Aprendizaje reforzado en *pair-trading* Aplicación para una estrategia *pair-trading*

Reinforcement learning in pair-trading Application for a pair-trading strategy

Cristian Quintero González*

Resumen

Este estudio explora las implicaciones de la implementación de técnicas de aprendizaje reforzado para el *trading* de acciones parte del índice S&P 500 al 15 de octubre de 2022, bajo una estrategia de *pair-trading*. A través de un proceso de selección de pares de acciones se investiga si modelos de aprendizaje reforzado ofrecen una ventaja frente estrategias más simples. Los resultados indicaron sorpresivamente que modelos entrenados con entornos sencillos como el que no permite posiciones en corto, producen una mayor y constante rentabilidad, si se compara con los agentes entrenados en entornos con mayor complejidad lógica como lo son el de proporción del saldo disponible para entrar en posiciones

* Magíster en Finanzas. Analista Cuantitativo - Validador de Modelos de Precios en ING Risk Hub. Polonia. [craquinterogofx@gmail.com] [ORCID ID: 0000-0003-3337-0182].
Artículo recibido: 14 de febrero de 2024.
Aceptado: 15 de mayo de 2024.

Para citar este artículo:
Quintero González, C. (2024). Aprendizaje reforzado en *pair-trading*. Aplicación para una estrategia *pair-trading*. *Odeon*, 26, 55-93.
DOI: https://doi.org/10.18601/17941113.n26.04

según el conjunto de acciones y β-balanceado. Adicionalmente se observó que un incremento en el número de pasos por episodio, que generalmente lleva a consumir un mayor tiempo de entrenamiento para el *hardware* usado, no es garantía de mejorar considerablemente la varianza de la distribución de rentabilidades potenciales en los datos de *trading*, como tampoco es una variable que permita mejorar significativamente la media del retorno u otros indicadores, lo que se evidencia también en los valores *value loss* y *policy loss*, los cuales se tornaban explosivos y más volátiles luego de un valor de episodios determinado.

Palabras clave: aprendizaje reforzado; arbitraje estadístico; *pair-trading*; *trading*.

Clasificación JEL: G11, G17, P4.

Abstract

This study explores the implications of implementing reinforcement learning algorithms on pair-trading strategies. Through a detailed analysis of three selected learning environments, it investigates how algortihms such as A2C and PPO perform when learning by being rewarded based on profit and loss. Findings suggest that a pure pair-trading strategy cannot outoperform a straigforward/ classical approach on pair-trading strategy, named delta distance. This paper discusses the challenges and opportunities associated with this implementation letting the reader to perform an extension considering the challenges presented thereof. The results surprisingly showed that models trained in simple environments, such as the one that does not allow short positions, produce greater and constant profitability compared to agents trained in environments of greater logical complexity, such as the proportion of the balance available to enter. in positions according to the set of actions and β-balanced. In addition, it was observed that increasing the number of steps per episode, which generally leads to consuming a greater training time for the hardware used, does not guarantee a significant improvement in the variance of the distribution of potential returns in the trading data, such as Nor is it a variable that allows a significant improvement in the average return or other indicators, also evidenced by the value loss and policy loss values, which became explosive and more volatile after a certain value of episodes.

Key words: Reinforcement learning; trading; statistical arbitrage; pair-trading.

JEL Classification: G11, G17, P4.

Introducción

Los mercados de valores son centro de atención para muchas personas cuyo interés yace en obtener ganancias por medio de la apertura y el cierre de posiciones sobre instrumentos financieros en los mercados de valores alrededor del mundo, donde el precio de estos instrumentos fluctúa en función de las expectativas de los agentes que intervienen. Sin embargo, aunque es un objetivo común entre los agentes de mercado, los mecanismos y las estrategias usados para alcanzarlo varían entre una cantidad inmensa de alternativas. Estas alternativas pueden considerarse clásicas como pueden ser el análisis técnico y fundamental, mientras otras menos comunes son apalancadas en desarrollos cuantitativos más recientes como aquellos que implementan aprendizaje máquina, aprendizaje reforzado, aprendizaje profundo, entre otros.

No es desconocido en el ámbito financiero que la dinámica de intercambio de los mercados es no determinista, y es por ello que algunos modelos abstractos que suelen ser generalmente deterministas, y con ello más sencillos de implementar, no pueden ser aplicados en tomas de decisión del día a día como en *trading*.

Ahora bien, incorporar elementos no previsibles y/o no recurrentes tales como desastres naturales, desabastecimientos, desorden social, guerras, entre otros, es un reto adicional para la toma de decisiones en el *trading*. En este sentido, algunas estrategias han buscado desacoplar su rendimiento del vaivén del mercado, originando las estrategias conocidas como mercado-neutrales, dentro de las cuales se enmarcan estrategias de *pair-trading* (PT).

De esta manera, este estudio implementa un conjunto de modelos de aprendizaje reforzado con el uso de información histórica de precios y algunos indicadores técnicos aplicados sobre un subconjunto de activos que hacen parte del índice S&P 500 a la fecha 15 de octubre de 2022. El objetivo de estos modelos es identificar los momentos de entrada y salida en una estrategia PT de forma tal que puedan generarse utilidades de estas operaciones. Sin embargo, la evidencia encontrada en el desarrollo del mismo da cuenta del desempeño mixto en términos de rentabilidad de los modelos de aprendizaje reforzado para la constitución de una estrategia de PT, aún más limitado por el hecho de que los modelos entrenados aquí puedan aprender una estrategia de PT completamente. Esto último para subrayar que los resultados obtenidos fueron más de una estrategia discrecional distinta a la de PT.

1. Estado del arte

La literatura reciente deja ver el creciente interés por aplicar modelos de aprendizaje reforzado para la toma de decisiones para *trading*. Diversos estudios han demostrado resultados mixtos. Sin embargo, la adopción de tales innovaciones cuantitativas conlleva desafíos significativos desde el marco de información y tiempos de procesamiento hasta gestión activa del desempeño de los modelos.

1.1. Modelos en *pair-trading*

Los modelos para tratar estrategias de PT han captado la atención de varios autores, de los cuales se distinguen algunos como los señalados por Krauss (2015), donde los clasifica en función del enfoque, de los cuales pueden subrayarse los siguientes[1]: distancia medida en desviaciones estándar de la media, cointegración, series de tiempo, control estocástico, *machine learning*, cópulas, PCA. Para todos los anteriores se asume una relación de largo plazo entre la serie de precios de los activos que componen un par.

El trabajo de Do y Faff (2010) cuestiona la utilidad de una estrategia PT con información desde 1962 hasta 2009, mostrando que hasta dicha fecha este tipo de estrategias presentaban una rentabilidad decreciente pero aún positiva, especialmente en periodos donde la volatilidad aumentaba con relativa persistencia temporal. Esto se debe a aspectos como la explotación concurrente de esta estrategia en pares y/o eventos relacionados con ingresos corporativos.

1.1.1. Distancia medida en desviaciones estándar de la media

El modelo de distancia es definido como aquel que analiza las señales de entrada y salida en términos de desviaciones estándar desde la media (Δ), calculada para una ventana de tiempo. En este sentido, algunos autores como Do y Faff (2010) usan dos desviaciones estándar.

En general, se dice que se compra una unidad del *spread* del par cuando este es menor o igual a $-\Delta$, o se vende una unidad del *spread* del par cuando es mayor o igual a Δ. La razón detrás de esta regla se debe a que se espera que en un *spread*

[1] Para más detalle sobre los distintos enfoques y sus objetivos particulares, remitirse al trabajo de Krauss (2015), sección 2.

que sigue un proceso gausiano, por ejemplo, la probabilidad de que el *spread* se desvíe más o menos Δ es justamente la integral del proceso gausiano, o lo que es lo mismo $1 - N(\Delta)$, y de forma análoga la probabilidad de que el nivel del *spread* del par sea menor que $-\Delta$ es de $N(-\Delta)$, y por simetría $1 - (\Delta) = N(-\Delta)$.

1.1.2. Cointegración

Estos modelos se relacionan con el concepto de tendencia estocástica entre dos series temporales (Vidyamurthi, 2004), también conocido como modelo de tendencias comunes desarrollado por Stock (Stock y Watson, 1988). En este se busca probar que dos series de tiempo que tienen una representación en VAR (Vector Autorregressive model), y dados dos componentes no estacionarios, se anularán[2].

En el caso de (Vidyamurthi, 2004), el autor hace uso de un test de Engle-Grander para determinar por mínimos cuadrados ordinarios el coeficiente de cointegración de la siguiente regresión:

$$log(P_t^A) = \mu + \beta log(P_t^B) + \epsilon_t \tag{1}$$

donde $\epsilon_t \sim N(0, \sigma^2)$, las variables P_t^A y P_t^B son los precios del activo A y B, respectivamente, y μ es una constante representando la tendencia. A su vez, esta definición la relaciona el autor con lo que sería el uso de Δ como medida de distancia desde la media μ para el caso del método de distancia[3].

1.1.3. Métodos estocásticos

Adicionalmente, en el ámbito de modelos estocásticos para PT se pueden encontrar comparaciones de una serie de modelos de reversión a la media aplicados en el mercado del oro y el petróleo, abarcando un modelos estocásticos de un solo factor hasta uno donde incorpora la dinámica de tasas de interés y volatilidad (Schwartz, 1997). Además, en línea con encontrar un método generalizado para solucionar procesos Ornstein-Uhlenbeck de reversión a la media vale destacar algunos que usan polinomios de Hermite para construir una secuencia explícita de funciones cerradas que convergen a la función de máxima log-verosimilitud compartiendo sus propiedades asintóticas (Ait-Sahala, 2002); al igual que trabajos relacionados con métodos de estimación donde se propone un marco de trabajo para la estimación simultánea

[2] Ver ecuación 5.3, capítulo 5 de Vidyamurthi (2004).
[3] Ver ecuación 5.8 de Vidyamurthi (2004).

de estimadores de los parámetros del modelo Ornstein-Uhlenbeck (Haress y Hu, 2021).

Sin embargo aunque el problema fue abordado desde la perspectiva de modelos Ornstein-Uhlenbeck (Bertram, 2009; Bertram, 2010), otros autores evalúan el modelo asumiendo que la variabilidad del proceso no cumple con los criterios de normalidad establecidos desde el proceso Ornstein-Uhlenbeck (Goncu y Akildirim, 2016), por lo cual el uso de un proceso de Levy para tener en cuenta las colas pesadas se muestra como alternativa, y ha sido puesto a prueba en el mercado de materias primas de varias bolsas de Estados Unidos. En la misma línea de modelos de reversión a la media, se encuentra otro trabajo que agrega al análisis la derivación de una transformada de Laplace, la cual le permite evaluar los límites de la estrategia, o lo que es lo mismo, los niveles óptimos de entrada basándose en tres tipos de dinámica del proceso estocástico (Zeng y Lee, 2014).

Otro trabajo encontrado menciona que los valores de los parámetros θ, μ, σ pueden ser estimados por máxima log-verosimilitud, usando los valores observados $(x_i^{\alpha,\beta})_{i=1,2,..,n}$ donde α y β son los momentos de ingreso y salida de la posición respectivamente (Leung y Li, 2016).

Es importante resaltar que el factor de difusión σdB representan los saltos aleatorios en la dinámica del proceso y que, por definición, en el movimiento Browniano, estos saltos siguen una distribución normal. Esto resulta relevante en la medida en que se han expuesto otros modelos complementarios a la dinámica de las rentabilidades en acciones, por ejemplo, se ajustan mejor al definir que el factor de difusión no sigue una distribución normal, sino que adopta una distribución Varianza-Gamma (VG), Inversa Normal Gausiana (NIG) y, en general, una hiperbólica (GHYP) ver (Konlack y Wilcox, 2014; Madan et al., 1999; Carr y Wu, 2004), respectivamente. Vale la pena mencionar que una característica respecto a estas tres distribuciones es que la distribución GHYP es la forma generalizada ante un parámetro de las distribuciones VG y NIG.

1.2. Aprendizaje reforzado en *trading*

En el marco de aprendizaje reforzado existen diversidad de configuraciones y tipologías aplicadas en *trading*. En ellas se pueden encontrar propuestas haciendo uso de modelos Q-network con espacios de acción discretos para compra, venta y no acción, y con variables estado dentro de las cuales resaltan una serie de aspectos

como la representación del tiempo en formato *timestamp* como una función sinusoidal con el fin de identificar las sesiones del mercado a lo largo del día (Huang, 2018). Respecto a la red profunda que ayuda en la representación de la función Q, esta es una de cuatro capas con funciones de activación ELU (Exponential Linear Unit)[4] salvo la última capa que es LSTM (Long-Short Term Memory). El *buffer* de eventos utilizado en el entrenamiento tuvo para los autores un mejor desempeño cuando este era pequeño, comparado con la metodología habitual que es de *buffers* medios o grandes.

