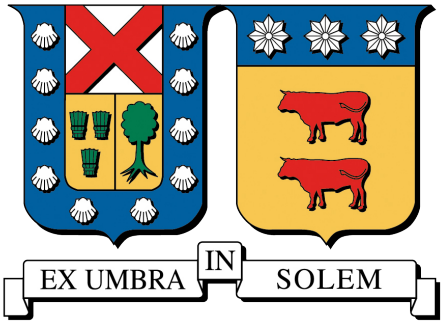
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA
VALPARAÍSO - CHILE



ESTRATEGIA DE INTERPRETACIÓN DE AGENTES DRL APLICADOS A REDES ÓPTICAS ELÁSTICAS

Presentado por:

JORGE ALBERTO BERMÚDEZ CEDEÑO

Como requisito para optar al grado de:

DOCTORADO EN INGENIERÍA ELECTRÓNICA

PROFESOR GUÍA : PhD. NICOLÁS JARA CARVALLO

AGOSTO 2025

CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☐ Memoria o trabajo de título ☒ Tesis de Postgrado

Título del trabajo: ESTRATEGIA DE INTERPRETACIÓN DE AGENTES DRL APLICADOS AREDES ÓPTICAS ELÁSTICAS

Nombre del candidato(a): Jorge Alberto Bermúdez Cedeño

Carrera / Grado: Doctorado en Ingeniería Electrónica

Campus: Casa Central **Departamento:** Electrónica

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Nicolás Alonso Jara Carvallo, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por **(marcar una opción)**:

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 04/12/2025 **Firma:**

Estudiante o Candidato(a):

Fecha: 28/11/2025 **Firma:**

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

AGRADECIMIENTOS

A la Universidad Técnica Federico Santa María y a la Agencia Nacional de investigación y Desarrollo, por brindarme la oportunidad y el apoyo necesario para cursar estudios de postgrado;

A los profesores del Departamento de Electrónica, por todo lo que me enseñaron en el desarrollo de sus clases;

A mi tutor, Dr. Nicolás Jara Carvallo, por todo su tiempo, dedicación, conocimiento y apoyo en la consecución del trabajo;

A mi familia y amigos, por todo su amor y cariño, por su apoyo incondicional, por darme fuerzas cuando más lo necesitaba, por confiar en mí cuando me faltó confianza;

A ustedes, muchas gracias por todo.

RESUMEN

La creciente complejidad de las redes ópticas elásticas (Elastic Optical Networks, EONs) y la necesidad de operar en entornos altamente dinámicos han impulsado el uso de técnicas basadas en inteligencia artificial, particularmente el Aprendizaje por Refuerzo Profundo (Deep Reinforcement Learning, DRL), como solución para problemas de gestión de recursos. Sin embargo, una de las principales barreras para su implementación práctica radica en su naturaleza de "caja negra", es decir, la dificultad para interpretar y justificar las decisiones tomadas por estos modelos. En este contexto, la presente tesis propone un marco metodológico que permite dotar de interpretabilidad a agentes DRL aplicados al problema RSA (Routing and Spectrum Allocation), uno de los desafíos centrales en redes ópticas modernas.

El enfoque desarrollado se basa en el uso de aprendizaje por imitación para entrenar clasificadores interpretables (como regresión lineal, regresión logística, árboles de decisión y bosques aleatorios) que emulan las políticas de decisión del agente DRL. Esta estrategia de tres etapas —entrenamiento del agente, imitación y análisis interpretativo— permite descomponer el comportamiento del modelo original en elementos comprensibles, identificando patrones, reglas y factores determinantes en la selección de rutas y asignación de espectro. El marco, además, integra un simulador que reproduce condiciones dinámicas de red, incluyendo impedimentos de la capa física y restricciones de calidad de transmisión (QoT), lo que permite evaluar los modelos en escenarios realistas.

Los resultados experimentales revelan que el subproblema de enrutamiento presenta mayor complejidad que la asignación espectral, evidenciado por una menor precisión en las predicciones de los clasificadores. Asimismo, todos los modelos imitadores tienden a reproducir consistentemente la elección de la primera ruta disponible, en línea con heurísticas tradicionales, mientras que la selección de rutas secundarias presenta un patrón más aleatorio, lo que sugiere una menor estructuración en esas decisiones por parte del agente. En cuanto a la importancia de características, se identificó que el tamaño del bloque espectral influye significativamente en la política del agente, mientras que atributos como el par origen-destino tienen escasa incidencia. Además, los clasificadores basados en árboles

de decisión destacaron la relevancia de las bandas centrales del espectro como zonas preferidas por el agente, posiblemente por su mayor flexibilidad operativa.

Este trabajo no solo permite interpretar modelos DRL en términos comprensibles, sino que también sienta las bases para construir versiones más ligeras, auditables y eficientes de estos agentes, facilitando su integración en entornos productivos y su validación en contextos críticos. El marco propuesto representa una contribución concreta al desarrollo de redes ópticas inteligentes y transparentes, promoviendo el uso de inteligencia artificial explicable en sistemas de telecomunicaciones avanzados. Finalmente, la investigación abre nuevas líneas de trabajo, como el diseño de modelos híbridos que combinen reglas interpretables con agentes DRL, y la extensión del enfoque a otros problemas de control y planificación en redes de próxima generación.

ABSTRACT

The increasing complexity of Elastic Optical Networks (EONs) and the need to operate in highly dynamic environments have driven the adoption of artificial intelligence–based techniques, particularly Deep Reinforcement Learning (DRL), as a promising solution for resource management problems. However, one of the main barriers to their practical deployment lies in their "black-box" nature, namely, the difficulty of interpreting and justifying the decisions made by such models. In this context, the present dissertation proposes a methodological framework designed to provide interpretability to DRL agents applied to the Routing and Spectrum Allocation (RSA) problem, one of the central challenges in modern optical networks.

The proposed approach relies on imitation learning to train interpretable classifiers (such as linear regression, logistic regression, decision trees, and random forests) that emulate the decision policies of the DRL agent. This three-stage strategy—agent training, imitation, and interpretative analysis—enables the decomposition of the original model's behavior into understandable elements, identifying patterns, rules, and key factors in route selection and spectrum assignment. The framework also integrates a simulator that reproduces dynamic network conditions, including physical-layer impairments and quality of transmission (QoT) constraints, thereby allowing the evaluation of models under realistic scenarios.

Experimental results reveal that the routing subproblem exhibits higher complexity than spectrum assignment, as evidenced by the lower accuracy of classifier predictions. Moreover, all surrogate models consistently reproduce the agent's preference for selecting the first available route, in line with traditional heuristics, whereas the selection of secondary routes displays a more random pattern, suggesting weaker structural decision-making by the agent in these cases. Regarding feature importance, it was found that spectral block size plays a significant role in the agent's policy, while attributes such as the source–destination pair have limited influence. Additionally, tree-based classifiers highlighted the relevance of central spectrum bands as preferred regions for the agent, possibly due to their greater operational flexibility.

This work not only enables the interpretation of DRL models in comprehensible terms but also lays the groundwork for building lighter, auditable, and more efficient versions of such agents, thereby facilitating their integration into production environments and their validation in critical contexts. The proposed framework constitutes a concrete contribution to the development of intelligent and transparent optical networks, fostering the adoption of explainable artificial intelligence in advanced telecommunication systems. Finally, the research opens new avenues for exploration, such as the design of hybrid models that combine interpretable rules with DRL agents, and the extension of the approach to other control and planning problems in next-generation networks.

Índice de Contenidos

Índice de Tablas

Índice de Figuras

1 | Introducción

Con la evolución constante de las tecnologías de comunicación y la aparición de nuevos servicios de red, la demanda global de tráfico de Internet ha experimentado un crecimiento exponencial en los últimos años Cisco (2018); López y Velasco (2016). Este fenómeno ha generado la necesidad de desarrollar arquitecturas de red más eficientes y escalables para gestionar el aumento continuo en los requerimientos de ancho de banda. Entre las soluciones propuestas, destacan las redes ópticas elásticas (EONs), las redes con multiplexación por división espacial (SDM) y las redes con multiplexación por división de banda (BDM) Mukherjee et al. (2020); Luo et al. (2017); Oliveira y da Fonseca (2017); Paz y Saavedra (2020), las cuales han sido diseñadas para operar tanto con tráfico estático como dinámico.

Estas nuevas arquitecturas superan en eficiencia y costos a las redes tradicionales basadas en la multiplexación por división de longitud de onda (WDM), al ofrecer una mejor asignación y aprovechamiento de los recursos ópticos disponibles. Sin embargo, su implementación no está exenta de desafíos. La mayor flexibilidad que proporcionan conlleva una complejidad adicional en la gestión, monitoreo y control de la red, lo que dificulta la automatización y aumenta la necesidad de intervención humana Ji et al. (2018). La creciente heterogeneidad de las redes ópticas y su constante evolución exigen nuevos enfoques para garantizar una operación eficiente y adaptativa en tiempo real.

Los métodos tradicionales de operación de redes ópticas se basan en reglas estáticas predefinidas, lo que resulta poco eficiente frente a las condiciones dinámicas de las redes modernas. La actualización de estas reglas requiere una intervención manual significativa y, en muchos casos, conlleva un rendimiento subóptimo Ouyang et al. (2025); Song

et al. (2025). Como alternativa, se han explorado métodos heurísticos y analíticos que mejoran la gestión de la red. Los métodos heurísticos aprovechan el conocimiento experto para encontrar soluciones aproximadas a problemas complejos, mientras que los métodos analíticos modelan estos problemas como tareas de optimización matemática.

No obstante, muchas de las tareas de control y gestión en redes ópticas son problemas NP-hard, lo que significa que su resolución es computacionalmente costosa y, en la práctica, inviable para escenarios en tiempo real. En el ámbito del monitoreo de redes ópticas, la percepción y estimación del estado de la red se basan comúnmente en modelos analíticos Tizikara et al. (2022), lo que impone un alto costo computacional y exige parámetros de red extremadamente precisos. En redes reales, estos parámetros pueden ser difíciles de obtener con exactitud, lo que introduce inexactitudes en el monitoreo y puede dar lugar a márgenes de error significativos o incluso a decisiones incorrectas en la operación de la red Pointurier (2017).

Para afrontar los desafíos de las redes ópticas del futuro, es esencial introducir un mayor nivel de inteligencia en sus procesos de monitoreo, control y gestión. La automatización de estas funciones permitirá reducir la dependencia de la intervención manual, mejorar la eficiencia operativa y aumentar la flexibilidad de la red Natalino et al. (2024). En este contexto, el uso de Machine Learning (ML) emerge como una solución prometedora, ya que permite a las redes ópticas aprender y adaptarse a entornos cambiantes sin necesidad de reprogramaciones constantes Cruzes (2025).

Los algoritmos de ML pueden resolver problemas complejos mediante un aprendizaje iterativo basado en datos históricos y en la retroalimentación del entorno. Si bien el entrenamiento de estos modelos puede ser costoso en términos de tiempo y recursos computacionales, este proceso puede llevarse a cabo anterior a la operación, permitiendo que los modelos preentrenados sean utilizados en línea para ejecutar cálculos en tiempo real con requisitos computacionales mínimos. Gracias a esta capacidad, el ML ha despertado un gran interés en la investigación aplicada a redes ópticas en los últimos años Yu et al. (2019); Chen et al. (2018); Martín et al. (2018).

Las arquitecturas actuales de redes ópticas pueden integrar técnicas de ML mediante la adopción de Redes Ópticas Definidas por Software (SDON), en las cuales los modelos

de ML pueden desempeñar funciones clave en los módulos de control. Esto permite una evolución hacia redes ópticas inteligentes, donde las decisiones pueden automatizarse y ajustarse dinámicamente según las necesidades de la red. Además, los módulos basados en ML pueden ser fácilmente añadidos, modificados o eliminados sin afectar la estabilidad de la red, lo que otorga una flexibilidad operativa sin precedentes.

A diferencia de las estrategias tradicionales basadas en reglas fijas y programación estática, las redes ópticas inteligentes potenciadas por ML pueden aprender y descubrir patrones ocultos en los datos de operación, lo que permite tomar decisiones informadas y eficientes en función de las condiciones de la red Amirabadi et al. (2024). Esta capacidad de adaptación continua facilita el uso eficiente de los recursos, la reducción de costos operativos y la mejora del rendimiento general del sistema.

Dentro de las técnicas de ML más prometedoras aplicadas a redes ópticas en los últimos años, se encuentran aquellas basadas en Aprendizaje por Refuerzo Profundo (Deep Reinforcement Learning, DRL). DRL ha mostrado resultados prometedores en problemas críticos como la asignación de espectro y enrutamiento (RSA), la estimación de la calidad de transmisión (QoT), la asignación de regeneradores y la detección de fallas en la red Mata et al. (2018); Zhang et al. (2020).

El aprendizaje por refuerzo es una subcategoría del ML que combina el aprendizaje profundo con el aprendizaje por refuerzo, permitiendo que un agente computacional aprenda a tomar decisiones eficientes mediante la interacción con su entorno y la retroalimentación de las consecuencias de sus acciones. Para ello, el agente emplea una red neuronal artificial (NN) de múltiples capas para transformar entradas en acciones, lo que le permite desarrollar estrategias eficientes a partir de la experiencia Luong et al. (2019).

Si bien la necesidad de sistemas automatizados capaces de tomar decisiones en tiempo real ha llevado a la integración de modelos de DRL, una de las principales barreras para su implementación en entornos operativos es la falta de transparencia en sus decisiones. La interpretabilidad de estos modelos no solo mejoraría la confianza en sus predicciones, sino que también facilitaría su validación, diagnóstico y mantenimiento Cheng et al. (2025). En redes ópticas, las decisiones automatizadas afectan la asignación de recursos críticos, como el espectro óptico, el enrutamiento y la gestión de fallos. Sin una adecuada inter-

pretabilidad, los operadores de red pueden enfrentarse a un efecto “caja negra”, donde las decisiones del modelo no pueden ser comprendidas ni justificadas. Esta opacidad puede generar incertidumbre en la operación de la red, dificultando la implementación de acciones correctivas cuando el rendimiento de la red no es el esperado. Además, en escenarios donde se requiere cumplir con regulaciones o normativas específicas, la falta de explicaciones claras sobre el comportamiento del modelo puede impedir su certificación y despliegue en infraestructuras críticas.

Una de las razones por las cuales la interpretabilidad es fundamental en redes ópticas es la necesidad de garantizar la confiabilidad y robustez de las decisiones automatizadas Glanois et al. (2021). En un entorno de red altamente dinámico, donde las condiciones del tráfico y la calidad del enlace pueden cambiar rápidamente, es esencial que los modelos de DRL sean capaces de explicar por qué se tomó una decisión específica y qué factores influyeron en ella. Si un modelo asigna una ruta óptica específica o cambia la asignación de espectro de manera inesperada, los operadores deben entender si la decisión fue tomada debido a congestión en otros enlaces, a degradación en la calidad de transmisión (Quality of Transmission, QoT) o a la necesidad de optimizar el balance de carga en la red.

Otro aspecto crítico de la interpretabilidad en DRL es la capacidad de depuración y corrección de errores en los modelos. Dado que el proceso de aprendizaje en DRL se basa en la interacción continua con el entorno, es posible que el modelo aprenda patrones incorrectos o sesgados en función de los datos de entrenamiento disponibles. Sin una capacidad de interpretación clara, detectar y corregir estos sesgos se convierte en una tarea extremadamente difícil Vouros (2022). Por ejemplo, si un modelo de DRL tiende a favorecer ciertas rutas en la red sin una justificación clara, podría generar cuellos de botella innecesarios o incluso fallos en la infraestructura debido a una utilización desigual de los recursos ópticos. Contar con herramientas que permitan analizar y explicar el comportamiento del modelo facilitaría la identificación de estos problemas y permitiría realizar ajustes en el entrenamiento del agente de aprendizaje.

Además, la interpretabilidad también juega un papel clave en la aceptación y adopción de DRL por parte de los operadores de red y tomadores de decisiones. En muchas organizaciones, la implementación de sistemas basados en inteligencia artificial enfrenta

resistencia debido a la falta de confianza en las decisiones automatizadas Cheng et al. (2025). Los operadores experimentados suelen preferir métodos tradicionales de gestión de redes, donde las reglas y procedimientos son explícitos y comprensibles. En este sentido, si un modelo de DRL puede proporcionar explicaciones claras sobre sus decisiones y demostrar su fiabilidad en la optimización de la red, es más probable que sea aceptado y utilizado en entornos operativos Glanois et al. (2021). De hecho, la interpretación de estos modelos no solo contribuye a generar confianza, sino que también permite extraer información relevante que puede transformarse en heurísticas útiles para la gestión de la red, lo cual refuerza la necesidad de avanzar en metodologías de interpretabilidad.

Desde una perspectiva de resiliencia de la red, la interpretabilidad también permite mejorar la respuesta ante fallos y eventos inesperados Cheng et al. (2025). En una red óptica, los fallos pueden deberse a múltiples factores, como la degradación de la fibra, problemas en los amplificadores ópticos o congestión excesiva en algunos enlaces. Si un modelo de DRL detecta un fallo y sugiere una acción de recuperación, es crucial que los operadores puedan entender la razón detrás de esta decisión. De lo contrario, la falta de confianza en el modelo podría llevar a los operadores a descartar sus recomendaciones, reduciendo el impacto positivo que el DRL podría tener en la resiliencia de la red.

Un ejemplo concreto de cómo la falta de interpretabilidad puede afectar la operación de una red óptica es el caso de la predicción de fallos en enlaces ópticos. Supongamos que un modelo de DRL detecta que un enlace está en riesgo de fallo y recomienda redirigir el tráfico hacia rutas alternativas. Si los operadores no pueden entender por qué se tomó esta decisión, podrían dudar en ejecutarla, especialmente si la acción recomendada implica reasignar una gran cantidad de tráfico y afectar potencialmente la calidad del servicio de otros clientes. En cambio, si el modelo puede proporcionar una explicación clara, como "se ha detectado una degradación progresiva en la calidad del enlace debido a un aumento en la atenuación óptica", los operadores pueden validar la predicción y tomar medidas proactivas para evitar una interrupción del servicio.

En consecuencia, además del uso del ancho de banda y de mejorar la eficiencia operativa, las redes ópticas basadas en DRL deben garantizar que sus decisiones sean comprensibles y justificables. La interpretabilidad de los modelos es un requisito esencial

para validar la confiabilidad de las soluciones y asegurar que las redes ópticas del futuro sean no solo más inteligentes, sino también más transparentes y confiables en su operación.

Esta tesis se centra en la interpretación de agentes de Deep Reinforcement Learning (DRL) entrenados para abordar problemas fundamentales en redes ópticas, con el objetivo de mejorar su aplicabilidad y confianza en entornos operativos, proporcionando explicaciones claras sobre sus decisiones y los factores que influyen en su proceso de toma de decisiones. Al abordar la interpretabilidad en agentes de DRL aplicados a redes ópticas, esta investigación no solo contribuirá a mejorar la confianza y adopción de estos modelos, sino que también sentará las bases para el desarrollo de redes ópticas inteligentes más transparentes y adaptables. La integración de métodos interpretables en DRL permitirá avanzar hacia una nueva generación de redes que sean capaces de operar de manera autónoma sin comprometer la supervisión y el control por parte de los expertos en redes.

En base a todo lo anterior, la pregunta de investigación de este trabajo de tesis es: **¿Cómo interpretar las decisiones de agentes de Deep Reinforcement Learning aplicados a la gestión de recursos en redes ópticas, de manera que se incremente la transparencia y confianza en sus resultados?**

1.1. Objetivos

1.1.1. Objetivo General

El objetivo general de este trabajo es: “Desarrollar y evaluar métodos de interpretabilidad para agentes de Deep Reinforcement Learning (DRL) aplicados a redes ópticas, con el fin de proporcionar explicaciones claras y comprensibles sobre sus decisiones, mejorando la confianza, supervisión y aplicabilidad de estos modelos en entornos operativos. ”.

1.1.2. Objetivos Específicos

1. Analizar, proponer y evaluar nuevos modelos de DRL aplicados a redes ópticas.
2. Identificar patrones, limitaciones y perspectivas explicativas basados en los resultados de la interpretación de los modelos DRL aplicados a redes ópticas.

