TRABAJO FIN DE MÁSTER

MÁSTER UNIVERSITARIO EN CIENCIA DE DATOS E INGENIERÍA DE COMPUTADORES

Aprendizaje por refuerzo profundo para control energético en smart cities

Aplicación de técnicas multiagente con enfoque independiente para el control energético en ciudades inteligentes

Autor
Yacine Brek Prieto

Directores
Juan Gómez Romero
MIguel Molina Solana

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

—

Granada, septiembre de 2023

Aprendizaje profundo por refuerzo para control energético en smart cities

Yacine Brek Prieto

Palabras clave: aprendizaje por refuerzo profundo, control energético, ciudades inteligentes, multiagente independiente, control descentralizado, emisiones de carbono, coste económico.

Resumen

Las *smart cities* están cada vez más cerca de sustituir al modelo de ciudad convencional. Esto constituye un nuevo paradigma para el control energético, presentando una oportunidad para combatir el calentamiento global, reduciendo las emisiones de carbono y demanda energética. Considerando el desarrollo del *hardware* y de las técnicas de *deep learning* en las últimas décadas, se dan las condiciones idóneas para introducir el aprendizaje por refuerzo profundo en el control energético de ciudades inteligentes.

En este proyecto se propone el uso de aprendizaje por refuerzo profundo para el control energético en *smart cities*. Se apuesta por una arquitectura de control multiagente independiente. Es decir, cada agente toma sus propias decisiones de manera aislada, sin comunicación directa con otros agentes.

Concretamente, el objetivo es reducir las emisiones de carbono y el coste económico para cada edificio del distrito. Para ello se entrenan agentes utilizando dos algoritmos de *DRL* (*SAC* y *MARLISA*) que posteriormente se comparan con un modelo de referencia, en este caso un *RBC*. Para la realización de un estudio más completo se proponen varios escenarios con diferentes características en los que entrenar estos agentes.

Finalmente, se analiza el desempeño de cada modelo considerando múltiples métricas de evaluación. Así se obtiene un análisis más completo que permite estudiar las emisiones de carbono y el coste económico de forma conjunta e individual.

Deep reinforcement learning for energy control in smart cities

Yacine Brek Prieto

Keywords: deep reinforcement learning, energy control, smart cities, independent multi-agent, decentralized control, carbon emissions, electricity price.

Abstract

Smart cities are getting closer to replacing the conventional city model. This represents a new paradigm for energy control, offering an opportunity to fight global warming by reducing carbon emissions and energy demand. Considering the recent development of hardware and deep learning techniques, deep reinforcement learning becomes a promising solution for energy control in smart cities.

This project proposes applying deep reinforcement learning for energy control in smart cities, using an independent multi-agent control architecture. In other words, each agent makes its own decisions without direct communication with other agents.

Specifically, the goal is to reduce carbon emissions and economic costs for each building in the district. To achieve this, agents are trained using two DRL algorithms -SAC and MARLISA-, and then compared with a reference model, in this case, an RBC (Rule-Based Controller). In order to develop a more complete study, various scenarios are proposed to train these agents.

Finally, the performance of each model is analyzed, considering multiple evaluation metrics. This yields a deeper analysis that allows studying carbon emissions and cost both jointly and separately.

Yo, **Yacine Brek Prieto**, alumno del Máster Universitario Oficial en Ciencia de Datos e Ingeniería de Computadores de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 77037521X, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Máster en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Yacine Brek Prieto

Granada a 6 de Septiembre de 2023.

D. **Juan Gómez Romero**, y **Miguel Molina Solana**, profesores del departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada.

Informan:

Que el presente trabajo, titulado ***Aprendizaje profundo por refuerzo para control energético en smart cities***, ha sido realizado bajo su supervisión por **Yacine Brek Prieto**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expedimos y firmamos el presente informe en Granada, a 6 de Septiembre de 2023.

Los directores:

Juan Gómez Romero Miguel Molina Solana

Agradecimientos

Me gustaría dar las gracias a mi familia y amigos por el apoyo incondicional durante la realización de este proyecto. Además, quiero hacer otra mención especial a mis tutores, Juan y Miguel, por ofrecerme esta gran oportunidad y ayudarme siempre que lo he necesitado.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

En este capítulo introductorio se desarrolla la motivación del proyecto, la descripción de problema, los objetivos planteados y la planificación tanto económica como temporal. Dedicaremos una sección a cada uno de los apartados mencionados.

1.1. Motivación

El **cambio climático**, el **calentamiento global** o el **aumento de la contaminación** son problemas que la comunidad científica enfrenta desde hace décadas. Estos fenómenos suponen una incesante preocupación para la humanidad, ya que amenazan nuestro entorno y el desarrollo de la vida en nuestro planeta. Problemas de tal magnitud conllevan grandes responsabilidades, por lo que que **las investigaciones dedicadas a la contaminación han aumentado de manera considerable** en los últimos años (véase la figura 1.1).

En 2021, el sector de **los edificios y la construcción** representó alrededor del **37 % de las emisiones de CO2** relacionadas con la energía y los procesos (figura 1.2) y más del **34 % de la demanda de energía** a nivel mundial (figura 1.3). En ese mismo año, las emisiones de CO2 referentes a la energía operativa del sector de los edificios alcanzaron un **máximo histórico de alrededor de 10 GtCO2**[1]. Este aumento supera el nivel de 2020 en aproximadamente un 5 % y el pico previo a la pandemia de 2019 en un 2 %. Además, la demanda de energía operativa en los edificios alcanzó un **máximo histórico de 135 EJ**[2]. Esto último, se traduce en un aumento de alrededor del 4 % con respecto al período de 2020 y supera el pico anterior de 2019 en más del 3 %. El *Global Buildings Climate Tracker* indica que el sector de los edificios y la construcción sigue estando lejos de lograr

[1] GtCO2 significa mil millones de toneladas de dióxido de carbono.
[2] 1 EJ (exajulio) equivale a 10^{18}J (julios).

la descarbonización para el año 2050. Además, conflictos como la guerra de Ucrania y la consiguiente crisis energética en Europa, agravan aun más este problema. Por otro lado, tenemos el factor económico, y es que la volatilidad de los precios mundiales de la energía plantea otros riesgos, junto con la crisis del costo de vida y las implicaciones de los aumentos de las tasas de interés en la inversión en la descarbonización por parte de los gobiernos, hogares y empresas. El aumento global en el costo de vida ejercerá presión sobre los costos de endeudamiento, pero **la eficiencia energética presenta un medio para moderar la volatilidad del costo de la energía y reducir las emisiones** [1].

Estos datos sugieren que la optimización en el control energético de ciudades es una buena alternativa para reducir los niveles de contaminación y abaratar el coste de la energía. Existen otras posibilidades como el uso de energía renovables o la construcción de edificios más eficientes. En este proyecto se busca aplicar conocimiento de inteligencia artificial y ciencia de datos, con la finalidad de colaborar en el ámbito de la energía. Concretamente, **este trabajo se enfoca en el control energético de ciudades inteligentes mediante el uso de aprendizaje por refuerzo profundo.** Si se obtienen buenos resultados, esto podría verse reflejado en un considerable ahorro económico y reducción de las emisiones de carbono. Por tanto, se trata de un tema de vital importancia que puede tener un gran impacto a nivel global.

El control energético en ciudades ha sido tratado con técnicas más rudimentarias como los controladores basados en reglas (*rule-based control* o *RBC*). Su principal desventaja es la complejidad que conlleva su programación y su rediseño en caso realizar una modificación. Esto no quiere decir que en algunos casos no sean herramientas interesantes cuyo rendimiento no es necesariamente malo.

Los modelos de aprendizaje por refuerzo profundo presentan un gran potencial pero conllevan un gran coste computacional en comparación con otras técnicas. Esta es una de las razones por las que no se ha podido explotar mucho este ámbito del aprendizaje automático hasta ahora. Gracias a los avances tecnológicos y al notable desarrollo del *hardware* en las últimas décadas, el aprendizaje por refuerzo profundo se puede utilizar con éxito en multitud de problemas actualmente.

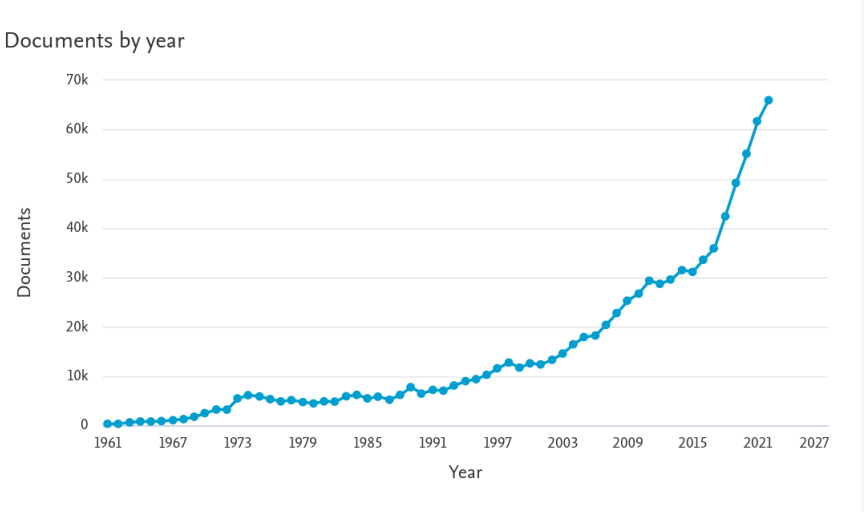

Figura 1.1: Número de artículos referentes al término *pollution* en Scopus (hasta 2022).

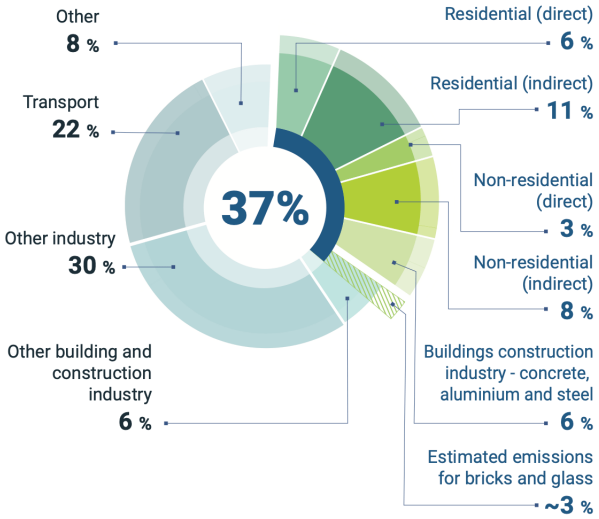

Figura 1.2: Cuota mundial de emisiones de CO2 de procesos y operaciones de edificios y construcción, 2021 [1].

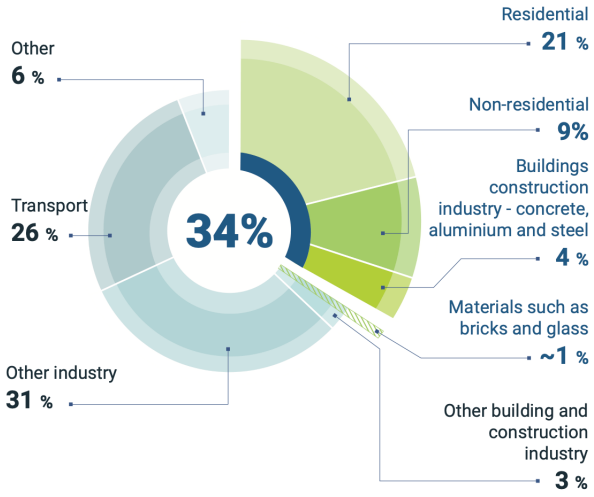

Figura 1.3: Participación mundial en la demanda de energía final de edificios y construcción, 2021 [1].

1.2. Descripción del problema

El problema planteado en este proyecto consiste en el **control energético de ciudades inteligentes mediante el uso de técnicas de aprendizaje por refuerzo profundo**. Estas técnicas se aplican con el objetivo de **reducir las emisiones de carbono y el coste económico**. Concretamente, se utiliza la arquitectura de control conocida como **descentralizada independiente**, donde cada agente se encarga de un edificio pero no hay comunicación directa entre ellos. Por lo que cada agente toma sus propias decisiones de manera independiente, sin comunicación directa con otros agentes. Se supone que cualquier interacción entre ellos es resultado de la dinámica del entorno. Para conseguir el objetivo mencionado se propondrá una función de recompensa que tenga en cuenta las emisiones de carbono y el precio de la electricidad para cada edificio.

1.3. Objetivos

Para la resolución del problema descrito en la sección anterior se plantean una serie de objetivos. Con esto se pretende optimizar la investigación para contribuir, en medida de lo posible, a la literatura científica.

Primer objetivo. Estudiar el campo del aprendizaje por refuerzo (*RL*) e investigar acerca del control energético en *smart cities*. Se pretende conocer las bases y los fundamentos del *RL* para poder plantear y

resolver el problema mediante el uso de estas herramienta. Además, se busca conocer el estado del arte sobre control energético.

Segundo objetivo. Entrar en contacto con *Autogrid* y *CityLearn* que son las plataformas software sobre las que se va a trabajar y realizar las simulaciones. En algunos casos, será necesario modificar el código para adaptar el entorno a nuestro problema.

Tercer objetivo. Realizar, ejecutar y analizar los experimentos propuestos para la resolución del problema. Estos se acompañarán de tablas, gráficas y comparativas con la finalidad de ser claros y concisos.

