for this purpose. The data from this entity has been widely studied in different ways, many of them with machine learning applications. Reference (Afzalan & Jazizadeh, 2020) proposes a method for inference of time-of-use on flexible loads, specifically an electric car. The method is proposed for getting a more detailed cause of some specific energy use patterns. A similar problem is approached but with a different method in (Wang, Du, Ye, & Zhao, 2020), They present a deep generative model for the same purpose of electric vehicle charging profiles.

The importance of solar generation data recording or quantification is raised in (Brown, Abate, & Rogers, 2020). For that purpose they propose a method for quantifying solar energy generation in censored smart sensors, in which there is no detail of solar energy generation records, contributing in generating more available data related to renewable sources. Also, reference (Henri & Lu, 2019) comparing four classic machine learning methods (neural network, support vector machine, logistic regression, and random forest) to predict and control photovoltaic panels and energy storage. They use 1-min interval data records concluding that machine learning can improve results over mode-based controllers and reduce computing efforts by training models on the cloud.

In (Mocanu et al., 2018) an on-line optimization using deep reinforcement learning and deep policy gradient is applied for making energy usage more efficient. In this particular case, results showed that deep policy gradient worked better for the data they used. Some of the methods, such as hyperparameter tuning (network configuration study) and approximation of the reinforcement learning scheme, are also applied in our work.

Motivated by the high potential of managing renewable energy sources and the availability of good quality data records of energy consumption, the present work proposes an online optimization model and applies machine learning regression to study the capability to use historical data to predict and optimize the use of electrical energy sources. The model is based on a reinforcement learning scheme, approximated using a Value Function Approximation (VFA). To deal with the data's nonlinearity, a Neural Network model is used to work as a regression function in the VFA scheme. The model is developed in the Python programming language, supported by the Python packages Pandas for reading historical data and Scikit-learn for machine learning.

1.1. Research questions and hypotheses

As energy consumption data is becoming more exploited (Avendano et al., 2018; Ebrahimi & Rastegar, 2020; Afzalan & Jazizadeh, 2020; Brown et al., 2020) by researchers, many methods have been tested. Energy consumption behavior has always been variable, but sometimes consumption data has patterns that are potentially discoverable with the help of algorithms. That made us elaborate out first research question as follows.

RQ1. Can energy consumption data in a household be generalized in order to approximate future consumption?

Literature shows that data-driven solutions(Vázquez-Canteli & Nagy, 2019; Kazmi et al., 2019; Liu et al., 2020; R. Lu et al., 2019) are feasible methods for dealing with energy optimization. Therefore we pose an hypothesis and a research question

H1: A data-driven reinforcement learning model can be approximated using neural networks, in order to optimize energy consumption and storage.

RQ2. Do neural network hyperparameters have an impact on the performance of the value function approximation?

Further questions are also asked in order to understand better how the predicting model will work and also how data behaves over time.

As the years present different seasons, each of them having very different solar energy and consumption profiles, two more questions are stated.

RQ3. Will the model work good despite of the epoch of the year?

RQ4. Will there be any difference between hours of the day?

1.2. Research contribution and structure

The research's main objective is the development and implementation of a flexible energy storage optimization model for managing solar and electric vehicle batteries in any building situation. In particular, the model has the ability to learn from data and generalize it for predicting future scenarios with good results in high solar energy generation. As a summary, this thesis:

- Proposes a reinforcement learning scheme for predicting the best battery energy usage policies.
- Applies neural networks for approximating the value function of the reinforcement learning model.
- Presents a case study with hyper-parameter tuning for improving neural network performance under the proposed policy and compares the model against other energy usage policies designed to take advantage of the nature of the test data.

The rest of the document is organized as follows. Chapter 2 gives a brief explanation of the methods applied in the proposed model. Chapter 3 states the problem definition and model, defining stages, states, costs, and actions and its nomenclature. Chapter 4 compares the model's configurations, tests the final model against three policies, and discusses the results. Finally, chapter 5 provides concluding remarks.

2. BACKGROUND

2.1. Dynamic Programming and Reinforcement Learning

Dynamic programming is a methodology for solving problems that are not static, that means they depend on stages and probabilities. Reinforcement learning is an Artificial Intelligence algorithm proposed for solving dynamic programming problems. An agent (the decision-maker) takes decisions based on its interaction with the environment, learning from data.

RL can be formalized as an MDP (Markov Decision Process) (Kim & Lim, 2018). States represent the current status of the environment and therefore imply the possible actions to be taken. Each action generates a reward (or cost) depending on the current state, defined by a cost function. The action taken will also define the transition to the next stage, which is also affected, in some cases, by a probability transition function.

In some cases, reinforcement learning schemes can be complicated to solve or take too much time. As the complexity depends on the number of states, actions, and stages, if any

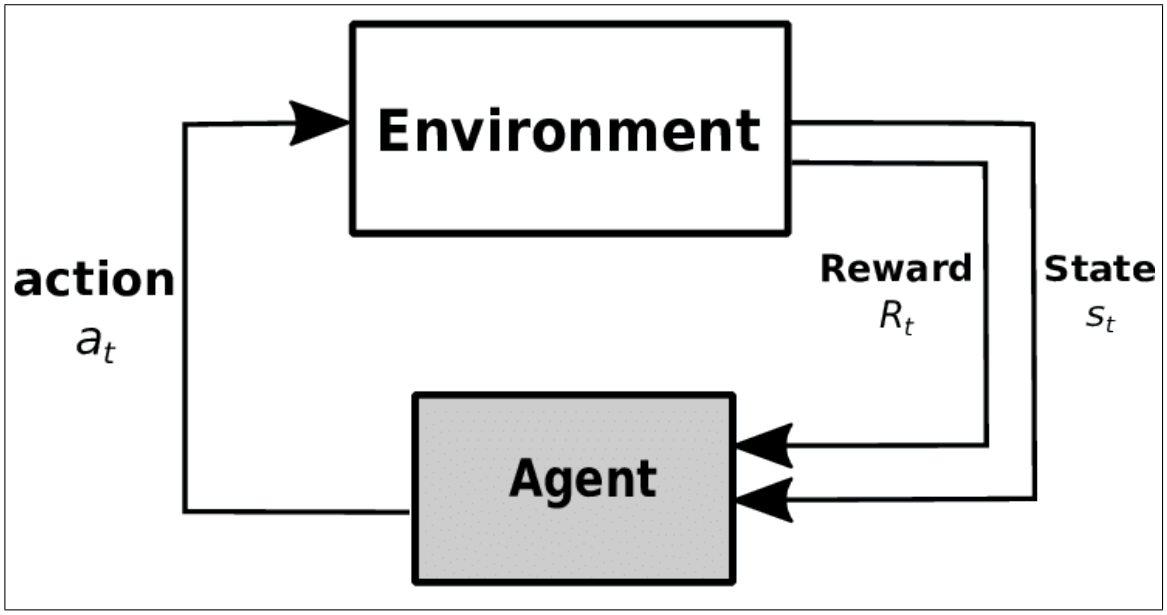


Figure 2.1. Reinforcement learning scheme

of them are too big, the problem becomes too big to be solved in a reasonable time with current computing power. This problem is called the curse of dimensionality.

2.1.1. Value Function Approximation

There are various methods to solve problems with a huge space of states, actions, or stages. One of them is called Value Function Approximation, which consists of estimating the long-term reward/cost of a given state (Sammut & Webb, 2017). To make that possible, the model uses a regression function trained with historical data to make the model converge to a good estimation.

The Value Function of a pair of state/action is commonly represented as follows.

$$V(s) = \min_{a} r(s, a) + V(s(a))$$

Where r represents the reward or cost of the current state, given an action choice and how it affects the environment, then the optimal value of a state s and taking action a is given by the sum of the reward of the current state and the best value of the function for the available space of actions, and the new state given by the chosen action.

For a large space of state and actions, this can be hard to be calculated, so an estimation should be done. In this work, the approximation is made by training a Value Function Approximator, establishing a limit for recursion, and using historical data for calculating these values.