3. Sintetizar los resultados de la interpretación en conclusiones prácticas que puedan orientar las mejoras en la asignación de recursos basada en DRL en redes ópticas.
4. Desarrollar una herramienta de trabajo que permita entrenar e interpretar agentes DRL aplicados a redes ópticas elásticas.

1.2. Aportes del trabajo de tesis

Este trabajo de tesis se enfoca en la interpretación de agentes DRL aplicados a redes ópticas. Entre los principales aportes del trabajo, destacan:

- Propuesta de una estrategia o marco de trabajo para la interpretación de agentes DRL. Se establecerán lineamientos y mejores prácticas para la implementación de agentes DRL interpretables en redes ópticas, considerando criterios de evaluación, validación y despliegue en entornos reales.
- Disponibilización de un framework para la simulación, evaluación e interpretación de agentes DRL. Durante el presente trabajo, se ha elaborado un framework que permite simular en redes ópticas dinámicas distintas estrategias para abordar problemas comunes en redes ópticas, así como la creación e interpretación de modelos de DRL.
- Evaluación del impacto de la interpretabilidad en la eficiencia de la red. Se analizará cómo la incorporación de métodos interpretables afecta el rendimiento de los modelos de DRL en términos de mejorar el uso del espectro y calidad de transmisión, estableciendo una relación entre transparencia y eficiencia.

1.3. Estructura de la tesis

El resto de esta tesis está estructurado de la siguiente manera: en el Capítulo 2 se realiza un análisis de los métodos y estrategias de interpretación comúnmente utilizados. Luego, en el Capítulo 3 se analiza y se propone una nueva estrategia de interpretación de modelos de DRL. Seguidamente, en el Capítulo 4 se muestran y analizan los resultados obtenidos, realizando comparaciones y observaciones sobre los mismos. Finalmente, en el Capítulo 5, se presentan las conclusiones del trabajo.

Agradecimientos

Este trabajo recibió apoyo financiero de ANID Fondecyt 1250775 y 11251432, del Programa de Becas ANID / DOCTORADO BECAS CHILE/2021 – 21210519, y de la Universidad Técnica Federico Santa María (USM) bajo el proyecto PI_LII_24_15.

Resultados obtenidos

En resumen, las publicaciones realizadas durante el período de tesis son:

Revistas científicas (Journals)

[1] J. Bermúdez, P. Guicharrousse-Vargas, H. Pempelfort, A. Leiva, R. Olivares, and N. Jara, "Efficient Band Management in Multiband Elastic Optical Networks: A Threshold-based Approach", *Journal of Optical Communications and Networking*, status: sent.

[2] N. Jara, J. Bermúdez, P. Morales, H. Pempelfort, R. Olivares, D. Bórquez-Paredes, A. Leiva, "Rethinking Band Management in Multiband Elastic Optical Networks: From Heuristics to AI-Driven Methods", *IEEE Communications Magazine*, status: sent.

[3] J. Bermúdez, P. Morales, H. Pempelfort, M. Araya, and N. Jara, "Understanding Deep Reinforcement Learning: Enhancing Explainable Decision-Making in Optical Networks", *ICT Express*, doi: 10.1016/j.icte.2025.08.002, 2025.

[4] J. Bermúdez, R. Vallejos, and N. Jara, "A Bandwidth-Balanced RMLSA Solution for Static Elastic Optical Network: A Two Stages Approach", *IEEE Access*, doi: 10.1109/ACCESS.2022.3188989, vol. 10, pp. 80092-80105, 2022.

Congresos internacionales (International Conferences)

[1] J. Chaffe, J. Bermúdez, P. Morales, N. Jara, "Enhancing Fault Tolerance in Optical Networks with DRL-Based 1+1 Protection: A Resilience-Driven Approach", 13th International Conference on Mathematical Methods in Reliability (MMR 2025), Viña del Mar, Chile, Jun 2025.

[2] N. Jara, J. Bermúdez, P. Morales, H. Pempelfort, R. Olivares, and A. Leiva, “Multiband Elastic Optical Networks: Comprehensive Insights into Band Resource Management”, *ONDM*, doi: 10.23919/ONDM65745.2025.11029339, pp. 1-6, 2025.

[3] J. Bermúdez, H. Pempelfort, P. Morales, M. Araya, and N. Jara, “Deciphering Deep Reinforcement Learning: Towards Explainable Decision-Making in Optical Networks”, *HPSR*, doi: 10.1109/HPSR62440.2024.10635946, pp. 80-86, 2024.

2 | Estado del arte

2.1. Necesidad de automatización e interpretabilidad

El crecimiento sostenido del tráfico de datos, impulsado por aplicaciones como el video bajo demanda, los servicios en la nube y el Internet de las Cosas (IoT), ha generado una presión significativa sobre las redes ópticas actuales, acercándolas a su límite físico de capacidad, fenómeno conocido como Capacity Crunch Won (2025). Ante esta situación, la expansión de la infraestructura mediante la instalación de nuevas fibras ópticas y equipamiento asociado representa una solución directa, pero económicamente inviable y operacionalmente compleja. En respuesta, la comunidad científica ha desarrollado diversas estrategias orientadas a mejorar el uso de los recursos ya desplegados, tales como la multiplexación espacial a través de fibras multicore y la expansión del espectro óptico hacia bandas adyacentes Yuan et al. (2024). Si bien estas alternativas incrementan el potencial físico de transmisión, su aprovechamiento eficiente requiere una gestión de recursos más flexible, dinámica y autónoma.

Históricamente, la operación de las redes ópticas ha estado basada en configuraciones estáticas y procedimientos manuales Andriolli et al. (2022). La asignación fija de canales espectrales, generalmente de 50 GHz conforme a los estándares de la Unión Internacional de Telecomunicaciones, la segmentación secuencial del problema de planificación, considerando enrutamiento, selección de modulación y asignación espectral de forma independiente, y el uso de heurísticas como First-Fit o Best-Fit han dado lugar a esquemas operativos limitados en su capacidad de adaptación. Aunque se han introducido paradigmas como las Redes Ópticas Elásticas (EON) y las plataformas de Redes Definidas por Software

(SDN), que habilitan cierto grado de reconfiguración y control centralizado, estos enfoques no alcanzan por sí solos el nivel de automatización necesario para enfrentar la complejidad operativa y la variabilidad del tráfico de las redes ópticas de nueva generación Cruzes (2023).

En este contexto, la integración de técnicas de Inteligencia Artificial, particularmente el Aprendizaje por Refuerzo Profundo (DRL), ha sido considerada una alternativa prometedora para dotar a las redes ópticas de capacidades de decisión autónoma y adaptativa Doherty et al. (2025). El enfoque DRL permite a un agente aprender políticas de asignación de recursos mediante la interacción directa con el entorno, mejorando su comportamiento en función de señales de recompensa acumulada. Esta propiedad es especialmente adecuada para abordar problemas de planificación en redes dinámicas, donde las condiciones de tráfico, topología y disponibilidad de recursos varían de forma no determinista. Así, las denominadas Redes Ópticas Inteligentes, habilitadas por SDN y potenciadas por DRL, se perfilan como una solución viable para lograr una operación más eficiente, resiliente y escalable«Pinto-Ríos et al. (2023); Terki et al. (2024).

Sin embargo, la adopción práctica de modelos basados en DRL enfrenta una limitación crítica: la falta de interpretabilidad de sus decisiones. A diferencia de los enfoques heurísticos o basados en optimización matemática, cuyas reglas de decisión son explícitas y rastreables, los modelos DRL tienden a comportarse como sistemas opacos, donde las decisiones son producto de redes neuronales profundas que no ofrecen una explicación comprensible de su razonamiento interno. Esta característica representa un obstáculo significativo en entornos operacionales donde la trazabilidad, la confiabilidad y la justificación técnica de cada decisión son requisitos esenciales, especialmente en casos de fallos de red o ante exigencias regulatorias Cheng et al. (2025); Vouros (2022). Incluso tras procesos extensos de entrenamiento, no existe plena garantía de que las decisiones del agente sean consistentes o estén alineadas con los objetivos de la red. Por ello, transformar las buenas prácticas de los agentes DRL en protocolos interpretables que permitan auditar, validar y ajustar el comportamiento de los agentes constituye un paso indispensable para lograr un desempeño sostenido, confiable y aplicable a redes ópticas en entornos reales.

Diversos trabajos han intentado aplicar DRL a problemas de planificación óptica, con

resultados dispares. Algunos agentes han sido entrenados para resolver subproblemas específicos como el enrutamiento, logrando mejoras en entornos controlados o de baja escala. Sin embargo, cuando se abordan problemas más complejos, como la asignación conjunta de espectro, banda y núcleo en arquitecturas multibanda o multicore, estos modelos frecuentemente no superan a heurísticas tradicionales como K-Shortest Path + First-Fit Doherty et al. (2025). Además, la mayoría de estos enfoques han sido validados únicamente en topologías pequeñas y bajo supuestos ideales de tráfico, lo que limita su aplicabilidad en escenarios reales. A esto se suma la falta de marcos metodológicos que permitan evaluar, depurar y validar las políticas aprendidas de manera sistemática Etezadi et al. (2023).

En consecuencia, la necesidad de dotar a los modelos DRL de capacidades explicativas se vuelve prioritaria. La interpretabilidad no solo facilita su validación técnica, sino que además permite su integración en flujos de operación donde la supervisión humana sigue siendo necesaria. En esta línea, se han propuesto enfoques como la destilación de políticas hacia modelos más simples y comprensibles, o el uso de técnicas de aprendizaje por imitación para extraer reglas desde agentes expertos. No obstante, estas metodologías aún no han sido plenamente desarrolladas ni aplicadas al contexto específico de redes ópticas, donde las decisiones deben considerar múltiples restricciones físicas y de calidad de servicio.

En síntesis, si bien las Redes Ópticas Inteligentes basadas en DRL representan una solución de alto potencial para abordar los desafíos de eficiencia, adaptabilidad y escalabilidad en la gestión de recursos, su despliegue efectivo exige avanzar en mecanismos que garanticen la interpretabilidad, confiabilidad y generalización de las políticas aprendidas. La automatización sin comprensión no es viable en sistemas críticos como las redes ópticas; por tanto, desarrollar modelos DRL explicables y adaptativos constituye un paso imprescindible hacia su adopción real en entornos productivos.

2.2. Interpretación de algoritmos de Machine Learning

Se dice que un modelo de aprendizaje por refuerzo (RL) es interpretable si todos sus componentes son inteligibles, incluyendo las entradas y su procesamiento, los modelos de transición y preferencia, y el modelo de toma de decisiones (por ejemplo, la política y las

funciones de valor) Glanois et al. (2021). En este sentido, se han propuesto diversas técnicas de interpretación para comprender los algoritmos de RL. A continuación, analizamos la interpretación de algunos algoritmos de aprendizaje automático (ML) comúnmente utilizados, así como diferentes métricas de categorización en DRL. Finalmente, presentamos algunos métodos de interpretación existentes en la literatura.

2.2.1. Regresión Lineal

La regresión lineal es una técnica de aprendizaje automático utilizada para modelar la dependencia de un objetivo en función de algunas características. Ha sido empleada por científicos informáticos, estadísticos y otros investigadores para resolver diversos problemas. En este modelo, las relaciones aprendidas para una única instancia se representa como la suma ponderada de las características presentadas en la entrada de la siguiente manera: Hastie et al. (2009):

$$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p + \epsilon \tag{2.1}$$

Una instancia se predice como la suma ponderada de p características. Los coeficientes beta (β_j) representan los pesos aprendidos de las características. En la ecuación 2.1, β_0 se conoce como el interceptor (intercept), la variable epsilon (ϵ) representa el error de predicción (es decir, la diferencia entre el resultado real y la predicción) y se supone que sigue una distribución Gaussiana Seber y Lee (2012). Seber y Lee (2012).

La linealidad de los modelos de regresión lineal hace que las tareas de estimación sean sencillas y, lo más importante, que los pesos de las ecuaciones lineales sean fáciles de comprender, lo que facilita la interpretación del modelo. Por esta razón, estos modelos son ampliamente utilizados en campos de investigación cuantitativa como la sociología, la psicología y la medicina.

La estimación de los pesos se realiza dentro de un intervalo de confianza específico (generalmente del 95 %), dentro del cual se espera que se encuentre el valor estimado de cada peso. En un modelo de regresión lineal bien definido, si se realizan 100 mediciones, el intervalo de confianza incluiría el peso correcto en 95 de ellas.

Interpretación

La interpretación de los pesos en un modelo de regresión lineal depende del tipo de característica Weisberg (2005).

- Característica numérica: Aumentar la característica numérica en una unidad modifica el resultado estimado en la magnitud de su peso. Un ejemplo de una característica numérica es el tamaño de una casa.
- Característica binaria: Una característica que toma uno de dos valores posibles para cada instancia. Un ejemplo es la característica "La casa tiene jardín". Uno de los valores actúa como categoría de referencia (en algunos lenguajes de programación, codificado con 0), como "Sin jardín". Cambiar la característica de la categoría de referencia a la otra categoría modifica el resultado estimado en la magnitud del peso de la característica.
- Característica categórica con múltiples categorías: Una característica con un número fijo de valores posibles. Un ejemplo es la característica "tipo de suelo", con categorías posibles como "alfombra", "laminado" y "parquet". Una solución para manejar múltiples categorías es el *one-hot encoding*, donde cada categoría tiene su propia columna binaria. Para una característica categórica con L categorías, solo se necesitan $L-1$ columnas, ya que la L-ésima categoría puede inferirse a partir de las demás (por ejemplo, si todas las columnas del 1 al $L-1$ tienen valor 0 para una instancia, se sabe que la característica toma la categoría L). La interpretación para cada categoría es la misma que en las características binarias. Algunos lenguajes, como R, permiten codificar características categóricas de diferentes maneras, como se describe más adelante en este capítulo.
- Intersección β_0: La intersección es el peso asignado a la "característica constante", que siempre tiene un valor de 1 para todas las instancias. La mayoría de los paquetes de software agregan automáticamente esta característica para estimar la intersección. Su interpretación es la siguiente: para una instancia donde todas las características numéricas tienen un valor de cero y las características categóricas están en sus categorías de referencia, la predicción del modelo corresponde al peso de la intersección.

Sin embargo, la interpretación de la intersección suele ser irrelevante, ya que es poco común encontrar instancias en las que todas las características sean exactamente cero. La interpretación adquiere más sentido cuando las características han sido estandarizadas (media de cero y desviación estándar de uno), ya que en ese caso la intersección representa el resultado predicho para una instancia donde todas las características están en su valor medio.

Importancia de las características

El valor absoluto de sus *t* estadísticas puede medir la importancia de una característica en un modelo de regresión lineal. El estadístico *t* es el peso estimado escalado con su error estándar

$$t_{\hat{\beta}_j} = \frac{\hat{\beta}_j}{SE(\hat{\beta}_j)} \tag{2.2}$$

donde $\hat{\beta}_j$ es el coeficiente estimado para la característica j, y $SE(\hat{\beta}_j)$ es el error estándar del coeficiente. A partir de 2.2, se puede entender que la importancia de una característica aumenta a medida que su peso estimado es mayor. Sin embargo, cuanto mayor sea la varianza del peso estimado (o menor sea la certeza sobre su valor correcto), menor será la importancia de la característica Yan y Su (2009).

2.2.2. Regresión Logística

La regresión logística puede modelar las probabilidades en problemas de clasificación con dos posibles resultados. Es una extensión del modelo de regresión lineal para problemas de clasificación. En lugar de ajustar una línea recta o un hiperplano, el modelo de regresión logística utiliza la función logística para restringir la salida de una ecuación lineal dentro del rango de 0 a 1. El algoritmo de regresión logística se define de la siguiente manera: Hastie et al. (2009):

$$P(y^{(i)} = 1) = \frac{1}{1 + exp(-(\beta_0 + \beta_1 x_1^{(i)} + \ldots + \beta_p x_p^{(i)}))} \tag{2.3}$$

En la ecuación 2.3, $P(y^{(i)} = 1)$ es la probabilidad de que la observación i pertenezca a la clase 1, β_0 es el intercepto o bias, $\beta_1, ..., \beta_p$ son los coeficientes del modelo logístico, y $x_1^{(i)}, ..., x_p^{(i)}$ son los valores de las características.

Interpretación

La interpretación de los pesos en la regresión logística difiere de la regresión lineal, ya que el resultado en la regresión logística es una probabilidad entre 0 y 1. Los pesos ya no influyen en la probabilidad de manera lineal. En su lugar, la función logística transforma la suma ponderada en una probabilidad. Sin embargo, la linealidad de la ecuación 2.3 se puede lograr considerando el valor logarítmico de la probabilidad de que ocurra un evento dividido por la probabilidad de que no ocurra (conocido como odds), de la siguiente manera: Hilbe (2009):

$$log(odds) = log\left(\frac{P(y=1)}{1-P(y=1)}\right) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p \tag{2.4}$$

Luego, comparamos lo que sucede cuando aumentamos en 1 el valor de una de las características. Sin embargo, en lugar de analizar la diferencia, observamos la razón entre las dos predicciones:

$$\frac{odds_{x_j+1}}{odds} = exp\left(\beta_j(x_j+1) - \beta_j x_j\right) = exp\left(\beta_j\right) \tag{2.5}$$

En última instancia, obtenemos una expresión tan simple como exp() del peso de una característica. Un cambio de una unidad en una característica modifica la razón de probabilidades de manera multiplicativa por un factor de $\exp(\beta_j)$. También se puede interpretar de la siguiente manera: un incremento de una unidad en x_j aumenta el logaritmo de la razón de probabilidades en el valor del peso correspondiente. Por ejemplo, si la razón de probabilidades (odds) es 2, la probabilidad de $y = 1$ es el doble que la de $y = 0$. Si el peso (log de la razón de probabilidades) es 0.7, entonces aumentar la característica respectiva en una unidad multiplica la razón de probabilidades por exp(0,7) (aproximadamente 2), lo que hace que las odds cambien a 4.

Finalmente, la interpretación del modelo de regresión logística para diferentes tipos de características se puede describir de la siguiente manera Hosmer Jr et al. (2013):

- Característica numérica: Si se incrementa el valor de la característica x_j en una unidad, la razón de probabilidades (odds) estimada cambia por un factor de $\exp(\beta_j)$.
- Característica categórica binaria: Uno de los dos valores de la característica actúa como la categoría de referencia (en algunos lenguajes de programación, el valor codificado como 0). Cambiar la característica x_j de la categoría de referencia a la otra categoría modifica la razón de probabilidades estimada por un factor de $\exp(\beta_j)$.
- Característica categórica con más de dos categorías: Una solución para manejar múltiples categorías es el one-hot encoding, donde cada categoría tiene su propia columna. Para una característica categórica con L categorías, solo se necesitan $L-1$ columnas; de lo contrario, el modelo estaría sobreparametrizado. La L-ésima categoría actúa como la categoría de referencia. La interpretación para cada categoría es equivalente a la interpretación de las características binarias.
- Intersección β_0: Cuando todas las características numéricas son cero y las características categóricas están en la categoría de referencia, la razón de probabilidades (odds) estimada es $\exp(\beta_0)$. La interpretación del peso de la intersección generalmente no es relevante.

2.2.3. Árboles de decisión

Los modelos basados en árboles dividen los datos varias veces según valores de corte específicos en las características. A través de estas divisiones, se crean diferentes subconjuntos del conjunto de datos, donde cada instancia pertenece a un subconjunto Hastie et al. (2009). Los subconjuntos finales se denominan nodos terminales o nodos hoja, mientras que los subconjuntos intermedios se conocen como nodos internos o nodos de división. Para predecir el resultado en cada nodo hoja, se utiliza el valor promedio de los datos de entrenamiento en ese nodo Maimon y Rokach (2014).