Cuarto objetivo. Extraer conclusiones sobre el trabajo realizado y los resultados obtenidos. Asimismo, se responderán las preguntas planteadas en la propuesta de solución al problema.

Quinto objetivo. Aportar propuestas para futuros trabajos con la intención de mejorar o dar otra perspectiva de este proyecto. Para ello, es fundamental la experiencia y el conocimiento que nos aporta la realización de este trabajo.

1.4. Cronograma

El trabajo se ha desarrollado durante un plazo de seis meses. A continuación, se muestra un diagrama de *Gantt* para la planificación seguida en este proyecto.

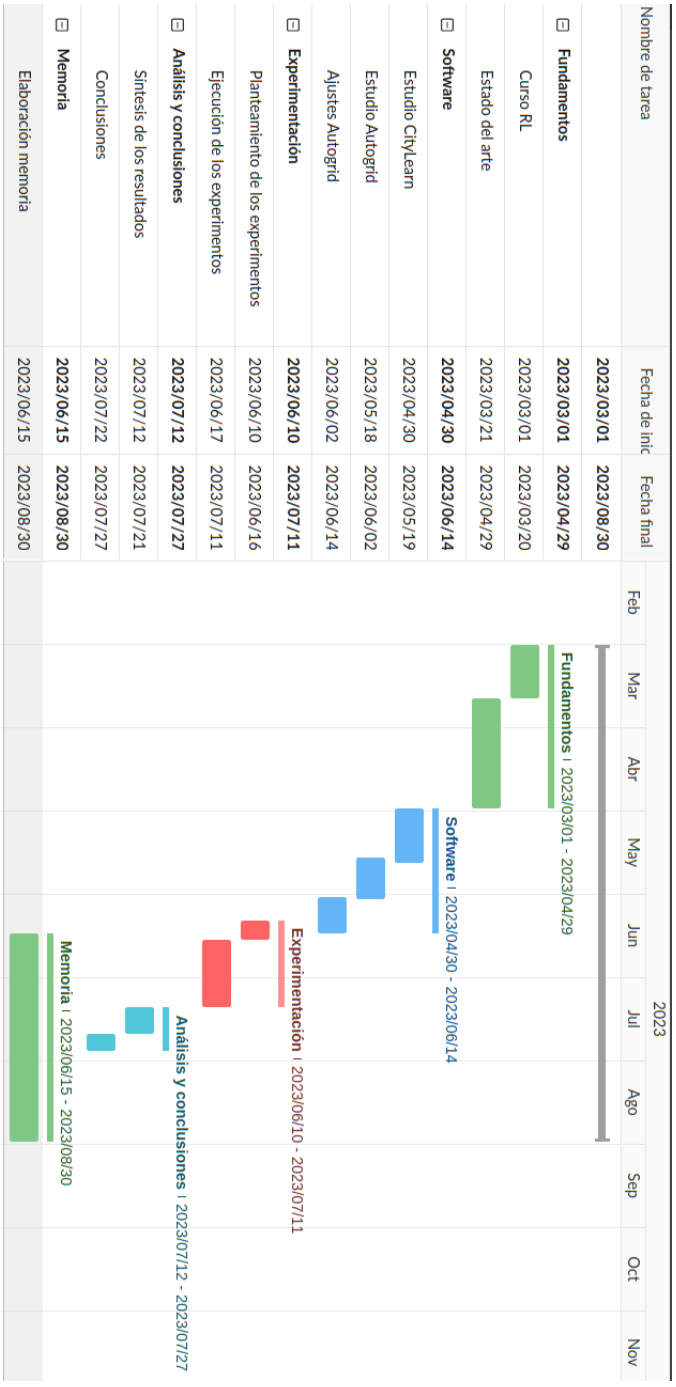

Figura 1.4: Diagrama de *Gantt* de la planificación del proyecto.

1.5. Presupuesto

El presupuesto de este proyecto se divide principalmente en **gastos de personal y de *hardware***. Se han consultado algunas fuentes para ser objetivos en las estimaciones realizadas. Los porcentajes referentes a la cotización de la Seguridad Social se han obtenido de la página web oficial. Para saber una aproximación del salario medio de un científico de datos en España, en 2023, se ha consultado GlassDoor. Por último, para conocer el IRPF y el sueldo neto que le corresponde a un trabajador de 24 años de edad, se ha utilizado esta herramienta. En la tabla 1.1 se muestra el desglose del presupuesto.

Para calcular el **coste mensual de la empresa** se ha aplicado la siguiente fórmula, siendo X el sueldo bruto mensual del trabajador:

$$X + X * (cotiz\ \ seg\ \ social\ \ trabajador)/100$$

Por otro lado, el **total de gastos en personal** se calcula multiplicando el coste mensual de la empresa por el número de meses que ha durado el proyecto, en este caso seis. Una vez obtenemos este valor, podemos determinar el presupuesto total sumando los gastos en personal y en hardware: 25.090'56 € + 1899 € = **26.989'56** €.

Presupuesto	**Cantidad**
Gastos en personal	**25.090'56 €**
Sueldo bruto anual	40.600 €
Número de pagas	6
Sueldo bruto mensual	3383'3 €
IRPF	19'75 %
Cotización Seguridad Social del trabajador	23'60 %
Cotización Seguridad Social de la empresa	4'7 %
Sueldo neto mensual del empleado	2.500'17 €
Coste mensual empresa	4.181'76 €
Gastos en hardware	**1899 €**
Ordenador portatil ASUS ROG Strix G15	1899 €

Tabla 1.1: Desglose del gasto en personal.

Capítulo 2

Antecedentes

2.1. Estado del arte

Cuando indagamos en el ámbito del control energético en ciudades inteligentes mediante el uso de técnicas de aprendizaje por refuerzo y aprendizaje por refuerzo profundo, hay algunos matices a tener en cuenta. El primero de ellos es que **los primeros trabajos datan del año 2013** (véase figura 2.1), lo que indica que es un campo relativamente reciente. Es decir, se trata de un tema que todavía no está tan explotado y no hay gran cantidad de trabajos al respecto, por lo que probablemente haya un gran margen de mejora. Por otro lado, resulta llamativo el **crecimiento que ha tenido en los últimos años.** Esto puede que tenga relación con el auge que está teniendo el aprendizaje por refuerzo profundo gracias al desarrollo del hardware que posibilita el uso de estas técnicas. También, hay que tener en cuenta que el control energético está adquiriendo cada vez más importancia porque es una posible forma de reducir la contaminación. Además, económicamente tiene un gran impacto en la sociedad puesto que la energía se ha convertido en un bien fundamental.

Dentro del aprendizaje profundo por refuerzo podemos trabajar con un único agente o con varios agentes. Este segundo enfoque es más conocido como **multiagente** y presenta dos tipos de **arquitectura de control**:

- **Descentralizado independiente**: en este enfoque, **cada agente del sistema toma decisiones basándose únicamente en sus propias observaciones y recompensas locales**. No existe comunicación o interacción directa entre los agentes. Cada uno sigue una estrategia individual de aprendizaje por refuerzo y busca maximizar su propia recompensa sin considerar las acciones o el estado de otros agentes en el sistema. En este enfoque no se busca una coordinación explícita entre los agentes y se supone que cualquier interacción entre ellos es resultado de la dinámica del entorno.

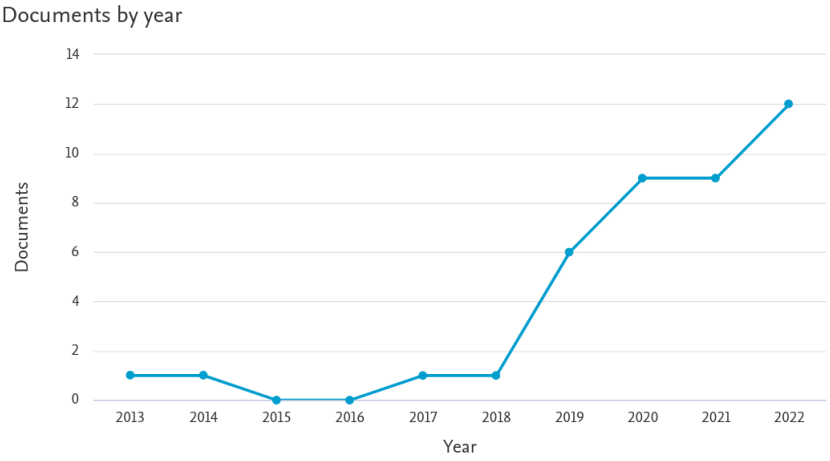

Figura 2.1: Número de artículos sobre control energético en smart cities mediante técnicas de RL o DRL en Scopus hasta 2022.

- **Descentralizado coordinado**: en este caso, **los agentes interactúan y se comunican entre sí para lograr objetivos comunes**. Si bien cada agente todavía tiene sus propias observaciones y recompensas locales, la diferencia clave aquí es que existe una coordinación activa entre ellos. Pueden compartir información, ajustar sus estrategias en función de las acciones de otros agentes y trabajar juntos para lograr objetivos que son difíciles de alcanzar individualmente. La comunicación y la colaboración son componentes esenciales en este enfoque.

A pesar de que no abundan los trabajos relacionados con esta temática, encontramos algunos artículos que han abierto camino en esta vertiente.

El trabajo [4] es uno de los más citados en lo referido a esta línea de investigación. Este artículo propone un sistema de gestión de energía basado en un algoritmo de aprendizaje por refuerzo, conocido como *Q-learning*, con la finalidad de proporcionar una política óptima para la gestión de energía en la red inteligente de una ciudad. Además, el trabajo también se centra en la privacidad de los datos de energía y propone técnicas para proteger los datos de energía en la red inteligente. Finalmente, se concluye que el uso de este algoritmo de aprendizaje por refuerzo puede **reducir los costos financieros en casi un 25 % a largo plazo**, lo que lo convierte en un enfoque prometedor para la gestión de energía en ciudades inteligentes.

El artículo [5] usa los dos tipos de arquitecturas de control mencionadas anteriormente. Este trabajo trata sobre el control energético en

edificios conectados a la red eléctrica en ciudades inteligentes mediante el uso de técnicas de aprendizaje por refuerzo. Concretamente, usa el algoritmo *MARLISA* bajo una arquitectura de control multiagente coordinado. Este algoritmo es comparado con un *RBC* (*rule based controller*) y otro algoritmo de *RL* bajo un enfoque descentralizado e independiente. El problema que se aborda es cómo **controlar de manera eficiente la demanda energética en ciudades inteligentes**, teniendo en cuenta la creciente electrificación, la integración de fuentes de energía renovable y el potencial cambio hacia vehículos eléctricos. La conclusión final es que el algoritmo *MARLISA* puede proporcionar un control efectivo y escalable de los sistemas de energía, sin la necesidad de costosos controladores basados en modelos. Además, se demostró que **los controladores de *RL* superaron al *RBC* en todas las métricas analizadas**, excepto en el consumo neto de electricidad, lo que sugiere que el enfoque de aprendizaje por refuerzo puede ser una herramienta valiosa para el control de la demanda de energía en edificios conectados a la red.

Por otro lado, el proyecto [6] se enfoca en la aplicación de técnicas de aprendizaje automático para lograr la **optimización energética en sistemas de calefacción urbana** (*District Heating* o *DH*). Concretamente, utiliza aprendizaje por refuerzo y aprendizaje supervisado en línea. El objetivo es mejorar la eficiencia energética en el lado del consumidor de la red *DH*, identificar posibles estrategias de ahorro de energía y parámetros relacionados en el sistema *DH*, y motivar la necesidad de un método de predicción en línea adaptable a cambios en los factores subyacentes que influyen en el patrón de carga de calor. Tras el estudio, los autores identifican que un método adecuado para lograr una mayor eficiencia energética de un sistema de calefacción urbana requiere una combinación de enfoques de aprendizaje automático. Específicamente, indican que el aprendizaje supervisado es un método apropiado para la predicción de la carga térmica, mientras que **el aprendizaje por refuerzo es idóneo para el control óptimo de las estrategias de almacenamiento de calor y equilibrio de carga**.

En [7] se trata la **gestión del consumo de electricidad** en un conjunto de diez edificios mediante el uso del aprendizaje por refuerzo multiagente para lograr una respuesta adaptativa de la demanda en ciudades inteligentes. De esta manera, se busca reducir el coste económico y mejorar la eficiencia energética. Como algoritmo base se utiliza el *RBC*, al igual que en trabajos anteriores, y para aprendizaje por refuerzo se propone *DDPG* y *MADPG*. El estudio concluyó que el enfoque de aprendizaje por refuerzo multiagente es efectivo para lograr una respuesta de demanda adaptativa en edificios inteligentes. Los resultados mostraron que el controlador *MADPG* superó al controlador *DDPG* en términos de costo de electricidad y desviación de tem-

peratura[1] . Además, se encontró que **el uso de múltiples controladores en lugar de uno solo puede mejorar aún más la eficiencia energética y reducir el costo de electricidad**. Sin embargo, el controlador ***RBC*** **optimizado manualmente tuvo un mejor rendimiento que ambos controladores de *RL***. En general, el trabajo demuestra el potencial del aprendizaje por refuerzo multiagente para abordar los desafíos de la gestión de la demanda de energía en ciudades inteligentes.

El proyecto planteado se considera un estudio completo respecto a la literatura que presenta este apartado. Esto se debe a la presencia de los siguientes factores:

- Análisis de los resultados con diferentes métricas.
- Estudio de la convergencia de los algoritmos.
- Experimentación extensa.
- Diferentes escenarios con distintas condiciones.
- Aplicación de algoritmos de *DRL* con enfoque descentralizado independiente.