2.1.2. Neural Networks in Reinforcement Learning schemes

Neural networks aim to simulate real human brains. They are composed of neurons that are interconnected, simulating neural synapses. Neural networks have layers of neurons of different sizes, where the neurons of a layer are usually connected to the neurons of the next layer. When every neuron of a layer is connected to all of the next layer's

neurons, they are called fully connected layers. Neurons receive input and give an output. Each connection between a neuron and another has a weight that will be trained and fitted until the$f$ neural network makes the best prediction of regression or classification. The input of the first layer (input layer) is defined by the model (e.g., temperature, energy levels, other neural network outputs, etc.). In contrast, each neuron's input of the next layers is the sum of the output of all neurons of the layer before times the weight of each connection.

Neural network training consists of giving training sample inputs to the neural network, looking for the model to adjust to all training sample outputs as closely as possible. These models can be used for regression or classification. As our work is based on regression, then for training, the neural network on each iteration calculates an output cost that is compared to the expected cost, based on a simulation. A loss function is applied to measure the loss of that iteration. For regression problems, common loss functions are the Mean Squared Error or the Absolute Mean Error. The magnitude of the loss will determine, among other parameters, how the neural network will adjust. For recalculating the parameters (weights) of the neural network, the backpropagation algorithm is applied (Hecht-Nielsen, 1992a; Werbos, 1990). More iterations of calculating loss with training samples are made until any convergence criteria.

Neural networks are widely studied and applied in many fields being energy one of them (Liu et al., 2020; Chung et al., 2020). Recent studies have focused on how deep learning can contribute to Reinforcement Learning topics, being one of the best examples of the Deep Q-Learning algorithm in (Mnih et al., 2015).

3. METHODOLOGY

3.1. Problem definition

The study's problem consists of a domestic household with smart sensors that measure the real-time energy consumption of the different home appliances and stores the data for further analysis. This house also has a photovoltaic panel for capturing solar energy, a house battery for solar energy storage, and an electric vehicle. In this case, it is supposed that the electric vehicle can give some of its energy to the house, which means serving as a battery, so the house can use its energy to reduce grid consumption, therefore reducing the money spent. Both solar energy, car charge consumption, and car discharge energy are also recorded with smart sensors. Houses like these also have flexible loads that can be controlled to minimize energy consumption from the grid and take advantage of renewable sources' energy storage.

With the current state of the art of monitoring electrical loads and also the capability of controlling some electrical devices, houses will soon have their own energy sources, more electric vehicles will also be able to give energy to the home, and energy will need to be used in a more smart way. The following model looks for controlling this kind of scenario.

3.2. Model

The scenario we approached is one in which there is a home with smart sensors combined with Internet-of-Things technology, capable of controlling how to use the distributed energy resources. In this context, we propose an online optimization model for deciding how to use distributed energy resources in a smart home context efficiently.

The model consists of a reinforcement learning scheme, using Value Function Approximation to calculate the cost of each pair of state and action chosen. The value function is

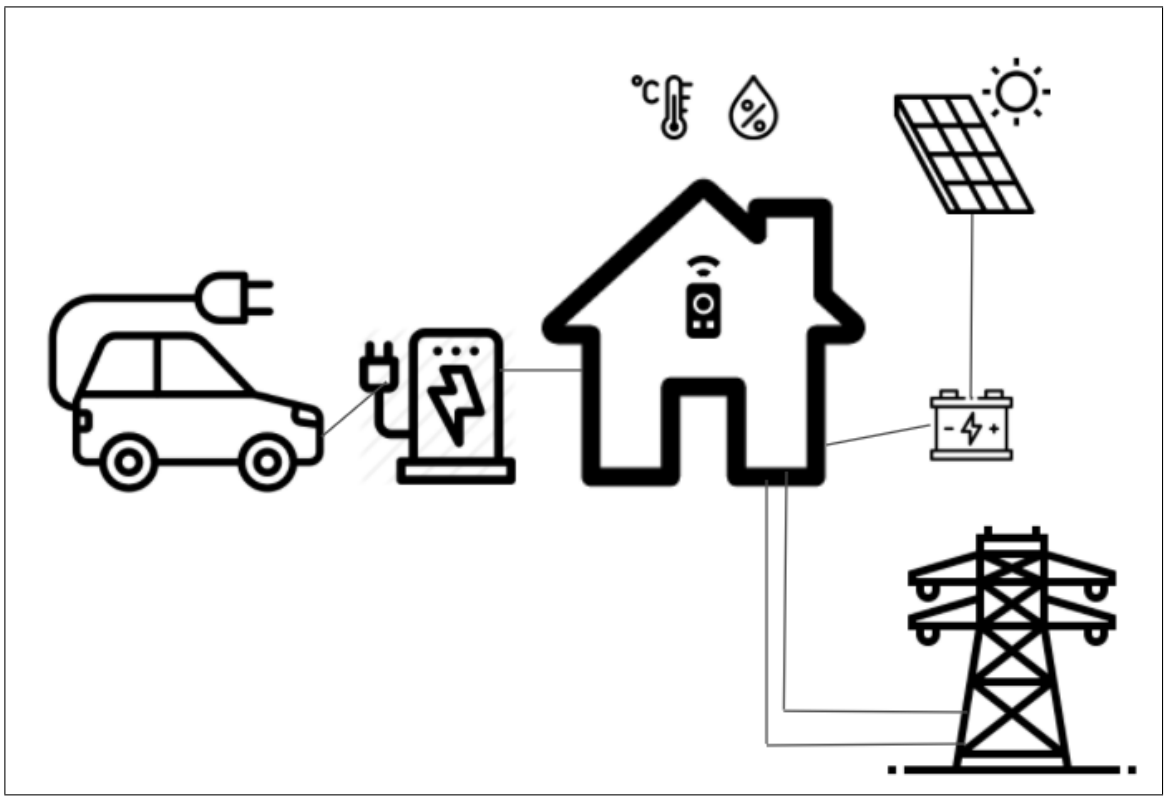


Figure 3.1. Scenario representation

approximated using historical data with a Neural Network model. The function approximation represents the future cost of energy usage, given a current state, and choosing one of the available actions.

The model can schedule the electricity usage from both the battery and the car by considering exogenous factors, such as weather conditions (temperature and humidity) and consumption history. The regression model will predict how much the action taken will cost in the future, so it optimizes over the actions space, looking for the minimum cost impact among all decisions.

3.3. Stages

The stages of the problem represent each period of time, hourly. The stages start from 1 and end at T, where T is the number of hours of the horizon of training or prediction.

$$t \in \{1, 2, 3, ...T\}$$

3.4. Parameters

Some parameters are used for giving context to the model. Energy cost, storage capacity in both car and battery, electrical consumption per kilometer, among others, are some of the parameters that can vary depending on the characteristics of the hardware used, such as batteries and electric cars. The parameters used in the model are listed as follows.

- C : Energy cost (constant) from grid ($/kWh).
- D: Car energy price, when charging car outside the home ($/kWh).
- E: Energy price (constant) for selling to the grid ($/kWh).
- χ: Value of unsatisfied energy demand ($/kWh).
- M : House battery maximum storage capacity (kWh)
- N : Car battery maximum storage capacity (kWh)
- W_{hc} : House battery maximum charge power (kW)
- W_{hd} : House battery maximum discharge power (kW)
- W_c : Car battery maximum charge power (kW)
- W_d : Car battery maximum discharge power (kW)
- W_a : Maximum power from the grid (kW)
- κ : Car average distance driven in an hour (km)
- ψ : Car average energy consumption by km (kWh/km)

The parameters listed above are grouped in four groups: costs of electricity (C, D, χ), energy storage capacity (M, N), power (W_{hc}, W_{hd}, W^c, W^d) and average measurements for cars (κ, ψ).

The value of D is the price of charging the car using an external charger, commonly on gas stations or public electric vehicle chargers on parking spots. The cost of unsatisfied

energy demand means that for any state if there is energy demand greater than W_a, that is the maximum power for a stage. The cost of that state will be increased by the unsatisfied demand penalty χ.