Existen varios algoritmos que pueden construir un árbol. Se diferencian en la estructura posible del árbol (por ejemplo, el número de divisiones por nodo), los criterios

para encontrar las divisiones, el momento en que se debe detener la división y la forma de estimar los modelos simples dentro de los nodos hoja. El algoritmo de Árboles de Clasificación y Regresión (CART) es probablemente el más popular para la inducción de árboles de decisión. La ecuación 2.6 representa la relación entre el resultado y, y las características x en un árbol de decisión CART.

$$\hat{y} = \hat{f}(x) = \sum_{m=1}^{M} c_m I\{x \in R_m\} \tag{2.6}$$

Cada instancia cae exactamente en un nodo hoja (subconjunto R_m). La función indicatriz $I_{\{x \in R_m\}}$ devuelve 1 si x pertenece al subconjunto R_m y 0 en caso contrario. Si una instancia cae en un nodo hoja R_l, el resultado predicho es $\hat{y} = c_l$, donde c_l es el promedio de todas las instancias de entrenamiento en el nodo hoja R_l Hastie et al. (2009).

El algoritmo CART toma una característica y determina qué valor de corte minimiza la varianza de y en una tarea de regresión o el índice de Gini en una tarea de clasificación. La Varianza indica qué tan dispersos están los valores de y en un nodo con respecto a su media. El índice de Gini mide la impureza de un nodo; por ejemplo, si todas las clases tienen la misma frecuencia, el nodo es impuro. Si solo hay una clase presente, el nodo es completamente puro. La varianza y el índice de Gini se minimizan cuando los puntos de datos en los nodos tienen valores de y muy similares.

En consecuencia, el mejor punto de corte es aquel que hace que los dos subconjuntos resultantes sean lo más diferentes posible en relación con el resultado objetivo Maimon y Rokach (2014). Para características categóricas, el algoritmo intenta crear subconjuntos probando diferentes combinaciones de categorías. Una vez determinado el mejor punto de corte para una característica, el algoritmo selecciona la característica para dividir, considerando la varianza o el índice de Gini. Luego, agrega esta división al árbol. El algoritmo continúa este proceso de búsqueda y división de forma recursiva en los nuevos nodos hasta que se alcanza un criterio de parada. Algunos criterios de parada posibles incluyen el número mínimo de instancias que deben estar en un nodo antes de realizar una división y el número mínimo de instancias que deben estar en un nodo terminal Joshi (2020).

Interpretación

La interpretación es sencilla: comenzando desde el nodo raíz, se avanza a los siguientes nodos según las condiciones establecidas en los bordes, que indican a qué subconjunto de datos se está accediendo. Una vez que se llega a un nodo hoja, este proporciona el resultado predicho. Todas las condiciones en los bordes están conectadas por 'AND', siguiendo la lógica siguiente: Si la característica x es [menor/mayor] que el umbral c Y ..., entonces el resultado predicho es el valor medio de y en ese nodo Perner (2011).

Importancia de las características

La importancia de una característica en un árbol de decisión se puede calcular de la siguiente manera: Recorrer todas las divisiones en las que se utilizó la característica. Medir cuánto ha reducido la varianza o el índice de Gini en comparación con el nodo padre. Sumar todas las contribuciones de importancia y escalarlas a un total de 100. De esta manera, cada importancia se puede interpretar como la proporción de la importancia total del modelo Kazemitabar et al. (2017).

Descomposición del árbol de decisión

Las predicciones individuales de un árbol de decisión pueden explicarse descomponiendo la ruta de decisión en un componente por característica. Se puede rastrear una decisión a lo largo del árbol y explicar una predicción mediante las contribuciones agregadas en cada nodo de decisión. El nodo raíz en un árbol de decisión es el punto de partida. Si solo se usara el nodo raíz para hacer predicciones, este devolvería el valor medio del resultado en los datos de entrenamiento. Con cada división subsiguiente, se suma o resta un nuevo término dependiendo del siguiente nodo en la ruta. Para obtener la predicción final, se debe seguir la ruta de la instancia de datos que se desea explicar, agregando sucesivamente las contribuciones en la fórmula Molnar (2020):

$$\hat{f}(x) = \bar{y} + \sum_{d=1}^{D} \text{split.contrib(d,x)} = \bar{y} + \sum_{j=1}^{p} \text{feat.contrib(j,x)} \tag{2.7}$$

La predicción de una instancia individual es la media del resultado objetivo más la

suma de todas las contribuciones de las D divisiones entre el nodo raíz y el nodo terminal donde termina la instancia. Sin embargo, el interés no radica en las contribuciones de cada división, sino en las contribuciones de cada característica. Una característica puede ser utilizada en más de una división o no ser utilizada en absoluto. Por lo tanto, se pueden sumar las contribuciones de cada una de las p características e interpretar su impacto en la predicción final.

2.3. Métricas de categorización

Los métodos de interpretación de aprendizaje reforzado se pueden categorizar en grupos Molnar (2020) atendiendo a:

1. **Tipo de explicación**: La interpretación se puede realizar de tres maneras diferentes Carvalho et al. (2019): utilizando otro método para generar explicaciones sin modificar el modelo original (Post-Hoc), introducir un nuevo modelo de aprendizaje interpretable en lugar del original (Intrínseca), o modificando el modelo original para hacerlo más interpretable sin reemplazarlo completamente (Combinada).

2. **Alcance de la interpretación**: Las interpretaciones de modelos de aprendizaje por refuerzo (RL) pueden explicar el comportamiento completo del modelo o su estrategia de toma de decisiones, en cuyo caso se denominan interpretaciones globales. Dado que reemplazar un modelo de caja negra por un nuevo modelo interpretable se aplica a todo el modelo y no solo a un subconjunto de muestras, la mayoría de los métodos intrínsecos generan interpretaciones globales. Sin embargo, también es posible que las interpretaciones se refieran solo a un subconjunto de acciones elegidas por el agente RL, lo que se conoce como interpretaciones locales Molnar (2020).

3. **Formato de la explicación**: El resultado de un método de interpretación puede generarse en diferentes formatos Chakraborti et al. (2019), como explicaciones textuales en lenguaje natural, explicaciones visuales mediante imágenes, listas de pares estado-acción o un conjunto de reglas que los humanos pueden aplicar para comprender la decisión final del agente.

4. **Fidelidad y exactitud**: La fidelidad se refiere al grado en que una interpretación

refleja con precisión el modelo real Kazhdan et al. (2020). Antes de interpretar un modelo, es fundamental asegurarse de que la interpretación explique correctamente su funcionamiento, por lo que la métrica de fidelidad es crucial. La medición de la fidelidad depende del alcance de los métodos de interpretación. Por ejemplo, los métodos de interpretación local reflejan con precisión el comportamiento del modelo de caja negra solo en la región limitada de entradas que están diseñados para explicar. En Papenmeier et al. (2019), los autores estudiaron el impacto de la fidelidad de las explicaciones en la confianza de los usuarios y encontraron que, aunque la precisión del modelo tiene un mayor impacto, las explicaciones con baja fidelidad pueden reducir la confianza de los usuarios.

2.4. Técnicas de interpretación de modelos de RL

2.4.1. Basadas en árboles de decisión

Dado que los árboles de decisión pueden visualizarse fácilmente de forma textual o gráfica (siendo inherentemente interpretables) Roth et al. (2019), se utilizan en diversos formatos para reemplazar el modelo de caja negra original en la interpretación. Sin embargo, entrenar un árbol de decisión es un proceso complejo Bastani et al. (2018). Por esta razón, en varios métodos de aprendizaje por refuerzo (RL) basados en árboles de decisión, se emplean estrategias como el imitation learning.

El aprendizaje por imitación (imitation learning o distillation) guía un modelo simple, como los árboles de decisión, a partir de un modelo más complejo, como las redes neuronales profundas (DNNs) Hinton et al. (2015). En lugar de entrenar el modelo externo directamente sobre el conjunto de datos, primero se entrena la DNN en los datos. Luego, la DNN entrenada transfiere su conocimiento al modelo más simple generando un conjunto de datos con etiquetas suavizadas (soft-labeled dataset). En este enfoque, el modelo complejo se conoce como el profesor (teacher), mientras que el modelo más simple es el estudiante (student).

Por ejemplo, en Bastani et al. (2018), se propuso un algoritmo de aprendizaje que aplica imitation learning a partir de un modelo DNN. Los autores utilizaron la función Q

de la DNN (teacher) para generar valores de entrenamiento para el modelo de árbol de decisión (student), además de las acciones óptimas para cada estado. Este método basado en imitation learning se denominó VIPER. Aunque el enfoque principal de VIPER es la verificación, también puede considerarse un método de interpretación en RL, ya que los árboles de decisión son inherentemente interpretables. No obstante, el alto número de nodos en la salida final (limitado a un máximo de 1000 nodos) dificulta la comprensión del resultado para los humanos.

Por otro lado, en Roth et al. (2019), los autores también interpretaron modelos de RL mediante árboles de decisión. Propusieron un modelo llamado Conservative Q-Improvement (CQI) para generar árboles más pequeños (menos de 100 nodos en un problema de navegación sencillo). En este enfoque, la adición de nuevos nodos al árbol final está restringida por la cantidad de recompensa futura descontada obtenida con estas adiciones. Como resultado, es necesario establecer un umbral que equilibre la longitud del árbol (y, por lo tanto, su interpretabilidad) con la precisión de la política de RL en la tarea a realizar.

Una limitación clave del método CQI es su posible pérdida de cobertura del espacio de estados. Al restringir la adición de nodos del árbol de decisión en función de la ganancia marginal en recompensas futuras descontadas, el algoritmo puede pasar por alto estados importantes que contribuyen a comportamientos poco frecuentes pero críticos, lo que reduce la capacidad de generalización del modelo. Además, aunque el CQI funciona bien en entornos simples (p. ej., tareas básicas de navegación), su escalabilidad a dominios complejos o de alta dimensión sigue siendo incierta, ya que equilibrar la interpretabilidad y la precisión de las políticas se vuelve cada vez más difícil en espacios de acción-estado más amplios.

El árbol de decisión resultante en CQI consta de dos tipos de nodos: nodos de rama (branch nodes) y nodos hoja (leaf nodes). Cada nodo de rama tiene dos hijos, y la división se basa en una de las características del vector de estado. Mientras tanto, un nodo hoja representa la acción que el agente toma en la trayectoria que finaliza en ese nodo. Debido a esta estructura, CQI solo puede aplicarse cuando el estado puede representarse mediante un vector de características.

En ambos ejemplos, VIPER y CQI, se utilizan árboles de decisión tradicionales para interpretar la salida del modelo. Sin embargo, los árboles de decisión pueden modificarse para interpretar modelos de aprendizaje por refuerzo (RL), como en Coppens et al. (2019). En este caso, los autores propusieron una técnica denominada distilling soft decision trees (SDT) para interpretar modelos de RL basados en redes neuronales profundas (DNNs).

El método SDTs combina árboles de decisión binarios con redes neuronales. En este enfoque, los nodos de bifurcación representan percepciones individuales, y la salida es la probabilidad p de moverse al subárbol derecho (y, en consecuencia, $1 - p$ es la probabilidad de moverse al subárbol izquierdo).

Por otro lado, los nodos de salida representan las acciones. Para que un humano comprenda las decisiones de este modelo, debe recorrer el árbol desde el nodo superior hasta los nodos hoja, lo cual se vuelve más complejo a medida que la profundidad del árbol aumenta. Además, los autores encontraron que la fidelidad del modelo disminuye con el aumento de la profundidad, lo que implica que se debe sacrificar fidelidad si se desean generar explicaciones más comprensibles.

Además de las estrategias mencionadas, en algunos casos se combinan varios árboles de decisión pequeños en diferentes formatos. Por ejemplo, la técnica Mixer of Expert Trees (MoET) Vasic et al. (2019) utiliza múltiples árboles de decisión más pequeños y especializados para interpretar una política de RL basada en una DNN. MoET consta de varios modelos expertos, cada uno especializado en un subespacio de las entradas. La especialización se logra mediante una función de compuerta que determina la contribución de cada experto en una muestra de entrada específica Yuksel et al. (2012). Luego, se calcula una suma ponderada de los expertos para obtener el resultado final.

De manera similar, los árboles de decisión de gradient boosting ofrecen otra forma de combinar múltiples modelos para obtener una decisión final. En este enfoque, el primer modelo se entrena con todos los datos, y el segundo modelo se entrena sobre la porción de datos donde el primer modelo cometió errores, y así sucesivamente. En otras palabras, cada modelo corrige los errores de su predecesor. Los modelos individuales resultantes se combinan en un formato de suma ponderada para generar la salida deseada para una

entrada determinada.

En Brown y Petrik (2018), los autores utilizaron este enfoque para interpretar modelos de RL mediante árboles de decisión. Además, propusieron reentrenar los árboles más recientes con el conocimiento de los árboles más antiguos para abordar el problema de manejar un gran número de árboles en el modelo final.

2.4.2. Explicaciones basadas en Visión por Computador

En esta categoría, se incluyen los métodos en los que la explicación se presenta en forma de una imagen, una lista de ideas o una imagen acompañada de información adicional.

SALIENCY MAPS

Un mapa de saliencia (saliency map) es una imagen que resalta la contribución de píxeles individuales en la decisión del modelo. Por ejemplo, los píxeles con valores más altos en la imagen de salida pueden indicar una mayor importancia en sus ubicaciones correspondientes en la imagen de entrada. Como resultado, cambiar los valores de estos píxeles en la imagen de entrada debería afectar significativamente la salida del modelo Nikulin et al. (2019).

La imagen generada puede ser local, explicando la decisión del modelo para una imagen específica, o global, identificando los píxeles más relevantes para una clase particular en todo el modelo. Esta última se usa para justificar por qué se seleccionó una clase en lugar de otras. Las interpretaciones basadas en saliency maps son comúnmente utilizadas para explicar modelos de aprendizaje automático, como redes neuronales profundas Shrikumar et al. (2017) y redes convolucionales Simonyan et al. (2013). Por ejemplo, en Simonyan et al. (2013), se introdujo un método que utiliza el gradiente de la salida con respecto a la imagen de entrada para generar tanto saliency maps locales como globales.

REGION-SENSITIVE RAINBOW

El método de interpretación Region Sensitive Rainbow (RS-rainbow) fue propuesto en Yang et al. (2018) para identificar las regiones más importantes en la imagen de entrada (rainbows) mientras se toman decisiones en un entorno de aprendizaje profundo con Q-learning. La diferencia principal entre los saliency maps y RS-rainbow es que, en RS-rainbow, la salida es una parte de la imagen original, mientras que en los mapas de saliencia los píxeles representan la importancia en términos de intensidad.

Las rainbows se generan mediante una arquitectura similar al método de atención utilizado en la creación de saliency maps integrados. Los autores incorporaron diversas técnicas de RL, como Deep Double Q-Networks Van Hasselt et al. (2016) y Dueling DQN Wang et al. (2016b), en una arquitectura que consta de tres componentes principales: codificador de imágenes, para transformar los fotogramas de entrada en mapas de características; módulo sensible a regiones, para generar probabilidades de imagen local para identificar las regiones más importantes; y una capa de política, las cuales producen los valores Q para la toma de decisiones.

Además de proporcionar una interpretación integrada, este método supera el rendimiento del método DQN, ya que tiene una representación más eficiente de los estados al incorporar las regiones más críticas en su representación.

COUNTERFACTUAL STATES

Un estado contrafactual (conterfactual state) es un estado que presenta una ligera diferencia con respecto a la condición original, pero que conduce a una acción distinta. En Dhurandhar et al. (2018), este enfoque se utiliza para explicar modelos de aprendizaje automático resaltando los píxeles críticos en la salida del modelo, es decir, aquellos cuya presencia es esencial y aquellos cuya ausencia es crucial para la clasificación del resultado.

En términos generales, el objetivo de este método es similar al de los saliency maps: identificar qué aspectos de la entrada visual son esenciales en las decisiones del agente. Sin embargo, en lugar de generar una imagen que indique la importancia de cada píxel

en la elección de acciones, este método produce una nueva imagen con el mínimo de modificaciones necesarias para cambiar la decisión del modelo.

EXPLICACIONES EN LENGUAJE NATURAL

Uno de los métodos de interpretación en RL consiste en explicar la política o algunas de sus partes utilizando lenguaje natural humano. En lugar de emplear un modelo basado en lenguaje comprensible de forma nativa, este enfoque agrega explicaciones en lenguaje natural a los modelos de caja negra ya existentes Wang et al. (2016a). Las explicaciones pueden utilizarse para justificar por qué se eligió una acción específica en una decisión individual, explicar la política completa del modelo, responder consultas particulares del usuario o justificar la selección de una decisión en lugar de otra (explicaciones contrastivas).

Por lo general, suelen emplearse explicaciones basadas en plantillas. Estas plantillas se utilizan para interpretar decisiones individuales en políticas de RL basadas en procesos de decisión de Markov (MDP). El objetivo es explicar por qué un agente de RL recomienda una acción específica completando una plantilla predefinida en lenguaje natural.

Los autores de Khan et al. (2009) propusieron una plantilla independiente del dominio. Para explicar una acción específica, la plantilla se rellena con los estados y escenarios que hacen más probable la elección de dicha acción. Luego, se completan los detalles específicos ejecutando el algoritmo desde el punto de partida (la acción que se quiere explicar) hasta el objetivo final. Además, los autores propusieron un marco para la generación de múltiples plantillas, con dos criterios fundamentales: criterio de suficiencia, en el cual dicen que una explicación es suficiente si puede justificar la optimalidad de una recomendación sin necesidad de usar más plantillas, y criterio de minimalidad, donde una explicación es mínima si incluye el menor número posible de plantillas para transmitir la información necesaria.

3 | Estrategia de Interpretación

En el contexto de las redes ópticas elásticas, uno de los principales objetivos es la asignación de recursos para la transmisión de datos, un proceso conocido como RSA (Routing and Spectrum Allocation) Galimberti y Sambo (2021). Este problema es especialmente desafiante debido a la complejidad inherente de las redes ópticas y las múltiples variables que deben considerarse para maximizar la eficiencia del sistema.

Las redes ópticas elásticas permiten una asignación flexible de ancho de banda y recursos, lo que les permite adaptarse de forma dinámica a los cambios en la demanda de tráfico. Sin embargo, resolver el problema RSA requiere considerar factores como la topología de la red, la capacidad de transmisión de cada enlace, la disponibilidad de diferentes niveles de modulación y la distribución del espectro óptico. Además, deben tomarse en cuenta las restricciones de calidad de servicio (QoS) para garantizar un rendimiento eficiente del sistema.

Las dificultades en la resolución del problema RSA radican en la necesidad de determinar, para cada solicitud de conexión, una ruta de transmisión, un formato de modulación adecuado y una porción de espectro disponible que mejore la utilización de recursos mientras se cumplen los requisitos de calidad de servicio. Esto implica el desarrollo de algoritmos eficientes y escalables, capaces de manejar grandes volúmenes de datos y tomar decisiones rápidas en tiempo real. Además, la complejidad computacional del problema aumenta con el tamaño y la densidad de la red, lo que hace necesario el uso de enfoques innovadores y técnicas avanzadas de optimización para encontrar soluciones viables en un tiempo razonable.

Dada la importancia y complejidad del problema, este trabajo se centra principal-

mente en la interpretación de agentes especializados en resolver el problema del RSA. No obstante, la estrategia aquí propuesta no se limita a obtener interpretaciones de este tipo de agentes, sino que se puede extender a diversos problemas de redes ópticas. En las siguientes secciones se detallan la estrategia de interpretación propuesta, así como los principales valores usados en las simulaciones.

3.1. Estrategia de interpretación propuesta

El enfoque de interpretación utilizado en este trabajo se muestra en la Fig. 3.1. La estrategia consta de tres etapas: entrenamiento del agente DRL, una técnica de aprendizaje por imitación y un análisis de interpretación.

Primero, entrenamos un agente DRL para resolver un problema específico en redes ópticas. El proceso de entrenamiento se desarrolló a través de distintas etapas experimentales orientadas a determinar configuraciones sólidas que garantizaran un buen desempeño del agente y, al mismo tiempo, facilitaran su interpretabilidad (ver Anexo. A). En este caso, nos centramos en el problema de enrutamiento y asignación de espectro (RSA). En la primera etapa, las acciones del agente no son interpretables, ya que actúa como un modelo de caja negra. A continuación, en el segundo paso, entrenamos diferentes clasificadores utilizando pares estado-acción recopilados del agente. Este proceso se conoce como aprendizaje por imitación, ya que los clasificadores intentarán imitar el comportamiento del agente. Evaluamos cuatro técnicas de clasificación en nuestros experimentos: regresión lineal, árboles de decisión, regresión logística y bosque aleatorio. Estos clasificadores tienen la ventaja de ser interpretables, y un conjunto de interpretaciones se utilizará en la etapa final para comprender el comportamiento del agente. De esta manera, obtenemos una interpretación basada en imitaciones del agente. A continuación, detallamos cada uno de estos pasos.