2.2. Fundamentos teóricos

2.2.1. Inteligencia Artificial

Actualmente, la inteligencia artificial se encuentra en pleno apogeo y es por eso que resulta habitual su constante aparición en los medios de comunicación. A pesar de la cantidad de propaganda diaria que recibimos sobre este tema, la mayor parte de la población desconoce qué es realmente la inteligencia artificial. Esto genera desconfianza y rechazo, en muchas ocasiones, por parte de estos ciudadanos.

Lo primero que debemos saber es que no se trata de un campo de estudio reciente. Su origen se remonta a poco después del fin de la Segunda Guerra Mundial y, concretamente, se le asignó su nombre en 1956. Resulta tan interesante este ámbito por parte de la comunidad científica que, junto a la biología molecular, es donde a la mayoría de científicos de otras disciplinas les gustaría trabajar [2]. La inteligencia artificial es un ámbito interdisciplinar, lo que lo convierte en una opción muy atractiva para los científicos de otros campos. Otro motivo por el que desata gran interés es que todavía hay

[1]La desviación de temperatura es la diferencia entre la temperatura real de un edificio y su temperatura objetivo.

diversos problemas a resolver y un gran potencial por explotar.

La figura 2.2, contiene una tabla con los cuatros enfoques de la IA y algunas definiciones que se han seguido a lo largo de su historia. Las definiciones que aparecen en la parte superior de la tabla hacen referencia al razonamiento y procesos mentales. En cambio, la parte inferior se corresponde con la conducta. Además, tenemos la parte izquierda de la tabla que alude a la forma de actuar del ser humano. Mientras que la parte derecha se basa en la racionalidad. Cuando hablamos de racionalidad, nos referimos a hacer lo correcto en función del conocimiento del que se disponga [2].

A lo largo del tiempo han surgido diversas controversias entre las perspectivas centradas en los humanos y las referentes a la racionalidad. Esto ocurre porque el enfoque humano es fundamentalmente una ciencia empírica que realiza hipótesis y confirmaciones mediante la realización de experimentos. En cambio, el enfoque racional es una mezcla de matemáticas e ingeniería. Por tanto, estos grupos se han apoyado y desaprobado el uno al otro durante las diferentes etapas de la IA [2].

Sistemas que piensan como humanos	**Sistemas que piensan racionalmente**
«El nuevo y excitante esfuerzo de hacer que los computadores piensen… máquinas con mentes, en el más amplio sentido literal». (Haugeland, 1985)	«El estudio de las facultades mentales mediante el uso de modelos computacionales». (Charniak y McDermott, 1985)
«[La automatización de] actividades que vinculamos con procesos de pensamiento humano, actividades como la toma de decisiones, resolución de problemas, aprendizaje…» (Bellman, 1978)	«El estudio de los cálculos que hacen posible percibir, razonar y actuar». (Winston, 1992)
Sistemas que actúan como humanos	**Sistemas que actúan racionalmente**
«El arte de desarrollar máquinas con capacidad para realizar funciones que cuando son realizadas por personas requieren de inteligencia». (Kurzweil, 1990)	«La Inteligencia Computacional es el estudio del diseño de agentes inteligentes». (Poole *et al.*, 1998)
«El estudio de cómo lograr que los computadores realicen tareas que, por el momento, los humanos hacen mejor». (Rich y Knight, 1991)	«IA… está relacionada con conductas inteligentes en artefactos». (Nilsson, 1998)

Figura 2.2: Definiciones de inteligencia artificial organizadas en 4 categorías [2].

2.2.2. Machine learning

El aprendizaje automático (*machine learning*) pertenece al ámbito de la inteligencia artificial. Su principal finalidad es conseguir que las máquinas mejoren en una tarea determinada aprendiendo de los datos. De esta manera se evita tener que codificar reglas explícitamente. En otras palabras, se podría definir el aprendizaje automático como el arte y la ciencia de programar computadoras para que puedan aprender de los datos. A continuación

exponemos una definición general y otra con un enfoque ingenieril [8]:

"*Machine learning* es el campo de estudio que brinda a las computadoras la capacidad de aprender sin ser programadas explícitamente". *Arthur Samuel, 1959*

"Se dice que un programa de computadora aprende de la experiencia E con respecto a alguna tarea T y alguna medida de desempeño P, si su desempeño en T, medido por P, mejora con la experiencia E". *Tom Mitchell, 1997*

Existen diferentes clasificaciones del *ML* dependiendo del criterio que se utilice [8]. A continuación, se exponen diferentes criterios con sus respectivas clasificaciones.

Dependiendo de si hay supervisión humana:

- Aprendizaje supervisado: el conjunto de entrenamiento está etiquetado. Además, se trata de clasificación cuando se predicen valores discretos y regresión cuando se predicen valores continuos.
- Aprendizaje no supervisado: el conjunto de entrenamiento no está etiquetado.
- Aprendizaje semisupervisado: el conjunto de entrenamiento está parcialmente etiquetado (generalmente muchos datos sin etiquetar y algunos pocos datos etiquetados).
- Aprendizaje por refuerzo: El sistema de aprendizaje puede observar el entorno, seleccionar y realizar acciones. A cambio, obtiene recompensas o sanciones. Por tanto, debe aprender por sí mismo cuál es la mejor estrategia para obtener la mayor recompensa a lo largo del tiempo.

Dependiendo de si pueden aprender gradualmente sobre la marcha:

- Batch learning: equivale a cuando el modelo realiza el aprendizaje de manera no incremental.
- Online learning: corresponde cuando el modelo realiza el aprendizaje de manera incremental.

Dependiendo de que funcionen simplemente comparando nuevos puntos de datos con puntos de datos conocidos o que, en su lugar, detecten patrones en los datos de entrenamiento y creen un modelo predictivo, como lo hacen los científicos:

- Basado en instancia: el sistema aprende los ejemplos de memoria, luego generaliza a nuevos casos usando una medida de similitud.
- Basado en modelo: generaliza a partir de un conjunto de ejemplos, construyendo un modelo de estos y luego usa ese modelo para hacer predicciones.

Cabe destacar que la filtración de *spam* fue una de las primeras aplicaciones de *machine learning* que contó con bastante popularidad y tuvo lugar en la década de 1990.

2.2.3. Aprendizaje por refuerzo

Como hemos visto en la sección anterior, el aprendizaje por refuerzo pertenece al campo del aprendizaje automático y al de la inteligencia artificial. Este tipo de aprendizaje es el más peculiar respecto a los otros tres y destaca principalmente por el sistema de recompensas que utiliza.

Al igual que la inteligencia artificial, se suele pensar que el *RL* es un campo reciente pero esto no es así. **El origen de este ámbito se remonta a los años 50 aproximadamente**. En aquel entonces, coexistían dos líneas de investigación que trataban el problema desde distintos enfoques. Una de ellas, estudiaba el aprendizaje mediante el paradigma de ensayo y error. En cambio, la otra vertiente se focalizaba en el problema de control óptimo y su solución mediante funciones de valor y programación dinámica, aunque no involucraba aprendizaje.

En los años 80 se fusionaron las dos vertientes mencionadas, surgiendo así un enfoque más similar al que conocemos hoy en día como aprendizaje por refuerzo. Gracias a los grandes avances que ha experimentado el *hardware* en las últimas décadas, el aprendizaje por refuerzo se encuentra en pleno auge actualmente. Esto se debe a que los avances tecnológicos han posibilitado el uso de técnicas de aprendizaje profundo por refuerzo que antes eran inviables por su elevado coste computacional.

La idea de que aprendemos interactuando con nuestro entorno es probablemente la primera que se nos ocurre cuando pensamos en la naturaleza del aprendizaje. Cuando un bebé juega o mira a su alrededor, no tiene un maestro, pero sí una conexión sensorial y motora directa con su entorno. Esta conexión produce una gran cantidad de información sobre la causa-efecto, las consecuencias de las acciones y sobre qué hacer para alcanzar objetivos. A lo largo de nuestra vida, estas interacciones son sin duda una fuente importante de conocimiento sobre nuestro entorno y sobre nosotros mismos.

Por tanto, somos muy conscientes de cómo responde nuestro entorno a lo que hacemos, y tratamos de influir en lo que ocurre a través de nuestro comportamiento. Concretamente, **el *RL* se basa en la interacción del entorno para entrenar al agente y que este sea capaz de alcanzar un objetivo determinado**. Esto es lo que hace tan singular este tipo de aprendizaje respecto a los otros tipos de aprendizaje del *ML*.

El aprendizaje por refuerzo consiste en aprender qué hacer (cómo asignar situaciones a acciones) para maximizar una señal numérica de recompensa. Al alumno no se le dice qué acciones debe realizar, sino que debe descubrir qué acciones producen la mayor recompensa probándolas. En los casos más interesantes y desafiantes, las acciones pueden afectar no sólo a la recompensa inmediata, sino también a la siguiente situación y, a través de ella, a todas las recompensas posteriores. Estas dos características, la búsqueda por ensayo y error y la recompensa diferida, son los dos rasgos distintivos más importantes del aprendizaje por refuerzo.

Uno de los mayores retos que plantea el aprendizaje por refuerzo, y no en otros tipos de aprendizaje, es el compromiso entre exploración y explotación. Para obtener muchas recompensas, un agente de aprendizaje por refuerzo debe preferir acciones que haya probado en el pasado y que le hayan resultado exitosas. Pero para descubrir esas acciones, tiene que probar acciones que no ha seleccionado antes. El agente tiene que explotar lo que ya ha experimentado para obtener recompensa, pero también tiene que explorar para hacer mejores selecciones de acciones en el futuro. El dilema es que ni la exploración ni la explotación pueden realizarse de forma exclusiva sin fracasar en la tarea. El agente debe probar diversas acciones y favorecer progresivamente las que le parezcan mejores. En una tarea estocástica, cada acción debe probarse muchas veces para obtener una estimación fiable de su recompensa esperada. El dilema exploración-explotación ha sido estudiado intensamente por los matemáticos durante muchas décadas, pero sigue sin resolverse. [9]

Elementos del aprendizaje por refuerzo

Los dos elementos básicos que incluye cualquier problema de *RL* son: el agente y el entorno con el que este interactúa.

El agente es el elemento que se encarga de tomar las decisiones, con la finalidad de aprender un comportamiento orientado a maximizar la señal de recompensa. Este aprendizaje se adquiere mediante la interacción con el entorno. Por lo tanto, en cada instante obtiene información del entorno para determinar:

- El estado en el que se encuentra.

- Una señal de recompensa (sirve para indicar cómo de bueno es ese estado).
- La acción que va a ejecutar en base a la información obtenida.

El entorno se puede definir como: "todo aquello que el agente no puede modificar de forma arbitrario"[2] [9]. Esto también incluiría la función de recompensa, a pesar de que el agente pueda conocerla. La dinámica del entorno define su comportamiento y esta no tiene porqué ser necesariamente conocida por el agente[3]. Si la dinámica cambia a lo largo del tiempo el entorno sería no estacionario, en caso contrario sería estacionario.

Además del agente y el entorno, se pueden identificar cuatro subelementos principales de un sistema de aprendizaje por refuerzo: una política, una señal de recompensa , una función de valor y, opcionalmente, un modelo del entorno.

La política define la forma de actuar del agente en un momento dado. A grandes rasgos, una política es una correspondencia entre los estados percibidos del entorno y las acciones que deben realizarse cuando se está en esos estados. Corresponde a lo que en psicología se conoce como conjunto de reglas o asociaciones estímulo-respuesta. En algunos casos, la política puede ser una simple función o una tabla de consulta. Mientras que en otros casos, puede implicar una gran cantidad de cálculos, como un proceso de búsqueda. La política es el núcleo de un agente de aprendizaje por refuerzo, ya que por sí sola es suficiente para determinar su comportamiento. En general, las políticas pueden ser estocásticas y por tanto, especificar probabilidades para cada acción.

La función de recompensa define el objetivo de un problema de aprendizaje por refuerzo. En cada paso temporal, el entorno envía al agente un único número denominado recompensa. El objetivo del agente es maximizar la recompensa total que recibe a largo plazo. Así pues, la función de recompensa define cuáles son los sucesos buenos y malos para el agente. Además, es la base principal para alterar la política; si una acción seleccionada por la política va seguida de una recompensa baja, entonces la política puede cambiarse para seleccionar alguna otra acción en esa situación en el futuro.

Mientras que la función de recompensa indica lo que es bueno en un sentido inmediato, **la función de valor** especifica lo que es bueno a largo plazo.

[2]La separación entre el entorno y el agente representa el límite del control absoluto del agente, no de su conocimiento.

[3]La dinámica suele ser parcial o totalmente desconocida, sobre todo en los problemas con aplicación en el mundo real.

A grandes rasgos, el valor de un estado es la cantidad total de recompensa que un agente puede esperar acumular en el futuro, partiendo de ese estado. Mientras que las recompensas determinan la conveniencia inmediata e intrínseca de los estados del entorno, los valores indican la conveniencia a largo plazo de los estados después de tener en cuenta los estados que probablemente seguirán y las recompensas disponibles en esos estados. Por ejemplo, un estado puede producir siempre una recompensa inmediata baja, pero tener un valor alto porque está seguido de otros estados que producen recompensas altas. Es importante señalar que esto no resta importancia a la función de recompensa, ya que sin esta sería imposible estimar la función de valor.