Otros trabajos se orientan a implementar un agente de aprendizaje reforzado con regla de actualización de política Q-learning que al igual que Huang (2018) usa una red neuronal con tres unidades de salida representando el espacio de estados, pero esta vez la función de activación predominante en las neuronas es ReLU (Rectified Linear Unit) (Carapuco et al., 2018). El mecanismo de entrenamiento es el usual, separación de datos de entrenamiento, validación y prueba, con ventanas de tiempo móviles en el último conjunto de datos, dada la característica no estacionaria de los mercados, ver figura 9 de (Carapuco et al., 2018). Un aspecto particular en el desarrollo de este trabajo es que medidas como regularización *L1*, *L2* y *dropout* no mejoraron el rendimiento al ser establecidos en capas medias de la red, sino solo en la capa final de esta. El enfoque de incorporar redes neuronales y algoritmos de aprendizaje reforzado hace parte de un subconjunto de algoritmos como Deep Reinforcement Learning (DRL), también definido por Plaat (2022) como la combinación de aprendizaje profundo y reforzado cuya arquitectura permite interactuar con entornos complejos y de alta dimensión, justamente delegando a la parte profunda, o *deep*, la operación de la alta dimensionalidad que métodos tabulares no pueden optimizar.

En términos de los tipos de regla de actualización de política *off-policy*[5] en el esquema *actor-critic*, trabajos como el de Kowalik et al. (2019), plantean el uso de Deep Deterministic Policy Gradient para el comercio de 4 índices en distintos países: DJI (USA), TSX (Canadá), JSE (Sudáfrica), Sensex (India). Usando datos diarios de precio, volumen y estadísticos de búsqueda de Google (Google Trends) y combinaciones de estos para los índices, compara tres modelos de estrategia: aprendizaje reforzado, regresión lineal y comprar y mantener *(buy-and-hold)*. Dentro de las conclusiones obtenidas está que la información desde Google Trends no aportó

[4]Ver https://ml-cheatsheet.readthedocs.io/en/latest/activation_functions.html

[5]La política de aprendizaje *off-policy* considera todo el conjunto de datos aprendidos en acciones pasadas, mientras la política *on-policy* considera las acciones más recientes.

mayor impacto en el resultado del modelo.

También existen modelos embebidos y multi-capa, donde las capas son usadas, por ejemplo, para: primera capa compuesta por un modelo de preprocesamiento de datos apoyado en redes convolucionales con el fin de obtener metacaracterísticas de las series de tiempo, una capa intermedia donde entra en juego el modelo de aprendizaje reforzado usando un algoritmo Deep Double Q-learning (Double DQN), y una tercera capa que fusiona las señales de distintas iteraciones del agente denominado meta-aprendiz (Carta et al., 2021).

El aprendizaje reforzado en *trading* ha tenido una variedad de enfoques, desde algunos puristas en el sentido de que usan solo un método de aprendizaje reforzado y otros que lo embeben junto con otras técnicas de aprendizaje máquina para reforzar su eficiencia o escoger el espacio de estados, como es el caso de Chakole et al. (2021), el cual usa algoritmos de aglomeración (K-Means) sobre estados preprocesados *[O,H,L,C,V]* y posteriormente Q-learning con acciones *[Compra, Venta y Mantener]*, en el *trading* de acciones. Los hay también de otros tipos los cuales incorporan otra información fuera de los precios, como por ejemplo indicadores técnicos, o variables macroeconómicas para el caso de *trading* de baja frecuencia.

Otros trabajos, referencian a modelos usados para el *trading* de acciones, monedas y materias primas para países como Estados Unidos, Corea, China, Brasil, entre otros (Sun et al., 2021).

2. Modelo

Para analizar el impacto de los modelos de aprendizaje reforzado en una estrategia de *pair-trading*, se propone el uso de dos algoritmos: Advantage Actor-Critic (A2C) y Proximal Policy Optimization (PPO). Estos modelos son aplicados en tres entornos de aprendizaje denominados: entorno sin posiciones cortas, entorno de variable acción continua y entorno con parámetro β. Además, con el fin de revisar si muestran una ventaja frente a métodos convencionales, estos son son comparados con un modelo base, anteriormente referido como distancias desde la media.

Se entiende estrategia de *pair-trading* aquella estrategia que hace uso de una señal compuesta por los precios de dos activos (ver ecuación 2), con el fin de entrar largo en uno de ellos y corto en el otro, con la expectativa de que los precios converjan nuevamente a su dinámica habitual.

$$Señal = \frac{P_A(t)}{\beta P_B(t)} \tag{2}$$

3. Experimento

Para validar los modelos propuestos, se diseñó un experimento que consiste en la implementación de dos agentes de aprendizaje reforzado haciendo uso de algoritmos A2C y PPO, aplicados sobre un subconjunto de pares de acciones que pertenecen al índice S&P 500 al 15 de octubre de 2022. El índice no es objeto del experimento, por lo cual no se considera el rebalanceo del mismo.

La información histórica de los activos es obtenida desde abril 1 de 2010 hasta abril 5 de 2021. Sin embargo, debido a que algunos de los activos en el índice S&P 500 pueden no tener una larga historia que sea de utilidad para la construcción de la señal, son descartados del conjunto.

3.1. Selección de pares

La selección de pares se realiza filtrando los pares bajo dos criterios: la distancia del modelo Dynamic Time Warping (DTW), estacionariedad bajo las pruebas estadisticas Augmented Dickey Fuller (ADF) y KPSS.

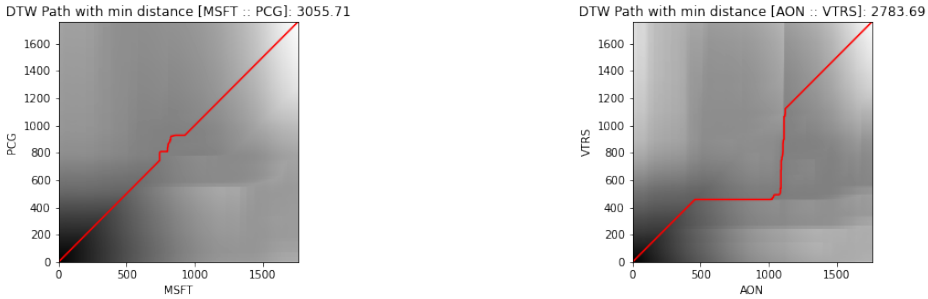
3.1.1. Distancia DTW

El primer paso es procesar las combinaciones de acciones del índice S&P 500 por el algoritmo DTW, solamente de aquellas acciones cuyo volumen diario promedio sea superior a un millón de dólares como criterio adicional de liquidez. Una vez filtradas las series, estas son estandarizadas con el objetivo de evitar ruidos por el efecto de escala en las cotizaciones al momento de calcular las diferencias, y del mismo modo para hacerlas comparables entre los resultados sobre distintos pares de acciones.

Durante la ejecución de este proceso se obtuvieron las matrices de DTW, las cuales representan las distancias entre cada uno de los puntos para cada una de las series de tiempo. En la figura 1, por medio de escala de grises, se muestran las distancias relativamente menores usando tonos claros, distancias relativamente menores al valor inicial infinito establecido en la creación de la matriz DTW. Con esto

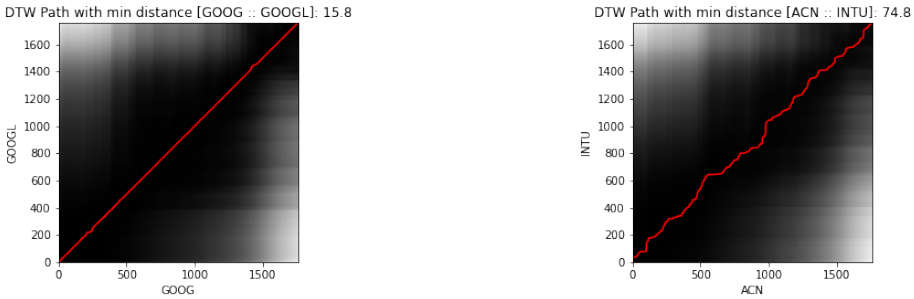
debe mencionarse que se espera que solamente los puntos muy cercanos a la diagonal secundaria de la matriz DTW tengan colores muy oscuros, lo cual representaría que los valores son demasiado altos para considerarlos dentro de la ruta mínima. Adicionalmente, la distancia mínima calculada por DTW se expresa visualmente a través de la linea roja de cada matriz, esperando que esta forme la diagonal secundaria lo más lineal posible sobre la matriz, lo que a su vez indica una distancia mínima justamente en los nodos equitemporales o casi equitemporales de las series en el cálculo.

Figura 1. Representación figura de matrices DTW para pares con grandes distancias.



Por otra parte, en la figura 2 se aprecia cómo las tonalidades oscuras que oscilan al rededor de la diagonal roja muestran que aunque existen distancias entre los nodos equitemporales de las series, estos son tan altos que no constituyen un nivel representativo para considerarlos en la ruta trazada por DTW.

Figura 2. Representación figura de matrices DTW para pares con pequeñas distancias



3.2. Test de estacionariedad

Con el fin de considerar solamente aquellos pares que muestren una variabilidad y promedio constante, se optó por realizar las pruebas de estacionariedad Dickey-Fuller aumentado (ADF) y KPSS sobre el conjunto total de pares filtrados en el paso anterior. De este modo, del total de 10.514 pares posibles para el S&P 500, cuyo volumen diario promedio es superior a un millón de dólares, resultaron solamente 94 pares que, bajo las pruebas de estacionariedad, se mostraban estacionarias con niveles de significancia del 5 % (tabla 1).

Tabla 1. Razón de precios de los pares de acciones del índice S&P 500, que bajo los estadísticos ADF y KPSS se muestran estacionarios