Finally, two parameters represent the average usage of an electric vehicle. The first one is ψ, which is the average kilometers driven by a car daily, based yearly. The second one, κ, represents the average energy consumption per kilometer by a car. Both are used together to estimate the consumption by a car when it is not at home, assuming a uniform distribution of the daily consumption to achieve the yearly average consumption.

3.5. States

- e_t^{NC} : energy consumption on stage t
- e_t^{AC} : air conditioning usage on stage t
- T_t : outside temperature on stage t
- H_t : relative humidity on stage t
- ρ_t : PV energy generated on stage t
- γ_t : car is connected or not on stage t
- ν_t : car's battery energy on stage t
- μ_t : house's battery energy on stage t
- h_t : hour of the day, at stage t

As said in section 2, the state represents the system's current condition at any given stage. Those conditions define the transition to the next stage and the cost for the current stage.

Energy used on period t, separated into two: total nonflexible consumption and the energy consumption (e_t^c) of an air conditioning unit (e_t^a). Exogenous factors are also used for the model. These are temperature (T_t) and relative humidity (H_T), both measured outside the home.

ρ_t stands for the photovoltaic energy captured by the solar panel on stage t. Finally, two state variables represent the stored energy status in both the car (ν_t) and the house (μ_t). γ_t, on the other hand, states if the car is available at the house, that means this vehicle is connected to the house charger so that the house can use its energy or the battery charged.

3.6. Actions

- b_t: action over house battery
- c_t: action over car battery

There is a set of possible actions $x_t(s) = \{b_t, c_t\}$, each one for the house battery and the car battery respectively.

For each stage, the house battery energy (b decision) has three kinds of decisions: charging with the current state solar energy, using energy for the house, or selling some energy to the grid. These decisions depend on the state of the energy level of the battery.

On the other hand, car battery energy is managed similarly, but the charging depends on the electric vehicle charger's power capacity. In summary, the car actions are charge, discharge (give energy to the house, and do nothing (idle connected car or car disconnected).

Unlike (Kim & Lim, 2018), the actions space is bigger, having more than one level of charging or discharging magnitudes. Nevertheless, when looking at more combinations of states, the problem becomes more complex, and one should make approximations. In (Vázquez-Canteli & Nagy, 2019), most of the solutions use Q-Learning or offline simulations. Our proposed method also differs as it employs a custom Value Function Approximation, using Neural Networks instead of Q-Learning.

3.7. Cost functions

As said in section 5.3, the state variable γ_t represents if the car is at home or not. For simplicity, every time the car is not at home ($\gamma_t = 0$), it is assumed that the energy spent equals the average distance driven by a car per hour (κ) times the average energy consumption per kilometer (ψ). This expression is described as q_t^c, in which q is the cost function, c means it is the cost of the car, and t is the stage.

For the following equations, γ represents that the car is connected or not to the power outlet, represented as a binary variable (1 for connected, 0 otherwise). Equation 3.1 represents the transition of state for variable ν, which depends on whether the car is connected or not. If it is, then nothing happens as 1-γ will be 0. If it is not, then the next state's energy of the car will be as described. For simplicity, it is assumed that the driver will make sure that it charges enough energy to the car to come back home at 0 energy.

Finally, equation 3.2 shows that at every stage that the energy consumption for charging the car outside the home is greater than the car energy, there is a cost produced equivalent to the cost of charging the car outside, times the over demand of energy.

$$\nu_{t+1} = \begin{cases} \nu_t - (1-\gamma_t)\kappa\psi \quad if \quad \nu_t \geq \kappa\psi \\ \\ 0 \qquad\qquad\qquad\qquad ioc \end{cases} \tag{3.1}$$

$$q_t^{ce}(s_t) = \begin{cases} D(\gamma_t\kappa\psi - \nu_t) \quad if \quad \nu_t < (1-\gamma_t)\kappa\psi \\ \\ 0 \qquad\qquad\qquad\qquad ioc \end{cases} \tag{3.2}$$

Equations 3.3 to 3.8 show the state's energy consumption, specifically the ones related to the car and home battery, based on the state decisions. In equation 3.3, e_t^{car} represents the energy used to charge the car battery. With that info, the total energy demand from

home is the sum of the charger's energy to the car plus the energy demand for noncontrollable usage and air conditioning, as shown in equation 3.4.

Depending on the battery decisions, there will be energy from the battery to the house, as shown in equation 3.5, represented as e_t^{bu} and from battery sold to the grid, equal to e_t^{bs} on equation 3.6. The model permits different magnitudes of power to be sold or used on the battery. For that purpose, the variables $w_s^r \in (0,1)$ and $w_u^r \in (0,1)$ stand for the fraction of power that will be sold or used respectively. The car can also give some of its battery energy to the house, representing its value as e_t^{cu} on equation 3.7.

Finally, an expression for the unsatisfied energy demand is expressed as e_t^{us} on equation 3.8, generated when the house's power added to the battery energy used for satisfying energy consumption is not enough. This energy is used later to calculate the penalization for unsatisfied energy demand.

$$e_t^{cc} = c_{1t}\min(W_c, N-\nu_t) + c_{2t}\min(W_c/4, N-\nu_t) \tag{3.3}$$

$$e_t^{Total} = e_t^{NC} + e_t^{AC} + e_t^{car} \tag{3.4}$$

$$e_t^{bu} = \sum_{r=1}^{n_u} b_t^r \min(e_t^{Total}, \mu_t, W_d w_u^r) \tag{3.5}$$

$$e_t^{bs} = \sum_{r=1}^{n_s} b_t^4 \min(\mu_t, W_d w_s^r) \tag{3.6}$$

$$e_t^{cu} = \max(\min((e_t^{Total} - e_t^{bu}), \nu_t, W_c), 0)) \tag{3.7}$$

$$e_t^{us} = \max(e_t^{Total} - e_t^{bu} - e_t^{cu} - W_h, 0)v \tag{3.8}$$

With the equations above, the cost function is defined as follows.

$$q_t(s_t, b_t, c_t) = \min(e_t^{Total} - e_t^{bu} - e_t^{cu}, W_a)C \\ + q_t^{ce}(s_t) + e_t^{us}\chi - Ee_t^{bs} \tag{3.9}$$

3.8. Transition functions

Three variables are affected by the actions taken for a given state: the house battery and the car battery. Other exogenous factors, such as temperature, PV generations, are given by historical data.

The transition of the car energy when the car is not at home was described in section 4.7. The transitions for both car and home battery are described below.

$$\nu_{t+1} = \min([\nu_t - e_t^{bu} - e_t^{bs} + \rho_t b_{0t}, M]) \tag{3.10}$$

$$\mu_{t+1} = \max([\mu_T + e_t^{cc} - e_t^{cu} - (1 - \gamma_t)\kappa\psi, 0]) \tag{3.11}$$

3.9. Bellman's equations

Bellman equations are presented from 3.12 to 3.14. They represent the initial states and actions for any recursion of the model. The value function takes the value of 0 on the last stage of recursion, that is when the end condition is met. Finally, on 3.14 it is shown that the decisions are made post cost calculation, and the decisions will affect the future and that the model looks for minimizing the cost of the decisions.

$$x_0 = (b_0, c_0) \tag{3.12}$$

$$V_T(*) = 0 \tag{3.13}$$

$$V_t^*(s_t, b_t, c_t) = q_t(s_t, b_t, c_t) + \min_x(V_{t+1}(s_{t+1}(b_t, c_t), b_{t+1}, c_{t+1})) \tag{3.14}$$

3.10. Value function approximation

The Value Function Approximation method (VFA) consists of approximating a state's value for a single set of chosen actions. The approximator is trained with existing data by using different regression methods until convergence.

In the proposed model, we use a machine learning model called Neural Network. As mentioned before, the neural network consists of layers of neurons (perceptrons) that perform optimizations over its parameters using the training data and loss functions, in this case, the Mean Squared Error with an algorithm called backpropagation (Hecht-Nielsen, 1992b). In this particular case, the Neural Network is used as a regression method, as it approximates the value function of the Reinforcement Learning scheme. The model will simulate a function $f(x) : X \rightarrow Y$ that receives a vector composed of the state and action variables of size $|S| + |A|$ and outputs a scalar of size 1 that represents the cost of the future decisions by choosing the actions for the given state. The Neural Network is fitted with historical data to adjust its parameters to make a good estimation of the future.

