Digital Object Identifier: 10.24054/rcta.v1i45.3131

Agentes de software basados en técnicas de aprendizaje automático. Perspectivas desde 2010 hasta 2023

Software agents based on machine learning technique. An outlook from 2010 to 2023

MSc. Hipatia Cazares Alegría[1], Dr. Pablo Pico Valencia[2]

[1] ***Pontificia Universidad Católica del Ecuador,*** *Maestría en Tecnologías de la Información, Esmeraldas, Ecuador.*
[2] ***Universidad de Granada,*** *Departamentos de Lenguajes y Sistemas Informáticos, Granada, Spain.*

Correspondencia: *pablo.pico@ugr.es*

Recibido: *04 septiembre 2024.* ***Aceptado:*** *15 diciembre 2024.* ***Publicado:*** *01 enero 2025.*

Cómo citar: *H. Cazares Alegría y P. Pico Valencia, «Agentes de software basados en técnicas de aprendizaje automático. Perspectivas desde 2010 hasta 2023», RCTA, vol. 1, n.º 45, pp. 39–56, ene. 2025.*
Recuperado de https://ojs.unipamplona.edu.co/index.php/rcta/article/view/3131

Esta obra está bajo una licencia internacional
Creative Commons Atribución-NoComercial 4.0.

Resumen: Este estudio tiene como objetivo analizar las principales propuestas teóricas y prácticas en las que se han integrado agentes de software con modelos de aprendizaje automático para determinar su alcance en términos de inteligencia, proactividad, colaboración y aprendizaje. Para el desarrollo de esta investigación se usó la metodología propuesta por Kofod-Peterson. Se analizaron 55 estudios los cuales mostraron que, en la interacción entre agentes de software y aprendizaje automático, los procesos cooperativos y colaborativos se han utilizado ampliamente en la resolución de problemas de control y en la optimización de datos en escenarios distribuidos como el hogar, juegos y las telecomunicaciones. También se evidenció que se utilizaron principalmente modelos de aprendizaje por refuerzo en comparación con los modelos de aprendizaje automático porque contribuyen de manera más significativa al modelado cooperativo de tareas en sistemas inteligentes.

Palabras clave: aprendizaje automático, agente software, sistema multiagente, inteligencia artificial.

Abstract: This study aims to analyze the main theoretical and practical proposals in which software agents have been integrated with machine learning models to determine their scope in terms of intelligence, proactivity, collaboration and learning. For the development of this research, the methodology proposed by Kofod-Peterson was carried out. Applying the methodology, 55 studies were analyzed. The studies showed that in the interaction between software agents and machine learning, cooperative and collaborative processes have been widely used in the resolution of control problems and in the optimization of data in distributed scenarios such as home, games and telecommunication. It was also found that mostly reinforcement learning models were used compared to machine learning models because they contribute more significantly to cooperative task modeling, which is widely used in intelligent systems.

Keywords: machine learning, software agents, multi-agent system, artificial intelligence.

1. INTRODUCCIÓN

Los agentes de software y los sistemas multiagente (MAS, por sus siglas en inglés) han sido ampliamente aplicados en los últimos años. Algunas de estas aplicaciones han sido propuestas teóricas, mientras que otras se han implementado para resolver problemas de producción en el mundo real en diversos escenarios, incluyendo la monitorización de la salud [1], la gestión de la educación en línea [2], el control inteligente de entornos [3], la microeconomía evolutiva [4], la gestión de la información en redes sociales [5], la vigilancia tecnológica [6], entre otros.

En estas aplicaciones, los agentes de software han implementado principalmente mecanismos de inteligencia basados en reglas, lógica y predicados, componentes deliberativos (i.e., creencias, deseos e intenciones), y ontologías. Sin embargo, los agentes de software también han mejorado estos niveles de inteligencia mediante la incorporación de técnicas de Inteligencia Artificial (IA), como el aprendizaje automático [7]. Esta integración ha permitido que los MAS utilicen mecanismos que permiten a las máquinas aprender de grandes conjuntos de datos, aumentando así su efectividad y adaptabilidad.

Actualmente, los MAS son una de las tecnologías de IA utilizadas en el desarrollo de sistemas reactivos orientados a la web, aplicaciones móviles y sistemas embebidos [8]. Las características inherentes de los agentes de software, como la proactividad, la autonomía, la colaboración y la inteligencia, permiten el desarrollo de sistemas cognitivos compatibles con aplicaciones en diversos escenarios como la Industria 4.0 [9], las ciudades inteligentes [9], los hogares inteligentes [10], los hospitales inteligentes [11], las universidades inteligentes [12], entre otros. Estos escenarios varían en complejidad y requieren mecanismos de inteligencia a diferentes escalas. En este contexto, los agentes de software juegan un papel crucial y cada vez más se integran en aplicaciones de computación en la nube, la niebla y el borde para gestionar proactivamente los recursos dentro de los ecosistemas del Internet de las Cosas (IoT, por sus siglas en inglés).

La inteligencia artificial (IA) está convirtiéndose en una parte integral de la vida cotidiana a través de aplicaciones que aprovechan estrategias de aprendizaje basadas en datos. Plataformas como Netflix, YouTube, Amazon, Spotify, Facebook y Tripadvisor, entre otras, ejemplifican esta tendencia. Generalmente, estos sistemas utilizan modelos de aprendizaje automático basados en algoritmos de regresión, clasificación o agrupamiento para entrenar predictores automáticos. Esto les permite realizar acciones con un alto grado de consistencia, como recomendaciones de productos, segmentación de clientes y predicción de precios, entre otras tareas. Por lo tanto, los agentes de software han incorporado modelos de aprendizaje automático para mejorar su nivel de inteligencia y, de este modo, convertirse en entidades proactivas con la capacidad de aprender y tomar decisiones más precisas, coherentes, y más cercanas a las que podría alcanzar un ser humano.

El reciente auge en la integración de agentes de software con modelos de aprendizaje supervisado, no supervisado y de refuerzo, ampliamente documentado en la literatura, ha motivado la realización de una revisión sistemática de la literatura. Esta revisión tiene como objetivo proporcionar una visión exhaustiva del alcance de los agentes de software y los sistemas multiagente (MAS) abordados mediante modelos de aprendizaje basados en datos. Al hacerlo, revela los dominios de conocimiento en los que se han aplicado estos modelos, las tareas que han optimizado, los algoritmos de aprendizaje que han resultado más efectivos y su papel en el apoyo a la toma de decisiones automáticas en los sistemas. Comprender estas experiencias de investigación es crucial para la próxima generación del IoT, conocida como el Internet de los Agentes, que incorporará modelos que mejoren la inteligencia y autonomía del IoT.

El objetivo de este estudio es analizar el alcance de las tecnologías orientadas a agentes combinadas con modelos de aprendizaje supervisado, no supervisado y de refuerzo para identificar las mejores prácticas en el desarrollo de sistemas inteligentes. La metodología utilizada para el proceso de revisión sistemática sigue las pautas propuestas por Kofod-Petersen [13]. Este enfoque proporciona instrucciones completas para realizar revisiones de literatura en el campo de la informática, incluyendo el diseño del estudio, la definición de una cadena de búsqueda científica, recomendaciones para bases de datos documentales especializadas y el establecimiento de criterios de inclusión y exclusión para los estudios recuperados.
Este artículo está organizado en cuatro secciones. La Sección 2 describe la metodología utilizada para llevar a cabo la revisión sistemática de la literatura. En esta sección, se formularon las preguntas de investigación, se explica el proceso de recuperación de estudios y se describen los criterios para la selección de estudios para su análisis. La Sección 3 presenta los resultados de la revisión sistemática,

proporcionando respuestas a cada una de las preguntas de investigación formuladas en la sección de metodología. También se discuten los resultados en términos del alcance y la efectividad de los agentes basados en aprendizaje automático para resolver problemas en entornos inteligentes. Finalmente, la Sección 4 presenta las principales conclusiones y sugerencias para trabajos futuros.

2. BASES TEÓRICAS

La inteligencia artificial (IA) se define como el campo de la informática que estudia la inteligencia en entidades artificiales. Desde una perspectiva de ingeniería, implica la creación de entidades que exhiban un comportamiento inteligente [14]. En esencia, la IA tiene como objetivo desarrollar sistemas que modelen comportamientos inteligentes similares a los humanos, como la comunicación, el aprendizaje, la colaboración, la proactividad y la toma de decisiones [15]. En este contexto, los agentes de software y las técnicas de aprendizaje automático juegan un papel crucial en el avance de la IA y la automatización de procesos.

La robótica ha sido beneficiada por los agentes de software desde sus inicios, ya que permiten a los sistemas soportar características humanas inherentes como la autonomía, la proactividad y la colaboración. Aunque los agentes no son una técnica nueva, siguen siendo fundamentales en la creación de entornos inteligentes, como el Internet de las Cosas (IoT) [16]. Normalmente, la inteligencia de los agentes se implementa utilizando métodos basados en lógica y predicados, lo que permite procesos de inferencia lógica. Sin embargo, en sistemas complejos y ecosistemas emergentes con grandes volúmenes de datos (big data), es crucial que los agentes modelen procesos de aprendizaje similares al cerebro humano. Como resultado, las redes neuronales artificiales (no cubiertas en este estudio) y los modelos de aprendizaje automático han ganado protagonismo. Esta sección describe en profundidad estas dos técnicas: agentes de software y aprendizaje automático.

2.1. Aprendizaje automático

El aprendizaje automático (ML, por sus siglas en inglés) es una estrategia para el análisis de información que automatiza la construcción de modelos de aprendizaje. También es una estrategia enfocada en desarrollar software flexible que se adapte cada vez que se incorporan nuevos datos a un modelo. En términos generales, existen tres tipos principales de aprendizaje automático (ML): supervisado, no supervisado y por refuerzo. A continuación, se describe con mayor detalle.