El modelo del entorno imita el comportamiento del entorno o permite hacer inferencias sobre cómo se comportará este. Por ejemplo, dado un estado y una acción, el modelo puede predecir el siguiente estado resultante y la siguiente recompensa. Los modelos de entorno se utilizan para planificar, es decir, para decidir una serie de acciones teniendo en cuenta posibles situaciones futuras antes de que se produzcan. Los métodos para resolver problemas de aprendizaje por refuerzo que utilizan modelos y planificación se denominan **métodos basados en modelos**, a diferencia de los métodos más sencillos sin modelos que son explícitamente aprendices de ensayo y error (métodos libres de modelo). Cabe destacar que para la mayoría de problemas del mundo real, no se dispone de un modelo del entorno, por lo que resultan necesarios los métodos libres de modelo.

Otros aspectos

Los problemas en aprendizaje por refuerzo pueden tener dos enfoques diferentes: **tareas de predicción y tareas de control**. El primer enfoque tiene como objetivo predecir la recompensa total esperable al partir de un estado determinado y seguir el comportamiento de la política establecida. En cambio, el segundo enfoque se centra en aproximar una política óptima que maximice la recompensa total esperable de cualquier estado.

Asimismo, en *RL* existen dos tipos de problema en función de como se divida el tiempo de interacción de un agente con su entorno. Concretamente, si se alcanza un estado terminal tras un número finito de pasos, se trata de un **problema episódico**. Mientras que si no se alcanza un estado terminal tras un número de pasos, se trata de un **problema continuado**.

Tras toda esta información, podemos concluir que el problema que se va a trabajar en este proyecto se trata de una tarea de control y un problema episódico.

Capítulo 3

Metodología

3.1. Problema y solución propuesta

El problema propuesto para este trabajo es el **control energético en *smart cities*** mediante el uso de técnicas de aprendizaje por refuerzo profundo. Concretamente, se pretende **reducir las emisiones de carbono y el coste económico**. Hasta ahora, la tendencia es utilizar sistemas basados en reglas (*RBC*) para afrontar este reto. De hecho, se propondrá un modelo base que use dicho algoritmo. El principal inconveniente del *RBC* radica en la complejidad que conlleva su programación y la ineficiencia de su rediseño para realizar cualquier modificación. Tras la resolución de este problema, se espera poder responder las siguientes cuestiones:

- ¿Pueden las técnicas de *DRL* igualar o incluso superar a un *RBC* en términos de efectividad en el control?
- Dentro de los algoritmos de *DRL*, ¿Cuál es superior? ¿En qué aspecto lo es?

Otras preguntas secundarias que surgen son:

- ¿Cuánto tiempo de simulación es necesario para obtener un buen rendimiento con los algoritmos de *DRL*?
- ¿Existen diferencias en la convergencia de los algoritmos de *DRL*?

Para resolver el problema y responder las preguntas se propone un **enfoque multiagente descentralizado independiente** de *DRL*. Esto implica que cada agente se ocupe de las tomas de decisiones de un solo edificio, contemplando únicamente sus propios estados y recompensas individuales. Es decir, no existe la comunicación o interacción entre agentes de manera explícita.

Para tomar las decisiones de cada edificio, los agentes pueden indicar la capacidad de carga o descarga de cada tanque de almacenamiento térmico y batería: refrigeración, calefacción, agua caliente y electricidad. Esto se representa como un número real entre -1 y 1, siendo los valores negativos las descargas y los positivos las cargas. Además, los agentes contemplan la posibilidad de usar energía renovable cuando esté disponible y en caso de que el edificio posea placas fotovoltaicas.

En cuanto a los experimentos, por cada uno de ellos se ejecutan tres algoritmos. Concretamente, ***RBC* se usa como modelo base** mientras que ***SAC* y *MARLISA* corresponden a los algoritmos de los modelos de *DRL*** propuestos. Cada ejecución proporciona los valores de la función de recompensa y de las diferentes funciones de coste de cada modelo. Esto permite realizar un análisis más amplio y robusto, de manera que se pueden corroborar los diferentes resultados obtenidos. Por otro lado, hay que señalar que **se utilizan tres escenarios diferentes** con el fin de añadir variabilidad al estudio y asegurar la adaptabilidad de los modelos a nuevos entornos con diferentes características. Con esto aseguramos que haya una buena capacidad de generalización por parte de los modelos.

Cabe destacar que un episodio representa un año completo, por lo que cada modelo ha aprendido a tratar el control energético con diferentes estaciones, climas y fenómenos meteorológicos. Concretamente, para cada modelo se han tratado los siguientes números de episodios: 3, 5, 10, 15, 20 y 25.

En resumen, se han realizado 39 experimentos: 2 algoritmos (*DRL)* x 3 escenarios x 6 longitudes de episodio + 1 *RBC* x 3 escenarios. Nótese que cada experimento corresponde a la combinación de un algoritmo, con un escenario y un número de episodios concreto. Para el entrenamiento de cada modelo se aplica el número de episodios seleccionados, mientras que la evaluación siempre se hace con un episodio. A pesar, de que *RBC* se ejecuta 6 veces por cada escenario, solo se cuenta como un experimento ya que sus condiciones no cambian. Esto se hace así por la implementación de *Autogrid*.

3.2. Software

Para la realización de este trabajo se han utilizado principalmente dos herramientas de software: ***CityLearn*** y ***Autogrid***.

CityLearn [3] es un entorno *OpenAI Gym* de código abierto para, principalmente, la implementación del aprendizaje por refuerzo multiagente destinado al control energético en ciudades [23, 25]. Un reto para *RL* en el control energético es la capacidad de comparar el rendimiento de los algorit-

mos [27]. Por lo tanto, esta herramienta posibilita y estandariza la evaluación de agentes de *RL*, de manera que diferentes algoritmos puedan ser fácilmente comparados entre sí.

Las ciudades y distritos tienen periodos de alta demanda energética que elevan el precio de la electricidad y el coste general de las redes de distribución de energía. Mediante el control energético se puede ajustar la demanda energética y como consecuencia reducir el coste de la generación, trasmisión y distribución eléctrica. *CityLearn* proporciona un entorno que permite la implementación del aprendizaje por refuerzo con uno o varios agentes encargados del control energético.

La herramienta *CityLearn*, ha sido utilizada en diversos proyectos y aplicaciones:

- Benchmarking para algoritmos de *RL* [10, 11, 12, 13, 14].
- Gestión de energía coordinada [15, 16, 17, 18, 19, 5, 20, 21].
- Respuesta-demanda basa en incentivos [22, 23].
- Gestión independiente de la energía. [24, 25].
- Meta-learning [26].
- Model predictive control [27].
- Transfer learning [28].
- Regulación de voltaje [29].

Autogrid es una herramienta en desarrollo por la Universidad de Granada. Su principal objetivo es ofrecer un entorno para facilitar la implementación de aprendizaje por refuerzo y control energético a dos niveles: regional y ciudad. Para el primer caso, se basa en la herramienta *Grid2Op* y para el segundo *CityLearn*. *Autogrid* facilita la automatización de grandes baterías de experimentos. Otro de sus objetivos es funcionar como intermediario entre *CityLearn*, *Grid2OP* y el usuario, de tal manera que sea más amigable para este.

Por último, cabe añadir que todo el código con las modificaciones necesarias para llevar acabo la experimentación propuesta se encuentra en este repositorio de github.

3.3. Formulación del problema

En esta sección vamos a describir los elementos básicos del aprendizaje por refuerzo en función de nuestro problema concreto. Cada subsección corresponderá a un elemento diferente que se explicará en detalle.

3.3.1. Entorno

CityLearn incluye modelos energéticos de edificios y recursos energéticos distribuidos (*DER*), como bombas de calor aire-agua, calentadores eléctricos y baterías. Una colección de modelos energéticos de edificios constituye un distrito virtual (también conocido como barrio o comunidad). En cada edificio, la refrigeración, la calefacción y el agua caliente pueden satisfacerse independientemente mediante bombas de calor aire-agua. Alternativamente, la calefacción y el agua caliente pueden satisfacerse mediante calentadores eléctricos.

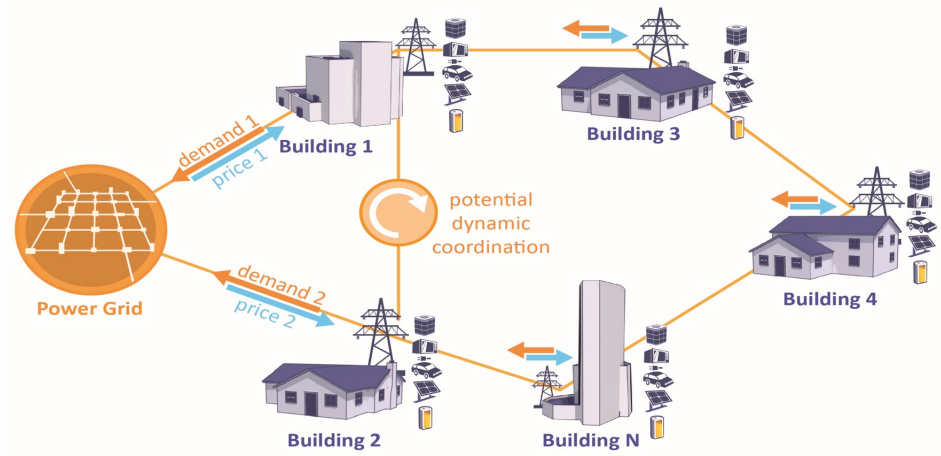

Figura 3.1: Ejemplo de una ciudad en cityLearn [3].

Los edificios pueden tener una combinación de tanques de almacenamiento térmico y baterías para almacenar la energía que se puede utilizar en periodos punta o más caros para satisfacer la refrigeración de espacios, la calefacción de espacios, el agua caliente y las cargas no desplazables (enchufes). Estos dispositivos de almacenamiento son cargados por el dispositivo eléctrico (bomba de calor o calentador eléctrico) que satisface el uso final para el que se destina la energía almacenada. Todos los dispositivos eléctricos, así como los enchufes, consumen electricidad de la red principal. En los edificios pueden instalarse paneles fotovoltaicos para compensar total o parcialmente el consumo de electricidad de la red, permitiendo a los edificios generar su

propia electricidad.

Los agentes *RBC*, *RL* o *MPC* controlan los tanques de almacenamiento térmico y las baterías determinando cuánta energía almacenar o liberar en cada momento. *CityLearn* garantiza que, en cualquier momento, la refrigeración, la calefacción, el agua caliente y las cargas no variables de los edificios se satisfacen independientemente de las acciones del controlador, utilizando la demanda precalculada o medida previamente de los edificios. Un controlador interno de reserva garantiza que los dispositivos eléctricos den prioridad a la satisfacción de las cargas del edificio antes de almacenar energía en los dispositivos de almacenamiento. El controlador de respaldo también garantiza que los dispositivos de almacenamiento no descarguen más energía de la necesaria para satisfacer las necesidades de los edificios.

3.3.2. Estados

Los estados de *CityLearn* **se agrupan en categorías de calendario, tiempo, distrito y edificio**. Los valores de estado de las categorías calendario, tiempo y distrito son iguales para todos los edificios del entorno, mientras que los estados de la categoría edificio son específicas de cada edificio. Los estados pueden calcularse, simularse o medirse previamente y suministrarse al entorno a través de archivos .csv planos. Otros estados dependen de las acciones realizadas por los agentes, por lo que se calculan durante el tiempo de ejecución de la simulación.

3.3.3. Acciones

Cada acción es un número real entre [-1.0, 1.0] que indica la proporción de la capacidad de un dispositivo de almacenamiento que debe cargarse o descargarse:

- Almacenamiento de refrigeración.
- Almacenamiento de calefacción.
- Almacenamiento agua caliente.
- Almacenamiento de electricidad.

3.3.4. Función de recompensa

La función de recompensa que se ha definido para este problema es la siguiente:

$$min(-(e \cdot \lambda_e \ + \ p \cdot \lambda_p), \ \ 0)$$

Siendo:

- e las emisiones de carbono.
- p el precio de la electricidad.
- λ el factor de normalización para que las emisiones de carbono y el precio estén en una misma escala.

Dada la implementación de los algoritmos, estos tienden siempre a maximizar la recompensa. Teniendo en cuenta que el objetivo final es minimizar las emisiones de carbono y el precio de la electricidad, la función de recompensa utilizada es la suma de estas dos componentes, con el signo invertido. Téngase en cuenta que se escoge el mínimo entre el valor obtenido y 0, porque no se puede reducir el precio de la electricidad o las emisiones de carbono más de 0.

3.3.5. Agentes

Para este problema se sigue un **enfoque multiagente**, concretamente **descentralizado independiente**. Por lo que cada agente del sistema toma decisiones basándose únicamente en sus propias observaciones y recompensas locales.

Por supuesto, cada agente se encarga de un edificio y no existe comunicación o interacción directa entre ellos. De hecho, se supone que cualquier interacción que ocurra entre estos, es resultado de la dinámica del entorno.

El objetivo de los agentes es maximizar su propia recompensa considerando sus propias acciones y estados. *Autogrid* y *CityLearn* permiten los siguientes algoritmos para el enfoque multiagente descentralizado indepen-dinete y son los que se han utilizado:

- *SAC*.
- *MARLISA*.
- *RBC*.

3.3.6. Funciones de coste

CityLearn proporciona un conjunto de funciones de coste (indicadores clave de rendimiento) que pueden utilizarse para cuantificar el rendimiento energético, medioambiental y económico de los edificios o distritos tras la simulación. Entre las funciones implementadas se incluyen las siguientes:

- Factor de carga.
- Pico medio diario.