4. COMPUTATIONAL EXPERIMENTS

The following experiments were modeled using Python 3 programming language, the library sci-kit-learn for the neural network models using the Multi-Layer Perceptron model, and Pandas for reading the CSV data.

All the experiments were run using, Google Colaboratory platform, with a single shared core, with two threads virtual machine with Intel(R) Xeon(R) CPU @ 2.30GHz and 12 GB of RAM, running over Linux.

4.1. Case of study

The data used for the experimental section belongs to Pecan Street Inc. Dataport, which was public for academic purposes by 2019. The data contains information about noncontrollable energy consumption, HVAC energy consumption, electric vehicle charger power, and weather conditions such as temperature and humidity. All this information is provided in intervals of one hour.

The dataset contains data from many houses for the United States of America, mainly from Texas and California. The experimental scenario chosen belongs to a household in San Diego, California, USA. The total amount of rows of data is 518.400, as it contains 360 days of 24 hours each. Only five days of the year did not have complete data. They had repeated hours or lost data (temperature, energy usage, among others). We removed all those days with incomplete or missing information from the dataset for cleaner information.

For the parameters presented in section 5, some assumptions are taken for simplicity and feasibility of solving the problem. First, the energy cost C from the network is set with a flat rate equal to the average price in Santiago, Chile, which is 105 CLP, which equals approximately 0,15 USD per kWh. Secondly, as seen in (Virta, 2019), the average distance driven by a car in a day is 46 km. Also, the average energy used per kilometer

Table 4.1. Instance values

C	\$105
D	\$280
M	10 kWh
N	15 kWh
W^{hc}	5 kW
W^{hd}	5 kW
W^{c}	4 kW
W^{d}	1 kW
W^{a}	10 kW
χ	630
κ	2,5 km
ψ	0,2 kWh/km

driven is about 0,2 kWh. Given that information and based on the training data, the average kilometers driven by a car, counting only hours when the car is not present at the house, is 2.5km. Therefore, every time the car is not parked in the house, the battery uses $2,5 \times 0,2 = 0,5$ KWh. In (Copec, n.d.), it is shown that the price of KWh in a fast charger is around 230 Chilean peso, which equals 0,3 USD approximately. The model assumes that the car cannot be out-of-gas, so, rationally, the car owner will charge the car with enough energy to come back to the house with 0 energy. Therefore the car will come back home with 0 energy. Still, the car owner pays for the overused energy (when the energy used in an hour block is greater than the energy stored in the car battery) with a price of 230, as a penalty for the model of making the car go outside home without enough energy. The cost penalty for every kWh of unsatisfied demand is defined as 630, which is 6 times the current cost.

4.2. Data preprocessing

The data used had to be cleaned, as it had more information than needed and missing data. As the dataset contains columns for many electrical appliances, but in this case, none of them were recorded separately except for the car electrical usage and air conditioning, we used only the aggregated electrical consumption air conditioning records.

Then, processed data consists of hourly electrical noncontrollable usage, air conditioning usage, solar energy captured by the photovoltaic panel, the car's energy from the charger, temperature, and humidity. As the proposed model decides whether or not to charge the electric vehicle, the energy from the dataset shows how much energy the real car used in each hour. It is then used to define the status of connected or disconnected to the house of the vehicle. Every time the energy is greater than zero, the car is connected to the house. Therefore, for simplicity, the car is at the house every time that it is connected and outside the house when it is not.

As for the vectors used as the model input, they are composed of 41 components, of which 8 of them are float numbers for every state except the hour of the day. The nature of each variable of the state is described as follows.

(i) $X_0 = e_t^{NC} \in R^+$
(ii) $X_1 = e_t^{AC} \in R^+$
(iii) $X_2 = T_t \in R^+$
(iv) $X_3 = H_t \in [0, 100]$
(v) $X_4 = \rho_t \in R^+$
(vi) $X_5 = \gamma_t \in \{0, 1\}$
(vii) $X_6 = \nu_t \in [0, N]$
(viii) $X_7 = \mu_t \in [0, M]$

The next 24 components stand for every hour of the day, from 0 to 23, in a binary representation. Among these 24 numbers, only one at a time can and must be one, while the others remain 0.

$$X_n \in \{0, 1\}, \forall n \in [8, 31] \tag{4.1}$$

$$\sum_{8}^{31} X_n = 1 \tag{4.2}$$

The Same technique is applied for the actions. For this particular model, actions were discretized to reduce decision space and simplify the problem. The action values are represented as binary vectors, where just one of each can equal to 1 at the time. Then the actions b and c are described as the following:

$$b_t = (b_{1t}, b_{2t}, b_{3t}, b_{4t}, b_{5t})$$

$$c_t = (c_{1t}, c_{2t}, c_{3t}, c_{4t})$$

Where $b_{nt} \in \{0, 1\}$, $\forall n \in \{1, 2, 3, 4, 5\}$ and $c_{nt} \in \{0, 1\}$, $\forall n \in \{1, 2, 3, 4\}$.

For each vector, the meanings of each action are described on the above lists.

For b_{nt}:

(i) battery is charged, which means the system will store all energy captured by the PV panel on the current stage in the battery.
(ii) battery energy is used at maximum power
(iii) battery energy is used at less power
(iv) battery energy is sold at maximum power
(v) battery energy is sold at less power

For c_{nt}:

(i) fast charging
(ii) slow charging
(iii) use car battery
(iv) do nothing (idle connected car or the default action when the car is not at home).

Then the cost functions are simplified and described as follows.

$$e_t^{bu} = b_{2t} \min(e_t^{Total}, \mu_t, W_d) + b_t^3 \min(e_t^{Total}, \mu_t, W_d/2) \tag{4.3}$$

$$e_t^{bs} = b_t^4 \min(\mu_t, W_d) + b_t^5 \min(\mu_t, W_d/2) \tag{4.4}$$

The next 9 components of the vector will represent this model's actions: the first five are for the house battery and the final four for the car battery. Each group represents the action chosen in order from one to five. For the first five, one and only one component can and must be one, while the others remain zero, representing the house battery's decision. The same occurs to the car decision on the four final vector components.

$$X_n \in \{0, 1\}, \forall n \in [32, 36] \tag{4.5}$$

$$\sum_{32}^{36} X_n = 1 \tag{4.6}$$

$$X_n \in \{0, 1\}, \forall n \in [36, 41] \tag{4.7}$$

$$\sum_{36}^{41} X_n = 1 \tag{4.8}$$

This technique is known as one-hot encoding and prevents the model from getting linear on categorical variables, such as the hour of the day and the actions (*One hot encoding*, n.d.).

For creating the training data, real info from the dataset is used. There is an infinite space of possible states on battery energy and car energy, which is not in the dataset, so for making this computable, a discretization is made. This means that for every hour block of data, which includes X_0 to X_5, the model generates some discrete values for the remaining state variables.