2.1.1. Aprendizaje supervisado

Es un subconjunto de las técnicas de ML cuyo objetivo es construir modelos que realicen predicciones basadas en evidencias [17]. Los algoritmos de aprendizaje supervisado trabajan con datos etiquetados, intentando encontrar una función de hipótesis que, dadas las variables de entrada, asigne la etiqueta de salida correspondiente [18]. Estos algoritmos cuentan con un corpus ordenado manualmente sobre el cual el algoritmo realiza dos procesos: encontrar los mejores parámetros para el modelo y evaluar el nivel de confiabilidad con esos parámetros. Esta fase se denomina fase de aprendizaje o de entrenamiento. Posteriormente, el modelo entrenado puede hacer predicciones a partir de nuevos datos, evidenciando así el aprendizaje.

Los algoritmos de aprendizaje supervisado pueden ser de regresión, clasificación y agrupamiento [19]. A continuación, se presenta una breve descripción de dichos algoritmos para ayudar a comprender mejor los resultados del estudio.

- **Máquina de vectores de soporte (SVM).** Algoritmo utilizado para realizar la clasificación de problemas en los que existen diferentes clases. El proceso de aprendizaje consiste en encontrar el hiperplano óptimo de separación en escalas dimensionales. Este hiperplano es una línea que divide un plano en dos partes, donde cada clase se encuentra en un lado del plano (ver Figura 1.a).

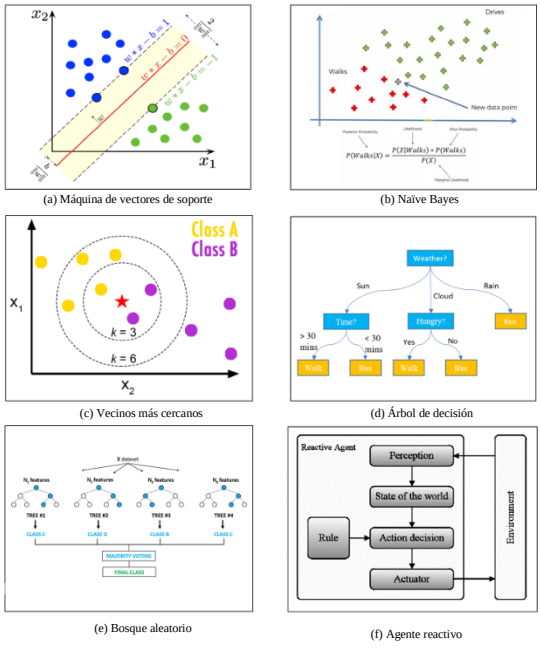
- **Naïve Bayes (NB).** Este algoritmo está basado en técnicas probabilísticas y se utiliza cuando hay entradas multidimensionales que proporcionan la distribución de probabilidad entre dos eventos. Aunque es un algoritmo muy simple, tiene ventajas competitivas al proporcionar mejores resultados que muchos de los otros algoritmos existentes. Por lo tanto, se utiliza en casos donde se requieren métodos de clasificación más avanzados (ver Figura 1.b).

- **K-vecinos más cercanos (KNN).** Es un algoritmo que implementa una técnica supervisada que asume que existen entidades similares en el vecindario. Este modelo permite clasificar valores buscando los puntos de datos más similares (por proximidad) y especificando

un valor de *k*, que corresponde al número de vecinos (ver Figura 1.c).

- **Árbol de decisión (DT).** Este algoritmo identifica la variable más significativa y el valor que proporciona los mejores conjuntos de la población. Los árboles de decisión parten del principio de crear un conjunto de decisiones en forma de árbol, de manera que los nodos intermedios representan soluciones y los nodos finales determinan la predicción del modelo (ver Figura 1.d).

- **Bosque aleatorio (RF).** Es un algoritmo que implementa un versátil método de aprendizaje por conjuntos, es decir, un grupo de modelos más débiles (árboles de decisión) se combinan para formar un modelo poderoso capaz de realizar métodos de reducción dimensional. Cada árbol proporciona una clasificación (vota por una clase) y el resultado es la clase con el mayor número de votos en todo el bosque formado (ver Figura 1.e).



(a) Máquina de vectores de soporte (b) Naïve Bayes (c) Vecinos más cercanos (d) Árbol de decisión (e) Bosque aleatorio (f) Agente reactivo

***Fig. 1**. Representación gráfica de los modelos de aprendizaje automático y arquitectura de un agente reactivo.*
***Fuente:** elaboración propia.*

2.1.2. Aprendizaje no supervisado

En este tipo de aprendizaje, a diferencia del aprendizaje supervisado, no se especifica al algoritmo cuál debe ser la salida; es decir, no hay una relación definida entre las entradas y la salida deseada. Además, en este modelo de aprendizaje no hay influencia externa, ya que no se informa si una salida fue correcta o no. Solo se le suministran grandes cantidades de datos y cada modelo construye sus propias asociaciones.

Los métodos de aprendizaje son diferentes a los utilizados en el modelo de entrenamiento supervisado. El algoritmo principal de este tipo es k-medias, que permite agrupar o segmentar observaciones en grupos basándose en características y distancias entre cada una de las observaciones. Esta agrupación consiste en realizar una minimización de la suma de las distancias entre cada uno de los objetos y el centroide de su grupo (clúster) [20].

2.2. Sistemas multiagentes

Un agente inteligente es una entidad autónoma capaz de realizar acciones en su entorno y percibir el estado del entorno para resolver un problema [21]. Un agente puede ser una entidad física, como un robot equipado con sensores y actuadores, o una entidad virtual, como los agentes de software. Estos últimos tienen propiedades fundamentales como la percepción, el razonamiento, el aprendizaje, la toma de decisiones, la resolución de problemas, la interacción y la comunicación [22].

Los agentes de software se clasifican según diversos criterios. Su categorización depende del investigador, de las tareas que realizan, de la forma en que aprenden, de su arquitectura, entre otros. Según la arquitectura, un agente de software puede ser de los siguientes tipos: agentes reactivos, deliberativos y agentes híbridos.

Las definiciones presentadas anteriormente corresponden a un agente simple. Sin embargo, estas entidades pueden agruparse para resolver problemas complejos. La asociación de varios agentes distribuidos, llamados sistemas multiagente (MAS, por sus siglas en inglés), puede lograr objetivos comunes. Este tipo de sistemas se han usado en una amplia variedad de aplicaciones informáticas, que van desde pequeños sistemas de control autónomos hasta sofisticados sistemas capaces de realizar negociaciones [23].

2.2.1. Agentes reactivos

Esta arquitectura se caracteriza por la ausencia de un elemento central de razonamiento o un modelo simbólico de su entorno. En lugar de ello, los agentes basados en esta arquitectura actúan y responden a los estímulos presentados por el estado

actual del entorno en el que están inmersos (Figura 1.f). Esto implica que los agentes manejan un mecanismo de gestión de eventos basado en reglas, que se activan cada vez que ocurre un cambio en el entorno [24]. Dicho mecanismo es sencillo de implementar. Sin embargo, existen algunos problemas con este modelo, entre ellos: no admite el aprendizaje continuo, no razona, no planifica a largo plazo y cada situación se registra únicamente en un sistema de reglas. El funcionamiento de un agente reactivo está impulsado por los datos que obtiene de los sensores que recogen información del entorno.

2.2.2. Agentes deliberativos

Esta arquitectura utiliza modelos de representación simbólica del conocimiento de manera explícita. Integra componentes BDI como creencias, deseos e intenciones [25]. Una creencia representa el estado del entorno del agente, un deseo representa sus motivaciones y una intención modela sus metas u objetivos. En esta arquitectura, las decisiones se toman utilizando mecanismos de razonamiento lógico basados en coincidencias de patrones y manipulación simbólica. Debido a la complejidad de los algoritmos de manipulación simbólica, estos agentes son difíciles de implementar.

2.2.3. Agentes híbridos

Estos agentes combinan dos o más filosofías dentro de un solo agente. Surgen de la necesidad de maximizar las capacidades y minimizar las deficiencias de cada arquitectura de agente descrita anteriormente para un propósito específico [26].

3. METODOLOGÍA

Una revisión sistemática es una síntesis de investigaciones en la cual el contenido de varios estudios sobre un tema particular es identificado, evaluado críticamente y sintetizado de manera sistemática según estrictos criterios metodológicos. Para llevar a cabo la revisión de la literatura propuesta en este estudio, se siguió la metodología propuesta por Kofod-Petersen [13]. Basados en esta metodología, que se especializa en revisiones de literatura en Ciencias de la Computación, se llevaron a cabo las siguientes tareas: (i) formulación de las preguntas de investigación, (ii) definición de las fuentes de información, (iii) definición de la cadena de búsqueda científica, y (iv) definición de los criterios de inclusión y exclusión para la selección de estudios primarios. Estas tareas se describen en detalle en esta sección.

3.1. Preguntas de investigación (RQs)

El estudio presenta la siguiente pregunta de investigación general: ¿Cómo se han integrado los agentes de software con el aprendizaje automático para mejorar sus capacidades de aprendizaje? Las preguntas específicas que se abordan en este estudio son:

- RQ1: ¿Cuál es el alcance de los agentes de software después de integrar técnicas de aprendizaje automático en su estructura o comportamiento?
- RQ2: ¿Qué tan bien se han utilizado los algoritmos de aprendizaje automático más populares para el desarrollo de sistemas inteligentes?
- RQ3: ¿Qué tipo de problemas se han resuelto con la integración de agentes de software y técnicas de aprendizaje automático?
- RQ4: ¿Qué tipo de aprendizaje han modelado los agentes inteligentes a través del aprendizaje automático?
- RQ5: ¿Cuáles son las fortalezas, oportunidades, debilidades y amenazas de los modelos de agentes basados en técnicas de aprendizaje automático?

3.2. Fuentes de información

Se utilizaron dos bases de datos documentales (Scopus y Web of Science) y cinco bibliotecas digitales (IEEE Xplore, Elsevier, ACM, Wiley y Springer) como fuentes de información para llevar a cabo el proceso de búsqueda científica y, de este modo, recuperar los estudios primarios a ser analizados.

