Universidad Nacional Autónoma de México

Maestría en Ciencia e Ingeniería de la Computación

Optimización de estrategias de comercio algorítmico a través de inteligencia artificial: Un análisis comparativo de modelos de aprendizaje profundo, algoritmos evolutivos y aprendizaje por refuerzo.

T E S I S

que para optar por el grado de

Maestro en Ciencia e Ingeniería de la Computación

PRESENTA:
Act. Ricardo Daniel Fernández Noguez

Tutor Principal:
Dr. Erik Molino Minero Re, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas.

Ciudad Universitaria, CD. MX. (Noviembre) 2023

Universidad Nacional Autónoma de México
UNAM
Biblioteca Central
Dirección General de Bibliotecas de la UNAM

UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

10 de noviembre de 2023

PROTESTA UNIVERSITARIA DE INTEGRIDAD Y HONESTIDAD ACADÉMICA Y PROFESIONAL

De conformidad con lo dispuesto en los artículos 87, fracción V, del Estatuto General, 68, primer párrafo, del Reglamento General de Estudios Universitarios y 26, fracción I, y 35 del Reglamento General de Exámenes, me comprometo en todo tiempo a honrar a la institución y a cumplir con los principios establecidos en el Código de Ética de la Universidad Nacional Autónoma de México, especialmente con los de integridad y honestidad académica.

De acuerdo con lo anterior, manifiesto que el trabajo escrito titulado Optimización de estrategias de comercio algorítmico a través de inteligencia artificial: Un análisis comparativo de modelos de aprendizaje profundo, algoritmos evolutivos y aprendizaje por refuerzo, que presenté para obtener el grado de Maestro en Ciencia e Ingeniería de la Computación, es original, de mi autoría y lo realicé con el rigor metodológico exigido por mi Programa de Posgrado, citando las fuentes, ideas, textos, imágenes, gráficos u otro tipo de obras empleadas para su desarrollo.

En consecuencia acepto que la falta de cumplimiento de las disposiciones reglamentarias y normativas de la Universidad, en particular las ya referidas en el Código de Ética, llevará a la nulidad de los actos de carácter académico administrativo del proceso de titulación/graduación

Atentamente

Ricardo Daniel Fernández Noguez, 312012975

Agradecimientos

A mis padres, cuyo apoyo incondicional, paciencia y enseñanzas han sido una constante a lo largo de mi vida, forjando los cimientos sobre los cuales he construido mis sueños y aspiraciones.

A mis hermanos, pilares de confianza y guía, siempre presentes para mostrarme el camino con su ejemplo y consejos.

A mis amigos, por ser la fuente de ánimo que me impulsa a seguir adelante, mostrándome el valor incalculable del trabajo en equipo y la perseverancia.

A mi lugar de empleo, por ofrecerme no solo una plataforma para aplicar mis conocimientos sino también la flexibilidad y el soporte necesarios para culminar este proyecto.

A mis profesores, en especial a mi tutor y sinodales, cuya dedicación, tiempo y sabiduría fueron fundamentales para que pudiera aprender y crecer a partir de mis errores.

A mi novia, compañera de vida, por estar a mi lado desde el comienzo hasta el final de esta travesía, inspirándome en cada paso con su amor y fortaleza.

Asimismo, deseo expresar mi sincera gratitud al Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT) por haberme otorgado una beca durante una etapa crucial de mis estudios. Esta beca no solo alivió mis preocupaciones financieras, sino que también me brindó la tranquilidad y la libertad para dedicarme plenamente a mi formación académica y profesional. Su confianza y apoyo han sido fundamentales en mi camino hacia la consecución de mis metas educativas y profesionales.

Finalmente, a la Universidad, por proporcionarme las herramientas, conocimientos y un entorno ético, moral y profesional que me ha formado como actuario y ahora como maestro, preparándome para contribuir a la sociedad con lo mejor de mis capacidades.

Resumen

En este proyecto se aborda el problema del comercio algorítmico que consiste en determinar los puntos de compra y venta de un activo financiero mediante modelos de inteligencia artificial como aprendizaje profundo, algoritmos evolutivos y aprendizaje por refuerzo analizando dos enfoques. El primer enfoque propone un preprocesamiento y etiquetamiento de los momentos de compra/-venta de la información, posteriormente se entrenó y evaluó la red neuronal y dos estrategias evolutivas, una de ellas con una modificación propuesta explicada más adelante. El segundo enfoque se implementó el marco de trabajo de *Financial Reinforcement Learning* (FinRL) para tres algoritmos con un enfoque multi-objetivo posteriormente se entrenaron y evalúaron dichos modelos.

Los algoritmos implementados fueron puestos a prueba en 8 portafolios que se configuraron y se comparó por medio del rendimiento acumulado. Siendo el segundo enfoque (algoritmos de FinRL) que tuvo un mayor rendimiento en la mayoría de los experimentos.

This project addresses the issue of algorithmic trading, which involves determining the buy and sell points of a financial asset using artificial intelligence models such as deep learning, evolutionary algorithms, and reinforcement learning by analyzing two approaches.

The first approach proposes preprocessing and labeling of the buy/sell moments of the information, followed by training and evaluating the neural network and two evolutionary strategies, one of which with a proposed modification explained later.

The second approach implemented the Financial Reinforcement Learning (FinRL) framework for three algorithms with a multi-objective focus; these models were subsequently trained and evaluated.

The implemented algorithms were tested on 8 configured portfolios and compared through cumulative performance. The second approach (Financial Reinforcement Learning (FinRL) algorithms) had the best performance in most of the experiments.

Tabla de contenido

Resumen **5**

Índice de Figuras **8**

Índice de Tablas **9**

1 Introducción **13**
1.1 Planteamiento del problema 14
1.2 Antecedentes 14
1.3 Objetivo general 17
1.4 Hipótesis 18
1.5 Contribución y relevancia 18
1.6 Estructura del documento 19

2 Marco teórico **21**
2.1 Introducción 21
2.2 Comercio algorítmico 22
2.3 Aprendizaje supervisado 25
2.4 Algoritmos evolutivos 27
2.5 Aprendizaje profundo por refuerzo 28
2.5.1 Advantage Actor Critic (A2C) 29
2.5.2 Optimización de política proximal (PPO) 30
2.5.3 Gradiente de política determinista profundos (Deep Deterministic Policy Gradients DDPG) 30
2.6 Optimización multi-objetivo 31

3 Metodología **33**
3.1 Introducción 33
3.2 Datos 34
3.3 Preprocesamiento 37
3.4 Etiquetamiento automático 45
3.5 Modelos 49
3.5.1 Selección de Variables 49
3.5.2 Red neuronal 53
3.5.3 Estrategia evolutiva 54
3.5.4 Estrategia evolutiva modificada 60

3.6 Financial Reinforcement Learning (FinRL) 62
3.6.1 Obtención de Datos y Preprocesamiento 63
3.6.2 Entrenamiento 64

4 Resultados **67**
4.1 Red neuronal 67
4.2 Estrategia evolutiva (ES) 69
4.3 Estrategia evolutiva modificada (ESM) 70
4.4 FinRL 71
4.4.1 A2C 71
4.4.2 PPO 73
4.4.3 DDPG 75
4.4.4 Comparación A2C, PPO, DDPG 76
4.5 Comparación General 79

5 Discusión **81**

6 Conclusión y trabajo futuro **85**

A Anexo A **89**

Bibliografía **91**

Índice de Figuras

2.1 Ejemplo red neuronal. 26

3.1 Diagrama Enfoque 1. 34
3.2 Diagrama Enfoque 2. 34
3.3 Ejemplo de gráfico de velas y volumen. 36
3.4 Ejemplo de la señal e intensidad generada por el algoritmo 1. 39
3.5 Ejemplo de la señal y variable objetivo generada por el algoritmo 2. 41
3.6 Ejemplo de la señal y variable ojetivo generada por el algoritmo 3. 43
3.7 Ejemplo de la señal y variable objetivo generada por el algoritmo 4. 45
3.8 Ejemplo de la variable objetivo resultante de unificar las respectivas de MACD, RSI y SMI. 46
3.9 Ejemplo de la volatilidad calculada en el algoritmo 4. 47
3.10 Ejemplo de la matriz de correlación. 50
3.11 Variables con mayor índice de información mutua. 52
3.12 Gráfico de convergencia iteraciones vs rendimiento de la estrategia evolutiva 59
3.13 Gráfico de convergencia, iteraciones vs recompensa y rendimiento de estrategia evolutiva modificada. 61
3.14 Frontera con el mayor hipervolumen encontrado. 66

4.1 Fronteras de mayor hipervolumen para cada uno de los portafolios con A2C. 72
4.2 Fronteras de mayor hipervolumen para cada uno de los portafolios con PPO. 74
4.3 Fronteras de mayor hipervolumen para cada uno de los portafolios con DDPG. 76
4.4 Comparación del retorno acumulado alcanzado por los algoritmos de A2C, PPO y DDPG en cada uno de los portafolios. 78
4.5 Comparación de algoritmos por retorno acumulado. 80

5.1 Comparación por retorno acumulado de las estrategias evolutivas implementadas. 82
5.2 Comparación por retorno acumulado de los métodos de aprendizaje por refuerzo implementados en el marco de trabajo de FinRL. 83

Índice de Tablas

3.1 Ejemplo de extracción con *yfinance*. 35
3.2 Ejemplo de Tabla con Indicadores calculados con *pandas_ta* [29]. 38
3.3 Variables añadidas después de las Tabla con indicadores calculados con *pandas_ta*. 47
3.4 Variables con correlación de Pearson menor a 0.6 en valor absoluto. 51
3.5 Variables con información mutua superior a 0.04. 52
3.6 Arquitectura de la red neuronal aplicada. 53
3.7 Resultados del modelo entrenado en el conjunto de prueba. 54
3.8 Resultados de la evaluación retrospectiva. 54
3.9 Arquitectura de la red neuronal aplicada en la estrategia evolutiva. 55
3.10 Resultados del modelo entrenado en el conjunto de prueba. 59
3.11 Resultados de la evaluación retrospectiva en el conjunto de prueba 60
3.12 Resultados del modelo entrenado en el conjunto de prueba. 61
3.13 Resultados de la evaluación retrospectiva. 62
3.14 Resultados de la evaluación retrospectiva A2C. 65
3.15 Hipervolumen generado por las fronteras encontradas en iteración de NSGA-II. 66

4.1 Resultados con respecto a la variable objetivo en el conjunto de prueba Red neuronal. 68
4.2 Resultados de la evaluación retrospectiva para el modelo de red neuronal. . 68
4.3 Resultados con respecto a la variable objetivo en el conjunto de prueba para estrategia evolutiva. 69
4.4 Resultados de la evaluación retrospectiva para el modelo de estrategia evolutiva. 70
4.5 Resultados con respecto a la variable objetivo en el conjunto de prueba para estrategia evolutiva modificada. 70
4.6 Resultados de la evaluación retrospectiva para el modelo de estrategia evolutiva modificada. 71
4.7 Resumen del hipervolumen generado en cada portafolio con A2C. 71
4.8 Resultado de la frontera con mayor hipervolumen y mayor retorno acumulado obtenido para A2C. 72
4.9 Resumen del hipervolumen generado en cada portafolio con PPO. 73
4.10 Resultado de la frontera con mayor hipervolumen y mayor retorno acumulado obtenido para PPO. 74
4.11 Resumen del hipervolumen generado en cada portafolio con DDPG. 75

4.12 Resultado de la frontera con mayor hipervolumen y mayor retorno acumulado obtenido para DDPG. 75
4.13 Comparación de los resultados de A2C, PPO y DDPG. 77
4.14 Comparación del porcentaje del retorno acumulado de los algoritmos implementados. 79

6.1 Algoritmo vencedor en cada uno de los portafolios. 86

A.1 Tabla de portafolios y activos 89
A.2 Lista de portafolios y sus activos correspondientes utilizados en la experimentación. 89

Índice de algoritmos

1 Algoritmo para generar la Señal e Intensidad con EMA o DEMA. 39
2 Algoritmo para generar la Señal y Objetivo con RSI. 40
3 Algoritmo para generar la Señal y Objetivo con MACD. 42
4 Algoritmo para generar la señal y objetivo con SMI. 44
5 Algoritmo para calcular la volatilidad normalizada/estandarizada en n períodos. 48
6 Algoritmo para entrenar los pesos del modelo mediante una estrategia evolutiva. 57
7 Algoritmo para obtener la recompensa dado un modelo específico. 58

Capítulo 1

Introducción

En finanzas el problema de compra y venta de activos financieros resulta altamente complejo debido a los componentes que conforman e impactan al mercado financiero. En la literatura, se ha abordado esta problemática bajo distintos enfoques dentro del ámbito de la inteligencia artificial, donde principalmente en función del precio de cierre, se miden tendencias, indicadores financieros e indicadores estocásticos, correlaciones entre índices u otros precios de activos, así como noticias buscando predecir los momentos en los cuales un activo se encuentra en sus valores óptimos de compra y venta. Los resultados en los distintos artículos consultados dentro de esta investigación han arrojado pruebas de que los modelos propios de la inteligencia artificial son lo suficientemente robustos para abordar eficientemente el problema.

Este proyecto se centra en explorar y evaluar diferentes modelos y técnicas de inteligencia artificial aplicadas al comercio algorítmico, con el objetivo de identificar los puntos óptimos de transacción.

Para abordar este reto, se adoptó dos enfoques, cada uno se analizó y se comparó con metodologías distintas. El primer enfoque se basa en el preprocesamiento y etiquetado de los datos financieros, seguido por la implementación y evaluación de una red neuronal y dos estrategias evolutivas, incluyendo una modificación específica detallada más adelante. El segundo enfoque se implementó el marco de trabajo *Financial Reinforcement Learning* (FinRL) [21], entrenándo y evaluando a tres algoritmos diferentes con un ajuste de hiperparámetros con perspectiva multi-objetivo.

El desempeño de los algoritmos implementados fue rigurosamente probado en ocho portafolios financieros distintos, configurados específicamente para este estudio. Los resultados obtenidos permitieron realizar comparativas basadas en el rendimiento acumulado de cada estrategia, destacando a los algoritmos implementados en el marco de trabajo de FinRL como los de mejor desempeño en la mayoría de los casos. Esta introducción sienta las bases para una exploración detallada de las metodologías aplicadas y los resultados obtenidos, con el fin de contribuir al avance del comercio algorítmico mediante metodologías propias de la inteligencia artificial.

1.1. Planteamiento del problema

Existen métodos y alternativas de modelos de inteligencia artificial como aprendizaje profundo, algoritmos evolutivos y aprendizaje por refuerzo para enfrentar el problema del comercio algorítmico. El problema del comercio algoritmico consiste en determinar los puntos de compra y venta óptimos de un activo financiero. Debido a que no es un problema trivial determinar cual es la mejor estrategia de compra venta de activos financieros, realizamos una comparativa de diferentes enfoques para encontrar el mayor rendimiento acumulado en un conjunto de portafolios de inversión que se ha definido para fines de la experimentación en este proyecto. Es por ello, que esta investigación se enfoca en implementar distintos algoritmos relevantes en estas áreas ya mencionadas. Cabe mencionar que estos proyectos construyen sus conjuntos de datos en función del precio de cierre se miden tendencias, indicadores financieros e indicadores estocásticos, correlaciones entre índices u otros precios de activos, así como noticias que buscan predecir los momentos en los cuales un activo se encuentra en sus valores óptimos de compra y venta.

Asimismo, cabe mencionar que la mayoría de los avances significativos en esta área no están disponibles de forma pública, a causa de la enorme cantidad de dinero en juego, es muy poco probable que las empresas *FinTech* privadas hagan públicos los últimos resultados de sus investigaciones [11].

1.2. Antecedentes

Los algoritmos de comercio algorítmico buscan encontrar las condiciones óptimas de compra y venta para poder obtener el máximo rendimiento sobre los activos financieros que operen. Antes de que se popularizara el uso de técnicas de inteligencia artificial para el problema del comercio algorítmico, este principalmente se fundamentaba en reglas duras basadas en distintos indicadores financieros que describen la tendencia y el comportamiento del precio del activo. Estos indicadores debían encontrarse bajo ciertas condiciones como cruzar umbrales específicos, cruces entre líneas de tendencias o cruces entre los indicadores mismos [1]. Sin embargo, ninguna regla era perfecta y dada la complejidad del problema derivado de los múltiples factores que pueden alterar el precio de un activo, un porcentaje significativo de las señales de compra y venta generadas eran falsas. Por esta razón, los métodos de inteligencia artificial abren diferentes caminos para enfrentar este problema, sin embargo, necesitan grandes volúmenes de información para poder determinar señales de compra y venta confiables.

En la literatura, se encuentran distintas formas de abordar el problema de comercio algorítmico, lo que cabe resaltar de los ejemplos que se describirán a continuación es que los trabajos más recientes y con mejores resultados utilizan algún método de aprendizaje profundo por refuerzo o aprendizaje de máquina.

- **Bosque aleatorio**: En 2014, Ash Booth , Enrico Gerding y Frank McGroarty [13] pro-

ponen un sistema de negociación automatizado basado en conjuntos ponderados por rendimiento de bosques aleatorios que mejora la rentabilidad y la estabilidad de la negociación de eventos estacionales. Se realiza un análisis de varias técnicas de regresión, así como una exploración de los méritos de varias técnicas para la ponderación experta. El rendimiento de los modelos se analiza utilizando una gran muestra de acciones del Deutscher Aktienindex (DAX). Los resultados expuestos en este artículo muestran que los conjuntos ponderados de actualidad de bosques aleatorios producen resultados superiores en términos de rentabilidad y precisión de predicción en comparación con otras técnicas de conjunto.

- **Máquina de soporte vectorial**: En 2018, Yingjun Chen y Yijie Hao [14] proponen un método que integra el análisis de componentes principales (PCA) en una máquina de vectores de soporte ponderada (WSVM) para pronosticar los puntos de negociación de la acción (PCA-WSVM). En primer lugar, se modela la predicción de las señales de negociación de acciones como un problema de clasificación ponderado de cuatro clases. Luego, se aplica PCA para limpiar el conjunto de datos original y reorganizarlo en una nueva estructura de datos. En tercer lugar, WSVM se utiliza con el conjunto de datos transformados para pronosticar los puntos de inflexión de la acción. Finalmente, se llevó a cabo una serie de experimentos entre PCA-WSVM, WSVM, PCA-ANN (PCA- Redes Neuronales Artificiales) y la estrategia Buy-and-Hold en acciones de dos mercados bursátiles chinos de Shanghai y Shenzhen, para probar el rendimiento de este modelo modelo establecido. Los resultados del experimento reflejan que con este modelo propuesto, la capacidad de predicción y la rentabilidad con diferentes estrategias de inversión son las mejores, lo que indica que PCA-WSVM es efectivo y se puede aplicar para pronosticar las señales de negociación de acciones en la aplicación del mundo real.

- **Algoritmos genéticos y redes neuronales**: En 2013, Cain Evans, Konstantinos Pappas y Fatos Xhafa [16] presentan un modelo de predicción y toma de decisiones basado en redes neuronales artificiales (ANN) y en algoritmos genéticos. El conjunto de datos utilizado para esta investigación comprende 70 semanas de tipos de cambio anteriores de los 3 pares de divisas más negociados: GBP/USD, EUR/GBP y EUR/USD. Las pruebas estadísticas iniciales confirmaron con una significación de más del 95 % que las series temporales de tipos de cambio diarios FOREX no se distribuyen aleatoriamente. Otro resultado importante es el modelo propuesto quelogró una precisión de predicción del 72,5 %. Además, al implementar la estrategia comercial óptima, este modelo produjo un rendimiento neto anualizado del 23,3 %.

- **Redes neuronales convolucionales (CNN)**: En este estudio realizado en 2018 por Omer Berat Sezer y Ahmet Murat Ozbayoglu [17] propone un nuevo modelo de comercio algorítmico CNN-TA utilizando una red neuronal convolucional 2-D basada en propiedades de procesamiento de imágenes. Para convertir series de tiempo financieras en imágenes 2-D, se utilizan 15 indicadores técnicos diferentes, cada uno con

diferentes selecciones de parámetros. Cada instancia de indicador genera datos para un período de 15 días. Como resultado, se construyen imágenes 2-D de tamaño 15×15. Luego, cada imagen se etiqueta como comprar, vender o mantener, según las colinas y valles de la serie temporal original. Los resultados indican que, en comparación con la estrategia *Buy & Hold* y otros sistemas comerciales comunes durante un largo período fuera de la muestra, el modelo entrenado brinda mejores resultados para acciones y ETF.

- **Series de tiempo y aprendizaje profundo por refuerzo (Time-driven feature-aware and deep reinforment learning TFJ-DRL)**: Este modelo no solo aborda el problema de comercio algorítmico sino que para la toma de decisiones hace que el agoritmo explore el entorno (series de tiempo altamente no estacionarias y ruidosas) de forma online sin ninguna información supervisada.

  Entonces la solución propuesta por Kai Lei, Bing Zhang, Yu Li, Min Yang y Ying Shen [18]; es un modelo consciente del tiempo que mediante aprendizaje profundo y aprendizaje por refuerzo, mejora el aprendizaje de representación de señales financieras y la toma de decisiones en el comercio algorítmico. Concretamente, el algoritmo aprende la representación ambiental seleccionando y volviendo a ponderar de manera adaptativa varias características de las señales financieras y resumiendo los valores de atención entre la información histórica y la tendencia cambiante según el estado actual. Además, el aprendizaje profundo supervisado y el aprendizaje por refuerzo se entrenan de forma conjunta e iterativa para hacer un uso completo de las señales supervisadas en los datos de entrenamiento y obtener más información actualizada y restricciones de función de pérdida más estrictas, lo que aumenta los rendimientos de la inversión. TFJ-DRL se evalúa en datos financieros del mundo real con diferentes tendencias de precios (al alza, a la baja y sin dirección obvia). Una serie de análisis muestran la robusta superioridad y la amplia aplicabilidad del método propuesto.

- **Aprendizaje profundo por refuerzo (DRL, DQN y Q-learning agent)**: En 2021 Cong Ma, Jiangshe Zhang, Junmin Liu, Lizhen Ji y Fei Gao [12] proponen un modelo novedoso llamado algoritmo de aprendizaje profundo por refuerzo de módulos múltiples paralelos (PMMRL). Aquí, se usan dos módulos paralelos para extraer y codificar la función: un módulo que emplea capas totalmente conectadas (FC) se usa para conocer el estado actual de los datos de mercado de las acciones negociadas y los datos fundamentales de la empresa emisora; otro módulo que utiliza capas de memoria a corto plazo (LSTM) tiene como objetivo detectar la tendencia histórica a largo plazo del mercado. El modelo propuesto puede extraer características de todo el entorno mediante los dos módulos anteriores simultáneamente, aprovechando las ventajas de las capas LSTM y FC. Extensos experimentos en el mercado de valores de China ilustran que el algoritmo PMMRL propuesto logra una mayor ganancia y una reducción menor que varios algoritmos de última generación.

  En otro artículo de investigación, Thibaut Théate y Damien Ernst [11] proponen una

nueva política comercial de DRL para maximizar el indicador de rendimiento resultante de la relación de Sharpe en una amplia gama de mercados bursátiles. Denominado algoritmo *Trading Deep Q-Network* (TDQN), este nuevo enfoque DRL está inspirado en el popular algoritmo DQN [19] y se adapta significativamente al problema comercio algorítmico específico en cuestión. El entrenamiento del agente de aprendizaje por refuerzo (RL) resultante se basa completamente en la generación de trayectorias artificiales a partir de un conjunto limitado de datos históricos del mercado de valores. Con el fin de evaluar objetivamente el rendimiento de las estrategias comerciales. Asimismo, se propone una metodología de evaluación del rendimiento novedosa y más rigurosa. Siguiendo este nuevo enfoque de evaluación del rendimiento, se informan resultados prometedores para el algoritmo TDQN.

Partiendo de la misma estrategia comercial que no es rentable para todas las acciones todo el tiempo, la selección de una estrategia de negociación para la acción en un instante de tiempo determinado es el principal problema de investigación en la negociación del mercado de valores. Una estrategia comercial dinámica óptima generada a partir del patrón actual de la tendencia del precio de las acciones puede intentar resolver este problema. El aprendizaje por refuerzo puede encontrar esta estrategia comercial dinámica óptima interactuando con el mercado de valores real como su entorno. La representación del estado del medio ambiente es crucial para el desempeño. Para ello, en 2021, el equipo conformado por Jagdish Bhagwan Chakole, Mugdha S. Kolhe, Grishma D. Mahapurush, Anushka Yadav y Manish P. Kurhekar [20] entrenó un agente comercial utilizando el algoritmo *Q-learning* de aprendizaje por refuerzo para encontrar estrategias comerciales dinámicas óptimas. Experimentaron con los dos modelos propuestos en datos bursátiles reales de los mercados bursátiles indio y estadounidense. Los modelos propuestos superaron la estrategia comercial basada en *Buy-and-Hold y Decision-Tree* en términos de rentabilidad.