Par	d-DTW	ADF	$ADF_{p-value}$	KPSS	$KSPP_{p-value}$	Par	d-DTW	ADF	$ADF_{p-value}$	KPSS	$KSPP_{p-value}$
A_AVGO	81.55	-3.63	0.01	0.23	0.10	AON_PHM	227.80	-3.28	0.02	0.46	0.05
ACN_NDAQ	84.71	-4.07	0.00	0.36	0.09	AOS_FDX	229.02	-3.28	0.02	0.45	0.06
AJG_BRO	91.78	-2.95	0.04	0.36	0.09	AKAM_APD	234.25	-4.98	0.00	0.29	0.10
ACN_PLD	92.51	-4.03	0.00	0.17	0.10	AEE_DRI	234.37	-3.77	0.00	0.27	0.10
ACN_CBRE	106.62	-3.66	0.00	0.32	0.10	AKAM_WMT	239.63	-4.76	0.00	0.27	0.10
AME_DOV	107.28	-3.85	0.00	0.34	0.10	AOS_EMN	241.72	-4.79	0.00	0.22	0.10
ADI_AME	110.13	-3.40	0.01	0.26	0.10	AMGN_PCAR	243.07	-3.76	0.00	0.29	0.10
A_SHW	118.53	-3.03	0.03	0.43	0.06	AOS_CFG	243.14	-3.39	0.01	0.36	0.09
ABT_NEE	121.05	-4.12	0.00	0.44	0.06	ALL_DRI	246.16	-3.43	0.01	0.43	0.07
ADSK_ISRG	122.49	-3.48	0.01	0.32	0.10	ADI_STX	246.17	-3.19	0.02	0.40	0.08
ACN_DHI	124.19	-4.05	0.00	0.23	0.10	ADP_PHM	254.70	-3.02	0.03	0.36	0.09
ADI_DOV	126.90	-5.02	0.00	0.37	0.09	ADM_HES	258.73	-3.70	0.00	0.22	0.10
ACN_HCA	131.13	-3.25	0.02	0.27	0.10	ABC_HES	264.01	-3.49	0.01	0.36	0.09
ADI_HLT	133.79	-4.04	0.00	0.28	0.10	AOS_MGM	264.09	-4.37	0.00	0.28	0.10
ADI_AON	134.44	-4.06	0.00	0.26	0.10	ALL_CI	264.90	-3.48	0.01	0.46	0.05
A_TMUS	134.64	-3.01	0.03	0.29	0.10	AKAM_EBAY	267.79	-3.50	0.01	0.43	0.06
ADI_HD	135.53	-3.91	0.00	0.36	0.09	CMCSA_DIS	268.35	-3.77	0.00	0.27	0.10
AON_HLT	136.45	-3.51	0.01	0.32	0.10	HBAN_HPE	274.95	-4.06	0.00	0.28	0.10
AME_AON	145.33	-3.47	0.01	0.23	0.10	AMGN_PEG	276.90	-3.50	0.01	0.46	0.05
ABT_GRMN	149.61	-2.90	0.04	0.23	0.10	AKAM_MCD	285.59	-3.66	0.00	0.38	0.08
ADP_APH	152.92	-3.10	0.03	0.40	0.08	AEE_MAR	291.55	-2.89	0.05	0.46	0.05
ADSK_MA	161.90	-3.15	0.02	0.33	0.10	AKAM_NEM	294.61	-5.19	0.00	0.32	0.10
ADP_HUM	165.24	-3.12	0.03	0.27	0.10	AKAM_KMX	295.53	-4.16	0.00	0.19	0.10
AON_FAST	166.54	-2.97	0.04	0.45	0.05	AMGN_EMR	297.00	-3.00	0.03	0.43	0.06
ADP_AXP	167.23	-4.20	0.00	0.42	0.07	ABC_LKQ	301.02	-3.45	0.01	0.32	0.10
AMT_UNP	169.73	-3.39	0.01	0.38	0.09	AEP_AMGN	301.24	-4.00	0.00	0.45	0.06
AES_APH	171.14	-2.87	0.05	0.45	0.06	AMGN_ETR	301.44	-3.73	0.00	0.19	0.10
AON_CBRE	172.22	-3.00	0.03	0.32	0.10	ALL_DFS	306.15	-3.66	0.00	0.44	0.06
AME_PHM	174.90	-4.42	0.00	0.37	0.09	AMGN_DFS	321.63	-3.07	0.03	0.40	0.08
ADBE_CZR	175.75	-4.62	0.00	0.24	0.10	ANET_CZR	325.61	-3.68	0.00	0.30	0.10
AES_EXPD	175.84	-2.95	0.04	0.39	0.08	AOS_STZ	329.81	-3.43	0.01	0.30	0.10
ADI_PHM	176.80	-3.94	0.00	0.25	0.10	AMGN_SYY	331.07	-3.20	0.02	0.34	0.10
ADI_HUM	187.72	-3.21	0.02	0.37	0.09	ALK_MO	336.80	-3.50	0.01	0.23	0.10
AKAM_BF-B	190.98	-3.78	0.00	0.45	0.06	AMGN_DTE	337.73	-3.80	0.00	0.39	0.08
AMT_PHM	196.21	-3.99	0.00	0.41	0.07	ABBV_DRI	352.01	-3.00	0.04	0.31	0.10
AFL_SYY	201.08	-4.02	0.00	0.43	0.06	AKAM_ATVI	357.00	-3.13	0.02	0.45	0.06
AFL_EXC	201.57	-3.48	0.01	0.18	0.10	CSX_TWTR	381.96	-3.32	0.01	0.35	0.10
ALK_BEN	202.14	-3.34	0.01	0.28	0.10	AFL_DLTR	398.42	-3.52	0.01	0.45	0.05
ADP_CNC	202.69	-3.03	0.03	0.36	0.09	AOS_MET	426.21	-3.07	0.03	0.38	0.09
AEE_TJX	204.23	-3.17	0.02	0.35	0.10	AMCR_WY	426.22	-3.23	0.02	0.40	0.08
AKAM_CCI	205.05	-4.69	0.00	0.16	0.10	AEE_APTV	429.24	-2.91	0.04	0.35	0.10
AON_DHI	208.46	-3.52	0.01	0.45	0.06	ABC_VMC	435.52	-3.41	0.01	0.40	0.08
AKAM_CHD	210.74	-3.72	0.00	0.12	0.10	AKAM_WRB	437.68	-2.92	0.04	0.43	0.06
AKAM_MKC	214.39	-4.51	0.00	0.18	0.10	AAP_SO	476.32	-3.07	0.03	0.32	0.10
AES_PH	218.03	-2.99	0.04	0.45	0.06	AMCR_EXPE	577.80	-2.86	0.05	0.17	0.10
ABBV_SCHW	220.23	-3.21	0.02	0.19	0.10	AIG_CTRA	633.03	-3.02	0.03	0.43	0.07
AKAM_LDOS	224.70	-4.55	0.00	0.22	0.10	AMCR_JNPR	637.70	-3.69	0.00	0.21	0.10

Como resultado, la tabla 2 muestra la distancia DTW calculada para cada uno de los pares seleccionados luego de pasar por los filtros anteriormente mencionados.

Tabla 2. Distancias DTW para los 82 pares de acciones del S&P seleccionadas

Par	Distancia	Par	Distancia	Par	Distancia	Par	Distancia
A_AVGO	81.5	ADSK_MA	161.9	AOS_FDX	229.0	AMGN_ETR	301.4
ACN_NDAQ	84.7	ADP_HUM	165.2	AKAM_APD	234.3	ALL_DFS	306.1
AJG_BRO	91.8	ADP_AXP	167.2	AEE_DRI	234.4	AMGN_DFS	321.6
ACN_PLD	92.5	AMT_UNP	169.7	AKAM_WMT	239.6	ANET_CZR	325.6
ACN_CBRE	106.6	AON_CBRE	172.2	AOS_EMN	241.7	AOS_STZ	329.8
AME_DOV	107.3	AME_PHM	174.9	AMGN_PCAR	243.1	AMGN_SYY	331.1
ADI_AME	110.1	ADBE_CZR	175.7	AOS_CFG	243.1	ALK_MO	336.8
A_SHW	118.5	AES_EXPD	175.8	ALL_DRI	246.2	AMGN_DTE	337.7
ABT_NEE	121.0	ADI_PHM	176.8	ADI_STX	246.2	ABBV_DRI	352.0
ADSK_ISRG	122.5	ADI_HUM	187.7	ADP_PHM	254.7	CSX_TWTR	382.0
ACN_DHI	124.2	AMT_PHM	196.2	ADM_HES	258.7	AOS_MET	426.2
ADI_DOV	126.9	AFL_SYY	201.1	ABC_HES	264.0	AMCR_WY	426.2
ACN_HCA	131.1	AFL_EXC	201.6	AOS_MGM	264.1	AEE_APTV	429.2
ADI_HLT	133.8	ALK_BEN	202.1	AKAM_EBAY	267.8	ABC_VMC	435.5
ADI_AON	134.4	ADP_CNC	202.7	CMCSA_DIS	268.4	AKAM_WRB	437.7
A_TMUS	134.6	AEE_TJX	204.2	HBAN_HPE	275.0	AAP_SO	476.3
ADI_HD	135.5	AKAM_CCI	205.0	AKAM_MCD	285.6	AMCR_EXPE	577.8
AON_HLT	136.5	AKAM_CHD	210.7	AKAM_NEM	294.6	AIG_CTRA	633.0
AME_AON	145.3	AKAM_MKC	214.4	AKAM_KMX	295.5	AMCR_JNPR	637.7
ABT_GRMN	149.6	ABBV_SCHW	220.2	AMGN_EMR	297.0		
ADP_APH	152.9	AKAM_LDOS	224.7	ABC_LKQ	301.0		

4. Agente de aprendizaje reforzado

Para la implementación de los algoritmos de aprendizaje reforzado se hizo uso de librerías como FinRL, Stable Baselines, ElegantRL y pyfolio (Quantopian) para la construcción y el entrenamiento de los mismos. Durante la fase de entrenamiento se llevó a cabo entrenamiento con agentes individuales y embebidos para cada serie de tiempo de pares en el intervalo de tiempo 1-abril-2010 a 1-enero-2021, con intervalo de validación del 2-enero-2021 a 5-abril-2021, y puesto a prueba para el periodo de desde 6-abril-2021 hasta 31-marzo-2022. El valor inicial de la cartera en dinero efectivo se asume en dólares (USD) por valor de $10.000 USD.

Dado que durante el entrenamiento se busca que el algoritmo consiga reconocer patrones para una estrategia PT, el entorno fue construido haciendo uso de la librería *gym*, tanto para los agentes con algoritmos individuales con el set de entrenamiento completo como para los agentes entrenados por ventanas de tiempo. El *script* base del ambiente implementado en este trabajo toma como base el usado por la librería FinRL para el caso de compra/venta de acciones, el cual fue modificado en dos

funciones de la clase *StockTradingEnv* [6].

4.1. Variables estado (S_t)

Las variables estado forman un vector compuesto por cantidades en un momento *t*, indicadores técnicos[7] y valor del disponible, variables identificadas por código y definición en la tabla 3.

Tabla 3. Código y descripción de variables estado usadas en los agentes de aprendizaje reforzado

Código	Descripción
account_value	Valor Disponible de la cartera
P_A	Precio en el instante t del activo A
P_B	Precio en el instante t del activo B
Q_A	Cantidad del activo A en la cartera
Q_B	Cantidad del activo B en la cartera
macd_A	indicador técnico MACD del activo A
macd_B	indicador técnico MACD del activo B
boll_ub_A	indicador técnico BB del activo A
boll_ub_B	indicador técnico BB del activo B
rsi_30_A	indicador técnico RSI 30d del activo A
rsi_30_B	indicador técnico RSI 30d del activo B
cci_30_A	indicador técnico CCI 30d del activo A
cci_30_B	indicador técnico CCI 30d del activo B
dx_30_A	indicador técnico DX del activo A
dx_30_B	indicador técnico DX del activo B
close_30_sma_A	indicador técnico SMA 30d del activo A
close_30_sma_B	indicador técnico SMA 30d del activo B
close_60_sma_A	indicador técnico SMA 60d del activo A
close_60_sma_B	indicador técnico SMA 60d del activo B

[6]Código base disponible en GitHub en: https://github.com/AI4Finance-Foundation/FinRL/blob/master/finrl/meta/env_stock_trading/env_stocktrading.py

[7]Indicadores como: MACD-Moving Average ConvergenceDivergence, BB-Bandas de Bollinger, RSI-Relative Strenght Index, CCI-Commodity Channel Index, DX-Directional Movement Index, SMA-Simple Moving Average.

4.2. Entornos de aprendizaje

Cabe mencionar que fueron ejecutados tres tipos de entornos: el primero es el disponible en la librería FinRL, el segundo donde tiene en cuenta la acción en valor continuo ($a_i \in [-1, 1]$) para definir el monto por negociar de cada acción, y un tercero que tiene en cuenta la razón del par para cumplir con la cuota de PT en la estrategia, es decir, haciendo uso de la razón de PT definida en la ecuación 2.

Para todos ellos se establecieron parámetros de mercado como costos transaccionales (ζ) de 0,1 % del valor de la transacción, un costo de préstamo para posiciones (ζ_{cortos}) cortas de 0,1 % del valor de la transacción. Algunos de ellos no tienen la posibilidad de entrar en operaciones cortas, sin embargo, para aquellos entornos en los que sí se permitió, se estableció un valor límite del valor inicial del portafolio (ϕ) hasta el cual se pueden tener posiciones cortas abiertas, de forma tal que simule una restricción por garantías y riesgo.

4.2.1. Entorno sin posiciones cortas

Como se mencionó en el apartado de marco teórico, las estrategias PT requieren la capacidad de producir órdenes de venta incluso cuando no existen en inventario tales acciones, o también conocido como venta en corto. Sin embargo, el entorno inicial provisto por FinRL tiene restricción sobre esta condición, por lo cual fue entrenado con fines de comparación. El seudoalgoritmo relacionado con *compra* y *venta* del entorno se muestra a continuación:

Algorithm 1 seudo-algoritmo de Venta, primer entorno entrenado

Require:
Estado $\mathcal{S} = \{C, P_1, P_2, Q_1, Q_2, IndTec_{1,1}, \ldots, IndTec_{1,n}, IndTec_{2,1}, \ldots, IndTec_{2,n}\}_t$
Acciones $\mathcal{A} = \{(0,0), (0,1), (0,-1), (1,0), (1,-1), (1,1), (-1,1), (-1,0), (-1,-1)\}$, $A : \mathcal{S} \Rightarrow \mathcal{A}$
Activo $i = \{1, 2\}$ ▷ Para una estrategia PT, serán solo dos activos con índice *i*
Comisión $\zeta = 0{,}1\,\%$
procedure VENTA(i, $\mathcal{S}$, ζ)
  if $Q_i > 0$ **then** ▷ Q_i es cantidad del activo *i* en el vector de estado $\mathcal{S}$
    $Q_{i,vender} \leftarrow \left\lfloor \frac{\mathcal{S}[0]}{P_i} \right\rfloor$ ▷ $\mathcal{S}[0]$ es el valor de la cartera
    $VlrVenta \leftarrow Q_{i,vender} * P_i * (1 - \zeta)$
    $\mathcal{S}[0] \leftarrow \mathcal{S}[0] + VlrVenta$
    $\mathcal{S}[2 + i] \leftarrow \mathcal{S}[2 + i] - Q_{i,vender}$ ▷ La cantidad Q_i es actualizada en el vector de estado $\mathcal{S}$
  else if $Q_i \leq 0$ **then**
    $Q_{i,vender} \leftarrow 0$
  end if
end procedure

En el algoritmo 1 se pueden distinguir elementos del aprendizaje reforzado, como son el vector de estado ($\mathcal{S}$) el cual contiene los valores en un instante de tiempo *t*, de: C dinero efectivo en la cartera, P_i el precio del activo *i*, Q_i la cantidad del

activo *i* en la cartera, $(IndTec_{i,1}, \ldots, IndTec_{i,n})$ los indicadores desde *1* hasta *n* del activo *i*.