3.2. Estrategia de búsqueda

Las preguntas de investigación definidas previamente proporcionaron las directrices para determinar un conjunto de términos que nos permitiera realizar el proceso de búsqueda. Los términos principales están constituidos por las palabras clave (*agente de software*, *sistema multiagente*), (*aprendizaje automático*, *ML*) e (*inteligente*).

3.3. Criterios de inclusión y exclusión

Los criterios de inclusión definidos para este estudio se centraron en considerar cómo los agentes de software utilizaron el aprendizaje automático como base para crear sistemas inteligentes. Adicionalmente, se establecieron cuatro criterios de

exclusión para descartar estudios que no proporcionaran información relevante. Estos criterios de exclusión fueron los siguientes: artículos repetidos, artículos escritos en un idioma distinto al inglés, artículos inaccesibles y artículos publicados antes de 2010. Los artículos recuperados con la cadena de búsqueda que no cumplían con estos criterios no fueron analizados. Después de aplicar los criterios descritos anteriormente, se analizaron 55 estudios.

4. RESULTADOS Y DISCUSIÓN

Esta sección describe los resultados de la revisión de la literatura. En resumen, esta sección proporciona respuestas a cada una de las preguntas de investigación (RQ1-RQ5) que se propusieron en la sección de metodología. Para presentar los resultados, se utilizan tablas que resumen los hallazgos.

4.1. Estudios analizados

La búsqueda científica aplicada en la metodología permitió recuperar 55 estudios en los que se propusieron integraciones de agentes y aprendizaje automático. Un resumen de los estudios seleccionados se muestra en la Tabla 1. Ambas tablas describen la etiqueta del estudio, la fuente de información, el año de publicación, el país donde se realizó el estudio, el título del estudio y la referencia.

Tabla 1: *Estudios analizados en la revisión.*

ID	FUENTE	AÑO	PAÍS	REF
E1	ACM	2010	EE. UU.	[27]
E2	ACM	2010	Canadá	[28]
E3	ACM	2011	China	[29]
E4	IEEE	2011	Francia	[30]
E5	ACM	2010	Canadá	[31]
E6	ACM	2012	España	[7]
E7	ACM	2012	EE. UU.	[32]
E8	ACM	2012	España	[33]
E9	ACM	2013	EE. UU.	[34]
E10	IEEE	2013	EE. UU.	[35]
E11	ACM	2013	EE. UU.	[36]
E12	IEEE	2014	Polonia	[17]
E13	ACM	2014	EE. UU.	[37]
E14	ACM	2016	Singapur	[38]
E15	ACM	2014	EE. UU.	[39]
E16	ACM	2017	EE. UU.	[40]
E17	ACM	2017	EE. UU.	[21]
E18	IEEE	2018	EE. UU.	[41]
E19	ACM	2018	Suecia	[42]
E20	ACM	2018	EE. UU.	[43]
E21	ACM	2018	Suecia	[44]
E22	ACM	2018	Suecia	[45]
E23	ACM	2018	Canadá	[46]
E24	SCOPUS	2018	EE. UU.	[47]
E25	ACM	2019	EE. UU.	[48]
E26	ACM	2019	Canadá	[49]
E27	ACM	2019	Canadá	[50]
E28	ACM	2019	Canadá	[51]
E29	ACM	2019	EE. UU.	[52]
E30	ACM	2019	Canadá	[53]
E31	ACM	2019	Ciprus	[54]
E32	ACM	2019	Canadá	[55]
E33	ACM	2019	Canadá	[56]
E34	ACM	2019	Canadá	[57]
E35	ACM	2019	EE. UU.	[58]
E36	ACM	2019	Canadá	[59]
E37	ACM	2019	Ciprus	[60]
E38	ACM	2019	Canadá	[61]
E39	ACM	2019	EE. UU.	[62]
E40	ACM	2019	Canadá	[63]
E41	ACM	2019	EE. UU.	[64]
E42	IEEE	2019	EE. UU.	[65]
E43	ACM	2019	EE. UU.	[66]
E44	SCOPUS	2020	EE. UU.	[64]
E45	ACM	2020	N. Zelanda	[67]
E46	ACM	2020	EE. UU.	[68]
E47	SCOPUS	2020	EE. UU.	[69]
E48	SCOPUS	2020	China	[70]
E49	ACM	2020	Suecia	[71]
E50	ACM	2020	Francia	[72]
E51	ACM	2020	EE. UU.	[73]
E52	ACM	2020	N. Zelanda	[74]
E53	ACM	2020	N. Zelanda	[75]
E54	ACM	2020	EE. UU.	[76]
E55	IEEE	2020	Australia	[77]

Fuente: *elaboración propia.*

Por otro lado, se ha evidenciado que el número de publicaciones en las que se han combinado agentes y técnicas de aprendizaje automático ha variado en la última década. Aunque las primeras publicaciones se realizaron entre 2010 y 2016, no fue hasta 2017 cuando se publicaron más estudios (23 estudios, como muestra la Figura 2). Una de las explicaciones para este fenómeno es que el aprendizaje automático se ha popularizado en los últimos años gracias a paradigmas emergentes como el Internet de las Cosas (IoT), la computación en la nube y el big data. Además, debido al desarrollo de la informática, se han incorporado varias bibliotecas de aprendizaje automático para crear modelos predictivos. Estos modelos han llegado al campo de los agentes para mejorar su aprendizaje basado en datos.

Fig. 2*. Estudios analizados por año.*
Fuente: *elaboración propia.*

Para resumir los datos de la Tabla 1, la Figura 3 muestra un gráfico que ilustra los países que han reportado más propuestas orientadas a la integración de agentes de software y técnicas de aprendizaje automático. Se destacan países como Estados Unidos, Suecia y Canadá con 22, 10 y 11 estudios, respectivamente.

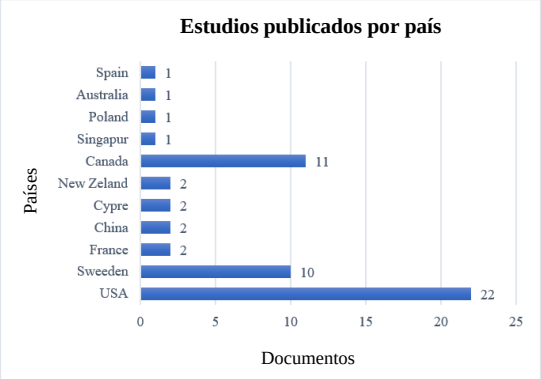


***Fig. 3**. Estudios analizados por país.*
***Source:** elaboración propia.*

4.2. Hallazgos

Para resumir los resultados se emplearon tablas como mecanismo de organización. Cada pregunta ha sido analizada y discutida lógicamente en función de la información obtenida tras la lectura de los 55 artículos seleccionados en este estudio, etiquetados como E1-E55.

4.2.1. ¿Cuál es el alcance de los agentes de software después de integrar técnicas de aprendizaje automático en su estructura o comportamiento?

La Tabla 2 muestra el alcance que han tenido los agentes de software tras ser integrados con técnicas de aprendizaje automático. En términos generales, dicha integración permitió cambios en la estructura de los agentes, logrando mejoras en los siguientes comportamientos: colaboración (58% de los estudios analizados), capacidad de aprendizaje (65%), inteligencia (51%) y, finalmente, la capacidad de actuar de manera autónoma (2% de todos los estudios).

***Tabla 2:** Alcance de los agentes basados en aprendizaje automático en los sistemas analizados. INT=inteligencia, COL=colaboración, LEA=aprendizaje, AUT=autonomía.*

ID	INT	COL	LEA	AUT	REF
E1	x	x	x		[27]
E2					[28]
E3			x		[29]
E4	x	x	x		[30]
E5		x			[31]
E6	x		x		[7]
E7	x		x		[32]
E8			x		[33]
E9	x	x	x		[34]
E10		x	x		[35]
E11	x	x		x	[36]
E12			x		[17]
E13	x		x		[37]
E14		x	x		[38]
E15		x			[39]
E16			x		[40]
E17		x			[21]
E18		x			[41]
E19	x		x		[42]
E20	x	x			[43]
E21			x		[44]
E22		x	x		[45]
E23	x	x	x		[46]
E24			x		[47]
E25	x				[48]
E26			x		[49]
E27	x	x	x		[50]
E28		x			[51]
E29			x		[52]
E30	x	x	x		[53]
E31					[54]
E32	x		x		[55]
E33			x		[56]
E34	x	x	x		[57]
E35	x	x	x		[58]
E36		x	x		[59]
E37	x		x		[60]
E38		x	x		[61]
E39	x	x			[62]
E40		x			[63]
E41	x	x			[65]
E42		x	x		[65]
E43	x	x			[66]
E44		x			[64]
E45	x		x		[67]
E46	x	x			[68]
E47			x		[69]
E48	x				[70]
E49		x			[71]
E50	x	x	x		[72]
E51		x	x		[73]
E52	x		x		[74]
E53	x	x			[75]
E54	x	x	x		[76]
E55	x		x		[77]

***Source:** elaboración propia.*

Es importante señalar que algunos estudios aprovecharon la integración de ambas tecnologías para potenciar dos o más de los comportamientos mencionados, de acuerdo con las necesidades de la aplicación desarrollada. Algunos ejemplos, que se muestran en la Tabla 2, describen casos donde se potenciaron dos (por ejemplo, E6, E10) o más (por ejemplo, E1, E4) aspectos en la misma entidad.

Contribución en términos de inteligencia. En cuanto a la inteligencia, los agentes de software se han centrado principalmente en mejorar su mecanismo de razonamiento lógico, desde el cual actúan de diferentes maneras. La inteligencia es una

característica fundamental lograda en varios estudios, incluyendo E1, E5, E16, E19, E20, E30, E36, E50, E54 y E55. Estos estudios emplearon diferentes enfoques para dotar a los agentes de la capacidad de tomar decisiones óptimas y resolver problemas complejos. Por ejemplo, en E55, se utilizó una combinación de técnicas de clasificación avanzadas, como SVM y bosque aleatorio, para permitir que los agentes analizaran grandes volúmenes de datos y tomaran decisiones informadas en redes de comunicación. De manera similar, en E54, la inteligencia fue fundamental para optimizar redes de semáforos, donde los agentes aprendieron a coordinar y ajustar sus decisiones en tiempo real para mejorar el flujo de tráfico. En comparación con E1 y E16, donde la inteligencia se centró en la adaptación continua a través de Q-Learning, los estudios E19 y E20 integraron técnicas como PCA y Q-Learning para abordar problemas de monitoreo y planificación urbana, demostrando que la inteligencia puede manifestarse de diversas maneras dependiendo del contexto y la complejidad del entorno.