1.3. Objetivo general

Comparar el comportamiento de distintos algoritmos de aprendizaje profundo por refuerzo, aprendizaje supervisado y algoritmos evolutivos en un problema complejo como lo es el comercio algorítmico.

El indicador de desempeño con la que se evaluarán y compararán cada uno de los algoritmos será el rendimiento generado.

Objetivos específicos

Como se ha expuesto existen diferentes estrategias para atacar el problema de comercio algorítmico aplicando modelos de inteligencia artificial, los objetivos específicos de este proyecto son:

- Explorar en la literatura las distintas propuestas de algoritmos que enfrentan el problema de comercio algorítmico.

- Exponer una propuesta para etiquetar de forma automática un grupo de puntos de compra y venta altamente rentables.

- Entrenar y evaluar distintos algoritmos con el fin de encontrar alguno que supere al menos a la tasa libre de riesgo actual 11 % (México junio de 2023).

- Proponer una modificación a algún algoritmo, anteriormente estudiado.

- Combinar un enfoque multi-objetivo con alguno de estos algoritmos implementados para ajustar hiperparámetros y conservar un grupo de hiperparámetros que formen una frontera de Pareto.

- Obtener rendimientos superiores a la inflación en los conjuntos de prueba.

1.4. Hipótesis

Se postula que, mediante la comparación y evaluación exhaustiva de diferentes algoritmos de aprendizaje profundo por refuerzo, aprendizaje supervisado y algoritmos evolutivos aplicados al comercio algorítmico, es posible identificar un algoritmo o una combinación de estos que no solo superen la tasa libre de riesgo actual del 11 % en México (junio de 2023), sino que también generen rendimientos superiores a la inflación en los conjuntos de prueba. Además, se espera que la implementación de un enfoque multi-objetivo para ajustar hiperparámetros contribuya a optimizar el rendimiento de algoritmos de aprendizaje por refuerzo, manteniendo un conjunto de hiperparámetros que formen una frontera de Pareto, y así maximizar la rentabilidad mientras se minimizan los riesgos asociados.

1.5. Contribución y relevancia

En este proyecto se consideran tres contribuciones de caracter relevante las cuales se explican a continuación:

- Algoritmo para etiquetamiento de compra/venta/mantener: en el Capítulo 3 de Metodología se describe una forma en la que al combinarse tres indicadores financieros y definiendo un umbral de rendimiento, puede existir un etiquetamiento de forma automatizada donde se etiqueten diferentes puntos de compra/venta que brindan un rendimiento alto. Regularmente, en los artículos relacionados a estos proyectos no se aborda a profundidad. Esta contribución es particularmente relevante ya que, a diferencia de muchos trabajos previos en la literatura, se ofrece un enfoque detallado y práctico para la identificación de estos puntos críticos, proporcionando así, una herramienta valiosa para los practicantes e investigadores del área.

- Modificación a la estrategia evolutiva: se propone una nueva forma de recompensar al agente basada en el rendimiento y el etiquetamiento anteriormente propuesto. Este enfoque dual permite una optimización más holística y alineada con los objetivos financieros del comercio algorítmico, y es una alternartiva competitiva con la estrategia evolutiva original.

- *Financial Reinforcement Learning (FinRL) + Non-dominated Sorting Genetic Algorithm (NSGA-II)* [52] para ajuste de hiperparámetros: Los resultados alcanzados con FinRL permiten una gama de opciones para diferentes apetitos de riesgo, las cuales forman una frontera de Pareto. Este enfoque multiobjetivo es relevante ya que proporciona una variedad de opciones estratégicas que pueden ser seleccionadas de acuerdo a diferentes perfiles y apetitos de riesgo, aportando flexibilidad y personalización a las decisiones de inversión.

1.6. Estructura del documento

La presente sección se dedicará a explciar una breve descripción sobre los componentes principales de cada uno de los capítulos que componen este proyecto.

En el Capítulo 2 es el detalle de la teoría básica que fundamenta la metodología implementada así como los modelos aplicados, con el fin de que este proyecto se encuentre autocontenido, en la medida de lo posible.

En el Capítulo 3 se profundizan los pasos que se siguen durante el desarrollo del proyecto, el proceso de etiquetado automático, los detalles de la estrategia evolutiva, así como de la red neuronal entrenada. Para dar paso a las bondades de FinRL que permite el entrenamiento de algoritmos de aprendizaje por refuerzo profundo en un ambiente de Open IA específico para comercio y la descripción el ajuste de hiperparámetros a un enfoque de optimización multi objetivo por medio de NSGA-II.

Después de especificar las condiciones que rigen a cada método, el Capítulo 4 exhibe las características de los experimentos realizados así como los resultados logrados para cada uno de los algoritmos implementados.

Lo anterior, para que en el Capítulo 5 se puede mostrar la discusión y limitaciones del proyecto para dar paso a las conclusiones y el trabajo futuro en el último (Capítulo 6).

Capítulo 2

Marco teórico

Este capítulo está integrado por el contenido básico de comercio algorítmico e inteligencia artificial enfocado particularmente en las definiciones y propiedades de los temas relacionados de este proyecto, los cuales, son: aprendizaje por supervisado, algoritmos evolutivos, aprendizaje profundo por refuerzo y optimización multi-objetivo.

2.1. Introducción

En el área de la inteligencia artificial, diferentes enfoques y métodos han sido desarrollados para abordar problemas complejos sobre optimización y toma de decisiones. Entre estos métodos habrá un mayor enfoque al aprendizaje profundo, los algoritmos evolutivos y el aprendizaje por refuerzo.

El aprendizaje profundo es un rubro de la inteligencia artificial que se basa en redes neuronales artificiales con múltiples capas, capaces de "aprender" y representar características y relaciones en los datos. A través del aprendizaje profundo, los modelos pueden analizar grandes volúmenes de información para reconocer patrones y estructuras relevantes, permitiendo el desarrollo de sistemas inteligentes capaces de tomar decisiones y realizar tareas complejas en diversas áreas, como el comercio algorítmico.

Por otro lado, los algoritmos evolutivos se inspiran en los procesos de evolución biológica y la selección natural. Estos métodos generan y mejoran soluciones de forma iterativa, por medio de la implementación de operadores como la cruza, mutación y selección. En el ámbito del comercio algorítmico, los algoritmos evolutivos pueden utilizarse para optimizar estrategias de inversión, encontrar configuraciones óptimas de cartera y adaptarse a condiciones cambiantes del mercado.

El aprendizaje por refuerzo es una técnica de aprendizaje automático que se especializa en la toma de decisiones (acciones) de forma secuencial en un entorno donde interactua con el ambiente. En el aprendizaje por refuerzo, un agente aprende a través de la interacción con el ambiente, recibiendo recompensas o penalizaciones (recompensas negativas) con base en sus acciones. En el contexto del comercio algorítmico, el aprendizaje por refuerzo ha sido

utilizado para desarrollar agentes capaces de optimizar estrategias de inversión, gestionar el riesgo y adaptarse a condiciones del mercado en tiempo real.

Por otro lado, la optimización multiobjetivo encuentra soluciones que equilibran y optimizan múltiples objetivos simultáneamente. En el comercio algorítmico, a menudo existen diferentes metas a tener en cuenta, como maximizar las ganancias, minimizar el riesgo y reducir los costos.

La optimización multiobjetivo permite explorar y obtener soluciones que representen compromisos y compensaciones entre estos objetivos, proporcionando a los inversores un conjunto de soluciones eficientes y no dominadas.

En resumen, este marco teórico busca contener la información relevante para este proyecto sobre el aprendizaje profundo, los algoritmos evolutivos, el aprendizaje por refuerzo y la optimización multiobjetivo los cuales son herramientas y enfoques para abordar problemas complejos en el comercio algorítmico debido a que estas técnicas ofrecen la capacidad de tomar decisiones, adaptarse a condiciones cambiantes y encontrar soluciones eficientes que consideren múltiples objetivos y restricciones.

2.2. Comercio algorítmico

El comercio algorítmico se refiere a programas de cómputo que permiten ejecutar algoritmos para automatizar algunos o todos los componentes de una estrategia de comercio. Los algoritmos fungen como una secuencia de pasos o reglas diseñadas para alcanzar un objetivo basado principalmente en el rendimiento, buscando facilitar la optimización a lo largo del proceso de inversión, desde la generación de un portafolio hasta la colocación de activos, ejecución de transacciones comerciales y administración de riesgos [2].

Muchas de las estrategias tradicionales en el comercio algorítmico están basadas en indicadores que buscan medir la tendencia, volatilidad, momentos de sobrecompra y sobreventa de ciertos activos, algunos de estos indicadores los describimos a continuación:

- **Índice de fuerza relativa (RSI)** por sus siglas en inglés. Oscila en una escala de cero a 100. El indicador fue desarrollado por J. Welles Wilder Jr. e introducido en su libro seminal de 1978, *New Concepts in Technical Trading Systems* [1].

  El RSI puede hacer más que señalar valores sobrecomprados y sobrevendidos. También puede indicar valores que pueden estar preparados para un cambio de tendencia o un retroceso correctivo en el precio. Puede indicar cuándo comprar y vender. Tradicionalmente, una lectura RSI de 70 o superior indica una situación de sobrecompra. Una lectura de 30 o menos indica una condición de sobreventa [3].

$$RSI_{t_1} = 100 - \frac{100}{1 + \frac{PG}{PP}}. \tag{2.1}$$

Donde PG es el promedio de variaciones a la alza y PP es el promedio de variaciones a la baja.

- **Media móvil exponencial (EMA)** es un tipo de media móvil que da un mayor peso a la información más reciente. Esta reacciona a los cambios significativos de precios más que una media móvil simple, que aplica un peso igual a las observaciones del periodo. Se utiliza para producir señales de compra y venta basadas en la media histórica.

$$EMA_{\text{t}} = (Valor_{\text{t}} \times k) + (EMA_{\text{t-1}} \times (1-k)) \tag{2.2}$$

donde:

  - $Valor_{\text{t}}$ es el precio de cierre del día t.
  - k es el factor de suavizado que se calcula como $k = \frac{2}{(N+1)}$, donde N es el número de periodos de la EMA.
  - $EMA_{\text{t-1}}$ es el valor de la EMA_t del día anterior.

El valor de k determina qué tanto peso se da a los precios más recientes. El valor de la EMA para el primer período se calcula generalmente usando una Media Móvil Simple (SMA) [5].

- **Convergencia o divergencia de la media móvil (MACD)** es un indicador de impulso que sigue una tendencia. Esta muestra la relación entre dos medias móviles del precio de un valor. El MACD se calcula restando la media móvil exponencial (EMA) de 26 periodos de la EMA de 12 periodos. En la Ecuación 2.3 se muestra la fórmula de MACD:

$$MACD = EMA_{12} - EMA_{26}. \tag{2.3}$$

La línea de señal es una EMA de nueve periodos de la línea MACD.

Este se utiliza para identificar tendencias a la alza o a la baja, así como condiciones de sobrecompra y sobreventa. Cuando la línea MACD cruza la línea de señal por encima, muestra que es una señal que la tendencia va a la alta. En caso contrario, cuando el MACD cruza por debajo, es una señal que va a la baja. Otro uso que se le da al MACD es para identificar divergencias, es decir, un cambio en las tendencias [5].

- **Índice de impulso estocástico (SMI)** es una herramienta para indicar los momentos en los que están los precios de las acciones. Es decir, indican la tendencia general. Este índice regularmente oscila entre valores $+100$ y -100. Y cuando el precio de cierre es más alto que la mediana o el valor del punto medio del rango alto o bajo, entonces el valor es mayor a 40, y es una tendencia a la alza. En caso contrario, los valores negativos a -40 muestran una tendencia a la baja [4].

- **Media móvil exponencial doble (DEMA)** es un indicador técnico ideado para reducir el retraso en los resultados producidos por una media móvil tradicional (SMA), la cual, es el promedio simple de los últimos n períodos al tiempo t. Se utiliza para disminuir

la cantidad de ruido que tienen los movimientos de precios. Así como para indicar la tendencia en el precio de una acción [5].

$$DEMA = 2 \cdot EMA - EMA(EMA). \tag{2.4}$$

- **Tasa de cambio (ROC)** es un indicador el cual mide el cambio porcentual en el precio entre el precio actual y el precio en un cierto número de periodos. A continuación la fórmula para calcular el ROC:

$$ROC = \left(\frac{\text{Precio de cierre}_p - \text{Precio de cierre}_{p-n}}{\text{Precio de cierre}_{p-n}} \right) \cdot 100. \tag{2.5}$$

  También se puede utilizar como un indicador de divergencia que señala un posible cambio de tendencia próximo [6].

- **Bandas de Bollinger (BBANDS)** se utiliza para trazar líneas de tendencia a dos desviaciones estándar del precio promedio móvil de un instrumento financiero. Así como demuestran los cambios en la volatilidad de un instrumento financiero [7].

$$\begin{aligned} Banda_{media} &= SMA_{20}, \\ Banda_{inferior} &= SMA_{20} - (std(P)_{20} \cdot 2), \\ Banda_{superior} &= SMA_{20} + (std(P)_{20} \cdot 2). \end{aligned} \tag{2.6}$$

- **Índice de fuerza real (TSI)** se utiliza principalmente para identificar tendencias e inversiones. Cuando el indicador diverge con el precio, la TSI puede señalar que la tendencia del precio está debilitando y puede invertirse.

$$\text{TSI} = 100 \times \frac{\text{EMA}(\text{EMA}(CambioPrecio, S), L)}{\text{EMA}(\text{EMA}(|CambioPrecio|, S), L)} \tag{2.7}$$

  donde:

  - EMA representa la media móvil exponencial.
  - *CambioPrecio* es la diferencia de precios de cierre de un período al siguiente.
  - S es el período corto para la primera EMA.
  - L es el período largo para la segunda EMA.
  - $|CambioPrecio|$ es el valor absoluto de los cambios de precios, lo que refleja la magnitud de los movimientos del precio sin considerar la dirección.

  Este indicador se basa en los movimientos de los precios, los niveles de sobrecompra y sobreventa y varían según el activo que se negocie. Es importante que es un indicador que se parece mucho al MACD, sin embargo, no se calculan de la misma manera, ya que cada una proporcionarán diferentes señales en diferentes momentos [4].

2.3. Aprendizaje supervisado

Presentamos el paradigma de ajuste de funciones desde el punto de vista del aprendizaje automático. Supongamos, para simplificar, que los errores son aditivos y que el modelo $Y = f(X) + \varepsilon$ es una suposición razonable. El aprendizaje supervisado intenta aprender f a través de ejemplos proporcionados por un "maestro". Se observa el sistema en estudio, tanto las entradas como las salidas, y se forma un conjunto de entrenamiento de observaciones $T = (x_i, y_i), i = 1,...,N$. Los valores de entrada observados al sistema x_i también se introducen en un sistema artificial, conocido como algoritmo de aprendizaje (generalmente un programa de computadora), que también produce salidas $\hat{f}(x_i)$ en respuesta a las entradas. El algoritmo de aprendizaje tiene la propiedad de que puede modificar su relación de entrada/salida $\hat{f}$ en respuesta a las diferencias $y_i - \hat{f}(x_i)$ entre las salidas originales y las generadas. Este proceso se conoce como aprendizaje, por ejemplo al finalizar el proceso de aprendizaje, se espera que las salidas artificiales y reales sean lo suficientemente cercanas como para ser útiles para todos los conjuntos de entradas que es probable que se encuentren en la práctica [8].

Un área muy particular dentro del aprendizaje supervisado es el aprendizaje profundo y para fines de este proyecto este proyecto tiene un enfoque particularmente en las redes neuronales artificiales de las cuales se detallará en la siguiente sección.

Redes neuronales artificiales

En el libro de Li Deng, Dong Yu , *"Deep Learning: Methods and Applications"* [36] los autores enuncian 5 definiciones de alto nivel relacionadas al aprendizaje profundo, de las cuales vamos a rescatar las dos siguientes:

- Definición 1: Una clase de técnicas de aprendizaje automático que explotan muchas capas de procesamiento de información no lineal para la extracción y transformación de características supervisadas o no supervisadas, y para el análisis y clasificación de patrones.
- Definición 2: El aprendizaje profundo es una nueva de investigación del aprendizaje automático (machine learning), que se ha introducido con el objetivo de acercar el aprendizaje automático más cerca de uno de sus objetivos originales denominado comúnmente como inteligencia artificial. El aprendizaje profundo aprende múltiples niveles de representación y abstracción que ayudan a dar sentido a datos como imágenes, sonido y texto.

El aprendizaje profundo se encuentra en las intersecciones entre las áreas de investigación de redes neuronales, inteligencia artificial, optimización, modelado gráfico, procesamiento de señales y reconocimiento de patrones. Tres razones importantes para la popularidad del aprendizaje profundo en la actualidad son el drástico aumento de las capacidades de procesamiento de los chips (por ejemplo, las unidades de procesamiento gráfico de propósito general (GPGPU por sus siglas en inglés)), el aumento significativo del tamaño de los datos utilizados para el entrenamiento y los recientes avances en la investigación sobre aprendizaje automático y procesamiento de señales e información. Estos avances han permitido a

los métodos de aprendizaje profundo explotar eficazmente funciones no lineales complejas y compositivas, aprender representaciones de características distribuidas y jerárquicas, y hacer un uso eficaz tanto de los datos etiquetados como de los datos no etiquetados [36].

El modelo más representivo del aprendizaje profundo es la red profunda de alimentación o perceptrón multicapa (MLP), esta es una función matemática que determina un conjunto de valores de entrada a valores de salida. La función se compone de múltiples funciones más simples, de tal forma que cada aplicación de una función matemática diferente proporciona una nueva representación de la entrada.

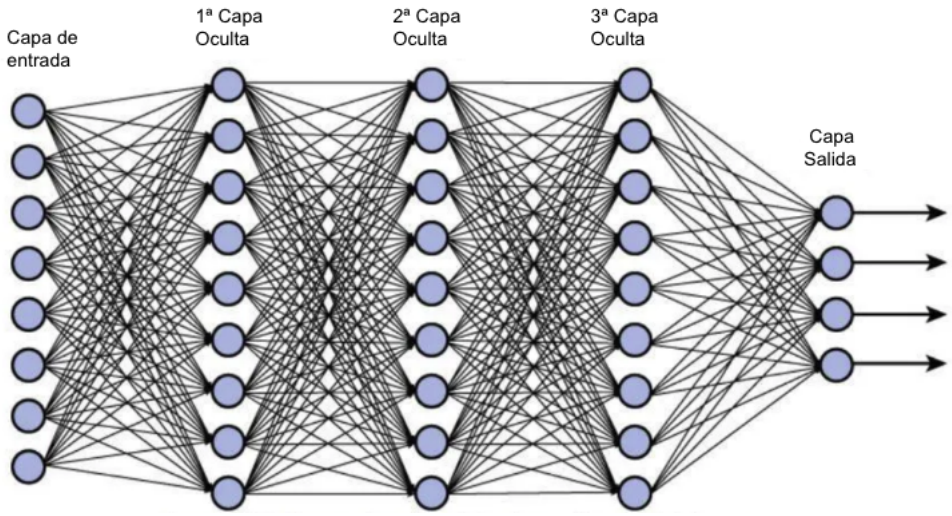

Figura 2.1: Ejemplo de red neuronal artificial, imagen adaptada de [15].

Cada capa de la red puede entenderse como el estado de la memoria del ordenador después de ejecutar instrucciones en secuencia. El poder de las redes neuronales se ve reflejado en las instrucciones posteriores pueden referirse a los resultados de las instrucciones anteriores. Dada esta situación el aprendizaje profundo permite que no toda la información de las funciones de activación de una capa afecta necesariamente los factores de variación que expliquen la entrada. De igual forma, la capa guarda información de estado que ayuda a ejecutar un programa que busca dar un sentido a la entrada.

Principalmente, se conocen dos formas de medir la profundidad de un modelo, una de ellas consiste en el número de instrucciones secuenciales que se ejecutan al evaluar la arquitectura, la cual se refiere a la longitud del camino más largo del diagrama de flujo que muestra cómo se calcula cada una de las salidas del modelo dadas sus entradas [38].

Bajo el enfoque del aprendizaje supervisado el problema de comercio algorítmico se plantea de forma matemática de la siguiente forma donde:

- $\mathbf{x}_t$ - Vector de características en el tiempo t. Esto puede incluir precios pasados, indicadores técnicos, volumen, entre otros.

- y_t - Etiqueta o valor objetivo en el tiempo t. Representa "1" si lo clasifica como compra, "0" para mantener y "-1" en el caso de venta.

- $f(\mathbf{x}_t)$ - Modelo que intenta predecir la etiqueta y_t basado en las características $\mathbf{x}_t$.

Dado que el objetivo es minimizar la función de pérdida:

$$L(y_t, f(\mathbf{x}_t)). \tag{2.8}$$

Donde L es una función de pérdida, que dado que el problema es predecir cuando comprar, vender o mantener se convierte en un problema de clasificación.

2.4. Algoritmos evolutivos

Los algoritmos evolutivos son un caso especial de metahurística donde se preserva una población de soluciones potenciales. La metaheurística regularmente está basada en procesos inspirados en la biología para generar poblaciones nuevas de soluciones las cuales preservan rasgos de la población inicial. Al igual que las metateurísticas de ascensión, estos procesos se repiten varias veces con la intención de encontrar una solución satisfactoria.

Las metaheurística destacan en algunos problemas por sobre otros métodos de búsqueda debido a que regularmente no se necesita saber mucho sobre el problema a resolver, una ventaja adicional es que las soluciones generadas por estos métodos no son intuitivas, así que, difícilmente hubieran sido propuestas por un ser humano [37].

Estrategia evolutiva (ES)

Las estrategias evolutivas tienen el mismo objetivo que el descenso de gradiente estocástico, es por ello que se les ve como una alternativa para optimizar. Estas estrategias se crearon a partir de la familia de algoritmos evolutivos [41]. Aplicamos la estrategia evolutiva (ES) para optimizar alguna función objetivo $f(x)$. El gradiente natural (por lo tanto, evolución de estrategia natural (por sus siglas en inglés, NES) [40]) trabaja con un espacio de distribución de probabilidad parametrizado por θ, $p_\theta(x)$. Y busca la dirección más pronunciada dentro de un pequeño paso en el espacio de distribución. Para encontrar una solución óptima para la función, se emplean las siguientes iteraciones [42]:

- Generar una población de muestras $D = (x_i, f(x_i))$ donde $x_i p_\theta(x)$.

- Evaluar la aptitud de las muestras en D.

- Seleccionar con base en su aptitud el mejor subconjunto de individuos y utilizarlos para actualizar θ, generalmente basado en la aptitud o el rango.

Con el fin de conocer la aptitud de las muestras en el paso dos, aplicamos un "jitter" (variación aleatoria) a cada una de las muestras del paso uno. El "jitter" se calcula a través de una varianza aleatoria basada en un hiperparámetro, σ, que controla cuánta varianza aleatoria existe en el "jitter". Posteriormente, en función de las muestras seleccionadas con esta variación aleatoria, se realiza el paso tres y luego se seleccionan las muestras con mejor comportamiento para la función $f(x)$.

Se ha demostrado que las estrategias evolutivas escalan de manera eficiente, lo que permite entrenar redes profundas para resolver problemas complicados de aprendizaje por refuerzo. También se ha descubierto que ES es una excelente técnica de optimización de caja negra cuando la función objetivo no tiene una forma analítica. Estos factores hacen que ES sea una alternativa efectiva al aprendizaje profundo por refuerzo y también más simple de implementar, ya que no se necesita retropropagación [41], sin embargo, dependiendo del objetivo y complejidad de la metaheurística implementada puede tener un costo computacional alto.

2.5. Aprendizaje profundo por refuerzo

El aprendizaje por refuerzo consiste en relacionar situaciones a acciones, con el objetivo de maximizar una recompensa numérica. No hay una supervisión sobre el agente aprendiz que le indique qué acciones tomar, sino que este debe descubrir qué acciones generan la mayor recompensa al intentarlas. En los experimentos más complicados e interesantes, las acciones suelen repercutir no solo la recompensa inmediata, sino también los siguientes estados y por ende todas las recompensas posteriores. Estas dos características: búsqueda mediante prueba y error y recompensa demorada, son las dos características más importantes que distinguen al aprendizaje por refuerzo [43].