- **Emisiones de carbono**.
- **Precio**.
- Consumo neto de electricidad.
- Pico de demanda.
- *Ramping*
- *Zero net energy*

Para nuestro proyecto solo resultan de interés las **emisiones de carbono** y el **precio de la electricidad**. Pero almacenamos también los demás valores para posibles futuros trabajos y una mayor aportación a la comunidad científica.

3.4. Algoritmos

3.4.1. SAC

El algoritmo ***Soft Actor Critic*** (SAC) pertenece a la familia ***actor-critic*** y se trata de uno de los algoritmos más populares del ámbito del aprendizaje por refuerzo profundo. Además, es un algoritmo ***off-policy***[1] que está basado en *Q-learning* y tiene la capacidad de optimizar políticas estocásticas [30].

Este algoritmo destaca por utilizar una función objetivo modificada, donde introduce el concepto de **entropía**, que tratará de maximizar junto a la función de recompensa. La entropía mide la aleatoriedad o impredecibilidad de una variable. Es decir, si siempre toma el mismo valor, su entropía será nula, y viceversa. De esta manera, podemos decir que la entropía consiste en una medida de aleatoriedad en el comportamiento del agente. *SAC* trata de lidiar con el problema de explotación y exploración maximizando dicha entropía. Por tanto, la función objetivo es la siguiente:

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}[r(s_t,a_t) + \alpha\mathcal{H}(\pi(\cdot\,|s_t))] \tag{3.1}$$

Además, *SAC* utiliza **tres redes neuronales**: una para aproximar la función estado-valor V parametrizada por ψ, otra para la función *soft Q* (Q(θ)) y otra función $\pi(\phi)$ para la política. Aunque el uso de aproximadores

[1]Hace referencia a aquellos metodos que utilizan dos políticas diferentes. Una se usa para aproximar y aprender de manera iterativa. Mientras que la otra, se utiliza para elegir acciones durante el aprendizaje e interaccionar con el entorno. En cambio, los métodos on-policy utiliza una sola política para dichas tareas

diferentes para V y Q no es estrictamente necesario, los autores sostienen que de esta manera en la práctica se puede conseguir una mejora [30]. A continuación, se muestra cómo se entrenan las redes:

- Para aproximar V, se minimiza la siguiente función de pérdida:

$$J_V(\psi) = \mathbb{E}_{s_t \sim \mathcal{D}}\left[\frac{1}{2}(V_\psi(s_t) - \mathbb{E}_{a_t \sim \pi_\phi}[Q_\theta(s_t, a_t) - \log \pi_\phi(a_t|s_t)])^2\right]$$

  La actualización se formula así:

$$\hat{\nabla}_\phi J_V(\psi) = \nabla_\psi V_\psi(s_t)(V_\psi(s_t) - Q_\theta(s_t, a_t) + \log \pi_\phi(a_t|s_t))$$

- La función Q se entrena minimizando el error:

$$J_Q(\theta) = \mathbb{E}_{s_t \sim \mathcal{D}}\left[\frac{1}{2}\left(Q_\theta(s_t, a_t) - (r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1} \sim p}[V_\psi(s_{t+1})])\right)^2\right]$$

  Con la actualización:

$$\hat{\nabla}_\theta J_Q(\theta) = \nabla_\theta Q_\theta(s_t, a_t)(Q_\theta(s_t, a_t) - r(s_t, a_t) - \gamma V_\psi(s_{t+1}))$$

- Para la red de la política, se considera la siguiente función de error[2]:

$$J_\pi(\phi) = \mathbb{E}_{s_t \sim \mathcal{D}}\left[D_{KL}\left(\pi_\phi(\cdot\,|s_t)\middle|\middle|\frac{\exp(Q_\theta(s_t, \cdot))}{Z_\theta(s_t)}\right)\right]$$

  Siendo en este caso la actualización:

$$\hat{\nabla}_\phi J_\pi(\phi) = \nabla_\phi \log \pi_\phi(a_t|s_t) + (\nabla_{a_t} \log \pi_\phi(a_t|s_t) - \nabla_{a_t} Q(s_t, a_t))\nabla_\phi f_\phi(\epsilon_t; s_t)$$

Los autores del algoritmo hicieron diferentes experimentos con diversos *benchmarks*, demostrando de manera empírica que este método era superior a las propuestas que presentaba el estado del arte en aquel momento. Por último, se añade el pseudocódigo de *SAC*:

[2]El término D_{KL} hace referencia a la divergencia de Kullback-Leibler [31].

Algoritmo 1: Soft Actor Critic (SAC)

Entrada: - θ: parámetros iniciales de la política
- ϕ_1, ϕ_2: parámetros de la Q-función
- $\mathcal{D}$: *buffer* de repetición

Asignar a los parámetros objetivo el valor de los parámetros principales: $\theta_{obj} \leftarrow \theta, \phi_{obj,1} \leftarrow \phi_1, \phi_{obj,2} \leftarrow \phi_2$
repetir
  Observar el estado s y seleccionar la acción $a \sim \pi_\theta(\cdot|s)$
  Ejecutar a
  Observar el siguiente estado s', recompensa r, y si es estado terminal (flag d)
  Almacenar (s, a, r, s', d) en el búfer de repetición ⌈
  Si s' es terminal, se resetea el estado del entorno
  si *se debe actualizar* **entonces**
    para cada $j \in [0, \textit{número actualizaciones})$ **hacer**
      Extraer un batch de transiciones B aleatorio
      Calcular los objetivos para las Q-funciones: $y(r, s', d) = r + \gamma(1-d)\left(min_{i=1,2} Q_{\phi_{obj,i}}(s', \tilde{a}') - \alpha \log \pi_\theta(\tilde{a}'|s')\right)$
      Actualizar las Q-funciones con un paso del gradiente descendente: $\nabla_{\phi_i} \frac{1}{|B|} \sum_{(s,a,r,s',d) \in B} (Q_{\phi_i}(s,a) - y(r,s',d))^2$ para $i = 1, 2$
      Actualizar la política con un paso del gradiente ascendente: $\nabla_\phi \frac{1}{|B|} \sum_{s \in B} (\text{mín}_{i=1,2}\, Q_{\phi_i}(s, \tilde{a}_\theta(s)) - \alpha \log \pi_\theta(\tilde{a}_\theta(s)|s)$
      donde $\tilde{a}_\theta(s)$ es una muestra de $\pi_\theta(\cdot|s)$, que es diferenciable con respecto a θ usando el truco de la reparametrización.
      Actualizar las redes objetivo con: $\phi_{obj,i} \leftarrow \rho\phi_{obj,i} + (1-\rho)\phi_i \quad \text{para } i = 1, 2$
    fin
mientras *no converja*

3.4.2. MARLISA

MARLISA es un algoritmo creado por los autores de *CityLearn* en la propia herramienta. Esta técnica es una extensión de *SAC* que permite la coordinación de los agentes mediante el reparto de recompensas, así como el intercambio mutuo de información. Para coordinarse, cada agente sólo necesita compartir dos variables con otro, lo que hace que el algoritmo sea escalable, ya que el número de variables que necesita cada agente no aumenta con el número de agentes. *MARLISA* puede aplicarse tanto a problemas con

un enfoque multiagente descentralizado independiente como descentralizado cooperativo. [5]. La implementación de *MARLISA* en *CityLearn* se muestra en la figura 3.2.

MARLISA se basa en el algoritmo de aprendizaje por refuerzo asimétrico multiagente (*AMRL*), que se ha adaptado para el problema de la formación de carga de energía. Este enfoque utiliza una función de recompensa con objetivos individuales y colectivos, y los agentes predicen su propio consumo de electricidad futuro y comparten esta información entre sí siguiendo un esquema de líder-seguidor.

Además, *MARLISA* utiliza un enfoque de selección de acción secuencial iterativa para coordinar las acciones de los agentes. Este enfoque permite a estos tomar decisiones basadas en la información que han compartido entre sí, lo que les permite coordinar su consumo de energía de manera efectiva. Además, este algoritmo permite la escalabilidad y descentralización del control de la energía, lo que lo hace adecuado para su uso en sistemas de energía de gran escala.

En la figura 3.3 se puede apreciar el pseudocódigo de *MARLISA* que se explica detalladamente en [5].

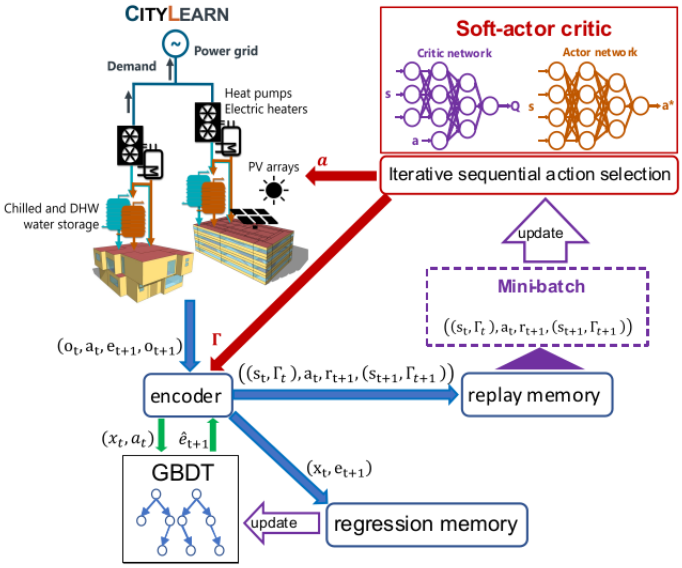

Figura 3.2: Implementación de *MARLISA* en *CityLearn*.

Algorithm 1: Action selection of MARLISA

Initialize: $\mathbf{k} \leftarrow 1$, $\hat{\mathbf{e}}_i^{k=0} \leftarrow 0 \; \forall \, i \in B = \{0, 1, \ldots, n\}$
$\hat{\mathbf{e}}^{cum} \leftarrow 0$
$\mathbf{s}_i \leftarrow \mathbf{f}_i^{\text{state encoder}}(\mathbf{o}_i)$, $\mathbf{x}_i \leftarrow \mathbf{f}_i^{\text{GBDT encoder}}(\mathbf{o}_i) \; \forall \, i \in B$
while $k \leq m$ **do**
  $i \leftarrow 0$; $\mathbf{c}^{cum} \leftarrow 0$
  for all agents $i \in B$ **do**
    $\mathbf{S}_{ext} \leftarrow (\mathbf{s}_i, \mathbf{c}^{cum}, \hat{\mathbf{e}}^{cum})$
    $\mathbf{a}_i^k \leftarrow \mathbf{SAC}_i\_\mathbf{ActionSelection}(\mathbf{S}_{ext})$
    $\hat{\mathbf{e}}_i^k \leftarrow \mathbf{GBDT}_i\_\mathbf{Predict}(\mathbf{x}_i, \mathbf{a}_i^k)$
    $\mathbf{c}_{i+1}^{cum} \leftarrow \mathbf{c}_i^{cum} + \mathbf{c}_i$
    if $i < n - 1$ **then**
      $\hat{\mathbf{e}}_{i+1}^{cum} \leftarrow \hat{\mathbf{e}}_i^{cum} + \hat{\mathbf{e}}_i^k - \hat{\mathbf{e}}_{i+1}^{k-1}$
    else
      if $i < m$ **then**
        $\hat{\mathbf{e}}_0^{cum} \leftarrow \hat{\mathbf{e}}_i^{cum} + \hat{\mathbf{e}}_i^k - \hat{\mathbf{e}}_0^{k-1}$
      else
        $\mathbf{A} \leftarrow (\mathbf{a}_0^k, \mathbf{a}_1^k, \ldots, \mathbf{a}_n^k)$
        $\Gamma \leftarrow$
  $k \leftarrow k + 1$
return A, Γ

Figura 3.3: Pseudocódigo de *MARLISA*.

Capítulo 4

Experimentos y discusión

4.1. Experimentación

Con la finalidad de realizar un estudio completo del problema planteado, **se han realizado 39 experimentos**: 2 algoritmos (*DRL)* x 3 escenarios x 6 longitudes de episodio + 1 *RBC* x 3 escenarios. Nótese que cada experimento corresponde a la combinación de un algoritmo, con un escenario y un número de episodios concreto. Para el entrenamiento de cada modelo se aplica el número de episodios seleccionados, mientras que la evaluación siempre se hace con un episodio. A pesar, de que *RBC* se ejecuta 6 veces por cada escenario, solo se cuenta como un experimento ya que sus condiciones no cambian. Esto se hace así por la implementación de *Autogrid*. Aun así, más adelante, cuando se calcula la mejora en porcentaje de los modelos de *DRL* respecto a *RBC*, se calcula la media de todas las ejecuciones realizadas, en cada escenario, para el modelo base. De esta manera, se intenta evitar el ruido del *baseline*.

Se ejecutan tres algoritmos, donde ***RBC* se usa como modelo base** mientras que ***SAC* y *MARLISA* corresponden a los algoritmos de los modelos de *DRL*** propuestos. Cada ejecución proporciona los valores de la función de recompensa y de las diferentes funciones de coste de cada modelo. Esto permite realizar un análisis más amplio y robusto, de manera que se pueden corroborar los diferentes resultados obtenidos.

Por otro lado, hay que señalar que **se utilizan tres escenarios diferentes** con el fin de añadir variabilidad al estudio y asegurar la adaptabilidad de los modelos a nuevos entornos con diferentes características. Con esto aseguramos que haya una buena capacidad de generalización por parte de los modelos.