Algorithm 2 seudo-algoritmo de Compra, segundo entorno entrenado

Require:
Estado $\mathcal{S} = \{C, P_1, P_2, Q_1, Q_2, IndTec_{A,1}, \ldots, IndTec_{A,i}, IndTec_{B,1}, \ldots, IndTec_{B,i}\}$
Acciones $\mathcal{A} = \{(0,0), (0,1), (0,-1), (1,0), (1,-1), (1,1), (-1,1), (-1,0), (-1,-1)\}$, $A : \mathcal{S} \Rightarrow \mathcal{A}$
Activo $i = \{1, 2\}$ ▷ Para una estrategia PT, serán solo dos activos con índice *i*
Comisión $\zeta = 0{,}1\,\%$
procedure COMPRA(i, $\mathcal{S}$, ζ)
$Q_{i,comprar} \leftarrow \left\lfloor \frac{\mathcal{S}[0]}{P_i} \right\rfloor$ ▷ $\mathcal{S}[0]$ es el valor de la cartera
$VlrCompra \leftarrow Q_{i,comprar} * P_i * (1 + \zeta)$
$\mathcal{S}[0] \leftarrow \mathcal{S}[0] - VlrCompra$
$\mathcal{S}[2 + i] \leftarrow \mathcal{S}[2 + i] + Q_{i,comprar}$ ▷ La cantidad Q_i es actualizada en el vector de estado $\mathcal{S}$
end procedure

Con base en esta implementación se evidenció que el algoritmo de entorno de FinRL para compra de acciones *StockTradingEnv* tiene como restricción que siempre tendrá como prioridad la compra de acciones cuando el conjunto de acciones $\mathcal{A}$ tenga al menos un *1* como componente de la misma y producto de la función de política $Q(s, a)$, y luego de esto ejecutará la venta. Esto restringe al modelo en el sentido de que para cuando se produzca un escenario tal que $\mathcal{A} = (0,1) \vee (-1,1) \vee (1,0) \vee (1,-1) \vee (1,1)$, el primer descuento por realizar será sobre la compra, por lo cual al pasar a la siguiente operación el valor disponible en efectivo ya estará reducido por el valor de la anterior compra.

4.2.2. Entorno con variable acción continua y venta en corto

A diferencia del entorno anterior, la acción que se obtiene desde la función de política $Q(s, a)$ ya no se muestra como valores enteros $(-1, 0, 1)$, representando las acciones de $Venta, Neutro, Compra$, sino que ahora representa la proporción del valor en efectivo dentro de la cartera (C). Ahora $a_i \in [-1, 1]$, siendo entonces *venta* cuando $a_i < 0$ y *compra* cuando $a_i > 0$. De este modo, ahora los seudoalgoritmos que determinan *compra* y *venta* son definidos como los algoritmos 3 y 4.

Si se compara el seudoalgoritmo de la librería FinRL 1 con el 3 de venta con acción en variable continua, se aprecia que en el segundo se incorporan dos nuevos parámetros: ζ_{corto} que corresponde al costo en el que incurre un *trader* por pedir prestadas unas acciones en corto, y ϕ cuyo rol es definir el valor máximo de exposición de la cartera en posiciones en corto como función del valor inicial de la cartera $C_{inicial}$. De igual manera, vale la pena mencionar que en el ejercicio de la representación de la función de densidad de probabilidad de este espacio de estado continuo

Algorithm 3 seudo-algoritmo de Venta, tercer entorno entrenado

Require:
Estado $\mathcal{S} = \{C, P_1, P_2, Q_1, Q_2, IndTec_{1,1}, \ldots, IndTec_{1,n}, IndTec_{2,1}, \ldots, IndTec_{2,n}\}_t$
Acciones $a_i \in \mathcal{A} | \{\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2), \mathcal{A}_i \in [-1, 1], i = 1, 2\}, \quad A : \mathcal{S} \Rightarrow \mathcal{A}$
Activo $i = \{1, 2\}$ ▷ Para una estrategia PT, serán solo dos activos con índice *i*
Comisión $\zeta = 0{,}1\,\%$
ComisiónCorto $\zeta_{corto} = 0{,}1\,\%$ ▷ Por prestamos en corto
MaxExposicionCorto $\phi = 30\,\%$
DineroEfectivoInicial $C_{inicial} = 10{,}000$
procedure VENTA(i, $\mathcal{S}$, ζ, a_i)
  $Q_{i,vender} \leftarrow \left\lfloor \frac{\mathcal{S}[0]}{P_i} \right\rfloor$ ▷ $\mathcal{S}[0]$ es el valor de la cartera
  if $Q_i \leq 0$ **then**
    if $\frac{a_i \mathcal{S}[0]}{P_i} \leq C_{inicial} * \phi$ **then** ▷ Venta $\leq$ máxima exposición en corto permitida
      $VlrVenta \leftarrow Q_{i,vender} * P_i * (1 - \zeta - \zeta_{corto})$
    else
      $Q_{i,vender} \leftarrow 0$
      $VlrVenta \leftarrow 0$
    end if
  else if $Q_i > 0$ **then** ▷ activo *i* en el inventario y la venta es menor que Q_i
    $VlrVenta \leftarrow Q_{i,vender} * P_i * (1 - \zeta)$
  end if
  $\mathcal{S}[0] \leftarrow \mathcal{S}[0] + VlrVenta$
  $\mathcal{S}[2 + i] \leftarrow \mathcal{S}[2 + i] - Q_{i,vender}$ ▷ La cantidad Q_i es actualizada en el vector de estado $\mathcal{S}$
end procedure

se usó una representación gaussiana similar a la señalada por (Sutton y Barto, 2020) en su sección 13.7, en el cual los parámetros $\mu(S, \theta)$ y $\sigma(S, \theta)$ son representaciones numéricas de la forma vectorial del espacio estado S.

Algorithm 4 seudo-algoritmo de Compra, segundo entorno entrenado

Require:
Estado $\mathcal{S} = \{C, P_1, P_2, Q_1, Q_2, IndTec_{A,1}, \ldots, IndTec_{A,i}, IndTec_{B,1}, \ldots, IndTec_{B,i}\}$
Acciones $a_i \in \mathcal{A} | \{\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2), \mathcal{A}_i \in [-1, 1], i = 1, 2\}, \quad A : \mathcal{S} \Rightarrow \mathcal{A}$
Activo $i = \{1, 2\}$ ▷ Para una estrategia PT, serán solo dos activos con índice *i*
Comisión $\zeta = 0{,}1\,\%$
procedure COMPRA(i, $\mathcal{S}$, ζ, a_i)
  $Q_{i,comprar} \leftarrow \left\lfloor \frac{a_i \mathcal{S}[0]}{P_i} \right\rfloor$ ▷ $\mathcal{S}[0]$ es el valor de la cartera
  $VlrCompra \leftarrow Q_{i,comprar} * P_i * (1 + \zeta)$
  $\mathcal{S}[0] \leftarrow \mathcal{S}[0] - VlrCompra$
  $\mathcal{S}[2 + i] \leftarrow \mathcal{S}[2 + i] + Q_{i,comprar}$ ▷ La cantidad Q_i es actualizada en el vector de estado $\mathcal{S}$
end procedure

Al igual que el anterior entorno de aprendizaje descrito, este sigue estando sujeto a que cuando una de las acciones sea de compra, esta seguirá siendo ejecutada primero, por lo cual la posterior orden estará restringida a calcular la cantidad de acciones por operar una vez ya fue descontada la primera operación en la variable dinero disponible C dentro del vector de estado $\mathcal{S}$.

4.2.3. Entorno con parámetro β

El siguiente entorno fue diseñado para resolver el problema de descontar instantáneamente la variable C en cada operación, especialmente en las compras. Para lograrlo,

se separaron los cálculos de compra y venta en una capa superior, y se determinó la cantidad de compra y venta de cada activo. Además, se calculó la razón más conservadora de la cantidad propuesta por la política $Q(s, a)$.

Se impuso una condición adicional debido a la explosión observada en las dinámicas de los agentes. Esta explosión se debía a una súbita ola de ventas en corto, lo que hacía parecer que había una fuente de ingresos infinita por activos que no estaban en la cartera. Por esta razón, se restringió la posibilidad de entrar en corto en ambos activos simultáneamente cuando la acción $(a = (a_1, a_2))$ tal que $(a_1 < 0, a_2 < 0)$.

Así es como, obteniendo las cantidades sugeridas en función del conjunto de acciones (a_1, a_2), se realiza el cálculo del parámetro β que recalculará las cantidades en función de si $\beta > 0$ o $\beta < 0$ considerando la existencia en inventario de cada uno de los activos. Cuando $\beta > 0$ se asume entonces que las cantidades del activo *i=1* serán un múltiplo del *i=2*, y viceversa cuando $\beta < 0$. De este modo se fuerza al agente a buscar cubrir posiciones existentes de modo tal que, por ejemplo, si en la variable estado $\mathcal{S}$ se tiene la cantidad $Q_1 = 10$ acciones del activo *i=1*, un $\beta = 2$, entonces se necesitará tener una posición en el activo *i=2* de $Q_2 = 2$, dando como resultado que si no se tiene y la acción $a_2 < 0$ se realizará la operación en corto que haga la razón muy cercana a 1. Otros casos pueden ser analizados con este mismo razonamiento.

4.3. Agentes implementados

Durante el desarrollo se probaron agentes con algoritmos como Policy Proximal Optimization (PPO) y Advantage Actor-Critic (A2C). Ambos algoritmos son de política estocástica y se probaron en los tres pares de activos con menor distancia DTW y en dos con mayor distancia DTW. Un mayor detalle del algoritmo *A2C* puede consultarse en Mnih, Kavukcuoglu et al. (2015) y Mnih, Puigdomènech et al. (2016), mientras que para detalles sobre el algoritmo *PPO*, puede consultarse en Schulman et al. (2017).

La selección de PPO fue alentada por los resultados de trabajos como los de Liang et al. (2018) y Yang et al. (2020). En el primero, se concluye que, si bien el algoritmo de aprendizaje llega a aprender una estrategia de *asset allocation*, esta no es sobresaliente ni óptima. En el segundo, se reporta un buen rendimiento al trabajar con algoritmos embebidos.

La primera implementación involucró el entrenamiento de los agentes con el 80 % de los datos históricos disponibles (entrenamiento y validación). Sin embar-

go, estas implementaciones mostraron una capacidad muy reducida para generar rentabilidades que superaran o al menos se acercaran al benchmark S&P 500.

Para ambos algoritmos, PPO y A2C, se usaron redes neuronales para el Actor y el Crítico, cada una con dos capas ocultas de 120 neuronas y funciones de activación tangente hiperbólico. La última capa de consolidación a la capa de salida de dos neuronas utilizó la función de activación ReLU.

4.4. Agentes por tramos

Dado el pobre rendimiento generado por agentes individuales al ser entrenados sobre el conjunto entero de datos, fue probado el rendimiento de agentes que realizan entrenamiento sobre ventanas de tiempo más cortas y excluyentes entre periodos.