En el aprendizaje por refuerzo además del agente y el entorno, se pueden identificar cuatro componentes principales de un sistema de aprendizaje por refuerzo: una política, una señal de recompensa, una función de valor (opcionalmente) y un modelo del entorno. Una política define la manera en que el agente de aprendizaje se comporta en un momento dado. A grandes rasgos, una política funciona como una función de correspondencia entre los estados percibidos del entorno y las acciones a tomar cuando se encuentra en esos estados. En algunos casos, la política se representa como una función simple o una tabla de búsqueda. La política es la parte más importante de un agente de aprendizaje por refuerzo ya que por sí sola es suficiente para determinar el comportamiento. Cabe señalar que la señal de recompensa muestra el beneficio de forma inmediata, a diferencia de la función de valor que busca el beneficio a largo plazo.

De forma general, el valor de un estado es la recompensa total que un agente espera acumular en el futuro, tomando como punto de partida el estado en cuestión. Las elecciones de acción toman decisiones basados en juicios de valor, es por ello, que siempre se buscan acciones que conduzcan a los estados de mayor valor debido a que siempre se prefiere la mayor cantidad de recompensa a largo plazo.

Sin embargo, estimar los valores es mucho más difícil que calcular las recompensas. Las recompensas se obtienen directamente al interactuar con el ambiente, por otro lado los valores deben aproximarse y actualizarce después de cada secuencia de observaciones realizadas por el agente. El modelo del entorno es el cuarto elemento del aprendizaje por refuerzo su función es imitar el comportamiento del ambiente donde se busca inferir cómo se comportará el ambiente [43].

Por otro lado, la complejidad de los algoritmos de aprendizaje por refuerzo no se suele describir en términos tradicionales de complejidad temporal (como $O(n)$, $O(n^2)$, etc.), debido a que su complejidad depende directamente del entorno, hiperparámetros, exploración modelo, etc.

Cabe mencionar que, dadas las características del aprendizaje por refuerzo el problema de comercio algorítmico se puede plantear de forma matemática de la siguiente manera:

- s_t - Estado en el tiempo t, que puede incluir información como precios históricos, indicadores técnicos, saldo, etc.
- a_t - Acción tomada en el tiempo t, que pertenece al conjunto de acciones posibles $A = \{\text{comprar}, \text{vender}, \text{mantener}\}$.
- r_t - Recompensa obtenida después de tomar la acción a_t en el estado s_t.
- $\pi(a|s)$ - Política, que es la probabilidad de tomar la acción a dado el estado s.
- γ es el factor de descuento que valora las recompensas inmediatas sobre las futuras.

El aprendizaje por refuerzo busca maximizar la recompensa esperada acumulativa, Ecuación 2.9, para ello, el objetivo es encontrar la política óptima π^* que maximice la recompensa esperada acumulativa, Ecuación 2.10:

$$J(\pi) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_t | s_0, \pi\right]. \tag{2.9}$$

$$\pi^* = \arg\max_{\pi} J(\pi). \tag{2.10}$$

2.5.1. Advantage Actor Critic (A2C)

El modelo actor crítico con ventaja (*Advantage Actor Critic*) tiene dos versiones principales: el Actor Crítico Asíncrono con Ventaja (*Asynchronous Advantage Actor Critic A3C*) y el actor crítico con ventaja (A2C).

A3C se describe en el artículo de Deepmind “Asynchronous Methods for Deep Reinforcement Learning"[44]. A grandes rasgos, A3C implementa un entrenamiento paralelo en el que varios trabajadores en entornos paralelos actualizan de forma independiente una función

de valor global, de ahí el término “asíncrono”. Una ventaja clave de contar con actores asíncronos es la exploración eficiente del espacio de estados.

A2C es como A3C pero sin la parte asíncrona, es decir, una versión que solo cuenta con un solo trabajador de A3C. Se ha comprobado empíricamente que A2C produce un rendimiento comparable al de A3C, al tiempo que es más eficiente. La implementación A2C síncrona funciona mejor que las implementaciones asíncronas - no ha sido observada alguna evidencia de que el ruido introducido por la asincronía proporcione algún beneficio en el rendimiento. Esta implementación A2C es más rentable que A3C cuando se utilizan máquinas con una sola GPU, y es más rápida que una implementación A3C con sólo CPU cuando se utilizan políticas más grandes [46].

2.5.2. Optimización de política proximal (PPO)

El algoritmo de optimización de política próximal combina ideas de A2C (tener múltiples trabajadores) y TRPO (utiliza una región de confianza para mejorar el actor). La idea principal es que, tras una actualización, la nueva política no debe alejarse demasiado de la anterior. Para ello, PPO utiliza el recorte para evitar una actualización demasiado grande [49] .

Existen dos versiones principales de PPO: *PPO-Penalty* y PPO-Clip . *PPO-Penalty* resuelve aproximadamente una actualización con restricciones Kullback-Leibler (KL) como TRPO, pero penaliza la divergencia KL en la función objetivo en lugar de convertirla en una restricción dura, y ajusta automáticamente el coeficiente de penalización a lo largo del entrenamiento para que se escale adecuadamente [47].

PPO-Clip no tiene un término de divergencia KL en el objetivo y no tiene ninguna restricción. En su lugar, se basa en un recorte especializado en la función objetivo para eliminar los incentivos para que la nueva política se aleje de la antigua. En este proyecto se centra sólo en PPO-Clip (la variante principal utilizada en OpenAI).

PPO entrena una política estocástica de forma on-policy. Esto significa que explora muestreando acciones según la última versión de su política estocástica. El grado de aleatoriedad en la selección de acciones depende tanto de las condiciones iniciales como del procedimiento de entrenamiento. A lo largo del entrenamiento, la política suele ser cada vez menos aleatoria, ya que la regla de actualización la anima a explotar las recompensas que ya ha encontrado. Esto puede hacer que la política quede atrapada en óptimos locales [48].

2.5.3. Gradiente de política determinista profundos (Deep Deterministic Policy Gradients DDPG)

DDPG se plantea en la publicación de Deepmind “Continuous Control With Deep Reinforcement Learning"[10] utiliza cuatro redes neuronales: una red Q, una red de política determinista, una red Q objetivo y una red de política objetivo.

La red Q y la red política son muy parecidas a la simple A2C, pero en DDPG, el actor asigna directamente los estados a las acciones (la salida de la red es directamente la salida) en lugar de emitir la distribución de probabilidad a través de un espacio de acción discreto.

Las redes objetivo son copias retardadas de sus redes originales que siguen lentamente a las redes aprendidas. El uso de estas redes de valor objetivo mejora enormemente la estabilidad en el aprendizaje. He aquí por qué: En los métodos que no utilizan redes objetivo, las ecuaciones de actualización de la red son interdependientes de los valores calculados por la propia red, lo que la hace propensa a la divergencia [50].

2.6. Optimización multi-objetivo

La optimización multi-objetivo consiste en encontrar varias soluciones que correspondan a la minimización (o maximización) de uno o varios objetivos definidos y que cumplan todas las restricciones especificadas (si las hay). En el caso del problema de optimización con un solo objetivo contiene una sola función objetivo y en la mayoría de los casos existe una única solución, a la que se conoce como solución óptima. Por lo tanto, una tarea de optimización multi-objetivo considera varios objetivos, los cuales, suelen estar en conflicto de manera simultánea.Es por ello, que difícilmente existe una única solución óptima, sino un conjunto de diferentes soluciones con distintos beneficios, conocidas como soluciones óptimas de Pareto o soluciones no dominadas. A pesar de la existencia de múltiples soluciones óptimas de Pareto, en la práctica sólo suele elegirse una de ellas. Por lo tanto, en comparación con los problemas de optimización de un solo objetivo, en la optimización multi-objetivo hay al menos dos tareas igualmente importantes: una tarea de optimización para encontrar soluciones óptimas de Pareto (mediante un procedimiento informático) y una tarea de toma de decisiones para elegir una única solución preferida. Esta última suele requerir información sobre las preferencias del responsable de la toma de decisiones.

En los años 60, varios investigadores propusieron de forma independiente adoptar los principios de la evolución natural, en particular la teoría de Darwin de la supervivencia del más apto, para la optimización. Los pioneros fueron Lawrence Fogel, John H. Holland, Ingo Rechenberg y Hans-Paul Schwefel. Una característica distintiva de los llamados algoritmos evolutivos (AE) es que trabajan con una población de soluciones. Esto es especialmente ventajoso en el caso de la optimización multi-objetivo, ya que pueden buscar varias soluciones óptimas de Pareto simultáneamente en una sola ejecución, proporcionando un conjunto de alternativas entre las que elegir. A pesar de algunas sugerencias y estudios iniciales, las principales actividades de investigación y aplicación de las EA en la optimización multiobjetivo, impulsadas por una sugerencia única de David E. Goldberg de una EA combinada con dominación y diferenciación de nichos, no comenzaron hasta principios de los años noventa. Pero en los últimos 15 años, el campo de la optimización evolutiva multi-objetivo (EMO por sus siglas en inglés) se ha desarrollado rápidamente, con una conferencia bianual dedicada, software comercial y más de 10 libros sobre el tema. Aunque los estudios anteriores se centraban en encontrar un conjunto representativo de soluciones en todo el conjunto óptimo de Pareto, las metodologías EMO también son buenas candidatas para encontrar sólo una parte del conjunto óptimo de Pareto [51].

NSGA-II

Los algoritmos evolutivos multiobjetivo (MOEA) que utilizan la ordenación no dominada y el reparto han sido criticados principalmente por: (1) su complejidad computacional $O(MN)$ (donde M es el número de objetivos y N el tamaño de la población); (2) su enfoque no elitista; y (3) la necesidad de especificar un parámetro de reparto. En el artículo de *A fast and elitist multiobjective genetic algorithm: nsga-II*, donde K. Deb, A. Pratap, S. Agarwal, y T. Meyarivan proponen un MOEA basado en la ordenación no dominante, denominado NSGA-II (Non-dominated Sorting Genetic Algorithm II), resuelve las tres dificultades anteriores.

Además, presentan un operador de selección que crea un conjunto de apareamiento en el cual, se combinan las poblaciones de padres e hijos y seleccionando las N mejores soluciones (con respecto a la aptitud y la dispersión). Los resultados de la simulación en problemas de prueba difíciles muestran que NSGA-II es capaz, para la mayoría de los problemas, de encontrar una mejor dispersión de soluciones y una mejor convergencia cerca del verdadero frente Pareto-óptimo en comparación con la estrategia de evolución Pareto-archivada y el algoritmo evolutivo fuerza-Pareto - otros dos MOEA elitistas que prestan especial atención a la creación de un frente Pareto-óptimo diverso. Además, modificamos la definición de dominancia para resolver problemas multi-objetivo restringidos de forma eficiente. Los resultados de simulación del NSGA-II restringido en una serie de problemas de prueba, incluido un problema no lineal de cinco objetivos y siete restricciones, se comparan con otro optimizador multiobjetivo restringido, y se observa un rendimiento mucho mejor del NSGA-II [52].

Con este tema concluimos este capítulo de marco teórico, el cual, procuró englobar los principales temas que fueron base para este proyecto de investigación y permitió desarrollar una metodología que se explica en el siguiente capítulo.

Capítulo 3

Metodología

Los capítulos anteriores permitieron describir las definiciones esenciales para abordar el problema de comercio algorítmico mediante algoritmos de inteligencia artificial expuesto en este proyecto.

A continuación se describe la metodología implementada para la importación, preprocesamiento de los datos, así como los modelos entrenados y puestos aprueba. Cabe mencionar, que el proyecto por completo fue desarrollado en Python y se incluyen en las secciones de código donde se adjuntará el código que se considere relevante así como los pseudo-códigos cuando sea necesario y su formulación matemática.

3.1. Introducción

En el presente capítulo se explica de forma detallada la manera en la que se obtuvieron los datos de los precios de las acciones, los indicadores financieros y estocásticos que se incluyen a partir de la información cualitativa. El preprocesamiento de la información para generar las señales y etiquetar de forma automática los momentos de compra y venta. Asímismo, se comentará acerca del proceso de selección de variables y se describirán las variables finales que formaron parte del entrenamiento, validación y prueba en la red neuronal así como en las estrategias evolutivas aplicadas. Se detalla además, las características de la red neuronal así como las estrategias evolutivas implementadas y los métodos que utilizamos para paralelizar la estrategia evolutiva con mútiples activos financieros. En la Figura 3.1 se muestra un diagrama que ilustra el flujo de trabajo bajo este enfoque.

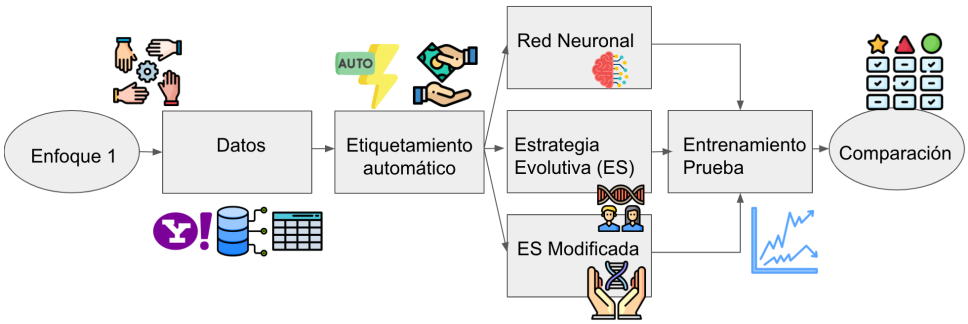

Figura 3.1: Diagrama de enfoque 1.

Posteriormente, con miras a mostrar una alternativa al nivel del estado del arte, se explica la forma en la que se implementó el marco de trabajo provisto por FinRL (Financial Reinforcement Learning) [21], donde se hizo uso de los métodos para extracción de los precios, cálculo de indicadores, inicialización del ambiente de *Gym OpenAI Trading* [22], conformación de los portafolios en función del giro empresarial de las acciones involucradas, configuración de los modelos provistos por FinRL [21], entrenamiento, una aplicación de NSGA-II [23] en el ajuste de hiperparámetros, validación, prueba y evaluación retrospectiva (*backtesting*) de los modelos ajustados.

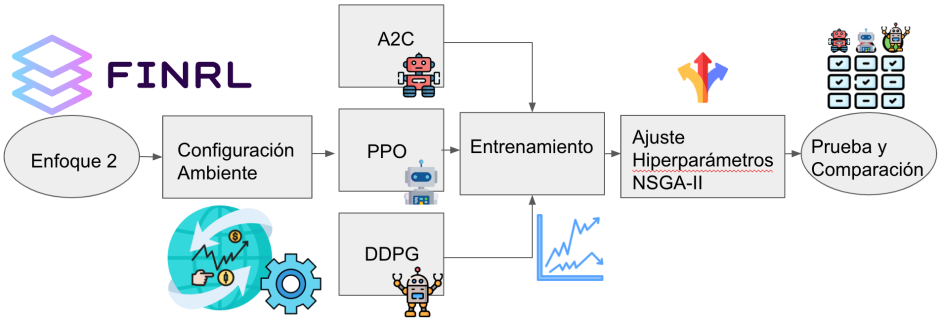

Figura 3.2: Diagrama de enfoque 2.

3.2. Datos

En la presente sección se mencionará acerca de la fuente donde se obtuvo la información histórica de los activos financieros con los cuales se experimentó, así como la representación de estas variables.

La obtención de datos resulta ser uno de los principales problemas para cualquier problema de ciencia de datos e inteligencia artificial, sin embargo, para abordar este problema en finanzas existen diferentes fuentes o API's que facilitan el acceso de forma gratuita a esta información, por ejemplo:

- Yahoo Finance [24].

- Quandl [25].
- Alpha Advantage [26].
- Google Finance [27].

Para fines de este proyecto se utilizó la paquetería *yfinance* [24], la cual, permite extraer directamente de Yahoo Finance la información cuantitativa de los precios de las acciones. A continuación se mostrará un pequeño ejemplo de uso en python.

```
import yfinance as yf
ticker='TSLA'
start_date = '2022-01-01'
data = yf.Ticker(ticker).history( start = start_date )[
    ['Open', 'High', 'Low', 'Close', 'Volume']
]
data.columns = map(str.lower, data.columns)
```

Código 3.1: Ejemplo de extracción de datos con *yfinance* [24].

Como se observa en el Código 3.1 uno de los parámetros es el *ticker* que corresponde a las siglas con que el activo se identifica en Yahoo finanzas, además para acotar la consulta se agrega la fecha inicial y limitamos a conservar solamente los campos Open, High, Low, Close y Volume, los cuales, se describen a continuación.

En la Tabla 3.1 se muestran los primeros cinco registros de la consulta resultante.

Date	open	high	low	close	volume
2022-01-03	382.58	400.35	378.67	399.92	103,931,400
2022-01-04	396.51	402.66	374.35	383.19	100,248,300
2022-01-05	382.21	390.11	360.33	362.70	80,119,800
2022-01-06	359.00	362.66	340.16	354.89	90,336,600
2022-01-07	360.12	360.30	336.66	342.32	84,164,700

Tabla 3.1: Ejemplo de extracción con *yfinance*.

Los campos que se recolectaron son los más utilizados en la literatura para construir la mayoría de los indicadores financieros, los precios se muestran en dólares o en su moneda de origen dependiendo donde se cotizan y representan lo siguiente:

- **Date:** fecha en formato *AAAA-MM-DD* a la que corresponde las variables cuantitativas.
- **open:** precio de apertura de la acción, regularmente los mercados del mundo operan de forma continua durante 9 horas. Así que, en función del mercado donde este activo cotice, este campo muestra el precio al que la acción aperturó en el mercado.

- **high:** es el precio más alto que el activo se logró cotizar durante esta fecha.
- **low:** es el precio más bajo que el activo se logró cotizar durante esta fecha.
- **close:** es el precio de cierre de la acción, en función del mercado donde este activo cotice, este campo muestra el precio último al que la acción se cotizó en el mercado.
- **volume:** se refiere a la cantidad de activos que fueron comerciados a lo largo del día, cuanto mayor es la cantidad se refiere a que es un activo altamente comerciado y por ende es fácil de intercambiar (comprar/vender).

Estas variables brindan información relevante sobre el comportamiento del activo financiero a lo largo del tiempo y al utilizar la paquetería de *mplfinance* [28] se puede generar una visualización interesante con el siguiente Código 3.2.

```
import mplfinance as mpf
mpf.plot(data.tail(100), type='candle', volume=True,style='yahoo')
```

Código 3.2: Ejemplo de gráfico con *mplfinance* [28].

Figura 3.3: Ejemplo de gráfico de velas y volumen con *mplfinance* [28] sobre el Ticker TSLA. El color rojo índica cuando el precio de apertura es mayor al precio de cierre, color verde indica lo opuesto, cuando el precio de cierre es mayor al de apertura.

La Figura 3.3 muestra en color rojo las fechas en donde el precio cerró por debajo del precio de apertura, y en color verde el caso opuesto, las líneas que sobrepasan los bordes representan al precio más alto en la parte superior y al precio más bajo en la parte inferior cotizado durante esa fecha. En la parte inferior, como un gráfico anexo se muestra a manera

de histograma los volúmenes operados en las fechas.

Ya que se abordó la descripción y el contexto de los datos cuantitativos que se utilizaron, en la siguiente sección se mencionará acerca del preprocesamiento que se le dió a esta información.

3.3. Preprocesamiento

En la presente sección se hablará del preprocesamiento de los datos, los indicadores que se calcularon, así como las señales que se diseñaron y el etiquetado automático para comprar/vender/mantener propuestos en este proyecto.

Con la información obtenida por medio de *yfinance* que se ejemplifica en la Tabla 3.1 se calcularon con la ayuda de la paquetería de *pandas_ta* [29] los siguientes indicadores:

- RSI.
- MACD.
- SMI.
- EMA.
- DEMA.
- BBANDS.
- ROC.
- TSI.

Los indicadores mencionados anteriormente se detallan en la sección 2.2 del Capítulo 2, así que en esta sección se ejemplificará en el Código 3.3 una forma de obtenerlos mediante la paquetería de *pandas_ta* [29].

```
import pandas_ta as pd

data.ta.rsi(close='close', length=14, append=True)
data.ta.macd(close='close', fast=12, slow=26, signal=9, append=True)
data.ta.smi(close='close', append=True)
data.ta.ema(close='close', append=True)
data.ta.dema(close='close', append=True)
data.ta.bbands(close='close', append=True)
data.ta.roc(close='close', append=True, window=1)
data.ta.tsi(close='close', append=True)
```

Código 3.3: Ejemplo de cálculo de indicadores con *pandas_ta* [29].

Lo anterior, da como resultado una tabla con los campos que se enlistan en la Tabla 3.2.

	Columna	Valores	nulos	Tipo
0	open	284	0	float64
1	high	284	0	float64
2	low	284	0	float64
3	close	284	0	float64
4	volume	284	0	int64
5	RSI_14	270	14	float64
6	MACD_12_26_9	259	25	float64
7	MACDh_12_26_9	251	33	float64
8	MACDs_12_26_9	251	33	float64
9	SMI_5_20_5	265	19	float64
10	SMIs_5_20_5	265	19	float64
11	SMIo_5_20_5	265	19	float64
12	EMA_10	275	9	float64
13	DEMA_10	275	9	float64
14	BBL_5_2.0	280	4	float64
15	BBM_5_2.0	280	4	float64
16	BBU_5_2.0	280	4	float64
17	BBB_5_2.0	280	4	float64
18	BBP_5_2.0	280	4	float64
19	ROC_10	274	10	float64
20	TSI_13_25_13	260	24	float64
21	TSIs_13_25_13	260	24	float64

Tabla 3.2: Ejemplo de Tabla con Indicadores calculados con *pandas_ta* [29].

Una vez que se cuenta con los indicadores que se utilizarán en este proyecto, se propone generar diferentes señales basadas en la documentación de los indicadores mismos, así que los primeros indicadores con los que se trabajarán estas señales son EMA y DEMA. La idea general de esta señal que se propone es indicar cuando el precio de cierre se encuentra por encima o por debajo del valor del EMA o DEMA, así como medir la "intensidad" de esta diferencia, con el fin de proporcionar total precisión del cálculo de esta señal. Se muestra a continuación el pseudo-código en en el Algoritmo 1.

Algoritmo: Señal EMA o DEMA.
Resultado: Señal (S) e intensidad (T)
Entrada: ;
Arreglo del precio de cierre (C), Arreglo de indicador EMA o DEMA (I);
Inicialización:
Sea N la longitud del arreglo C;
Se define al arreglo S;
Se define al arreglo T;
para $k \leftarrow 1$ **a** N **hacer**
  si $I[k] < C[k]) \& (I[k-1] < C[k-1])$ **entonces**
    $S[k] = 1$;
    $T[k] = C[k]/I[k]$;
  si no, si $(I[k] > C[k]) \& (I[k-1] > C[k-1])$ **entonces**
    $S[k] = -1$;
    $T[k] = -C[k]/I[k]$;
  en otro caso
    $S[k] = 0$;
    $T[k] = -C[k]/I[k]$;
  fin
fin
devolver S, T

Algoritmo 1: Algoritmo para generar la Señal e Intensidad con EMA o DEMA.

En la Figura 3.4 se observa gráficamente el comportamiento de la señal y la intensidad generada con el Algoritmo 1 utilizando como entrada el indicador EMA de parámetro (12).

Figura 3.4: Ejemplo de la señal e intensidad generada por el algoritmo 1.

La siguiente señal que se propone es utilizar el indicador RSI, la idea general de la señal es marcar los momentos de sobrecompra y sobreventa los cuales se consideran que son cuando el indicador se encuentra en valores inferiores a 30 y en sobrecompra cuando rebasa el valor de 70, la señal indica estos cruces marcando con -1 los momentos de sobrecompra y con $+1$ los momentos de sobreventa. Por otro lado, se genera una variable objetivo la cual consiste en aquellos puntos de sobreventa/sobrecompra que representen una utilidad superior al umbral definido del $1\,\%$ esto con el fin de evitar las falsas alertas de compra o venta de la señal. A continuación, se describe en el Algoritmo 2 con el cual se genera la señal y variable objetivo a partir del indicador RSI.

