NEGOCIACIÓN ALGORÍTMICA DE ACCIONES POR MEDIO DE APRENDIZAJE POR REFUERZO PROFUNDO

Elaborado por: Santiago Alberto Giraldo Escobar

Director: Fernán Alonso Villa Garzón, PhD

Codirectora: Lina Marcela Cortés Durán, PhD. UNIVERSIDAD EAFIT

Trabajo para optar por el título de Magíster en Ingeniería

MAESTRIA EN INGENIERÍA - INGENIERÍA DE SISTEMAS

UNIVERSIDAD NACIONAL DE COLOMBIA

SEDE MEDELLÍN

2021

TABLA DE CONTENIDO

RESUMEN 4

ABSTRACT 5

1. COMPRENSIÓN DEL NEGOCIO 6

1.1 INTRODUCCIÓN 6

1.2 JUSTIFICACIÓN 7

1.3 OBJETIVO 10

1.4 OBJETIVOS ESPECÍFICOS 10

1.5 REVISIÓN DE LA LITERATURA 10

1.6 CARACTERIZACIÓN DEL MODELAMIENTO DE SERIES DE TIEMPO MEDIANTE ARQUITECTURAS GAN 12

1.7 CARACTERIZACIÓN DE LOS MODELOS DE APRENDIZAJE POR REFUERZO PROFUNDO 17

1.8 EL APRENDIZAJE POR REFUERZO APLICADO A LAS FINANZAS 21

2. ENTENDIMIENTO DE LOS DATOS 22

2.1 PREPROCESO DE LOS DATOS 26

2.2 ANÁLISIS EXPLORATORIO DE DATOS 28

2.3 SELECCIÓN DE REGLAS DE NEGOCIACIÓN O TRADING 31

2.3.1 ESTRATEGIA DE PROMEDIO MÓVIL SIMPLE 31

3. MODELAMIENTO DE LOS DATOS 36

3.1 GENERACIÓN DE DATOS SINTÉTICOS MEDIANTE UN MODELO GAN 36

4. IMPLEMENTACIÓN DEL ALGORITMO DE APRENDIZAJE POR REFUERZO PARA COMPRA DE ACCIONES. 49

5. EVALUACIÓN DEL MODELO 52

6. CONCLUSIONES 66

6.1 CUMPLIMIENTO DE OBJETIVOS 66

6.1.1 OBJETIVO ESPECÍFICO 1. 66

6.1.2 OBJETIVO ESPECÍFICO 2. 66

6.1.3 OBJETIVO ESPECÍFICO 3. 66
6.1.4 OBJETIVO GENERAL 67
6.2 POSIBLES USOS DEL APRENDIZAJE POR REFUERZO PROFUNDO APLICADO A NEGOCIACIONES ELECTRÓNICAS 68
7. BIBLIOGRAFÍA Y FUENTES DE INFORMACIÓN 68

RESUMEN

Este trabajo de grado tiene como finalidad explorar la utilización de series de tiempo financieras sintéticas generadas por un modelo de Redes Neuronales Generativas Adversarias (GAN por sus siglas en inglés) para entrenar un algoritmo de Aprendizaje Profundo Q Por Refuerzo que ejecute acciones de compra y venta para un título del mercado de valores del índice de Standard & Poor's 500. Para el desarrollo del trabajo se empleó la metodología CRISP DM propuesta por IBM, entendiendo primero el negocio y la teoría necesaria para desarrollar los modelos, para continuar con la exploración y conocimiento de los datos disponibles que concordaran con los objetivos del estudio. En este se desarrolla un procedimiento para la selección de series ficticias y para el entrenamiento de un algoritmo por refuerzo con estos datos. Se utiliza la métrica de Kolmogorov - Smirnov como componente esencial para entrenar las redes GAN. Se explican los resultados de los experimentos, y se evidencia la dificultad para calibrar modelos generativos adversarios y de agentes entrenados por refuerzo. Por último, se presentan las conclusiones derivadas del trabajo y posibles investigaciones futuras.

Palabras claves: Aprendizaje profundo; Aprendizaje por refuerzo profundo; Redes neuronales generativas adversarias; Negociación algorítmica; Aprendizaje de máquina; Negociación de acciones;

ALGORITHMIC STOCK TRADING THROUGH DEEP REINFORCEMENT LEARNING

ABSTRACT

This degree work aims to explore the use of synthetic financial time series generated by a Generative Adversarial Neural Networks (GAN) model to train a Deep Reinforcement Learning algorithm that executes buy and sell actions for a stock in the Standard & Poor's 500 index. For the implementation of the study, we used the CRISP methodology proposed by IBM, understanding first the business and the theory necessary to develop the models, to continue with the exploration and knowledge of the available data that matched the objectives of the project. In this paper, a procedure for selecting synthetic series and training a reinforcement algorithm with these data is developed. The Kolmogorov-Smirnov metric is used as an essential component to train GANs. The results of the experiments are explained, and the difficulty in calibrating generative adversarial and reinforcement network models is shown. Finally, conclusions derived from the project and possible future research are presented.

Key words: Deep learning; Deep reinforcement learning; Generative Adversarial Networks; Algorithmic trading; Machine learning, Stock trading;

1. COMPRENSIÓN DEL NEGOCIO

1.1 INTRODUCCIÓN

Una negociación algorítmica (conocida también como trading algorítmico o comercio algorítmico) es un conjunto de reglas diseñadas para ejecutar acciones de compra o venta de un activo o producto financiero en un mercado electrónico [1][2]. Se caracterizan por ser aplicaciones, que operan o ejecutan acciones según la lectura que se haga de las condiciones de mercado en el que operan [2]. Una alternativa a los programas de reglas empleados en las negociaciones algorítmicas tradicionales es la metodología propuesta por un campo de la inteligencia artificial llamada Aprendizaje por Refuerzo, y que resulta atractiva para ser aplicada en la negociación de activos financieros, pues permite desarrollar sistemas expertos especializados en la toma de decisiones de inversión en tiempo real en mercados financieros [3].

En un modelo de Aprendizaje por Refuerzo, un agente entrena para aprender a reconocer en qué momento es bueno o malo ejecutar una acción (comprar, vender o retener un activo financiero) en un mercado financiero electrónico, lo que posibilita a dicho agente a tomar decisiones de forma secuencial a medida que la información actualizada le llega de la situación de su entorno (noticias, volúmenes, retornos, etc.) en tiempo real [3]. Uno de los inconvenientes de los modelos de Aprendizaje por Refuerzo es que requieren de una ingesta grande de datos para entrenar el modelo [4], y en finanzas, este problema se hace más complejo, pues los entornos de entrenamiento no son los adecuados dadas las características propias de las series de tiempo financieras [3].

Generalmente, los métodos para crear trayectorias sintéticas de series de tiempo que alimentan el entorno del agente tienen que ver con el remuestreo de series históricas (*Boostraping*) o con la simulación de Monte Carlo [3][5]. El primero restringe los datos a información conocida, y asume que el futuro es igual al pasado, y el segundo, generalmente, utiliza modelos de caminatas aleatorias geométricas o modelos econométricos como los denominados ARMA o ARIMA para citar algunos, que asumen la normalidad de los datos y la independencia en la distribución de los retornos de la serie [8]. Como alternativa y por su facilidad para modelar relaciones no lineales, se han propuesto métodos de Aprendizaje por Refuerzo con redes neuronales profundas, integradas en el proceso de entrenamiento del algoritmo.

Pero estas técnicas también presentan inconvenientes como los de asumir que la distribución es idéntica y estacionaria, ocasionando sobreentrenamiento o sobreajuste [4].

De igual forma, algunas publicaciones de López de Prado [5][6][7], quien ha sido uno de los investigadores que abogan por el uso de nuevas técnicas en las finanzas, como es el caso del aprendizaje de máquina y de su uso correcto. Ha criticado el uso de metodologías como la validación cruzada como herramienta de entrenamiento para el análisis de series de tiempo financieras, pues esta técnica emplea el remuestreo para entrenar un algoritmo de Aprendizaje de Máquina [7]. Es este punto el que lleva a plantear la pregunta del problema del trabajo, y que a su vez conduce a explorar la validez de utilizar modelos generativos como alternativa para suplir la falta de métodos en la generación de datos sintéticos adecuados [3], diferentes a las medidas de remuestreo o de simulación de Monte Carlo con modelos de Redes Generativas Adversarias.

De lo anterior se desprende la siguiente pregunta: ¿Es posible encontrar una técnica o método que permita incrementar los ejemplos simulados en un modelo de Aprendizaje por Refuerzo que preserve las dinámicas temporales de las series de tiempo reales para entrenar un agente de negociación algorítmica?

1.2 JUSTIFICACIÓN

La negociación algorítmica se refiere a programas que realizan de manera automática o semiautomática parte del ciclo de negociación de productos financieros en mercados electrónicos. Se utilizan para diversas tareas, como optimizar ejecución de órdenes, negociaciones de alta frecuencia, optimización o rebalanceo de portafolio, y se han vuelto relevantes porque toman decisiones importantes como el monto a invertir o vender, el precio del activo o el momento de ejecutar una orden en el mercado. Estos programas modelan las condiciones y características del mercado en donde operan, y normalmente, son una serie de líneas de código que reflejan la idiosincrasia, sesgos, conocimientos y restricciones políticas o de riesgo de quienes los programan, y a veces son de una complejidad tal, que los hacen muy difícil de mantener y actualizar [2][3][8][9]. Los procesos de la negociación algorítmica pueden ser modelados con técnicas de programación dinámica, pues estos se dan en contextos de toma de decisiones secuenciales, pero

al momento de implementarlos presentan obstáculos, porque los supuestos necesarios de un conocimiento preciso del entorno y la necesidad de tener los recursos computacionales adecuados para ponerlos en práctica, no se dan en la realidad [3][11].

De forma alternativa, el aprendizaje de máquina se utiliza para el diseño de aplicaciones de negociación algorítmica de activos, y se emplea en sus diferentes formas (Aprendizaje Supervisado y No Supervisado) como un proceso estático de dos etapas [9][10]. En la primera etapa se desarrolla un modelo que utiliza información histórica (Máquinas de Soporte Vectorial, Xgboost, Aprendizaje Profundo, etc.) para predecir cuándo es el momento adecuado para ejecutar una acción o no de una negociación, mientras que la segunda etapa, involucra un sistema de negociación que ejecuta la acción según la predicción del modelo de aprendizaje [10]. A pesar de su popularidad, este enfoque tiene varios inconvenientes que son relevantes en el momento de ejecución de los programas, como pueden ser que la minimización del error no necesariamente esté en línea con la meta del inversionista de maximizar las utilidades, o que en el momento de modelar el algoritmo se descarte información significativa, o que existan restricciones exógenas como la falta de liquidez y los costos de transacción y que éstos no sean considerados en su implementación [10].

Dadas las características del Aprendizaje por Refuerzo, las metodologías agrupadas por esta etiqueta se han convertido en métodos promisorios para resolver los problemas antes mencionados, pues dichas técnicas buscan aprender a ejecutar acciones de acuerdo con una situación dada por una señal numérica de recompensa. El agente aprende haciendo, ejecuta acciones y evalúa el resultado de éstas, con la ventaja de que también recompensa la acción actual, y a través de ella, las acciones posteriores [12]. Las características de ensayo y recompensa son extremadamente útiles para los problemas secuenciales que se presentan en las finanzas [11]. En el caso de las negociaciones algorítmicas, el agente tratará de maximizar las ganancias al llevar a cabo acciones de comprar o vender en un mercado simulado [9].

Uno de los principales retos en el Aprendizaje por Refuerzo en general y más cuando es aplicado en las finanzas, es la posibilidad de construir un simulador que represente de mejor manera la realidad [5][13], y es aquí en donde las Redes

Generativas Adversarias se hacen interesantes para afrontar la problemática anteriormente planteada [9].

Esta situación no es exclusiva del Aprendizaje por Refuerzo, de hecho, López de Prado [14] lo menciona como uno de los problemas de las finanzas empíricas, pues los investigadores no cuentan con modelos que representen las condiciones de mercado en ambientes controlados como en la física [5]. De ahí la importancia de generar datos sintéticos que permitan hacer análisis en entornos simulados de escenarios que sean cercanos a lo que pasa en la realidad [15].

Una alternativa para el entrenamiento de los modelos de Aprendizaje por Refuerzo son los modelos de Redes Generativas Adversarias [9][15] (GAN por su sigla en inglés). Estas arquitecturas se caracterizan por aprender los parámetros de un conjunto de datos existentes y ser capaces de reproducir los hechos estilizados de la distribución empírica de la serie de tiempo real como lo son los agrupamientos o clústeres de volatilidad, los efectos de apalancamiento o autocorrelaciones seriales, características que son difíciles de reproducir con los modelos econométricos tradicionales [17]. Es así como a partir de lo escrito por Karpe [9], Assefa *et al.* [15], Hansen [16] y Wiese *et al.* [17], se infiere que los modelos de GAN son una aproximación adecuada para solucionar la falta de datos confiables como insumo del entorno de entrenamiento de modelos de Aprendizaje por Refuerzo.

En este estudio se enlaza el método de Aprendizaje por Refuerzo para el entrenamiento de un modelo de inteligencia artificial con una Red Generativa Adversaria, encargada de la generación de datos sintéticos para el entrenamiento del agente. Ambas metodologías se aplican en el contexto de las finanzas, más concretamente en el de las negociaciones algorítmicas uniendo dos contextos de dominio: primero, el de las finanzas cuantitativas, que busca ejecutar órdenes de venta o compra de productos financieros en mercados electrónicos; segundo, el de las ciencias de la computación, que busca hacer la ejecución de las órdenes de manera automática o semiautomática.

Con base en la revisión y análisis realizado a continuación se plantean los objetivos de este trabajo.

1.3 OBJETIVO

Desarrollar un modelo de Redes Generativas Adversarias como entorno de entrenamiento para un algoritmo de Aprendizaje por Refuerzo aplicado en el contexto de la negociación algorítmica.

1.4 OBJETIVOS ESPECÍFICOS

- Caracterizar las metodologías de Redes Generativas Adversarias y de Aprendizaje por Refuerzo empleadas para la negociación algorítmica.

- Desarrollar un modelo de Aprendizaje por Refuerzo Profundo para negociación algorítmica que incluya una arquitectura de Redes Generativas Adversarias para el entrenamiento del agente.

- Validar la modelación propuesta con datos de una acción de un mercado de acciones electrónico.