Contribución en términos de aprendizaje. En términos de aprendizaje, un agente puede observar su entorno y, en función de los cambios que ocurren, internalizarlos para ejecutar procesos que le den la capacidad de modificar sus habilidades de razonamiento. El aprendizaje es una característica observada en los estudios E1, E5, E19, E20, E28, E30, E36, E50, E54 y E55, donde los agentes mejoraron su rendimiento a través de la experiencia y la adaptación continua. En E28, se aplicó el aprendizaje evolutivo, permitiendo a los agentes explorar y optimizar soluciones con el tiempo, similar al enfoque adaptativo observado en E36, donde se utilizó el enfoque Actor-Crítico para mejorar la coordinación a largo plazo. En E54 y E30, los agentes aprendieron a través de la experiencia en la optimización del tráfico, mejorando sus políticas a medida que acumulaban más datos y obtenían más conocimientos. En contraste, E5 y E55 utilizaron técnicas de aprendizaje supervisado más tradicionales para mejorar la clasificación y gestión de redes, mientras que E19 y E20 integraron el aprendizaje y la optimización en tareas de monitoreo y planificación, demostrando cómo el aprendizaje puede aplicarse en una variedad de contextos, desde la optimización técnica hasta la planificación estratégica urbana.

Contribución en términos de colaboración. Un agente es colaborativo si puede determinar dinámicamente acciones de coordinación en diferentes situaciones para cumplir objetivos sin afectar su rendimiento. La colaboración fue un elemento clave en estudios como E1, E5, E16, E20, E30, E50 y E54, donde los agentes trabajaron juntos para alcanzar objetivos comunes. En E54, la colaboración fue esencial para la optimización de la red de señales de tráfico, donde los agentes coordinaron sus acciones para mejorar la eficiencia del tráfico. En contraste, en E50, la colaboración se utilizó en la gestión del espectro en redes IoT, permitiendo a los agentes compartir recursos y evitar interferencias. En E30, la colaboración fue fundamental en la simulación del tráfico a nivel microscópico, donde los agentes cooperaron para gestionar el flujo vehicular en entornos urbanos complejos. A diferencia de estudios como E1 y E5, donde la colaboración se centró más en la coordinación básica entre agentes, E20 y E16 exploraron cómo los agentes pueden colaborar de maneras más complejas, compartiendo información y adaptándose a los cambios en el entorno para optimizar la planificación urbana y la predicción en entornos dinámicos.

Contribución en términos de autonomía. Un agente autónomo no solo actúa en respuesta a estímulos en el entorno, sino que exhibe un comportamiento dinámico y escalable orientado a objetivos. La autonomía se manifestó en los estudios E1, E20, E27, E30 y E36, donde los agentes pudieron operar de manera independiente, sin necesidad de intervención externa constante. En E27, la autonomía permitió a los agentes tomar decisiones basadas en su propia evaluación de las acciones, lo cual es similar a lo observado en E36, donde los agentes desarrollaron capacidades de coordinación autónoma a largo plazo. En E30, los agentes demostraron autonomía en la gestión del tráfico, tomando decisiones independientes que optimizaban el flujo vehicular en tiempo real. Por otro lado, en E1 y E20, la autonomía fue clave en la adaptación y optimización en entornos complejos, donde los agentes actuaron de manera independiente para resolver problemas de control y planificación sin necesidad de coordinación externa. Estos estudios destacan cómo la autonomía permite a los agentes actuar de manera más efectiva en entornos donde no es posible la supervisión constante, aunque esta capacidad requiere algoritmos y técnicas avanzadas para garantizar que los agentes puedan mantener un rendimiento óptimo de manera independiente.

4.2.2. RQ2: ¿En qué medida se han utilizado los algoritmos de aprendizaje automático más populares para el desarrollo de sistemas inteligentes?

Una de las motivaciones para abordar este estudio fue conocer los algoritmos de aprendizaje automático utilizados para crear modelos predictivos que aprenden a partir de datos, desde la perspectiva de los agentes que tienen capacidades proactivas de toma de decisiones. A continuación, se describe cómo los agentes han utilizado algoritmos de aprendizaje automático, es decir, aprendizaje supervisado, no supervisado y por refuerzo.

Aprendizaje supervisado. Los algoritmos supervisados permiten el reconocimiento de patrones que se extraen de grandes conjuntos de datos. La Tabla 3 presenta una lista de los estudios principales que han integrado aprendizaje supervisado para alcanzar sus objetivos.

Tabla 3: *Algoritmos de aprendizaje automático utilizados en los estudios analizados. EMPC= Control Predictivo de Modelo Explícito, control basado en EMPC, XGBoost=Gradient Boosting, RNN= Red neuronal, KNN=K-vecinos más cercanos, SVN=Máquina de vectores de soporte, RF=Bosque aleatorio.*

ID	EMPC	XGBOOST	RNN	KNN	SVM	RF
E18	x					
E25		x				
E36			x			
E37			x			
E55				x	x	x

Fuente: *elaboración propia.*

En el estudio E25, se utilizó *XGBoost* para la estimación del esfuerzo en el desarrollo de software. Este algoritmo fue seleccionado por su capacidad para manejar grandes volúmenes de datos y generar predicciones precisas, mejorando significativamente la exactitud en la estimación del esfuerzo requerido para completar proyectos de software.

Por otro lado, las Redes Neuronales Recurrentes (RNN) se aplicaron en los estudios E36 y E37 para capturar dependencias temporales en entornos de aprendizaje por refuerzo multiagente y en problemas donde la información sobre el estado completo del sistema no siempre está disponible. Estas redes permitieron a los agentes aprender patrones en secuencias temporales y predecir estados futuros, mejorando así la toma de decisiones en entornos dinámicos y parcialmente observables.

Asimismo, el algoritmo de K-vecinos más cercanos (KNN) se utilizó en el estudio E55 para clasificar la calidad de la transmisión en una red gestionada por múltiples agentes, basado en parámetros como latencia y jitter. KNN demostró ser eficaz en la clasificación en tiempo real de estos parámetros, lo cual es esencial para mantener la calidad del servicio en redes de comunicación.

Finalmente, en el estudio E55, se usaron de manera complementaria los algoritmos SVM, bosque aleatorio y el clasificador de vectores de soporte Nu (Nu-SVC) para mejorar la precisión en la clasificación de parámetros de transmisión en redes gestionadas por múltiples agentes. SVM fue seleccionado por su capacidad para manejar datos de alta dimensión y realizar clasificaciones precisas en escenarios complejos. El algoritmo bosque aleatorio contribuyó con su robustez y capacidad para reducir el sobreajuste, lo que mejoró la fiabilidad del sistema al clasificar conjuntos de datos complejos y ruidosos. Nu-SVC se destacó por su flexibilidad para ajustar los márgenes de separación, lo que permitió una clasificación más precisa y adaptable en un entorno de red con variaciones de datos.

Aprendizaje no supervisado. Dentro del análisis de los estudios, también se evidenció la aplicación del aprendizaje no supervisado. Sin embargo, debido a que existen pocas variantes de algoritmos de esta naturaleza y debido a que estos algoritmos parten de un conjunto de datos del cual no hay conocimiento a priori, su uso no está ampliamente evidenciado. Dado que el objetivo de estos algoritmos es analizar la comprensión de los datos o su transformación automática, los estudios analizados han aplicado generalmente modelos más supervisados y de refuerzo (Figura 4).

Tabla 4: *Algoritmos no supervisados de aprendizaje automático utilizados en los estudios analizados. PCA=Análisis de Componentes Principales, HC= Agrupación Jerárquica.*

ID	K-MEANS	PCA	HC
E5	x		
E8		x	
E13			x
E19		x	
E42		x	
E43	x		

Fuente: *elaboración propia.*

Uno de los estudios que sí empleó este tipo de algoritmo son las propuestas descritas en E5 y E43. Estas utilizaron el algoritmo K-medias para agrupar datos en clústeres con características similares. En el estudio E5, K-medias se aplicó para clasificar diferentes tipos de tráfico en redes cognitivas, facilitando una gestión eficiente del espectro. Los agentes en este contexto actuaron como controladores que gestionan y adaptan la asignación de espectro en tiempo real, basados en los clústeres identificados por el algoritmo.

Por otro lado, también se empleó el algoritmo de Análisis de Componentes Principales (PCA). En E8, PCA se utilizó para simplificar el espacio de características en un entorno de aprendizaje por refuerzo multiagente, mejorando la eficiencia del aprendizaje al reducir la complejidad del modelo. Los agentes en este estudio aprovecharon las representaciones reducidas para aprender más rápido y con mayor precisión.

Finalmente, el algoritmo de Clúster Jerárquico también fue empleado por el estudio E13. El algoritmo ayudó a organizar los datos en una estructura jerárquica, lo que permitió una clasificación más detallada y progresiva de los datos. En este estudio, los agentes se emplearon en un entorno de simulación donde la jerarquía de clústeres ayudó a los agentes a identificar patrones y tomar decisiones basadas en el nivel de detalle necesario. Esto fue particularmente útil en la simulación de comportamientos humanos en entornos urbanos, donde los agentes necesitaban interpretar y actuar sobre datos agrupados en diferentes niveles de granularidad.

Estos algoritmos no supervisados permitieron a los agentes organizar y simplificar los datos, lo que a su vez mejoró la capacidad de los agentes para aprender, adaptarse y actuar en sus respectivos entornos. Estos algoritmos sirvieron como herramientas fundamentales para mejorar la efectividad y eficiencia de los agentes en la resolución de problemas complejos en tiempo real.