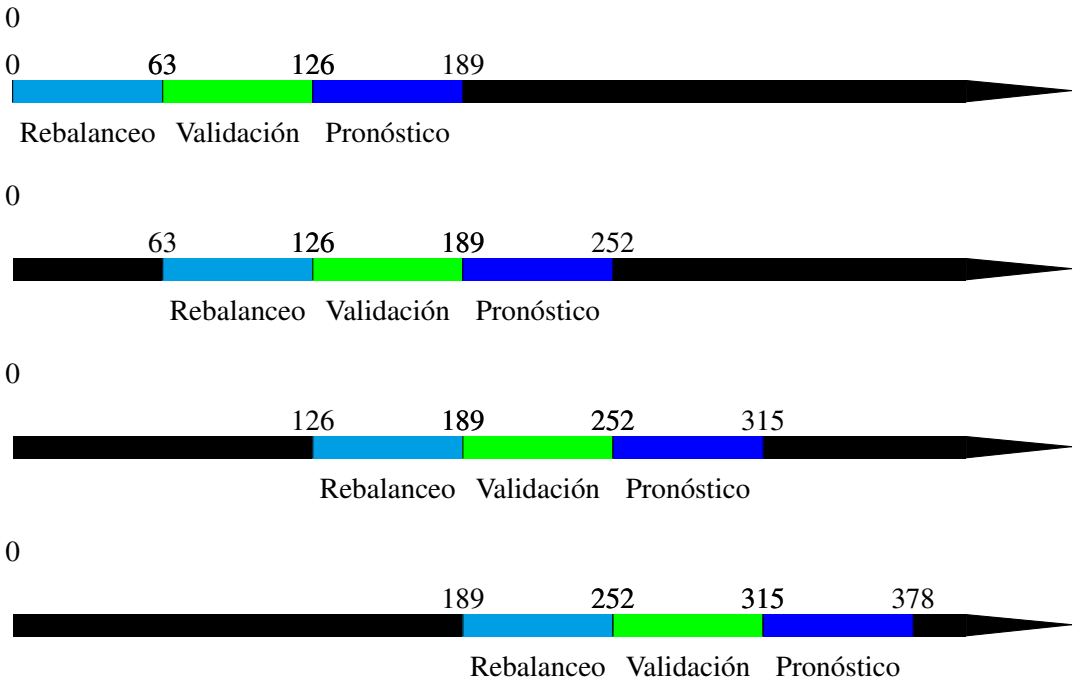
Este enfoque se construye dividiendo los datos en segmentos temporales para entrenar los agentes. Para este fin, la *ventana de rebalanceo* fue determinada en 63, que resulta en reentrenar el agente cada 3 meses (ver figura 3). Estos parámetros cubren tanto parámetros del agente como de la política. Un ejemplo de parámetros de política pueden ser el optimizador que se usa en las redes neuronales y las tasas de aprendizaje de las mismas.

Para evaluar cada uno de estos agentes fueron entrenados para cada par de activos mencionados como alcance de este trabajo, subrayando que tanto PPO como *A2C* aprenden la probabilidad de $Q(s, a)$ a través del *Actor*, razón por la cual al momento de realizar las pruebas fue necesario implementar diversas ejecuciones para, por medio del método Monte Carlo, de 1.000 trayectorias cada uno, encontrar el valor medio de la rentabilidad generada por los modelos como promedio de cada una de las trayectorias en cada instante *t*. Como resultado, usando el entorno predefinido para que el agente aprendiera a reconocer estrategias de PT puras, se obtuvieron rentabilidades mixtas, especialmente para los pares, los tres mayores y tres menores distancias DTW (tabla 2).

4.5. Hiperparámetros

En lo referente a los hiperparámetros usados en los modelos, estos fueron establecidos por medio de un enfoque heurístico debido al complejo establecimiento de los mismos, dada la alta dimensionalidad, complejidad que también fue registrada en algunos trabajos previos como el de Mnih, Kavukcuoglu et al. (2015), donde se menciona que los métodos tradicionales para *Deep Learning* suelen ser costosos

Figura 3. Representación de los agentes por tramos en ventanas temporales, en número de día



para hallar los parámetros en aprendizaje reforzado. Los establecidos finalmente pueden verse en la tabla 4.

En esta etapa fueron consideradas métricas de entrenamiento en cada una de las ventanas, tales como: pérdida de la función valor, pérdida de la función Q, Varianza explicada, pérdida de entropía y divergencia Kullback-Liebler *(KL)* aproximada. De particular interés fue el comparar estas métricas entre los modelos entrenados con dos diferentes entornos de aprendizaje como por ejemplo aquel sin posibilidad de realizar operaciones en corto y aquel que toma el valor de la acción (a_1, a_2) como proporción a invertir del saldo en efectivo disponible.

Tabla 4. Hiperparámetros usados en cada uno de los agentes bajo los algoritmos *A2C* y *PPO*

Hiper-parámetro	A2C	PPO	Descripción
time_steps	450000	450000	Máx. número de pasos por episodio
n_steps	5	2048	Número de pasos en el buffer
optimizador	RMSprop	–	Optimizador usado para la red neuronal del actor y el critico
ortho_init	True	–	Determina si o no se usaran valores iniciales ortogonales
gae_lambda	1	0.95	Factor para trade-off de sesgo-varianza para GAE
ent_coef	0.005	0.01	Coef. de entropía para el calculo de pérdida
vf_coef	0.5	0.5	Coef. de la función-valor para el calculo de pérdida
max_grad_norm	0.5	0.5	Máximo valor para el recorte del gradiente
rms_prop_eps	0.00001	–	Parámetro ϵ de RMSprop
batch_size	–	128	Minibatch usado para actualizar el gradiente
buffer_size	–	1000	Tamaño del buffer para el replay
clip_range	–	0.2	Parámetro de recorte para la función valor
n_epochs	–	10	Numero de episodios para pérdida sustituta
learning_rate	0.0007	0.00025	Tasa de aprendizaje en las redes neuronales
net_arch	[64,64]	[400,300]	Arquitectura de la red neuronal profunda

4.5.1. Algoritmo A2C

En el caso del algoritmo *A2C*, las métricas consideradas para determinar el número máximo de pasos por episodio *(time_steps)* fueron la pérdida de función valor *(value loss)*[8], y la pérdida de función-Q *(policy loss)*. Esto se debe a que dentro del análisis de estas funciones en un modelo que tiene un verdadero aprendizaje se espera que el *value loss* incremente o se mantenga estable debido al aprendizaje que está adquiriendo y, una vez la recompensa se estabiliza, este valor comienza a decrecer, (figura 4).

Por otro lado, el nivel de *value loss* se relaciona con cuánto cambia la función de política, donde en un estado de aprendizaje óptimo se espera que este sea decreciente. De este modo se buscará establecer el número de pasos en un número de iteraciones inmediatamente antes de que *value loss* caiga súbitamente a cero, y *policy loss* se vuelva explosivamente variante, como puede verse en las figura 5 para el caso del par A-AVGO con el algoritmo *A2C* y el ambiente β-balanceado, el cual determinó un *time step* menor o igual a 480 mil pasos.

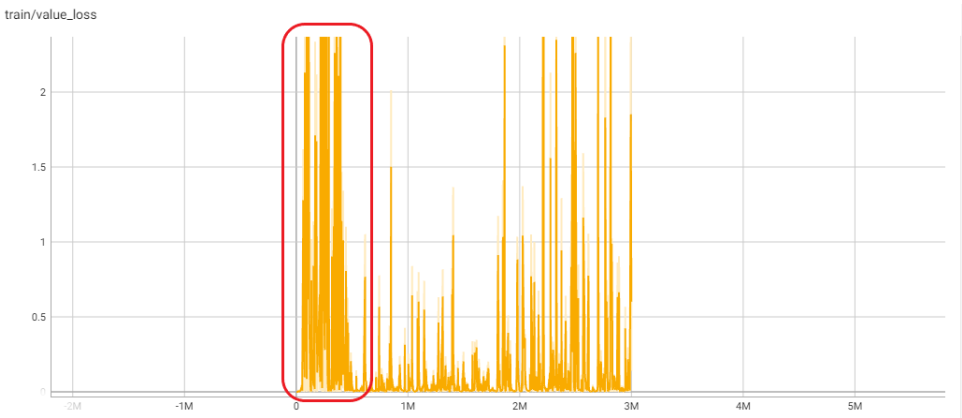
4.5.2. Algoritmo PPO

Por otra parte, para el algoritmo *PPO* fueron consideradas las métricas de divergencia aproximada de Kullback-Liebler *(approx kl)* y varianza explicada *(explained*

[8]Esta pérdida es definida como la diferencia entre el valor esperado por la función valor (red neuronal del crítico) y el valor realmente observado una vez una acción a sido realizada.

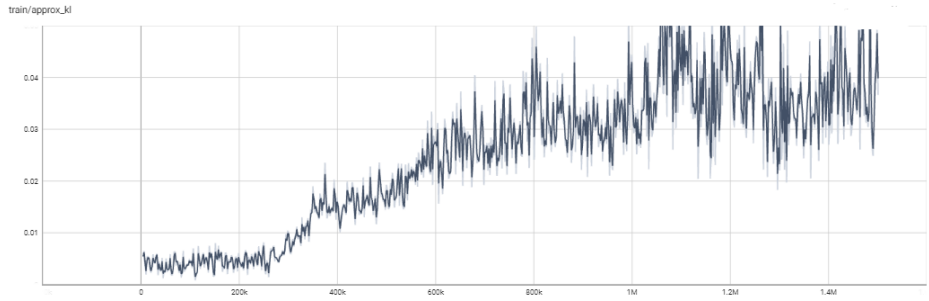
Figura 4. Rango *time step* considerado como óptimo en el entrenamiento del agente para el par A-AVGO en el ambiente β-balanceado, bajo *policy loss* para el algoritmo **A2C**

Figura 5. Rango *time step* considerado como óptimo en el entrenamiento del agente para el par A-AVGO en el ambiente β-balanceado, bajo *value loss* para el algoritmo A2C



variance). La primera de ellas considerada relevante debido a que estima la diferencia que tiene la política como función de distribución del par Estado-acción entre actualizaciones, donde un repentino incremento de esta divergencia muestra que la política está dejando de aprender secuencialmente y, en cambio, puede estar entrando en un proceso de aprendizaje aleatorio. Un ejemplo se puede ver en la figura 6.

Figura 6. Rango *time step* considerado como óptimo en el entrenamiento del agente para el par AAP-SO en el ambiente β-balanceado, bajo *approx kl* para el algoritmo PPO



La segunda es considerada relevante puesto que permite medir si el valor estimado de la función valor se acerca al observado, y esperando que este se acerque a 1 (figura 7).

Un ejemplo de la relevancia para determinar el parámetro *(time_steps)* es que al observar algunos resultados sobre agentes entrenados con *(time_steps)* un tanto superiores a los límites establecidos, estos mostraban un deterioro significativo sobre la rentabilidad promedio en la simulación Monte Carlo, como por ejemplo, el observado en la figura 8 para el par A-AVGO para el modelo entrenado con el algoritmo *PPO* con número de pasos 450.000 y 1 millón. Vale la pena mencionar que una rentabilidad de -100 % automáticamente resulta en el límite inferior del valor de la cartera, pues esto implicará que su valor ha sido completamente reducido, sin embargo, ese límite fue liberado para el ejercicio de comparación a fin de ver el nivel mínimo al que podría llegar el entrenamiento netamente en términos numéricos.

Figura 7. Rango *time step* considerado como óptimo en el entrenamiento del agente para el par AAP-SO en el ambiente β-balanceado, bajo *explained variance* para el algoritmo PPO

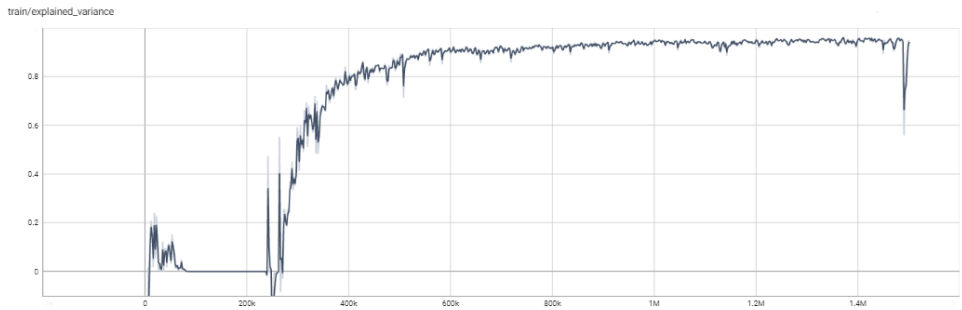
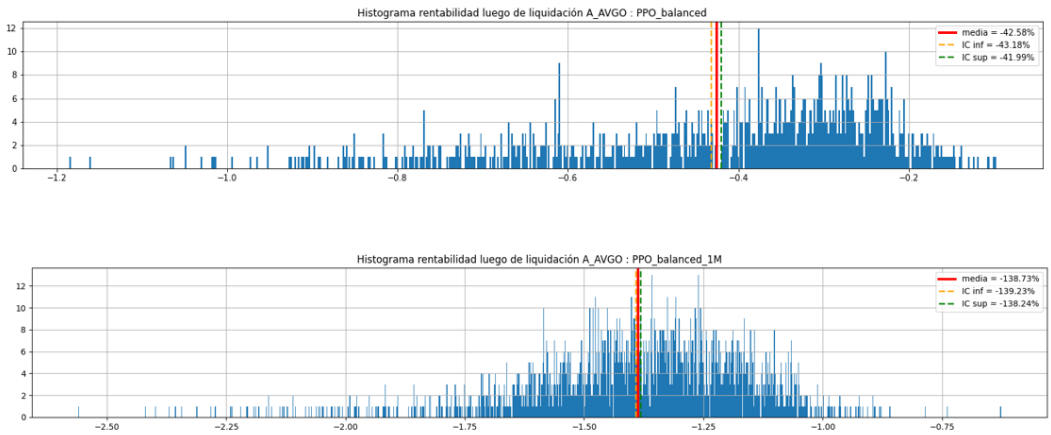


Figura 8. Diferencia del valor promedio de rentabilidad obtenida al establecer el parámetro *(time_steps)* en 450 mil (arriba) comparado con 1 millón (abajo), para el par A-AVGO usando el algoritmo PPO



Cabe destacar que al tratarse de funciones de política estocástica, el resultado de estos valores muestra un mayor ruido que un algoritmo con política determinista, debido al ruido que genera para la red neuronal el aprender desde la función *Advantage*[9], que sirve para el agente como medida de qué tan diferente o mejor fue el resultado de la acción tomada frente a lo que se esperaba al tomarla.