1.5 REVISIÓN DE LA LITERATURA

La información sobre la negociación algorítmica es abundante en la literatura de divulgación, en especial aquella que se refiere a “cómo hacerse rico”, pero es limitada sobre revisiones de literatura académica al respecto. Hasta donde se ha podido corroborar, la revisión de las negociaciones algorítmicas publicada por *Communications of ACM* [8] es el mejor resumen sobre este tópico de las finanzas. Dicho estudio está compuesto de una definición de términos utilizados en la jerga financiera relacionados con las negociaciones algorítmicas, explora las estructuras y características de los datos financieros, resume los diferentes tipos de modelos que se implementan en este campo, explica las arquitecturas empleadas y las pruebas de desempeño para validar resultados de los mismos en la práctica. En general, el artículo da una amplia visión de cómo los algoritmos funcionan en las finanzas de los mercados de capitales y es un buen punto de partida para entender la utilidad y riesgos de estos para el presente trabajo.

En cuanto a la aplicación de Aprendizaje de Máquina y Aprendizaje Profundo en las finanzas, la literatura es diversa, en parte, por la novedad que se ha dado en los últimos años por su aplicación en todo tipo de cosas. Ejemplos de resúmenes se encuentran en [16], [18], [19], [20], [21], [22], y [23] que en general son similares, solo que cambian el contexto en donde se aplica la revisión de la literatura. Estos estudios, básicamente comparan metodologías tradicionales econométricas y de series de tiempo con técnicas recientes en diferentes contextos de las finanzas, como la optimización de portafolio, análisis de riesgos, valoración de activos, predicción de precios y análisis de sentimientos. Los modelos van desde técnicas tradicionales como las Máquinas de Soporte Vectorial, pasando por las Redes Convolucionales a modelos recurrentes como los LSTM.

Las Redes Generativas Adversarias han ganado mucha popularidad en años recientes, dada su capacidad de generar datos inexistentes y creíbles como rostros de personas o videos falsos [28][29][30], y por esto, investigadores de diversos campos académicos han empezado a examinar las posibilidades de generación sintética de datos con este tipo de arquitecturas, que produzcan escenarios nuevos de series de tiempo, pero que a su vez, sean capaces de reproducir los hechos estilizados o propiedades estadísticas de dichas series [9] [15] [16] [17] [28] [29] [30] [31].

Las redes GAN fueron propuestas originalmente por Schmidhuber en 1990 como un caso es especial de “Redes Adversarias Curiosas” y fueron desarrolladas en un contexto de Aprendizaje por Refuerzo [32], pero fueron popularizadas posteriormente por Goodfellow [33][34] a quien usualmente se le da al crédito por el invento de esta arquitectura, pues el trabajo de Schmidhuber no fue conocido hasta que la propuesta de Goodfellow fue popular en los medios académicos[1].

El Aprendizaje por Refuerzo, junto con el Aprendizaje Supervisado y Aprendizaje No supervisado conforman el Aprendizaje de Máquina [12]. Que es un área del conocimiento de las ciencias de la computación que estudia la forma de encontrar patrones en espacios multidimensionales sin ser dirigidos específicamente [22].

[1] https://analyticsindiamag.com/lstm-juergen-schmidhuber-artificial-intelligence-switzerland-goedel-machines/

El Aprendizaje Supervisado busca aprender a partir de conjuntos de ejemplos etiquetados por un supervisor externo. Una vez el modelo es entrenado, éste es utilizado para predecir (extrapolar) a que etiqueta corresponde los nuevos datos nuevos ejemplos. Los ejemplos supervisados pueden ser del tipo Regresión o Clasificación. En el primer caso se da cuando lo que se busca predecir es un valor real (continuo), y en el segundo caso, se presenta cuando se requiere encontrar a que categoría o clase el conjunto de ejemplo pertenece el dato evaluado [12].

El Aprendizaje No Supervisado, aprende de las mismas características de los datos ocultas en los mismos, no utiliza etiquetas para catalogar los datos, en lugar de eso busca similitudes entre los datos y de esta manera generar conjuntos similares. Típicamente a estas técnicas también se les conoce como algoritmos de Agrupamiento, Segmentación o *Clustering* [12].

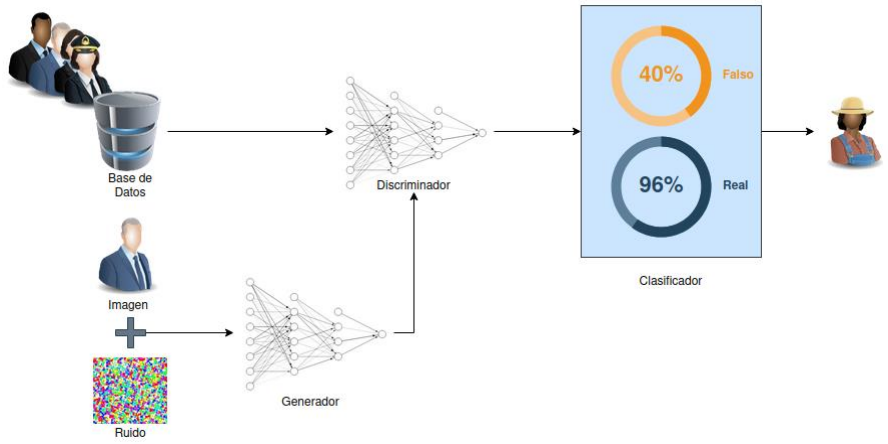
El Aprendizaje por Refuerzo es diferente de las otras dos técnicas. En este caso, el aprendizaje se da por medio del ensayo y error. En pocas palabras, el algoritmo aprende por interacción con un entorno, que es evaluado por una función que le indica cual es la mejor respuesta o acción a unas condiciones predefinidas. Aquí la compensación (*trade-off*) entre la explotación y exploración es un asunto importante. El agente debe utilizar (explotar) lo aprendido, pero debe encontrar (explorar) nuevas decisiones para maximizar la recompensa esperada [12].

1.6 CARACTERIZACIÓN DEL MODELAMIENTO DE SERIES DE TIEMPO MEDIANTE ARQUITECTURAS GAN

Una red GAN en esencia es un modelo en el que dos agentes (arquitecturas de redes neuronales) compiten entre sí en un juego teórico de suma cero. Ambas redes son alimentadas con la misma información y no son atadas a ningún tipo de variable dependiente que ayude a las redes a encontrar el mejor patrón (es por esta razón que se clasifica este tipo de modelos como Aprendizaje No Supervisado). Una de las redes recibe el nombre de generador y a la otra se le llama discriminador (o adversaria). La primera recibe un dato de la base de datos (imagen o serie de tiempo, por ejemplo) el cual es alterado al aplicarle ruido de forma aleatoria, y de esta manera modificar la naturaleza de la información, que va a permitir entrenar el

generador en su intento de crear un dato similar al que alimento la red, pero sin ruido [17].

El resultado del entrenamiento de la primera red alimenta la segunda red que ha sido entrenada con los datos originales o sin transformación y que es la encargada de realizar un proceso de clasificación. El objetivo del discriminador, tal y como su nombre lo dice, es discriminar (clasificar) si la información suministrada por el generador es real o falsa [29][30]. En otras palabras, el generador debe tratar de generar datos que el discriminador debe determinar si pertenece o no a la muestra (misma distribución del conjunto de datos u objetivo) [17]. En el Gráfico 1, se muestra el esquema de cómo funcionan las Redes Generativas Adversarias en el caso de generación de imágenes sintéticas.



Fuente: elaboración propia.

Gráfico 1. -Esquema del algoritmo GAN para generación de imágenes.

En cuanto a la modelación de series de tiempo financieras con modelos GAN se ha escrito relativamente poco, es posible encontrar las investigaciones de Wise *et al.* [17], Takahashi *et al.* [29], Fu *et al.* [30] y Villarraga [31].

El proceso para series de tiempo es similar al aplicado en otros contextos como el de visión por computador, pero a su vez presenta otras dificultades por las características intrínsecas de las series de tiempo financieras: el primero, es que lo que se modela es la variación de los precios con respecto a un intervalo de tiempo

dado, ratio que se le conoce como retorno o rendimiento del activo financiero; el segundo, es que el ratio tiene unas características propias que en la literatura financiera se conoce como hechos estilizados [17] y que se refieren a que las series de los rendimientos poseen las siguientes propiedades [17] [29] [30]:

- Los retornos presentan colas pesadas, es decir valores extremos.
- Los retornos presentan leptocurtosis en su distribución, dado a que la mayoría de los datos tienden a concentrarse casi en 0.
- A través del tiempo se presentan clústeres de volatilidad, porque no es constante.
- Los retornos presentan un efecto de palanca, es decir, la volatilidad esta negativamente correlacionada con los retornos.
- La distribución empírica de los retornos se considera no correlacionada, pero tampoco se considera que estos sean independientes.

Takahashi *et al.* [29] definen un modelo GAN como la optimización minmax aplicada a la Ecuación 1.

$$\min_{\boldsymbol{G}} \max_{\boldsymbol{D}} \boldsymbol{V}\,(\boldsymbol{D},\boldsymbol{G}) = \mathbb{E}_{\boldsymbol{x}\sim \boldsymbol{p}_{\boldsymbol{data}(\boldsymbol{x})}}\,[\boldsymbol{log}\boldsymbol{D}(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\,\boldsymbol{p}_{\boldsymbol{z}}(\boldsymbol{z})}[\boldsymbol{log}(\mathbf{1}-\boldsymbol{D}(\boldsymbol{G}(\boldsymbol{z})))]$$

Ecuación 1. Función de Costo de un Modelo GAN

En donde:

D: Discriminador.

G: Generador.

x: Variable aleatoria objetivo (Dato).

z: Variable aleatoria de ruido apriori.

El proceso de optimización ocurre en dos etapas, en la primera el discriminador aprende a distinguir los datos reales de los falsos y en la segunda el generador busca que los datos generados sean identificados como reales [17].

En cuanto al tipo de arquitecturas utilizadas en la literatura se encuentran diversos enfoques, Wise *et al.* [17], propone una arquitectura de varios bloques compuestos de dos capas convoluciones temporales o convoluciones con dilatación ***d*** de una dimensión, esto es, convoluciones con huecos que permiten modelar dependencias de largo plazo (memoria), la dimensión ***d*** es el hiperparámetro que determina el "hueco" o rezago entre las capas, permitiendo que se conserve información en la secuencia de la serie de tiempo. Takahashi *et al.* [29] proponen tres modelos: multiperceptrón, convolucional y uno mixto. Fu *et al.* [30] proponen un modelo de Red Generativa Adversaria Condicional o CGAN por sus siglas en inglés. Villarraga [31] hace una revisión de los modelos implementados por Fu *et al*. y otros aplicando las diferentes metodologías a diversas series de tiempo de periodicidades distintas (diarias e intradiarias).

Una parte fundamental de un algoritmo GAN es encontrar una medida de evaluación que permita saber si el generador crea datos sintéticos realistas con respecto al discriminador. Normalmente, cuando se optimiza un modelo de redes neuronales, lo que se hace es minimizar una función de costo hasta que este converge, el asunto con los modelos generativos es que emplean dos redes que compiten entre sí y las cuales deben mantener una suerte de equilibrio entre ellas, por lo que no existe una función objetivo única a optimizar. Brownlee [34] hace un resumen extenso acerca las medidas empleadas para evaluar los generadores, pero éstas se refieren a medidas aplicada en general a modelos enfocados en la visión por computador, y es algo que se está modificando permanentemente en la literatura especializada.

En cuanto a los modelos enfocados en generar series de tiempo sintéticas no existe un consenso acerca de cuál es la mejor metodología para determinar la calidad de la serie sintética, se encontró que Wise *et al.* [17] utilizan dos medidas para evaluar los resultados de los modelos que son la distancia Wasserstein-1 o distancia del movimiento de la tierra (conocida también como métrica de Kantorovich - Rubinstein[2]) y la métrica Drăgulescu - Yakovenko (conocida también como métrica DY). El primer evaluador, busca medir la distancia entre las masas de dos distribuciones conjuntas, parámetros que serían minimizados en el proceso de generación de datos sintéticos. La segunda medida computa la función de probabilidad de densidad de los retornos de las distribuciones históricas y generadas en un segmento diferenciado ***t***. Takahashi *et al.* [29] no hacen referencia alguna a un tipo de medida en específico, más allá de evaluar los hechos estilizados

[2] https://es.abcdef.wiki/wiki/Wasserstein_metric

de las series sintéticas una vez generadas, y no indican como son utilizados para evaluar la selección de un modelo. Fu *et al.* [30] utiliza la actualización de la función de costo maximizando la probabilidad de que las series sintéticas sean reales, mientras que Villarraga [31] emplea medidas de precisión (*accuracy*).

Para este trabajo se implementan las medidas de precisión y una medida adicional que es empleada normalmente en el análisis de series de tiempo y de Monte Carlo, que es la métrica de Kolmogorov - Smirnov (KS) [36][37]. Esta prueba no paramétrica, compara dos distribuciones de probabilidad acumuladas evaluando la bondad de ajuste entre ellas. Básicamente, mide la distancia entre ambas distribuciones, la teórica y la empírica. La hipótesis nula (**H_0**) para esta prueba dice que no existe diferencia entre la distribución teórica y la distribución empírica, mientras que la hipótesis alternativa (**H_1**) plantea que existe una diferencia entre ambas distribuciones. El valor crítico que normalmente se emplea para esta prueba y que rechaza la hipótesis nula es **α** = 0.05, por lo tanto, se debe rechazar la **H_0** si el valor **p** obtenido en la prueba es menor que **α** [38].

Con base en lo anterior se establece para el estudio el siguiente procedimiento para evaluar una arquitectura de redes neuronales GAN:

1. Selección de una muestra de precios históricos de una acción.
2. Transformación de los retornos de la acción.
3. Estandarización de la serie de los retornos de la acción.
4. Selección de una arquitectura específica GAN y correr ésta con los retornos estandarizados de la acción.
5. Evaluación de la precisión para el modelo discriminador y generador. Aquí se define como umbral un 90% de precisión para cada red, es decir, ambos modelos deben ser capaces de reconocer cuando una serie es real o falsa el 90% de las veces[3].
6. Generación de n series sintéticas con la arquitectura actual. El tamaño de la muestra lo definirá la capacidad computacional disponible para entrenar al agente.

[3] No se encontró ninguna recomendación al respecto, pero lo que se busca con este umbral es garantizar que la generación de datos sea lo más realista posible, y que no sea tan preciso que produzca casi ningún o ningún modelo GAN, pero que garanticen la creación de datos con las propiedades esperadas.