Aprendizaje por refuerzo. Los algoritmos de aprendizaje por refuerzo también se han aplicado de manera integrada con agentes para llevar a cabo procesos de toma de decisiones. El uso de los principales algoritmos se detalla en el resumen de la Tabla 5.

Tabla 5: *Principales algoritmos de aprendizaje por refuerzo utilizados por los estudios analizados.*

ID	Q-LEARNING	DEEP Q-LEARNING (DQN)	SARSA	ACTOR-CRITIC	POLICY GRADIENT	MONTE CARLO METHODS
E1	x					
E16	x					
E27				x		
E30	x	x		x	x	
E36				x	x	
E44	x	x	x	x	x	x
E50	x		x			x
E23		x				
E13	x					x
E20	x				x	x

Fuente: *elaboración propia.*

El algoritmo Q-Learning se utilizó en estudios como E1, E16, E30, E44, E50, E13 y E20. Q-Learning es un método basado en valores que permite a los agentes aprender una política óptima iterando directamente sobre la función de valor de acción. En estos estudios, Q-Learning demostró ser una opción popular debido a su simplicidad y efectividad, especialmente en entornos multiagente donde los agentes necesitan explorar y explotar el entorno para encontrar la mejor política de acción.

Otro algoritmo utilizado fue la red Q Profunda (DQN). Este algoritmo, una extensión de Q-Learning que utiliza redes neuronales profundas, se utilizó en los estudios E23, E30 y E44. DQN es especialmente útil en situaciones donde el espacio de estados es continuo o muy grande, como en el estudio E30, donde los agentes tenían que manejar la simulación del tráfico a nivel microscópico. DQN permitió a los agentes aproximar la función de valor en espacios de estados complejos, mejorando significativamente la capacidad del agente para tomar decisiones en entornos de alta dimensión en comparación con el Q-Learning tradicional. La integración de redes neuronales permitió a los agentes manejar entornos más ricos en información, aunque a costa de una mayor complejidad computacional y requisitos de datos para el entrenamiento.

También en la misma línea, el algoritmo SARSA, empleado en los estudios E44 y E50. SARSA es un método basado en valores que, a diferencia de Q-Learning, aprende la política directamente a partir de la experiencia del agente, lo que lo hace más conservador en la exploración. En estos estudios, SARSA fue ventajoso en escenarios donde los agentes necesitaban un enfoque más seguro, minimizando el riesgo de tomar decisiones que pudieran llevar a resultados negativos en entornos inciertos. Por ejemplo, en E50, SARSA se utilizó en la gestión del espectro de redes IoT, donde era

crucial evitar decisiones que pudieran comprometer la calidad de servicio. En comparación con Q-Learning, SARSA ofrece mayor estabilidad en entornos donde las recompensas pueden ser volátiles.

El enfoque Actor-Crítico también se aplicó en los estudios E27, E30, E36 y E44. Dicho enfoque combina lo mejor de los métodos basados en políticas (actor) y basados en valores (crítico). Este algoritmo permite a los agentes aprender cuál es la mejor acción para tomar, y también evaluar la calidad de esa acción, lo que reduce la varianza en la estimación de la política y mejora la estabilidad del aprendizaje. En el estudio E36, Actor-Crítico se utilizó para la coordinación a largo plazo en entornos multiagente, donde la necesidad de reducir la varianza y garantizar decisiones consistentes era crítica. En comparación con Q-Learning y SARSA, Actor-Crítico tiene la ventaja de ser más eficiente en entornos continuos y de alta dimensión, pero requiere un cuidadoso equilibrio entre las actualizaciones del actor y el crítico para evitar inestabilidades.

Los métodos de Gradiente de Política, utilizados en E30, E36, E44 y E20, optimizan directamente la política del agente como una función del gradiente de la recompensa esperada. Estos métodos son particularmente útiles en escenarios donde las políticas deben ajustarse de manera suave y continua, como en E36, donde la coordinación de múltiples agentes requería ajustes constantes en un entorno dinámico. Los agentes se beneficiaron de la capacidad de los métodos de Gradiente de Política para manejar políticas estocásticas, lo que permite una exploración más efectiva y una mejor adaptación a los cambios en el entorno. En comparación con Q-Learning y SARSA, el método de Gradiente de Política es más adecuado para entornos continuos y complejos, aunque es más susceptible a problemas de lenta convergencia y alta varianza en las actualizaciones de la política.

Finalmente, los métodos de Monte Carlo también se emplearon en los estudios E44, E50, E13 y E20. Básicamente, estos métodos se utilizaron para estimar funciones de valor basadas en el retorno promedio de episodios completos. Estos métodos fueron especialmente útiles en estudios como E13, donde las tareas eran episódicas y se requería una estimación precisa del valor de estado-acción. Los agentes en estos estudios se beneficiaron de la simplicidad de la actualización de la política basada en muestras completas de episodios, lo que es ventajoso en entornos donde las recompensas pueden ser altamente variables. Sin embargo, en comparación con métodos como Q-Learning o el método de Gradiente de Política, los métodos de Monte Carlo pueden ser menos eficientes en entornos donde los episodios son largos o donde la retroalimentación de recompensas es esporádica.

4.2.3. RQ3: ¿Qué tipo de problemas se han resuelto con la integración de agentes de software y técnicas de aprendizaje automático??

Muchos de los modelos en los que se han integrado agentes con aprendizaje automático son propuestas conceptuales. Los modelos de agentes que se han llevado al campo práctico han permitido resolver algunos problemas interesantes, particulares y útiles en el mundo moderno. La Tabla 6 describe algunos de los problemas en los que se han utilizado los agentes estudiados en este trabajo.

Tabla 6: *Principales problemas en los que se ha utilizado el aprendizaje automático basado en agentes.*

ID	DOMINIO	PROBLEMA RESUELTO
E1	IA y Multi-Agente	Desarrollo de algoritmos de aprendizaje automático para la coordinación en sistemas multi-agente.
E4	IA y Comunicación	Mejora de la comunicación entre agentes en un entorno multi-agente utilizando aprendizaje por refuerzo.
E5	Telecomunicaciones	Gestión de espectro en radios cognitivas utilizando técnicas de aprendizaje.
E7	Aprendizaje por Refuerzo	Mejora de la generalización en el aprendizaje por refuerzo multi-agente a través de la factorización de tensores.
E10	Control y Automatización	Implementación de aprendizaje cooperativo en sistemas de control continuo.
E11	Gestión de Tráfico Urbano	Optimización del control de tráfico urbano utilizando aprendizaje por refuerzo multi-agente.
E13	Simulación Basada en Agentes	Simulación y predicción de comportamientos colectivos en entornos urbanos.
E14	Logística y Optimización	Optimización del despacho dinámico en operaciones logísticas utilizando aprendizaje por refuerzo multi-agente.
E16	Entornos Dinámicos	Desarrollo de técnicas de predicción para mejorar el rendimiento del aprendizaje por refuerzo en entornos dinámicos.
E18	Gestión de Tráfico Urbano	Implementación de un enfoque de aprendizaje automático compartido para la gestión del tráfico urbano.

E19	Rastreo y Monitoreo	Desarrollo de un sistema multi-agente para el rastreo y monitoreo de múltiples objetos utilizando aprendizaje por refuerzo.
E20	Planificación Urbana	Aplicación del aprendizaje por refuerzo multi-agente para la planificación y desarrollo de proyectos urbanos.
E22	Inteligencia Artificial y Multi-Agente	Propuesta de redes de descomposición de valor para mejorar la cooperación en entornos multi-agente.
E24	Desarrollo de Software	Aplicación de técnicas de aprendizaje automático para la estimación de esfuerzo en el desarrollo de software.
E25	Gestión de Riesgos	Desarrollo de enfoques de aprendizaje por refuerzo aversos al riesgo en entornos multi-agente mixtos.
E27	Control y Optimización	Desarrollo de algoritmos actor-crítico para el aprendizaje por refuerzo multi-agente en entornos restringidos.
E28	Inteligencia Artificial y Multi-Agente	Capacitación de agentes cooperativos en sistemas de aprendizaje por refuerzo multi-agente.
E29	Conducción Autónoma	Arquitectura de aprendizaje por refuerzo con compartición de parámetros para la conducción autónoma en entornos multi-agente.
E30	Gestión de Tráfico	Simulación y optimización del tráfico microscópico utilizando aprendizaje por refuerzo multi-agente profundo.
E34	Aprendizaje por Refuerzo Profundo	Desarrollo de técnicas de aprendizaje por refuerzo profundo competitivo en entornos multi-agente.
E35	Inteligencia Artificial y Aprendizaje	Mejora de los algoritmos de aprendizaje por refuerzo cooperativo mediante la incorporación de demostraciones mixtas.
E36	Aprendizaje por Refuerzo Profundo	Implementación de aprendizaje por refuerzo profundo jerárquico para la coordinación a largo plazo en entornos multi-agente.
E37	Control Adaptativo	Control adaptativo de sistemas multi-agente utilizando aprendizaje por refuerzo profundo.
E38	Simulación y Atención Médica	Evaluación de servicios de atención médica a través de simulación basada en agentes y aprendizaje automático.
E39	Gestión de Tráfico Urbano	Creación de un entorno de aprendizaje por refuerzo multi-agente para la gestión de tráfico urbano a gran escala.
E41	Logística y Transporte	Optimización del despacho de pedidos a través de la asignación de vehículos utilizando aprendizaje por refuerzo multi-agente.
E42	Energía y Redes Eléctricas	Desarrollo de un esquema de protección para microrredes de CA basado en sistemas multi-agente y aprendizaje automático.
E43	Inteligencia Artificial y Aprendizaje	Implementación de un sistema de aprendizaje multi-agente adaptativo basado en razonamiento incremental híbrido basado en casos.
E45	Gestión de Tráfico Urbano	Aplicación de aprendizaje profundo feudal para la gestión del tráfico en entornos urbanos.
E46	Logística y Optimización	Desarrollo de un sistema de aprendizaje por refuerzo multi-agente cooperativo para optimizar sistemas exprés.
E47	Inteligencia Artificial y Comunicación	Desarrollo de técnicas para aprender comunicación multi-agente a través de juegos de emisor-receptor fundamentados.
E50	Telecomunicaciones	Gestión cooperativa del espectro en redes cognitivas de IoT utilizando aprendizaje por refuerzo multi-agente.
E52	Gestión de Tráfico Urbano	Optimización de la gestión del tráfico utilizando aprendizaje por refuerzo multi-agente con comunicación emergente.
E54	Gestión de Tráfico Urbano	Optimización de redes de señales de tráfico a través de la integración de aprendizaje por refuerzo multi-agente independiente y centralizado.
E55	Redes y Comunicaciones	Aplicación de técnicas de aprendizaje automático para la clasificación de parámetros de transmisión en redes administradas por múltiples agentes.