Algoritmo: Señal y Objetivo RSI.
Resultado: Señal RSI (S)
Entrada: ;
Arreglo de indicador RSI (I);
Arreglo de precios de cierre (C);
Inicialización:
Sea N la longitud del arreglo C;
Se define al arreglo S de dimensión N;
Se define T como un arreglo de 0's con dimensión N;
Se define al arreglo B;
Se define $MIN()$ como la función que regresa el mínimo de un arreglo.;
para $k \leftarrow 1$ **a** N **hacer**
  si $I[k] < 30$) **entonces**
    $S[k] = 1$;
    Agregar al arreglo B el valor de $C[k]$;
  si no, si $(I[k] > 70)$ **entonces**
    $S[k] = -1$;
    $compra = MIN(B)$;
    $venta = C[k]$;
    si $venta/compra - 1 > 0{,}01$ **entonces**
      Guardar un $+1$ en la posición j de T tal que $compra = C[j]$;
      $T[k] = -1$;
      Eliminar el elemento $compra$ del arreglo B ;
  en otro caso
    $S[k] = 0$;
  fin
fin
devolver S, T

Algoritmo 2: Algoritmo para generar la Señal y Objetivo con RSI.

En la Figura 3.5 se puede observar gráficamente como se comporta la señal y la variable objetivo generada con el Algoritmo 2 utilizando como entrada el indicador RSI de parámetro 14. Se puede observar que dada la condición de solo conservar aquellas operaciones de

compra/venta que generen más del umbral de rendimiento definido 1 %, permite evadir las falsas señales propias del indicador, y que esta variable objetivo obtenida no se perjudique por las falsas señales.

Figura 3.5: Ejemplo de la señal y variable objetivo generada por el algoritmo 2.

La siguiente señal que se propone es utilizando el indicador MACD, la idea general de la siguiente señal es marcar los momentos donde las dos medias móviles que conforman al indicador se cruzan, si se cruzan por debajo es una señal de compra, si se cruzan por encima es una señal de venta, así que aprovechando la interpretabilidad del indicador, la señal indica estos cruces marcando con $+1$ las compras y con -1 las ventas, de igual forma que el algoritmo 2, se genera una variable objetivo basada en el indicador, siempre y cuando las operaciones de compra/venta superen el umbral de rendimiento que se definió como el 1 %. A continuación se mostrará en el Algoritmo 3 la generación de esta señal con su variable objetivo.

Algoritmo: Señal y objetivo MACD.
Resultado: Señal MACD (S)
Entrada: ;
Arreglo de indicador MACD (I);
Arreglo de indicador MACD señal (Is);
Arreglo de precios de cierre (C);
Inicialización:
Sea N la longitud del arreglo C;
Se define el arreglo S de dimensión N;
Se define T como un arreglo de 0's con dimensión N;
Se define al arreglo B;
Se define $MIN()$ como la función que regresa el mínimo de un arreglo.;
para $k \leftarrow 1$ **a** N **hacer**
  si $(I[k-1] < Is[k-1])\&(I[k] > Is[k])$ **entonces**
    $S[k] = 1$;
    Se agrega al arreglo B el valor de $C[k]$;
  si no, si $(I[k-1] > Is[k-1])\&(I[k] < Is[k])$ **entonces**
    $S[k] = -1$;
    $compra = MIN(B)$;
    $venta = C[k]$;
    si $venta/compra - 1 > 0{,}01$ **entonces**
      Se guarda un $+1$ en la posición j de T tal que $compra = C[j]$;
      $T[k] = -1$;
      Se elimina el elemento $compra$ del arreglo B ;
  en otro caso
    $S[k] = 0$;
  fin
fin
devolver S, T

Algoritmo 3: Algoritmo para generar la Señal y Objetivo con MACD.

En la Figura 3.5 se puede observar gráficamente como se comporta la señal y la variable objetivo generada con el Algoritmo 3 utilizando como entrada el indicador MACD de parámetro $(12,26,9)$.

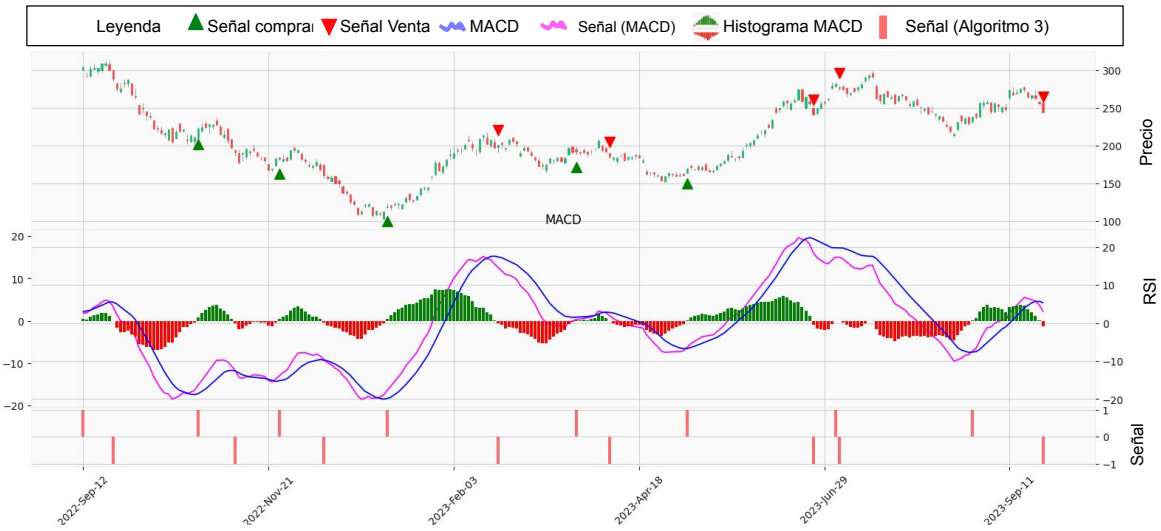

Figura 3.6: Ejemplo de la señal y variable ojetivo generada por el algoritmo 3.

La siguiente señal que se propone es utilizando el indicador SMI, la idea general de la siguiente señal es marcar los óptimos locales presentes en el SMI señal donde los mínimos locales serán momentos de compra, los máximos locales de venta y se etiquetará con $+1$ y -1 respectivamente. Además, se genera una variable objetivo basada en la señal siempre y cuando la operación compra/venta rebase el umbral de rendimiento de $1\,\%$.

A continuación se mostrará en el Algoritmo 4 la generación de la señal y la variable objetivo.

Algoritmo: Señal y objetivo SMI.
Resultado: Señal SMI (S) y variable objetivo (T)
Entrada: ;
Arreglo de indicador SMI señal (Is);
Arreglo de precios de cierre (C);
Inicialización:
Sea N la longitud del arreglo C;
Se define al arreglo S de dimensión N;
Se define T como un arreglo de 0's con dimensión N;
Se define al arreglo B;
Se define $MIN()$ como la función que regresa el mínimo de un arreglo.;
Se define al arreglo MX como el conjunto de índices donde se encuentran los máximos locales de Is.
Se define al arreglo MN como el conjunto de índices donde se encuentran los mÍnimos locales de Is.
para $k \leftarrow 1$ **a** N **hacer**
  si $MN[k] = k$ **entonces**
    $S[k] = 1$;
    Se agrega al arreglo B el valor de $C[k]$;
  si no, si $MX[k] = k$ **entonces**
    $S[k] = -1$;
    $compra = MIN(B)$;
    $venta = C[k]$;
    si $venta/compra - 1 > 0{,}01$ **entonces**
      Guardar un $+1$ en la posición j de T tal que $compra = C[j]$;
      $T[k] = -1$;
      Eliminar el elemento $compra$ del arreglo B ;
  en otro caso
    $S[k] = 0$;
  fin
fin
devolver S, T

Algoritmo 4: Algoritmo para generar la señal y objetivo con SMI.

Con el fin de mostrar una visualización del comportamiento de la señal basada en SMI así como la variable objetivo se genera la Figura 3.7.

En resumen, se busca que los indicadores y señales anteriormente mostrados brinden información relevante sobre los patrones y tendencias de los activos financieros.

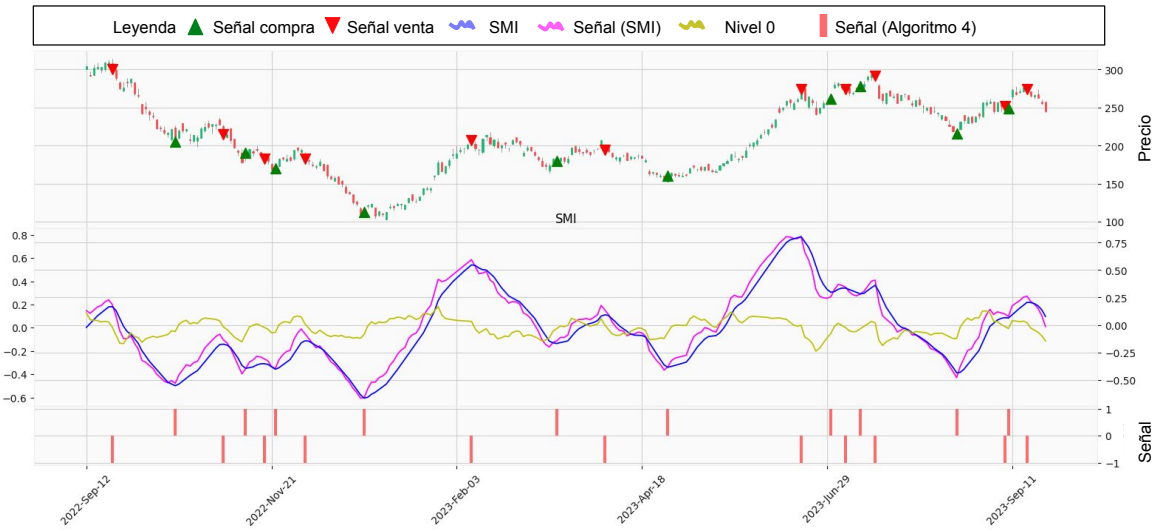

Figura 3.7: Ejemplo de la señal y variable objetivo generada por el algoritmo 4.

3.4. Etiquetamiento automático

Dentro del ámbito del aprendizaje supervisado, uno de los retos más complejos es la obtención de datos etiquetados de manera precisa que sirvan para el entrenamiento, la validación y la prueba de modelos de aprendizaje. La importancia de los datos etiquetados adecuadamente es de suma importancia, ya que esto constituye la base sobre la cual las máquinas pueden identificar los patrones correctos y hacer predicciones precisas. Existen numerosas aplicaciones y compañías que han emergido para facilitar el proceso de etiquetamiento de datos, dirigidas específicamente a mejorar los procedimientos del aprendizaje automático. Por lo tanto, esta sección tiene como finalidad presentar una metodología simplificada que permita el etiquetado automático de nuestra información de forma eficiente y efectiva.

En el contexto específico del comercio algorítmico, el desafío reside en identificar los puntos óptimos de sobrecompra y sobreventa para realizar operaciones que maximicen los rendimientos financieros. Se entiende por sobrecompra a aquel escenario donde el precio de un activo está inflado, lo cual sugiere que podría ser un momento propicio para la venta. Inversamente, la sobreventa se refiere a la situación en la que un activo se percibe como infravalorado, lo que podría señalar una oportunidad de compra. La habilidad para identificar y actuar en estos momentos es lo que fundamenta las estrategias para aumentar las ganancias.

Dentro de este proyecto, hemos empleado indicadores conocidos como osciladores, que son herramientas analíticas diseñadas para señalar estados de sobrecompra y sobreventa. Sin embargo, dichos indicadores no son infalibles y pueden producir señales erróneas, por lo que no deben considerarse como reglas absolutas de inversión.

El proceso de etiquetado que se propone en este proyecto utiliza un umbral definido junto con las señales generadas por osciladores tales como el MACD, el RSI y el SMI, tal como

se detalla en los Algoritmos 2, 3 y 4. Este umbral está diseñado para asegurar un etiquetado que se enfoque en una estrategia de inversión que busq a rendimientos financieros atractivos.

En esta metodología, las señales de los indicadores sugieren momentos óptimos para realizar operaciones de compra o venta. No obstante, el umbral establecido nos permite filtrar aquellas operaciones que no alcanzan un rendimiento superior a un porcentaje "X%", permitiendo así una capa de estrategia y adaptabilidad para el inversor encargado del etiquetado.

Finalmente, al obtener las señales objetivo de los indicadores SMI, RSI y MACD, se procede a su consolidación en una sola señal objetivo. Esto implica que si al menos uno de los indicadores sugiere una acción de compra o venta, la señal objetivo consolidada lo reflejará. El resultado de este etiquetado compuesto se visualiza en la Figura 3.8, y la Ecuación 3.1 detalla el cálculo pertinente para obtener esta señal objetivo unificada. Con esto, se busca proveer un marco de referencia claro y aplicable para la toma de decisiones en el comercio algorítmico basado en datos etiquetados de forma inteligente y estratégica.

$$Objetivo_t^{general} = \begin{pmatrix} -1 & \text{Si } Objetivo_t^{SMI} = -1 || Objetivo_t^{MACD} = -1 || Objetivo_t^{RSI} = -1 \\ 1 & \text{Si } Objetivo_t^{SMI} = 1 || Objetivo_t^{MACD} = 1 || Objetivo_t^{RSI} = 1 \\ 0 & \text{e. o. c.} \end{pmatrix}. \tag{3.1}$$

Donde $Objetivo_t^i$ se refiere a la variable objetivo generada por el indicador i, observada al tiempo t.

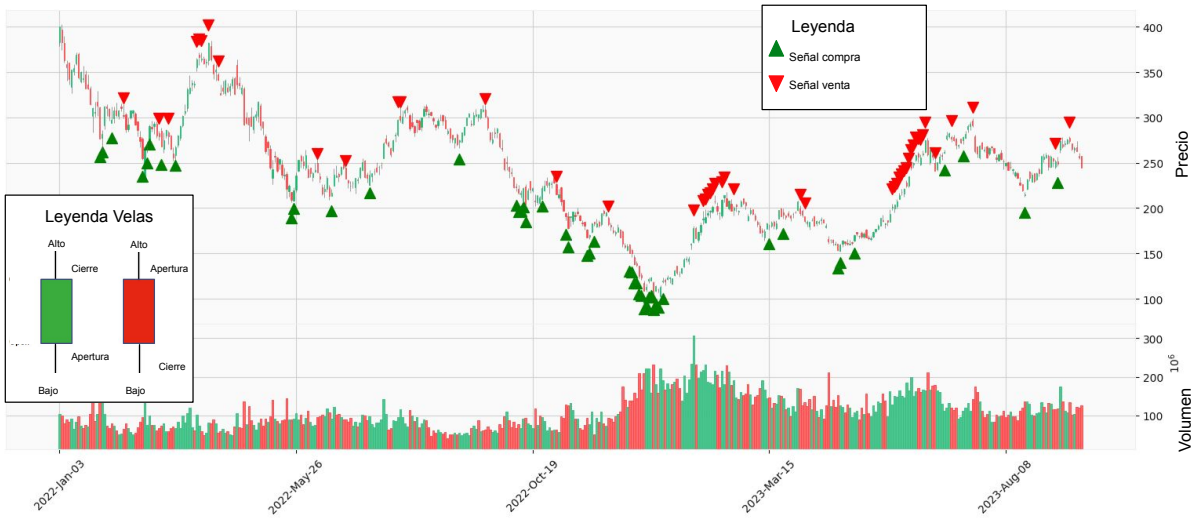

Figura 3.8: Ejemplo de la variable objetivo resultante de unificar las respectivas de MACD, RSI y SMI.

En el ejemplo mostrado, el combinar las 3 variables objetivo generadas con el fin de obtener un mayor número de operaciones compra/venta exitosas, del cual si se hubiera seguido esta estrategia se obtendría un rendimiento del 928 % la ventana de tiempo del ejercicio.

Consolidada la variable objetivo, en la Tabla 3.4 se muestran las variables que se agregan al conjunto de datos a trabajar.

	name	type
22	rsi_s	int64
23	macd_s	int64
24	smi_s	int64
25	target	int64
26	ror	float64
27	ema_s	int64
28	ema_perc	float64
29	dema_s	int64
30	dema_perc	float64

Tabla 3.3: Variables añadidas después de las Tabla con indicadores calculados con *pandas_ta*.

Posteriormente, el precio de cierre, el MACD y el SMI, se aplicó al siguiente algoritmo con el fin de obtener una escalarización de los datos (normalización o estandarización según sea el caso), de las variaciones de estos valores en función a una ventana de n periodos. A continuación, en el Algoritmo 5 se muestra como generar este indicador con cada variable.

El Algoritmo 5, brinda una matriz de tamaño NxV, para cada una de los indicadores para el cual se efectúe, lo cual aumenta el número de campos a trabajar. A continuación se muestra la representación de esta medición con respecto a los indicadores que se aplicaron.

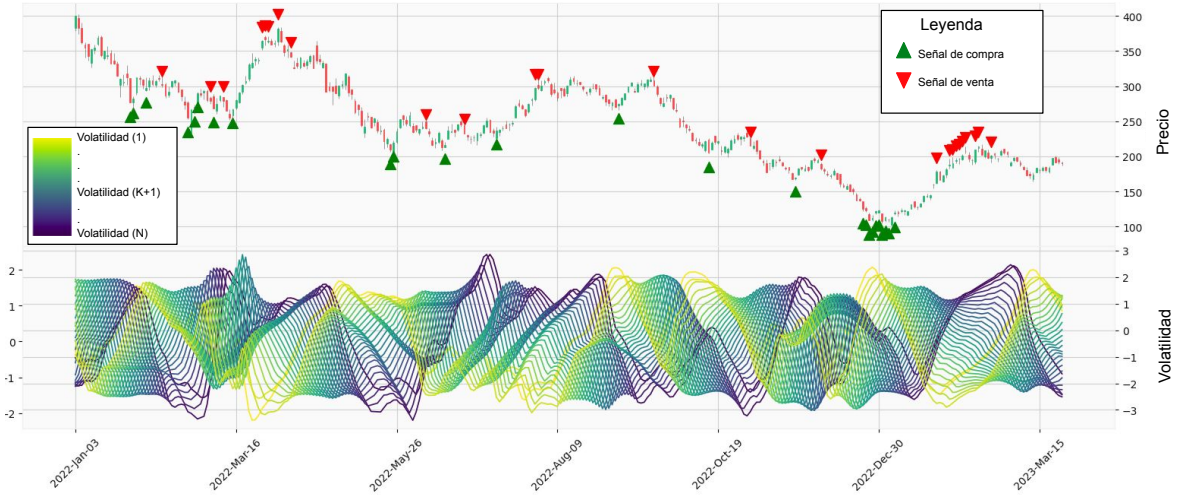

Figura 3.9: Ejemplo de la volatilidad calculada en el algoritmo 4, señales de compra y venta resultante de unificar las señales de MACD, RSI y SMI de los algoritmos anteriores.

En la siguiente sección se mencionará acerca de la selección de variables que se utilizaron

Algoritmo: Indicador de volatilidad normalizada/estandarizada.
Resultado: Matriz de dimensión NxV (M)
Entrada: ;
Arreglo con precio de cierre, MACD o SMI (S);
Dimensión de la ventana (V);
Bandera de normalización (B) ;
Inicialización:
Sea N la longitud del arreglo (S);
Se define la matriz de dimensión NxV (M);
Se define a $MIN()$ como la función que obtiene el mínimo de una lista;
Se define a $MAX()$ como la función que obtiene el máximo de una lista;
Se define a $MEAN()$ como la función que obtiene la media de los elementos de una lista;
Se define a $ST()$ como la función que obtiene la desviación estándar de los elementos de una lista;
para $k \leftarrow 0$ **a** $N-1$ **hacer**
    $d = k - V$;
    si $d < 0$ **entonces**
        Se repiten $-d$ veces el primer elemento en S, (F) ;
        Se toman los primeros j elementos de S , (L) ;
        Se unifican F,L en una sola lista T;
    en otro caso
        Se toman los elementos de la posición d a k del arreglo S, (T);
    fin
    Se inicializa el arreglo R ;
    para $j \leftarrow 0$ **a** $V-1$ **hacer**
        $R[j] = T[j+1] - T[j]$;
    fin
    si $B = 1$ **entonces**
        $R = (R - MIN(R))/(MAX(R) - MIN(R))$;
    en otro caso
        $R = (R - MEAN(R))/ST(R)$;
    fin
    Se agrega a R a la fila k-ésima de la matriz M
fin
devolver M

Algoritmo 5: Algoritmo para calcular la volatilidad normalizada/estandarizada en n períodos.

para los modelos aplicados, así como los detalles de cada uno de los modelos.

3.5. Modelos

En la presente sección, se describen los modelos que se entrenaron para clasificar los momentos de compra/venta para diferentes activos.

Los modelos que se proponen para clasificar los momentos de compra/venta son:

- Red neuronal [36].
- Estrategia evolutiva.
- Estrategia evolutiva con recompensa ponderada por la variable objetivo.

3.5.1. Selección de Variables

A este punto, el conjunto de datos contiene todas las variables con las que se trabajará, así que del total de 117 variables es necesario una selección que permita utilizar tan solo las que brinden información relevante para entrenar al modelo. Con esto en mente, se realizó la normalización de las variables que no se encontraran en intervalos de $[-1,1]$, *MACD, Volume y open* por mencionar algunas, esto con el fin de que no sesgar al modelo con la información de alguna variable en particular.

Se inicia la selección de variables con un análisis de la correlación entre cada una de las variables con el fin de identificar aquellas que no estuvieran relacionadas y que de esta forma aportaran información nueva, la Figura 3.10 ilustra la dimensión de la matriz de correlación que se analizaron, para delimitar el conjunto de variables.

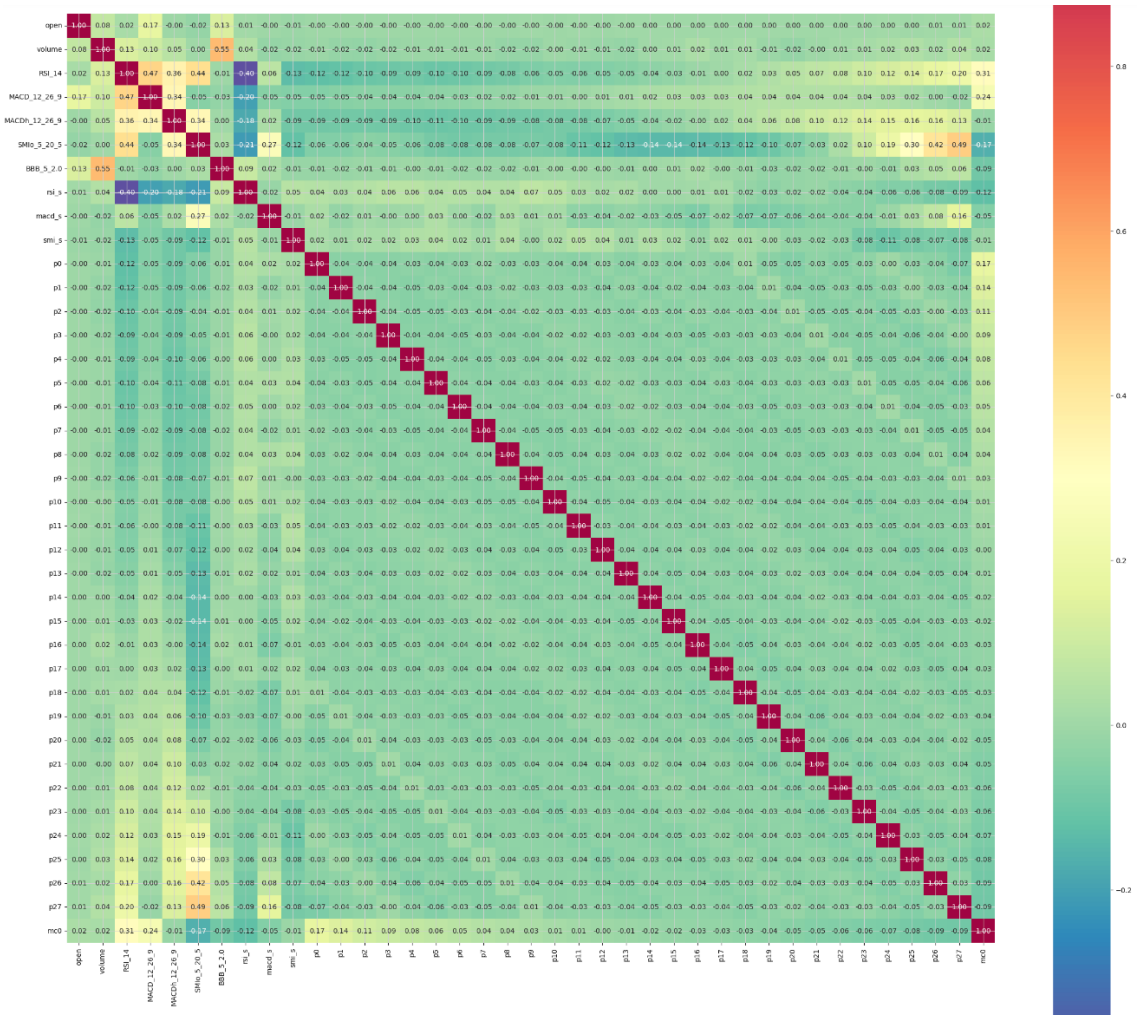

Figura 3.10: Ejemplo de la matriz de correlación el orden de los campos es el mismo que en la Tabla 3.4.