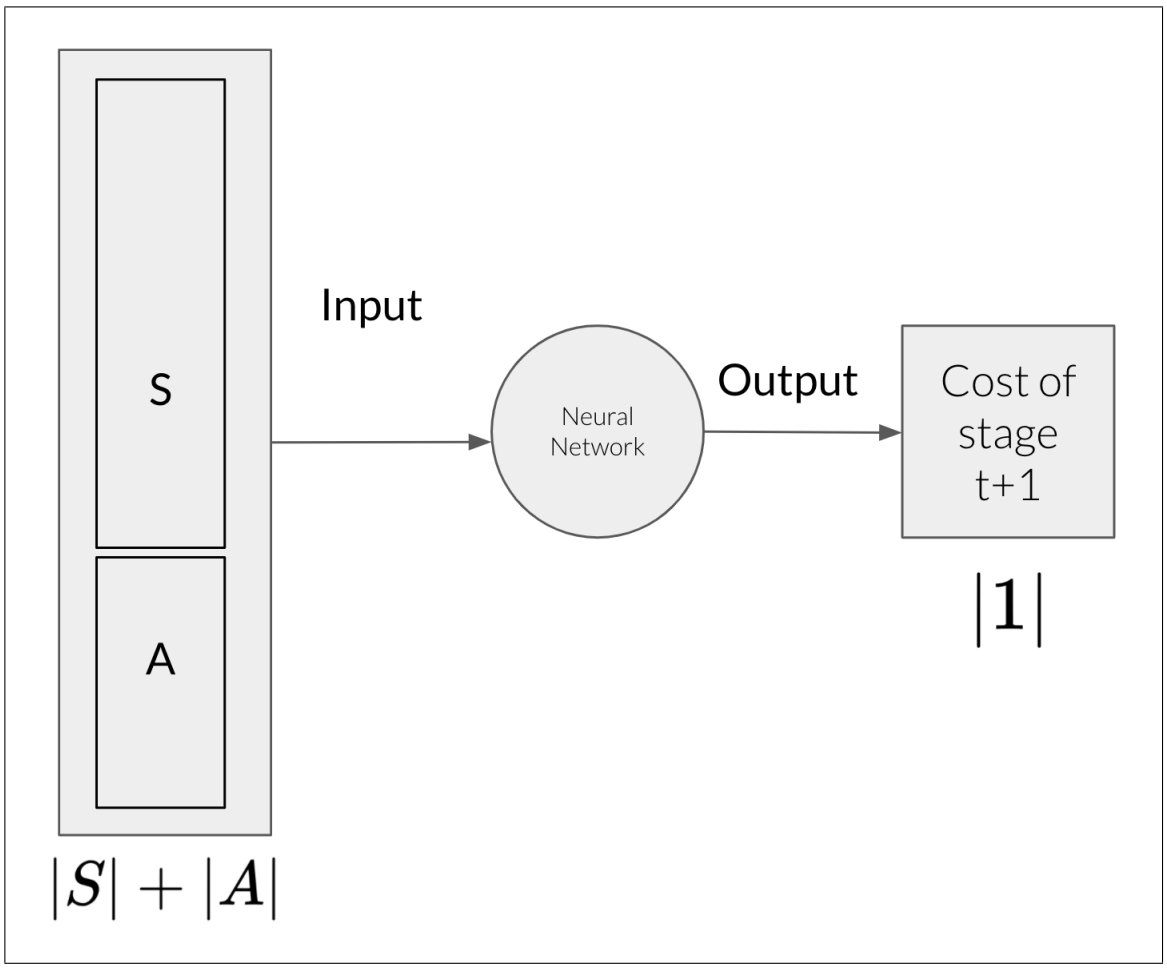


Figure 4.1. Neural network input and output

As this is a reinforcement learning scheme, the model goes into recursion for a defined horizon of time to approximate the value function by only looking 12 hours ahead for this particular case.

Dataset generation is presented in algorithms 1 and 2.

Both functions $feasibleB$ and $feasibleC$ return the feasible actions for the current stage, related to the current state (car and battery energy), and prevent the model from overcharging or using more energy than the batteries have.

Algorithm 1: Data creation algo

X = [];
Y = [];
for $t \leftarrow 0$ **to** T **do**
    if *car energy* > 0 **then**
        $\rho = 1$
    else
        $\rho = 0$
    end
    for $\nu \in N^{+} \leq N$ **do**
        for $\mu \in N^{+} \leq M$ **do**
            for $b \in feasibleB(t)$ **do**
                for $c \in feasibleC(t)$ **do**
                    $x = [e_t^{NC}, e_t^{AC}, T_t, H_t, \rho_t, \gamma_t] + OneHot(hour_t) +$
                    $OneHot(\nu) + OneHot(\mu) + b_t + c_t$;
                    $X.append(s)$;
                    $y = StateRecursion(t+1, b, c, d = 0)$;
                    $Y.append(y)$
                end
            end
        end
    end
end

4.3. Model and architecture calibration

As mentioned before, to approximate the cost of a pair of state-actions, a neural network is used. Neural networks can use different optimization methods for recalculating their parameters. The *scikit-learn* library provides three of them: lbfgs, Stochastic Gradient Descent (SGD), and Adam (Kingma & Ba, 2014). In this model, SGD, Adam, and lbfgs are compared as lbfgs is the best on short datasets in training time and accuracy, and SGD and Adam perform better in longer datasets. The vector of state/action and the result of the recursion are the data used to train the neural network, so it approximates the future cost of the current decisions.

Algorithm 2: StateRecursion

input : t the stage and d the deepness of the recursion, b_t and c_t the actions for the current stage
output: The cost y of the recursion

```
X = [];
for t ← 0 to T do
    if d == 12 then
        return 0
    else
        cost, s_{t+1} = q_t(s_t, b_t, c_t);
        bestCost = ∞;
        for b ∈ feasibleB(t) do
            for c ∈ feasibleC(t) do
                y = StateRecursion(t + 1, b, c, d + 1);
                if y < bestCost then
                    bestCost = y;
                else
                    continue
                end
            end
        end
        return bestCost;
    end
end
```

Neural networks work well with specific structures depending on the data nature. Hyper Parameter Tuning consists of training different neural network configurations and testing its results, so the best combination is selected as a final model. In this case, five hyperparameters were analyzed: batch size, neural network layer structure, initial learning rate, epsilon (for Adam), and optimizer.

The batch size is the number of vectors given to the model for calculating the loss and then optimizing the network weights. The loss is calculated based on each iteration's result, comparing the model output on each vector with the expected result on the training set. On each epoch, only the batch size of vectors is given for training. That means that the model is backpropagated (parameters are readjusted based on loss of that iteration) on each epoch several times, given by the total vectors divided by the batch size. Therefore, more

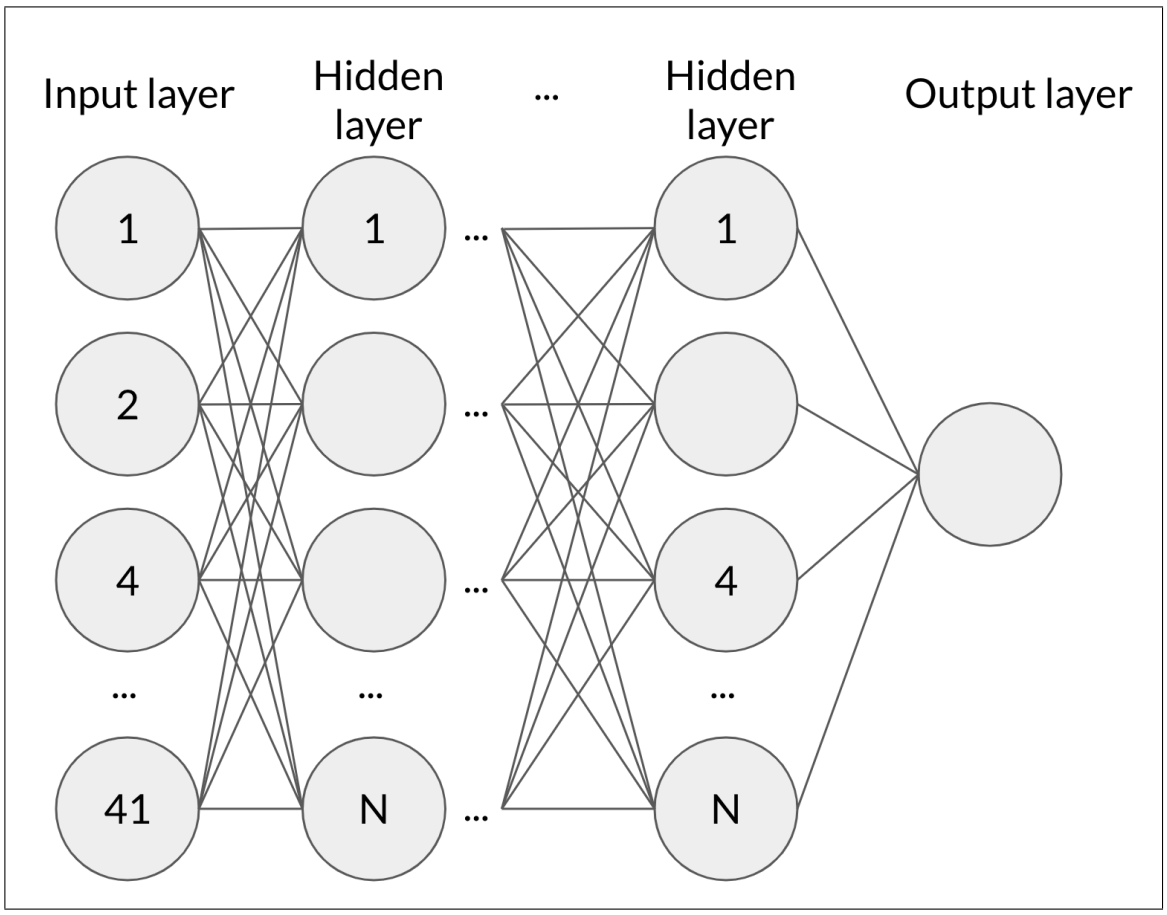


Figure 4.2. Hourly average

or fewer times that the model is trained on each epoch will affect the model's convergence time.