7. Construcción de la distribución de probabilidad acumulada (DPA) para toda la muestra.
8. Evaluación la DPA con el estadístico KS.
9. Si el modelo cumple 5 y 8 es guardado, para ser utilizado posteriormente en la generación de datos sintéticos, caso contrario se pasa a entrenar otro nuevo modelo GAN.

1.7 CARACTERIZACIÓN DE LOS MODELOS DE APRENDIZAJE POR REFUERZO PROFUNDO

Un algoritmo de Aprendizaje por Refuerzo está compuesto de dos elementos como lo son el agente y el entorno. En esencia un agente es un algoritmo (programa) que toma las decisiones para o en el modelo, es el que interactúa con lo que le rodea y aprende con base en esas interacciones, es el aprendiz y el decisor [12]. Éste es el que utiliza la experiencia y explora nuevas acciones para maximizar los resultados. Mientras que el entorno es todo con lo que interactúa el agente para tomar una decisión. Los componentes físicos, de información o de programación de un problema de aprendizaje hacen parte del entorno. Por ejemplo, en un vehículo autónomo, la dirección de éste o el programa que la mueve puede pensarse como parte del agente, pero en realidad es parte del entorno pues no toma ninguna decisión, ésta la toma el agente.

El agente ejecuta una acción ($\boldsymbol{a_t}$), que es predefinida con anterioridad, y que es la forma como el agente interactúa con el entorno, en otras palabras, es la "materialización" de la decisión del agente. La acción ocurre en un tiempo **t**, al que se le llama paso o estado ($\mathbf{s_t}$), en donde el agente tiene a su disposición cierta información del entorno que solo es visible o parcialmente visible en el momento **t**, y a partir de ésta, el agente ejecuta una acción. Al agente no le importa la información anterior del paso o estado previo, es por esta razón que al Aprendizaje por Refuerzo se le enmarca formalmente como un Proceso de Decisión de Márkov Finito (MDP por sus siglas en inglés) [12].

El agente debe tener criterios o elementos que le permitan evaluar que está tomando las decisiones adecuadas y acordes con lo que se le pide que aprenda el algoritmo. Esta evaluación se da por los siguientes elementos del algoritmo como lo son la política ($\boldsymbol{\pi}$), la recompensa ($\mathbf{R_t}$) y la función de valor ($\mathbf{V_t}$).

Una política es un mapeo del estado del entorno percibido para el agente, y que permite definir la forma en como el agente se debe comportar en ese estado. Ésta puede ser una función o una tabla de búsqueda, y se pueden presentar de manera determinística o estocástica especificando probabilidades para cada acción recomendada [12]. Matemáticamente se puede expresar la acción en función de la política como la Ecuación 2.

$$\boldsymbol{a_t = \pi(s_t)}$$

Ecuación 2. Acción del agente en el tiempo t.

Donde:

$\boldsymbol{a_t}$: Es la acción se toma en tiempo **t**

$\boldsymbol{\pi}$: Es la política que mapea la información disponible en el momento **t**.

$\boldsymbol{s_t}$: Es el estado o información observable en ese instante de tiempo **t**.

Las señales de recompensa definen el objetivo del problema, dicen que es bueno o malo para el agente. Su importancia radica en que define una base para alterar la política basado en que tan buena es o no una recompensa. La señal de recompensa, así como la política puede ser carácter discreto o estocástico [12].

La recompensa es el resultado (un valor numérico) dado por el entorno como reacción a la ejecución de la acción del agente. El objetivo último del agente es maximizar la recompensa $\mathbf{G_t}$ [12], ésta se representa mediante la Ecuación 3.

$$\boldsymbol{G_t = R_{t+1} + \gamma R_{t+2} + \cdots = \sum_{0}^{\infty} \gamma^k R_{t+k+1}}$$

Ecuación 3. Recompensa del agente.

Donde:

$\boldsymbol{G_t}$: Es el valor acumulado de la recompensa.

$\boldsymbol{g_t}$: Es el coeficiente de descuento que penaliza la recompensa futura, y que se define como un valor entre 0 y 1.

$\boldsymbol{R_t}$: Recompensa en el tiempo **t**.

La función de valor se encarga de sumar la cantidad total de recompensa en un estado, y que un agente puede esperar acumular en el futuro, comenzando desde el estado evaluado ($\mathbf{s_t}$) y que generalmente no es visible, prediciendo las recompensas futuras. La diferencia del valor de la función con la recompensa es que ésta última es un valor inmediato de lo generado por la acción del agente en el estado **t**, mientras que el valor de la función tiene en cuenta el valor del estado evaluado actual y los valores de los estados futuros [12]. Una función de valor se representa como la Ecuación 4.

$$\boldsymbol{V(s) = \sum_{a \in A} Q(s,a)\pi(a|s)}$$

Ecuación 4. Función de valor del agente.

Donde:

Q(s,a): Es la función estado-acción, y es el valor esperado de la recompensa $\boldsymbol{G_t}$ dado el estado $\mathbf{s_t}$ y la acción $\boldsymbol{a_t}$.

π(a|s): Política según la acción dado un estado ***s***.

De acuerdo a Sutton *et al*. [12], la estimación eficiente de este valor (Q) es el concepto más relevante en el Aprendizaje por Refuerzo, pues debe ser estimado y reestimado en la medida que se van evaluando las secuencias de observaciones hechas por el agente durante su tiempo de vida. Este valor se puede expresar de manera más formal como la Ecuación 5.

$$\boldsymbol{Q(s,a) = \mathbb{E}[G_t|S_t = s, A_t = a]}$$

Ecuación 5. Función de valor *Q* en función de una acción *a* en un estado *s*.

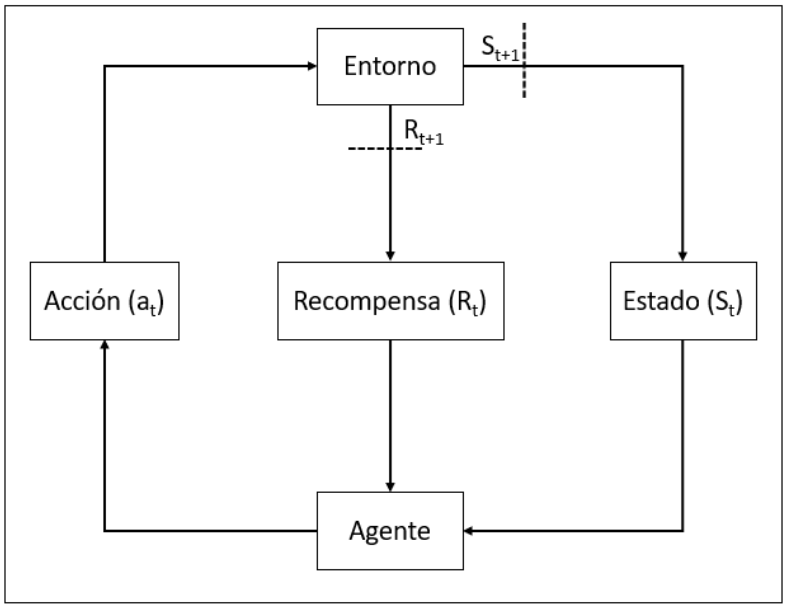
El agente, el entorno, la política y la función de valor se agrupan en un modelo, con el cual se busca imitar el comportamiento del entorno, y permite hacer inferencias de cómo debe comportarse el agente. Se usan para la planeación del curso de acción del agente. A este tipo de modelos se les llama métodos basados en modelos, y cuándo el algoritmo es diseñando para que ejecute acciones por ensayo y error se le llama libre de modelos [12].

Luego de algunas manipulaciones algebraicas se llega a la siguiente Ecuación 6:

$$\boldsymbol{V(s)} = \mathbb{E}[\boldsymbol{R_{t+1}} + \boldsymbol{\gamma V(S_{t+1})} | \boldsymbol{S_t} = \boldsymbol{s}]$$

Ecuación 6. Ecuación de retorno óptimo o Ecuación de Bellman.

Ésta es la ecuación de Bellman [39], y representa un sistema recursivo, que va aprendiendo en cada iteración del proceso de entrenamiento, y que es la piedra angular en el algoritmo de Aprendizaje por Refuerzo. La Ecuación 6 se resuelve mediante programación dinámica, en un proceso iterativo que maximiza la recompensa del agente, y que le permite al agente aprender de manera secuencial las mejores acciones dado un estado del entorno. En el Gráfico 2 se muestra de manera esquemática cómo funciona el proceso iterativo de un algoritmo de Aprendizaje por Refuerzo clásico [12].



Fuente: Sutton *et al.*

Gráfico 2. Esquema de un algoritmo de Aprendiza por Refuerzo.

El ejercicio entonces, en un problema de Aprendizaje por Refuerzo, es optimizar la función de valor y la función de política, la forma como se afronte el problema definirá que tipo de algoritmo se empleará para entrenar al agente; si éste se

concentra en la primera opción se les llama métodos basados en función de valor, y en el otro caso, se les llama métodos de búsqueda de política, un hibrido entre los dos métodos se les llama actor-crítico [40].

1.8 EL APRENDIZAJE POR REFUERZO APLICADO A LAS FINANZAS

La bibliografía de Aprendizaje por Refuerzo es extensa, y su aplicación en las finanzas ha sido analizada bastante en los últimos años. Existen varios estudios de revisión de literatura y entre ellos se encuentran [10], [11], [24], [25], [26] y [27]. En estos se explora las diversas aplicaciones del Aprendizaje por Refuerzo en el ámbito económico y financiero, que no son muy diferentes de las analizadas en un contexto de Aprendizaje de Máquina clásico (Aprendizaje Supervisado y No Supervisado) o de Aprendizaje Profundo, como lo son la optimización de portafolio, análisis de coberturas, valoración de activos y negociaciones algorítmicas. La diferencia en este caso es que los modelos de Aprendizaje por Refuerzo se usan para la toma de decisiones secuenciales, y en donde en su mayoría, son procesos dinámicos de Márkov, pero eso no quiere decir que no se estén desarrollando modelos híbridos como el llamado Aprendizaje Profundo Q, en donde el algoritmo por refuerzo tiene un componente importante de una red neuronal que modela relaciones no lineales, reemplaza el almacenamiento habitual en la teoría clásica de estados para la evaluación del valor **Q** y que sirven para ser evaluadas por una función de control de valor [11][26].

Para este trabajo se implementará el método propuesto por Hilpisch[4] [41] con algunas modificaciones para adaptarlo a los objetivos propuestos. El método corresponde a un modelo de Aprendizaje Profundo que predice el valor **Q** con redes neuronales. La novedad en la utilización de este tipo de metodologías es que , si bien, desde la teoría clásica de Aprendizaje por Refuerzo se sabe que el valor de **Q** puede ser encontrado mediante programación dinámica, se ha encontrado que los resultados obtenidos de la manera tradicional tienden a ser inestables y divergentes, lo que hace muy difícil de estimar dicho coeficiente [40], por este motivo diversos autores han propuesto emplear redes neuronales para calcular éste parámetro

[4] https://github.com/yhilpisch/aiif/tree/main/code

valiéndose de las capacidades del teorema de aproximación universal de las redes neuronales [41].

Conforme a lo anterior, para este estudio se plantea desarrollar el siguiente procedimiento:

1. Seleccionar una cantidad n de trayectorias sintéticas de los retornos de la acción seleccionada en el entendimiento de los datos y generadas por los modelos GAN según la capacidad computacional.
2. Seleccionar una serie de las n trayectorias.
3. Calcular los indicadores asociados a la trayectoria derivados del entendimiento de los datos.
4. Segmentar los datos en dos conjuntos, entrenamiento y validación según análisis realizado para el entendimiento de los datos.
5. Definir una arquitectura de una red profunda para estimar el valor Q del algoritmo.
6. Entrenar el agente con los datos de entrenamiento.
7. Evaluar con un conjunto de validación los resultados del modelo.
8. Repetir el paso 1 hasta que se recorran las n trayectorias sintéticas.
9. Evaluar el modelo de Aprendizaje por Refuerzo con datos reales de la acción seleccionada (conjunto de prueba) en el entendimiento de los datos y que no hayan sido utilizadas en para el entrenamiento del modelo GAN.

2. ENTENDIMIENTO DE LOS DATOS

Determinar qué tipo de datos son los adecuados para este trabajo, implica responder primero tres preguntas:

- ¿Cuál acción escoger para recolectar su data?
- ¿Cuál frecuencia de datos se debe recolectar?
- ¿Qué señales se utilizarán para complementar los precios para entrenar el agente?

La primera implica escoger una acción que tenga un comportamiento aceptable desde el punto de vista de la rentabilidad, y que además tenga la suficiente liquidez para poder ser analizada. Cuando se habla de liquidez se quiere decir, que ésta se

tranza diariamente durante un periodo de tiempo predefinido en el momento de consecución de los registros históricos de la misma.

La segunda pregunta, también relacionada con la liquidez, busca entender que tipo de frecuencia de datos seleccionar. Una frecuencia diaria es más propensa a no contener información (es eficiente) y una periodicidad de alta frecuencia (*tick*), es difícil de conseguir y cuando se puede obtener, requiere de gran capacidad de almacenamiento. Esto deja entonces, la posibilidad de frecuencias que van desde un rango de un minuto, cinco minutos, 15 minutos, 30 minutos y la hora.

La tercera pregunta está relacionada con la información adicional que complementa la serie de precios en el algoritmo de Aprendizaje por Refuerzo, al cual es posible agregar secuencias de datos que sean de alguna manera relevantes en el estado para entrenar el agente. Para este estudio se seguirá el procedimiento propuesto por Hilpisch [42] para evaluar indicadores que acompañaran los precios en el entrenamiento del modelo.

Respecto a la primera pregunta se resolvió primero tomando de publicaciones especializadas en internet, las acciones de mejor y peor desempeño de las listadas en el índice del Standard & Poor's 500 y Nasdaq de los años 2019 y 2020[5] de varios sitios web, que se consolidaron en una única lista de 98 títulos. Esta selección sirvió de base para descargar datos de diferentes frecuencias por medio de la api de Alpaca[6], que a la vez transforma los datos crudos publicados en el mercado IEX, los registros de transacciones en éste y que son publicados de forma gratuita. En la Tabla 1 se resumen los símbolos de las 98 acciones consideradas en la etapa preliminar y que sirvieron para el análisis de selección de la serie de tiempo en el trabajo.