Para un mayor entendimiento de algunos parámetros del optimizador RMSprop en el algoritmo *A2C* ver Mnih, Puigdomènech et al. (2016), sección 4. Este mismo comportamiento se observó de forma similar para los demás pares de activos como también para el entorno denominado balanceado al ser comparado con el entorno sin posiciones cortas.

5. Resultados

Los resultados obtenidos para cada uno de los agentes entrenados, expuestos a los tres entornos de aprendizaje mencionados anteriormente, presentaron una dinámica mixta. Con esto, es necesario mencionar que durante la prueba de los modelos se hicieron 1.000 ejecuciones de cada uno de los modelos en el periodo de prueba de los mismos para, por medio de Monte Carlo, obtener el valor medio de la respuesta de los agentes, esto debido a que las Q-funciones para los modelos *A2C* y *PPO* son representadas por una distribución del espacio acción a un estado S_i, o lo que es igual, implementan funciones de política estocástica por lo cual no existe un valor determinístico a un estado S_t observado. En adición, con fines de comparación, fue realizado el *backtesting* de una estrategia PT basado en las distancias respecto a la media de los 5 pares de activos.

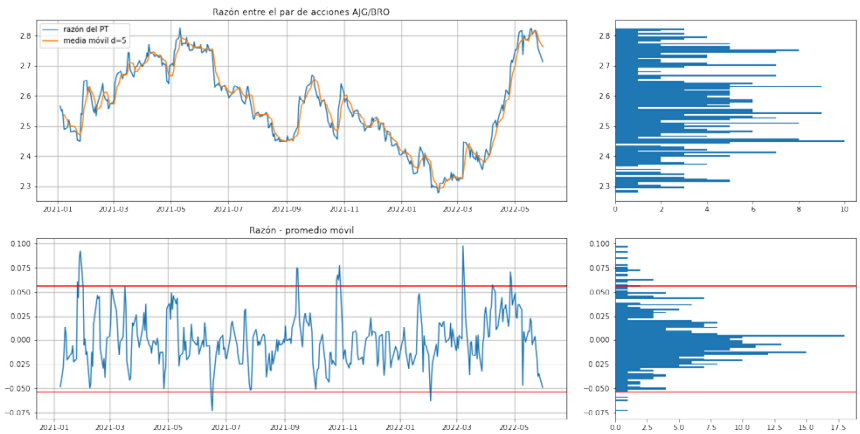
5.1. Distancia desde la media

Esta estrategia de *pair-trading* es ampliamente conocida y para esta implementación se basa en primero restar de la razón de los precios del par de activos la media móvil de 5 días, con el fin de remover la tendencia estocástica de la serie. Luego de esto se calcula la media y desviación estándar de la nueva serie a fin de encontrar la banda

[9]La definición de la función Advantage puede verse en la fórmula número 10 de Schulman et al. (2017), donde se representa como el la diferencia del valor descontado de recompensas siguiendo una política π_τ y el valor estimado por la función valor (red neuronal). También ver Plaat (2022) sección 3.2.4.3.

superior e inferior de la estrategia, definidas como la media $\pm 2\sigma$. Acto seguido, se asume que cuando la razón de precios ajustada sea superior a la banda superior se entrará en una posición corta sobre el par o, lo que es lo mismo, será una posición corta para la acción A y una posición larga para la acción B. Del mismo modo, cuando la razón de precios ajustada sea menor a la banda inferior, se entrará en una posición larga del par. Un ejemplo de este método puede verse en la figura 9.

Figura 9. Bandas para una estrategia PT para el par de activos AJG-BRO



Para la ejecución de la estrategia se tuvieron en cuenta los mismos costos transaccionales considerados para los agentes en los algoritmos de aprendizaje reforzado, con el fin de tener una base comparable de rentabilidad y demás medidas. Así, la tabla 5 expone los resultados obtenidos con esta estrategia para los pares de activos seleccionados.

5.2. Resultados para el entorno sin posiciones cortas

Los modelos entrenados en el entorno con restricción a posiciones cortas produjo en su gran mayoría rentabilidades anuales positivas al cierre del año de prueba. Con este enfoque llama la atención que en promedio la rentabilidad es más alta que la rentabilidad observada del *benchmark*, en varias de las ejecuciones individuales,

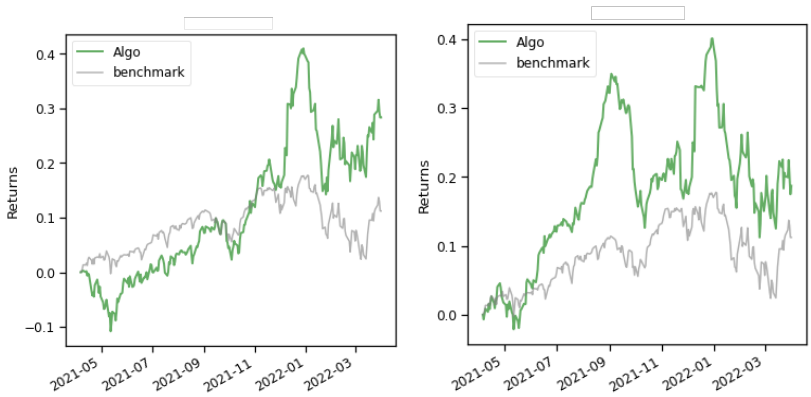
Tabla 5. Medidas bajo la estrategia simple de desviación desde la media, para los 5 pares seleccionados del S&P 500

	Retorno (%)	Volatilidad (%)	SR	CR	SoR	MD (%)
A_AVGO	-0,004	8,099	0,039	-0,000	0,0581	-7,291
ACN_NDAQ	21,370	17,489	1,193	1,772	2,3440	-12,055
AJG_BRO	-1,350	7,890	-0,132	-0,150	-0,1855	-8,992
AAP_SO	11,940	20,903	0,646	0,749	0,8842	-15,930
AIG_CTRA	-13,889	29,739	-0,351	-0,377	-0,4788	-36,798

SR: Sharpe ratio, CR: Calmar ratio, SoR: Sortino Ratio, MD: Max Drawdown

como se ve en la figura 10. Sin embargo, las métricas de ejecuciones individuales no garantizan una constante rentabilidad que supere al *benchmark*. Por esta razón, un mejor acercamiento es considerar los rangos de rentabilidad generados en diversas ejecuciones para el algoritmo A2C y PPO. Es relevante hacer notar que el algoritmo A2C generó una leve heterogeneidad entre los agentes para las distintas ventanas.

Figura 10. Rentabilidad acumulada en el par A-AVGO bajo algoritmos A2C (izquierda) y PPO (derecha) usando el entorno de aprendizaje sin posiciones en corto, comparado con el *benchmark* S&P 500



Del mismo modo, analizando el intervalo de confianza[10] de la rentabilidad me-

[10]Calculado como $\mu \pm \frac{(1-\alpha)\sigma}{N_\tau}$, siendo μ la media de la rentabilidad, σ la desviación estándar de

dia obtenida a través de 1.000 ejecuciones de cada agente se estimó en un rango de entre $\pm$ 0,35 % sobre la media de 28 % anual para este entorno de aprendizaje. La tabla 6 muestra el promedio de varias de las métricas de rentabilidad y volatilidad a fin de tener un panorama más amplio del rendimiento de cada uno de los agentes entrenados. Sin embargo, aunque esta puede inducir a considerar que existen ventanas con un mayor rendimiento a la hora de medir la rentabilidad, este no es una regla general para todos los pares de activos, como tampoco lo es que los agentes entrenados con el algoritmo A2C tienen un mejor rendimiento por sobre PPO.

Tabla 6. Medidas bajo la estrategia simple de desviación desde la media

	Retorno (%)	Volatilidad (%)	SR	CR	SoR	MD (%)
A_AVGO	-0,004	8,099	0,039	-0,000	0,0581	-7,291
ACN_NDAQ	21,370	17,489	1,193	1,772	2,3440	-12,055
AJG_BRO	-1,350	7,890	-0,132	-0,150	-0,1855	-8,992
AAP_SO	11,940	20,903	0,646	0,749	0,8842	-15,930
AIG_CTRA	-13,889	29,739	-0,351	-0,377	-0,4788	-36,798

SR: Sharpe ratio, CR: Calmar ratio, SoR: Sortino Ratio, MD: Max Drawdown

Ahora, si bien es cierto que los agentes entrenados con este entorno de aprendizaje tienen rendimientos positivos que pueden sobrepasar el *benchmark* en rentabilidad, este no constituye un curso natural para una estrategia PT, debido a que la estrategia requiere entrar en posiciones cortas, razón por la que estos modelos sirven como referente de comparación a los posteriores.

5.2.1. Variables relevantes en el modelo

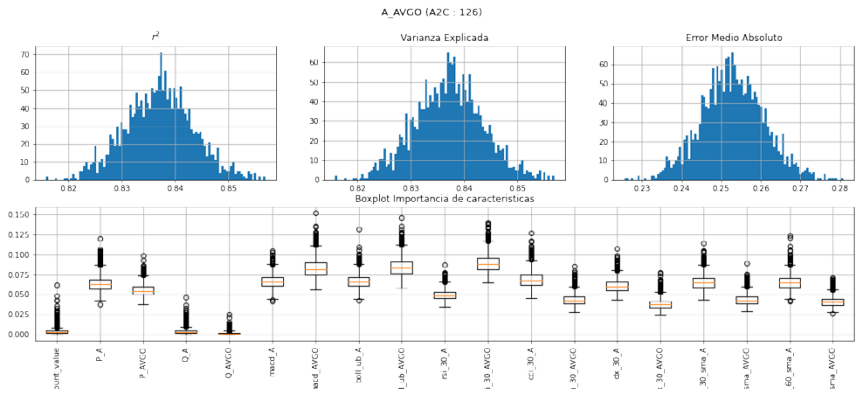
En esta ocasión se construyó un bosque aleatorio *(Random Forest)* de regresión con los mismos valores de las simulaciones realizadas para encontrar los valores medios de rentabilidad. Algunos parámetros del bosque son: número de estimadores es 100, error medio absoluto (MAE, por sus siglas en inglés) como criterio de división de características y sin límite al número de niveles en el árbol como tampoco número de hojas para cada estimador. Cabe destacar que al igual que para el análisis de los demás entornos de aprendizaje, las características relevantes del modelo fueron analizadas en función de la simulación, esto debido a que algunos parámetros tales

la rentabilidad y N_τ el número de ejecuciones o trayectorias.

como la cantidad disponible en la cartera de cada uno de los activos, y el nivel de disponible en la cartera varían en función de cada simulación, dependiendo de las decisiones tomadas por cada agente. Un ejemplo de esto puede verse en la figura 11, la cual hace referencia a las características relevantes para el agente usando el algoritmo *A2C* en el par A-AVGO, cuya puntuación r^2 en promedio es de 0,83, la varianza explicada promedio es de 0,829 y el error medio absoluto es en promedio 0,25.

Los demás modelos entrenados para las ventanas 189, 252 y 315 en el par A-AVGO, usando tanto el algoritmo *A2C* como *PPO*, generan valores de relevancia de las variables muy similares.