The layers of the model explain how neurons are connected in the inner layers of the neural network. For this experiment, the model is evaluated in two ways: the size of one hidden layer and the number of hidden layers of the same size. We chose three configurations to measure how good the model was with one hidden layer: 16, 32, and 64 neurons. On the other hand, for the number of layers of the same size, configurations of two layers of 16 neurons were added to the experiments, giving a total of four configurations (see Figure 4.2).

Table 4.2. Optimizer Grid Search

Optimizer	**Training Time**	**Mean R^2 score**	**Std. R^2 score**
SGD	155s	0.296	0.079
lbgfs	74s	0.32	0.118

Table 4.3. Epsilon, using Adam, 0.001 chosen.

Epsilon	**Training Time**	**Mean R^2 score**	**Std. R^2 score**
0.0001	199s	0.317	0.042
0.001	199s	0.355	0.07
0.00001	176s	0.312	0.041

Table 4.4. Learning rate, using Adam

Initial LR	**Training Time**	**Mean R^2 score**	**Std. R^2 score**
0.01	155s	0.313	0.117
0.005	74s	0.383	0.017
0.05	74s	0.296	0.05
0.001	74s	0.434	0.046

On learning rate, as it is used only in SGD, was manually tested. Learning rate means how much the weights of the model will change on each iteration. Results show that initial learning rates greater than $1e-5$ make the model diverge almost instantly. Lower rates make the model too slow for convergence. Then $1e-5$ is chosen for all tests.

For all these experiments, Grid Search (*3.2. Tuning the hyper-parameters of an estimator*, n.d.) is used to get the best combination. There is a total of four fits (model training) for each option (e.g., four events of training using lbfgs and four using SGD). The data used are the first to months of the dataset, as two is the number of months used on the rest of the experiments. The first experiment was to choose the best optimizer between lbfgs and adam. As shown in the table 4.2, lbfgs perform better on average and in all data partitions done by the Grid Search. For the remaining experiments, Adam is chosen as those parameters only affect SGD-based algorithms.

In tables 4.4, 4.6, 4.3, 4.5 and 4.2, the results of the grid search are shown. As it can be seen, the parameters chosen for the final models are the batch size of 32, a learning rate

Table 4.5. Batch size, using Adam

Batch Size	**Training Time**	**Mean R^2 score**	**Std. R^2 score**
16	155s	0.332	0.037
32	74s	0.374	0.06
64	74s	0.339	0.093
128	74s	0.343	0.042

Table 4.6. Hidden layer structure, using Adam

Layers	**Training Time**	**Mean R^2 score**	**Std. R^2 score**
(16)	109s	0.34	0.051
(32)	173s	0.103	0.16
(64)	298s	-0.049	0.224
(16 -> 16)	160s	0.048	0.181

of 0.00001 for SGD and 0.001 for Adam, one layer of 16 neurons for layer structure, and an epsilon 0.001 for both Adam and SGD.

Despite the results of the optimizer, every optimization model is tested in the next section.

4.4. Practical advantages of the proposed method.

For having knowledge about the effectiveness of the proposed model, some constant policies were developed and tested against the reinforcement learning scheme proposed. The following policies were made with information of the test dataset, so they are made to take advantage of the nature of the data.

The first policy consists of charging the battery whenever it is empty; otherwise, it is used for house consumption. The second policy selects hours of the day for charging and discharging the car, using Policy I for the home battery. Hours of the day are chosen by test data analysis, selecting the hours with more energy usage, but considering that the car needs to be charged during the night to be used in the morning. The third policy is set to charge the house battery when there are peak hours of solar energy, uses energy when it

Table 4.7. Results comparison

Policy	**Yearly cost**	Improvement (%)
Model SGD	1,397,417	-
Model lbfgs	1,483,407	-
Model Adam	1,422,779	-
Policy I	1,630,345	16,7%
Policy II	1,542,444	10,4%
Policy III	1,405,426	0,6%

is better than charging by looking 24 hours ahead. It uses the same policy as policy II for the car.

- Policy I: charge the battery when disconnected and use the energy when there is energy greater than 0 kWh.
- Policy II: always charge the battery and use it when it is greater than 0. Charge the car when energy is under threshold $\tau = 8$ when the hour is between 00:00 and 7:00 AM. Use the car energy between 21:00 and 00:00.
- Policy with future information: charge battery in peak generation hours, and use its energy when it is better than charging, by looking 24 hours ahead in the future. For the car, it is the same as Policy II.

This experiment consists of simulating 10 months of data in a real-time situation, in which the model is trained with two months of data once a month for predicting the decisions for the whole next month. This experiment aims to determine how the model performs along the year while retraining once a month. Months are approximated, and they are all composed of 30 days.

Training time takes 20 minutes, while preprocessing data takes around 6 hours for generating simulation vectors by searching all possible states for the set of actions, with a recursivity depth of 12 hours (blocks of time). For a real-time scenario, both preprocessing and training have to be made as a pipeline. Value function calculation is then instantaneous (less than a second) when the model is already trained.

Table 4.8. Results Model against Policy II per month, months 3 to 11.

Month	Model	Policy II	Improvement	Total Solar energy
3	279626	301717	8%	912 kWh
4	292070	314385	8%	1075 kWh
5	252910	284559	13%	1152 kWh
6	226252	266316	18%	1176 kWh
7	249648	282444	13%	1077 kWh
8	266246	296946	12%	964 kWh
9	279071	316020	13%	776 kWh
10	301049	332590	10%	611 kWh
11	327949	348842	6%	469 kWh
12	182120	192564	6%	195 kWh

Table 4.9. Results Model against complete information policy per month, months 3 to 11.

Month	Model	Complete Info	Improvement	Total Solar energy
3	279,626	276,826	-1%	912 kWh
4	292,070	292,798	0%	1075 kWh
5	252,910	263,374	4%	1152 kWh
6	226,252	241,092	7%	1176 kWh
7	249,648	255,664	2%	1077 kWh
8	266,246	266,879	0%	964 kWh
9	279,071	283,173	1%	776 kWh
10	301,049	300,519	0%	611 kWh
11	327,949	318,911	-3%	469 kWh
12	182,120	177,853	-2%	195 kWh

It is noticeable that SGD was the best optimizer for the model in this experiment. Unlike grid search, what matters in this experiment is making the best decisions to minimize costs. In contrast, on grid-search, the criteria are based on the R^2 score, which only measures how good the regression of the neural network is, according to the training data. Answering our second research question (RQ2), neural network hyper-parameters selection matter for a better prediction.

As seen in table 4.7, the model overcomes all policies by the percentages presented at the table, specifically a 10,4% against the best policy, in the year overall. It also overcomes the complete information policy of 0,6%. This policy is very close to the results of

Table 4.10. Results Model against Policy II per hour average.