Con el fin de evitar la colinealidad e incluir en este modelo las variables que aporten información relevante, se realizaron dos métodos de selección de variables que posteriormente se explicará brevemente:

- Información mutua: La información mutua (MI) entre dos variables aleatorias es un valor no negativo, que mide la dependencia entre las variables. Es igual a cero si y solo si dos variables aleatorias son independientes, y los valores más altos significan una mayor dependencia [30].

  La función se basa en métodos no paramétricos basados en la estimación de la entropía a partir de las distancias de k vecinos más cercanos, como se describe en [31] y [32]. Ambos métodos se basan en la idea propuesta originalmente en [33].

- Correlación de Pearson: La correlación bivariada de Pearson produce un coeficiente de correlación de muestra, r, que mide la fuerza y la dirección de las relaciones lineales entre pares de variables continuas. Por extensión, la Correlación de Pearson evalúa si

existe evidencia estadística de una relación lineal entre los mismos pares de variables en la población, representada por un coeficiente de correlación poblacional, (“rho”) [34]. La correlación se calcula como se muestra en la Ecuación 3.2.

$$r = \frac{\Sigma(x - m_x)(y - m_y)}{\sqrt{\Sigma(x - m_x)^2 \Sigma(y - m_y)^2}}. \tag{3.2}$$

Donde m_x se refiere a la media del vector x y m_y la media del vector y. [35]

Los resultados de excluir a las variables que su correlación de Pearson fuera superior a un umbral de 0.6, permitieron seleccionar 36 variables las cuales son mostradas en la Tabla 3.4.

	Variable
0	open
1	volume
2	RSI_14
3	MACD_12_26_9
4	MACDh_12_26_9
5	SMIo_5_20_5
6	BBB_5_2.0
7	rsi_s
8	macd_s
9	smi_s
10	p0 al p27
38	mc0

Tabla 3.4: Variables con correlación de Pearson menor a 0.6 en valor absoluto.

Por otro lado, se realizó una selección mediante el método de información mutua y los resultados arrojaron 18 variables con un valor superior a 0.04 de información mutua.

	name
0	smi_s
1	rsi_s
2	macd_s
3	SMIs_5_20_5
4	SMIo_5_20_5
5	RSI_14
6	SMI_5_20_5
7	MACD_12_26_9
8	mc27
9	MACDh_12_26_9
10	mc28
11	TSI_13_25_13
12	sm26
13	ema_perc
14	sm27
15	sm28
16	mc26
17	ema_s

Tabla 3.5: Variables con información mutua superior a 0.04.

En conjunto las variables relevantes para entrenar el modelo, según los métodos de selección de variables implementados, son 50, se logra reducir la dimensionalidad en un 56 %, las cuales se muestran en la Tabla 3.5 y 3.4.

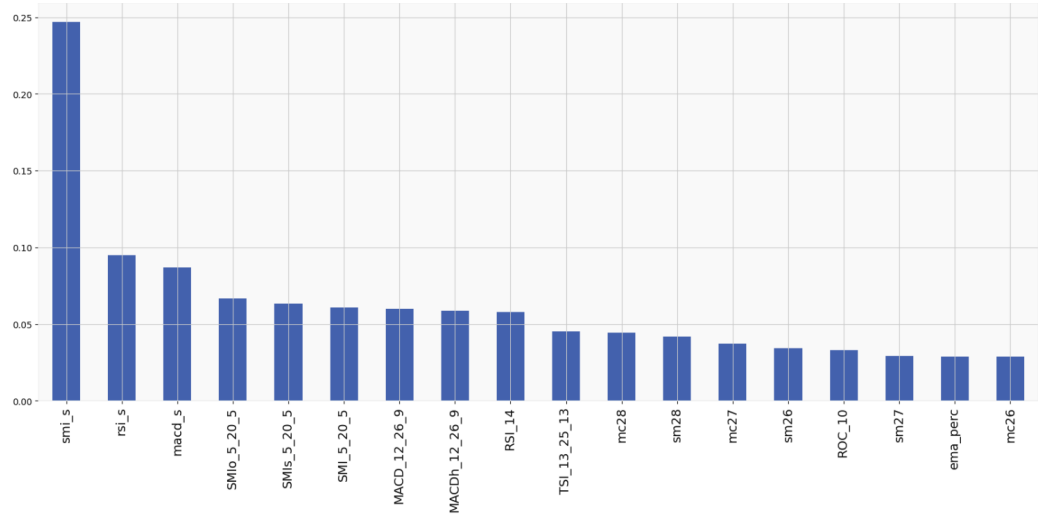

Figura 3.11: Variables con mayor índice de información mutua.

Ahora, en la siguiente sección se abordan los pormenores de la red neuronal implementada dado el conjunto de variables seleccionadas.

3.5.2. Red neuronal

Es importante recordar que el objetivo del modelo debe ser diferenciar correctamente los momentos de compra/venta y para fines de las pruebas con la red neuronal se medirá de dos formas:

- Puntaje f1: Es la media armónica entre la precisión y la exhaustividad (recall).

$$F1 = 2 \cdot \frac{\text{Precisión} \times \text{Recall}}{\text{Precisión} + \text{Recall}}. \tag{3.3}$$

  Donde: Precisión $= \frac{TP}{TP+FP}$ y Recall $= \frac{TP}{TP+FN}$, siendo TP los verdaderos positivos y FN los falsos negativos.

- Rendimiento: Cálcula el porcentaje de ganancia obtenido al tiempo t.

$$R = \frac{V_t - V_0}{V_0}. \tag{3.4}$$

  Donde: V_t es el valor del portafolio al tiempo t y V_0 es el valor inicial.

Más allá de que el modelo logre imitar con precisión el comportamiento de la variable objetivo, se busca que el rendimiento generado por las acciones tomadas sea atractivo con respecto a la inflación. Así que los indicadores de exactitud y precisión serán relevantes para darn una idea de si el algoritmo está aprendiendo correctamente, sin embargo, el rendimiento generado por la estrategia dictada por el algoritmo dará mayor claridad sobre el objetivo alcanzado por el modelo.

La arquitectura de la red neuronal que se propone es densa de tipo secuencial con 4 capas ocultas como se muestra en la Tabla 3.6.

Capa	Tipo	Dimensión	Función de Activación	Parámetros
1	Densa	(328)	Relu	16,728
2	Densa	(128)	Relu	42,112
3	Densa	(64)	Relu	8,256
4	Densa	(32)	Relu	2,080
5	Salida	(3)	Softmax	99

Tabla 3.6: Arquitectura de la red neuronal aplicada.

El entrenamiento se realizó con el 80 % de los datos, la red utilizó un optimizador Adam con una tasa de aprendizaje de 0.001 y función de pérdida la entropía cruzada categórica.

Además se agregó una función de paro en función de la pérdida en el conjunto de validación, donde después de 20 épocas donde no mejorara mantuviera los pesos de la red que mejor se ajustó. Los resultados en el conjunto de prueba se muestran en la Tabla 3.7, donde como se observa en el soporte más del 80 % de los resultados están clasificados como “Mantener", lo que significa que los datos están bastante imbalanceados, es por ello, que el puntaje f1 da un indicador de desempeño más alineado a las necesidades de este proyecto. Dicho lo anterior, el modelo tiene dificultades especialmente para predecir los momentos de “Compra".

	Precisión	Exhaustividad	Puntaje f1	Soporte
Venta	0.887	0.875	0.881	72
Mantener	0.979	0.907	0.941	518
Compra	0.535	0.979	0.692	47
Exactitud	0.909	0.909	0.909	0.909
Promedio macro	0.801	0.920	0.838	637
Promedio ponderado	0.936	0.909	0.917	637

Tabla 3.7: Resultados del modelo entrenado en el conjunto de prueba.

Se toma el resultado de la predicción para realizar la evaluación retrospectiva, donde se supone un dinero inicial de 100,000, además como estrategia cada que se predice un momento de venta, se considera vender todas las acciones compradas hasta el momento, y así se obtienen los resultados de la Tabla 3.8, donde debido a que el modelo tiene dificultades en predecir los momentos de compra, esto lo límita a tener un rendimiento mayor al 11 %.

Ganancia	Acciones restantes	Último precio comprado	% Ganado	Monto comprado
1,241.03	2	108.09	8.93 %	18,147.76

Tabla 3.8: Resultados de la evaluación retrospectiva.

Este es el ejemplo paso a paso de la metodología que se implementó en cada uno de los activos con los que se realizó este proyecto. En la siguiente sección se aborda los pormenores de la estrategia evolutiva.

3.5.3. Estrategia evolutiva

En esta sección se describen los pormenores de las dos estrategias evolutivas implementadas, cabe mencionar que la segunda estrategia evolutiva, presenta una modificación en la forma en la que se recompensa al agente, en función del etiquetamiento propuesto en este trabajo.

Las estrategias evolutivas implementadas se conforman de los siguientes componentes:

- Agente.
- Población.

- Estados/Observaciones: Basados en el preprocesamiento anteriormente trabajado.
- Recompensa.
- Modelo.

La estrategia evolutiva propuesta utiliza un modelo de red neuronal con las características mostradas en la Tabla 3.9, la cual se inicializa de forma aleatoria:

Capa	Tipo	Dimensión	Función de Activación	Parámetros
1	Densa	(500)	Relu	25,500
2	Salida	(3)	Softmax	1,501

Tabla 3.9: Arquitectura de la red neuronal aplicada en la estrategia evolutiva.

La estrategia evolutiva buscará ajustar los pesos de la red neuronal del modelo en busca de los pesos que máximicen la recompensa, o en este caso el rendimiento. La forma en la que realizará el ajuste será con ayuda de una población, donde cada individuo consta de su propio modelo (red neuronal) con el cual se enfrentará al ambiente en busca del mayor rendimiento. Como se muestra en la Ecuación 3.5, el modelo presente en cada individuo, no es más que el modelo actual del agente sumado a un sesgo aleatorio multiplicado por un peso $\delta \in (0,1)$.

$$\begin{aligned} W^i_{k,j,l} &= W_{k,j,l} + \delta * N, \\ N &\sim \mathcal{N}(0,1). \end{aligned} \tag{3.5}$$

Donde W^i se refiere a los pesos de la red neuronal del i-ésimo individuo, W son los pesos de la red neuronal general del agente, k,j,l se refiere a la entrada j,l de la k-ésima capa y N es un número aleatorio de una distribución normal estándar.

Una vez que cada individuo tiene definida su red neuronal, estos se enfrentan al ambiente de forma independiente y obtienen cada uno una recompensa (r) 3.6, en función del dinero inicial y el dinero resultante después de comerciar.

$$r = D_f / D_i - 1. \tag{3.6}$$

Donde D_f es el dinero final resultante de las operaciones del agente, D_i es el dinero inicial con el que el agente apertura sus operaciones.

Posteriormente, con una recompensa asignada a cada uno de los individuos, se estandarizan las recompensas y se actualizan los pesos de W en función de los pesos del modelo de cada individuo ponderando con respecto a la recompensa obtenida, como se muestra en la Ecuación 3.7.

$$W_k = W_k + \frac{\alpha}{P * \delta} * (W^A_k \cdot R)^\top. \tag{3.7}$$

Donde k, se refiere a la k-ésima capa de la red neuronal, α corresponde a la tasa de aprendizaje, P es el tamaño de la población, W_k^A es el arreglo de las k-ésimas capas del modelo de cada individuo, así como R es el vector con las recompensas estandarizadas, con el fin de ilustrar la Ecuación 3.8 muestra los arreglos W_k^A y R:

$$\begin{aligned} W_k^A &= [W_k^0, W_k^1, \ldots, W_k^P]. \\ R &= [r_1, r_2, \ldots, r_P]. \end{aligned} \tag{3.8}$$

De esta forma, los individuos que generen mayores recompensas serán aquellos que sesguen hacia esa dirección el modelo.
Ya que se actualizó los pesos del modelo W, se itera el proceso. Para fines de este estudio se realizaron 500 iteraciones sobre cada activo. De acuerdo a lo anterior y con los principales pasos descritos de la estrategia, se muestra el Algoritmo 6 así como el pseudocódigo de la estrategia evolutiva implementada.

Cabe mencionar que la función de recompensa es muy parecida a la utilizada en la evaluación retrospectiva.

Algoritmo: estrategia evolutiva.
Resultado: Modelo ajustado W
Entrada: ;
Arreglo con precio de cierre (C);
Matriz de estados (S);
Función de recompensa (F) ;
Tamaño de la población (P) ;
δ ;
Tasa de aprendizaje (α) ;
Número de iteraciones (T) ;
Inicialización:
Se inicializa el modelo W donde cada entrada del modelo se distribuye $\mathcal{N}(0,1)$;
Se inicializa el arreglo que contendrá a los modelos de cada individuo (W^A) ;
Se inicializa el arreglo que contendrá las recompensas de cada individuo (R) ;
Se inicializa $Score = -\infty$;
para $k \leftarrow 0$ **a** $T-1$ **hacer**
    para $i \leftarrow 0$ **a** $P-1$ **hacer**
        Se genera al individuo (i), es decir, se genera el Modelo W^i como se muestra en la Ecuación 3.5. ;
        Se obtiene la recompensa $r_i = F(S,C,W^i)$;
        Se guarda el Modelo W^i en el arreglo W^A ;
        Se guarda el valor de r_i en el arreglo R ;
    fin
    Se estandariza R ;
    Se actualiza W como se indica en 3.7. ;
    Se evalúa W, $N_Score = F(S,C,W)$;
    si $N_Score > Score$ **entonces**
        $Best_W = W$;
fin
$W = Best_W$
devolver W

Algoritmo 6: Algoritmo para entrenar los pesos del modelo mediante una estrategia evolutiva.

Una vez que se describió la metodología de este modelo, ahora se mostrarán algunos resultados para ejemplificar los datos resultantes de la ejecución. Para mantener la consistencia con el ejemplo de la red neuronal, se seguirá trabajando con la información del ticket de TSLA.

El entrenamiento se realizó con 500 iteraciones, $\alpha = 0{,}03$, $\delta = 0{,}1$ y una población de tamaño 30. En la Figura 3.12 se muestra el gráfico de convergencia del rendimiento contra el número de iteraciones, donde se observa como conforme aumenta el número de épocas

Algoritmo: Función de Recompensa.
Resultado: Recompensa (r)
Entrada: ;
Arreglo con precio de cierre (C);
Matriz de Estados (S);
Modelo (W);
Dinero inicial (D_0);
Inicialización:
Sea N el tamaño del arreglo de precios de cierre (C) ;
Se define el dinero final $D_F = D_0$;
Se inicializa el arreglo que contendrá los precios de compra B ;
Se inicializa $b = 0$;
para $k \leftarrow 0$ **a** $N-1$ **hacer**
    Sea a la acción dictada por el Modelo W dado el estado $S[k]$;
    si $a = Compra$ & $D_F > C[t]$ **entonces**
        Se agrega el precio $C[t]$ al arreglo B ;
        Se actualiza el dinero final, $D_F = D_F - C[t]$;
        Se actualiza $b = b+1$;
    si no, si $a = Venta$ & $b > 0$ **entonces**
        Se actualiza el dinero final, $D_F = D_F + C[t] * b$;
        Se actualiza $b = 0$;
fin
devolver $D_F/D_0 - 1$

Algoritmo 7: Algoritmo para obtener la recompensa dado un modelo específico.

el rendimiento del agente converge o se estabiliza hacia una solución que en el mejor de los casos sería óptima o cerca de óptima a medida que siga el entrenamiento.

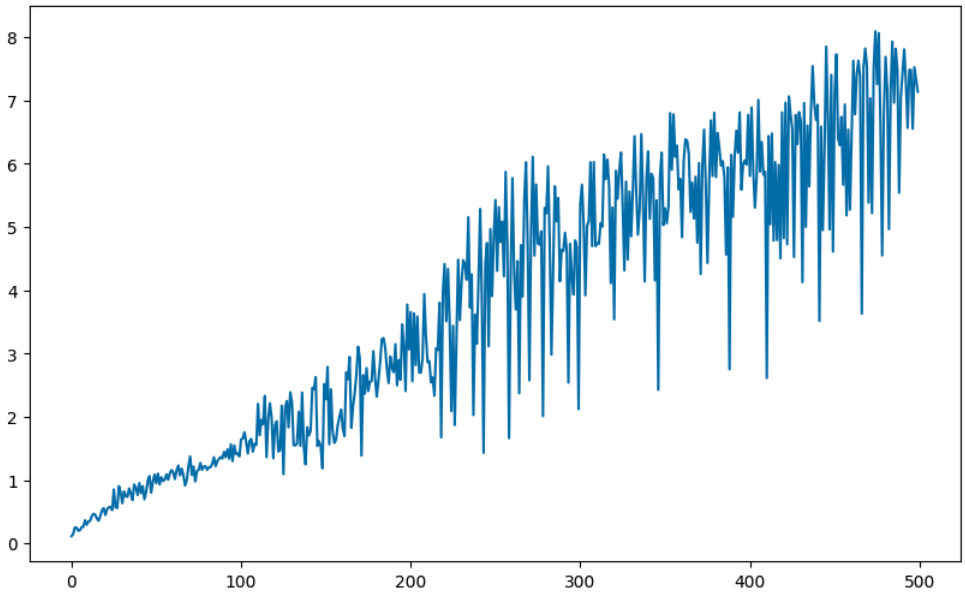

Figura 3.12: Gráfico de convergencia iteraciones vs rendimiento de la estrategia evolutiva, el gráfico de convergencia busca mostrar el progreso de un agente en términos de, por ejemplo, la recompensa acumulada, el error en la estimación de valor, entre otros, a lo largo del tiempo o episodios.

El gráfico de convergencia da la impresión que el modelo podría seguir mejorando, sin embargo, se observa que es errático. Además como se observa en la Tabla 3.10 la coincidencia es mínima con respecto a la estrategia propuesta por el etiquetado definido en la metodología Capítulo 3 de este proyecto.

	Precisión	Exhaustividad	Puntaje f1	Soporte
Venta	0.108553	0.452055	0.175066	73.000000
Mantener	0.839286	0.181118	0.297940	519.000000
Comprar	0.112108	0.531915	0.185185	47.000000
Exactitud	0.237872	0.237872	0.237872	0.237872
Promedio Macro	0.353315	0.388362	0.219397	639.000000
Promedio ponderado	0.702320	0.237872	0.275609	639.000000

Tabla 3.10: Resultados del modelo entrenado en el conjunto de prueba.

Por otro lado, en la Tabla 3.11 se observa que el agente logra un 2,51 % de ganancia y tan solo 2 acciones las cuales el precio comprado es inferior al precio último visto del activo.

En la siguiente subsección se menciona acerca de la modificación que se propuso en la estrategia evolutiva descrita anteriormente, así como un ejemplo de los resultados obtenidos para un activo.

Ganancia	Acciones restantes	Último precio comprado	% Ganado	Monto comprado	Último precio
882.43	2	173.44	2.51 %	49,749.1	184.30

Tabla 3.11: Resultados de la evaluación retrospectiva en el conjunto de prueba

3.5.4. Estrategia evolutiva modificada

La modificación que se propone a la estrategia evolutiva anteriormente descrita, se enfoca en la forma en la que se cálcula la recompensa.

Básicamente, lo que se propone a diferencia de la Ecuación 3.6, es utilizar durante el entrenamiento la variable objetivo generada en el prepocesamiento para recompensar al agente siempre y cuando haga acciones que coincidan con el etiquetamiento. Sin embargo, si el algoritmo logra un rendimiento superior al generado con el etiquetamiento no se le penalizará si este difiere, de esta forma la modificación a la recompensa se define en la Ecuación 3.9 :

$$r = \begin{pmatrix} D_f/D_i - 1, & \text{si } D_f/D_i - 1 \geq r_e \\ (D_f/D_i - 1) * F1_{ponderado} + F1_{ponderado} * 10, & \text{si } D_f/D_i - 1 > 0 \\ (D_f/D_i - 1) * F1_{ponderado} - F1_{ponderado} * 10, & \text{e. o. c.} \end{pmatrix}. \tag{3.9}$$

Donde D_f se refiere al dinero final resultante de la operación del agente, considerando el monto de las acciones compradas y que aún no han sido vendidas, D_i es el dinero inicial con el que el agente aperturó operaciones, r_e se refiere a la recompensa obtenida si se siguiera estrictamente el etiquetamiento, esta última calculada como se define en la Ecuación 3.6. $F1_{ponderado}$ corresponde al puntaje f1 obtenido entre la variable objetivo y las acciones realizadas por el agente, ponderado como se muestra en la Ecuación 3.10.

$$F1_{ponderado} = 0.4 * F1^{Compra} + 0.4 * F1^{Venta} + 0.2 * F1^{Mantener}. \tag{3.10}$$

Donde $F1^a$ se refiere al puntaje f1 correspondiente a la acción a.

Se muestra un ejemplo de los resultados generados por este algoritmo, comenzando por la Figura 3.13 donde se observa el comportamiento de la recompensa con cada iteración en el conjunto de entrenamiento. Hay que recordar que en este ejercicio se tiene la recompensa y el rendimiento obtenido por el agente.

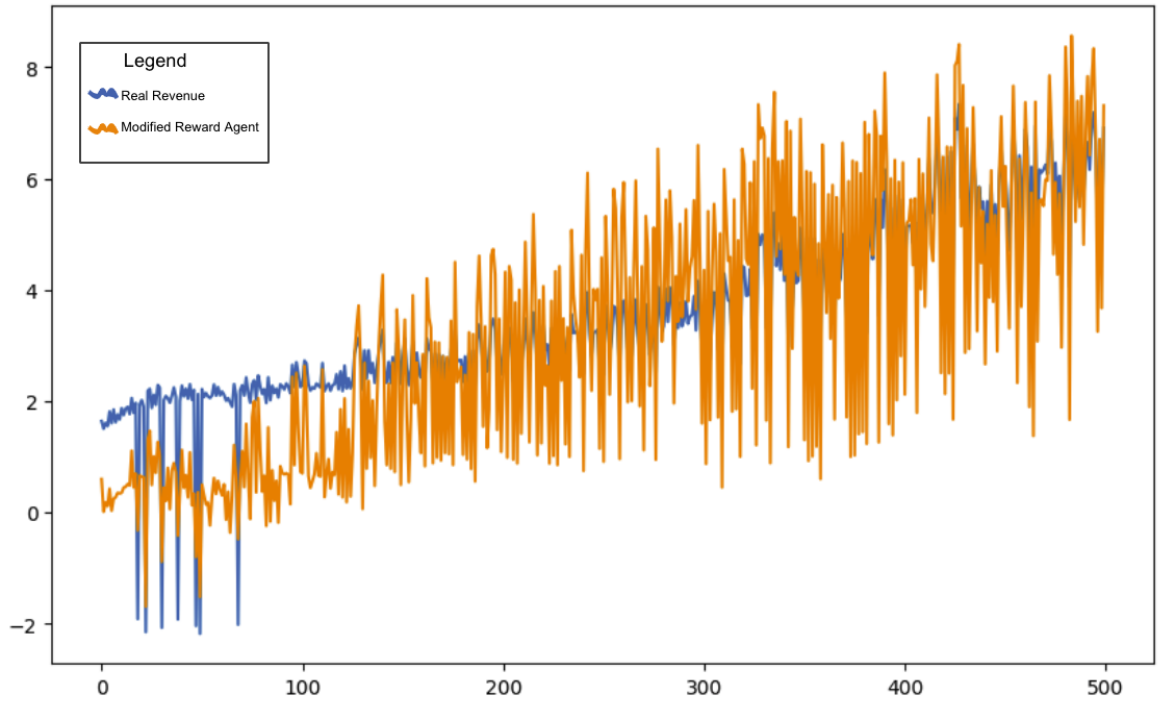

Figura 3.13: Gráfico de convergencia, iteraciones vs recompensa y rendimiento de estrategia evolutiva modificada.