---

[5] https://www.marketwatch.com/story/here-are-the-best-and-worst-dow-and-sp-500-stocks-of-2019-2019-12-30, https://www.marketwatch.com/story/these-are-the-20-worst-performing-sp-500-stocks-of-2020-analysts-see-double-digit-rebounds-for-6-of-them-in-2021-2020-12-29, y https://www.marketwatch.com/story/these-are-the-best-performing-nasdaq-and-sp-500-stocks-of-2020-2020-12-29.

[6] https://alpaca.markets/

AAL	CSCO	JPM	NOW	TGT
AAPL	CTLT	KHC	NVDA	TMUS
AAPL	CVX	KLAC	OKE	TRV
ABMD	DIS	KO	OKTA	TRV
ADS	DOCU	LB	OXY	TSLA
ALB	DXC	LRCX	PDD	UAL
ALGN	EOG	M	PFE	UNH
AMAT	ETSY	MCD	PG	V
AMD	FANG	MELI	PG	VLO
AMGN	FCX	MMM	PTON	VNO
AMZN	FTI	MOS	PWR	VZ
APA	GPS	MRK	PYPL	WBA
AXP	GS	MRNA	QRVO	WFC
BA	HD	MRO	RCL	WMT
CAT	HFC	MRVL	ROL	WST
CCL	HON	MSFT	SLB	XOM
CDNS	IBM	NCLH	SNPS	XRX
CMG	IDXX	NDX	SPG	ZM
CPRT	INTC	NKE	SPX	
CRM	JNJ	NOV	TEAM	

Tabla 1. Símbolos de las acciones consideradas en el análisis de selección de la serie de tiempo para el trabajo.

La api de Alpaca permite descargar precios de acciones en frecuencias de un minuto, cinco minutos, 15 minutos y diarias, y tiene un límite de descarga de 1,000 datos por cada consulta; las descargas tienen también una fecha límite histórica al 2 de enero de 2015 para ninguna acción consultada, por lo que ningún registro anterior a este día puede ser descargado. Para poder bajar los datos con la api, se desarrolló un código en Python que permitiera descargar los registros de las acciones seleccionadas en las frecuencias antes mencionadas (exceptuando las diarias por lo explicado anteriormente).

Para seleccionar cual es el título se analizó la frecuencia en los registros de un minuto de cada título descargado. La selección de este rango obedece a que si una acción tiene pocos registros en el día de dicha frecuencia la acción debe ser descartada. De igual forma, mayor frecuencia en los datos, indican que existen menos baches entre estas, y que barras de mayor rango de tiempo estarán más

completas sin necesidad de volver a verificar la información. Esto se explica, por ejemplo, analizando una barra[7] de cinco minutos. Esta está compuesta de muchos registros *tick* que puede ser agregada en intervalos de tiempo regulares. En el caso de una barra de cinco minutos la agregación sería de 6 barras de un minuto. El mismo razonamiento aplica para cualquier rango de tiempo superior al minuto.

En la Tabla 2 se puede apreciar las estadísticas de los primeros diez títulos ordenados según la cantidad de registros descargados con la api de Alpaca. Lo que muestra esta tabla es la calidad desde un punto de vista frecuentista de registros por día. El conteo muestra cuántos días tenían registros en ese periodo de descarga de datos. La media muestra el promedio de barras de minuto se contabilizaron cada día. La desviación muestra la variabilidad en las barras por día. El rango entre mínimo y máximo, y sus respectivos cuartiles muestran la distribución de los registros por minutos por cada día en la muestra.

Símbolo	Conteo	Media	Desv.	Mín.	25%	50%	75%	Máx.
MSFT	1260.0	386.00	22.15	0.0	388.00	390.0	390.00	391.0
AAPL	**1260.0**	**385.81**	**20.09**	**54.0**	**388.00**	**390.0**	**390.00**	**391.0**
INTC	1260.0	385.06	19.11	75.0	386.00	389.0	390.00	391.0
CSCO	1260.0	382.89	20.31	54.0	383.00	388.0	390.00	391.0
PFE	1260.0	382.18	21.08	17.0	382.00	387.0	389.00	391.0
WFC	1260.0	380.51	23.07	17.0	380.00	388.0	390.00	391.0
JPM	1260.0	378.41	22.79	46.0	376.00	385.0	389.00	391.0
FCX	1260.0	377.15	23.98	15.0	376.00	384.0	388.00	390.0
VZ	1260.0	376.67	23.90	15.0	373.00	385.0	389.00	391.0
MRO	1260.0	372.14	29.09	12.0	370.00	382.0	387.25	391.0

Tabla 2. Estadísticos del conteo de registros de las primeras acciones 10 acciones más completas descargados del sitio de internet Alpaca.

Después de analizar la regularidad en las frecuencias de registros de un minuto por día de los activos, se concluyó que la acción de Apple era la más adecuada, pues ésta tiene las mejores estadísticas junto con Microsoft, de registros en su media y cuantiles, y se destaca además de marcar precio todos los días.

[7] Una barra en finanzas se entiende como un registro que contiene generalmente una tupla de 6 datos: Estampa de Tiempo, Apertura, Máximo, Mínimo, Cierre y Volumen. El contenido de esta barra estará definida por la forma en como se agrupen los datos, pueden ser por tiempos (barra de tiempo), por volumen (barra de volumen) o por cantidad de moneda transada (barra de dólar) [7].

Dado que la fuente de información proviene únicamente del mercado IEX y que esta bolsa no consolida las transacciones con registros de otras bolsas o *dark pools*, los precios históricos en general presentan muchos baches entre los períodos intradiarios. Después de analizar las frecuencias para un minuto, cinco minutos y, 15 minutos, se llegó a la conclusión que ésta última era la periodicidad que presentaba menos baches, y que, por lo tanto, los datos faltantes podías ser reemplazados con técnicas estándares para el relleno de datos y que son utilizadas en finanzas para complementar las series. La selección de estas medidas como evaluadores de la calidad de la información, radica en qué, tener mayor información permite modelar mejor los mercados analizados. Como puede verse en la Tabla 2, son las acciones de las grandes compañías las que presentan una mejor dinámica (mayor número de transacciones), lo cual redunda en mayores posibilidades de mejores pronósticos.

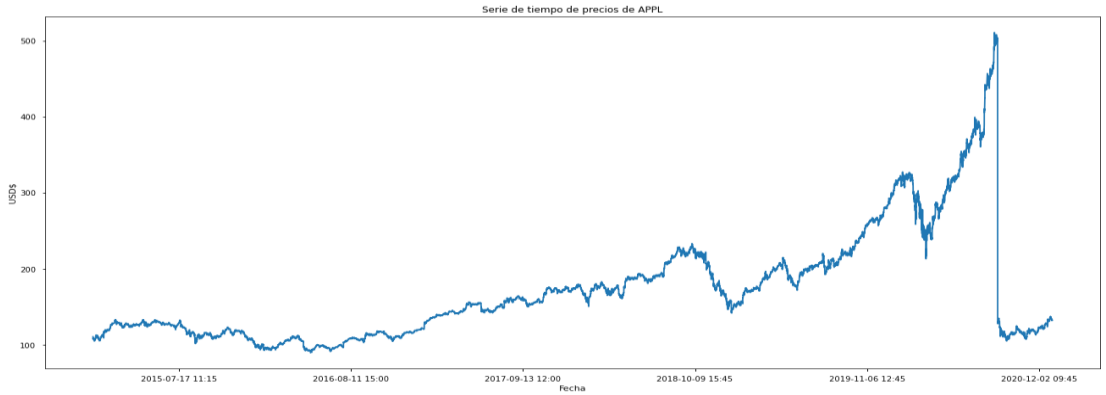
Lo anterior también se soporta en el hecho de que un objetivo de este estudio es el de implementar un generador de series de tiempo sintéticas. En ese sentido, tener espacios entre los registros de manera frecuente, hace más difícil modelar los hechos estilizados. En consecuencia, la presencia de baches en la serie histórica podría disminuir la calidad de las series generadas por el modelo de redes neuronales.

2.1 PREPROCESO DE LOS DATOS

Seleccionada la acción de Apple (AAPL) y la frecuencia de los registros de 15 minutos por día, se procede a consolidar los archivos descargados en uno sólo. Los registros comprenden información desde 2 de enero del 2015 a 2831 de diciembre del 2020. En el Gráfico 2 se observa en comportamiento de los precios de cierre de la acción durante este período. En éste se aprecia un quiebre abrupto el día 30 de agosto del 2020. La explicación a este fenómeno es que el precio se dividió en una razón de 4:1 (cuatro acciones por una acción)[8], lo que es una práctica común en mercados desarrollados, en donde se reparte un “dividendo” vía aumento de acciones en el mercado [43], de tal forma que quien posee una acción, recibe a

[8] https://www.nasdaq.com/articles/stock-alert%3A-apple-up-after-4-1-stock-split-2020-08-31

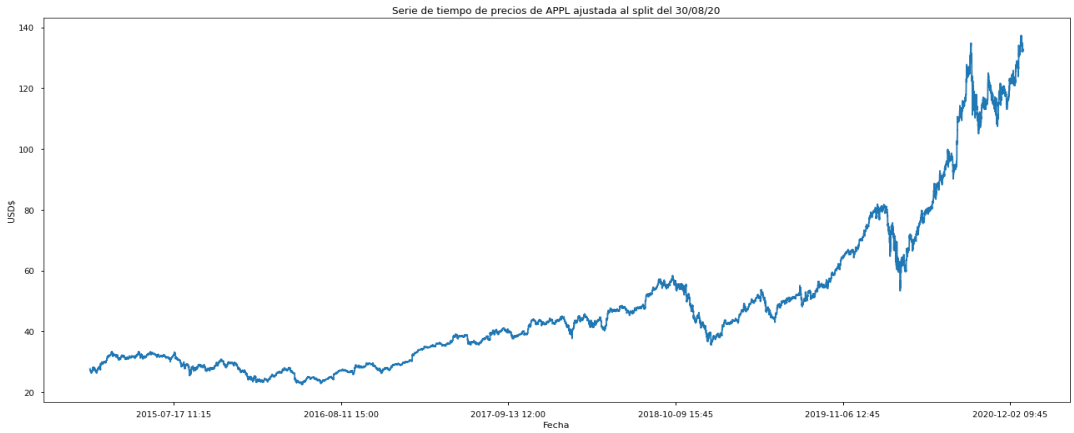
cambio “x” acciones, según la razón de división definida en el momento del *split* o división de la acción.



Fuente: elaboración propia.

Gráfico 2. Serie de tiempo del precio de 15 minutos diarios de la acción de Apple entre el 02/01/15 a 31/12/20 antes de corrección.

Para corregir este fenómeno es necesario dividir el precio de la acción por 4 desde el 30 de agosto de 2020 hasta el inicio de la serie, esto es para mantener coherencia con los precios a partir de dicha fecha. En el Gráfico 3 se pude observar la serie de tiempo corregida.



Fuente: elaboración propia.

Gráfico 3. Serie de tiempo del precio de 15 minutos diarios de la acción de Apple entre el 02/01/15 a 31/12/20 después de corrección.

La serie transformada al corregir el *split* en los precios del 30 de agosto de 2020 al inicio de la muestra, es la que se utilizará como base para todos los análisis en este trabajo.