3.1.1. Aprendizaje Reforzado Profundo (DRL)

Los algoritmos de Deep Reinforcement Learning (DRL) se representan comúnmente como un proceso de decisión de Markov finito (MDP) Levine et al. (2020); Haarnoja

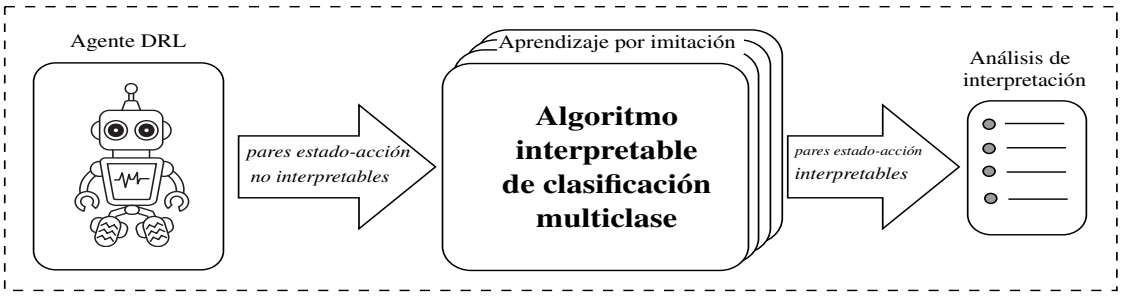


Figura 3.1: Estrategia de interpretación de tres etapas.

et al. (2018). Esta formulación del problema se define mediante la tupla $\mathcal{S}, \mathcal{A}, \mathcal{R}$, donde $\mathcal{S}$ representa el espacio de estados, $\mathcal{A}$ el espacio de acciones, y $\mathcal{R}(S, A)$ la función de recompensa escalar. El problema consiste en un agente que interactúa con su entorno en una serie de pasos de tiempo $1, 2, ..., t-1, t, t+1, ...$ El objetivo del agente es aprender una política óptima Π^*, que mapea el estado observado actual $S_t \in \mathcal{S}$ del entorno a la acción óptima $A^*(\Pi^* : S \rightarrow A^*)$. Para cada acción $A_t \in \mathcal{A}$, se proporciona una recompensa numérica $R_{t+1} \in \mathcal{R} \subset \mathbb{R}$, que brinda retroalimentación al agente sobre la efectividad de cada acción. El agente busca maximizar la recompensa futura acumulada G_t en el paso de tiempo t, definida como:

$$G_t = \sum_{\tau=t+1}^{T} \kappa^{\tau-t-1} R_\tau \tag{3.1}$$

donde T denota el número total de pasos de tiempo y $\kappa \in [0, 1)$ representa el factor de descuento Zhang et al. (2019); Sutton y Barto (2018).

En este trabajo, el agente DRL consiste en un agente multi-discreto que toma acciones en un entorno de redes ópticas elásticas (EON) para resolver el problema de enrutamiento y asignación de espectro (RSA) en redes ópticas elásticas. El agente aprende a seleccionar un conjunto de acciones dentro de la red óptica en función del estado de la red en cada instante de tiempo. Los componentes clave de nuestro modelo de Reinforcement Learning (RL) son los siguientes:

Episodio Un episodio consiste en una serie de pasos de tiempo. En cada episodio de entrenamiento, se comienza con una red vacía y se reciben solicitudes de conexión de manera secuencial a una tasa de una solicitud por cada paso de tiempo, las cuales el agente debe atender.

Representación del Estado En general, la representación del estado es un subconjunto del estado completo del entorno de la red óptica. Este estado incluye datos de demanda, datos de utilización de la red, características del grafo de la red y el estado físico de la capa óptica (por ejemplo, la relación señal-ruido óptica OSNR de los enlaces). La selección de características del estado depende del problema de optimización específico Liu et al. (2021).

Acción Las acciones dentro del marco de Reinforcement Learning representan las decisiones disponibles para el agente en cada paso de tiempo. El conjunto específico de acciones y sus definiciones dependen del problema en cuestión. En muchos enfoques, se considera un vector fijo de acciones, lo que implica que cualquier acción $A \in \mathcal{A}$ puede seleccionarse en cualquier instante de tiempo.

Recompensa Siguiendo la definición del MDP, la recompensa total $\mathcal{R}_{tot(t)}$ en un paso de tiempo t se define como la suma de todas las recompensas anteriores:

$$\mathcal{R}_{tot(t)}(\mathcal{S}_t, \mathcal{A}_{t-1}) = \sum_{t=1}^{T} R_t(\mathcal{S}_{t-1}, \mathcal{A}_{t-1}), \tag{3.2}$$

donde $\mathcal{R}_t(\mathcal{S}_{t\text{-}1}, \mathcal{A}_{t\text{-}1})$ representa la recompensa individual en el paso de tiempo t, dado el estado $\mathcal{S}_{t-1}$ y la acción $\mathcal{A}_{t-1}$.

Dada una acción del agente, la conexión se asigna en el índice especificado si el bloque correspondiente está disponible en la ruta. En este caso, el agente recibe una recompensa positiva (+1). Si la asignación falla, se penaliza al agente con una recompensa negativa (-1).

3.1.2. Aprendizaje por Imitación

El aprendizaje por imitación es una técnica de aprendizaje automático en la que un modelo o agente aprende a realizar una tarea observando ejemplos de cómo un experto ejecuta dicha tarea Hussenot et al. (2020). Esencialmente, el modelo de aprendizaje (también llamado estudiante) intenta imitar las acciones y decisiones tomadas por el experto

para lograr un rendimiento similar en la tarea objetivo. El aprendizaje por imitación generalmente involucra varios pasos: recolección de datos, diseño del modelo, entrenamiento, evaluación y ajuste fino Peng et al. (2021). Se pueden prever varias ventajas del aprendizaje por imitación. Por ejemplo, puede ser útil cuando el diseño manual de algoritmos es difícil o costoso, permitiendo que los modelos aprendan directamente de la experiencia. Sin embargo, también presenta limitaciones, como la incapacidad de adaptarse a situaciones nuevas o imprevistas que el experto no haya encontrado antes.

En los últimos años, se han presentado avances en las técnicas de aprendizaje profundo mediante la aplicación del aprendizaje por imitación a técnicas de DRL, como el algoritmo DAgger Ross et al. (2011). Este algoritmo combina elementos de aprendizaje por imitación y aprendizaje por refuerzo para abordar algunos de estos desafíos y mejorar la adaptabilidad del modelo.

El algoritmo DAgger intenta recopilar demostraciones del agente bajo la distribución de estados inducida por la política aprendida, es decir, el agente proporciona las acciones correctas a tomar. Sin embargo, la distribución de entrada de los ejemplos proviene del comportamiento del aprendiz. La forma más simple de DAgger procede como se muestra en el algoritmo 1.

Algorithm 1 : DAgger algorithm

1: **procedure** DAgger(a, b) ▷ initial demonstrations ($\mathcal{D}$), agent policy (π^A), mixer (β)
2: *train* π_1^L on $\mathcal{D}$
3: **for** $i = 1$ to $\mathcal{N}$ **do**
4: *let* $\pi_i = \beta_i \pi^E + (1 - \beta_i)\pi_i^L$
5: *sample trajectories:* $\tau[(s_0, a_0), ..., s_T, a_T]$ using π_i
6: $\mathcal{D}_i = [\pi^E(\tau_s^0), ..., \pi^E(\tau_s^T)]$
7: *aggregate dataset:* $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{D}_i$
8: *train learner policy:* π_{i+1}^L *on* $\mathcal{D}$
9: **return** *best* π_{i+1}^L

La política del aprendiz se inicializa utilizando un conjunto de datos $\mathcal{D}$ de demostraciones de expertos, lo que da como resultado una política π_1^L en la línea 2. Posteriormente, itera $\mathcal{N}$ veces desde las líneas 3 a 9. En la primera iteración, la política $\pi_1 = \beta_1 \pi^A + (1-\beta_1)\pi_1^L$ (una mezcla estocástica del agente y el aprendiz) se usa para recopilar nuevas trayectorias (línea 5), que se agregan a $\mathcal{D}$ (en la línea 7), la cual a su vez, se usa para entrenar una política de aprendiz π_2^L en la línea 8. En la iteración $n - 1$, una política π_{n-1} se usa para

recopilar más trayectorias, y esas trayectorias se agregan a $\mathcal{D}$, de modo que la política π_n^L pueda imitar al agente basándose en todo el conjunto de datos.

En el núcleo del proceso de aprendizaje por imitación se encuentra la política imitadora. Cualquier algoritmo de clasificación multiclase puede utilizarse para nuestro propósito. Cuanto más simple sea el algoritmo, más interpretaciones se podrán obtener. Sin embargo, los modelos más simples suelen presentar bajas capacidades de imitación, lo que puede hacer que los conocimientos extraídos del agente difieran de la realidad. Este equilibrio entre la interpretabilidad del modelo y su rendimiento se muestra en 3.2.

Como se observa en la Fig. 3.2, los clasificadores RF presentan un buen compromiso entre rendimiento e interpretabilidad Rudin (2019). A diferencia de modelos complejos como las redes neuronales, los RF son transparentes en su proceso de toma de decisiones, lo que los hace valiosos en escenarios donde comprender el razonamiento detrás de las predicciones es esencial, al mismo tiempo que mantienen un buen desempeño. En comparación, los clasificadores de regresión lineal son simples y ampliamente interpretables, pero carecen del poder predictivo y la versatilidad de los clasificadores RF, especialmente cuando se trata de encontrar relaciones complejas entre los datos.

La selección del método adecuado depende de la complejidad y las necesidades específicas del problema en cuestión. Por lo tanto, es fundamental evaluar diversos clasificadores para lograr una interpretación precisa de los algoritmos de DRL cuando se aplican a redes ópticas. Esta evaluación garantiza que el modelo elegido se ajuste bien a los requisitos

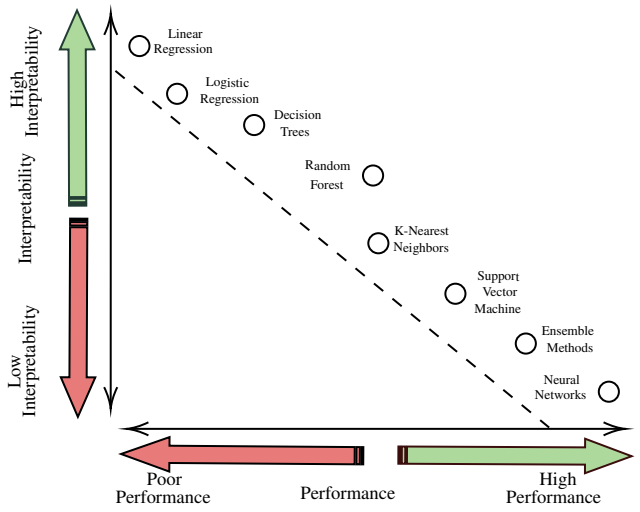


Figura 3.2: Relación de compromiso entre interpretabilidad y rendimiento de algunos clasificadores

del problema, equilibrando de manera eficaz la interpretabilidad y el rendimiento. Para ello, en este trabajo experimentamos con regresión lineal, árboles de decisión, regresión logística y clasificadores de bosque aleatorio para imitar el comportamiento del agente y evaluar su comportamiento.

3.1.3. Análisis de interpretación

En este apartado, detallamos cómo pueden ser usados los clasificadores de regresión lineal (LR), árboles de decisión (DT), regresión logística (LGR) y bosque aleatorio (RF) en el ambiente multiclase, así como el proceso de interpretación de sus resultados.

Regresión Lineal (LR)

Para la tarea de clasificación utilizando LR, en este trabajo consideramos el clasificador Ridge Hastie et al. (2009), ya que ofrece un equilibrio entre simplicidad, interpretabilidad y rendimiento en tareas de clasificación. El clasificador Ridge incorpora regularización L2 para mitigar el sobreajuste y mejorar la capacidad de generalización en problemas de clasificación. Al igual que otros clasificadores lineales, aprende un límite de decisión lineal que separa las diferentes clases en el espacio de características. Sin embargo, además de minimizar el error entre las etiquetas de clase predichas y las reales, también penaliza los coeficientes grandes de las variables de características, reduciéndolos efectivamente hacia cero. Este término de regularización, controlado por un hiperparámetro llamado fuerza de regularización o alpha, ayuda a evitar que el modelo se vuelva demasiado sensible al ruido en los datos de entrenamiento.

Interpretar los resultados de un clasificador Ridge implica examinar los coeficientes aprendidos de las características. Los coeficientes positivos indican características que contribuyen positivamente a la predicción de una clase particular, mientras que los coeficientes negativos sugieren que la característica contribuye negativamente. Sin embargo, algunas bibliotecas de regresión lineal pueden forzar los coeficientes a ser positivos, lo que facilita el proceso de interpretación. La magnitud de los coeficientes refleja la importancia de cada característica en la determinación de las etiquetas de clase. Además, ajustar la fuerza de regularización permite equilibrar el ajuste de los datos de entrenamiento y mantener la

simplicidad y capacidad de generalización del modelo.

Regresón Logística (LGR)

La regresión logística multinomial (LGR) es una extensión de la regresión logística diseñada para manejar tareas de clasificación con más de dos clases. Modela las probabilidades de cada clase utilizando la función softmax, la cual generaliza la función logística a múltiples clases. Similar a la regresión logística binaria, el modelo calcula una combinación lineal de las características de entrada y sus pesos asociados para cada clase, y luego aplica la función softmax para obtener las probabilidades de cada clase. Durante el entrenamiento, el modelo aprende los pesos que minimizan la diferencia entre las probabilidades predichas y las etiquetas de clase reales mediante técnicas como la estimación de máxima verosimilitud o descenso de gradiente.

La interpretación de los resultados de la regresión logística multinomial implica examinar los coeficientes aprendidos (pesos) asociados con cada característica para cada clase. Coeficientes positivos indican que un aumento en el valor de la característica incrementa la probabilidad de pertenecer a la clase respectiva, mientras que coeficientes negativos indican lo contrario. La magnitud de los coeficientes refleja la fuerza de la relación entre la característica y la clase resultante. Además, comparar las probabilidades entre clases permite comprender la probabilidad relativa de cada clase dadas las características de entrada.

Árboles de Decisión (DT)

Los árboles de decisión (DT) operan segmentando los datos en subconjuntos en función de los valores de las características. Comenzando en el nodo raíz, seleccionan iterativamente la característica más informativa para dividir los datos, creando ramas que conducen a nodos hijos. Este proceso continúa recursivamente, con cada división buscando maximizar la homogeneidad de los subconjuntos en relación con la variable objetivo. Finalmente, el árbol llega a los nodos hoja, donde asigna etiquetas de clase para tareas de clasificación. Durante la predicción, una instancia recorre el árbol desde el nodo raíz hasta un nodo hoja según sus valores de características, y la etiqueta asociada a ese nodo hoja se

convierte en la clase predicha.

Los árboles de decisión ofrecen transparencia e interpretabilidad, pero pueden sobreajustarse si no se restringen adecuadamente. Técnicas como poda (prunning) y limitación de la profundidad del árbol ayudan a mitigar este problema, haciendo de los árboles de decisión un algoritmo versátil y ampliamente utilizado en clasificación.

Interpretar un árbol de decisión implica rastrear la lógica de toma de decisiones codificada dentro de la estructura del árbol. Comenzando desde el nodo raíz, cada punto de ramificación representa una decisión basada en una característica específica y su valor. Siguiendo las ramas a través de los nodos internos, se revelan las condiciones bajo las cuales los datos se particionan en subconjuntos. Finalmente, llegar a los nodos hoja proporciona la decisión de clasificación final. Analizar la distribución de clases en los nodos hoja ayuda a comprender cómo el árbol de decisión asigna etiquetas a las instancias. Además, la profundidad del árbol y la importancia de las características brindan información sobre la complejidad del proceso de toma de decisiones y la relevancia de las diferentes características en los resultados. Al interpretar un árbol de decisión, los investigadores pueden descubrir patrones subyacentes en los datos y obtener transparencia en el proceso de toma de decisiones, facilitando tanto la predicción como la comprensión en tareas de aprendizaje automático.

Random Forest (RF)

El bosque aleatorio (RF) consiste en un conjunto de árboles de decisión. Cada árbol se entrena independientemente sobre una muestra bootstrap del conjunto de datos original, lo que significa que cada árbol ve un subconjunto ligeramente diferente de los datos Wei et al. (2021). Al crecer un árbol de decisión individual, divide repetidamente los datos en subconjuntos según los valores de las características. Estas divisiones se determinan encontrando la característica y el umbral que mejor separan los datos, utilizando métricas como la impureza de Gini o la entropía Criminisi et al. (2011). A medida que crece cada árbol de decisión, genera un conjunto de reglas de decisión. Estas reglas representan el camino que sigue un punto de datos a través del árbol, desde el nodo raíz hasta un nodo hoja, basado en sus valores de características. Por ejemplo, una regla de decisión podría

verse así: "Si la Característica A > 5 y la Característica B < 10, entonces Clase X."

Durante la predicción, cada árbol de decisión genera su propia predicción basada en las reglas de decisión que ha aprendido. Para tareas de clasificación, esta predicción puede ser una etiqueta de clase. RF combina las predicciones de todos los árboles individuales mediante un proceso llamado "votación". La predicción final es la clase que recibe más votos entre todos los árboles.

En este trabajo, realizamos un análisis de importancia de características, determinando qué características son más influyentes en las predicciones al examinar con qué frecuencia aparecen en las reglas de decisión en todo el conjunto de árboles. De esta manera, proporcionamos transparencia e interpretabilidad al modelo DRL.

3.2. El simulador

La simulación de este trabajo de tesis se realizó usando las herramientas proporcionadas en el framework *Multiband*[1]. Este framework gestiona tanto el entrenamiento de agentes DRL aplicados a redes ópticas, como el aprendizaje por imitación de diferentes clasificadores multiclases. Para el entrenamiento del agente especializado en el problema RSA, consideramos tres rutas alternativas (caminos más cortos) y los cinco primeros bloques disponibles en la topología de red NSFNet(3.3). Esto significa que, en cada paso de tiempo, el agente debe decidir entre 1 de 15 acciones posibles (cinco bloques posibles en cada una de las tres rutas) para resolver correctamente el problema RSA.

Esta investigación sigue el modelo uniforme de comunicación entre todos los nodos propuesto en Ives et al. (2015) para la generación de tráfico. Esto implica que cada nodo puede solicitar comunicación con la misma probabilidad en cada paso de tiempo. Las solicitudes de conexión siguen una distribución de variable exponencial, con tiempo medio entre llegadas de λ y tiempo medio de servicio de la conexión μ, para una carga de tráfico de la red de λ/μ.

[1] https://github.com/jbcedeno930806/multiband.git

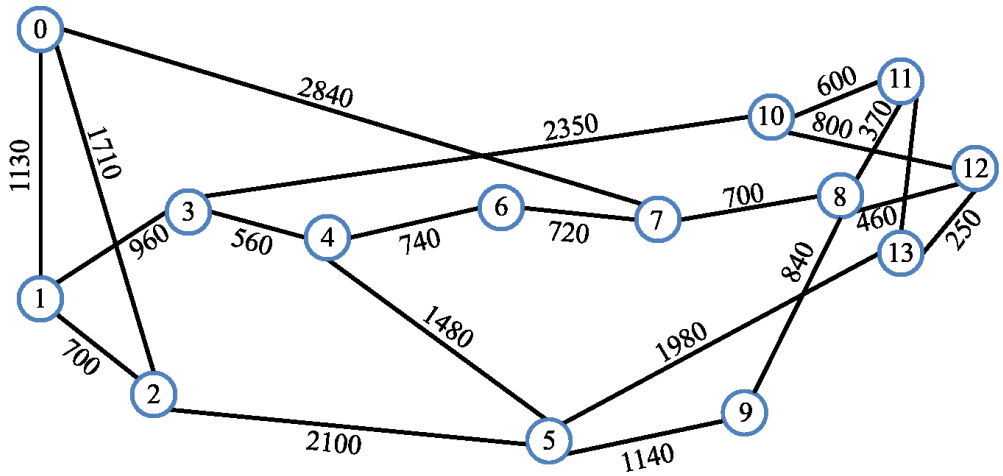


Figura 3.3: Topología de la red NSFNet usada en las simulaciones.