Además como se observa en la Tabla 3.12 la coincidencia es mínima con respecto a la estrategia propuesta por el etiquetado.

	Precisión	Exhaustividad	Puntaje f1	Sporte
Vender	0.142	0.698	0.236	73
Mantener	0.448	0.049	0.089	521
Comprar	0.0357	0.173	0.059	46
Exactitud	0.132	0.132	0.132	0.13
Promedio Macro	0.208	0.307	0.128	640
Promedio ponderado	0.383	0.132	0.104	640

Tabla 3.12: Resultados del modelo entrenado en el conjunto de prueba.

Por otro lado, en la Tabla 3.13 se observa que el agente logra un 3,9 % de ganancia y 7 acciones las cuales si las vendiera hoy tendría una ganancia total de $1,969. Si comparamos este escenario con el logrado por la estrategia evolutiva original, debido a que hay aún 7 acciones por vender en la estrategia evolutiva modificada esto le permite tener una mayor oportunidad al momento en el que fue detenida la prueba, además de tener a ese momento una rentabilidad superior en un 1,4 % (contemplando las 7 acciones retenidas vendidas a ese momento).

Ganancia	Acciones Restantes	Último precio comprado	% Ganado	Monto comprado	Último precio
681.30	7	189.19	3.9 %	50,002.02	184.30

Tabla 3.13: Resultados de la evaluación retrospectiva.

Los ejemplos que se han abordado anteriormente sirven para ejemplificar como fueron tratados los datos, entrenados y evaluados los modelos. Lo anterior, con el fin de dar claridad sobre los procesos e indicadores de desempeño que se utilizaron para evaluar cada uno de modelos. Por otro lado, los resultados obtenidos por los algoritmos en términos de rendimiento se encuentran por debajo de la meta (11 %) y tan solo el puntaje f1 macro para la red neuronal se encuentra por encima del 80 %. Además, en los modelos de estrategia evolutiva se encuentra el puntaje f1 por debajo de 20 %. Es importante decir, que las arquitecturas de las redes así como los entrenamientos tienen diferentes características, donde por motivos de costo computacional y naturaleza de los modelos no fue posible homologarlos.

A continuación se aborda el marco de trabajo proporcionado por FinRL, así como algunos ejemplos de los resultados.

3.6. Financial Reinforcement Learning (FinRL)

En esta sección se describe el marco de trabajo brindado por FinRL [21], así como la forma en la que se utiliza. No sin antes brindar una breve introducción a esta paquetería.

FinRL es una paquetería de aprendizaje por refuerzo que facilita a los principiantes exponerse a las finanzas cuantitativas y desarrollar sus propias estrategias de negociación de acciones, mediante:

- Tutoriales fácilmente reproducibles.
- Optimizar desarrollos propios y compararlos fácilmente con los esquemas existentes.
- Entornos virtuales configurados a partir del mercado de valores que permite especificar costo de transacción, la liquidez del mercado y el grado de aversión al riesgo del inversor.
- Proporciona algoritmos de Aprendizaje Profundo por Refuerzo de última generación ajustados tales como: DQN, DDPG, PPO, SAC, A2C, TD3, entre otros.
- Además incorpora tres aplicaciones, negociación de acciones individuales, negociación de múltiples acciones y asignación de cartera.

Dadas las bondades de FinRL se propuso seguir la siguiente metodología para generar distintos experimentos que permitieron enfrentar este desafío con agentes de aprendizaje profundo por refuerzo.

Es importante mencionar que el preprocesamiento y el etiquetado vistos en las secciones, no se aplicaron en el marco de trabajo de FinRL, debido a que se desea probar este marco de trabajo sin ninguna modificación significativa en su ambiente y obtención de datos. Además que en los modelos vistos en las secciones no se hizo ningún trabajo de ajuste de hiperparámetros tarea que sí se realizó en este marco de trabajo con un enfoque de Optimización multi-objetivo donde se busca optimizar:

- Rendimiento acumulado.
- Volatilidad anual.
- Estabilidad.

A continuación, se detallará la metodología aplicada para la obtención de datos, preprocesamiento, entrenamiento, ajuste de hiperparámetros con enfoque multi-objetivo (NSGA-II) y evaluación de los modelos aplicados.

3.6.1. Obtención de Datos y Preprocesamiento

Para la obtención de datos, FinRL cuenta con un método de extracción con el cual, descarga los datos de Yahoo Finanzas.

Posteriormente parte del preprocesamiento implica definir cuáles son los indicadores que aportaran información al agente como parte del ambiente, para este estudio particular se incluye lo siguiente:

- MACD.
- Bandas de Bollinger.
- RSI.
- CCI.
- DX.
- SMA.

Dichos indicadores se generán para cada uno de los activos que se incluyan al portafolio, hay que recordar que FinRL permite entrenar en función de un portafolio con más de un activo.

3.6.2. Entrenamiento

FinRL permite realizar el entrenamiento a partir de la configuración de un ambiente, el cual, trata de simular las condiciones del mercado, los parámetros que se utilizaron fueron los siguientes:

- HMAX: 100.
- Dinero Inicial: 100,000.
- Número de activos disponibles: Depende del portafolio que haya confirgurado.
- Costo de Compra: 0.001.
- Costo de Venta: 0.001.
- Espacio de Estados.
- Dimensión de activos.
- Indicadores: Enlistados en la Subsección 4.5.1.
- Espacio de acciones.
- Escala de recompensa: 0.0001.

Los algoritmos que fueron entrenados durante 2,500 episodios fueron A2C y PPO, en su versión de comercio de multiples activos financieros disponible en FinRL, y los únicos hiperparámetros que fueron ajustados son

- A2C: Número de Pasos, Coeficiente de entropía y Tasa de Aprendizaje.
- PPO: Número de Pasos, Coeficiente de entropía ,Tasa de Aprendizaje y tamaño de Batch.

La forma en la que se ajustaron los hiperparámetros fue a través de NSGA-II donde cada uno de los hiperparámetros fue trabajado en los siguientes espacios de búsqueda, con el propósito de optimizar los objetivos: rendimiento acumulado , la volatilidad anualizada y la estabilidad.

- Número de pasos: [500,3000].
- Coeficiente de entropía: [0,0000001,0,1].
- Tasa de aprendizaje: [0,0001,0,1].
- Tamaño de Batch: Como un porcentaje [0,1,1] del número de pasos.

Los hiperparámetros de NSGA-II fueron:

- Generaciones: 100.

- Tamaño de la población: 20.
- Número de descendientes: 5.
- Cruza: SBX probabilidad de 0.9 y eta 15.
- Mutación: PM eta 20.

Y cada uno de estos experimentos se ejecutaron 20 veces con el fin de tener una muestra sobre las distintas fronteras de soluciones encontradas y con ellas realizar pruebas estadísticas.

Dicho lo anterior, en la Tabla 3.14 se muestra un ejemplo de los resultados de la implementación del marco de trabajo de FinRL con el modelo de A2C, un portafolio constituido solo por una acción: *TSLA*, con el fin de guardar consistencia con los otros ejemplos, donde por medio de NSGA-II se encuentran los hiperparámetros que optimizan los objetivos de retorno acumulado, volatilidad anualizada y estabilidad.

Objetivos	A2C	A2C Ajustado con NSGA-II
Retorno acumulado	-2.26 %	0 %
Volatilidad anualizada	22.11 %	0 %
Estabilidad	0	0

Tabla 3.14: Resultados de la evaluación retrospectiva A2C.

Como se observa después del ajuste de hiperparámetros, el algoritmo sugiere que es preferible no comerciar con ese activo ya que no encontró una solución que permitiera generar rendimientos.

Cabe mencionar, que esta solución fue tomada de la frontera con el hipervolumen más alto y fue la que mayor rendimiento acumulado alcanzaba. En la Tabla 3.15 se muestra el hipervolumen generado por cada frontera obtenida en las distintas iteraciones del experimento donde el punto de referencia fue [10,10,10].

Los parámetros correspondientes a esta solución encontrada son:

- Número de pasos : 1,259.
- Coeficiente de entropía: 0,000181.
- Tasa de Aprendizaje : 0,032132.

Además en la Figura 3.14 se muestra la frontera de la que se deriva esta solución.

Estos resultados y la forma en la que fueron tratados y analizados es el ejemplo de la metodología implementada y los criterios tomados para la selección de parámetros.

	Iteración	Hipervolumen	Semilla
19	19	1042.95	140
6	6	1042.89	49
12	12	1042.70	91
5	5	1042.51	42
8	8	1042.30	63
9	9	1042.30	70
1	1	1042.30	14
16	16	1042.30	119
13	13	1042.29	98
14	14	1042.29	105
10	10	1042.28	77
17	17	1042.27	126
7	7	1042.25	56
2	2	1041.18	21
18	18	1041.18	133
3	3	1040.53	28
15	15	1038.97	112
11	11	1038.79	84
0	0	1036.75	7
4	4	1014.75	35

Tabla 3.15: Hipervolumen generado por las fronteras encontradas en iteración de NSGA-II.

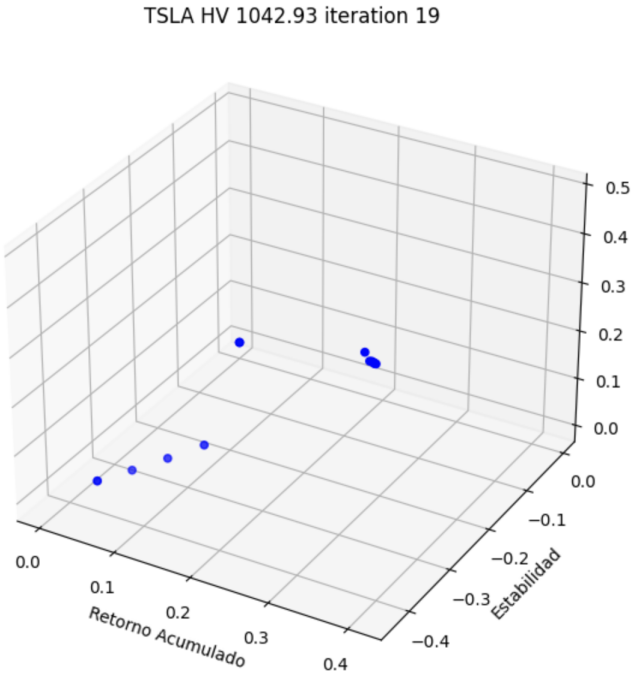

Figura 3.14: Frontera con el mayor hipervolumen encontrado.

Capítulo 4

Resultados

En este capítulo se mencionará acerca de los resultados de experimentar con 8 portafolios distintos conformadas por acciones de sectores comerciales específicos. Esto con el fin de comparar los modelos que se implementaron en este caso de estudio.

Los portafolios que se define para trabajar están basados en algunos de los sectores comerciales más importantes del mercado de valores y cada uno está conformado por al menos 8 activos de las empresas principales de cada sector. A continuación se muestran los resultados para cada uno de los modelos implementados de forma individual, y posteriormente se realizarán las comparaciones correspondientes.

4.1. Red neuronal

En la presente sección se mostrará los resultados de la red neuronal puesta aprueba en los 8 portafolios definidos para este proyecto, es importante destacar que se entrenó una red neuronal en cada uno de los activos que conforman al portafolio, y posteriormente se sumarizan los resultados para cada portafolio.

Este modelo utilizó para el entrenamiento y prueba al etiquetamiento propuesto en el Capítulo 3, Sección 3.3 Preprocesamiento, En la Tabla 4.1 se muestran los resultados sobre el conjunto prueba. Debido a que la variable objetivo está altamente imbalanceada para todos los escenarios, el puntaje f1 macro resulta la más conveniente de los datos como se observa los portafolios con el puntaje f1 macro más alto son minerales energéticos con (86.9 % puntaje f1 macro), Comercio al por menor (82.5 %) y Finanzas (82.5 %).

Sector	Macro precisión promedio	Macro recall promedio	Macro puntaje f1 promedio
Comunicación	0.772	0.795	0.755
Bienes de consumo duraderos	0.673	0.643	0.628
Bienes de consumo no duraderos	0.757	0.860	0.788
Servicios de consumo	0.765	0.835	0.789
Tecnologías electrónicas	0.786	0.860	0.812
Minerales energéticos	0.869	0.872	0.869
Finanzas	0.802	0.883	0.824
Comercio al por menor	0.795	0.877	0.825

Tabla 4.1: Resultados con respecto a la variable objetivo en el conjunto de prueba Red neuronal.

Por otro lado, la Tabla 4.2 presenta los rendimientos obtenidos bajo la evaluación retrospectiva que se describen en el Capítulo 3. Cabe destacar, que a pesar de que consumos duraderos tiene el macro puntaje f1 más bajo, el resultado en cuanto a retorno fue de los más altos con 12.71 % , esta situación es interesante ya que aunque el etiquetado propone una estrategia con un retorno atractivo, no significa que no haya otras soluciones si el objetivo es el rendimiento.

Sector	Valor total	Invertido	Retorno promedio	Retorno acumulado
Comunicación	605.71	20,532.06	6.50 %	2.95 %
B. consumo duraderos	3,001.45	23,599.27	10.89 %	12.71 %
B. consumo no duraderos	2,353.26	52,265.36	4.51 %	4.50 %
Servicios de consumo	19,223.77	249,507.64	8.14 %	7.70 %
Tecnologías electrónicas	6,308.59	65,888.46	9.21 %	9.57 %
Minerales energéticos	412.04	6,134.64	6.71 %	6.71 %
Finanzas	8,881.62	110,068.40	6.72 %	8.06 %
Comercio al por menor	10,863.68	222,403.98	5.59 %	4.88 %

Tabla 4.2: Resultados de la evaluación retrospectiva para el modelo de red neuronal.

El valor total se compone por la ganancia generada por el algoritmo más el valor en riesgo que es el valor de las acciones que aún no ha vendido el algoritmo al último precio observado. El Invertido se refiere al acumulado de las compras realizadas por el algoritmo durante la evaluación retrospectiva, el retorno promedio se refiere al promedio de los activo del portafolio y el retorno acumulado no es más que la proporción del Valor Total con respecto al monto invertido. Dado este breve detalle sobre la Tabla 4.2 es importante resaltar que las estrategias propuestas por la red neuronal logran aprender un comportamiento que aunque no es óptimo libera a cualquiera de los portafolios de la pérdida siendo el retorno acumulado en el sector de comunicación el más bajo con 2.95 %.

En la siguiente sección se mostrará los resultados generados por el modelo de estrategia evolutiva implementado.

4.2. Estrategia evolutiva (ES)

En la presente sección se mostrará los resultados de la estrategia evolutiva que se puso a aprueba en los 8 portafolios definidos en este proyecto, cabe mencionar que el entrenamiento de este modelo se realizó de forma independiente con cada uno de los activos que conforman al portafolio, y posteriormente se resumen los resultados para cada portafolio.

Este modelo no toma en cuenta el etiquetamiento propuesto en el Capítulo 4, Sección 4.3 Preprocesamiento, sin embargo, en la Tabla 4.3 se compara la predicción del modelo contra el etiquetado propuesto donde se observa que esta estrategia no es comparable contra la propuesta por el etiquetado automático y tiene muy pocas coincidencias a tal grado que el macro puntaje f1 para cualquiera de los portafolios no sobrepasa el 0.23.

Sector	Macro precisión Promedio	Macro recall Promedio	Macro puntaje f1 Promedio
Comunicación	0.350	0.376	0.194
B. consumo duraderos	0.354	0.441	0.191
B. consumo no duraderos	0.347	0.375	0.164
Servicios de consumo	0.323	0.399	0.182
Tecnologías electrónicas	0.348	0.420	0.191
Minerales energéticos	0.344	0.373	0.238
Finanzas	0.351	0.404	0.175
Comercio al por menor	0.336	0.395	0.174

Tabla 4.3: Resultados con respecto a la variable objetivo en el conjunto de prueba para estrategia evolutiva.

Por otro lado, la Tabla 4.4 presenta los rendimientos obtenidos bajo la evaluación retrospectiva que se describen en el Capítulo 3, siendo el portafolio de minerales energéticos quien obtiene el retorno acumulado más alto 8.73 %.

Sector	Valor total	Invertido	Retorno promedio	Retorno acumulado
Comunicación	650.91	31,279.70	10.16 %	2.08 %
B. consumo duraderos	1,657.29	60,192.42	-5.39 %	2.75 %
B. consumo no duraderos	2,619.19	121,227.3	2.25 %	2.16 %
Servicios de consumo	54,252.59	1,142,538	3.72 %	4.74 %
Tecnologías electrónicas	3,869.38	141,541.2	3.19 %	2.73 %
Minerales energéticos	7,711.78	882,928.6	4.98 %	8.73 %
Finanzas	6,019.56	234,338.1	2.84 %	2.56 %
Comercio al por menor	53,490.28	848,425.5	3.88 %	6.30 %

Tabla 4.4: Resultados de la evaluación retrospectiva para el modelo de estrategia evolutiva.

4.3. Estrategia evolutiva modificada (ESM)

En la presente sección se mostrará los resultados de la estrategia evolutiva modificada, la cual, fue probada en los 8 portafolios definidos para este proyecto, es importante mencionar que el entrenamiento de este modelo se realizó de forma independiente con cada uno de los activos que conforman al portafolio, y posteriormente se resumen los resultados para cada portafolio.

Este modelo toma en cuenta el etiquetamiento propuesto en el Capítulo 3 para el cálculo de la recompensa, así que, en la Tabla 4.5 se compara la predicción del modelo contra el etiquetado propuesto donde se observa que a pesar de utilizar el etiquetamiento para ponderar la recompensa del agente, no logra alcanzar un puntaje f1 macro sobresaliente al menos en el número de iteraciones con las cuales se experimentó.

Sector	Macro precisión Promedio	Macro recall Promedio	Macro puntaje f1 Promedio
Comunicación	0.271	0.300	0.128
B. consumo duraderos	0.274	0.333	0.153
B. consumo no duraderos	0.339	0.351	0.157
Servicios de consumo	0.310	0.316	0.172
Tecnologías electrónicas	0.307	0.339	0.160
Minerales energéticos	0.251	0.246	0.127
Finanzas	0.303	0.276	0.129
Comercio al por menor	0.351	0.382	0.159

Tabla 4.5: Resultados con respecto a la variable objetivo en el conjunto de prueba para estrategia evolutiva modificada.

Por otro lado, la Tabla 4.6 presenta los rendimientos obtenidos bajo la evaluación retrospectiva que se describen en el Capítulo 3 donde se observa que el retorno acumulado más alto lo obtiene en el portafolio de comercio al por menor con 9.58 %.

Sector	Valor Total	Invertido	Retorno promedio	Retorno acumulado
Comunicación	277.12	37,228.62	-2.98 %	0.74 %
B. consumo duraderos	2,300.49	69,967.31	-0.15 %	3.28 %
B. consumo no duraderos	3,370.87	125,700.64	2.81 %	2.68 %
Servicios de consumo	20,500.08	772,782.36	2.91 %	2.65 %
Tecnologías electrónicas	4,324.12	180,662.31	2.69 %	2.39 %
Minerales energéticos	4,148.94	106,346.84	3.47 %	3.90 %
Finanzas	11,281.61	281,873.00	3.23 %	4.00 %
Comercio al por menor	76,374.10	796,769.03	4.02 %	9.58 %

Tabla 4.6: Resultados de la evaluación retrospectiva para el modelo de estrategia evolutiva modificada.

4.4. FinRL

En la presente sección se aborda los resultados del entrenamiento y la evaluación retrospectiva implementado bajo el marco de trabajo de FinRL sobre los modelos con los que se experimentó: A2C y PPO.

4.4.1. A2C

Se abordan los resultados del hipervolumen obtenido de las fronteras resultantes en cada uno de los 10 experimentos con NSGA-II para cada portafolio, siendo la Tabla 4.7 donde se presenta el hipervolumen (HV) promedio obtenido de cada frontera tomando como punto de referencia el $[1,1,1]$, la desviación estándar del hipervolumen y la mediana del mismo, donde se observa la mayor mediana del hipervolumen son 4.094 y se encuentra en el portafolio de minerales energéticos.

Portafolio	HV Promedio	Desv. Estandar HV	Mediana HV
Comunicación	1.891180	0.087925	1.887791
B. consumo duraderos	1.063414	0.011549	1.057970
B. consumo no duraderos	2.336192	0.045713	2.322972
Servicios de consumo	2.063026	0.072456	2.090768
Tecnologías electrónicas	1.795944	0.116269	1.821639
Minerales energéticos	4.118165	0.073539	4.094504
Finanzas	1.823735	0.124473	1.889599
Comercio al por menor	3.357731	0.078996	3.352358

Tabla 4.7: Resumen del hipervolumen generado en cada portafolio con A2C.

Por otro lado, en la Figura 4.1 se muestra como luce la frontera de mayor hipervolumen para cada uno de estos portafolios.

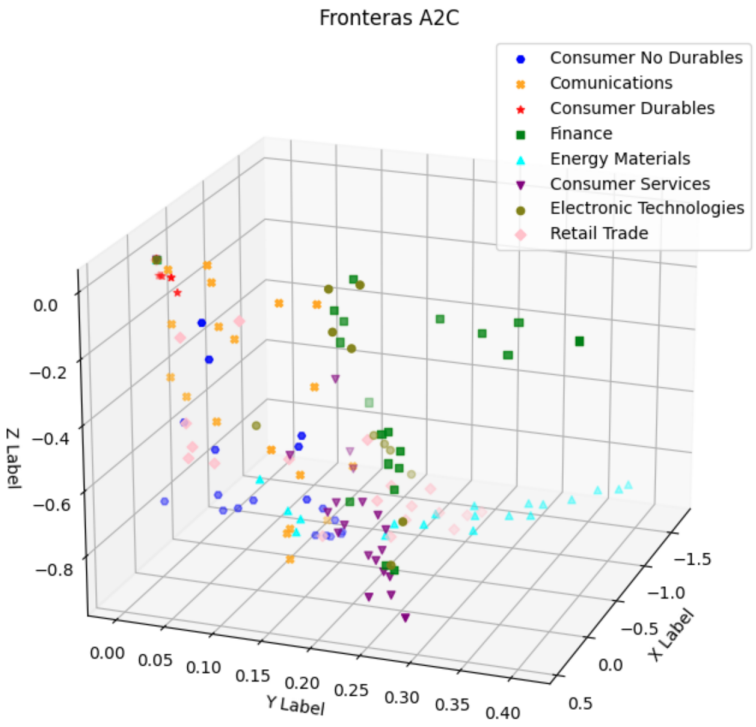

Figura 4.1: Fronteras de mayor hipervolumen para cada uno de los portafolios con A2C.

Una vez que se conocen las fronteras en función a su hipervolumen como se muestra en la Tabla 4.7 y se visualiza la distribución en la Figura 4.1, se seleccionan estas fronteras, así como las respuestas que contengan al retorno acumulado más alto que se muestra en la Tabla 4.8 a fin de comparar bajo el mismo criterio, siendo minerales energéticos, seguido de comercio minorista los portafolios con mayor retorno acumulado 171 % y 108 % respectivamente.

Portafolio	# Pasos	C. entropía	T. Aprend.	# Iter.	Ret. Acum.	Vol. An.	Estab.
Comunicación	1464	0.051	0.097	10	35.06 %	0.164	0.84
B. Con. no D.	1070	0.089	0.026	7	18.34 %	0.187	0.59
B. Con. D.	621	0.028	0.072	2	00.00 %	0.000	0.00
Serv. de Con.	738	0.045	0.040	0	35.51 %	0.343	0.13
Min. Energ.	1428	0.099	0.035	9	171.25 %	0.380	0.89
Tec. Elect.	1918	0.038	0.033	8	33.11 %	0.173	0.58
Finanzas	2301	0.020	0.005	7	30.31 %	0.239	0.62
Com. min.	1893	0.029	0.000	5	108.40 %	0.261	0.90

Tabla 4.8: Resultado de la frontera con mayor hipervolumen y mayor retorno acumulado obtenido para A2C.

4.4.2. PPO

En la presente subsección se muestran los resultados del hipervolumen obtenido de las fronteras resultantes en cada uno de los 10 experimentos con NSGA-II para cada portafolio, siendo la Tabla 4.9 donde se presenta el hipervolumen promedio obtenido de cada frontera tomando como punto de referencia el $[1,1,1]$, la desviación estándar del hipervolumen y la mediana del mismo. Dicha Tabla muestra que la mayor mediana del hipervolumen son 4.14 y la se encuentra en el portafolio de minerales energéticos, seguido por un hipervolumen de 3.49 de comercio al por menor.