2.2 ANÁLISIS EXPLORATORIO DE DATOS

La acción seleccionada para este trabajo es el título de Apple con el símbolo AAPL y los registros descargados por medio de la api de Alpaca y comprenden un período entre el 2 de enero de 2015 a 31 de diciembre de 2020, y están dados en barras de una frecuencia de 15 minutos por día. Este mercado a las 9:30 am y cierra a las 4:00 pm (Hora estándar del Este, Nueva York). En números significa disponer 40,743 registros en toda la muestra. En la Tabla 3 se resumen el contenido descargado.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40743 entries, 0 to 40742
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Date    40743 non-null  object
 1   open    40743 non-null  float64
 2   high    40743 non-null  float64
 3   low     40743 non-null  float64
 4   close   40743 non-null  float64
 5   volume  40743 non-null  int64
dtypes: float64(4), int64(1), object(1)
memory usage: 1.9+ MB
```

Tabla 3. Descripción de los datos de la acción de Apple entre 02/01/2015 a 31/12/2020 con frecuencia de 15 minutos.

En finanzas se acostumbra a modelar los cambios relativos en los precios de cierre de la barra, llamados retornos o rendimientos, en lugar del precio en sí. Matemáticamente se entiende el retorno $\mathbf{r_t}$ de una acción como el logaritmo natural del cambio de precios entre el inicio y fin de un período de tiempo **t**, en la Ecuación 7 se muestra la fórmula para este cálculo.

$$r_t = ln\left(\frac{p_t}{p_{t-1}}\right)$$

Ecuación 7. Rendimiento de un activo en el tiempo t.

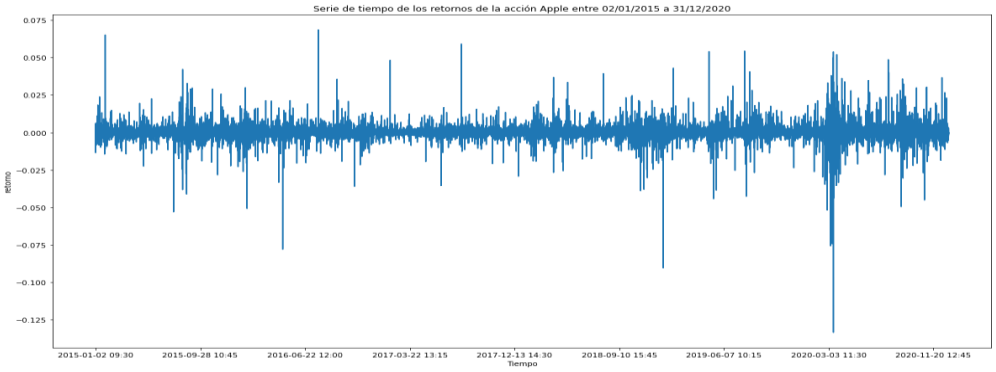
Donde:

r_t: Es el retorno en el período **t**

p_t: Es el precio en el precio **t**

p_{t-1}: Es e precio en el precio **t-1**

Para esta serie se calcularon los retornos o rendimientos para cada período analizado. En el Gráfico 4 se muestra la evolución en el tiempo de los precios y retornos de la acción durante el período de estudio. Como es de esperarse, la acción presenta alta volatilidad, agrupamientos por periodos y saltos a través del tiempo.



Fuente: elaboración propia.

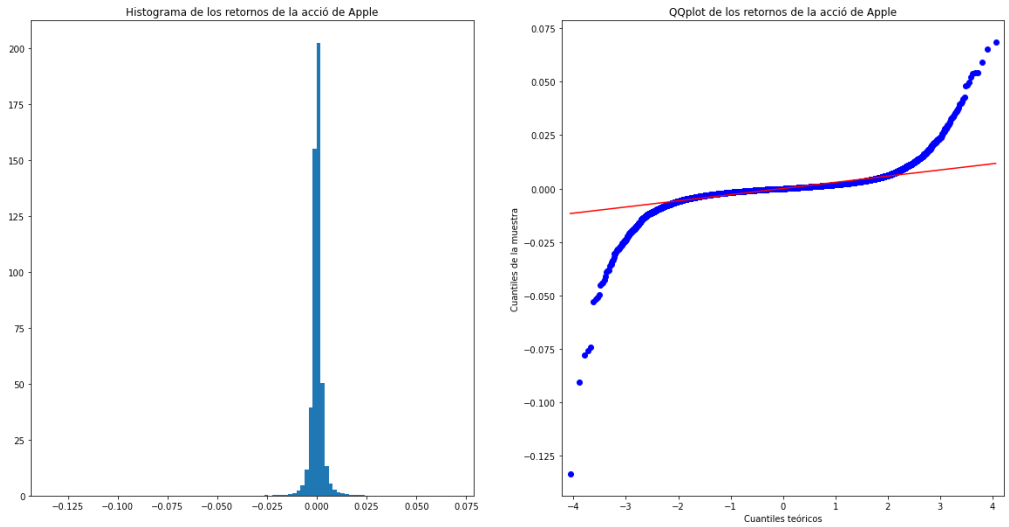
Gráfico 4. Serie de tiempo de los retornos de 15 minutos diarios de la acción de Apple entre el 02/01/15 a 31/12/20 después de corrección.

Los estadísticos descriptivos de la serie histórica se presentan en la Tabla 4 en el periodo de tiempo del estudio. De esta información puede observase que los retornos cumplen las características de los hechos estilizados, como lo son una alta concentración alrededor de la media, así como alta dispersión entre sus valores máximos y mínimos.

Estadístico	Retornos
Conteo	40,743.00
Media	0.000045
Desv.	0.003506
Mín.	-0.124905
25%	-0.001056
50%	0.000000
75%	0.001105
Max.	0.070891
Curtosis	105.490108
Sesgo	-1.131351

Tabla 4. Estadísticos descriptivos de los retornos de la acción de Apple entre 02/01/2015 a 31/12/2020 con frecuencia de 15 minutos.

Analizando la distribución empírica y el QQplot de la serie de los retornos, tal y como puede apreciarse en el Gráfico 5, es posible evidenciar lo que muestra la Tabla 4, y observar la presencia de colas pesadas en ésta. La distribución presenta valores extremos, una alta concentración de retornos cercanos a cero haciendo que la misma tenga un exceso de curtosis alto, en donde su valor es de USD$ 105.50, lo que explica el pico que se aprecia en el mismo gráfico. Los valores extremos en ambas colas (-12%, 7%) son amplios para cambios en el precio de una serie de 15 minutos en donde las variaciones tienden a ser pequeñas, esto apoya la necesidad de generar precios sintéticos que tiendan a emular estos comportamientos de forma ficticia, y no con comportamientos normales que es lo que se acostumbra en las simulaciones que generan secuencias brownianas geométricas.



Fuente: elaboración propia.

Gráfico 5. Histograma y diagrama de cuantiles de los retornos de la serie de tiempo de los retornos de la acción de Apple entre 02/01/2015 a 31/12/2020 con frecuencia de 15 minutos.

2.3 SELECCIÓN DE REGLAS DE NEGOCIACIÓN O TRADING

2.3.1 ESTRATEGIA DE PROMEDIO MÓVIL SIMPLE

Existe extensa literatura alrededor de cuáles deben ser las señales que indiquen cuando comprar o vender un activo en un mercado financiero, que va desde la astrología, pasando por análisis técnico o de patrones hasta los métodos más robustos como los estadísticos. Para analizar qué tipo de secuencia diferente a la de los retornos y después de revisar la literatura, se decidió utilizar el método de *vectorized backtesting* o pruebas de respaldo vectorizado (por su traducción al español del inglés) propuesto por Yves Hilpisch [41][42] para definir qué tipo de indicador puede ser agregado al entorno del agente.

Básicamente, las pruebas de respaldo vectorizado, se refiere a códigos en donde se utiliza el algebra matricial para replicar las utilidades de una estrategia de trading en un vector de posición que contiene valores de 1 (comprar) y -1 (vender) [42]. Esta forma de modelar las ganancias de una estrategia es útil porque permite

evaluar de manera simple si una técnica de modelado rinde frutos o no sirviéndose del algebra lineal.

Sin embargo, el análisis vectorizado tiene una falla que para algunos puede ser fundamental y es que no considera costos transaccionales, y por lo tanto los resultados pueden ser irreales o ingenuos [41]. En una aplicación real de estas metodologías llevada a un entorno productivo, estos costos deben ser considerados teniendo en cuenta las condiciones o fricciones propias del mercado estudiado, pero se considera que es una buena aproximación para evaluar en una situación de laboratorio para el entrenamiento de un algoritmo de Aprendizaje por Refuerzo que es el objetivo de este estudio. Como el abanico de alternativas de estrategias es amplio, se optó por trabajar con la técnica de promedios móviles simples, que es empleada por los practicantes en el mercado [40].

La estrategia de los promedios móviles es una técnica que utiliza un promedio retrasado, el cual es calculado en una ventana de tiempo **T.** El comportamiento con respecto al movimiento de los precios reales, se considera emiten señales de compra o venta, y que si son seguidos de forma sistemática, deben producir utilidades para quien las aplique. Un promedio móvil de un precio de acciones se puede definir como la media de los precios de una acción en un rango de tiempo estricto o cerrado **T**, el cual se recalcula al desplazarse un registro hacia adelante, generando una serie paralela a los precios. La función matemática de este indicador se expresa en la Ecuación 8.

$$\boldsymbol{PMS(T)_t = \frac{1}{T}\sum_{t=1}^{T} p_t}$$

Ecuación 8. Promedio Móvil Simple.

Donde:

$\boldsymbol{PMS(T)_t}$: Promedio móvil simple en la ventana de tiempo **T**

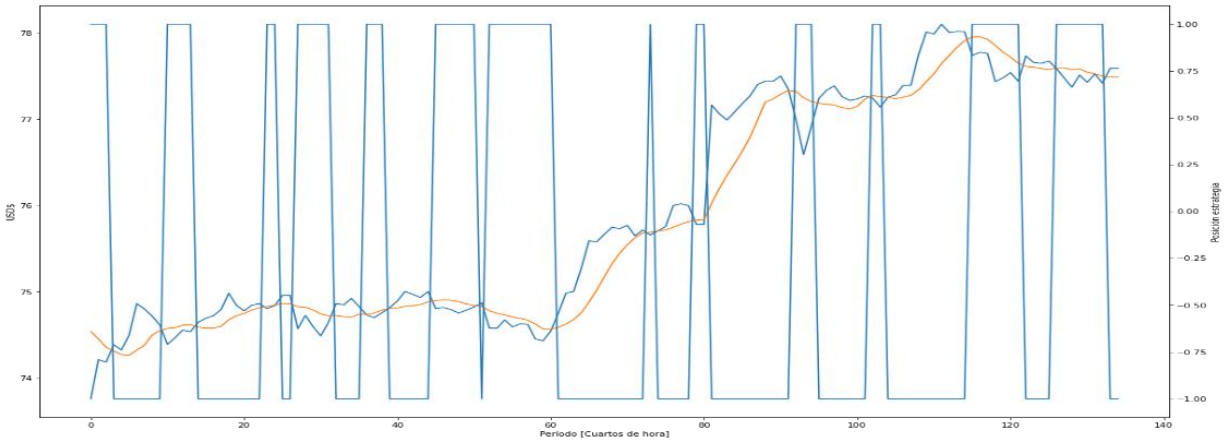
$\boldsymbol{T}$: Es la ventana de tiempo a estimar el promedio

$\boldsymbol{p_t}$: Precio en el tiempo **t**

Como premisa inicial se tiene un promedio móvil simple **PMS** para un periodo **T** [38], que emitirá una señal de venta de compra o venta de la acción asignando un signo (1 o -1) según se cumpla alguna de las siguientes condiciones [44]:

- Compra cuando $\mathbf{PMS(T)_t} > \mathbf{p_t}$
- Venta cuando $\mathbf{PMS(T)_t} < \mathbf{p_t}$

En este caso $\mathbf{p_t}$ es el precio del activo analizado en el tiempo **t**. Para evaluar una estrategia de promedios móviles simples mediante el proceso de prueba respaldo vectorizado, se debe determinar la posición (signo) generada por este conjunto de reglas, calculando un vector (columna). En el Gráfico 6 se aprecia un ejemplo de la posición generada por una estrategia de promedio móvil simple en un segmento de la serie analizada en este trabajo. Los valores de la posición van desde -1 a 1, donde el primer valor se refiere cuando $\mathbf{PMS(T)_t} < \mathbf{p_t}$ (señal de compra) y el último corresponde a cuando $\mathbf{PMS(T)_t} > \mathbf{p_t}$ (señal de venta). El cambio de régimen se da cuando hay un cambio de tendencia.



Fuente: elaboración propia.

Gráfico 6. Series de Precio y Promedios Móviles Simple, y Posición generada por estrategia para un segmento de la acción de Apple.

Una vez se identifica la posición de los datos históricos, se hace un desplazamiento del vector posición de un período y se multiplica por los retornos de las series, el sentido de este desplazamiento es que el cruce de promedios se da en un tiempo **t**, el cual origina una señal de ejecución de una operación (venta o compra) y los

resultados de la aplicación de ésta se reflejan en el tiempo **t+1**. El beneficio de la estrategia se da cuando se calcula el interés compuesto acumulado de este nuevo vector.

En finanzas el interés compuesto es la acumulación de rendimientos a través de un periodo de tiempo n. Para este trabajo se utiliza el interés compuesto continuo pues es habitual en el cálculo de rentabilidades para periodicidades inferiores a un día, y en este caso los intervalos de la serie están definidos por diseño en períodos de 15 minutos cada uno. el interés continuo compuesto acumulado se expresa mediante la Ecuación 10.

$$\boldsymbol{R} = \boldsymbol{e}^{\sum_{t=1}^{T} r_t}$$

Ecuación 10. Rendimiento compuesto continúo acumulado.

Donde:

R: Rendimiento compuesto

t: Periodo total analizado

$\boldsymbol{r_t}$: Rendimiento del periodo analizado

El desempeño de la estrategia dependerá de contrastar los rendimientos acumulados durante ese período, acción que se llamará estrategia pasiva (se asume se compra el título al inicio de la muestra o período evaluado y se retiene hasta el final del período analizado), con los rendimientos acumulados de la estrategia del Promedio Móvil Simple (PSM). Para el trabajo se realizaron varios experimentos, en donde se compararon estrategias para los períodos 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, y 36 cuartos de hora. En la Tabla 5 se resumen los resultados de los retornos acumulados obtenidos por cada estrategia en el último registro de la serie de tiempo analizada ordenadas según el retorno de la estrategia. Los valores presentados en la esta tabla corresponden primero a los promedios móviles calculados. La columna de Retornos Pasivo muestra los rendimientos de una estrategia de comprar y retener hasta el final del período de análisis[9], mientras que la columna de Retorno de Estrategia refleja el rendimiento obtenido por las señales derivadas del sistema de Promedios Móviles Simples. Como puede verse, seguir

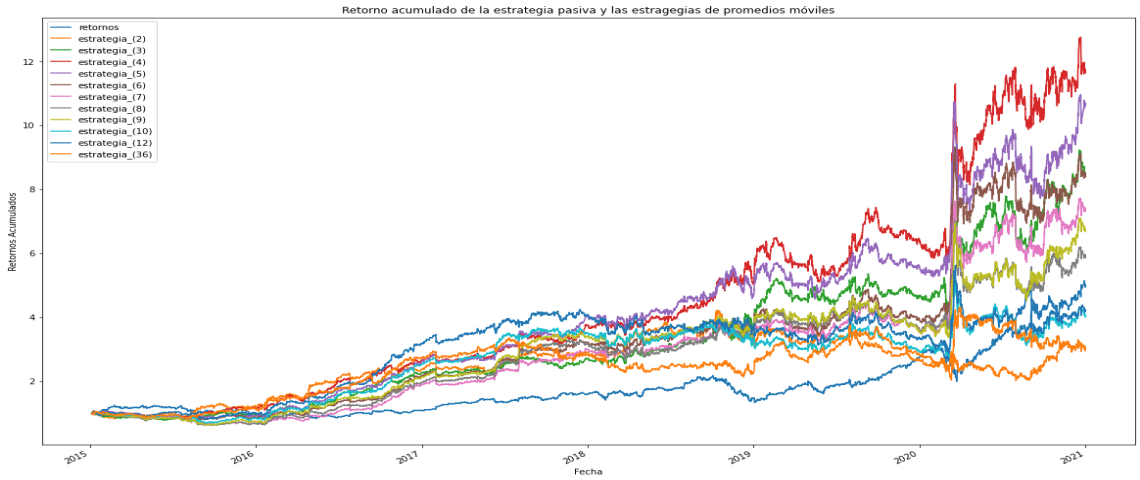
[9] Las Tablas de datos de cada análisis se truncaron con la estampa de tiempo mínima del PMS mayor (64) para poder hacer equiparables los datos y poderlos comparar.

esta estrategia presenta buenos resultados, con excepción de las estrategias de promedios móviles de 2 y 36 cuartos de hora respectivamente.

Estrategia	Retorno Pasivo	Retorno Estrategia
4	**5.14**	**12.76**
5	5.14	10.97
6	5.14	9.33
3	5.14	9.23
7	5.14	7.73
9	5.14	7.11
8	5.14	6.98
12	5.14	5.63
10	5.14	5.19
36	5.14	4.32
2	5.14	4.23

Tabla 5. Retornos compuestos de las estrategias de Promedios Móviles Simples analizadas.

Para entender mejor el desempeño de los diferentes experimentos con los promedios móviles se graficó la evolución de los rendimientos acumulados a través del tiempo, el resultado se presenta en el Gráfico 7. De éste puede inferirse que la mejor estrategia es la de utilizar el promedio móvil 4 períodos o cuartos de hora, por esta razón, éste será el indicador que se utilizará para entrenar el agente como complemento a los precios. Éste tendrá la tarea de ser referencia de un patrón de negociación para el agente, con el cual éste podrá aprender de manera óptima las señales de compra y venta y así obtener la mejor recompensa.



Fuente: elaboración propia.

Gráfico 7. Evolución del retorno acumulado en el tiempo de la estrategia de momento para diferentes ventanas para la acción de Apple.

3. MODELAMIENTO DE LOS DATOS

3.1 GENERACIÓN DE DATOS SINTÉTICOS MEDIANTE UN MODELO GAN

Para implementar un modelo de series de tiempo GAN, es necesario considerar la longitud de las series de tiempo sintética a generar por la arquitectura, pues de esta depende los resultados del modelo de Aprendizaje por Refuerzo, lo que convierte la decisión de la selección del tamaño de la serie en lo que dice el famoso dicho popular: “qué fue primero el huevo o la gallina”. Para poder tomar esta decisión hubo que hacer diferentes experimentos completos y de acuerdo a los resultados decidir cuál era el más conveniente para el estudio, pues no hay una regla teórica o empírica que ayude a decidir un tamaño óptimo o adecuado.

Es importante resaltar que la serie histórica está compuesta por lecturas del mercado de cuartos de hora, por lo tanto, si lo que se requiere modelar es una hora, se deben generar cinco registros, dos horas equivalen a nueve registros, un día a 27 registros, una semana a 135 registros, un mes a 540 registros y dos meses a 1,080 registros. Por lo tanto, la selección de longitud en modelo GAN se debe hacer teniendo en cuenta una magnitud que represente un período regular de tiempo.

Después de diversos ensayos se selecciona un período de 216 registros, lo que equivale a ocho días. La selección de éste intervalo se hizo tomando como guía lo realizado por Hilpisch [41] en su modelo para el mercado Forex. En donde selecciona una serie cuyo valor unitario es de 30 segundos y para entrenar el modelo de Aprendizaje por Refuerzo, escoge un conjunto de datos con información de cinco horas. La validación es hecha con datos de una hora y, por último, la prueba él autor la hace con registros dos horas. De este procedimiento se puede deducir una regla empírica de {5x para entrenamiento, x para validación, 2x para prueba} en donde x es el período de tiempo que emplea el agente para validar la estrategia. Como normalmente las evaluaciones de un participante del mercado se hace por resultados diarios, es coherente pensar que, para este trabajo, x es igual a 27 cuartos de hora (lo que equivale a un día). Lo que lleva al aplicar este a la regla empírica deducida del texto a establecer un horizonte de tiempo y dimensión de entrada para el modelo GAN del estudio en: 5 x 27 + 27 + 2 x 27 = 216.

Definido el horizonte de tiempo de la serie sintética, el siguiente paso es determinar qué datos de la muestra van a alimentar al modelo GAN, y para este trabajo se decidió trabajar con toda la muestra. La razón para esta decisión fue la de permitir que el generador contara con la mayor cantidad de información disponible posible y que los resultados fueran lo más realistas y, además estuvieran acorde con los datos de la distribución de la muestra de los retornos de la acción.

Un modelo GAN tiene tres componentes: red generadora, red discriminadora y red enlace; siendo ésta última, la red que conecta las dos primeras redes y produce una información de salida (clasifica). En la Tabla 6 se aprecia los elementos principales con los que se definió la arquitectura del discriminador.

Capa	**Filtro**	***Kernel***	**Paso**	**d**	***Padding***	**Activación**	***Dropout***
Densa	64	1	1	1	*same*	*LeakyRelu* (α = 0.2)	0.4
Convolucional 1d	128	1	1	1	*same*	*LeakyRelu* (α = 0.2)	0.4
Convolucional 1d	128	2	1	2	*same*	*LeakyRelu* (α = 0.2)	0.4
Convolucional 1d	128	2	1	4	*same*	*LeakyRelu* (α = 0.2)	0.4
Plana	*None*	*None*	*None*	*None*	*None*	*None*	*None*
Densa	1	2	1		*same*	*None*	*None*

Tabla 6. Parámetros de la arquitectura del discriminador.

Tensorflow arroja unos resúmenes útiles para el momento de la reproducibilidad de los resultados, como puede verse en el Gráfico 8. En éste se puede apreciar las diferentes capas utilizadas en la implementación de la arquitectura del discriminador, además de las formas de entrada a cada capa y los parámetros entrenados en cada una de ellas.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 217, 64)           320
_________________________________________________________________
leaky_re_lu (LeakyReLU)      (None, 217, 64)           0
_________________________________________________________________
dropout (Dropout)            (None, 217, 64)           0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 217, 128)          32896
_________________________________________________________________
leaky_re_lu_1 (LeakyReLU)    (None, 217, 128)          0
_________________________________________________________________
dropout_1 (Dropout)          (None, 217, 128)          0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 217, 128)          65664
_________________________________________________________________
leaky_re_lu_2 (LeakyReLU)    (None, 217, 128)          0
_________________________________________________________________
dropout_2 (Dropout)          (None, 217, 128)          0
_________________________________________________________________
conv1d_3 (Conv1D)            (None, 217, 128)          32896
_________________________________________________________________
leaky_re_lu_3 (LeakyReLU)    (None, 217, 128)          0
_________________________________________________________________
flatten (Flatten)            (None, 27776)             0
_________________________________________________________________
dense (Dense)                (None, 1)                 27777
=================================================================
Total params: 159,553
Trainable params: 159,553
Non-trainable params: 0
_________________________________________________________________
```