Hour	Model	Policy	Improvement
0	281	303	7.59%
1	302	413	36.8%
2	365	453	24.2%
3	278	355	27.6%
4	192	251	31.25%
5	159	191	19.69%
6	126	157	24.22%
7	113	145	27.7%
8	117	141	20.64%
9	126	146	15.94%
10	145	158	8.91%
11	183	193	5.25%
12	206	233	12.82%
13	211	239	13.19%
14	178	209	17.24%
15	178	167	-6.18%
16	150	148	-1.63%
17	158	137	-13.24%
18	149	141	-5.55%
19	185	148	-19.62%
20	182	150	-18.04%
21	187	157	-15.97%
22	210	216	2.92%
23	243	256	5.46%

the proposed model, as this policy was created according to the data nature, knowing in advance what was going to happen in the simulation.

It is also noticeable that the proposed model performs better in months with more solar energy generation, showing that for question (RQ3), the season of year affects how the model performs. This can be related to the fact that the model is trained with data from the past two months. It can be seen that solar energy decreases considerably from month 8, so for months 11 and 12, the data used to train the model is from months with almost double the solar energy produced in the case of the 11th month and three times on month 12. Data from past years of the same months could have improved the model performance.

For answering RQ4, the hourly average shown in figure 4.3 shows how the model (blue line) performs better along the day than Policy II (orange line). Nevertheless, between 15 and 21 hours, the policy overcomes the proposed model, on average. This could be related to how these hours, the policy uses the energy from the battery whenever its energy is greater than zero. Hence, it gets empty for the night, having only energy from the car battery. Instead, the proposed model learns that it needs to save energy for night hours when there is no solar generation.

In figure 4.4, the blue line represents the cost generated by the model in an example week, while the orange line represents the Policy II costs. It can be seen that there is a peak where Policy II cost is much more than the proposed model. That is related to a car recharge since Policy II is forced to charge the car when its battery level is under a threshold of energy, while the proposed model decides it according to its own training.

Our first question RQ1 was if energy consumption data in a household be generalized in order to approximate future consumption. As the experiments show, it depends strongly on the epoch of the year. The model is developed in a way that it takes advantage of energy storage, nevertheless in months with less energy consumption the data of the past months differed too much to make a good approximation. Data can be used for predicting better the future, but probably different models will be required for different types of scenarios.

With the fast development of smart sensors with energy consumption recording, added to cheap and powerful enough hardware like IoT controllers such as Raspberry Pi, the proposed model can be easily implemented in residential buildings.

Although the model is better in costs than the policy in this experiment, initial costs are not considered, such as the solar panel installation and purchase cost, microcontrollers if they are not previously installed, the house battery, and the car charger. Therefore the model is meant to work in an environment with the technical requirements previously implemented.

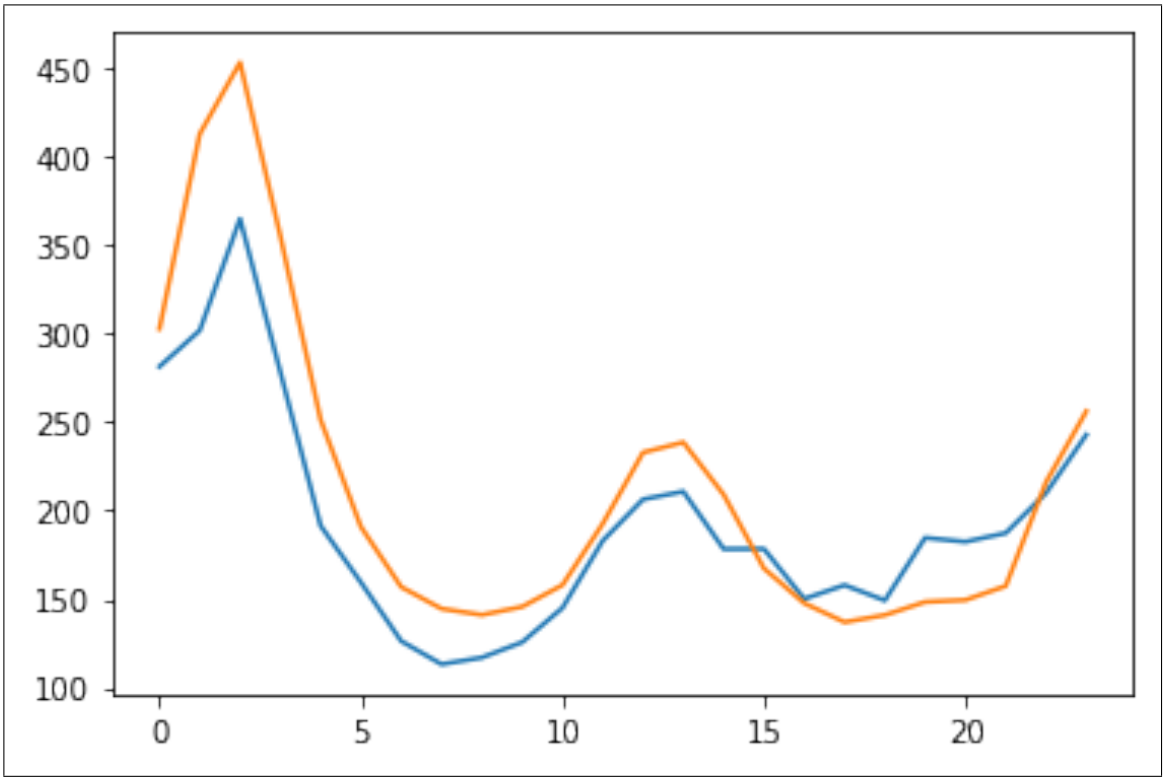


Figure 4.3. Model (blue) vs Policy II (orange) hourly average cost for a full day.

Further work can include different pricing policies with variable prices varying on the amount of energy used, the hour of the day, or the year's seasons. Battery damage or wear can also be a topic to include in further research and consider other costs not related to operational energy costs.

Since the experiments are only one instance of study, we encourage new scenarios with different configurations of the reinforcement learning scheme and techniques other than Value Function Approximation to deal with the curse of dimensionality of this problem.

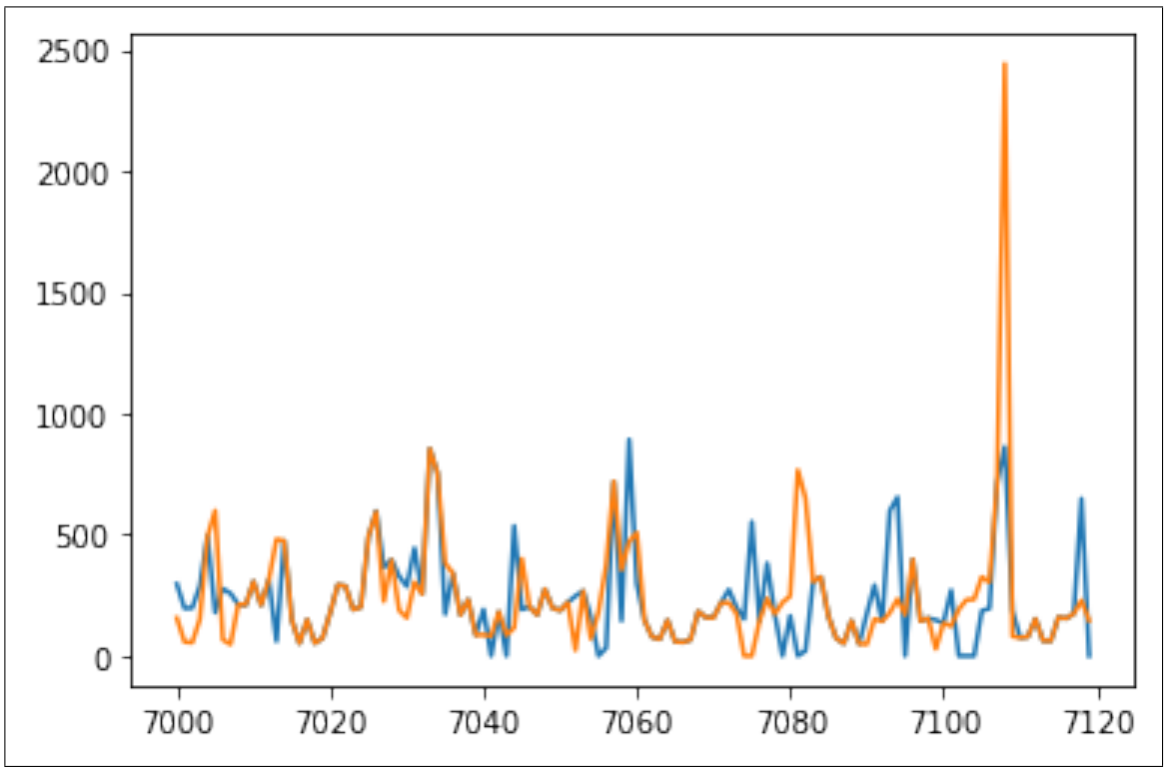


Figure 4.4. Example 5 days of costs for Model (blue) and Policy II (orange) againt the hour block of the data set (hour of the year)