Portafolio	HV Promedio	Desv. Estandar HV	Mediana HV
Comunicación	1.79	0.034	1.78
B. consumo duraderos	1.05	0.000	1.05
B. consumo no duraderos	2.41	0.022	2.41
Servicios de consumo	2.24	0.085	2.24
Tecnologías electrónicas	1.99	0.128	1.96
Minerales energéticos	4.17	0.130	4.14
Finanzas	2.07	0.079	2.04
Comercio al por menor	3.49	0.022	3.49

Tabla 4.9: Resumen del hipervolumen generado en cada portafolio con PPO.

Complementando, en la Figura 4.2 se muestra la distribución de los puntos de la frontera de mayor hipervolumen para cada uno de estos portafolios.

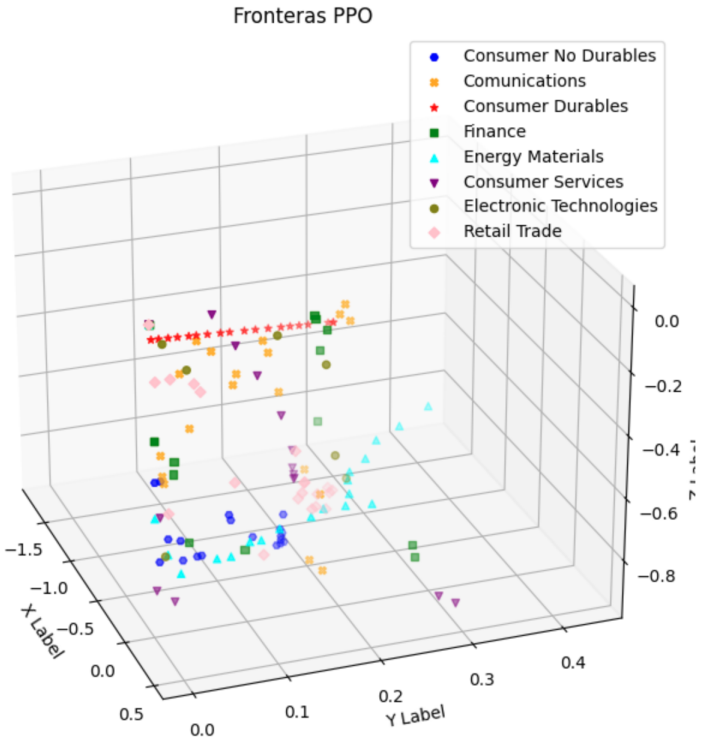

Figura 4.2: Fronteras de mayor hipervolumen para cada uno de los portafolios con PPO.

Con las fronteras de mayor hipervolumen se considera solo la respuesta que contengan al retorno acumulado más alto y se muestran en la Tabla 4.10 para cada portafolio, junto a los hiperparámetros encontrados por NSGA-II, cabe resaltar en este punto que los minerales energéticos muestran el retorno acumulado más alto con 178 % seguido de comerio al por menor con 107 %. Por otro lado, el portafolio con menor rendimiento es bienes de consumo duradero donde se obtuvo 0 %.

Portafolio	# Pasos	C. entropía	T. Aprend.	# Iter.	Ret. Acum.	Vol. An.	Estab.
Comunicación	2461	0.031	0.057	52.7 %	39.53 %	0.165	0.83
B. Con. no D.	817	0.098	0.021	59.4 %	12.70 %	0.220	0.11
B. Con. D.	690	0.077	0.043	65.1 %	00.00 %	0.000	0.00
Serv. de Con.	879	0.051	0.039	89.7 %	43.57 %	0.207	0.49
Min. Energ.	1326	0.006	0.090	39.0 %	178.09 %	0.433	0.87
Tec. Elect.	721	0.019	0.019	30.7 %	37.15 %	0.175	0.61
Finanzas	1437	0.054	0.022	75.4 %	47.00 %	0.241	0.70
Com. min.	2646	0.097	0.007	33.7 %	107.60 %	0.263	0.90

Tabla 4.10: Resultado de la frontera con mayor hipervolumen y mayor retorno acumulado obtenido para PPO.

4.4.3. DDPG

En la presente subsección se muestran los resultados del hipervolumen obtenido de las fronteras resultantes en cada uno de los 10 experimentos con NSGA-II para cada portafolio, siendo la Tabla 4.9 donde se presenta el hipervolumen promedio obtenido de cada frontera tomando como punto de referencia el $[1,1,1]$, la desviación estándar del hipervolumen y la mediana del mismo. En esta misma Tabla se muestra que la mayor mediana del hipervolumen son 3.94 que se encuentra en el portafolio de minerales energéticos, seguido por un hipervolumen de 3.25 del comercio al por menor.

Portafolio	HV Promedio	Desv. Estandar HV	Mediana HV
Comunicación	1.615	0.006	1.61
B. consumo duraderos	1.044	0.000	1.04
B. consumo no duraderos	2.266	0.002	2.26
Servicios de consumo	2.077	0.027	2.08
Tecnologías electrónicas	1.746	0.076	1.75
Minerales energéticos	3.936	0.159	3.94
Finanzas	1.742	0.081	1.76
Comercio al por menor	3.257	0.027	3.25

Tabla 4.11: Resumen del hipervolumen generado en cada portafolio con DDPG.

Por otro lado, en la Figura 4.3 se muestra la frontera de mayor hipervolumen para cada uno de estos portafolios.

Con las fronteras de mayor hipervolumen se considera solo la respuesta que contenga al retorno acumulado más alto y las se muestra en la Tabla 4.10 para cada portafolio, junto a los hiperparámetros encontrados por NSGA-II. Se observa en esta Tabla que minerales energéticos muestran el retorno acumulado más alto con 173 % seguido de comerio al por menor con 103 %. Sin embargo, no todos los portafolios tienen un comportamiento sobresaliente como es el caso de bienes al consumo duradero donde se obtuvo 0 %.

Portafolio	Batch Size	% T. aprendizaje	# Iter	Ret. Acum.	Vol. An.	Estab.
Com.	1105	0.58 %	4	13.63 %	0.2289	0.087
B. Con. D.	1344	6.11 %	10	00.00 %	0.0000	0.000
B. Con. no D.	530	7.24 %	2	35.43 %	0.1684	0.845
Serv. de Con.	697	0.65 %	0	31.99 %	0.1749	0.629
Tec. Elect.	795	4.75 %	2	27.77 %	0.2445	0.608
Min. Energ.	799	0.11 %	1	173.81 %	0.3983	0.877
Finanzas	596	0.47 %	5	31.76 %	0.3151	0.135
Com. min.	1186	0.97 %	10	103.56 %	0.2686	0.900

Tabla 4.12: Resultado de la frontera con mayor hipervolumen y mayor retorno acumulado obtenido para DDPG.

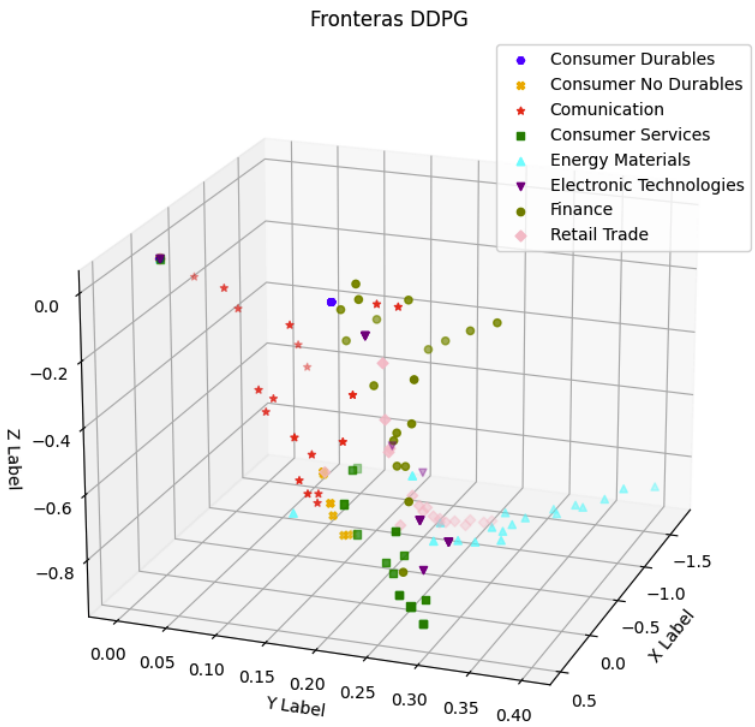

Figura 4.3: Fronteras de mayor hipervolumen para cada uno de los portafolios con DDPG.

4.4.4. Comparación A2C, PPO, DDPG

Acontinuación en la Tabla 4.13 se comparan los resultados de los algoritmos implementados bajo el marco de trabajo de FinRL y ajustados mediante un enfoque multi-objetivo mediante NSGA-II, siendo el retorno acumulado, la estabilidad y la volatilidad anualizada las metas. En esta comparación una de las situaciones más interesantes es que los tres algoritmos deciden no comerciar con los activos del portafolio de Bienes al consumo duradero, definiendo esta como la mejor estrategia. Por otro lado, todos los demás portafolios consiguen rendimientos superiores al 12.7 %, el cuál ya es superior al objetivo de 11 % de inflación actual. Por otro lado, la volatilidad más alta se observa en el portafolio de minerales energéticos esto tiene sentido debido a que tiene el rendimiento más elevado.

Portafolio	Retorno acumulado			Volatilidad An.			Estabilidad		
	A2C	PPO	DDPG	A2C	PPO	DDPG	A2C	PPO	DDPG
B. Con. no D.	35.06 %	39.53 %	35.43 %	0.164	0.165	0.168	0.84	0.83	0.84
Com.	18.34 %	12.70 %	13.63 %	0.187	0.220	0.228	0.59	0.11	0.08
B. Con. D.	00.00 %	00.00 %	00.00 %	0.000	0.000	0.000	0.00	0.00	0.00
Serv. de Con.	35.51 %	43.57 %	31.99 %	0.343	0.207	0.174	0.13	0.49	0.62
Min. Energ.	171.25 %	178.09 %	173.81 %	0.380	0.433	0.398	0.89	0.87	0.87
Tec. Elect.	33.11 %	37.15 %	27.77 %	0.173	0.175	0.244	0.58	0.61	0.60
Finanzas	30.31 %	47.00 %	31.76 %	0.239	0.241	0.315	0.62	0.70	0.13
Com. min.	108.40 %	107.60 %	103.56 %	0.261	0.263	0.268	0.90	0.90	0.90

Tabla 4.13: Comparación de los resultados de A2C, PPO y DDPG, se toma el resultado con mayor retorno acumulado de la frontera con mayor hipervolumen.

En la Figura 4.4 reducimos la comparación de la Tabla 4.13 a solamente el retorno acumulado, con el fin de ilustrar las diferencias entre estos dos algoritmos implementados.

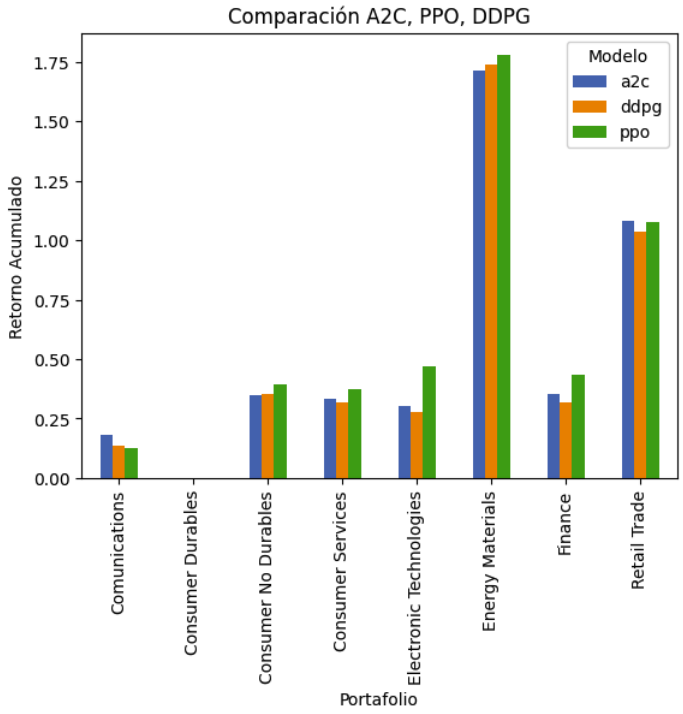

Figura 4.4: Comparación del retorno acumulado alcanzado por los algoritmos de A2C, PPO y DDPG en cada uno de los portafolios, se toma el resultado con mayor retorno acumulado de la frontera con mayor hipervolumen.

Como se puede observar en la Figura 4.4 en términos de rendimiento acumulado el algoritmo PPO encuentra un rendimiento mayor en los portafolios de bienes de consumo no duraderos, servicios de consumo, tecnologías electrónicas, minerales energéticos, finanzas, las cualidades de PPO que le permiten tener cierta ventaja se atribuyen principalmente a los siguientes puntos:

- puede aprender políticas efectivas con menos datos en comparación con otros algoritmos. Esto es crucial en el comercio algorítmico donde cada muestra puede ser costosa.
- Gracias a la “restricción de la razón de probabilidad recortada"(clipped probability ratio), PPO logra mantener las actualizaciones de la política dentro de un rango predefinido, evitando cambios bruscos que podrían ser perjudiciales para el aprendizaje. Esto ayuda a evitar actualizaciones de políticas que podrían llevar a una caída en el rendimiento, algo particularmente importante en mercados financieros donde decisiones erráticas pueden ser muy costosas.
- PPO puede manejar eficientemente espacios de acciones continuas, lo que es útil en el comercio donde las decisiones pueden incluir cuánto comprar o vender de un activo, en lugar de simplemente si comprar o vender [49].

4.5. Comparación General

Los algoritmos implementados comparten un indicador de desempeño en común que es el retorno acumulado, este indicador brinda una idea de como se comporta cada uno de los algoritmos y bajo este enfoque se comparan en la siguiente Tabla 4.14.

Portafolio	Red Neuronal	ES	ESM	A2C	PPO	DDPG
Comunicación	2.95	2.08	0.74	18.34	12.70	13.63
Bienes cons. duradero	12.71	2.75	3.28	00.00	00.00	00.00
Bienes cons. no duradero	4.50	2.16	2.68	35.51	39.53	35.43
Servicios de consumo	7.70	4.74	2.65	33.11	37.15	31.99
Tecnologías electrónicas	9.57	2.73	2.39	30.31	47.00	27.77
Minerales energéticos	6.71	8.73	3.90	171.25	178.09	173.81
Finanzas	8.06	2.56	4.00	35	43.57	31.76
Comercio minorista	4.88	6.30	9.58	108.4	107.60	103.56

Tabla 4.14: Comparación del porcentaje del retorno acumulado de los algoritmos implementados.

Basados en el retorno acumulado se observa en la Figura 4.5 y la Tabla 4.14 que (como era de esperarse) no hay un algoritmo que domine el problema de comercio algorítmico en todos los portafolios con los que se experimentó. Sin embargo, cabe resaltar que en cada portafolio hay al menos un algoritmo que supera la meta del rendimiento superior al 11 %.

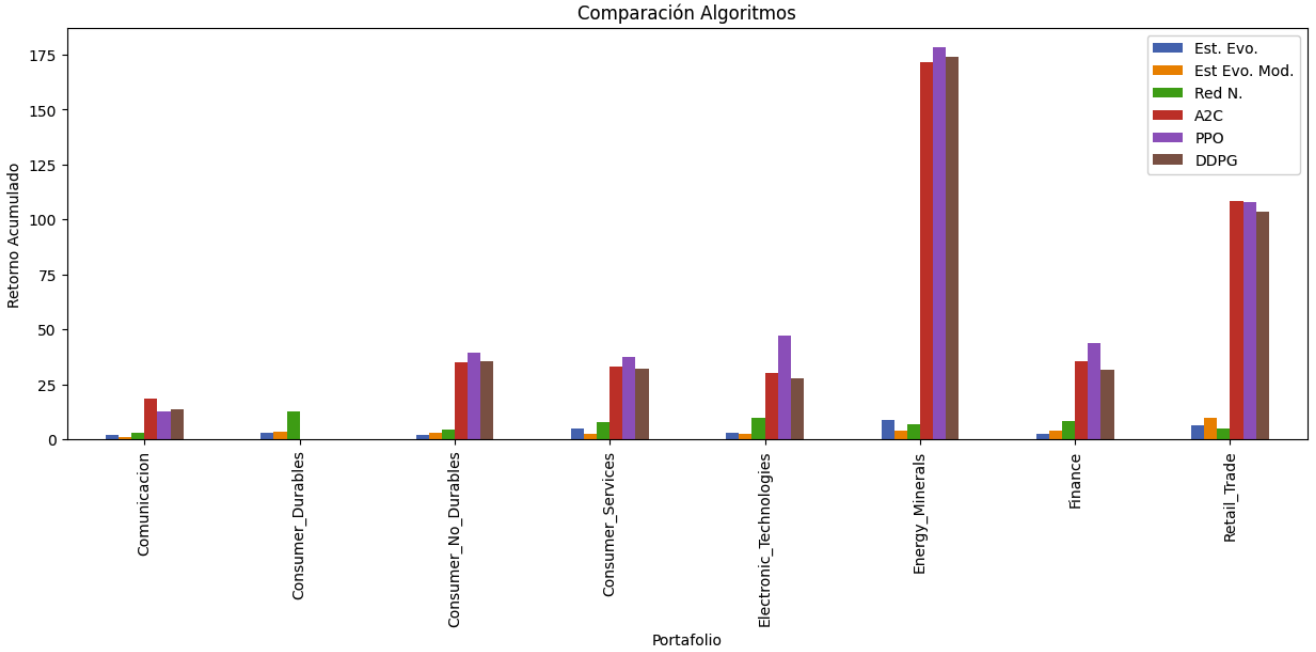

Figura 4.5: Comparación de algoritmos por retorno acumulado.

Capítulo 5

Discusión

Con el fin de dar mayor claridad al trabajo realizado en esta tesis, consta de 3 fases:

1. La primera donde expone una metodología para trabajar con los datos financieros prácticamente desde cero, encontrar patrones y señales a partir de indicadores comúnmente usados en el ámbito del comercio algorítmico, para después lograr proponer una forma automática de etiquetar momentos de compra y venta. Este etiquetado automático es utilizado como variable objetivo para entrenar una red neuronal (siguiendo el enfoque del aprendizaje supervisado).

   Por otro lado, se excluye el etiquetamiento y con la base de datos se prueba una estrategia evolutiva inicial (siguiendo el enfoque de Estrategias de Evolución Natural (NES)), la cual, después de analizarla se propone realizar una modificación que retome el etiquetamiento anteriormente propuesto, con el fin de encaminar las recompensas apartir de un conocimiento previo que no busca perturbar la exploración natural del agente. Cabe mencionar que la modificación propuesta muestra resultados similares y en algunos escenarios logra superar a la estrategia evolutiva original, es por ello que resulta una alternativa interesante.

   Considerando el retorno acumulado los modelos utilizados de red neuronal, estrategia evolutiva y la estrategia evolutiva modificada cada uno consigue resultados superiores en los portafolios de:

   - Red neuronal en comunicación (2.95 % retorno acumulado), bienes de consumo duraderos (12.71 %), bienes de consumo no duraderos (4.5 %), servicios de consumo (7.7 %) y tecnologías electrónicas (9.57 %).
   - Estrategia evolutiva en minerales energéticos (8.73 %).
   - Estrategia evolutiva modificada en comercio al por menor (9.58 %).

   Cabe mencionar que todos los valores entre paréntesis hacen referencia al retorno acumulado alcanzado en ese portafolio por el método en cuestión. Por otro lado, se muestra en la Figura 5.1 una comparación entre la estrategia evolutiva y la modificación propuesta donde en los siguientes portafolios es superior:

- Estrategia evolutiva en comunicación (2.08 %), servicios de consumo (4.74 %), minerales energéticos (8.73 %) y tecnologías electrónicas (2.73 %).
- Estrategia evolutiva modificada en bienes de consumo duraderos (3.28 %), bienes de consumo no duraderos (2.68 %), finanzas (4 %) y comercio al por menor (9.58 %).

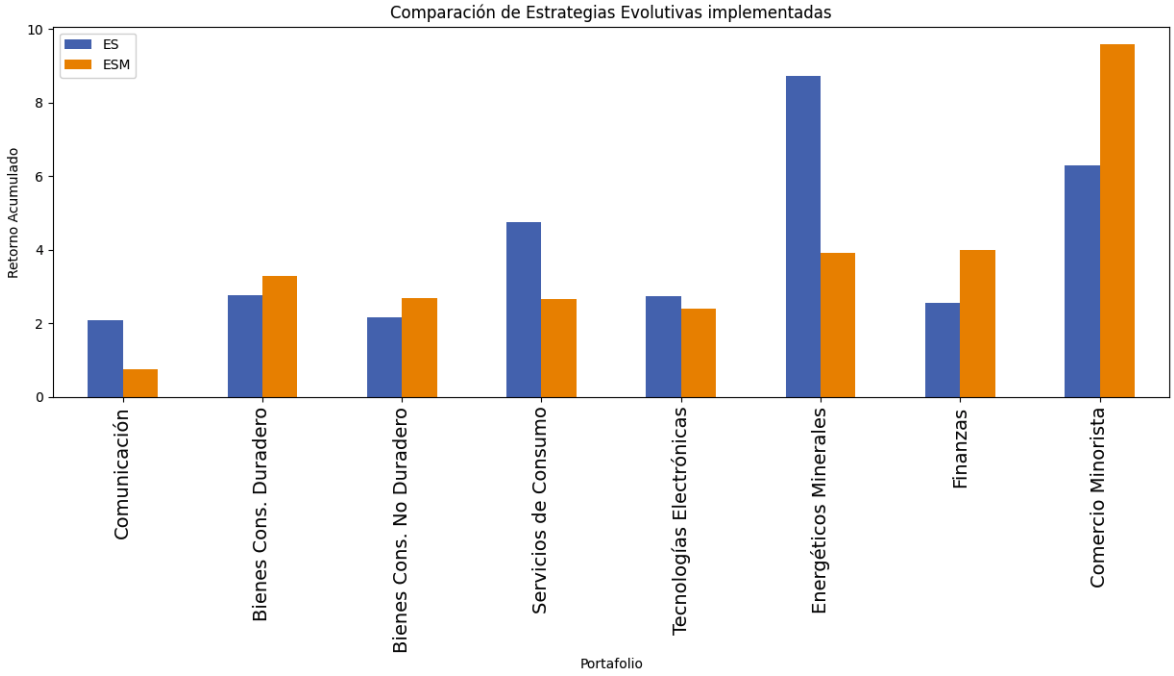

Figura 5.1: Comparación por retorno acumulado de las estrategias evolutivas implementadas.

Se observa que el integrar la información del etiquetamiento automatizado a la recompensa que recibe el agente le otorga cierta ventaja con algunos portafolios.

De igual manera este etiquetamiento, así como la forma en la que se realizó el preprocesamiento de los datos le brinda cierta superioridad a la red neuronal debido a que el optimizador ADAM le permite converger a los pesos de manera más rápida, lo que a la estrategia evolutiva le tomaría muchas generaciones.

2. En la segunda fase, se adopta un marco de trabajo a nivel del estado del arte para el aprendizaje por refuerzo. En este marco, se aborda el problema del comercio algorítmico utilizando los algoritmos A2C, PPO y DDPG. Posteriormente, se llevó a cabo un ajuste de hiperparámetros de estas versiones básicas en FinRL. Para ello, se empleó un enfoque multi-objetivo. Aunque este enfoque resulta computacionalmente costoso, los resultados obtenidos justifican el esfuerzo. Las fronteras identificadas por NSGA-II ofrecen un panorama completo de opciones, permitiendo definir un perfil de riesgo para un usuario final.

   Los resultados encontrados mediante NSGA-II para los algoritmos de A2C, PPO y DDPG permiten bajo la combinación de los criterios de retorno acumulado, volatilidad

anual y estabilidad al seleccionar algún algoritmo que se adecúe a necesidades más específicas, en este caso, la fase 1 se basa en la comparación pura al obtener el mayor retorno acumulado, como se observa en la Figura 5.2, también se observa que los siguientes algoritmos alcanzan la superioridad en los siguientes portafolios:

- PPO en bienes de consumo no duraderos (39.53 %), Servicios de consumo (37.15 %), Tecnologías electrónicas (37.15 %), Finanzas (43 %) y minerales energéticos (178.09 %).
- A2C en comunicación (18.34 %), y Comercio al por menor (108.4 %).
- DDPG no logra vencer en ninguno de los portafolios propuestos a los algoritmos implementados en este proyecto.