3.2.1. Impedimentos de la capa física

Las tasas de bits se asignan a cada solicitud de conexión utilizando una variable aleatoria uniforme que selecciona una de varias posibilidades. Los requisitos de tasa de bits disponibles se enumeran en la tabla 3.1. La tabla 3.1 considera el impacto de las degradaciones de la capa física (PLI) acumuladas por la señal durante la propagación para calcular el número necesario de unidades de ranura de frecuencia (FSUs), como en Calderon et al. (2020). Para garantizar una calidad de transmisión (QoT) adecuada, se considera una probabilidad de tasa de error de bits de 10^{-6} en el análisis de cálculo. Según la tabla 3.1, la demanda de solicitud de conexión (en FSUs) es una función de la tasa de bits solicitada y del formato de modulación seleccionado para transmitir la conexión.

Tabla 3.1: Requisitos de espectro en términos de FSUs y alcance máximo alcanzable (MAR) para cada par de tasa de bits y formato de modulación.

Modulation Format		Bit rate (Gbps)				
Type	MAR (km)	10	40	100	400	1000
BPSK	5520	1	4	8	32	80
QPSK	2720	1	2	4	16	40
8-QAM	1360	1	2	3	11	27
16-QAM	560	1	1	2	8	20
32-QAM	240	1	1	2	7	16
64-QAM	80	1	1	2	6	14

3.2.2. Configuración del agente DRL

El agente DRL consiste en el algoritmo Proximal Policy Optimization (PPO) Schulman et al. (2017). El agente fue entrenado en el entorno rmsa_env (ver repositorio) para resolver el problema RMSA utilizando el algoritmo PPO proporcionado por la biblioteca Stable Baselines 3 Raffin et al. (2021). Durante el entrenamiento, solo se consideraron cuatro hiperparámetros: el número de pasos (n_steps) = 1024, el factor de descuento (κ) = 0.99, una tasa de aprendizaje de 3×10^{-6} y un tamaño de lote de 16. Se llevó a cabo un entrenamiento con una duración de diez millones de pasos de tiempo, equivalente a 5.000 episodios de 2.000 pasos de tiempo. Todos los parámetros no descritos se mantuvieron consistentes con la configuración predeterminada especificada en Stable Baselines 3. Para fines de validación, se emplearon 30 episodios, cada uno con 10.000 pasos de tiempo.

En cada paso de tiempo, la representación del estado se proporciona al agente como un vector de información $\mathcal{X}$ de 67 características, como se muestra en la Fig. 3.4. Las características de $\mathcal{X}[1]$ a $\mathcal{X}[14]$ y de $\mathcal{X}[15]$ a $\mathcal{X}[28]$ indican el nodo de transmisión y el nodo de destino utilizando un formato de codificación one-hot. Luego, las características restantes se dividen en tres grupos de 13 elementos, un grupo para cada posible camino considerado en el problema RSA. Para cada grupo, las características de 1 a 10 indican el índice inicial y el número de ranuras contiguas de los primeros cinco bloques disponibles, mientras que las características 11, 12 y 13 muestran la demanda de la conexión en términos de FSUs, el número total de ranuras disponibles en la ruta y el número promedio de ranuras libres en todos los bloques disponibles.

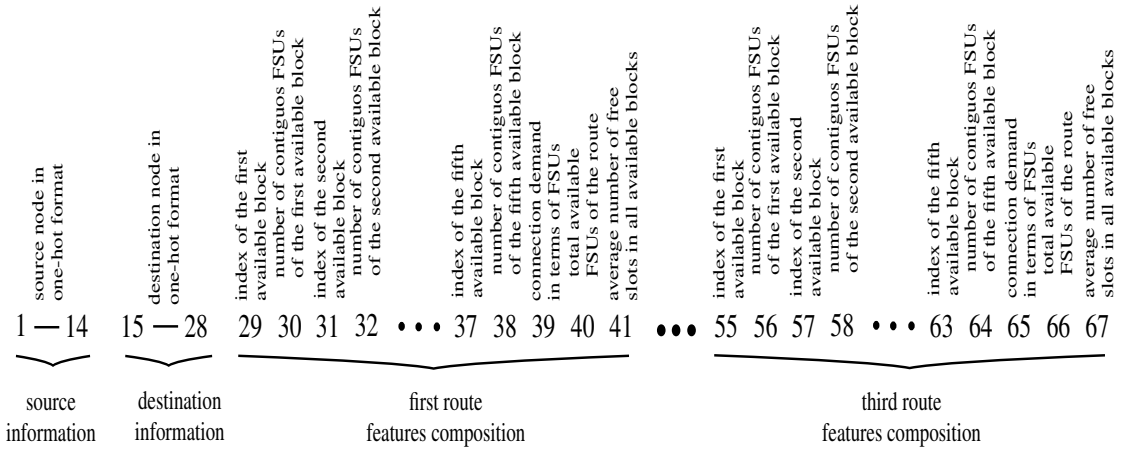


Figura 3.4: Descripción de la representación del estado utilizado para entrenar el agente DRL

3.2.3. Configuración del Aprendizaje por Imitación

La etapa de aprendizaje por imitación comprende el algoritmo DAgger, que interactúa con un algoritmo de clasificación. Para ello, evaluamos cuatro técnicas de clasificación diferentes. Todas las implementaciones utilizadas aquí son proporcionadas por la biblioteca sklearn Pedregosa et al. (2011). Los clasificadores fueron entrenados utilizando el algoritmo DAgger (ver Algoritmo 1) con un conjunto de datos inicial de 200.000 pares de muestras estado-acción recopiladas del agente. El algoritmo DAgger se ejecutó durante un total de $N = 10$ rondas utilizando una función mezcladora estocástica decreciente lineal β, que varía de 1 a 0. A continuación, describimos los principales parámetros utilizados para cada clasificador.

Clasificador Ridge: El clasificador Ridge primero convierte los valores objetivo en -1, 1 y luego trata el problema como una tarea de regresión multi-salida. Para nuestros experimentos, personalizamos solo dos parámetros. Primero, utilizamos una fuerza de regularización de 1. Cabe destacar que la regularización mejora la condición del problema y reduce la varianza de las estimaciones. Además, usamos un *tol* de 0.0001. Este parámetro especifica un criterio de convergencia para el modelo de clasificación, lo que a su vez controla la precisión de la solución. Finalmente, forzamos todos los coeficientes de peso a ser positivos, estableciendo el parámetro "positive" en True. Todos los demás parámetros de entrada permanecen en su valor predeterminado.

Clasificador de Regresión Logística Multinomial: Este clasificador es una extensión del clasificador de regresión logística, configurado para ser un clasificador *multiclase*. En el caso multiclase, el algoritmo de entrenamiento utiliza la opción de pérdida de entropía cruzada. Además, configuramos parámetros adicionales para considerar una función de penalización L2, la cual es una regularización que penaliza los coeficientes grandes añadiendo al costo un término proporcional al cuadrado de los parámetros, un *tol* de 0.0001, y escogemos el algoritmo L-BFGS Liu y Nocedal (1989) como método de optimización. No se realizaron personalizaciones adicionales.

Clasificador de Árbol de Decisión: Este algoritmo divide recursivamente un conjunto de muestras según un criterio de división. Las muestras pueden dividirse aleatoriamente o seleccionando la mejor división posible, utilizando diferentes criterios como gini, entropía o *log_loss*. Para este caso, configuramos los parámetros *splitter* y *criterion* para usar la mejor división según el criterio gini. Cabe destacar que los clasificadores de árboles de decisión pueden generar árboles excesivamente complejos que no generalizan bien a partir de los datos de entrenamiento. Esta situación genera muchas reglas de decisión y puede complicar la interpretación y transparencia del modelo. En estos casos, se pueden emplear mecanismos como la poda o la limitación de la profundidad del árbol para evitar este problema. Sin embargo, la poda o la limitación de la profundidad del árbol pueden influir en las capacidades de imitación. La solución adecuada a la relación de compromiso entre la capacidad de imitación y la interpretabilidad del modelo puede variar según el problema y las necesidades del usuario. En nuestros experimentos, observamos que los árboles generados eran lo suficientemente cortos para facilitar la interpretación del modelo; por lo tanto, no se consideró ninguna personalización adicional de parámetros para limitar el clasificador de árbol de decisión.

Clasificador Random Forest: Nuestra implementación de Random Forest se configuró para utilizar 300 estimadores. Cada estimador está configurado como un árbol de decisión único, siguiendo la descripción previa dada en esta sección. Todos los demás parámetros relacionados con el clasificador RF se mantuvieron en su valor predeterminado.

Complejidad computacional

Al comparar la complejidad computacional de los clasificadores Ridge, árboles de decisión, regresión logística multinomial y random forest para imitar el comportamiento de un agente DRL, es crucial considerar tanto la fase de entrenamiento como la de predicción para cada modelo.

El clasificador Ridge, una variante de la regresión lineal con regularización L2, típicamente tiene una complejidad de entrenamiento de $O(n^2m + n^3)$ Hastie et al. (2009), donde n es el número de características y m es el número de ejemplos de entrenamiento. Esta complejidad surge de la resolución del problema de mínimos cuadrados regularizados,

que implica la inversión de matrices. Durante la predicción, el clasificador Ridge tiene una complejidad de $O(n)$, ya que solo requiere calcular el producto punto entre el vector de características y el vector de pesos.

En contraste, los árboles de decisión presentan una complejidad de entrenamiento de $O(mn \log m)$ Molnar (2020). Esta complejidad se debe al proceso recursivo de división de los datos en cada nodo, lo que implica ordenar y seleccionar las mejores divisiones de características. La predicción con árboles de decisión es más eficiente, con una complejidad de $O(\log m)$ Hastie et al. (2009), ya que implica recorrer el árbol desde la raíz hasta un nodo hoja.

El clasificador de Regresión Logística Multinomial, utilizado para clasificación multiclase, tiene una complejidad de entrenamiento de $O(knm)$, donde k representa el número de iteraciones hasta la convergencia. Este proceso iterativo involucra la optimización mediante el gradiente descendente. Al igual que los clasificadores Ridge, la complejidad de predicción para la Regresión Logística Multinomial es $O(n)$, ya que implica calcular las probabilidades para cada clase y seleccionar la clase con la mayor probabilidad.

Por último, el clasificador Random Forest, un conjunto de árboles de decisión, tiene una complejidad de entrenamiento mayor, de $O(kmn \log m)$ Hastie et al. (2009), donde k es el número de árboles en el bosque. Cada árbol se construye de manera independiente, lo que conlleva un aumento en las demandas computacionales. Sin embargo, los bosques aleatorios ofrecen robustez y un mejor desempeño al promediar las predicciones de múltiples árboles. La complejidad de predicción para los bosques aleatorios es $O(k \log m)$, lo que refleja la necesidad de agregar predicciones de k árboles, cada uno recorriendo desde la raíz hasta un nodo hoja.

En resumen, los clasificadores Ridge y de Regresión Logística Multinomial ofrecen una menor complejidad computacional durante la predicción, pero pueden carecer de la capacidad para modelar relaciones complejas con la misma eficacia que los árboles de decisión o los bosques aleatorios. Los árboles de decisión proporcionan un equilibrio entre complejidad e interpretabilidad, mientras que el Random Forest, es computacionalmente complejo durante el entrenamiento. Sin embargo, ofrece un mejor rendimiento y mayor interpretabilidad, lo que lo hace adecuado para escenarios que requieren toma de decisiones

robustas y transparentes. Evaluar estos clasificadores ayuda a seleccionar el modelo más adecuado según las necesidades y la complejidad específica del problema en cuestión.

4 | Resultados Experimentales

En este capítulo se presentan los resultados derivados de la estrategia de interpretación propuesta para el análisis del comportamiento de agentes aplicados al problema de Enrutamiento y Asignación de Espectro (RSA). Antes de abordar la interpretación del agente de Aprendizaje por Refuerzo Profundo (DRL), es necesario demostrar que la metodología desarrollada es válida y capaz de reproducir patrones de decisión conocidos. Para ello, incorporamos un experimento preliminar orientado a imitar el comportamiento de la heurística clásica K-Shortest Path First-Fit (K-SP-FF) con K=3. Este experimento permite verificar que la estrategia implementada —basada en el empleo de cuatro clasificadores complementarios— puede capturar adecuadamente la lógica interna de un algoritmo cuyo funcionamiento es completamente transparente.

Una vez validada la metodología en este escenario controlado, extendemos el análisis al caso central del presente trabajo: la interpretación del agente DRL entrenado para resolver el problema RSA. Utilizando nuevamente los mismos cuatro clasificadores, examinamos su comportamiento desde tres perspectivas: (i) una interpretación general del proceso de toma de decisiones en el problema RSA completo, (ii) un análisis específico del subproblema de enrutamiento, y (iii) un análisis independiente del subproblema de asignación de espectro. Esta estructura comparativa permite contrastar la consistencia, divergencia o alineación entre las decisiones del agente DRL y las heurísticas establecidas, proporcionando así una visión profunda y fundamentada sobre su proceso de decisión.

4.1. Evaluación Preliminar de la Metodología mediante 3-SP-FF

Antes de aplicar la estrategia de interpretación al agente DRL, es fundamental validar que la metodología propuesta puede capturar correctamente patrones de decisión conocidos. Para ello, se construyó un escenario controlado en el cual se imita el comportamiento de la heurística *K-Shortest Path First-Fit* (3-SP-FF), con $K = 3$. Esta heurística opera siguiendo dos reglas estrictas:

1. Seleccionar la primera ruta disponible entre las K rutas más cortas.
2. Asignar el primer bloque espectral disponible mediante el criterio *First-Fit*.

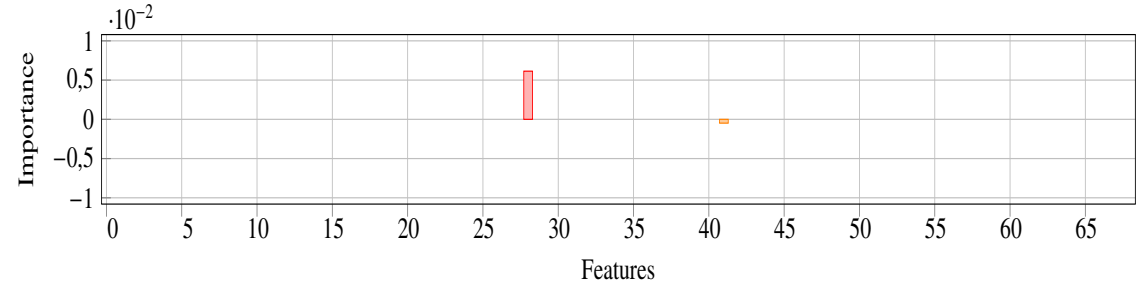
Cualquier clasificador capaz de imitar la heurística debería asignar importancia estas dos reglas intrínsicas del algoritmo KSPFF. A continuación, se presentan los resultados obtenidos para los cuatro clasificadores empleados en la metodología.

4.1.1. Resultados del Clasificador Linear Regression Multiclase

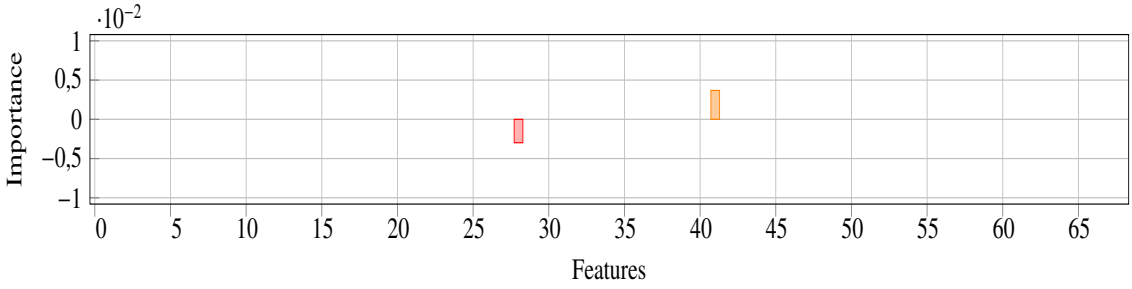
La Figura 4.1 muestra la importancia asignada por el modelo de Regresión Lineal Multiclase. Aunque los coeficientes son de magnitud reducida debido a la normalización del modelo, se observa que:

- Solo dos características muestran coeficientes significativamente diferentes de cero: la característica 28 y la característica 41.
- La relación entre los signos es coherente con el comportamiento de la heurística:
  - La disponibilidad de la primera ruta (característica 28) es la más influyente, como se espera, dado que KSP-FF selecciona siempre la primera ruta válida.
  - La segunda ruta (característica 41) recibe importancia únicamente en los casos donde la primera no es viable.

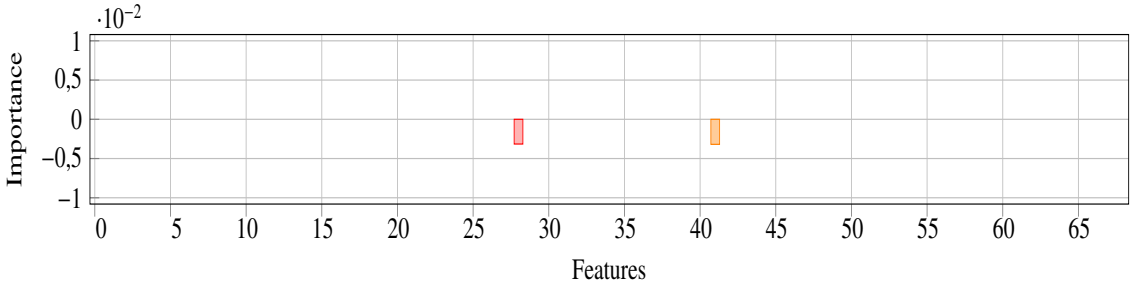
Los resultados confirman que el modelo captura fielmente la lógica de la heurística, limitando la importancia a las dos variables relevantes.



(a) Importancia de las características obtenida para la primera clase.



(b) Importancia de las características obtenida para la segunda clase.



(c) Importancia de las características obtenida para la segunda clase.

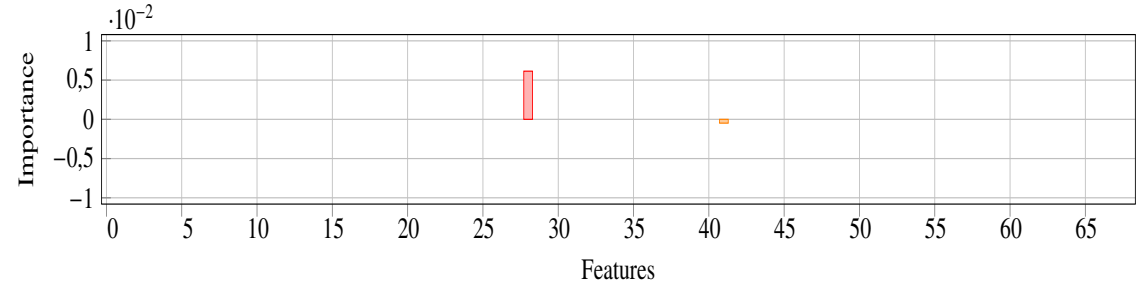
Figura 4.1: Importancia promedio de las características obtenida para el clasificador LR.