Figura 11. Variables relevantes en el modelo *A2C* para el par A-AVGO, ventana 126

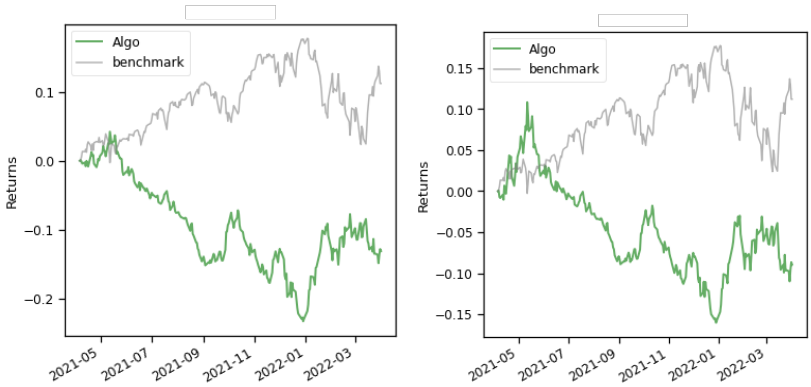


5.3. Resultados para el entorno con acciones (a_i) como proporción del disponible

Contrario al comportamiento del modelo que es entrenado con restricción a operaciones en corto, los agentes entrenados considerando el valor de la acción (a_1, a_2), que pertenecen al espacio de acciones A_i, como la proporción del presupuesto disponible en efectivo para realizar operaciones, mostraron un rendimiento inferior en el valor medio de rentabilidad y varias veces negativos, como puede verse en la figura 12, aunque al comienzo del intervalo de prueba superó al *benchmark* por un tiempo reducido. Adicionalmente, se pudo ver que el valor medio para cada una

de las ventanas entrenadas mostraba ocasionalmente un comportamiento mixto, es decir, para determinadas ventanas se mostraban rendimientos positivos, mientras en otras se encontraban negativos para los pares A-AVGO, AIG-CTRA y ACN-NDAQ particularmente con el algoritmo A2C.

Figura 12. Rentabilidad acumulada en el par A-AVGO bajo algoritmos A2C (izquierda) y PPO (derecha) usando el entorno de aprendizaje a_i como proporción del disponible, comparado con el *benchmark* S&P 500



En este entorno de aprendizaje se observó por primera vez el comportamiento mixto que adoptaron los agentes al incorporar lógica de una estrategia PT. Un elemento que llamó inicialmente la atención en este fue que al no restringir por medio de una exposición máxima del portafolio a posiciones cortas, debido el mecanismo de recompensa, el agente optaba por enteramente entrar en posiciones cortas en cerca de un 98 % de las operaciones realizadas. Este fenómeno se consideró como aquel en el cual el agente considera que este es dinero *gratis* ingresando al saldo en efectivo disponible del portafolio, por esto fue incorporada la cota máxima $(C_{inicial} * \phi)$ vista en el seudoalgoritmo 3. El comportamiento mixto mencionado puede verse en los valores promedio del cuadro 7, donde en todos ellos se ve una disminución comparado con los valores del cuadro 6.

5.3.1. Variables relevantes en el modelo

A diferencia de lo sucedido en el análisis de variables relevantes en el entorno sin posiciones en corto, durante el análisis en el presente entorno se evidenció que par-

Tabla 7. Promedio de medidas de rendimiento anual para el par A - AVGO bajo el entorno (a_i) como proporción del disponible

Agente	Ventana	Retorno (%)	Volatilidad (%)	SR	CR	SoR	MD (%)
A2C	126	33,11	26,01	1,23	1,65	1,89	-20,10
	189	-10,31	8,32	-1,58	-0,68	-1,97	-14,73
	252	-5,47	4,39	-1,24	-0,64	-1,52	-7,20
	315	28,30	24,89	1,10	1,42	1,69	-19,93
PPO	126	16,49	22,33	0,77	0,86	1,12	-20,28
	189	13,82	22,07	0,65	0,73	0,96	-20,82
	252	15,80	22,25	0,74	0,82	1,08	-20,45
	315	16,50	22,36	0,76	0,86	1,11	-20,42

SR: Sharpe ratio, CR: Calmar ratio, SoR: Sortino Ratio, MD: Max Drawdown

ticularmente para el agente entrenado con la ventana 252, las variables explicativas cambiaron, ahora ponderando mayormente la cantidad del activo *A* y también *AVGO* en el mecanismo de generación de acción, como puede verse en la figura 14. Destaca en esa observación que el rango de variabilidad de la relevancia en las variables mencionadas incrementó sustancialmente. Paralelamente destaca que al observar la figura 13, las ventanas 189 y 252 presentaron el retorno negativo, potencialmente relacionado con el cambio presentado en el modelo en cuanto a relevancia de las variables parte del espacio observado para el agente (S_t).

Figura 13. Histograma de rentabilidad anual en el par A-AVGO con algoritmos *A2C (izquierda)* y *PPO (derecha)*. Valores en porcentaje

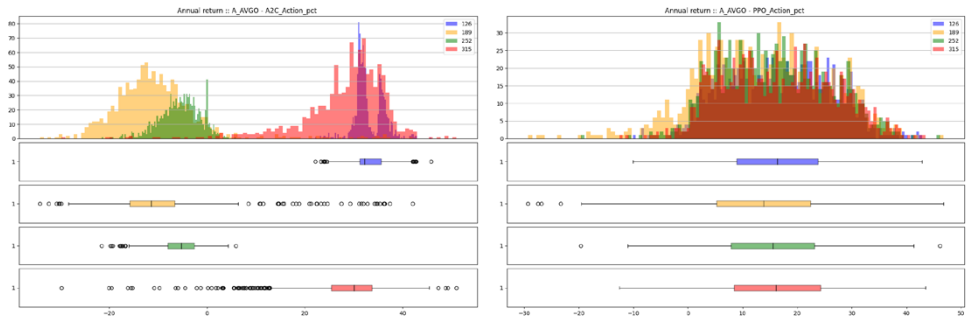
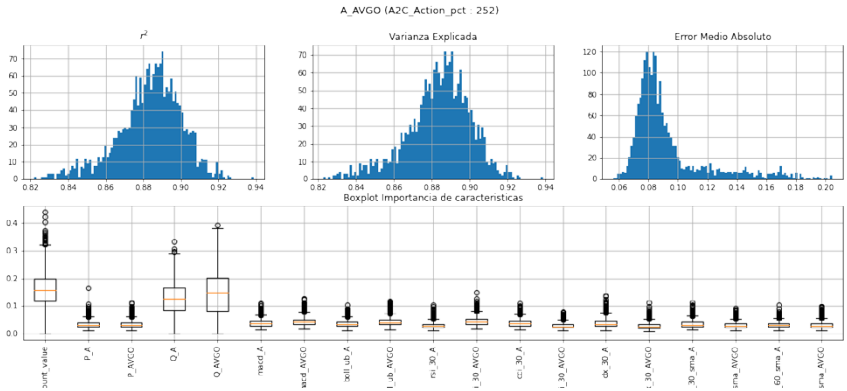


Figura 14. Variables relevantes en el modelo *A2C* en el entorno de acción como proporción del disponible, para el par A-AVGO y ventana 252



5.4. Resultados para el entorno β-balanceado

Este entorno generó también resultados mixtos al igual que el anterior. Pueden apreciarse algunos de ellos en la figura 15. Del mismo modo, las métricas promedio de las ejecuciones de Monte Carlo para el par A-AVGO realizadas se pueden ver en la tabla 9, y para los demás pares en la tabla 8.

5.4.1. Variables relevantes en el modelo

En relación con las variables relevantes para los modelos entrenados bajo el entorno β-balanceado llama la atención que los modelos que generaron en promedio rentabilidades negativas presentaban también dentro de las variables relevantes una mayor ponderación en las variables *Saldo disponible*, *Cantidad del activo A* y *Cantidad del activo B*. (ejemplo de esto puede verse en el la figura 17). Este particular comportamiento parece no ser el único, dado que en una observación del par ACN-NDAQ para la ventana 315 (figura 16), presenta un deterioro en el error medio absoluto, pero no se explica por las cantidades, y, por el contrario, este caso parece tener similares niveles de relevancia en las variables que los casos donde se generan retornos positivos.

Tabla 8. Promedio de medidas de rentabilidad y volatilidad para el par A - AVGO bajo el entorno aprendizaje β-balanceado

Par	Agente	Segmento	Retorno anual (%)	Volatilidad anual (%)	SR	CR	SoR	MD (%)
AAP_SO	A2C	126	-13,45	16,48	-0,91	-0,62	-1,18	-21,46
		189	-20,30	25,13	-0,91	-0,69	-1,19	-30,20
		252	-17,83	13,19	-1,45	-0,91	-1,94	-19,35
		315	-15,47	17,42	-0,94	-0,77	-1,30	-20,50
	PPO	126	-11,93	15,53	-0,77	-0,66	-1,07	-18,18
		189	-4,61	26,96	-0,02	-0,17	-0,03	-22,96
		252	-11,93	16,64	-0,73	-0,65	-1,02	-18,61
		315	-6,76	30,71	-0,07	-0,26	-0,10	-24,52
ACN_NDAQ	A2C	126	-16,29	77,95	0,16	-0,25	0,24	-66,34
		189	-8,78	26,25	-0,22	-0,28	-0,32	-31,07
		252	-11,95	44,38	-0,07	-0,26	-0,10	-46,17
		315	-25,75	12683,45	0,22	-0,24	21,49	-109,24
	PPO	126	-14,82	30,09	-0,42	-0,40	-0,60	-37,05
		189	-14,34	21,78	-0,68	-0,48	-0,95	-30,68
		252	-14,66	33,31	-0,33	-0,37	-0,48	-39,45
		315	-14,65	25,36	-0,55	-0,44	-0,78	-33,69
AIG_CTRA	A2C	126	-29,70	26,85	-1,25	-0,91	-1,66	-32,57
		189	-23,82	27,19	-0,89	-0,91	-1,24	-25,90
		252	-30,19	23,46	-1,42	-0,84	-1,85	-35,81
		315	-19,78	31,69	-0,53	-0,78	-0,79	-24,95
	PPO	126	-25,06	26,26	-1,01	-0,90	-1,39	-27,80
		189	-22,27	24,97	-0,90	-0,89	-1,26	-24,77
		252	-27,09	26,61	-1,09	-0,86	-1,47	-31,16
		315	-26,38	26,75	-1,06	-0,87	-1,44	-29,90
AJG_BRO	A2C	126	-26,24	19,28	-1,52	-1,00	-2,06	-26,17
		189	6,63	16,95	0,37	0,53	0,57	-16,08
		252	-22,25	19,43	-1,22	-0,95	-1,67	-23,36
		315	-30,91	28,75	-1,15	-1,01	-1,56	-30,74
	PPO	126	-22,49	18,74	-1,30	-1,00	-1,76	-22,51
		189	-18,71	17,43	-1,10	-1,00	-1,50	-18,70
		252	-29,42	25,12	-1,29	-1,00	-1,76	-29,35
		315	-19,63	16,37	-1,28	-0,99	-1,74	-19,68
A_AVGO	A2C	126	68,15	233,68	1,13	1,54	2,06	-55,73
		189	-39,00	1029,92	-0,50	-0,64	0,00	-60,61
		252	-28,89	193,56	-0,15	-0,48	-0,10	-58,13
		315	57,23	89,64	1,11	1,64	2,01	-42,99
	PPO	126	-43,06	1012,26	-0,37	-0,65	-0,21	-66,17
		189	-40,35	1228,92	-0,33	-0,62	-0,09	-64,49
		252	-40,04	316,38	-0,48	-0,65	-0,35	-60,93
		315	-42,16	571,29	-0,44	-0,67	0,34	-63,03

SR: Sharpe ratio, CR: Calmar ratio, SoR: Sortino Ratio, MD: Max Drawdown

Figura 15. Rentabilidad acumulada en el par A-AVGO bajo algoritmos A2C (izquierda) y PPO (derecha) usando el entorno de aprendizaje sin posiciones en corto, comparado con el *benchmark* S&P 500