Fuente: *elaboración propia.*

En el campo de la optimización del tráfico urbano, varios estudios utilizaron técnicas de aprendizaje por refuerzo multiagente. Entre ellos, los estudios E11, E39, E45, E54 y E52 comparten el objetivo de mejorar la gestión del tráfico en entornos urbanos mediante la optimización de las señales de tráfico y los flujos vehiculares. Sin embargo, cada uno aborda el problema desde una perspectiva diferente.

Por ejemplo, E39 presenta un entorno específico para la simulación y optimización de tráfico a gran escala, mientras que E54 propone una combinación de aprendizaje independiente y centralizado para optimizar las redes de señales de tráfico. Por otro lado, E45 emplea un enfoque de aprendizaje profundo feudal para gestionar el tráfico, centrándose en la estructura jerárquica de la toma de decisiones.

Dentro del campo de la inteligencia artificial aplicada a sistemas multiagente, los estudios E1, E4, E22 y E28 presentan avances significativos. Estos estudios se centran en desarrollar algoritmos y arquitecturas que permitan una cooperación y comunicación eficientes entre los agentes dentro de un entorno compartido. En particular, E4 y E28 se concentran específicamente en mejorar la comunicación entre agentes mediante el uso de

técnicas de aprendizaje por refuerzo. Por el contrario, E22 propone una red de descomposición de valor que facilita la cooperación entre agentes. E1, sin embargo, adopta un enfoque más amplio, explorando el desarrollo de algoritmos de aprendizaje para la coordinación en sistemas multiagente sin restringirse a un dominio específico.

Los estudios E14, E46 y E41 demuestran aún más la aplicabilidad del aprendizaje por refuerzo en la optimización de la logística. Estas investigaciones exploran cómo estos algoritmos pueden optimizar tareas como el despacho dinámico de recursos y la asignación de vehículos específicamente para la logística y el transporte. Por ejemplo, E14 aborda el desafío más amplio del despacho dinámico en las operaciones logísticas, mientras que E41 se centra en optimizar la asignación de vehículos para la entrega de pedidos. E46, por otro lado, utiliza un enfoque de aprendizaje por refuerzo cooperativo para optimizar los sistemas de mensajería exprés.

En el ámbito de la energía y las redes eléctricas, el estudio E42 propone un enfoque para la protección de microrredes de CA utilizando sistemas multiagente y aprendizaje automático. Este estudio es único en su aplicación y muestra cómo los enfoques de aprendizaje por refuerzo pueden extenderse a dominios más técnicos y específicos como la gestión y protección de infraestructuras críticas.

Finalmente, en la simulación basada en agentes, el estudio E13 se centra en la predicción del comportamiento colectivo en entornos urbanos utilizando modelos basados en agentes. Este enfoque es diferente de otros estudios más orientados a la optimización de procesos, ya que se centra en comprender y predecir el comportamiento humano a través de la simulación.

4.2.4. RQ4: ¿Qué otro tipo de aprendizaje se modeló en los agentes para que fueran más inteligentes?

El análisis de los estudios permitió determinar que los agentes, gracias a otras técnicas adicionales al aprendizaje automático, pudieron modelar comportamientos más complejos e inteligentes. A continuación, se describe cómo se aplicaron estas técnicas o métodos:

Una de las técnicas más extendidas utilizadas en los estudios es el Aprendizaje Basado en Reglas. Esta técnica se utiliza en el estudio E5, donde los agentes toman decisiones siguiendo un conjunto predefinido de reglas. Esta técnica es especialmente útil en entornos donde el comportamiento esperado puede ser codificado explícitamente.

Posteriormente, el Aprendizaje Bayesiano también se aplica en el estudio E12. En este enfoque, los agentes actualizan continuamente sus creencias sobre el mundo a medida que adquieren nueva evidencia. Esta técnica es fundamental en escenarios caracterizados por una alta incertidumbre, donde la probabilidad juega un papel crucial en la toma de decisiones.

El Razonamiento Basado en Casos también se estudia en E13 y E43. Esta técnica permite a los agentes resolver nuevos problemas adaptando soluciones de problemas similares resueltos en el pasado. Es especialmente útil en situaciones donde se dispone de un rico historial de experiencias previas.

Por otro lado, el Aprendizaje de Árboles de Decisión se aplica en el estudio E24. Esta técnica divide el espacio de decisión en subconjuntos más manejables, permitiendo a los agentes tomar decisiones basadas en una jerarquía de preguntas.

Continuando con el análisis, se presenta el Aprendizaje por Transferencia en E41. Esta técnica permite a los agentes aprovechar el conocimiento adquirido en un contexto para mejorar su rendimiento en otro contexto relacionado, lo que es especialmente útil cuando los datos en el nuevo dominio son limitados.

El estudio E43 combina el razonamiento basado en casos con métodos incrementales en el Razonamiento Incremental Híbrido Basado en Casos (IHCBR), permitiendo a los agentes mejorar continuamente su base de conocimiento a medida que adquieren nueva experiencia.

Finalmente, el Aprendizaje por Imitación es utilizado por los autores del estudio E53. Esta técnica permite a los agentes aprender comportamientos observando a otros agentes o humanos, lo que es útil en entornos donde el comportamiento deseado puede ser observado, pero no fácilmente programado.

4.2.6. RQ5: ¿Cuáles son las fortalezas, oportunidades, debilidades y amenazas de los modelos de agentes basados en técnicas de aprendizaje automático?

Se han determinado las siguientes fortalezas, oportunidades, debilidades y amenazas con respecto al desarrollo de agentes que utilizan aprendizaje automático para mejorar las características inherentes de los agentes: inteligencia, colaboración, aprendizaje, adaptación y proactividad.

Fortalezas (F)

- F1. Existencia de modelos y algoritmos de aprendizaje automático a nivel de implementación compatibles con varios lenguajes de programación que facilitan su integración con herramientas de desarrollo de agentes. Por ejemplo, JADE puede implementar modelos de aprendizaje automático Weka o Deeplearning4j (DL4J), y SPADE puede implementar modelos de aprendizaje automático en Python utilizando scikit-learn.
- F2. Uso de servicios en la nube para distribuir mecanismos de aprendizaje sin la necesidad de que los agentes los integren como parte de su estructura. Esto permite que los agentes sean livianos y que se apliquen nuevas configuraciones sin modificar a los agentes mismos. Además, los procesos de entrenamiento de los modelos se realizarán en infraestructuras sofisticadas. Esto es de suma importancia para el desarrollo de sistemas de automóviles autónomos compatibles con la nube y el procesamiento en el borde.
- F3. Disponibilidad de estándares para el desarrollo de agentes, lo que permite a los agentes de aprendizaje compartir su conocimiento con agentes homólogos dentro del ecosistema en el que el agente se ejecuta y colabora. Un caso específico es el estándar FIPA-ACL para establecer comunicación entre agentes y los protocolos de comunicación FIPA para modelar procesos de interacción complejos en sistemas heterogéneos, entre otros.

Debilidades (D)

- D1. Los modelos de aprendizaje automático suelen ser demasiado pesados para que un agente los integre en su estructura. Esta complejidad los hace difíciles de usar en sistemas embebidos y dependientes de la computación en la nube.
- D2. No existen metodologías o herramientas formales que permitan el desarrollo de este tipo de agente sin la necesidad de integrar varias herramientas disponibles.

Oportunidades (O)

- O1. Existe una necesidad de entidades inteligentes que aprendan de los big data para los sistemas modernos. La disponibilidad de datos generados por las redes sociales y el IoT posiciona a los agentes que utilizan técnicas de aprendizaje automático como entidades útiles para operar en ecosistemas inteligentes avanzados.
- O2. Existe disponibilidad de plataformas orientadas hacia el desarrollo de agentes en múltiples lenguajes de programación. Esto permite el desarrollo de agentes que mejoran su capacidad de aprendizaje para que puedan ser utilizados en diferentes entornos emergentes como aplicaciones web, aplicaciones móviles, sistemas embebidos, la nube, la computación de borde, la computación de niebla, entre otros.
- O3. Los agentes que integran mecanismos de aprendizaje basados en datos pueden emplear blockchain para determinar la confiabilidad de los datos utilizados para modelar acciones de aprendizaje. Esto sería de gran importancia para evitar que los agentes aprendan de información poco confiable y, en consecuencia, tomen decisiones inadecuadas que afecten el entorno en el que operan.

Amenazas (A)

- A1. Los agentes son vulnerables a ataques por parte de expertos en informática. Aunque las técnicas de aprendizaje automático ayudan a mejorar la capacidad de aprendizaje del entorno del agente, no contribuyen significativamente a mejorar los mecanismos de seguridad y privacidad de datos. Sin embargo, utilizando aprendizaje automático, los agentes podrían aprender a identificar comportamientos maliciosos, detectar noticias falsas y más.

5. CONCLUSIONES

Este artículo analizó cómo los agentes de software mejoran su nivel de inteligencia, colaboración, autonomía y adaptación al integrar modelos de aprendizaje automático. Se examinaron los fundamentos teóricos y prácticos de las tecnologías orientadas a agentes y el aprendizaje automático para la creación de sistemas inteligentes. Se presentaron propuestas que incorporan algoritmos de aprendizaje supervisado y no supervisado. Sin

embargo, en la mayoría de los casos, los agentes y los sistemas multiagente utilizaron algoritmos de aprendizaje por refuerzo. Al combinar estos algoritmos, los agentes pudieron optimizar tareas relacionadas con la comunicación y la coordinación en escenarios conocidos y desconocidos.