4.1.2. Resultados del Clasificador Logistic Regression (LGR)

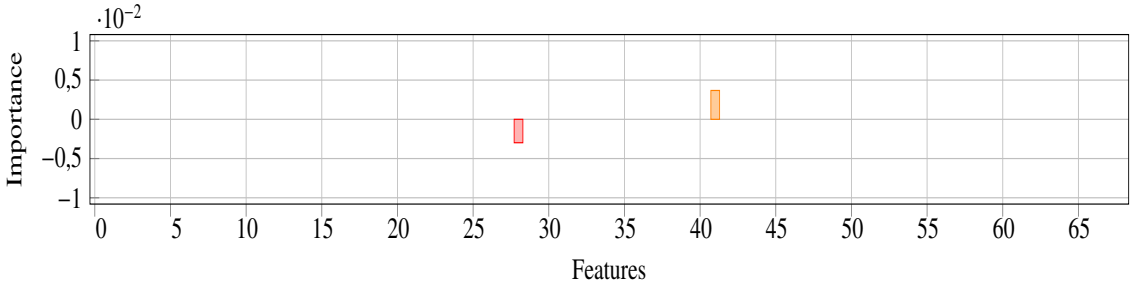
El comportamiento de la Regresión Logística Multiclase es aún más revelador, como se aprecia en la Figura 4.2. El modelo asigna valores altos (positivos o negativos) únicamente a las característics 28 y 41, mientras que el resto de las características poseen coeficientes exactamente nulos.

La distribución observada es coherente con:

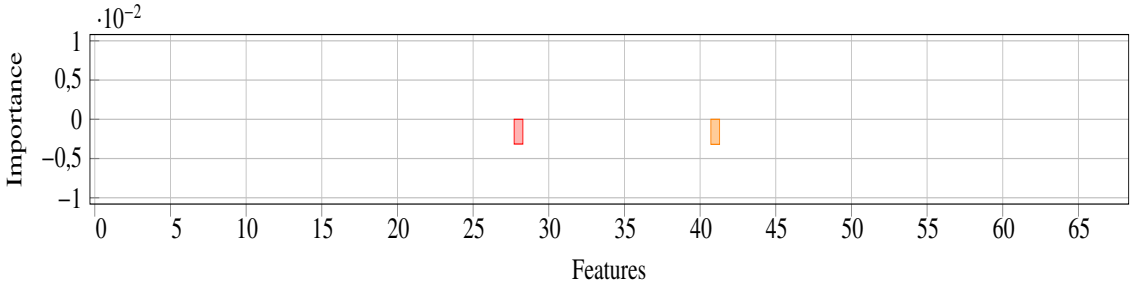
- la regla determinista de KSP-FF,
- la separación lineal entre clases, y
- la prioridad absoluta dada a la primera ruta disponible.



(a) Importancia de las características obtenida para la primera clase.



(b) Importancia de las características obtenida para la segunda clase.



(c) Importancia de las características obtenida para la segunda clase.

Figura 4.2: Importancia promedio de las características obtenida para el clasificador LGR.

Interpretación específica:

- En Clase-0 (ruta 1), la característica 28 recibe el coeficiente positivo dominante, indicando que disponibilidad en la primera ruta aumenta la probabilidad de seleccionar esa acción.
- En Clase-1 (ruta 2), la característica 41 toma relevancia cuando la característica 28 no es viable.
- En Clase-2 (ruta 3), ambas características tienen contribuciones negativas, coherentes con el hecho de que esta clase solo se elige si las dos primeras fallan.

4.1.3. Resultados del Clasificador Decision Trees

El clasificador Decision Trees (Figura 4.3) reproduce de forma jerárquica la regla determinista de la heurística:

- La característica con mayor importancia es la característica 28 (ruta 1 disponible).
- La característica 41 aparece como segunda división relevante.

Esto coincide con la estructura de un árbol de decisión: primero se evalúa la característica más discriminativa, que aquí es la disponibilidad de la primera ruta, y solo en los casos negativos se pasa a considerar la disponibilidad de la segunda.

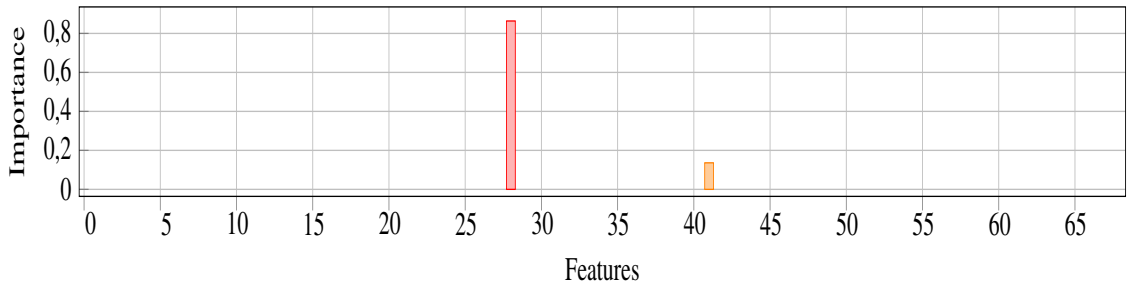


Figura 4.3: Importancia promedio de las características obtenidas para el clasificador DT.

4.1.4. Resultados del Clasificador Random Forest

El modelo Random Forest (Figura 4.4) suaviza las importancias debido al voto conjunto de múltiples árboles, pero mantiene exactamente el mismo patrón observado anteriormente:

- La característica 28 concentra la mayor parte de la importancia.
- La característica 41 aparece como segunda característica relevante.
- El resto de las características tienen importancia prácticamente nula.

Esto confirma que incluso en presencia de múltiples árboles y variaciones aleatorias, el modelo converge hacia la estructura lógica de la heurística que se pretende imitar.

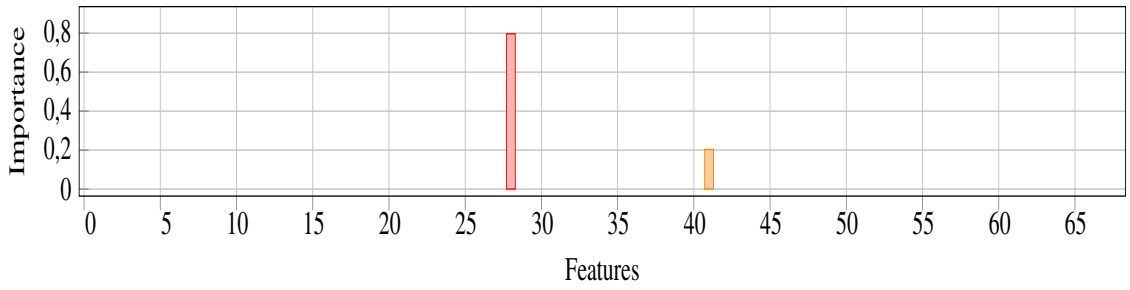


Figura 4.4: Importancia promedio de las características obtenidas para el clasificador RF.

4.1.5. Conclusión Parcial

Los cuatro clasificadores coinciden de manera absoluta en que solo dos características influyen en la toma de decisiones de la heurística K-SP-FF. Esta observación valida la metodología de interpretación bajo un escenario controlado y confirma que:

- la metodología no introduce patrones espurios,
- los clasificadores capturan correctamente la prioridad entre rutas,
- la lógica interna de la heurística queda reproducida en las importancias.

Con esta verificación, se procede al análisis interpretativo del agente DRL, cuyo comportamiento es inherentemente más complejo y no está regido por reglas deterministas.

4.2. Visión general

El objetivo principal de este trabajo consiste en ofrecer un conjunto de interpretaciones válidas de un agente entrenado para abordar problemas en redes ópticas. Para garantizar la validez de dichas interpretaciones, es fundamental que el agente demuestre un rendimiento competitivo, comparable o superior al de las soluciones existentes. En este contexto, la Fig. 4.5 presenta una evaluación de la probabilidad de bloqueo obtenida por el agente (considerado como experto), en comparación con la alcanzada mediante la heurística `K=(1,3)` de rutas más cortas disponibles con asignación tipo first-fit. Adicionalmente, con fines comparativos, se incluyen los resultados obtenidos por cuatro clasificadores distintos entrenados mediante aprendizaje por imitación.

Para diferentes cargas de tráfico, se puede observar que tanto el agente como nuestros clasificadores presentan un rendimiento intermedio entre las probabilidades de bloqueo alcanzadas por la heurística `1-SAP-FF` (la peor) y la de `3-SAP-FF` (la mejor). Esta observación nos permite asegurar que los resultados obtenidos por los distintos clasificadores son comparables con las heurísticas actualmente existentes, y por lo tanto, cualquier análisis y/o resultado referente a estos clasificadores puede ser relevante a nivel práctico. En cuanto a las capacidades de imitación, se puede ver en la Fig. 4.5 que, entre los distintos imitadores, el bosque aleatorio fue el que realizó la mejor aproximación del agente, seguido por los imitadores DT, LGR y LR, en ese orden. Esta situación se aprecia con mayor claridad en la Fig. 4.6, que muestra la matriz de confusión obtenida para cada imitador a 1000 Erlangs.

En general, el comportamiento de todos los imitadores es muy similar. En todos los casos, la mayoría de las predicciones se mantienen a lo largo de la diagonal principal de la matriz, aunque existen algunas instancias mal clasificadas. Por ejemplo, las clases 0, 5 y 10 presentan un buen desempeño, mientras que otras clases muestran más errores. Se observaron imitaciones críticas en el caso del imitador LR, donde solo la primera

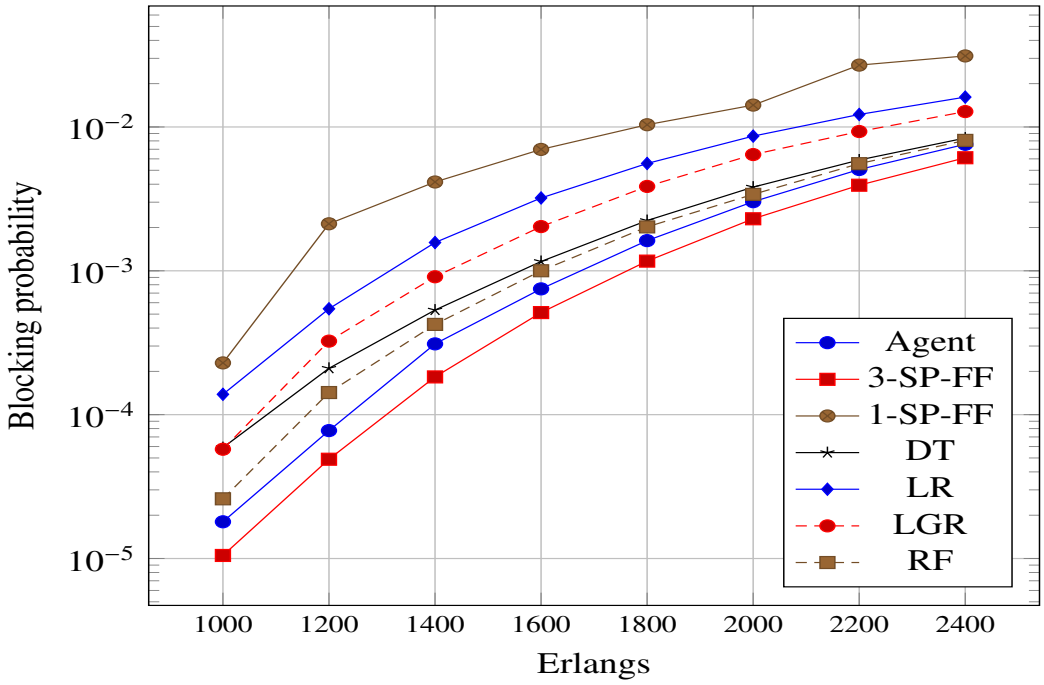
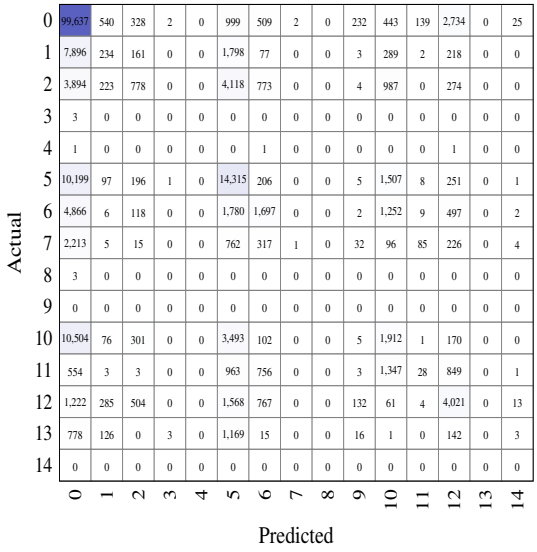
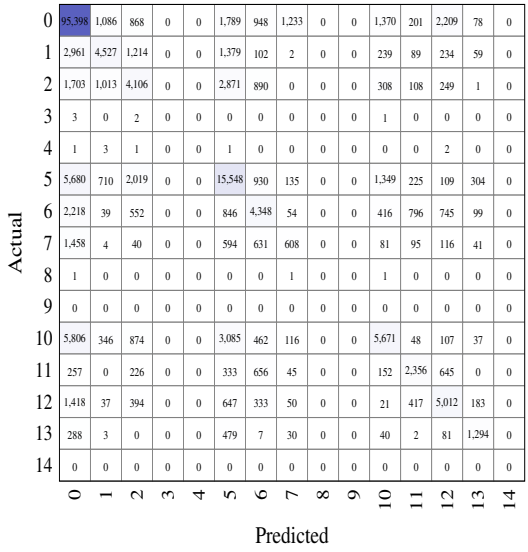


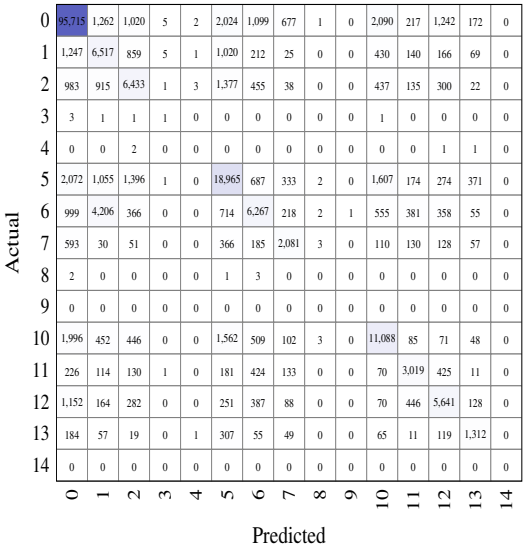
Figura 4.5: Comparación de la probabilidad de bloqueo de los distintos clasificadores frente a la heurística first-fit de rutas más cortas disponibles `K=(1,3)`, considerada como el estado del arte.



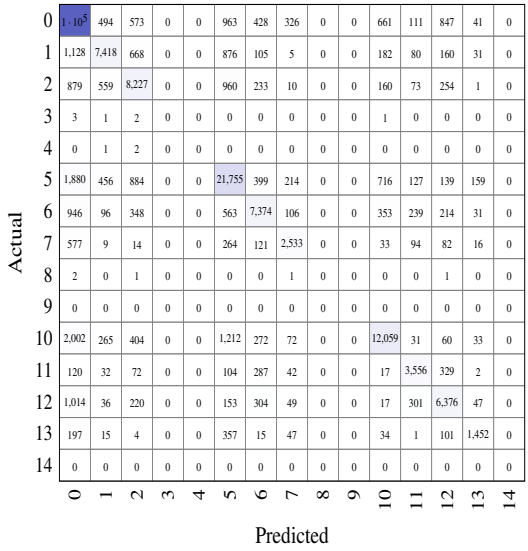
(a) Matrix de confusión del agente (actual) vs. estudiante LR (predicho)



(b) Matrix de confusión del agente vs. estudiante LGR student (predicho)



(c) Matrix de confusión del agente (actual) vs. estudiante DT (predicho)



(d) Matrix de confusión del agente (actual) vs. estudiante RF (predicho)

Figura 4.6: Comparación de las matrices de confusión de los imitadores LR, LGR, DT y RF respecto a la política del agente.

clase logró un buen rendimiento de imitación. Además, los errores de imitación ocurren con mayor frecuencia en clases con menos muestras. Consideramos que esto se debe al desbalanceo en los datos de entrenamiento. Una forma de resolver este problema es utilizar una cantidad similar de instancias durante el proceso de imitación. Sin embargo, las instancias se recopilan del agente, el cual, a su vez, prefiere el uso de algunas acciones sobre

Tabla 4.1: Métricas de clasificación obtenidas para los distintos imitadores.

	Metric	Actions														
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
LR	F1-score	0.81	0.04	0.12	-	-	0.50	0.22	0.00	-	-	0.16	0.01	0.45	-	-
	Precision	0.70	0.15	0.32	0.00	-	0.46	0.33	0.33	-	0.00	0.24	0.10	0.43	-	0.00
	Recall	0.94	0.02	0.07	0.00	0.00	0.53	0.17	0.00	0.00	-	0.12	0.01	0.47	0.00	-
LGR	F1-score	0.86	0.49	0.38	-	-	0.57	0.45	0.20	-	-	0.43	0.52	0.56	0.60	-
	Precision	0.81	0.58	0.40	-	-	0.56	0.47	0.27	-	-	0.59	0.54	0.53	0.62	-
	Recall	0.91	0.42	0.37	0.00	0.00	0.58	0.43	0.17	0.00	-	0.34	0.50	0.59	0.58	-
DT	F1-score	0.91	0.61	0.58	0.10	-	0.71	0.61	0.56	-	-	0.67	0.64	0.65	0.59	-
	Precision	0.91	0.60	0.58	0.07	0.00	0.71	0.61	0.56	0.00	0.00	0.67	0.64	0.65	0.58	-
	Recall	0.91	0.61	0.58	0.14	0.00	0.70	0.62	0.56	0.00	-	0.68	0.64	0.66	0.60	-
RF	F1-score	0.92	0.79	0.72	-	-	0.80	0.77	0.74	-	-	0.85	0.77	0.75	0.80	-
	Precision	0.97	0.70	0.72	-	-	0.81	0.72	0.68	-	-	0.73	0.78	0.75	0.65	-
	Recall	0.97	0.70	0.72	-	-	0.81	0.72	0.68	-	-	0.73	0.78	0.75	0.65	-

otras. En este sentido, recolectar la misma cantidad de instancias para el entrenamiento implica ejecutar un gran número de simulaciones y reducir los resultados. Este enfoque, aunque más efectivo, excede el alcance de nuestro análisis.

En cambio, los resultados de clasificación presentados en la Tabla 4.1 destacan diferencias notables en el rendimiento de los modelos evaluados. El clasificador de Regresión Lineal (LR) muestra una efectividad limitada en la predicción multiclase. Aunque alcanza un F1-score alto (0.81) para la acción 0, con una precisión de 0.70 y un recall de 0.94, su rendimiento es considerablemente deficiente en la mayoría de las demás clases. Varias acciones presentan puntuaciones cercanas a cero o simplemente no son predichas, lo que indica una capacidad de generalización insuficiente, especialmente en presencia de distribuciones de clases desequilibradas o escasas.

El modelo de Regresión Logística (LGR) tiene un desempeño mejor que LR, con puntuaciones más altas en un conjunto más amplio de clases. Se observan F1-scores superiores a 0.4 en las acciones 0, 1, 2, 5, 11, 12 y 13, lo que refleja un balance más estable entre precisión y recall. Sin embargo, el modelo aún falla en capturar acciones menos frecuentes, como las clases 4 y 8. A pesar de estas limitaciones, LGR constituye una línea base más confiable que LR para escenarios que requieren precisión moderada y capacidad de interpretación.

En comparación, el clasificador basado en Árboles de Decisión (DT) ofrece una alternativa más robusta e interpretable. Alcanza valores de F1-score superiores a 0.5 en

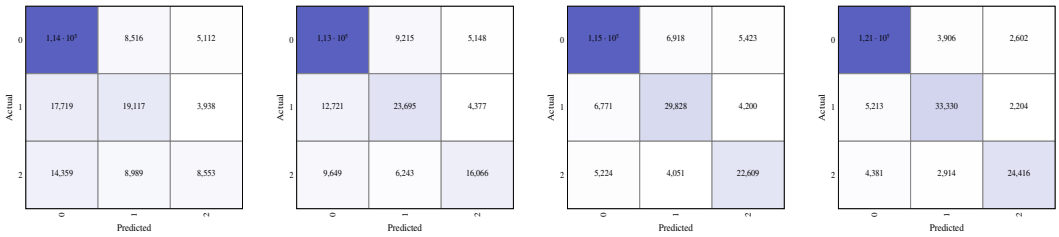
la mayoría de las clases, incluyendo las acciones 0, 2, 5, 7, 11 y 12, y mantiene valores consistentes de precisión y recall. Aunque no predice ciertas clases (por ejemplo, 4 y 8) y tiene un rendimiento bajo en la clase 3, DT proporciona una relación favorable entre complejidad del modelo e interpretabilidad, lo que lo hace adecuado para comprender el problema de ruteo y asignación espectral.