5. CONCLUSIONS

This research introduces a way of modeling a reinforcement learning scheme for optimizing energy consumption in an environment with electric vehicles and renewable energies. In (Vázquez-Canteli & Nagy, 2019) it can be seen that most techniques that aim to solve similar problems only focus on electric vehicles alone or in home appliances. Our work goes further by proposing a way of taking advantage of energy storage, expandable to other distributed energy resources. The availability of data records of energy consumption and weather conditions opens computer and data science opportunities to contribute to demand-side energy management topics. In this line, historical data to train models give results that cannot be simply concluded by simple human inspection, such as how better to manage solar energy in months of more generation. The development of computational open-source tools makes it easier to implement good and fast solutions for energy problems, like the model proposed, without too many computational resources. That could be an opportunity to deploy this type of solution in computer devices inside homes.

REFERENCES

3.2. tuning the hyper-parameters of an estimator. (n.d.). Retrieved from https://scikit-learn.org/stable/modules/grid_search.html

Afzalan, M., & Jazizadeh, F. (2020). A machine learning framework to infer time-of-use of flexible loads: Resident behavior learning for demand response. *IEEE Access*, *8*, 111718–111730.

Avendano, D. N., Ruyssinck, J., Vandekerckhove, S., Van Hoecke, S., & Deschrijver, D. (2018). Data-driven optimization of energy efficiency and comfort in an apartment. In *2018 international conference on intelligent systems (is)* (pp. 174–182).

Brown, J., Abate, A., & Rogers, A. (2020). Disaggregation of household solar energy generation using censored smart meter data. *Energy and Buildings*, 110617.

Chile, C. (2020). *Gobierno presenta programa casa solar que facilita el acceso a paneles con sistemas fotovoltaicos.* Retrieved from https://www.cnnchile.com/pais/programa-casa-solar-paneles-sistema-fotovoltaicos_20201006/

Chung, H.-M., Maharjan, S., Zhang, Y., & Eliassen, F. (2020). Distributed deep reinforcement learning for intelligent load scheduling in residential smart grid. *IEEE Transactions on Industrial Informatics*.

Copec. (n.d.). *Electromovilidad donde necesites.* Retrieved from https://ww2.copec.cl/copecvoltex

de Chile, G. (2018). *Gob.cl - artículo: Presidente piñera presentó plan para cerrar todas las centrales energéticas a carbón para que chile sea carbono neutral.* Retrieved from https://www.gob.cl/noticias/presidente-pinera-presento-plan-para-cerrar-todas-las-centrales-energeticas-carbon-para

-que-chile-sea-carbono-neutral/

Di Giorgio, A., Liberati, F., & Pietrabissa, A. (2013). On-board stochastic control of electric vehicle recharging. In *52nd ieee conference on decision and control* (pp. 5710–5715).

Ebrahimi, M., & Rastegar, M. (2020). Data-driven charging load estimation of behind-the-meter v2g-capable evs. *IEEE Transactions on Industry Applications*.

Gao, J. (2014). Machine learning applications for data center optimization.

Google. (2018). *Moving toward 24x7 carbon-free energy at google data centers: Progress and insights.* Retrieved from https://storage.googleapis.com/gweb-sustainability.appspot.com/pdf/24x7-carbon-free-energy-data-centers.pdf

He, Y., Venkatesh, B., & Guan, L. (2012). Optimal scheduling for charging and discharging of electric vehicles. *IEEE transactions on smart grid*, *3*(3), 1095–1105.

Hecht-Nielsen, R. (1992a). Theory of the backpropagation neural network. In *Neural networks for perception* (pp. 65–93). Elsevier.

Hecht-Nielsen, R. (1992b). Theory of the backpropagation neural network. In *Neural networks for perception* (pp. 65–93). Elsevier.

Henri, G., & Lu, N. (2019). A supervised machine learning approach to control energy storage devices. *IEEE Transactions on Smart Grid*, *10*(6), 5910–5919.

Kazmi, H., Suykens, J., Balint, A., & Driesen, J. (2019). Multi-agent reinforcement learning for modeling and control of thermostatically controlled loads. *Applied energy*, *238*, 1022–1035.

Kim, S., & Lim, H. (2018). Reinforcement learning based energy management algorithm for smart energy buildings. *Energies*, *11*(8), 2010.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Liu, Y., Guan, X., Li, J., Sun, D., Ohtsuki, T., Hassan, M. M., & Alelaiwi, A. (2020). Evaluating smart grid renewable energy accommodation capability with uncertain generation using deep reinforcement learning. *Future Generation Computer Systems*, *110*, 647–657.

Lu, R., & Hong, S. H. (2019). Incentive-based demand response for smart grid with reinforcement learning and deep neural network. *Applied energy*, *236*, 937–949.

Lu, R., Hong, S. H., & Yu, M. (2019). Demand response for home energy management using reinforcement learning and artificial neural network. *IEEE Transactions on Smart Grid*, *10*(6), 6629–6639.

Lu, X., Liu, Z., Ma, L., Wang, L., Zhou, K., & Feng, N. (2020). A robust optimization approach for optimal load dispatch of community energy hub. *Applied Energy*, *259*, 114195.

Mena, R., Escobar, R., Lorca, Á., Negrete-Pincetic, M., & Olivares, D. (2019). The impact of concentrated solar power in electric power systems: A chilean case study. *Applied Energy*, *235*, 258–283.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... others (2015). Human-level control through deep reinforcement learning. *nature*, *518*(7540), 529–533.

Mocanu, E., Mocanu, D. C., Nguyen, P. H., Liotta, A., Webber, M. E., Gibescu, M., & Slootweg, J. G. (2018). On-line building energy optimization using deep reinforcement learning. *IEEE transactions on smart grid*, *10*(4), 3698–3708.

Ning, C., & You, F. (2019). Optimization under uncertainty in the era of big data and deep learning: When machine learning meets mathematical programming. *Computers & Chemical Engineering*, *125*, 434–448.

One hot encoding. (n.d.). Retrieved from `https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html`

Rahman, M. M., Oni, A. O., Gemechu, E., & Kumar, A. (2020). Assessment of energy storage technologies: A review. *Energy Conversion and Management*, *223*, 113295.

Sammut, C., & Webb, G. I. (2017). *Encyclopedia of machine learning and data mining*. Springer.

Simsek, Y., Sahin, H., Lorca, Á., Santika, W. G., Urmee, T., & Escobar, R. (2020). Comparison of energy scenario alternatives for chile: Towards low-carbon energy transition by 2030. *Energy*, 118021.

Vázquez-Canteli, J. R., & Nagy, Z. (2019). Reinforcement learning for demand response: A review of algorithms and modeling techniques. *Applied energy*, *235*, 1072–1089.

Virta. (2019, Apr). *Ev charging 101 - how much electricity does an electric car use?* Liikennevirta Oy (Ltd.). Retrieved from `https://www.virta.global/blog/ev-charging-101-how-much-electricity-does-an-electric-car-use`

Wang, S., Du, L., Ye, J., & Zhao, D. (2020). A deep generative model for non-intrusive identification of ev charging profiles. *IEEE Transactions on Smart Grid*.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, *78*(10), 1550–1560.

Zhang, D., Li, S., Sun, M., & O'Neill, Z. (2016). An optimal and learning-based demand response and home energy management system. *IEEE Transactions on Smart Grid*, *7*(4),

1790–1801.