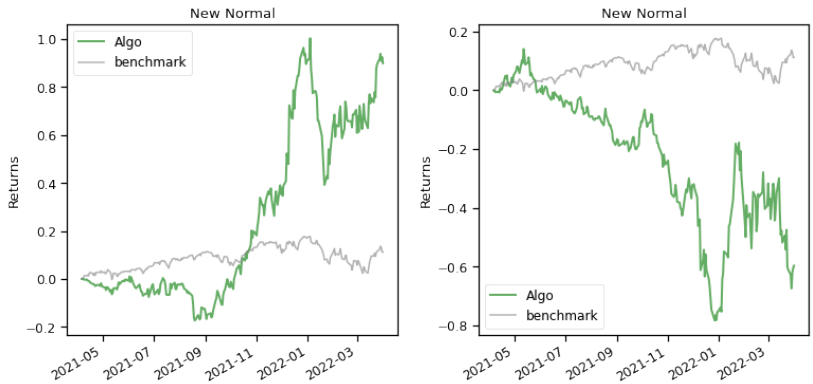


Tabla 9. Promedio de medidas de rendimiento anual para el par A - AVGO bajo el entorno β-balanceado

Agente	Ventana	Retorno (%)	Volatilidad (%)	SR	CR	SoR	MD (%)
A2C	126	68,15	233,68	1,13	1,54	2,06	-55,73
	189	-39,00	1029,92	-0,50	-0,64	0,00	-60,61
	252	-28,89	193,56	-0,15	-0,48	-0,10	-58,13
	315	57,23	89,64	1,11	1,64	2,01	-42,99
PPO	126	-43,06	1012,26	-0,37	-0,65	-0,21	-66,17
	189	-40,35	1228,92	-0,33	-0,62	-0,09	-64,49
	252	-40,04	316,38	-0,48	-0,65	-0,35	-60,93
	315	-42,16	571,29	-0,44	-0,67	0,34	-63,03

SR: Sharpe ratio, CR: Calmar ratio, SoR: Sortino Ratio, MD: Max Drawdown

Figura 16. Variables relevantes para el par ACN-NADQ, usando el algoritmo *A2C* para la ventana 315

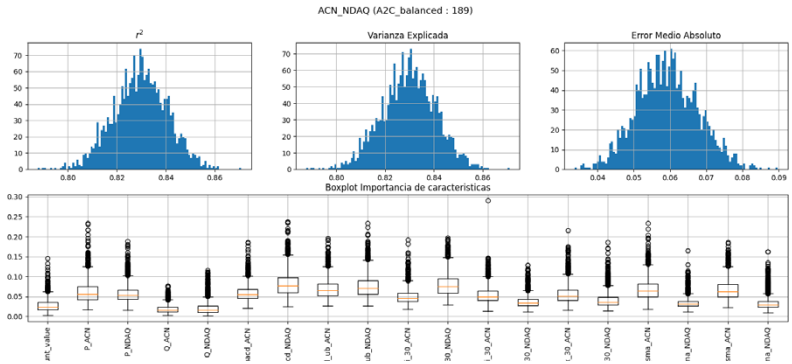
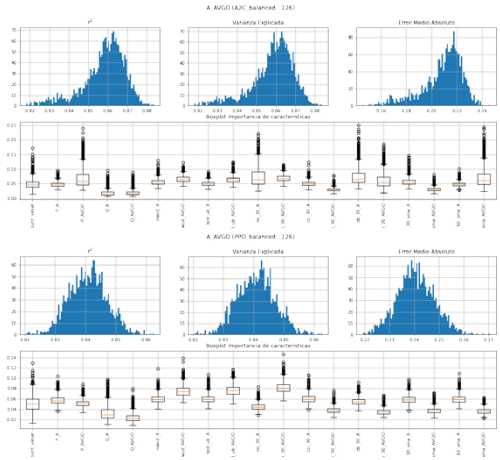


Figura 17. Variables relevantes en el modelo *A2C (izq.)* y *PPO (der.)* para el par A-AVGO, ventana 126



6. Conclusiones

Los modelos de aprendizaje reforzado tienen un soporte cuantitativo bastante robusto y, en muchas ocasiones, comparten metodologías usadas en las finanzas cuantitativas estocásticas, como son la ecuación de Bellman, el método de Monte Carlo y los procesos de decisión de Markov. Sin embargo, aunque tienen marcos de trabajo compartidos, los modelos de aprendizaje reforzado apalancan el motor de aprendizaje mediante el modelado de decisiones óptimas a través del ensayo y error. Esto supone un reto adicional al modelar ambientes de entrenamiento específicos de un problema determinado y a los algoritmos que puedan hacer frente a ese entorno, definido, entre otras cosas, por el tipo de espacio de acciones disponible en el entrenamiento. Esta complejidad se expuso al determinar los hiperparámetros óptimos para obtener una rentabilidad aceptable, cuando fuera posible.

Al analizar el rendimiento del conjunto de modelos entrenados, fue sorprendente notar que los agentes entrenados en entornos sencillos, como el que no permite posiciones en corto, producen una mayor y constante rentabilidad en comparación con los agentes entrenados en entornos con mayor complejidad lógica, como los de proporción del saldo disponible para entrar en posiciones según el conjunto de acciones y β-balanceado. Adicionalmente, se observó que un incremento en el número de pasos por episodio, que generalmente lleva a consumir más tiempo de entrenamiento para el *hardware* usado, no garantiza una mejora considerable en la varianza de la distribución de rentabilidades potenciales en los datos de *trading*, ni es una variable que permita mejorar significativamente la media del retorno u otros indicadores. Esto también se evidenció en los valores de *value loss* y *policy loss*, los cuales se tornaban explosivos y más volátiles después de un cierto número de episodios.

Dicho lo anterior, los agentes presentados en este trabajo no lograron aprender una estrategia de *pair-trading* que supere la rentabilidad generada por el *benchmark* S&P 500, e incluso no logran tener un retorno positivo constante. Esto fue observado tanto para los pares con menor distancia DTW como los seleccionados con mayor distancia DTW del subconjunto expuesto en la sección 2, aun cuando todos los pares procesados aquí son estadísticamente estacionarios bajo las pruebas ADF y KSPP. Sin embargo, cabe destacar que la rentabilidad generada por una estrategia simple como la de desviaciones estándar desde la media no produjo tampoco una rentabilidad superior al 10 % con excepción del par ACN-NDAQ, en el cual los dos entornos entrenados con reglas más complejas produjeron una rentabilidad mixta casi simétrica centrada en cero. No obstante, si fue claro que en orden de rentabili-

dad los entornos fueron, de mayor a menor: sin posiciones en corto, a_i-proporcional y β-balanceado.

Cabe destacar de lo anterior que el presente trabajo desarrolla los entornos de aprendizaje de forma tal que sea el agente el que aprenda a determinar el parámetro β y, así mismo, construir la señal del par, a diferencia de la estrategia base de distancia desde la media donde se construye de forma explícita en el modelo la señal de forma previa.

Hasta el momento, y con los parámetros establecidos y analizados en este trabajo, no se evidenció una notable mejoría en una estrategia *pair-trading* haciendo uso de algoritmos de aprendizaje reforzado como Advantage Actir-Critic (A2C) o Proximal Policy Optimization (PPO), tanto para pares de activos con mínima distancia DTW, como para los de alta distancia DTW. Adicionalmente, se observó que variables relevantes para los modelos tales como saldo disponible en caja y cantidad de activos, bien sea el activo A o el activo B en el portafolio, suelen correlacionarse con rendimientos más negativos y dispersos en el promedio de las simulaciones.

De igual manera, se revisó que no existiera sesgo *forward-looking* especialmente para el entorno sin posiciones cortas, dados los resultados positivos que tuvo este entorno respecto a los demás. Con esto se confirmó que dentro del entorno construido no se evidencia dicho sesgo.

6.1. Trabajos futuros

Si bien el trabajo actual busca exponer si existe o no una oportunidad de obtener un retorno mayor al de desviaciones desde la media en una estrategia *PT* haciendo uso de algoritmos de aprendizaje reforzado, se dejan para trabajos futuros algunos aspectos potenciales que pueden ser profundizados y que no fueron analizados aquí. Entre estos aspectos se encuentran: la selección de un factor distinto al cambio en el valor del portafolio como medida de recompensa en los entornos de entrenamiento; el uso de modelos embebidos basados en alguna métrica como el Sharpe Ratio, el Sortino Ratio o el Calmar Ratio; y el uso de la razón del par de activos como una variable interna del estado de forma explícita, ajustada por alguna métrica de tendencia móvil para hacerla lo más estacionaria posible.

Otro trabajo futuro potencial es implementar los agentes de manera que reciban la señal preconstruida como parámetro de entrada en el espacio de estado de los entornos de aprendizaje. Esto tiene como objetivo hacer más homogénea la comparación entre los modelos de aprendizaje reforzado y el modelo *benchmark*.

Referencias

Ait-Sahala, Y. (2002). Maximum likelihood estimation of discretely sampled diffusions: A closed-form approximation approach. *Econometrica, 70*(1), 223-262.

Bertram, W. (2009). Optimal trading Strategies for Ito diffusion processes. *Physica A: Statistical mechanics and its applications, 338*, 2865-2873.

Bertram, W. (2010). Analytic solution for optimal statistical arbitrage trading. *Physica A: Statistical mechanics and its applications, 389*, 2234-2243.

Carapuco, J., Neves, R. y Horta, N. (2018). Reinforcement learning applied to forex trading. *Applied Soft Computing Journal, 73*, 783-794.

Carr, P. y Wu, L. (2004). Time-changed Levy processes and option pricing. *Journal of financial economics, 1*(71), 113-141.

Carta, S., Corriga, A., Ferreira, A. y Podda, A. (2021). A multi-layer and multi-ensemble stock trader using deep learning. *Applied Science, 51*, 889-905.

Chakole, J., Kolhe, M., Mahapurush, G., Yadav, A. y Kurhekar, M. (2021). A Q-learning agent for automated trading in equity stock markets. *Expert Systems with Applications, 163*, 1-12.

Do, B. y Faff, R. (2010). Does simple pairs trading still work? *Financial Analysts Journal, CFA Institute, 66*(4), 83-95.

Goncu, A. y Akildirim, E. (2016). A stochastic model for commoditiy pairs trading. *Quantitative Finance, 16*(12), 1843-1857.

Haress, E. M. y Yaozhong, H. (2021). Estimation of all parameters in the fractional Ornstein-Uhlenbeck model under discrete observations. *Statistical inference for stochastic processes, 24*, 327-351.

Huang, C. Y. (2018). Financial trading as a game: A deep reinforcement learning approach. *arXiv preprint arXiv, 1807*(02787), 1-15.

Konlack, V. y Wilcox, D. (2014). A comparison of generalized hyperbolic distribution models for equity returns. *Journal of applied mathematics* (15).

Kowalik, P., Kjellevold, A. y Gropen, S. (2019). A deep reinforcement learning approach for stock trading [Tesis de maestría]. Norwegian University of Science y Technology.

Krauss, C. (2015). Statistical arbitrage pairs trading strategies: Review and outlook. En *Working Paper 09*. Institut für Wirtschaftspolitik und Quantitative Wirtschaftsforschung.

Leung, T. y Xin, L. (2016). *Optimal mean reversion: Mathematical analysis and practical applications*. World Scientific.

Liang, Z., Chen, H., Zhu, J., Jiang K. y Li, Y. (2018). Adversarial deep reinforcement learning in portfolio management. En *arXiv*.

Madan, D., Carr, P. y Chang, E. (1999). The Variance Gamma process and option pricing. *European Finance Review, 2*(1), 79-105.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A., Veness, J., Bellemare, M. […] Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature, 518*, 529-533.

Mnih, V., Puigdome`nech, A., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D. y Kavukcuoglu, K. (2016). Asynchronous Methods for Deep Reinforcement Learning. *CoRR*. arXiv, *1602*(01783). http://arxiv.org/abs/1602. 01783

Plaat, A. (2022). *Deep reinforcement learning*. Springer.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A. y Klimov, O. (2017). "Proximal Policy Optimization Algorithms". *CoRR* abs, *1707*(06347). http://arxiv.org/abs/1707.06347

Schwartz, E. (1997). Stochastic behavior of commodity prices: Implications for valuation and hedging. *Journal of Finance, 52*(2), 923-973.

Stock, J. y Watson, M. (1988). Testing for common trends. *Journal of ameri- can Statistical Association, 83*(404), 1097-1107.

Sun, S., Wang, R. y An, B. (2021). Reinforcement learning for quantitative trading. *arXiv ePrint*.

Vidyamurthi, G. (2004). *Pairs trading: Quantitative methods and analysis*. Wiley Finance.

Yang, H., Liu, X., Zhong, S. y Walid, A. (2020). Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy. *SSRN*. 10.2139/ssrn.3690996.

Zeng, Z. y Lee, C. C. (2014). Pairs trading: optimal threshold and profitability. *Quantitative Finance, 14*(11), 1881-1893.