Cada escenario tiene una estructura diferente, por lo que el número de

edificios varía. Los escenarios 1 y 2 tienen cinco edificios, mientras que el escenario 3 se compone de siete edificios.

Cabe destacar que un episodio representa un año completo, por lo que cada modelo ha aprendido a tratar el control energético con diferentes estaciones, climas y fenómenos meteorológicos. Concretamente, para cada modelo se han tratado los siguientes números de episodios: 3, 5, 10, 15, 20 y 25.

4.2. Síntesis de los resultados

A continuación, se presentan dos subsecciones donde se mostrará los resultados de la función de recompensa y las funciones de coste. Esto irá acompañado de un análisis y estudio completo que determinará cuál es el mejor modelo y otras cuestiones interesantes que serán de gran utilidad para la extracción de conclusiones de este trabajo.

4.2.1. Función de recompensa

En las figuras 4.1, 4.2 y 4.3, se muestran los valores de la función de recompensa media en función del número de episodios por cada algoritmo de *DRL*. Hay tres imágenes porque cada una corresponde a un escenario diferente. De estos gráficos se pueden extraer algunas ideas interesantes. Cabe destacar que el algoritmo *RBC* no aparece representado porque no le afecta el número de episodios, es decir, no cuenta con este parámetro.

En los tres escenarios, se puede apreciar que ***MARLISA* converge antes que *SAC***. De hecho, si observamos la evolución de cada algoritmo, apreciamos que **con *MARLISA* los valores obtenidos no varían mucho del primer al último experimento en un mismo escenario**. Mientras que con *SAC* ocurre lo contrario, es decir, hay una gran diferencia al usar 3 episodios frente a utilizar 25. Si a esto le añadimos que **inicialmente *SAC* parte con un mejor rendimiento, esta diferencia se hace aún más notable con el aumento de episodios**. Por lo que se ve que *MARLISA* aporta peores resultados. Esto no quiere decir que sean necesariamente malos. A continuación, compararemos los mejores modelos de *DRL* con el modelo base que usa *RBC* para ilustrar mejor el rendimiento obtenido.

La tabla 4.1 recoge las mejores recompensas medias obtenidas por cada algoritmo y escenario. Es decir, selecciona el mejor modelo para cada algoritmo por escenario. Como se ha mencionado anteriormente el mejor rendimiento de los modelos que usan *DRL* coinciden con el último experimento para cada escenario, lo que sugiere que los algoritmos convergen con 25 episodios. Los resultados recogidos en la tabla para *RBC*, son la media de

todas las ejecuciones en cada ciudad, ya que en este algoritmo no interfiere el número de episodios.

En la tabla 4.1, se aprecia que los modelos que usan el algoritmo *SAC* tienen un mejor desempeño respecto a los demás en las tres ciudades. Después le sigue *Marlisa*, marcando así una clara predominancia de los modelos de *DRL* frente al modelo base.

La tabla 4.2, contiene el porcentaje que cuantifica la mejoría de los mejores modelos de *DRL* respecto al modelo base *RBC* en cada escenario. Resulta llamativo que **los modelos con *SAC* son entorno al 50 % mejores que el modelo base de cada ciudad**, siendo 52'53 %, 51'74 % y 45'04 % la mejoría respectivamente. Esto señala que se mejora entorno al 50 % el conjunto del ahorro económico y la reducción de emisiones de carbono. Por otro lado, **los modelos con *MARLISA* son entorno al 36 % mejores que el modelo base de cada ciudad**, siendo 38'99 %, 35'33 % y 34'22 % la mejoría respectivamente en cada escenario. Por lo tanto, *MARLISA* también aporta una mejora notable, aunque no tenga tan buen rendimiento como *SAC*.

	Escenario 1	**Escenario 2**	**Escenario 3**
SAC	-0'235	-0'250	-0'310
MARLISA	-0'302	-0'335	-0'371
RBC	-0'495	-0,518	-0'564

Tabla 4.1: Mejores recompensas medias obtenidas por algoritmo y escenario.

	Escenario 1	**Escenario 2**	**Escenario 3**
SAC	52,53 %	51'74 %	45'04 %
MARLISA	38,99 %	35'33 %	34'22 %

Tabla 4.2: Mejores porcentajes obtenidos en función de la recompensa de cada algoritmo respecto a *RBC* en cada escenario.

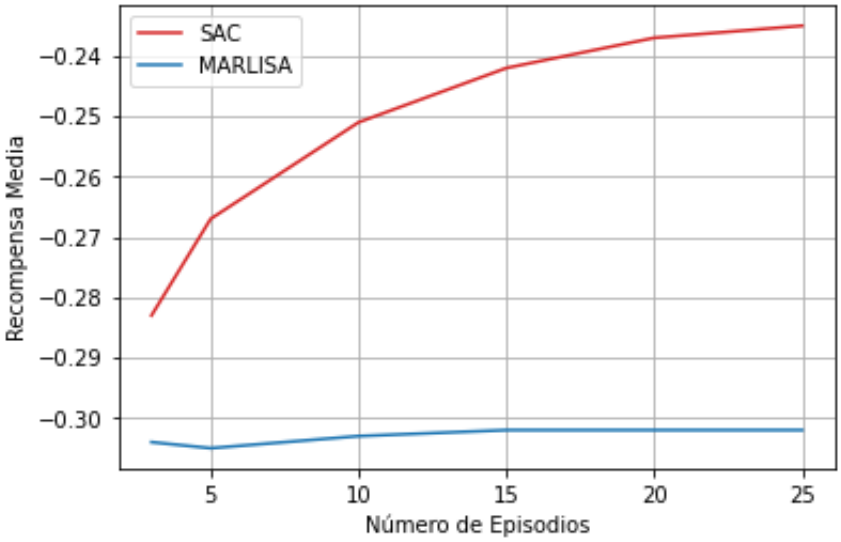

Figura 4.1: Convergencia de los algoritmos de DRL en el escenario 1.

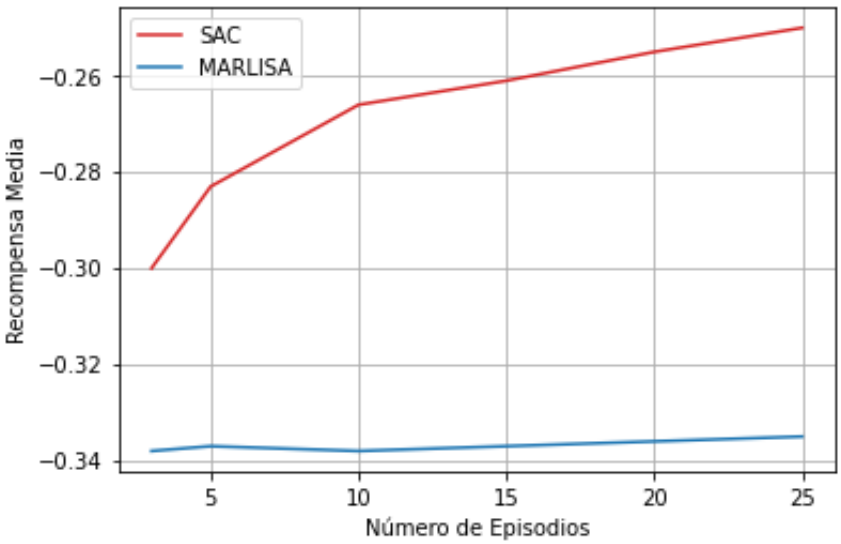

Figura 4.2: Convergencia de los algoritmos de DRL en el escenario 2.

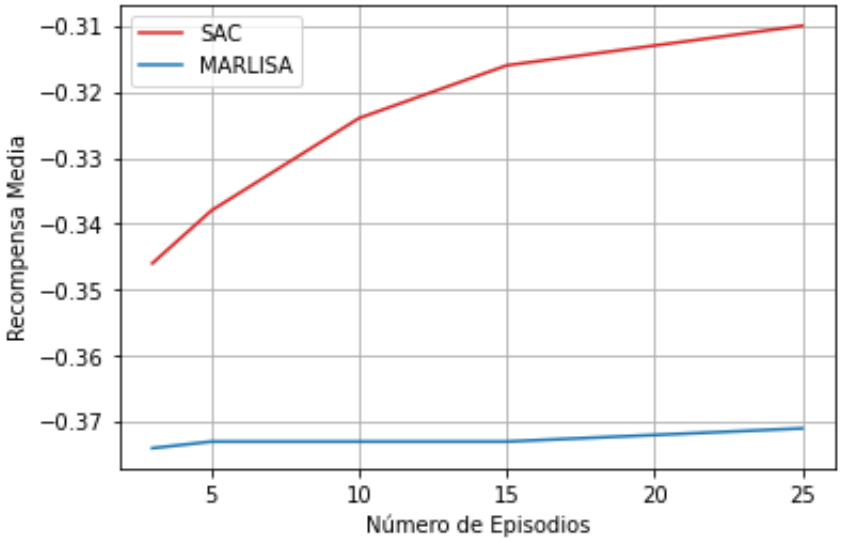

Figura 4.3: Convergencia de los algoritmos de DRL en el escenario 3.

4.2.2. Funciones de coste

En este subapartado se van a analizar las funciones de coste referentes a las emisiones de carbono y al coste económico energético. Esto se hace ya que **si únicamente se utiliza la función de recompensa para el análisis, no se puede diferenciar si existe un mejor modelo en cada aspecto de forma individual**. De esta manera, obtendremos un estudio más completo y amplio del que podremos sacar más conclusiones.

En la tabla 4.3, se muestran los mejores valores para la función de coste referente a las emisiones de carbono obtenidas por cada algoritmo y escenario. En cada ciudad, los modelos con *SAC* vuelven a tener un mejor rendimiento. Además, *MARLISA* también supera al modelo base. La mejoría de los modelos de *DRL* sobre el modelo base, se puede apreciar mejor en la tabla 4.4, ya que contiene los mejores porcentajes obtenidos en función de las emisiones de carbono de cada algoritmo respecto a *RBC* en cada ciudad. **Los modelos con *SAC* son entorno a un 48 % mejores que el modelo base de cada ciudad**, siendo 49'79 %, 50'43 % y 42'51 % la mejoría respectivamente. Por otro lado, **los modelos con *MARLISA* son entorno al 39 % mejores que el modelo base de cada ciudad**, siendo 41'02 %, 38'68 % y 36'71 % la mejoría respectivamente en cada escenario. Por tanto, podemos afirmar que **el mejor modelo con *SAC*, en cada escenario, tiene un mejor desempeño en cuanto a la reducción de emisiones de carbono respecto a los otros modelos**. Ahora vamos a comprobar si esto es así también para el coste económico de la energía.

De manera análoga a como hemos realizado el análisis de la función de coste de las emisiones de carbono, vamos ahora a hacerlo para la función de coste referente al coste económico. La tabla 4.5 recoge los mejores valores para el coste económico obtenido por cada algoritmo en cada escenario. De nuevo, ocurre que en cada ciudad los modelos que usan *SAC*, tienen mejores resultados. De hecho, *MARLISA* también vuelve a superar al modelo base. Para apreciar la mejoría de los modelos de *DRL* respecto al modelo base de cada ciudad, podemos fijarnos en la tabla 4.6 que expresa dicha mejora en porcentajes. Al analizar esta tabla, se observa que **los modelos con *SAC* son entorno a un 54 % mejores que el modelo base de cada ciudad**, siendo 55'11 %, 55'62 % y 49'72 % la mejoría respectivamente. En cambio, **los modelos con *MARLISA* son entorno al 38 % mejores que el modelo base de cada ciudad**, siendo 39'48 %, 36'91 % y 38'91 % la mejoría respectivamente en cada escenario. Tras esto, confirmamos que **el mejor modelo con *SAC*, en cada escenario, tiene un mejor desempeño en cuanto a la reducción de costes económicos respecto a otros modelos**.

En el subapartado anterior podíamos afirmar, gracias a la función de recompensa, que cuando un modelo era mejor que otro, lo era respecto al conjunto del ahorro económico y la reducción de emisiones de carbono. Tras realizar el análisis de esta subsección, podemos concluir que **los mejores modelos de *DRL* superan también al modelo base de cada ciudad en cada uno de estos aspectos por separado: ahorro económico y reducción de emisiones de carbono**. Esto también ocurre entre los mejores modelos de *DRL*. ***SAC* también es superior en ambas aristas, por separado, a *MARLISA* en cada escenario**.

	Escenario 1	**Escenario 2**	**Escenario 3**
SAC	0'841	0'801	0'902
MARLISA	0'988	0'991	0'993
RBC	1'675	1'616	1'569

Tabla 4.3: Mejores valores de emisiones de carbono obtenidas por algoritmo y escenario.

	Escenario 1	**Escenario 2**	**Escenario 3**
SAC	49'79 %	50'43 %	42'51 %
MARLISA	41'02 %	38'68 %	36'71 %

Tabla 4.4: Mejores porcentajes obtenidos con la función de coste referente a las emisiones de carbono de cada algoritmo respecto a *RBC* en cada escenario.

	Escenario 1	**Escenario 2**	**Escenario 3**
SAC	0'73	0'695	0'814
MARLISA	0'984	0'988	0'989
RBC	1'626	1'566	1'619

Tabla 4.5: Mejores valores de costes económicos obtenidos por algoritmo y escenario.