Finalmente, el clasificador de Bosques Aleatorios (RF) demuestra el rendimiento general más alto. Produce de forma consistente F1-scores superiores a 0.70 en la mayoría de las clases predichas, con valores de precisión y recall que alcanzan hasta 0.97. Sus resultados son especialmente sólidos en las acciones 0, 2, 5, 7, 11, 12 y 13. Aunque comparte algunas de las mismas limitaciones que DT al manejar clases poco frecuentes, RF destaca por su precisión predictiva y robustez. Su principal desventaja es su menor interpretabilidad en comparación con los modelos de árbol único; sin embargo, su confiabilidad general lo convierte en un fuerte candidato para lograr interpretaciones confiables de modelos DRL.

4.3. Análisis de imitación en el ruteo y la asignación espectral

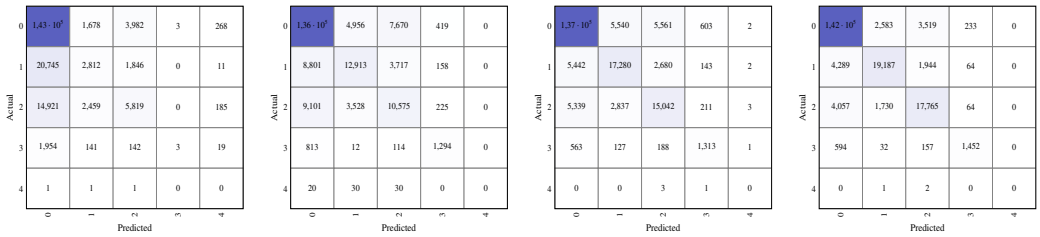
Un análisis más profundo del proceso de imitación puede llevarse a cabo considerando por separado los problemas de enrutamiento y asignación espectral. Las Figuras 4.7 y 4.8 muestran la matriz de confusión obtenida por cada uno de los cuatro clasificadores para el ruteo y la asignación espectral, respectivamente.

Por un lado, a partir de la Fig. 4.7, se puede observar que todos los clasificadores tienden a seleccionar la primera ruta para resolver el subproblema de ruteo. Muchos métodos heurísticos comúnmente utilizados para resolver problemas de ruteo aprovechan esta tendencia. Por lo tanto, prevemos que estos clasificadores exhiben un comportamiento que se alinea con los métodos existentes. Sin embargo, distinguimos un comportamiento inusual al utilizar las rutas 2 y 3. El uso de las rutas 2 y 3 es muy similar entre los distintos clasificadores, lo que sugiere que si el experto no puede asignar la demanda en la primera ruta, intenta una transmisión aleatoria entre las rutas restantes. Esta estrategia, a pesar de lograr buenos resultados, nunca ha sido estudiada en la literatura.



(a) LR accuracy: 0.706835 (b) LGR accuracy: 0.763235 (c) DT accuracy: 0.837065 (d) RF accuracy: 0.8939

Figura 4.7: Actual (agent) vs. predicted (student) confusion matrix comparison for the routing subproblem



(a) LR accuracy: 0.758215(b) LGR accuracy: 0.80239 (c) DT accuracy: 0.85377 (d) RF accuracy: 0.903655

Figura 4.8: Actual (agent) vs. predicted (student) confusion matrix comparison for the spectrum assignment subproblem

Un análisis paralelo puede extenderse para comprender la toma de decisiones del experto al seleccionar un bloque de FSUs para la asignación de la demanda. Según la Fig. 4.8, todos los clasificadores prefirieron la primera acción, lo que indica una preferencia por el primer bloque disponible dentro de una ruta. Este comportamiento está estrechamente alineado con heurísticas establecidas para la asignación espectral, como `3-SAP-FF`, donde se pretende compactar el uso del espectro tanto como sea posible para mantener libre el extremo del espectro para futuras solicitudes de conexión. Además, en todos los casos, es notable que la probabilidad de uso disminuye a medida que aumenta el número de bloques. Como resultado, este comportamiento, junto con la observación anterior, sugiere que el agente no solo intenta estrictamente utilizar el primer bloque, sino que bajo ciertas condiciones, también utiliza otros bloques disponibles que ofrecen mejores resultados. Este comportamiento es similar al algoritmo de ajuste óptimo (best-fit), donde el objetivo es encontrar el espacio más adecuado (es decir, el que tiene el tamaño exacto o más cercano a la solicitud) para reducir la fragmentación. En consecuencia, observamos que el agente combina dos enfoques para resolver el subproblema de asignación espectral. Finalmente, como se muestra en la Fig. 4.8, ninguno de los cuatro imitadores fue capaz de replicar

exitosamente la quinta acción. Entendemos que esto se debe principalmente a dos factores: la cantidad limitada de muestras de entrenamiento y la gran complejidad involucrada en distinguir esta acción particular del resto del conjunto de acciones.

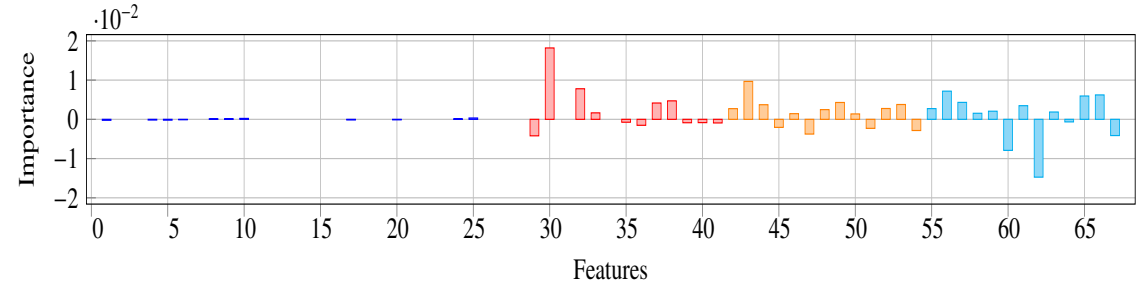
Una observación importante a partir de las Fig. 4.7,4.8 puede hacerse si comparamos la precisión obtenida para cada uno de los subproblemas de ruteo y asignación espectral. En este sentido, cada uno de los cuatro clasificadores obtuvo una mejor precisión en el subproblema de asignación espectral. Esto significa que el subproblema de enrutamiento es más complejo de predecir y, por lo tanto, es más difícil entender por qué el agente decide usar una ruta sobre otra.

4.4. Interpretación de las acciones de ruteo y asignación espectral

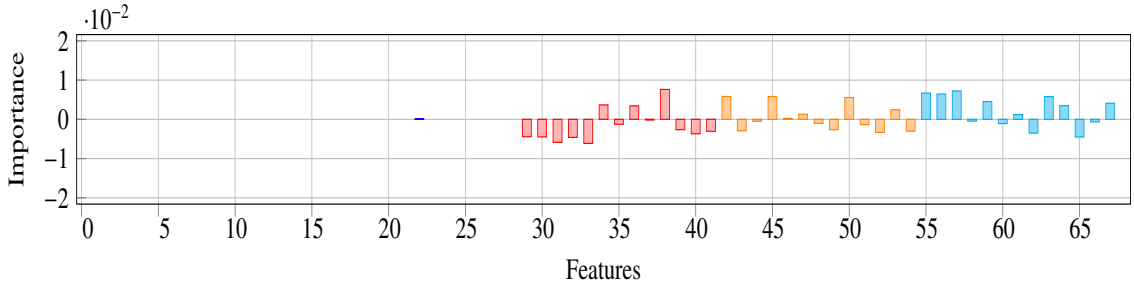
Nuestra investigación demuestra que los cuatro clasificadores son capaces de imitar el comportamiento del agente experto basado en DRL. Esto sugiere que los conocimientos obtenidos del análisis de los clasificadores pueden aplicarse al agente experto con un grado razonable de confianza, siendo el nivel de incertidumbre determinado por la precisión de los imitadores. Los cuatro clasificadores analizados se dividen en dos grupos: dos basados en métodos de regresión y dos en algoritmos de árboles de decisión. Debido a las similitudes operativas y resultados comparables dentro de estos grupos, nos enfocaremos en los dos clasificadores con mejor rendimiento de cada uno (regresión logística y bosque aleatorio) para proporcionar una comprensión más clara y completa de los resultados, destacando su rendimiento y características distintivas.

4.4.1. Interpretaciones basadas en algoritmos de regresión

En los clasificadores de regresión logística multinomial, la clase de salida se determina por la mayor probabilidad predicha asociada a cada clase. Dado un conjunto de características de entrada, la probabilidad para cada clase depende de los pesos asociados a cada característica. En este sentido, el peso asignado a cada característica refleja su contribución a la probabilidad de que la muestra pertenezca a una clase particular. Por ejemplo, la Fig. 4.9 refleja la contribución de los pesos de cada característica en la predic-



(a) Importancia de las características obtenida para la primera clase.



(b) Importancia de las características obtenida para la segunda clase.

Figura 4.9: Importancia promedio de las características obtenida para el clasificador LGR.

ción de la primera (Fig. 4.9a) y segunda (Fig. 4.9b) clase. Las características se agrupan en dos bloques: aquellas indexadas desde $\mathcal{X}[1]$ hasta $\mathcal{X}[28]$, que se relacionan con los nodos de origen y destino de las solicitudes de conexión, y aquellas desde $\mathcal{X}[29]$ hasta $\mathcal{X}[67]$, que entregan información sobre la disponibilidad espectral a lo largo de las tres rutas analizadas.

Para la primera clase, el primer bloque de características (1–28) muestra baja importancia, con todos los valores cercanos a cero, lo que indica que la información de ruteo origen-destino tiene una influencia mínima en la predicción de esta clase. A medida que avanzamos a los bloques segundo a cuarto (características desde la 29 en adelante), hay una variación notable en la importancia. En particular, las características 30, 43 y 62 muestran picos significativos, lo que sugiere que ciertas condiciones o características asociadas a estas variables influyen fuertemente en la predicción de la primera clase. En particular, las características 30 y 43 corresponden a información relacionada con el número de espacios libres consecutivos del primer bloque disponible en las dos primeras rutas; mientras que la característica 62 corresponde al tamaño del cuarto bloque disponible en la tercera ruta. Por tanto, la probabilidad de ocurrencia de esta clase está altamente influenciada por el tamaño del primer bloque disponible en las dos primeras rutas, mientras que el tamaño del cuarto

bloque en la tercera ruta (característica 62) impacta negativamente esta tendencia.

Por el contrario, en la Fig. 4.9b, los patrones de importancia en el primer bloque de características son igualmente bajos, lo que demuestra un patrón consistente de mínima influencia de la información de ruteo origen-destino en ambas clases. Sin embargo, en los bloques posteriores, los niveles de importancia son en general más bajos comparados con la primera clase, particularmente alrededor de las características donde la primera clase mostró picos significativos. Esta tendencia sugiere que, si bien estas características son críticas para predecir la primera clase, tienen menor peso decisivo en la predicción de la segunda clase.

Finalmente, los picos pronunciados en el primer gráfico comparados con la importancia relativamente atenuada en el segundo podrían indicar una inclinación del modelo hacia la primera clase. Esta observación está alineada con los comentarios realizados anteriormente, donde los agentes priorizan la primera clase sobre las demás.

4.4.2. Interpretaciones basadas en algoritmos de árboles de decisión

En la Fig. 4.10, mostramos los puntajes de importancia de características derivados del bosque aleatorio para una carga de tráfico de 1000 Erlangs. Entre estas características, las primeras 28 resultaron ser poco significativas, cada una con puntajes por debajo de 5×10^{-3}. Incluso las características $\mathcal{X}[14]$ y $\mathcal{X}[15]$ fueron consideradas irrelevantes en el proceso de decisión del bosque aleatorio. Esto sugiere que el conocimiento de los nodos de origen y destino podría no ser crucial para el proceso de toma de decisiones del experto, y eliminarlas podría mejorar la eficiencia del entrenamiento de la política del agente.

En contraste, las características que abarcan desde $\mathcal{X}[29]$ hasta $\mathcal{X}[67]$ encapsulan un 97.9 % de toda la información relevante del experimento. Dentro de este subconjunto, identificamos tres grupos de 13 características cada uno. Las primeras diez características de cada grupo representan el índice y tamaño de los primeros cinco bloques de FSUs disponibles en cada ruta. En comparación, las tres características restantes indican la demanda de slots de servicio, el total de slots disponibles en la ruta, y el promedio de slots dentro de los bloques disponibles (como se discute en la Sección 3.2.2, *Configuración del agente DRL*).

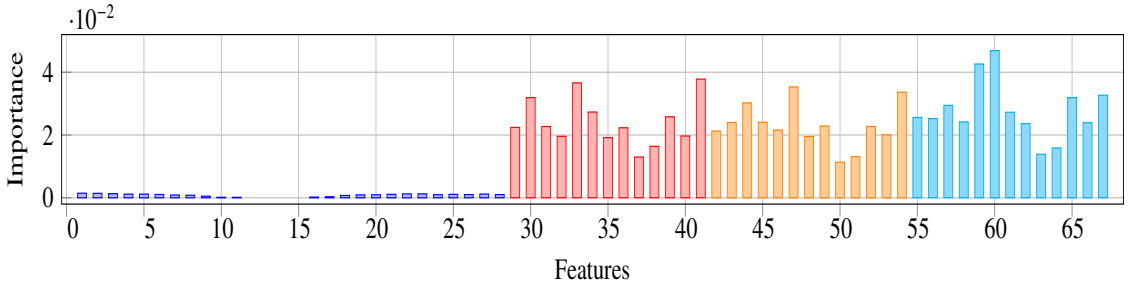


Figura 4.10: Importancia promedio de las características obtenidas para el clasificador RF.

En general, las características que indican los índices de los bloques fueron más importantes que aquellas que indican el tamaño de los bloques en todas las rutas. Esto sugiere que el proceso de toma de decisiones del experto está influenciado en gran medida por la posición de los bloques dentro del espectro de frecuencias más que por la cantidad de slots que contienen. Además, las características críticas se concentraron alrededor de los terceros bloques de índice. Estos bloques suelen aparecer en la región central del espectro. Por ejemplo, en la segunda y tercera rutas, las características clave fueron $\mathcal{X}[47]$ y $\mathcal{X}[60]$, que denotan el índice del tercer bloque disponible de FSUs. Esta observación sugiere que los bloques ubicados en el centro del espectro entregan ligeramente más información relevante para la toma de decisiones del experto en comparación con otras características.

Además, la última característica dentro de cada grupo mostró una importancia notable, con un puntaje promedio de $3{,}3 \times 10^{-2}$. Esta característica indica el número promedio de slots libres en los bloques disponibles, por lo que un valor alto en esta característica sugiere la importancia del tamaño del bloque en el proceso de toma de decisiones del modelo experto DRL.

5 | Conclusiones

El tema propuesto en este trabajo aborda el desafío de la caja negra presente en los algoritmos de aprendizaje por refuerzo profundo (DRL) aplicados a redes ópticas. Utilizando aprendizaje por imitación para entrenar diferentes clasificadores, se logran distintos niveles de interpretabilidad que permiten comprender el proceso de toma de decisiones de un modelo DRL entrenado para resolver el problema RSA. Al interpretar el comportamiento del agente en elementos comprensibles, se identifican estrategias efectivas que pueden transformarse en soluciones implementables para optimizar la gestión de recursos en redes reales. Esto incrementa la confianza en las soluciones generadas y facilita la creación de métodos más livianos y explicables, adecuados para escenarios operativos más amplios.

Nuestro análisis revela que el subproblema de enrutamiento es más difícil de predecir que el de asignación espectral, ya que todos los clasificadores obtienen una mayor precisión en este último. Todos los clasificadores favorecieron sistemáticamente la primera ruta en sus decisiones, lo que es coherente con los métodos heurísticos comúnmente utilizados. Sin embargo, la selección de la segunda y tercera ruta mostró un patrón más aleatorio, lo que sugiere que, más allá de la primera ruta, el proceso de toma de decisiones es menos estructurado.

El análisis de la importancia de las características mostró que la información del par origen-destino tuvo poco impacto en el comportamiento del agente, mientras que el tamaño del bloque jugó un papel clave en su proceso de decisión. Los clasificadores basados en árboles de decisión destacaron la relevancia de los bloques ubicados en las regiones centrales del espectro, lo que indica que estas características influyen en las decisiones del

agente para mantener la disponibilidad del espectro en el futuro. Estos hallazgos ofrecen información valiosa sobre el proceso de decisión del agente y pueden orientar mejoras en los modelos DRL aplicados a redes ópticas.

En general, el marco presentado en este estudio no solo proporciona una vía para interpretar las decisiones de los agentes DRL en redes ópticas, sino que también permite extraer patrones valiosos que pueden servir como base para diseñar protocolos, reglas o heurísticas más comprensibles y aplicables en contextos prácticos.

Desde una perspectiva metodológica, este trabajo demuestra que es posible integrar mecanismos de interpretabilidad a modelos DRL sin comprometer su aplicabilidad. Los clasificadores utilizados no solo permitieron emular el comportamiento del agente, sino también proporcionar explicaciones útiles para los operadores de red, favoreciendo la confianza en las decisiones automatizadas. A nivel operativo, esto puede traducirse en una mayor aceptación de tecnologías basadas en inteligencia artificial en contextos críticos como la gestión de infraestructura óptica, donde la capacidad de justificar acciones es tan importante como su eficacia.

En términos de aportes prácticos, este estudio contribuye con un framework completo de simulación, entrenamiento e interpretación, que puede ser extendido a otros escenarios o aplicado a diferentes tipos de redes y políticas. Las técnicas de interpretación utilizadas aquí podrían integrarse a sistemas SDN (Software Defined Networks) para construir capas de control explicables y auditables, un requisito fundamental para el despliegue de soluciones autónomas en redes de producción.

Finalmente, esta investigación sienta las bases para futuras líneas de trabajo. Entre ellas se destacan: (i) el diseño de nuevas representaciones del estado que incorporen variables más explicativas del entorno, (ii) el desarrollo de modelos híbridos que combinen agentes DRL con reglas explícitas derivadas de los clasificadores imitadores, y (iii) la evaluación del framework en escenarios de mayor escala y realismo. También se abre la posibilidad de aplicar este enfoque a otros dominios de telecomunicaciones donde la transparencia y la fiabilidad sean factores clave.

En conclusión, esta tesis demuestra que la interpretabilidad no es un atributo incompatible con la inteligencia artificial aplicada a redes ópticas, sino una condición necesaria

para su adopción responsable. Al dotar de explicaciones comprensibles a modelos complejos como DRL, no solo se mejora su supervisión y validación, sino que se promueve un avance hacia redes más autónomas, resilientes y humanas en su interacción.