Gráfico 8. Resumen de la implementación de la red para el discriminador.

En la Tabla 7 y en el Gráfico 9 se muestran los elementos de la red generadora, necesarios para la reproducibilidad del estudio. La arquitectura es similar a la diseñada para el discriminador. Nótese, que la entrada de datos es diferente a la del discriminador. En ambas redes se utilizó el parámetro **d** o dilatación, que permite que el modelo retenga información (memoria) de capas anteriores. Cuando se activa este campo en la capa los pasos (*stride*), estos no pueden ser mayores a uno. Está es la razón por la cual no hay pasos diferentes entre las capas. Los coeficientes de dilatación de varias capas del generador son diferentes, y su especificación se da por la experimentación, buscando que las redes convolucionales capturen las dependencias temporales de las series de entrenamiento.

Capa	Filtro	*Kernel*	Paso	d	*padding*	Activación	*Dropout*
Densa	128	1	1	1	*same*	*LeakyRelu* (α = 0.2)	*None*
Convolucional 1d transpuesta	128	1	1	1	*same*	*LeakyRelu* (α = 0.2)	*None*
Convolucional 1d transpuesta	128	2	1	8	*same*	*LeakyRelu* (α = 0.2)	*None*
Convolucional 1d transpuesta	128	2	1	12	*same*	*LeakyRelu* (α = 0.2)	*None*
Convolucional 1d transpuesta	128	2	1	36	*same*	*LeakyRelu* (α = 0.2)	*None*
Convolucional 1d	1	2	1	1	*same*	None	*None*

Tabla 7. Parámetros de la arquitectura del generador.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 27776)             305536
_________________________________________________________________
leaky_re_lu_4 (LeakyReLU)    (None, 27776)             0
_________________________________________________________________
reshape (Reshape)            (None, 217, 128)          0
_________________________________________________________________
conv1d_transpose (Conv1DTran (None, 217, 128)          16512
_________________________________________________________________
leaky_re_lu_5 (LeakyReLU)    (None, 217, 128)          0
_________________________________________________________________
conv1d_transpose_1 (Conv1DTr (None, 217, 128)          16512
_________________________________________________________________
leaky_re_lu_6 (LeakyReLU)    (None, 217, 128)          0
_________________________________________________________________
conv1d_transpose_2 (Conv1DTr (None, 217, 128)          32896
_________________________________________________________________
leaky_re_lu_7 (LeakyReLU)    (None, 217, 128)          0
_________________________________________________________________
conv1d_transpose_3 (Conv1DTr (None, 217, 128)          32896
_________________________________________________________________
leaky_re_lu_8 (LeakyReLU)    (None, 217, 128)          0
_________________________________________________________________
conv1d_4 (Conv1D)            (None, 217, 1)            257
=================================================================
Total params: 404,609
Trainable params: 404,609
Non-trainable params: 0
_________________________________________________________________
```

Gráfico 9. Resumen de la implementación de la red para el generador.

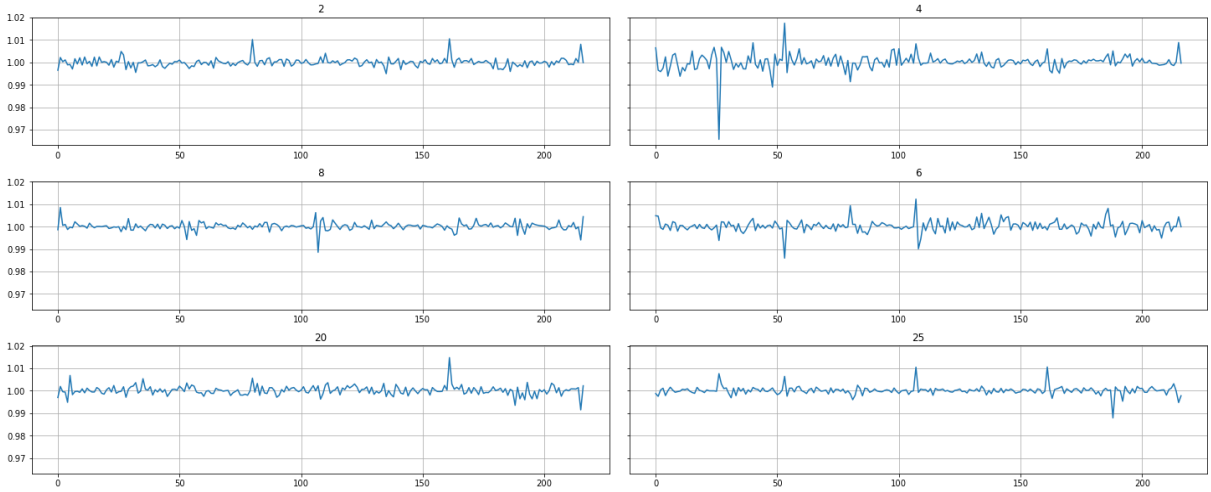
Como se están modelando dos redes por separado, es necesario diseñar otra red que permita hacer el empalme entre el discriminador y el generador, el Grafico 10 se muestra la especificación de esta red de enlace y que permite clasificar la red generada como real o falsa.

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
sequential_1 (Sequential)    (None, 217, 1)            404609
_________________________________________________________________
sequential (Sequential)      (None, 1)                 159553
=================================================================
Total params: 564,162
Trainable params: 404,609
Non-trainable params: 159,553
_________________________________________________________________
```

Gráfico 9. Resumen de la implementación de la red para el generador.

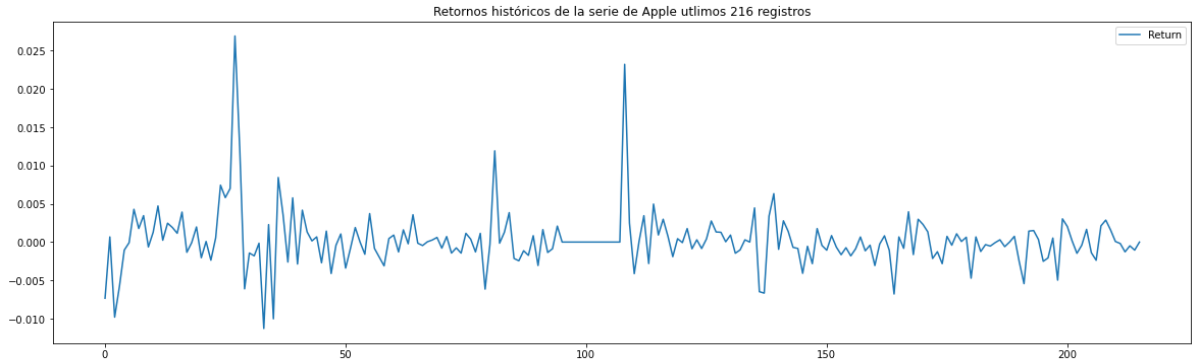
El modelo se optimiza con el método Adam y con una tasa de aprendizaje **lr** igual a 0.0001. Finalmente, se corre el modelo GAN para 10,000 épocas de 60 lotes cada una. En cada corrida se evalúa la distribución sintética comparada con la distribución real. De este ejercicio se guardaron 2,638 modelos que cumplían los dos criterios establecidos en la sección 1.6 que son: una precisión del 90% en la clasificación de las series sintéticas para los modelos generador y discriminador; y que el valor-p del estadístico Kolmogorov - Smirnov de las distribuciones fuera mayor al 5%, es decir, rechazando $\mathbf{H_0}$ y poder asumir que los datos sintéticos si pertenecen a la distribución empírica obtenida de los datos históricos.

En el Gráfico 10 se presentan seis ejemplos de retornos sintéticos generados por uno de los modelos GAN que cumplieron con los criterios de selección definidos en la sección 1.6. Que al ser comparados con el Gráfico 11 (en el que se dibujan retornos históricos de los últimos 216 registros de la muestra), es posible apreciar visualmente que las series sintéticas a través del tiempo, tienen un comportamiento similar a una muestra de igual tamaño obtenida de la serie real. En los datos sintéticos se presentan saltos bruscos o picos, así como agrupamientos (clústeres) de volatilidad.



Fuente: elaboración propia.

Gráfico 10. Muestra de retornos sintéticos de 6 series generadas por un modelo GAN.



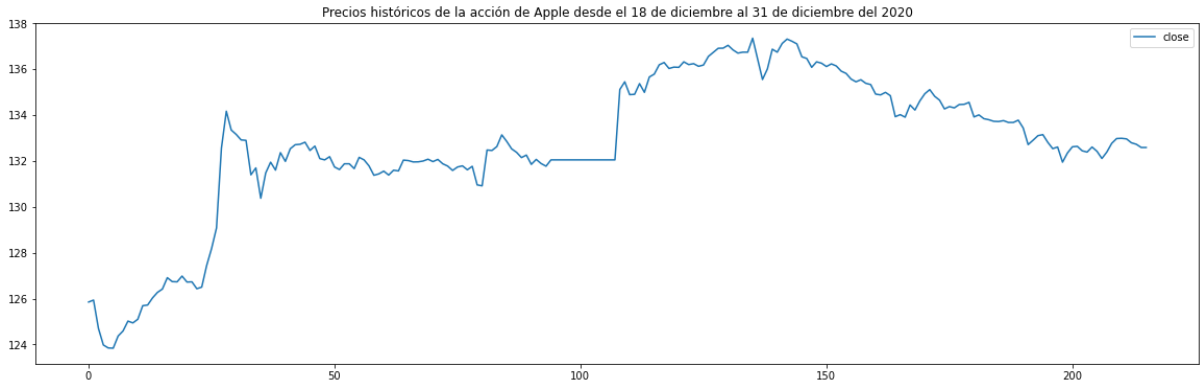
Fuente: elaboración propia.