	Escenario 1	**Escenario 2**	**Escenario 3**
SAC	55'11 %	55'62 %	49'72 %
MARLISA	39'48 %	36'91 %	38'91 %

Tabla 4.6: Mejores porcentajes obtenidos con la función de coste referente a los costes económicos de cada algoritmo respecto a *RBC* en cada escenario.

Capítulo 5

Conclusión

El control energético en ciudades inteligentes es un problema complejo que cada vez cobra más relevancia en nuestra sociedad. Este ámbito, tiene bastante potencial a desarrollar gracias a los avances tecnológicos que han surgido en las últimas décadas. A pesar de esto, no abunda el uso de técnicas de aprendizaje por refuerzo profundo en este campo. Aunque, es cierto que en los últimos años se ha experimentado un incremento en cuanto al número de estudios que introducían estas técnicas para el control energético en *smart cities*.

Recordemos que la incesante preocupación por el medio ambiente está más que justificada, ya que fenómenos como el aumento de la contaminación, el calentamiento global o el cambio climático pueden acarrear consecuencias irreversibles en el futuro. Si tenemos en cuenta que **en 2021 el sector de los edificios y la construcción presentó, aproximadamente, el 37 % de las emisiones de CO2 relacionadas con la energía y los procesos**, nos percatamos de la relevancia que puede llegar a tener el control energético en ciudades inteligentes. Además, ese mismo año se alcanzó un máximo histórico de 10GtCO2 en las emisiones de CO2 referentes al sector de los edificios. Lo que confirma la necesidad de nuevas propuestas para el control energético en este ámbito.

Por otro lado, el tema económico en lo referente al consumo energético también ha sufrido consecuencias en los últimos años. Factores como la guerra de Ucrania y la consiguiente crisis energética en Europa, ha aumentado considerablemente el precio de la energía. Esto se puede ver reflejado, por ejemplo, en la factura de la luz que en España se incrementó más del 67 % en 2022, según la web tarifasgasluz.

El proyecto realizado se considera un estudio inicial, aunque completo respecto a la literatura que presenta el estado del arte. Esto se debe a la

presencia de los siguientes factores:

- Análisis de los resultados con diferentes métricas.
- Estudio de la convergencia de los algoritmos.
- Experimentación extensa.
- Diferentes escenarios con distintas condiciones.
- Aplicación de algoritmos de *DRL* con enfoque descentralizado independiente.

Con la realización de este trabajo **se ha demostrado que el aprendizaje por refuerzo profundo es una herramienta útil, prometedora y adecuada para la reducción de las emisiones de carbono y el costo energético**. Además, se han obtenido respuestas para las preguntas planteadas en el apartado de metodología:

- ¿Pueden las técnicas de *DRL* igualar o incluso superar a un *RBC* en términos de efectividad en el control?
- Dentro de los algoritmos de *DRL*, ¿Cuál es superior? ¿En qué aspecto lo es?
- ¿Cuánto tiempo de simulación es necesario para obtener un buen rendimiento con los algoritmos de *DRL*?
- ¿Existen diferencias en la convergencia de los algoritmos de *DRL*?

El estudio realizado con la función de recompensa nos revela que, aproximadamente, ***SAC* mejora un 50 %** respecto al modelo base de cada ciudad. Esto significa que de media se mejora entorno al 50 % el conjunto del ahorro económico y la reducción de emisiones de carbono. En cambio, ***MARLISA* mejora entorno a un 36 %** respecto al *baseline* de cada escenario. Esto responde a la primera pregunta: **las técnicas de *DRL* superan al *RBC* de cada ciudad**.

Si únicamente se analiza la función de recompensa, no se puede determinar si existe un mejor modelo para la reducción de las emisiones de carbono o el ahorro económico de manera individual. Por tanto, vamos a estudiar los resultados obtenidos mediante el uso de las funciones de coste respectivas a ambos aspectos.

Los resultados referentes a la función de coste de las **emisiones de carbono**, presentan que ***SAC* mejora entorno a un 48 %** respecto al modelo

base de cada escenario. Mientras que ***MARLISA*** **mejora aproximadamente un 39 %** respecto al *baseline* de cada ciudad. Por lo que podemos afirmar que ***SAC*** **tiene el mejor desempeño en cuanto a la reducción de emisiones de carbono** comparado con los otros algoritmos.

Por otro lado, si estudiamos la función de coste respectiva al **costo económico**, vemos que, aproximadamente, ***SAC*** **mejora un 54 %** respecto al *baseline* de cada ciudad. En este caso, ***MARLISA*** **mejora entorno al 38 %** respecto al modelo base de cada escenario. Por lo que confirmamos que también ***SAC*** **tiene el mejor rendimiento en cuanto a la reducción de costos económicos**.

Anteriormente afirmamos, gracias a la función de recompensa, que cuando un modelo era mejor que otro, lo era respecto al conjunto del ahorro económico y la reducción de emisiones de carbono. Tras analizar individualmente las funciones de coste, podemos concluir que **los modelos de *DRL* superan al *RBC* de cada ciudad en: ahorro económico y reducción de emisiones de carbono por separado**. Al igual ocurre entre los mejores modelos de *DRL*. Es decir, ***SAC*** **es superior en ambas aristas, por separado, a *MARLISA*** en cada escenario.

Para responder la tercera y cuarta pregunta, es necesario analizar la convergencia de los algoritmos de *DRL*. Esto se hace a partir de los resultados obtenidos mediante la función de recompensa en los diferentes experimentos. Lo primero que que se puede destacar es que ***MARLISA*** **converge antes que *SAC*** (véase figura 4.1, 4.2 y 4.3). Resulta llamativo que **con *MARLISA* los resultados no varían mucho del primer experimento al ultimo** en un mismo escenario. En el caso de ***SAC*** **sucede lo contrario**, ya que hay una diferencia notable entre los valores del primer y último experimento. Otro detalle interesante, es que ***SAC*** **parte con un mejor rendimiento que *MARLISA***, lo que aumenta más todavía la superioridad de este algoritmo sobre *MARLISA*. Por último, cabe señalar que **los algoritmos de *DRL* han demostrado tener su mejor rendimiento con 25 episodios**.

Cabe destacar que durante este proyecto se obtienen varias funciones de coste, en concreto ocho, pero solo son necesarias dos de ellas para el estudio realizado: emisiones de carbono y costo económico. Aun así, estos datos se presentan en el apéndice con la intención de aportar información útil para otras investigaciones.

5.1. Trabajo Futuro

A raíz del trabajo realizado, surgen una serie de propuestas que podrían resultar interesantes para futuros trabajos. La primera de ellas consiste en un **proyecto que compare el enfoque multiagente descentralizado independiente con el coordinado**. De esta manera se valorarían más opciones de mejora sobre este problema. Aunque es cierto que en muchas tareas de la vida real, no se puede elegir el enfoque a utilizar debido a limitaciones existentes.

Otra alternativa a tener en cuenta, sería **realizar experimentos sobre ciudades más grandes**. Esta limitación viene dada por *CityLearn*, ya que permite escoger diferentes entornos pero el número de edificios suele ser pequeño. De hecho, sería interesante contar con una herramienta que permita una mayor configuración del entorno para poder alterar las condiciones libremente.

Probar una mayor diversidad de algoritmos de *DRL* sería otra idea para futuros trabajos. Así se podría enriquecer más el análisis y comprobar cómo funcionan otras alternativas frente a los mejores modelos de este estudio. Para ello también sería necesario que *CityLearn* añadiera más algoritmos de *DRL* (actualmente solo incluye *SAC* y *MARLISA*).

Por otro lado, se podría **añadir otro elemento más a la función de recompensa**. Esto cambiaría la naturaleza del problema, ya que tendría en cuenta otro factor más. Una opción sería el disconfort que se ha añadido hace relativamente poco a *CityLearn*.

También se podría hacer un **estudio profundo de los resultados obtenidos con las otras funciones de coste** que no se han analizado. En este problema dichas métricas no acompañaban al objetivo planteado, pero en el futuro podría resultar interesante un trabajo enfocado en esto.

5.2. Conocimiento adquirido

Durante la realización del proyecto de fin de máster se ha adquirido una serie de conocimientos relevantes. Esto hace que el trabajo realizado se considere enriquecedor y fructífero desde el punto de vista académico y científico. Principalmente, he aprendido las **bases y los fundamentos del aprendizaje por refuerzo**. Este campo no lo había estudiado ni en el máster, ni en la especialidad de inteligencia artificial del grado de ingeniería informática en la UGR. Además, he afrontado un **problema de aprendizaje por refuerzo profundo** con un **enfoque multiagente descentralizado**

independiente. Gracias a esto he desarrollado destrezas como el análisis y la minería de datos en ámbitos con los que no estoy tan familiarizado. También, he explorado el **campo del control energético** y diferentes enfoques con los que se suele tratar esta tarea. Además, he profundizado sobre los problemas actuales que conlleva la contaminación que produce la energía en edificios y ciudades. Esto me ha concienciado de la responsabilidad que implica el trabajo de científico de datos en estos caso y del relevante papel que juega la ciencia en problemas de esta magnitud.

Apéndice

	Average reward
SAC	-0'283
MARLISA	-0'304
RBC	-0'494

Tabla 5.1: Recompensas medias obtenidas en el escenario 1 con 3 episodios.

	SAC	MARLISA	RBC
1 - load factor	0'985	0'999	1'026
Average daily peak	0'958	0'996	1'659
Carbon emissions	0'989	0'996	1'676
Cost	0'898	0'993	1'623
Electricity consumption	1'005	0'996	1'688
Peak demand	0'946	1	1'518
Ramping	1'128	0'999	3'169
Zero net energy	1'114	1'007	1'316

Tabla 5.2: Funciones de coste obtenidas en el escenario 1 con 3 episodios.

	Average reward
SAC	-0'267
MARLISA	-0'305
RBC	-0'494

Tabla 5.3: Recompensas medias obtenidas en el escenario 1 con 5 episodios.

	SAC	MARLISA	RBC
1 - load factor	0'977	0'998	1'024
Average daily peak	0'894	0'998	1'67
Carbon emissions	0'948	0'997	1'673
Cost	0'842	0'994	1'623
Electricity consumption	0'966	0'998	1'684
Peak demand	0'939	1	1'499
Ramping	1'08	1'014	3'154
Zero net energy	1'114	1'01	1'317

Tabla 5.4: Funciones de coste obtenidas en el escenario 1 con 5 episodios.

	Average reward
SAC	-0'251
MARLISA	-0'303
RBC	-0'496

Tabla 5.5: Recompensas medias obtenidas en el escenario 1 con 10 episodios.

	SAC	MARLISA	RBC
1 - load factor	0'976	0'998	1'026
Average daily peak	0'856	0'995	1'657
Carbon emissions	0'894	0'994	1'68
Cost	0'789	0'99	1'628
Electricity consumption	0'911	0'994	1'692
Peak demand	0'973	1	1'47
Ramping	0'975	1	3'125
Zero net energy	1'11	1'011	1'316

Tabla 5.6: Funciones de coste obtenidas en el escenario 1 con 10 episodios.

	Average reward
SAC	-0'242
MARLISA	-0'302
RBC	-0'496

Tabla 5.7: Recompensas medias obtenidas en el escenario 1 con 15 episodios.

	SAC	MARLISA	RBC
1 - load factor	0'973	0'998	1'023
Average daily peak	0'845	0'99	1'649
Carbon emissions	0'862	0'989	1'68
Cost	0'759	0'985	1'631
Electricity consumption	0'881	0'989	1'696
Peak demand	0'924	1	1'537
Ramping	0'894	0'99	3'182
Zero net energy	1'113	1'011	1'321

Tabla 5.8: Funciones de coste obtenidas en el escenario 1 con 15 episodios.

	Average reward
SAC	-0'237
MARLISA	-0'302
RBC	-0'496

Tabla 5.9: Recompensas medias obtenidas en el escenario 1 con 20 episodios.

	SAC	MARLISA	RBC
1 - load factor	0'967	0'997	1'021
Average daily peak	0'828	0'99	1'637
Carbon emissions	0'847	0'99	1'673
Cost	0'741	0'986	1'634
Electricity consumption	0'866	0'99	1'689
Peak demand	1'002	1	1'496
Ramping	0'866	0'999	3'117
Zero net energy	1'121	1'013	1'316

Tabla 5.10: Funciones de coste obtenidas en el escenario 1 con 20 episodios.

	Average reward
SAC	-0'235
MARLISA	-0'302
RBC	-0'492

Tabla 5.11: Recompensas medias obtenidas en el escenario 1 con 25 episodios.

	SAC	MARLISA	RBC
1 - load factor	0'974	0'998	1'026
Average daily peak	0'839	0'987	1'645
Carbon emissions	0'841	0'988	1'666
Cost	0'73	0'984	1'618
Electricity consumption	0'86	0'988	1'681
Peak demand	0'981	1	1'52
Ramping	0'844	0'99	3'146
Zero net energy	1'119	1'012	1'314

Tabla 5.12: Funciones de coste obtenidas en el escenario 1 con 25 episodios.

	Average reward
SAC	-0'300
MARLISA	-0'338
RBC	-0'521

Tabla 5.13: Recompensas medias obtenidas en el escenario 2 con 3 episodios.

	SAC	MARLISA	RBC
1 - load factor	0'967	1'003	1'003
Average daily peak	0'888	0'994	1'546
Carbon emissions	0'946	0'999	1'624
Cost	0'856	0'996	1'574
Electricity consumption	0'966	0'999	1'649
Peak demand	0'971	1	1'524
Ramping	1'063	1'003	2'783
Zero net energy	1'177	1'01	1'481

Tabla 5.14: Funciones de coste obtenidas en el escenario 2 con 3 episodios.