Bibliografía

Amirabadi, MA; Nezamalhosseini, SA; Kahaei, MH; y Chen, Lawrence R (2024). A survey on machine and deep learning for optical communications. *arXiv preprint arXiv:2412.17826*. 1

Andriolli, Nicola; Giorgetti, Alessio; Castoldi, Piero; Cecchetti, Gabriele; Cerutti, Isabella; Sambo, Nicola; Sgambelluri, Andrea; Valcarenghi, Luca; Cugini, Filippo; Martini, Barbara; et al. (2022). Optical networks management and control: A review and recent challenges. *Optical Switching and Networking*, 44, 100652. 2.1

Bastani, Osbert; Pu, Yewen; y Solar-Lezama, Armando (2018). Verifiable reinforcement learning via policy extraction. *Advances in neural information processing systems*, 31. 2.4.1

Brown, Alexander y Petrik, Marek (2018). Interpretable reinforcement learning with ensemble methods. *arXiv preprint arXiv:1809.06995*. 2.4.1

Calderon, Felipe Ignacio; Lozada, Astrid; Borquez-Paredes, Danilo; Olivares, Ricardo; Davalos, Enrique Javier; Saavedra, Gabriel; Jara, Nicolas; y Leiva, Ariel (2020). BER-Adaptive RMLSA Algorithm for Wide-Area Flexible Optical Networks. *IEEE Access*, 8, 128018–128031. 3.2.1

Carvalho, Diogo V.; Pereira, Eduardo M.; y Cardoso, Jaime S. (2019). Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8). 1

Chakraborti, Tathagata; Sreedharan, Sarath; Grover, Sachin; y Kambhampati, Subbarao (2019). Plan explanations as model reconciliation – an empirical study. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)* (pp. 258–266). 3

Chen, X.; Proietti, R.; Lu, H.; Castro, A.; y Yoo, S. J. B. (2018). Knowledge-based autonomous service provisioning in multi-domain elastic optical networks. *IEEE Communications Magazine*, 56(8), 152–158. 1

Cheng, Zelei; Yu, Jiahao; y Xing, Xinyu (2025). A survey on explainable deep reinforcement learning. *arXiv preprint arXiv:2502.06869*. 1, 2.1

Cisco (2018). Cisco visual networking index: forecast and methodlogy, 2012-2017. 1

Coppens, Youri; Efthymiadis, Kyriakos; Lenaerts, Tom; Nowé, Ann; Miller, Tim; Weber, Rosina; y Magazzeni, Daniele (2019). Distilling deep reinforcement learning policies in

soft decision trees. In *Proceedings of the IJCAI 2019 workshop on explainable artificial intelligence* (pp. 1–6). 2.4.1

Criminisi, Antonio; Shotton, Jamie; y Konukoglu, Ender (2011). Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 7(2-3), 81–227. 3.1.3

Cruzes, Sergio (2023). Optical networks automation overview: A survey. *J Sen Net Data Comm*, 3(1), 144–162. 2.1

Cruzes, Sergio (2025). Revolutionizing optical networks: The integration and impact of large language models. *Optical Switching and Networking*, 57, 100812. 1

Dhurandhar, Amit; Chen, Pin-Yu; Luss, Ronny; Tu, Chun-Chen; Ting, Paishun; Shanmugam, Karthikeyan; y Das, Payel (2018). Explanations based on the missing: Towards contrastive explanations with pertinent negatives. *Advances in neural information processing systems*, 31. 2.4.2

Doherty, Michael; Matzner, Robin; Sadeghi, Rasoul; Bayvel, Polina; y Beghelli, Alejandra (2025). Reinforcement learning for dynamic resource allocation in optical networks: hype or hope? *Journal of Optical Communications and Networking*, 17(9), D1–D17. 2.1

Etezadi, Ehsan; Natalino, Carlos; Diaz, Renzo; Lindgren, Anders; Melin, Stefan; Wosinska, Lena; Monti, Paolo; y Furdek, Marija (2023). Deep reinforcement learning for proactive spectrum defragmentation in elastic optical networks. *Journal of Optical Communications and Networking*, 15(10), E86–E96. 2.1

Galimberti, Gabriele y Sambo, Nicola (2021). Rmlsa algorithms for elastic optical networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 23(2), 968–1002. 3

Glanois, Claire; Weng, Paul; Zimmer, Matthieu; Li, Dong; Yang, Tianpei; Hao, Jianye; y Liu, Wulong (2021). A survey on interpretable reinforcement learning. *arXiv preprint arXiv:2112.13112*. 1, 2.2

Haarnoja, Tuomas; Zhou, Aurick; Abbeel, Pieter; y Levine, Sergey (2018). Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*. 3.1.1

Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome H; y Friedman, Jerome H (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer. 2.2.1, 2.2.2, 2.2.3, 2.2.3, 3.1.3, 3.2.3

Hilbe, Joseph M (2009). *Logistic regression models*. Chapman and hall/CRC. 2.2.2

Hinton, Geoffrey; Vinyals, Oriol; y Dean, Jeff (2015). Distilling the knowledge in a neural network (2015). *arXiv preprint arXiv:1503.02531*, 2. 2.4.1

Hosmer Jr, David W; Lemeshow, Stanley; y Sturdivant, Rodney X (2013). *Applied logistic regression*, volume 398. John Wiley & Sons. 2.2.2

Hussenot, Robin; Marchand, Nicolas; y Laurent, Jérémy (2020). A survey on imitation learning for robotics. *Robotics and Autonomous Systems*, 133, 103647. 3.1.2

Ives, David J; Bayvel, Polina; y Savory, Seb J (2015). Routing, modulation, spectrum and launch power assignment to maximize the traffic throughput of a nonlinear optical mesh network. *Photonic Network Communications*, 29(3), 244–256. 3.2

Ji, Yuefeng; Zhang, Jiawei; Wang, Xin; y Yu, Hao (2018). Towards converged, collaborative and co-automatic (3c) optical networks. *Science China Information Sciences*, 61(12), 1–19. 1

Joshi, Ameet V. (2020). *Decision Trees*, (pp. 53–63). Springer International Publishing: Cham. 2.2.3

Kazemitabar, Jalil; Amini, Arash; Bloniarz, Adam; y Talwalkar, Ameet S (2017). Variable importance using decision trees. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, y R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, volume 30: Curran Associates, Inc. 2.2.3

Kazhdan, Dmitry; Shams, Zohreh; y Lio, Pietro (2020). Marleme: A multi-agent reinforcement learning model extraction library. In *2020 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8). 4

Khan, Omar; Poupart, Pascal; y Black, James (2009). Minimal sufficient explanations for factored markov decision processes. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 19 (pp. 194–200). 2.4.2

Levine, Sergey; Finn, Chelsea; Darrell, Trevor; y Abbeel, Pieter (2020). Reinforcement learning and control as probabilistic inference: Tutorial and review. *Foundations and Trends in Robotics*, 8(1-2), 1–179. 3.1.1

Liu, Dong C y Nocedal, Jorge (1989). On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1), 503–528. 3.2.3

Liu, Hongfang; Xiao, Yong; Wang, Xinbo; Xiao, Guoqiang; Guo, Lei; y Zhang, Wei (2021). Optical network resource optimization: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 23(2), 867–900. 3.1.1

López, Víctor y Velasco, Luis, Eds. (2016). *Elastic Optical Networks*. Optical Networks. Cham: Springer International Publishing. 1

Luo, Xiao; Zhao, Yang; Chen, Xue; Wang, Lei; Zhang, Min; Zhang, Jie; Ji, Yuefeng; Wang, Huitao; y Wang, Taili (2017). Manycast routing, modulation level and spectrum assignment over elastic optical networks. *Optical Fiber Technology*. 1

Luong, Nguyen Cong; Hoang, Dinh Thai; Gong, Shimin; Niyato, Dusit; Wang, Ping; Liang, Ying-Chang; y Kim, Dong In (2019). Applications of deep reinforcement learning in communications and networking: A survey. *IEEE communications surveys & tutorials*, 21(4), 3133–3174. 1

Maimon, Oded Z y Rokach, Lior (2014). *Data mining with decision trees: theory and applications*, volume 81. World scientific. 2.2.3, 2.2.3

Martín, I.; Hernández, J. A.; Troia, S.; Musumeci, F.; Maier, G.; y de Dios, O. G. (2018). Is machine learning suitable for solving rwa problems in optical networks? In *2018 European Conference on Optical Communication (ECOC)* (pp. 1–3). 1

Mata, Javier; de Miguel, Ignacio; Durán, Ramón J.; Merayo, Noemí; Singh, Sandeep Kumar; Jukan, Admela; y Chamania, Mohit (2018). Artificial intelligence (AI) methods in optical networks: A comprehensive survey. *Optical Switching and Networking*, 28, 43–57. 1

Molnar, Christoph (2020). *Interpretable machine learning*. Lulu.com. 2.2.3, 2.3, 2, 3.2.3

Mukherjee, B.; Tomkos, I.; Tornatore, M.; Winzer, P.; y Zhao, Y. (2020). *Springer Handbook of Optical Networks*. Springer Handbooks. Springer International Publishing. 1

Natalino, Carlos; Panahi, Ashkan; Mohammadiha, Nasser; y Monti, Paolo (2024). Ai/ml-as-a-service for optical network automation: use cases and challenges. *J. Opt. Commun. Netw.*, 16(2), A169–A179. 1

Nikulin, Dmitry; Ianina, Anastasia; Aliev, Vladimir; y Nikolenko, Sergey (2019). Free-lunch saliency via attention in atari agents. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)* (pp. 4240–4249).: IEEE. 2.4.2

Oliveira, Helder MNS y da Fonseca, Nelson LS (2017). Routing, spectrum, core and modulation level assignment algorithm for protected sdm optical networks. In *GLOBECOM 2017-2017 IEEE Global Communications Conference* (pp. 1–6).: IEEE. 1

Ouyang, Kangao; Tang, Fengxian; Yuan, Zhilin; Li, Jun; y Li, Yongcheng (2025). Static and dynamic routing, fiber, modulation format, and spectrum allocation in hybrid ull fiber-ssmf elastic optical networks. *IEEE Access*. 1

Papenmeier, Andrea; Englebienne, Gwenn; y Seifert, Christin (2019). How model accuracy and explanation fidelity influence user trust. *arXiv preprint arXiv:1907.12652*. 4

Paz, Esteban y Saavedra, Gabriel (2020). Maximum transmission reach for optical signals in elastic optical networks employing band division multiplexing. *arXiv preprint arXiv:2011.03671*. 1

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; y Brucher (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830. 3.2.3

Peng, Xue; Tan, Mingkui; Zhu, Jingqing; Liu, Yaodong; Hong, Kai; Yang, Zihan; Tang, Jianda; Liu, Qihang; Zhang, Yuhao; Zhang, Jianye; et al. (2021). Deep reinforcement learning for imitation learning: A survey. *arXiv preprint arXiv:2108.01453*. 3.1.2

Perner, Petra (2011). How to interpret decision trees? In P. Perner (Ed.), *Advances in Data Mining. Applications and Theoretical Aspects* (pp. 40–55). Berlin, Heidelberg: Springer Berlin Heidelberg. 2.2.3

Pinto-Ríos, Juan; Calderón, Felipe; Leiva, Ariel; Hermosilla, Gabriel; Beghelli, Alejandra; Bórquez-Paredes, Danilo; Lozada, Astrid; Jara, Nicolás; Olivares, Ricardo; y Saavedra, Gabriel (2023). Resource allocation in multicore elastic optical networks: A deep reinforcement learning approach. *Complexity*, 2023(1), 4140594. 2.1

Pointurier, Yvan (2017). Design of low-margin optical networks. *Journal of Optical Communications and Networking*, 9(1), A9–A17. 1

Raffin, Antonin; Hill, Ashley; Gleave, Adam; Kanervisto, Anssi; Ernestus, Maximilian; y Dormann, Noah (2021). Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268), 1–8. 3.2.2

Ross, Stéphane; Gordon, Geoffrey; y Bagnell, Drew (2011). A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (pp. 627–635).: JMLR Workshop and Conference Proceedings. 3.1.2

Roth, Aaron M; Topin, Nicholay; Jamshidi, Pooyan; y Veloso, Manuela (2019). Conservative q-improvement: Reinforcement learning for an interpretable decision-tree policy. *arXiv preprint arXiv:1907.01180*. 2.4.1

Rudin, Cynthia (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5), 206–215. 3.1.2

Schulman, John; Wolski, Filip; Dhariwal, Prafulla; Radford, Alec; y Klimov, Oleg (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*. 3.2.2

Seber, George AF y Lee, Alan J (2012). *Linear regression analysis*. John Wiley & Sons. 2.2.1

Shrikumar, Avanti; Greenside, Peyton; y Kundaje, Anshul (2017). Learning important features through propagating activation differences. In *International conference on machine learning* (pp. 3145–3153).: PMLR. 2.4.2

Simonyan, Karen; Vedaldi, Andrea; y Zisserman, Andrew (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*. 2.4.2

Song, Yuchen; Zhang, Min; Zhang, Yao; Shi, Yan; Shen, Shikui; Tang, Xiongyan; Huang, Shanguo; y Wang, Danshi (2025). Lifecycle management of optical networks with dynamic-updating digital twin: a hybrid data-driven and physics-informed approach. *IEEE Journal on Selected Areas in Communications*. 1

Sutton, Richard S y Barto, Andrew G (2018). Reinforcement learning: An introduction. *MIT press*. 3.1.1

Terki, Abdennour Ben; Pedro, João; Eira, António; Napoli, Antonio; y Sambo, Nicola (2024). Deep reinforcement learning for resource allocation in multi-band optical networks. In *2024 International Conference on Optical Network Design and Modeling (ONDM)* (pp. 1–4).: IEEE. 2.1

Tizikara, Dativa K; Serugunda, Jonathan; y Katumba, Andrew (2022). Machine learning-aided optical performance monitoring techniques: A review. *Frontiers in communications and networks*, 2, 756513. 1

Van Hasselt, Hado; Guez, Arthur; y Silver, David (2016). Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30. 2.4.2

Vasic, Marko; Petrovic, Andrija; Wang, Kaiyuan; Nikolic, Mladen; Singh, Rishabh; y Khurshid, Sarfraz (2019). Moet: Interpretable and verifiable reinforcement learning via mixture of expert trees. 2.4.1

Vouros, George A (2022). Explainable deep reinforcement learning: state of the art and challenges. *ACM Computing Surveys*, 55(5), 1–39. 1, 2.1

Wang, Ning; Pynadath, David V; y Hill, Susan G (2016a). Trust calibration within a human-robot team: Comparing automatically generated explanations. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)* (pp. 109–116).: IEEE. 2.4.2

Wang, Ziyu; Schaul, Tom; Hessel, Matteo; Hasselt, Hado; Lanctot, Marc; y Freitas, Nando (2016b). Dueling network architectures for deep reinforcement learning. In *International conference on machine learning* (pp. 1995–2003).: PMLR. 2.4.2

Wei, Yibo; Zou, Deqing; Ma, Yu; y Wang, Zhong (2021). Ensemble learning and its applications to network security: A comprehensive review. *Future Generation Computer Systems*, 117, 298–319. 3.1.3

Weisberg, Sanford (2005). *Applied linear regression*, volume 528. John Wiley & Sons. 2.2.1

Won, Rachel (2025). Beating the capacity crunch in optical links: Silicon photonics. *Nature Reviews Electrical Engineering*, (pp. 1–1). 2.1

Yan, Xin y Su, Xiaogang (2009). *Linear regression analysis: theory and computing*. world scientific. 2.2.1

Yang, Zhao; Bai, Song; Zhang, Li; y Torr, Philip HS (2018). Learn to interpret atari agents. *arXiv preprint arXiv:1812.11276*. 2.4.2

Yu, J.; Cheng, B.; Hang, C.; Hu, Y.; Liu, S.; Wang, Y.; y Shen, J. (2019). A deep learning based rsa strategy for elastic optical networks. In *2019 18th International Conference on Optical Communications and Networks (ICOCN)* (pp. 1–3). 1

Yuan, Junling; Wu, Zixuan; Li, Xuhong; Zhang, Qikun; y Hao, Xuyang (2024). A rmsca algorithm for space division multiplexing elastic optical networks with core switching. *Optical Fiber Technology*, 88, 103978. 2.1

Yuksel, Seniha Esen; Wilson, Joseph N; y Gader, Paul D (2012). Twenty years of mixture of experts. *IEEE transactions on neural networks and learning systems*, 23(8), 1177–1193. 2.4.1

Zhang, Yuhang; Shah, Nilay; Wen, Zhenkun; Chen, Yuxi; y Xing, Eric (2019). Dac: The double actor-critic architecture for learning options. *arXiv preprint arXiv:1909.01963*. 3.1.1

Zhang, Yongjun; Xin, Jingjie; Li, Xin; y Huang, Shanguo (2020). Overview on routing and resource allocation based machine learning in optical networks. *Optical Fiber Technology*, 60, 102355. 1

A | Obtención de Parámetros de Simulación

El presente anexo describe el proceso seguido para definir los parámetros utilizados en las simulaciones realizadas en esta tesis. La elección adecuada de dichos parámetros fue fundamental para garantizar un entrenamiento estable del agente DRL, así como para obtener resultados comparables y reproducibles en los distintos escenarios evaluados. El procedimiento estuvo compuesto por varias etapas experimentales, cuyo objetivo fue identificar configuraciones robustas tanto desde el punto de vista del rendimiento como de la interpretabilidad del modelo.

A.1. Optimización de Hiperparámetros mediante *Optuna*

En la primera etapa se llevó a cabo un proceso sistemático de optimización de hiperparámetros utilizando la herramienta *Optuna*. Esta plataforma permitió explorar el espacio de configuraciones del agente de manera eficiente mediante técnicas de búsqueda bayesiana y adaptación dinámica de parámetros. Entre los hiperparámetros ajustados se incluyeron la tasa de aprendizaje (learning_rate), el tamaño del *batch* (batch_size), el coeficiente de entropía, entre otros. El criterio principal de evaluación fue el desempeño obtenido durante el entrenamiento, medido en términos de recompensa acumulada y estabilidad de la política. Este proceso permitió identificar un conjunto de hiperparámetros que mejoraban el rendimiento sin comprometer la convergencia del modelo.

A.2. Evaluación del uso de *Action Masking*

Se evaluó también el uso de *action masking* como mecanismo para restringir al agente a seleccionar únicamente acciones válidas en cada estado. Si bien esta técnica aceleró significativamente la velocidad de entrenamiento y redujo la exploración de decisiones inválidas, se observó que, una vez alcanzada cierta madurez en el entrenamiento, el desempeño del agente con y sin *masking* convergía a valores similares. Además, el uso de *action masking* introducía complejidades adicionales en el proceso de aprendizaje por imitación, particularmente en la construcción del conjunto de datos y la replicación precisa de la política del agente. Por estas razones, y con el objetivo de facilitar el análisis interpretativo, se decidió no emplear *action masking* en la versión final del modelo.

A.3. Comparación de Distintas Políticas de Entrenamiento

Con el fin de seleccionar el algoritmo de aprendizaje más adecuado, se llevaron a cabo múltiples experimentos utilizando diversas políticas de entrenamiento disponibles en la librería *Stable-Baselines3*. Entre las políticas consideradas se encuentran PPO, DQN, TRPO, A2C y ACER. Cada política fue evaluada bajo las mismas condiciones de simulación, analizando su capacidad para converger de manera estable, su sensibilidad a los hiperparámetros y su compatibilidad con el entorno de red óptica simulado. Los resultados indicaron que PPO ofrecía el mejor balance entre rendimiento, estabilidad y reproducibilidad, por lo que fue seleccionada como la política principal para los experimentos presentados en este trabajo.

A.4. Diseño y Prueba de Esquemas de Recompensa

Finalmente, se exploraron diferentes heurísticas simples para definir la función de recompensa del agente. Estas incluían combinaciones de recompensas positivas y negativas, tales como: recompensas binarias (+1 para éxito y -1 para fallo), esquemas exclusivamente positivos (por ejemplo, +1 para asignación exitosa), así como escalas discretas intermedias (0, 0.5 y 1). El objetivo de esta fase fue identificar un esquema de recompensas que

incentivara comportamientos coherentes sin introducir sesgos perjudiciales o dificultar la convergencia. A través de pruebas experimentales, se determinó que ciertas configuraciones producían políticas inestables o excesivamente aleatorias, mientras que otras facilitaban la formación de patrones consistentes. El esquema adoptado en la versión final del agente corresponde a aquel que mostró el mejor compromiso entre simplicidad, estabilidad y claridad interpretativa.

Conclusión del Proceso de Selección

En conjunto, este proceso experimental permitió establecer una configuración de simulación óptima y justificable, tanto desde el punto de vista del rendimiento del agente DRL como de los objetivos de interpretabilidad planteados en esta tesis. Las decisiones aquí documentadas aseguran que los resultados presentados se sustentan en un diseño metodológico riguroso y transparente.

B | Topologías de red utilizadas

En este apartado, se detallan las distancias de cada uno de los enlaces de la topología de red NSFNet.

B.1. NSFNet

Nodo origen	Nodo destino	Distancia (km)	Nodo origen	Nodo destino	Distancia (km)
0	1	1130	5	13	1980
0	2	1710	6	7	720
0	7	2840	7	8	700
1	2	700	8	9	840
1	3	960	8	11	370
2	5	2100	8	12	460
3	4	560	10	11	600
3	10	2350	10	12	800
4	5	1480	11	13	460
4	6	740	12	13	250
5	9	1140			