Gráfico 11. Retornos históricos de los últimos 216 registros de la acción de Apple de la muestra analizada.

Otra forma interesante de ver los datos es comparar una serie de precios histórica de igual longitud con series de precios simuladas. La primera se muestra en el Gráfico 12 en donde se grafica una secuencia continua de los últimos 216 registros históricos para la acción de Apple. En el Gráfico 13 se dibujan 6 series sintéticas, que se construyeron con un precio inicial a USD$ 127.78, que corresponde al inicio de la serie utilizada para construir el Gráfico 12. Como puede observarse, las series sintéticas (Gráfico 13) son similares en su evolución a la serie real (Gráfico 12) para

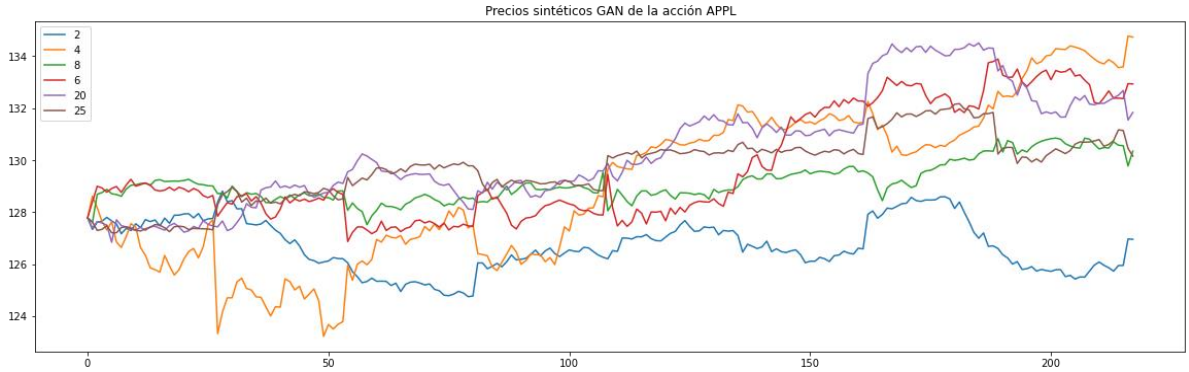
ese período de tiempo. En estas se presentan saltos bruscos y hay correcciones en el tiempo muy parecidas a las series históricas.

.



Fuente: elaboración propia.

Gráfico 12. Precios históricos de la acción de Apple desde el 17 de diciembre al 31 de diciembre del 2020.

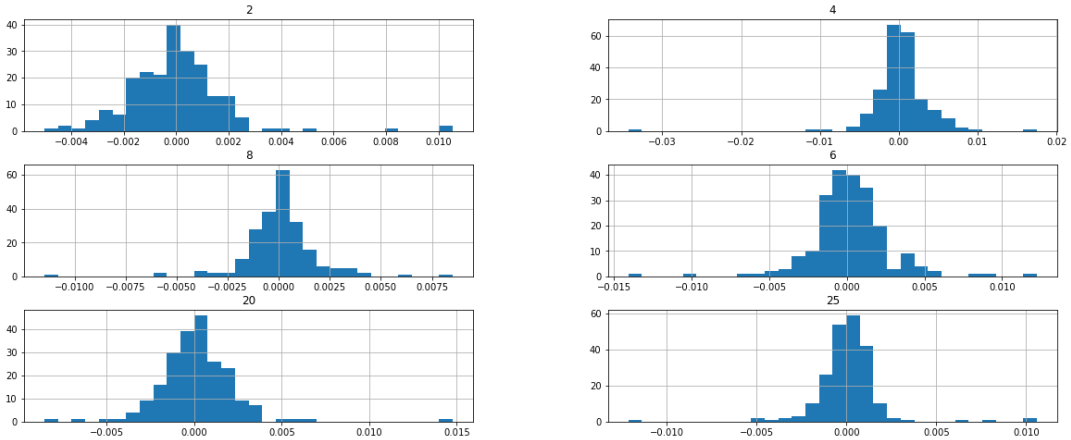


Fuente: elaboración propia.

Gráfico 13. Muestra de precios sintéticos generados con un modelo GAN de la para la acción de Apple

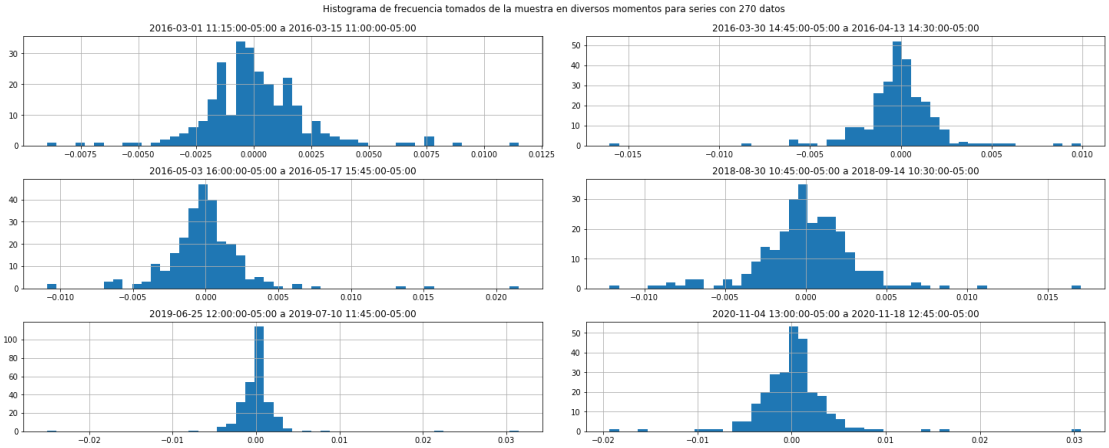
En los Gráficos 14 y 15 se hace un ejercicio comparativo de los histogramas de las series de retornos sintéticas y con tramos de la serie histórica de igual longitud (216 registros) y que fueron seleccionados aleatoriamente de la serie histórica. Al observar ambos gráficos, se puede observar que las muestras sintéticas son similares. Los datos simulados reproducen colas pesadas, no se distribuyen normalmente, la media es cercana a 0 como en las series históricas, y también hay

evidencia de leptocurtosis en todos los histogramas, características que son las que se buscan reproducir desde la definición del problema de generación de datos artificiales.



Fuente: elaboración propia.

Gráfico 14. Histograma de frecuencia de seis muestras generadas por el modelo GAN.



Fuente: elaboración propia.

Gráfico 15. Histograma de frecuencia de serie históricas escogidas aleatoriamente con 216 registros cada una de Apple.

Del Gráfico 13 es importante anotar algo al respecto de los resultados sintéticos, a pesar de la evolución de los retornos, y por lo tanto de los precios, estos son

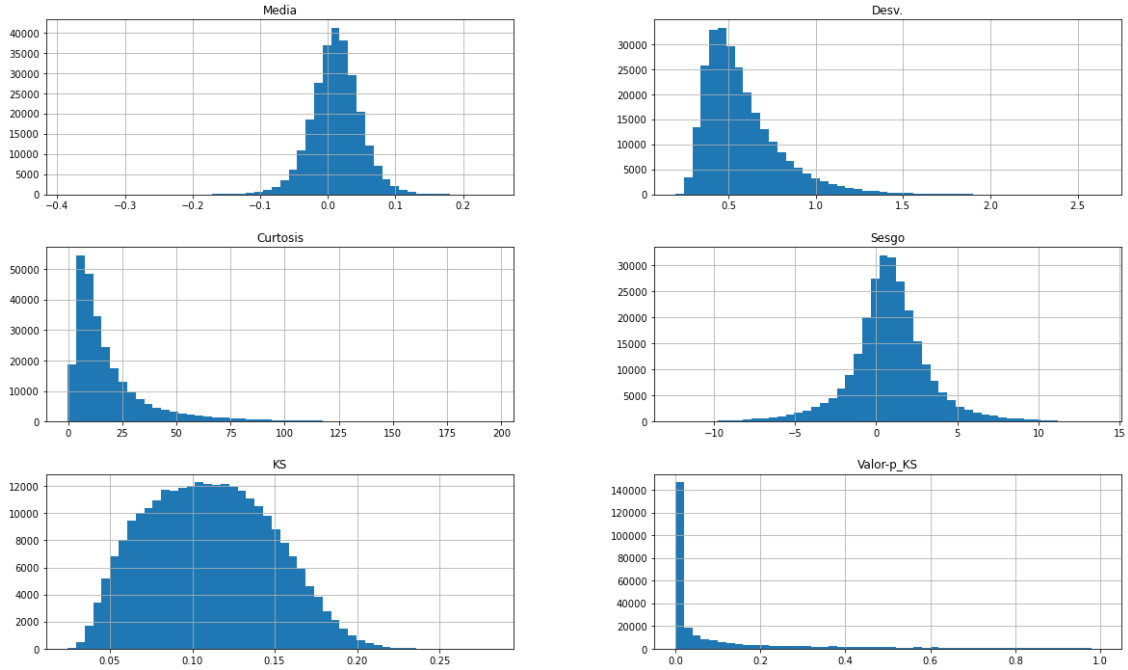
similares a los reales, se nota que el modelo GAN tiende a producir ciertas regularidades en su comportamiento que deben ser evitadas en el momento de entrenar un agente, pues sí no se descartan, se estaría entrenando para seguir patrones pseudoaleatorios, y eso es algo que se debe evitar, ya que lo que se busca con las muestras simuladas es que los datos sintéticos generen información similar a la real, y que no sea predecible, es decir, que lo que aprenda el agente sea evaluar una serie cualquiera que provenga de la distribución de los datos, y no sesgos ocasionados por la generación sintética de datos.

Para poder evitar estos sesgos generados por los modelos GAN y el riesgo de un mal entrenamiento del agente, se evaluaron los estadísticos de cada modelo. Esto se hizo generando en cada modelo GAN seleccionado 100 muestras sintéticas, a las cuales se les calculó su media, desviación estándar, exceso de curtosis y sesgos respectivos. Cada parámetro se agrupo a su vez, obteniendo un conjunto de estadísticos propios de cada parámetro calculado. En la Tabla 8 se presentan un resumen estadístico de cada parámetro. De ésta puede evidenciarse la dispersión en los parámetros estadísticos con respecto a la muestra. Un ejemplo se puede apreciar al observar la media de retornos mínimos y máximos (-38.5%, 24.4%) vs. los datos históricos (-12.5%, 7.1%), las diferencias entre ellos son considerables.

Estadístico	Media	Desv.	Moda	Exceso de Curtosis	Sesgo	KS	Valor-p_KS
Media	0.012	0.579	-2.911	19.343	0.796	0.110	0.111
Desv.	0.037	0.220	2.096	20.075	2.505	0.038	0.199
Min.	-0.385	0.195	-27.006	-0.176	-12.381	0.024	2.33E-15
25.00%	-0.010	0.425	-3.535	7.169	-0.393	0.081	0.001
50.00%	0.012	0.525	-2.247	12.668	0.779	0.109	0.011
75.00%	0.034	0.675	-1.556	23.401	2.026	0.138	0.119
Max.	0.244	2.629	0.984	195.950	13.802	0.282	1.000

Tabla 8. Estadísticos de los resultados obtenidos al simular 100 series sintéticas en 2,638 modelos GAN.

Para entender mejor los resultados de la Tabla 8, se graficaron los histogramas de frecuencia y las cajas de bigotes de cada parámetro estadístico de las muestras generadas. En el Gráfico 16 se muestran los histogramas de los estadísticos de las series generadas sintéticamente. Cada parámetro presenta distintos tipos de distribuciones. Todos los histogramas muestran gran dispersión entre sí, con sesgos, y colas pesadas.

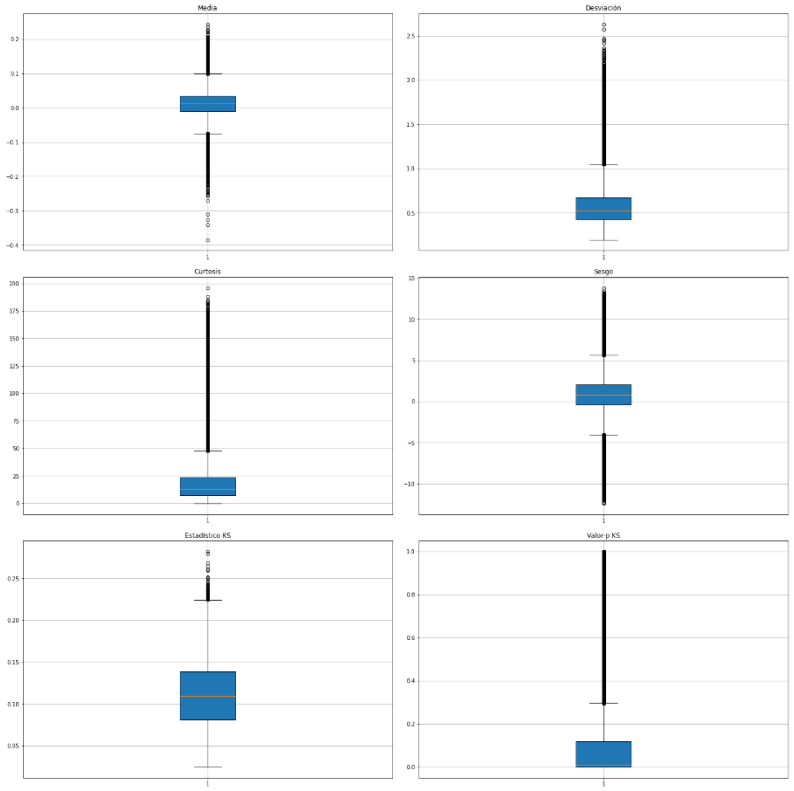


Fuente: elaboración propia.

Gráfico 16. Histograma de frecuencia de las series sintéticas generadas por 2,663 modelos GAN.