Las tecnologías orientadas a agentes tienen amplias oportunidades de desarrollo a la luz del auge de paradigmas como la Inteligencia Artificial, la Computación en la Nube, Blockchain y Big Data. Se ha demostrado que los agentes, a través de algoritmos de aprendizaje automático, pueden aprender a partir de datos y ejecutar tareas de predicción automática para alcanzar sus objetivos de manera más eficaz; es decir, coordinando mejor las acciones en grupos de agentes e identificando patrones en los datos del entorno en el que operan para tomar mejores decisiones. Además, se ha demostrado que los modelos de aprendizaje profundo, muchos de ellos basados en aprendizaje supervisado, también han sido utilizados por agentes y sistemas multiagente. Sin duda, este proceso de integración permitirá la mejora de escenarios emergentes como el Internet de las Cosas y el Internet de los Agentes, un paradigma en el que los agentes inteligentes son los actores predominantes.

Para trabajos futuros, se propone que la dinámica de los agentes inteligentes con tecnologías de aprendizaje automático se incorpore a la interacción con big data en la computación en la nube para implementar sistemas de apoyo a la toma de decisiones proactivos, útiles en diferentes ecosistemas de Internet, como el Internet de las Cosas y sus respectivas aplicaciones en ciudades, universidades, hospitales e industrias inteligentes.

REFERENCIAS

[1] S. K. Polu, “Modeling of efficient multi-agent based mobile health care system,” *Int J Innov Res Sci Technol*, vol. 5, no. 8, pp. 10–14, 2019.

[2] S. Munawar, S. K. Toor, M. Aslam, and E. Aimeur, “PACA-ITS: A Multi-agent system for intelligent virtual laboratory courses,” *Applied Sciences (Switzerland)*, vol. 9, no. 23, 2019, doi: 10.3390/app9235084.

[3] S. Cho and F. Zhang, “An adaptive control law for controlled lagrangian particle tracking,” *WUWNet 2016 - 11th ACM International Conference on Underwater Networks and Systems*, 2016, doi: 10.1145/2999504.3001077.

[4] R. K. Jain *et al.*, “Stability analysis of piezoelectric actuator based micro gripper for robotic micro assembly,” *ACM International Conference Proceeding Series*, no. c, 2013, doi: 10.1145/2506095.2506105.

[5] M. Tanti, S. Fossey, L. Madrid-Briand, P. Carrieri, B. Spire, and P. Roux, “Une analyse de Twiter pour mieux comprendre les acteurs de la communication des nouvelles drogues et leurs discussions,” *ACM International Conference Proceeding Series*, pp. 36–38, 2018, doi: 10.1145/3240431.3240438.

[6] Y. Islen and S. Juan, “Componente para la extracción y transformación de datos en el proceso de vigilancia tecnológica Component for the data mining and transformation within the technological surveillance process,” no. June 2017, 2016.

[7] M. Kaisers, D. Bloembergen, and K. Tuyls, “A Common Gradient in Multi-agent Reinforcement Learning (Extended Abstract),” *Proc. of 11th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 1393–1394, 2012.

[8] S. H. Chen and T. K. Fu, “Eliminating artificial-natural dichotomy a formal study on a core cognitive process in artificial intelligence,” *ACM International Conference Proceeding Series*, vol. Part F1285, 2017, doi: 10.1145/3080845.3080866.

[9] S. R. Hamidi, E. N. M. Ibrahim, M. F. B. A. Rahman, and S. M. Shuhidan, “Industry 4.0 urban mobility: goNpark smart parking tracking module,” *ACM International Conference Proceeding Series*, pp. 503–507, 2017, doi: 10.1145/3162957.3163042.

[10] C. Cappelli, G. V. Pereira, M. B. Bernardes, F. Bernardini, and A. Gomyde, “Building a reference model & an evaluation method for cities of the Brazilian network of smart & human cities,” *ACM International Conference Proceeding Series*, vol. Part F1282, pp. 580–581, 2017, doi: 10.1145/3085228.3085257.

[11] A. A. F. Brandão, L. Vercouter, S. Casare, and J. Sichman, “Exchanging reputation values among heterogeneous agent reputation models: An experience on ART testbed,” *Proceedings of the International Conference on Autonomous Agents*, vol. 5, pp. 1047–1049, 2007, doi: 10.1145/1329125.1329405.

[12] I. Menchaca, M. Guenaga, and J. Solabarrieta, “Using learning analytics to assess project management skills on engineering degree courses,” *ACM International Conference Proceeding Series*, vol. 02-04-Nove, pp. 369–376, 2016, doi: 10.1145/3012430.3012542.

[13] A. Kofod-petersen, “How to do a Structured Literature Review in computer science,” 2014. [Online]. Available: https://research.idi.ntnu.no/aimasters/files/SLR_HowTo2018.pdf

[14] E. Saadatian, T. Salafi, H. Samani, Y. De Lim, and R. Nakatsu, “An affective telepresence system using smartphone high level sensing and intelligent behavior generation,” *HAI 2014 - Proceedings of the 2nd International Conference on Human-Agent Interaction*, pp. 75–82, 2014, doi: 10.1145/2658861.2658878.

[15] J. Jumadinova, P. Dasgupta, and L. K. Soh, “Strategic capability-learning for improved multi-agent collaboration in ad-hoc environments,” *Proceedings - 2012 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2012*, vol. 2, pp. 287–292, 2012, doi: 10.1109/WI-IAT.2012.57.

[16] J. A. Manrique, J. S. Rueda-Rueda, and J. M. T. Portocarrero, “Contrasting Internet of Things and Wireless Sensor Network from a Conceptual Overview,” *Proceedings - 2016 IEEE International Conference on Internet of Things; IEEE Green Computing and Communications; IEEE Cyber, Physical, and Social Computing; IEEE Smart Data, iThings-GreenCom-CPSCom-Smart Data 2016*, pp. 252–257, 2017, doi: 10.1109/iThings-GreenCom-CPSCom-SmartData.2016.66.

[17] T. H. Teng, A. H. Tan, J. A. Starzyk, Y. S. Tan, and L. N. Teow, "Integrating motivated learning and k-winner-take-all to coordinate multi-agent reinforcement learning," *Proceedings - 2014 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IAT 2014*, vol. 3, pp. 190–197, 2014, doi: 10.1109/WI-IAT.2014.167.

[18] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-supervised learning with deep generative models," *Adv Neural Inf Process Syst*, vol. 4, no. January, pp. 3581–3589, 2014.

[19] E. Levy, O. E. David, and N. S. Netanyahu, "Genetic algorithms and deep learning for automatic painter classification," *GECCO 2014 - Proceedings of the 2014 Genetic and Evolutionary Computation Conference*, no. Dl, pp. 1143–1150, 2014, doi: 10.1145/2576768.2598287.

[20] H. Kim, Y. Kim, and J. Hong, "Cluster management framework for autonomic machine learning platform," *Proceedings of the 2019 Research in Adaptive and Convergent Systems, RACS 2019*, pp. 128–130, 2019, doi: 10.1145/3338840.3355691.

[21] R. Rădulescu, P. Vrancx, and A. Nowé, "Analysing congestion problems in multi-agent reinforcement learning," *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 3, pp. 1705–1707, 2017.

[22] S. Kim, Y. K. Row, and T. J. Nam, "Thermal interaction with a voice-based intelligent agent," *Conference on Human Factors in Computing Systems - Proceedings*, vol. 2018-April, pp. 1–6, 2018, doi: 10.1145/3170427.3188656.

[23] J. Yan, D. Hu, S. S. Liao, and H. Wang, "Mining agents' goals in agent-oriented business processes," *ACM Trans Manag Inf Syst*, vol. 5, no. 4, 2015, doi: 10.1145/2629448.

[24] K. Hassani and W. S. Lee, "On designing migrating agents: From autonomous virtual agents to intelligent robotic systems," *SIGGRAPH Asia 2014 Autonomous Virtual Humans and Social Robot for Telepresence, SA 2014*, 2014, doi: 10.1145/2668956.2668963.

[25] D. Singh, L. Padgham, and B. Logan, "Integrating BDI Agents with Agent-Based Simulation Platforms," *Auton Agent Multi Agent Syst*, vol. 30, no. 6, pp. 1050–1071, 2016, doi: 10.1007/s10458-016-9332-x.

[26] A. Leite, R. Girardi, and P. Novais, "Using ontologies in hybrid software agent architectures," *Proceedings - 2013 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IATW 2013*, vol. 3, pp. 155–158, 2013, doi: 10.1109/WI-IAT.2013.172.

[27] R. Amin and S. Khalid, "Machine Learning Algorithms for Depression".

[28] A. Wilson and A. Fern, "Bayesian Role Discovery for (Extended Abstract)," *Learning*, pp. 1587–1588.

[29] M. E. Taylor, B. Kulis, and F. Sha, "Metric learning for reinforcement learning agents," *10th International Conference on Autonomous Agents and Multiagent Systems 2011, AAMAS 2011*, vol. 2, pp. 729–736, 2011.

[30] S. Hoet and N. Sabouret, "Reinforcement learning of communication in a multi-agent context," *Proceedings - 2011 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2011*, vol. 2, pp. 240–243, 2011, doi: 10.1109/WI-IAT.2011.125.

[31] C. Wu *et al.*, "Spectrum Management of Cognitive Radio Using Multi-agent Reinforcement Learning," in *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, 2010, pp. 10–14. [Online]. Available: www.ifaamas.org

[32] S. Bromuri, "A tensor factorization approach to generalization in multi-agent reinforcement learning," *Proceedings - 2012 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2012*, vol. 2, pp. 274–281, 2012, doi: 10.1109/WI-IAT.2012.21.

[33] W. T. L. Teacy *et al.*, "Decentralized Bayesian reinforcement learning for online agent collaboration," *11th International Conference on Autonomous Agents and Multiagent Systems 2012, AAMAS 2012: Innovative Applications Track*, vol. 1, pp. 312–319, 2012.