	Average reward
SAC	-0'283
MARLISA	-0'337
RBC	-0'521

Tabla 5.15: Recompensas medias obtenidas en el escenario 2 con 5 episodios.

	SAC	MARLISA	RBC
1 - load factor	0'972	0'999	0'997
Average daily peak	0'861	0'995	1'516
Carbon emissions	0'905	0'996	1'627
Cost	0'798	0'993	1'578
Electricity consumption	0'926	0'996	1'651
Peak demand	1'009	1	1'404
Ramping	1'025	1	2'752
Zero net energy	1'192	1'01	1'485

Tabla 5.16: Funciones de coste obtenidas en el escenario 2 con 5 episodios.

	Average reward
SAC	-0'266
MARLISA	-0'338
RBC	-0'516

Tabla 5.17: Recompensas medias obtenidas en el escenario 2 con 10 episodios.

	SAC	MARLISA	RBC
1 - load factor	0'974	0'999	1'002
Average daily peak	0'827	0'993	1'526
Carbon emissions	0'85	0'999	1'609
Cost	0'743	0'995	1'562
Electricity consumption	0'871	1	1'633
Peak demand	1'04	1	1'472
Ramping	0'914	1'003	2'782
Zero net energy	1'171	1'015	1'475

Tabla 5.18: Funciones de coste obtenidas en el escenario 2 con 10 episodios.

	Average reward
SAC	-0'261
MARLISA	-0'337
RBC	-0'514

Tabla 5.19: Recompensas medias obtenidas en el escenario 2 con 15 episodios.

	SAC	MARLISA	RBC
1 - load factor	0'97	0'999	0'999
Average daily peak	0'818	0'989	1'502
Carbon emissions	0'835	0'995	1'607
Cost	0'73	0'992	1'552
Electricity consumption	0'858	0'995	1'633
Peak demand	1'027	1	1'444
Ramping	0'868	1'004	2'732
Zero net energy	1'167	1'014	1'475

Tabla 5.20: Funciones de coste obtenidas en el escenario 2 con 15 episodios.

	Average reward
SAC	-0'255
MARLISA	-0'336
RBC	-0'516

Tabla 5.21: Recompensas medias obtenidas en el escenario 2 con 20 episodios.

	SAC	MARLISA	RBC
1 - load factor	0'973	0'998	1'0
Average daily peak	0'803	0'989	1'517
Carbon emissions	0'813	0'992	1'608
Cost	0'709	0'989	1'562
Electricity consumption	0'835	0'993	1'635
Peak demand	1'036	1	1'432
Ramping	0'835	0'993	2'701
Zero net energy	1'184	1'02	1'477

Tabla 5.22: Funciones de coste obtenidas en el escenario 2 con 20 episodios.

	Average reward
SAC	-0'250
MARLISA	-0'335
RBC	-0'519

Tabla 5.23: Recompensas medias obtenidas en el escenario 2 con 25 episodios.

	SAC	MARLISA	RBC
1 - load factor	0’971	0’998	0’994
Average daily peak	0’802	0’989	1’507
Carbon emissions	0’801	0’991	1’618
Cost	0’695	0’988	1’569
Electricity consumption	0’824	0’991	1’642
Peak demand	1’008	1	1’473
Ramping	0’811	0’992	2’805
Zero net energy	1’181	1’019	1’481

Tabla 5.24: Funciones de coste obtenidas en el escenario 2 con 25 episodios.

	Average reward
SAC	-0’346
MARLISA	-0’374
RBC	-0’564

Tabla 5.25: Recompensas medias obtenidas en el escenario 3 con 3 episodios.

	SAC	MARLISA	RBC
1 - load factor	0’953	0’999	1’056
Average daily peak	0’907	0’989	1’529
Carbon emissions	0’986	0’998	1’571
Cost	0’917	0’996	1’619
Electricity consumption	1’007	0’999	1’601
Peak demand	0’945	1	1’613
Ramping	1’246	1’012	3’557
Zero net energy	1’07	1’005	1’185

Tabla 5.26: Funciones de coste obtenidas en el escenario 3 con 3 episodios.

	Average reward
SAC	-0’338
MARLISA	-0’373
RBC	-0’565

Tabla 5.27: Recompensas medias obtenidas en el escenario 3 con 5 episodios.

	SAC	MARLISA	RBC
1 - load factor	0'946	0'999	1'061
Average daily peak	0'883	0'991	1'546
Carbon emissions	0'967	0'996	1'565
Cost	0'892	0'993	1'625
Electricity consumption	0'984	0'997	1'593
Peak demand	1'009	1	1'515
Ramping	1'166	1'015	3'532
Zero net energy	1'063	1'006	1'184

Tabla 5.28: Funciones de coste obtenidas en el escenario 3 con 5 episodios.

	Average reward
SAC	-0'324
MARLISA	-0'373
RBC	-0'564

Tabla 5.29: Recompensas medias obtenidas en el escenario 3 con 10 episodios.

	SAC	MARLISA	RBC
1 - load factor	0'946	0'998	1'051
Average daily peak	0'845	0'986	1'559
Carbon emissions	0'935	0'998	1'572
Cost	0'854	0'995	1'616
Electricity consumption	0'952	0'999	1'6
Peak demand	0'866	1	1'378
Ramping	1'044	1'011	3'598
Zero net energy	1'061	1'005	1'185

Tabla 5.30: Funciones de coste obtenidas en el escenario 3 con 10 episodios.

	Average reward
SAC	-0'316
MARLISA	-0'373
RBC	-0'563

Tabla 5.31: Recompensas medias obtenidas en el escenario 3 con 15 episodios.

	SAC	MARLISA	RBC
1 - load factor	0’927	0’998	1’06
Average daily peak	0’828	0’985	1’553
Carbon emissions	0’914	0’997	1’565
Cost	0’831	0’994	1’613
Electricity consumption	0’931	0’998	1’594
Peak demand	0’856	1	1’758
Ramping	0’976	1’02	3’633
Zero net energy	1’061	1’008	1’185

Tabla 5.32: Funciones de coste obtenidas en el escenario 3 con 15 episodios.

	Average reward
SAC	-0’313
MARLISA	-0’372
RBC	-0’566

Tabla 5.33: Recompensas medias obtenidas en el escenario 3 con 20 episodios.

	SAC	MARLISA	RBC
1 - load factor	0’929	0’998	1’049
Average daily peak	0’82	0’984	1’56
Carbon emissions	0’909	0’995	1’575
Cost	0’824	0’991	1’627
Electricity consumption	0’927	0’996	1’605
Peak demand	0’857	1	1’462
Ramping	0’962	1’009	3’665
Zero net energy	1’061	1’006	1’186

Tabla 5.34: Funciones de coste obtenidas en el escenario 3 con 20 episodios.

	Average reward
SAC	-0’310
MARLISA	-0’371
RBC	-0’563

Tabla 5.35: Recompensas medias obtenidas en el escenario 3 con 25 episodios.

	SAC	MARLISA	RBC
1 - load factor	0'918	0'999	1'049
Average daily peak	0'822	0'985	1'536
Carbon emissions	0'902	0'993	1'564
Cost	0'814	0'989	1'614
Electricity consumption	0'918	0'994	1'594
Peak demand	0'879	1	1'448
Ramping	0'941	1'005	3'543
Zero net energy	1'062	1'006	1'184

Tabla 5.36: Funciones de coste obtenidas en el escenario 3 con 25 episodios.

Bibliografía

[1] The global alliance for buildings and construction, 2022.

[2] S.J. Russell and Norvig. *Inteligencia artificial: un enfoque moderno*. 2004.

[3] José R. Vázquez-Canteli, Jérôme Kämpf, Gregor Henze, and Zoltan Nagy. Citylearn v1.0: An openai gym environment for demand response with deep reinforcement learning. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. Association for Computing Machinery, 2019.

[4] Zhuzhu Wang, Yang Liu, Zhuo Ma, Ximeng Liu, and Jianfeng Ma. Lipsg: Lightweight privacy-preserving q-learning-based energy management for the iot-enabled smart grid. *IEEE Internet of Things Journal*, 2020.

[5] Jose R. Vazquez-Canteli, Gregor Henze, and Zoltan Nagy. Marlisa: Multi-agent reinforcement learning with iterative sequential action selection for load shaping of grid-interactive connected buildings. 2020.

[6] Samuel Idowu, Christer Ahlund, and Olov Schelen. Machine learning in district heating system energy optimization. 2014.

[7] Jose Vazquez-Canteli, Thomas Detjeen, Gregor Henze, Jérôme Kämpf, and Zoltan Nagy. Multi-agent reinforcement learning for adaptive demand response in smart cities. *Journal of Physics: Conference Series*, 2019.

[8] Aurélien Géron. *Hands-on machine learning with Scikit-Learn and TensorFlow concepts, tools, and techniques to build intelligent systems*. OŔeilly Media, 2017.

[9] Richard S.Sutton and Andrew G.Barto. *Reinforcement Learning: An introduction (2nd ed.)*. The MIT Press, 2018.

[10] Gauraang Dhamankar, Jose R. Vazquez-Canteli, and Zoltan Nagy. Benchmarking multi-agent deep reinforcement learning algorithms on

a building energy demand coordination task. Association for Computing Machinery, 2020.

[11] Fazel Khayatian, Zoltán Nagy, and Andrew Bollinger. Using generative adversarial networks to evaluate robustness of reinforcement learning agents against uncertainties. *Energy and Buildings*, 2021.

[12] Kingsley Nweye, Bo Liu, Peter Stone, and Zoltan Nagy. Real-world challenges for multi-agent reinforcement learning in grid-interactive buildings. *Energy and AI*, 2022.

[13] Giuseppe Pinto, Anjukan Kathirgamanathan, Eleni Mangina, Donal P. Finn, and Alfonso Capozzoli. Enhancing energy management in grid-interactive buildings: A comparison among cooperative and coordinated architectures. *Applied Energy*, 2022.

[14] Rongjun Qin, Songyi Gao, Xingyuan Zhang, Zhen Xu, Shengkai Huang, Zewen Li, Weinan Zhang, and Yang Yu. Neorl: A near real-world benchmark for offline reinforcement learning, 2021.

[15] Davide Deltetto. *PhD thesis, Politecnico di Torino*, 2020.

[16] Ruben Glatt, Felipe Leno da Silva, Braden Soper, William A. Dawson, Edward Rusu, and Ryan A. Goldhahn. Collaborative energy demand response with decentralized actor and centralized critic. In *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. Association for Computing Machinery, 2021.

[17] Anjukan Kathirgamanathan, Kacper Twardowski, Eleni Mangina, and Donal P. Finn. A centralised soft actor critic deep reinforcement learning approach to district demand side management through citylearn. In *Proceedings of the 1st International Workshop on Reinforcement Learning for Energy Management in Buildings & Cities*. Association for Computing Machinery, 2020.

[18] Giuseppe Pinto, Davide Deltetto, and Alfonso Capozzoli. Data-driven district energy management with surrogate models and deep reinforcement learning. *Applied Energy*, 2021.

[19] Giuseppe Pinto, Marco Savino Piscitelli, José Ramón Vázquez-Canteli, Zoltán Nagy, and Alfonso Capozzoli. Coordinated energy management for a cluster of buildings through deep reinforcement learning. *Energy*, 2021.

[20] Cheng Yang, Jihai Zhang, Fangquan Lin, Li Wang, Wei Jiang, and Hanwei Zhang. Combining forecasting and multi-agent reinforcement

learning techniques on power grid scheduling task. In *2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA)*, 2023.

[21] Yude Qin, Ji Ke, Biao Wang, and Gennady Fedorovich Filaretov. Energy optimization for regional buildings based on distributed reinforcement learning. *Sustainable Cities and Society*, 2022.

[22] Davide Deltetto, Davide Coraci, Giuseppe Pinto, Marco Savino Piscitelli, and Alfonso Capozzoli. Exploring the potentialities of deep reinforcement learning for incentive-based demand response in a cluster of small commercial buildings. *Energies*, 2021.

[23] Filip Tolovski. Advancing renewable electricity consumption with reinforcement learning, 2020.

[24] Bingqing Chen, Weiran Yao, Jonathan Francis, and Mario Bergés. Learning a distributed control scheme for demand flexibility in thermostatically controlled loads. In *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2020.

[25] Kingsley Nweye, Sivashunmugam Sankaranarayanan, and Zoltán Nagy. Merlin: Multi-agent offline and transfer learning for occupant-centric energy flexible operation of grid-interactive communities using smart meter data and citylearn. *ArXiv*, 2022.

[26] Huiliang Zhang, Di Wu, and Benoit Boulet. Metaems: A meta reinforcement learning-based control framework for building energy management system, 2022.

[27] Sicheng Zhan, Yue Lei, and Adrian Chong. Comparing model predictive control and reinforcement learning for the optimal operation of building-pv-battery systems. *E3S Web of Conferences*, 2023.

[28] Kingsley Nweye, Siva Sankaranarayanan, and Zoltan Nagy. MERLIN: Multi-agent offline and transfer learning for occupant-centric operation of grid-interactive communities. *Applied Energy*, 2023.

[29] Aisling Pigott, Constance Crozier, Kyri Baker, and Zoltan Nagy. Grid-learn: Multiagent reinforcement learning for grid-aware building energy management. *Electric Power Systems Research*, 2022.

[30] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Procs. 35th International Conference on Machine Learning*, 2018.

[31] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 1951.