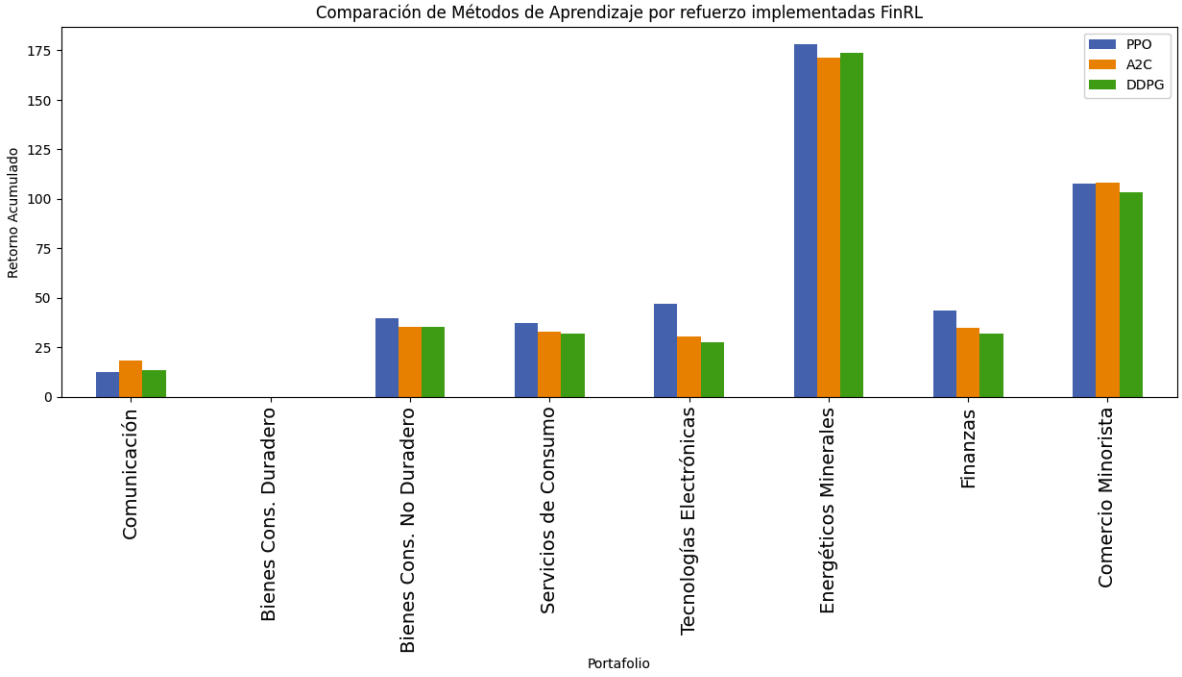

Figura 5.2: Comparación por retorno acumulado de los métodos de aprendizaje por refuerzo implementados en el marco de trabajo de FinRL.

Lamentablemente, para el portafolio de bienes de consumo duradero, los tres algoritmos sugieren no comerciar, lo que impide encontrar un parámetro bajo el cual la comparación implique una ganancia, este comportamiento se le atribuye a la forma en la que se entrenan los algoritmos dado que buscan una política que maximice una recompensa, en el escenario donde no se encuentra una estrategia que permita un rendimiento positivo, la estrategia de no comerciar es aquella que optimiza el riesgo (dado que el riesgo al no comerciar es 0), y al no encontrar una estrategia con rendimiento positivo, no comerciar maximiza el rendimiento (al ser 0).

3. En la tercera fase, el objetivo fue comparar los seis algoritmos implementados como se muestre en la Tabla 4.14 y Figura 4.5, teniendo en cuenta sus especificaciones y características únicas. Se sometieron a una prueba con los diferentes portafolios definidos,

que se puede apreciar la versatilidad de FinRL mostrando una notable adaptabilidad en la mayoría de los portafolios, incluso utilizando sus versiones más básicas de cada algoritmo. Cabe destacar la importancia de ajustar los hiperparámetros y cómo el enfoque multi-objetivo ofrece una perspectiva más versátil y amplia en cuanto a la dirección de la optimización. Al comparar estos algoritmos, queda en evidencia el dominio de la metodología de FinRL. Su habilidad para configurar un entorno que emule de manera fiel un ambiente real de comercio le proporciona la ventaja necesaria para destacar como un método del estado del arte.

Limitaciones

En esta sección se discuten los principales problemas y limitaciones que enfrenta este proyecto.

Inicialmente, se debe considerar que las condiciones de entrenamiento y las arquitecturas de las redes neuronales varían entre los algoritmos propuestos. Se optó por mantener cada algoritmo en su configuración básica o predeterminada. Esta decisión implica que la comparación entre algoritmos es posible únicamente a nivel de la tarea específica que realizan, es decir, la creación de una estrategia de compra y venta de activos financieros, aunque cada uno opera desde diferentes perspectivas y marcos de trabajo.

Si se profundiza en estas perspectivas y marcos de trabajo, es claro que el enfoque 2, basado en un modelo altamente reconocido de aprendizaje por refuerzo, presentado en NEURIPS en 2019, difiere significativamente del Enfoque 1, el cual ha sido diseñado e implementado específicamente para este proyecto. Podríamos comparar esta situación con la de un aprendiz que, utilizando señales construidas a partir de indicadores básicos, entrena algoritmos en sus configuraciones elementales y los confronta con un sistema experto en aprendizaje por refuerzo financiero. Esta comparación subraya la reflexión necesaria entre un método tradicional y manual de preparación de datos frente a la creación de entornos que simulan comportamientos del mercado real.

Como se menciona en los objetivos del proyecto el indicador de desempeño es el retorno acumulado o ganancia acumulada. Sin embargo, para efectos de su comparación fue considerado la estrategia o resultado que presentó el máximo rendimiento encontrado, lo cual, no es relevante en significancia estadística o incluso estadística descriptiva, a pesar de que este resultado tiene relavancia ya que fue observado en un conjunto de datos de prueba.

El proyecto enfrenta desafíos significativos relacionados con las condiciones de entrenamiento y las arquitecturas de red entre los algoritmos propuestos. Además de la diferencia entre los enfoques adoptados que conlleva a una reflexión crítica sobre la metodología de preparación de datos y la simulación de entornos de mercado. Asimismo, la metodología para evaluar el desempeño, sin considerar su significancia estadística, plantea cuestiones sobre la robustez de los resultados. Es esencial reconocer que, aunque la estrategia de máximo rendimiento se identificó dentro de un conjunto de prueba, esto no necesariamente se puede correlacionar con una superioridad metodológica o práctica.

Capítulo 6

Conclusión y trabajo futuro

Este proyecto ha recorrido un camino analítico y práctico a través del complejo panorama del comercio algorítmico, desglosado en tres fases distintivas que han revelado hallazgos sobre el uso de algoritmos avanzados y metodologías en la toma de decisiones financieras.

En la primera fase, se destaca el desarrollo de una metodología pionera para el preprocesamiento y etiquetado automático de puntos de compra y venta en datos financieros. Esta innovación permitió el entrenamiento efectivo de una red neuronal, cuyos resultados sobresalieron en diversos portafolios, evidenciando el potencial del aprendizaje supervisado cuando se complementa con un preprocesamiento de datos meticuloso y una adecuada elección de un optimizador como ADAM.

La introducción de una estrategia evolutiva modificada que incorpora el conocimiento previo a través del etiquetado propuesto, sin inhibir la exploración propia del algoritmo, surgió como un enfoque híbrido prometedor. Los resultados comparables, y en algunos casos superiores, a la estrategia evolutiva original, sugieren una variante en la dirección de estrategias evolutivas más informadas y, por ende, potencialmente más efectivas.

En la segunda fase, la adopción de FinRL como marco de trabajo da un vistazo del estado del arte en comercio algorítmico mediante aprendizaje por refuerzo, utilizando algoritmos reconocidos como A2C, PPO y DDPG. La aplicación de técnicas de optimización multiobjetivo, aunque costosas en términos computacionales, se justificó plenamente por los resultados obtenidos. La capacidad de personalizar el perfil de riesgo para el usuario final, ofreciendo un espectro de estrategias basadas en diferentes combinaciones de retorno acumulado, volatilidad y estabilidad mediante NSGA-II, representa un avance considerable sobre enfoques unidimensionales previos.

En la tercera fase, la comparativa completa entre seis algoritmos, considerando sus propiedades únicas y rendimientos en distintos portafolios, demostró la versatilidad y robustez del enfoque FinRL. La superioridad mostrada en esta metodología destaca la relevancia de simular los mercados financieros y la importancia de una calibración de hiperparámetros orientada a múltiples objetivos.

No obstante, el proyecto no estuvo libre de limitaciones, tales como las diferencias en las condiciones de entrenamiento y características de los modelos y la comparativa a un nivel funcional, y no profundo en términos de configuración y optimización de los mismos.

En la Tabla 6.1 destacamos los algoritmos vencedores con su respectivo enfoque y rendimiento acumulado.

Portafolio	Algoritmo vencedor	Enfoque	Retorno acumulado
Comunicación	A2C	Enfoque 2 (FinRL)	18.34 %
Bienes cons. duradero	Red Neuronal	Enfoque 1	12.71 %
Bienes cons. no duradero	PPO	Enfoque 2 (FinRL)	39.53 %
Servicios de consumo	PPO	Enfoque 2 (FinRL)	37.15 %
Tecnologías electrónicas	PPO	Enfoque 2 (FinRL)	47.0 %
Minerales energéticos	PPO	Enfoque 2 (FinRL)	178.09 %
Finanzas	PPO	Enfoque 2 (FinRL)	43.57 %
Comercio minorista	A2C	Enfoque 2 (FinRL)	108.4 %

Tabla 6.1: Algoritmo vencedor en cada uno de los portafolios.

Este proyecto, no solo evidencia el progreso tecnológico en el ámbito del comercio algorítmico sino que también ilustra cómo el rigor científico y la innovación son una guía para descubrir y explotar las complejidades del mercado financiero.

Trabajo futuro

Uno de los aspectos más relevantes para el futuro desarrollo de esta investigación es la mejora y comparación de diferentes algoritmos de aprendizaje computacional aplicados al comercio algorítmico. Si bien este proyecto ha explorado métodos como el aprendizaje profundo, algoritmos evolutivos y aprendizaje por refuerzo, aún existen oportunidades para optimizar y perfeccionar estos enfoques. Es fundamental examinar a detalle cada enfoque y evaluar su desempeño en una variedad de escenarios y mercados financieros.

El aprendizaje profundo podría beneficiarse de la experimentación con arquitecturas más complejas, como redes neuronales recurrentes o modelos transformer, que permitan capturar patrones a lo largo del tiempo y considerar relaciones a largo plazo en los datos de mercado. Por otro lado, los algoritmos evolutivos pueden mejorar mediante la implementación de técnicas de optimización más avanzadas y la exploración de métodos de selección de operadores genéticos más eficientes. Además, se incluyen datos adicionales con el fin de enriquecer los modelos utilizados en el comercio algorítmico. Si bien el proyecto actual se ha enfocado en el análisis de datos de precios de activos, existe un basto conjunto de información adicional que podría ser relevante para mejorar la precisión y robustez de los modelos como lo son: información de sentimiento del mercado, noticias financieras, indicadores macroeconómicos y eventos políticos podría proporcionar una visión más completa del contexto en el que se desarrollan las operaciones financieras. Estos datos capturan la percepción general del mercado y las influencias externas que afectan los precios de los activos. La utilización de fuentes de datos no convencionales, como datos de redes sociales y datos alternativos, representan una oportunidad para descubrir señales ocultas que podrían ser valiosas para las estrategias de comercio, y posteriormente generar la aplicación práctica de estrategias de comercio en tiempo real. La transición desde un entorno de pruebas y simulación a un entorno de operación en vivo donde se monitorea y evalúa constantemente el desempeño de las es-

trategias, permitirá realizar ajustes y mejoras de manera ágil. La retroalimentación continua y el análisis de resultados en tiempo real serán fundamentales para mantener la efectividad de las estrategias en un entorno dinámico y cambiante.

En resumen, la mejora y comparación de enfoques de inteligencia artificial, la incorporación de datos adicionales y la adopción en tiempo real son las vertientes donde desarrolla este proyecto. Estas líneas de investigación contribuyen al fortalecimiento de las estrategias de comercio algorítmico basadas en inteligencia artificial, abriendo nuevas oportunidades para una toma de decisiones más informada y eficiente en el mercado financiero.

Apéndice A

Anexo A

En este anexo se muestran los activos que conformaron cada uno de estos portafolios experimentados.

Tabla A.1: Tabla de portafolios y activos

Portafolio	Abreviación	Activos
Bienes de consumo no duradero	B. Con. no D.	'PG' ,'KO' ,'PEP' ,'NKE' ,'UL' ,'BUD' ,'CL'
Comunicación	Com.	'T', 'VZ', 'LUMN', 'VOD', 'TMUS', 'GSAT', 'VIV', 'AMX'
Bienes de consumo duraderos	B. Con. D.	'F', 'NIO', 'TSLA', 'FFIE', 'LCID', 'RIVN', 'GM', 'LI', 'ROKU', 'ARVL'
Servicios de consumo	Serv. de Con	'MCD', 'DIS', 'SBUX', 'BKNG', 'ABNB', 'MAR', 'WBD', 'V', 'MA'
Tecnologías electrónicas	Tec. Elect.	'AAPL', 'TSM', 'NVDA', 'INTC', 'QCOM', 'TXN', 'RTX', 'AMD'
Energéticos minerales	Min. Energ	'XOM', 'SHEL', 'PTR', 'EQNR', 'BP', 'PBR', 'SNP', 'CNQ', 'OXY', 'PXD'
Finanzas	Finanzas	'JPM' ,'BAC', 'RY', 'MS', 'HSBC', 'BX', 'AXP', 'BLK', 'C'
Comercio minorista	Com. Min.	'AMZN', 'WMT', 'BABA', 'HD', 'COST', 'TJX', 'ORLY', 'AZO'

Tabla A.2: Lista de portafolios y sus activos correspondientes utilizados en la experimentación.

Bibliografía

[1] J. Welles Wilder, *New Concepts in Technical Trading Systems*, Trend Research, Universidad de Indiana, 1978.

[2] Stefan Jansen. *Machine Learning for Algorithmic Trading: Predictive models to extract signals from market and alternative data for systematic trading strategies with Python*. Apress, Berkeley, CA. 2020

[3] Chande, Tushar S., and Stanley Kroll. *The New Technical Trader: Boost Your Profit by Plugging into the Latest Indicators.* Wiley. 1994.

[4] Blau, W. *Momentum, Direction, and Divergence.* John Wiley & Sons. 1993.

[5] Murphy, J. J. *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications.* New York, New York Institute of Finance. 1999.

[6] Achelis, S. B. *Technical Analysis from A to Z.* McGraw-Hill. 2000.

[7] Bollinger, J. Bollinger on Bollinger Bands. McGraw-Hill. 2001.

[8] Hastie, T., Tibshirani, R., Friedman, J. (2009). *Overview of Supervised Learning. In: The Elements of Statistical Learning.* Springer Series in Statistics. Springer, New York, NY.

[9] Yaser S. Abu-Mostafa, Malik Magdon-Ismail, Hsuan-Tien Lin, *Learning From Data*, AMLBook, marzo de 2012.

[10] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, Daan Wierstra, *Continuous control with deep reinforcement learning*, Computer Science, Machine Learning, 2015.

`https://arxiv.org/abs/1509.02971`

[11] Thibaut Théate y Damien Ernst. *An application of deep reinforcement learning to algorithmic trading*, Expert Systems with Applications, 2021

`https://www.sciencedirect.com/science/article/pii/S0957417421000737`

[12] Cong Ma , Jiangshe Zhang , Junmin Liu , Lizhen Ji y Fei Gao (2021). *A Parallel multi-module deep reinforcement learning algorithm for trading algorithm*, Neurocomputing, 2021.

https://www.sciencedirect.com/science/article/pii/S0925231221005233

[13] Ash Booth , Enrico Gerding y Frank McGroarty *Automated trading with performance weighted random forests and seasonality* , Expert Systems with Applications, 2014.

https://www.sciencedirect.com/science/article/pii/S0957417413009731

[14] Yingjun Chen y Yijie Hao *Integrating principle component analysis and weighted support vector machine for stock trading signals prediction*, Neurocomputing, 2018.

https://www.sciencedirect.com/science/article/pii/S092523121831066X.

[15] Kinza Yasar, *Neural Network*, Tech Target , última actualización agosto 2023, retomado 2023.

https://www.techtarget.com/searchenterpriseai/definition/neural-network

[16] Cain Evans , Konstantinos Pappas , Fatos Xhafa, *Utilizing artificial neural networks and genetic algorithms to build an algo-trading model for intra-day foreign exchange speculation* ,Mathematical and Computer Modelling, 2013.

https://www.sciencedirect.com/science/article/pii/S0895717713000290

[17] Omer Berat Sezer y Ahmet Murat Ozbayoglu, *Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach* ,Applied Soft Computing, 2018.

https://www.sciencedirect.com/science/article/pii/S1568494618302151

[18] Kai Lei , Bing Zhang and Yu Li , Min Yang y Ying Shen *Time-driven feature-aware jointly deep reinforcement learning for financial signal representation and algorithmic trading*, Expert Systems with Applications, 2020.

https://www.sciencedirect.com/science/article/pii/S0957417419305822

[19] Adam Paszke (retomado 2022) *REINFORCEMENT LEARNING (DQN) TUTORIAL*, Pytorch, 2021.

`https://pytorch.org/tutorials/intermediate/reinforcement_q_learning.html`

[20] Jagdish Bhagwan Chakole , Mugdha S. Kolhe , Grishma D. Mahapurush , Anushka Yadav y Manish P. Kurhekar , *A Q-learning agent for automated trading in equity stock markets*, Expert Systems with Applications, 2021.

`https://www.sciencedirect.com/science/article/pii/S0957417420305856`

[21] Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, Christina Dan Wang. *FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance*, Quantitative Finance , Trading and Market Microstructure, Deep Reinforcement Learning Workshop, 34th Conference on Neural Information Processing Systems (NeurIPS2020), Vancouver, Canada.

`https://arxiv.org/abs/2011.09607`

[22] Willies Ogola *Leveraging OpenAI Gym and the Anytrading Environment for Trading* , Febrero 8, 2022, última visita 2023.

`https://www.section.io/engineering-education/leveraging-openai-gym-and-the-anytrading-environment-for-trading/#getting-started-with-gym-anytrading-environment`

[23] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, *.A fast and elitist multiobjective genetic algorithm: NSGA-II,"* in IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197, April 2002, doi: 10.1109/4235.996017.

[24] Ran Aroussi, *yfinance 0.2.18*, PyPI, 16 abril 2023, última vez consultado em 2023 `https://pypi.org/project/yfinance/`

[25] Bernard Brenyah *"Quandl: Getting End of Day Stock Data with Python"*, Medium, Oct 14, 2017, última vez consultado 2023.

[26] Leo Smigel *"Alpha Vantage API Python Tutorial"*, Analyzing Alpha, enero 2023, última vez consultado 2023.

`https://analyzingalpha.com/alpha-vantage-api-python-tutorial`

[27] Hongtao Cai, *"googlefinance 0.7"*,PyPI, 16 abril 2015, última vez consultado en 2023

`https://pypi.org/project/googlefinance/`

[28] Daniel Goldfarb, *"mplfinance 0.12.9b7"*, PyPI, 8 dic 2022, última vez consultado en 2023

`https://pypi.org/project/mplfinance/`

[29] Kevin Johnson, Lluís Salord Quetglas, Mohd Daniel Saufi Bin Ezani, *"Pandas TA - A Technical Analysis Library in Python 3"*, Github, twopirllc/pandas-ta, 28 jul 2021, retomado en 2023.

https://github.com/twopirllc/pandas-ta#indicators-by-category

[30] Guillaume Lemaitre, Tim Head, Thomas J. Fan,jeremie du boisberranger, *sklearn.feature_selection.mutual_info_classif*, scikit-learn 1.2.2, última actualización Nov 16, 2022, retomado en 2023.

https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_classif.html#r50b872b699c4-2

[31] A. Kraskov, H. Stogbauer and P. Grassberger, *Estimating mutual information*. Phys. Rev. E 69, 2004.

[32] B. C. Ross, *Mutual Information between Discrete and Continuous Data Sets*. PLoS ONE 9(2), 2014.

[33] L. F. Kozachenko, N. N. Leonenko, *"Sample Estimate of the Entropy of a Random Vector"*, Probl. Peredachi Inf., 23:2 (1987), 9-16

[34] Kent State University, *"SPSS TUTORIALS: PEARSON CORRELATION"*, última actualización enero 27, 2023 , retomado 2023. https://libguides.library.kent.edu/SPSS/PearsonCorr

[35] Matt Haberland, Gary Strangman *"scipy.stats.pearsonr"*, Scipy última actualización enero, 2021 , retomado 2023. https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html

[36] Li Deng, Dong Yu *"Deep Learning: Methods and Applications"*, Microsoft Research, One Microsoft Way, Redmond, WA 98052; USA.

https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/DeepLearningBook_RefsByLastFirstNames.pdf

[37] Conor Ryan, *Evolutionary Algorithms and Metaheuristics*, Encyclopedia of Physical Science and Technology (Third Edition), Academic Press, 2003.

https://www.sciencedirect.com/science/article/pii/B0122274105008474

[38] Ian Goodfellow & Yoshua Bengio & Aaron Courville, *Deep Learning*, MIT Press, 2016.

https://www.deeplearningbook.org/

[39] Snow, Derek, *"Machine Learning in Asset Management"* (July 16, 2019). JFDS: https://jfds.pm-research.com/content/2/1/10, Available at SSRN: https://ssrn.com/abstract=3420952 or http://dx.doi.org/10.2139/ssrn.3420952

[40] Daan Wierstra, Tom Schaul , Tobias Glasmachers , Yi Sun , Jan Peters, Jürgen Schmidhuber, *Natural Evolution Strategies*, Una-May O'Reilly, Journal of Machine Learning Research 15 (2014) 949-980. https://www.jmlr.org/papers/volume15/wierstra14a/wierstra14a.pdf

[41] Salimans, T., Ho, J., Chen, X., Sidor, S. and Sutskever, I. *Evolution Strategies as a Scalable alternative to Reinforcement Learning*, 2017.

[42] Wierstra, D., Schaul, T., Glasmachers, T., Sun, Y., Peters, J. and Schmidhuber, J. , *Natural Evolution Strategies.* Journal of Machine Learning Research, 2014.

[43] Erik Sorensen, Ryan Ozzello, Rachael Rogan, Ethan Baker, Nate Parks, Wei Hu, *"Meta-Learning of Evolutionary Strategy for Stock Trading"*, published by Journal of Data Analysis and Information Processing, Vol.8 No.2, 2020.

[44] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., & Kavukcuoglu, K. *Asynchronous methods for deep reinforcement learning*. In International conference on machine learning, 2016.

[45] Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning, An Introduction.* second edition, 2018.

[46] Chris Yoon, *Understanding Actor Critic Methods and A2C*, Towards Data Science, Feb 5, 2019, retomado 2023.

[47] Stable Baselines3, *PPO*,DLR-RM, última actualización May 8, 2023, retomado 2023. https://github.com/DLR-RM/stable-baselines3/blob/master/docs/modules/ppo.rst

[48] Josh Achiam, *Proximal Policy Optimization*,Open IA Spinning Up, última actualización May 8, 2023, retomado 2023.

https://spinningup.openai.com/en/latest/algorithms/ppo.html

[49] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov, *Proximal Policy Optimization Algorithms*,Computer Science, Machine Learning, 2017.

https://arxiv.org/abs/1707.06347

[50] Chris Yoon *Deep Deterministic Policy Gradients Explained Reinforcement Learning in Continuous Action Spaces*, Towards Data Science, 20 marzo 2019, retomado 2023.

[51] Juergen Branke, Kalyan Deb, Kaisa Miettinen, Roman Slowinski, *Multiobjective Optimization, Interactive and Evolutionary Approaches* , Research Gate, enero 2008. https://www.researchgate.net/publication/242505658_Multiobjective_Optimization_Interactive_and_Evolutionary_Approaches_outcome_of_Dagstuhl_seminars

[52] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. *A fast and elitist multiobjective genetic algorithm: nsga-II*. Trans. Evol. Comp, 6(2):182–197, April 2002. URL: http://dx.doi.org/10.1109/4235.996017,doi:10.1109/4235.996017.