[34] X. Zhu, C. Zhang, and V. Lesser, "Combining dynamic reward shaping and action shaping for coordinating multi-agent learning," *Proceedings - 2013 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2013*, vol. 2, pp. 321–328, 2013, doi: 10.1109/WI-IAT.2013.127.

[35] L. Torrey and M. E. Taylor, "Teaching on a Budget: Agents advising agents in reinforcement learning," *12th International Conference on Autonomous Agents and Multiagent Systems 2013, AAMAS 2013*, vol. 2, pp. 1053–1060, 2013.

[36] C. Zhang and V. Lesser, "Coordinating multi-agent reinforcement learning with limited communication," *12th International Conference on Autonomous Agents and Multiagent Systems 2013, AAMAS 2013*, vol. 2, no. Aamas, pp. 1101–1108, 2013.

[37] W. Rand, "Machine Learning Meets Agent-Based Modelling: When Not To Go: When Not To Go To a Bar," 2006. [Online]. Available: https://ccl.northwestern.edu/papers/agent2006rand.pdf

[38] P. Mannion, K. Mason, S. Devlin, J. Duggan, and E. Howley, "Multi-objective dynamic dispatch optimisation using Multi-Agent Reinforcement Learning," *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, pp. 1345–1346, 2016.

[39] H. Wang *et al.*, "Integrating reinforcement learning with multi-agent techniques for adaptive service composition," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 12, no. 2, 2017, doi: 10.1145/3058592.

[40] A. Marinescu, I. Dusparic, and S. Clarke, "Prediction-based multi-agent reinforcement learning in inherently non-stationary environments," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 12, no. 2, 2017, doi: 10.1145/3070861.

[41] G. Henri and N. Lu, "A Multi-Agent Shared Machine Learning Approach for Real-time Battery Operation Mode Prediction and Control," *IEEE Power and Energy Society General Meeting*, vol. 2018-Augus, pp. 1–5, 2018, doi: 10.1109/PESGM.2018.8585907.

[42] P. Rosello and M. J. Kochenderfer, "Multi-agent reinforcement learning for multi-object tracking," *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 2, pp. 1397–1413, 2018.

[43] B. Khelifa and M. R. Laouar, "Multi-agent reinforcement learning for urban projects planning," *ACM International Conference Proceeding Series*, 2018, doi: 10.1145/3330089.3330134.

[44] H. Kazmi, J. Suykens, and J. Driesen, "Valuing knowledge, information and agency in multi-agent reinforcement learning: A case study in smart buildings: Industrial applications track," *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 1, pp. 585–587, 2018.

[45] P. Sunehag *et al.*, "Value-decomposition networks for cooperative multi-agent learning based on team

reward," *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 3, pp. 2085–2087, 2018.

[46] G. Palmer and K. Tuyls, "Lenient Multi-Agent Deep Reinforcement Learning," no. Aamas, pp. 443–451, 2018.

[47] J. Wang and L. Sun, "Dynamic holding control to avoid bus bunching: A multi-agent deep reinforcement learning framework," *Transp Res Part C Emerg Technol*, vol. 116, no. April, p. 102661, 2020, doi: 10.1016/j.trc.2020.102661.

[48] W. Amaral, G. Braz, L. Rivero, and D. Viana, "Using machine learning technique for effort estimation in software development," *ACM International Conference Proceeding Series*, 2019, doi: 10.1145/3364641.3364670.

[49] G. Palmer, R. Savani, and K. Tuyls, "Negative update intervals in deep multi-agent reinforcement learning," *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 1, pp. 43–51, 2019.

[50] R. B. Diddigi, K. J. Prabuchandran, D. Sai Koti Reddy, and S. Bhatnagar, "Actor-critic algorithms for constrained multi-agent reinforcement learning," *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 4, pp. 1931–1933, 2019.

[51] S. Bhalla, S. G. Subramanian, and M. Crowley, "Training cooperative agents for multi-agent reinforcement learning," *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 3, pp. 1826–1828, 2019.

[52] M. Kaushik, N. Singhania, S. Phaniteja, and K. M. Krishna, "Parameter sharing reinforcement learning architecture for multi agent driving," *ACM International Conference Proceeding Series*, pp. 0–6, 2019, doi: 10.1145/3352593.3352625.

[53] G. Bacchiani, D. Molinar, and M. Patander, "Microscopic traffic simulation by cooperative multi-agent deep reinforcement learning," *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 3, pp. 1547–1555, 2019.

[54] T. Molderez, B. Oeyen, C. De Roover, and W. De Meuter, "Marlon - a domain-specific language for multi-agent reinforcement learning on networks," *Proceedings of the ACM Symposium on Applied Computing*, vol. Part F1477, pp. 1322–1329, 2019, doi: 10.1145/3297280.3297413.

[55] M. Zhou *et al.*, "Factorized Q-Learning for Large-Scale Multi-Agent Systems," 2019.

[56] D. S. K. Reddy, A. Saha, S. G. Tamilselvam, P. Agrawal, and P. Dayama, "Risk averse reinforcement learning for mixed multi-agent environments," *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 4, no. 2, pp. 2171–2173, 2019.

[57] Y. Zhao and X. Ma, "Learning efficient communication in cooperative multi-agent environment," *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 4, pp. 2321–2323, 2019.

[58] W. Zhou, Y. Chen, and J. Li, "Competitive Evolution Multi-Agent Deep Reinforcement Learning," in *CSAE2019*, China, 2019.

[59] H. R. Lee and T. Lee, "Improved cooperative multi-agent reinforcement learning algorithm augmented by mixing demonstrations from centralized policy," *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 2, no. Aamas, pp. 1089–1098, 2019.

[60] M. Ossenkopf, M. Jorgensen, and K. Geihs, "Hierarchical multi-agent deep reinforcement learning to develop long-term coordination," *Proceedings of the ACM Symposium on Applied Computing*, vol. Part F1477, pp. 922–929, 2019, doi: 10.1145/3297280.3297371.

[61] J. Castellini, R. Savani, F. A. Oliehoek, and S. Whiteson, "The representational capacity of action-value networks for multi-agent reinforcement learning," *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 4, no. 1, pp. 1862–1864, 2019.

[62] H. Zhang *et al.*, "CityFlow: A multi-agent reinforcement learning environment for large scale city traffic scenario," *The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019*, pp. 3620–3624, 2019, doi: 10.1145/3308558.3314139.

[63] X. Li, J. Zhang, J. Bian, Y. Tong, and T. Y. Liu, "A cooperative multi-agent reinforcement learning framework for resource balancing in complex logistics network," *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 2, pp. 980–988, 2019.

[64] J. Hook, V. De Silva, and A. Kondoz, "Deep Multi-Critic Network for accelerating Policy Learning in multi-agent environments," *Neural Networks*, vol. 128, pp. 97–106, 2020, doi: 10.1016/j.neunet.2020.04.023.

[65] M. Uzair, L. Li, J. G. Zhu, and M. Eskandari, "A protection scheme for AC microgrids based on multi-agent system combined with machine learning," *2019 29th Australasian Universities Power Engineering Conference, AUPEC 2019*, pp. 17–22, 2019, doi: 10.1109/AUPEC48547.2019.211845.

[66] N. El Ghouch, M. Kouissi, and E. M. En-Naimi, "Multi-agent adaptive learning system based on incremental hybrid case-based reasoning (IHCBR)," *ACM International Conference Proceeding Series*, 2019, doi: 10.1145/3368756.3369030.

[67] J. Ma and F. Wu, "Feudal multi-agent deep reinforcement learning for traffic signal control," *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 2020-May, pp. 816–824, 2020.

[68] Y. Li, Y. Zheng, and Q. Yang, "Cooperative Multi-Agent Reinforcement Learning in Express System," *International Conference on Information and Knowledge Management, Proceedings*, pp. 805–814, 2020, doi: 10.1145/3340531.3411871.

[69] H. Mao, Z. Zhang, Z. Xiao, Z. Gong, and Y. Ni, *Learning multi-agent communication with double attentional deep reinforcement learning*, vol. 34, no. 1. Springer US, 2020. doi: 10.1007/s10458-020-09455-w.

[70] C. Hu, "A confrontation decision-making method with deep reinforcement learning and knowledge transfer for multi-agent system," *Symmetry (Basel)*, vol. 12, no. 4, pp. 1–24, 2020, doi: 10.3390/SYM12040631.

[71] O. Batata, V. Augusto, and X. Xie, "Mixed Machine learning and Agent-based Simulation for Respite Care Evaluation," in *Proceedings of the 2018 Winter Simulation Conference*, 2016, pp. 1–23.

[72] D. Dašić, M. Vučetić, M. Perić, M. Beko, and M. Stanković, "Cooperative Multi-Agent Reinforcement Learning for Spectrum Management in IoT Cognitive Networks," *ACM International Conference Proceeding Series*, vol. Part F1625, no. Cm, pp. 238–247, 2020, doi: 10.1145/3405962.3405996.

[73] J. Yang, I. Borovikov, and H. Zha, "Hierarchical cooperative multi-agent reinforcement learning with skill discovery," *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 2020-May, pp. 1566–1574, 2020.

[74] S. Gupta, R. Hazra, and A. Dukkipati, "Networked multi-agent reinforcement learning with emergent communication," *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 2020-May, no. i, pp. 1858–1860, 2020.

[75] D. E. Hostallero, D. Kim, S. Moon, K. Son, W. J. Kang, and Y. Yi, "Inducing cooperation through reward reshaping based on peer evaluations in deep multi-agent reinforcement learning," *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 2020-May, pp. 520–528, 2020.

[76] Z. Zhang, J. Yang, and H. Zha, "Integrating independent and centralized multi-agent reinforcement learning for traffic signal network optimization," *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 2020-May, pp. 2083–2085, 2020.

[77] D. Zelasko, P. Plawiak, and J. Kolodziej, "Machine learning techniques for transmission parameters classification in multi-agent managed network," *Proceedings - 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing, CCGRID 2020*, pp. 699–707, 2020, doi: 10.1109/CCGrid49817.2020.00-20.