En el Gráfico 17 se dibujan las cajas de bigotes para cada estadístico calculado, mostrando, que cada parámetro presenta valores extremos, lo que corrobora lo visto en la Tabla 8 y el Gráfico 16. De estas observaciones es posible inferir que los modelos GAN, a pesar de haberse seleccionado con criterios como el de la precisión y el Kolmogorov – Smirnov, no presentan uniformidad en los datos generados en el sentido estadístico. Esta dispersión en los retornos sintéticos hace necesario tomar precauciones adicionales en el momento de generar muestras sintéticas, pues la evidencia indica que éstas pueden estar por fuera de los límites históricos y que en la práctica son difícil ocurrencia.

Estos límites empíricos dados por los mínimos y máximos de la muestra de retornos de los activos se pueden explicar al tener en cuenta los mecanismos de protección de corto circuito propios de mercados organizados, creadores de mercados y coberturas sobre los activos financieros, que hacen que variaciones abruptas se mantengan dentro ciertos límites, que son propios de cada mercado y que garantizan la estabilidad de los mismos, en especial cuando se habla de frecuencias menores a un día.



Fuente: elaboración propia.

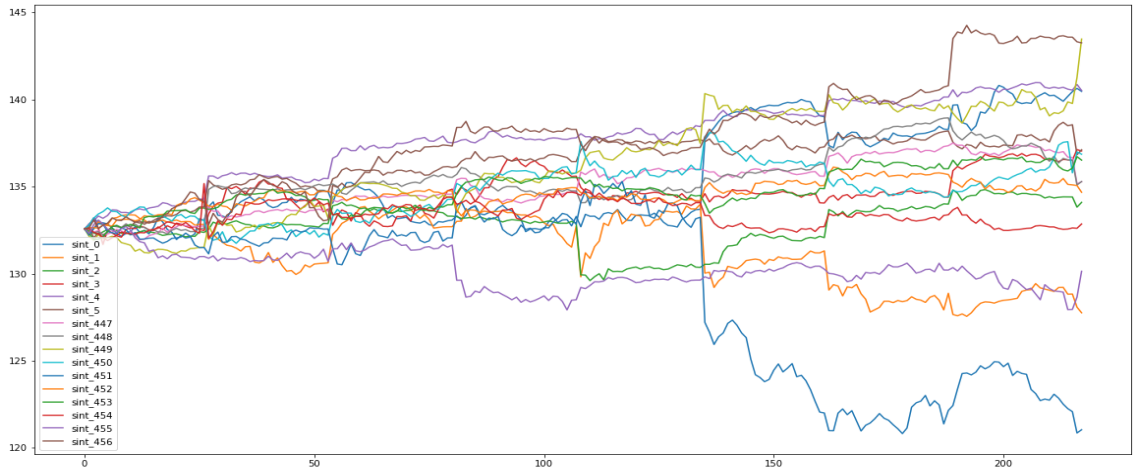
Gráfico 17. Cajas de bigotes de las series sintéticas generadas por 2663 modelos GAN.

Si bien es posible que un retorno anormal suceda ante un descalabro financiero o una noticia de resultados exorbitantes, la probabilidad de estén por fuera de a los límites históricos es infinitamente pequeña, dados los mecanismos de protección de cada mercado. Es por eso que generar muestras simuladas con retornos superiores a los límites de los datos históricos, debe alarmar a quien este modelando series sintéticas y, por lo tanto, estas deben ser omitidas de los datos que entrenan un agente de Aprendizaje por Refuerzo.

Inicialmente se pensó en incorporar los modelos GAN dentro del algoritmo de Aprendizaje por Refuerzo, pero en la práctica no fue posible dada las limitaciones computacionales. Es por esta razón, que en lugar embeber el modelo en el entorno del agente, se planteó la necesidad de crear una base de registros simulados como fuente de datos.

Para este trabajo, se generaron diez secuencias de 216 registros sintéticos cada una, por cada modelo GAN, para un total de 26,380 series. Luego se filtraron los datos que no cumplían el criterio del retorno máximo y mínimo de los datos históricos, afectados cada uno por un incremento del 10%, es decir, al valor mínimo y máximo se decrementó e incrementó un 10% respectivamente (-13%, 8%), para fijar los límites de filtrado para la selección de una serie sintética. Este valor del 10% es arbitrario, y se empleó como un factor de seguridad, en caso de que se pudiera dar en la realidad un retorno superior a los obtenidos de las estadísticas de los datos históricos. Después de este filtrado, se pasaron a un total 13,181 series, que se ordenaron de forma aleatoria, para así evitar que cuando se entrene el agente, este aprendiera algún sesgo originado por el modelo.

Adicionalmente, teniendo en cuenta que el código de Aprendizaje por Refuerzo requiere que la información de entrada sea una serie de precios, se transformó cada simulación pasándola de retornos de precios. Esto se hizo al aplicarle los retornos al último precio de la serie histórica que es de USD$ 132.58. En el Gráfico 18 se muestran ejemplos de series de precios sintéticas seleccionadas de forma aleatoria a partir de los retornos GAN seleccionados para entrenar el agente. Estas series evidencian trayectorias en el tiempo similares a las que ven en la muestra histórica.



Fuente: elaboración propia.

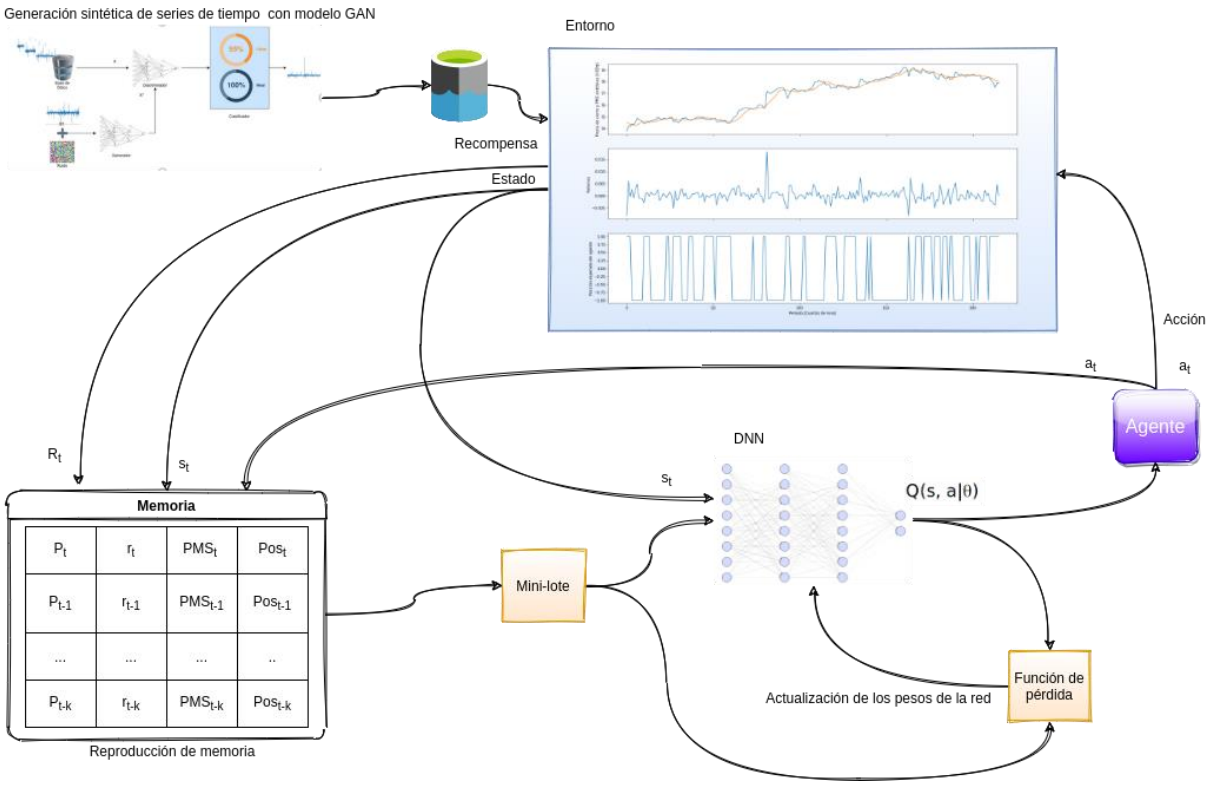
Gráfico 18. Ejemplo de series de tiempo de precios sintéticas generadas por los modelos GAN.

4. IMPLEMENTACIÓN DEL ALGORITMO DE APRENDIZAJE POR REFUERZO PARA COMPRA DE ACCIONES.

El procedimiento que se sigue en la implementación es el que se detalló en la sección 1.8 sobre el Aprendizaje por Refuerzo aplicado a las finanzas y está basado en el modelo planteado por Hilpisch [41]. En el Gráfico 19 se muestra el diagrama del algoritmo de Aprendizaje por Refuerzo empleando series de tiempo sintéticas generadas por una arquitectura GAN para entrenar un agente que compre y venda acciones. El flujo del esquema comienza con la implementación de unas series de retornos por medio de un modelo GAN, los cuales alimentan una base de datos de registros sintéticos, que constituyen la fuente de datos para el modelo de Aprendizaje por Refuerzo.

Dicho algoritmo está compuesto de dos partes: La primera corresponde al generador de entornos o gimnasio financiero, éste se encarga de preparar los datos, eso incluye los cómputos de las señales adicionales al precio. Para este estudio se definieron como señales adicionales para entrenar el agente: los retornos de los precios sintéticos, el promedio móvil simple de cuatro períodos y la volatilidad móvil de 4 períodos. Este último se calculó porque es sabido que las series tienden a producir clústeres de volatilidad que van cambiando en el tiempo y son fuentes de información que pueden servirle al agente como patrón para su aprendizaje. Adicionalmente, se calcula la acción que debe tomar el agente según el estado, ésta es una señal binaria, 1 para adquirir o retener el archivo, y 0 para vender o no tener el activo. El gimnasio financiero, como se llama el entorno en este trabajo, también es el encargado de actualizar sucesivamente al agente con el estado o paso, que es la información que el agente puede ver, son las señales de la instantánea del momento **t** (precio, retorno, promedio móvil, volatilidad móvil y señal de compra) y de calcular la recompensa resultante de la acción en cada paso.

La segunda parte del algoritmo es el agente. Éste es el encargado de definir como aprende el modelo (si está en modo exploración o explotación), en él se construye la arquitectura de la red neuronal profunda para el cálculo de **Q**, ejecuta una acción, reproduce los estados para el entrenamiento del agente y valida que la red si esté retornando los valores deseados.



Fuente: elaboración propia.

Gráfico 19. Diagrama del algoritmo de Aprendizaje por Refuerzo empleando series de tiempo sintéticas generadas por una arquitectura GAN para entrenar agente.

En el Gráfico 20 se presenta la arquitectura elegida para este agente. La cual está conformada por 3 capas densas. La primera inicializa la forma de la red con los datos de entrada de unidades ocultas y el número de señales modeladas en el entorno. Estos valores pueden cambiar según el escenario que se desee modelar. Para el estudio se plantean 24 unidades ocultas y 4 señales o características, y su función de activación es del tipo Relu. La siguiente capa es igual a la primera, y su función de activación es del tipo Relu. La última capa es densa también, pero las unidades de salida son 2 y su función de activación es lineal. La pérdida del modelo se calcula con los errores mínimos cuadrados o MSE (por sus siglas en inglés), el optimizador seleccionado para este caso es RMSprop y que es función de una tasa de aprendizaje **lr** igual a 0.0001.

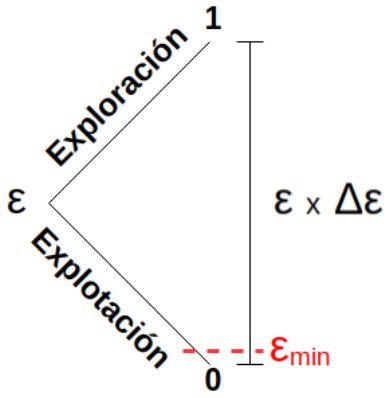
```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_12 (Dense)             (None, 3, 24)             120
_________________________________________________________________
dropout_8 (Dropout)          (None, 3, 24)             0
_________________________________________________________________
dense_13 (Dense)             (None, 3, 24)             600
_________________________________________________________________
dropout_9 (Dropout)          (None, 3, 24)             0
_________________________________________________________________
dense_14 (Dense)             (None, 3, 2)              50
=================================================================
Total params: 770
Trainable params: 770
Non-trainable params: 0
_________________________________________________________________
```

Gráfico 20. Resumen de la implementación de la red del agente Q.

Además de la red, el agente tiene varios atributos o hiperparámetros que se definen dentro de la clase y que son claves en el comportamiento del agente, estos son ε, disminución de $\Delta\varepsilon$, el ε mínimo y γ. Los tres primeros parámetros controlan el *trade-off* entre la exploración y la explotación.

En Aprendizaje por Refuerzo, este dilema se refiere concretamente en como el agente selecciona las acciones, pues éste busca generar la mayor cantidad de recompensa posible, por lo que seleccionará acciones de éxito que ya han sucedido en el pasado (explotación). El problema radica que para conocer estas acciones el agente debe entonces probar nuevas acciones (explotación) de manera estocástica. Y cada acción se evalúa de manera iterativa para encontrar las mejores decisiones de selección [12].

En el Gráfico 21 se presenta de forma esquemática el costo de oportunidad del dilema exploración *vs*. explotación. Para explicar cómo funciona este *trade-off*, hay que entender que ε es un umbral dinámico que se compara con un valor aleatorio, y según sea la comparación de este último con el valor de ε, la acción va a ser una predicción o una acción conocida de un estado anterior. El valor de ε está entre 0 y 1, y va cambiando en cada iteración al ser afectado por un coeficiente de disminución $\Delta\varepsilon$, actualizando su valor en cada paso hasta alcanzar un valor mínimo. De ahí en adelante el algoritmo se concentra en la explotación, dejando de predecir nuevos valores, y por lo tanto tomando las acciones de la memoria.



Fuente: elaboración propia.

Gráfico 21. Esquema del costo del dilema de explotación de ε.

En el trabajo ε comienza desde un máximo de 1 a un valor mínimo de 0.05, la forma como se reduce el parámetro está dada por el coeficiente de disminución que para este caso se fijó en Δε = 0.925.

El valor γ es el parámetro que define como se afecta a las recompensas futuras. Si el valor es en magnitud baja, se da más peso a las recompensas pasadas, en caso contrario, se da más peso a las recompensas recientes [39]. La memoria que almacena los estados es una lista con capacidad para 2700 elementos, su selección fue arbitraria.

5. EVALUACIÓN DEL MODELO

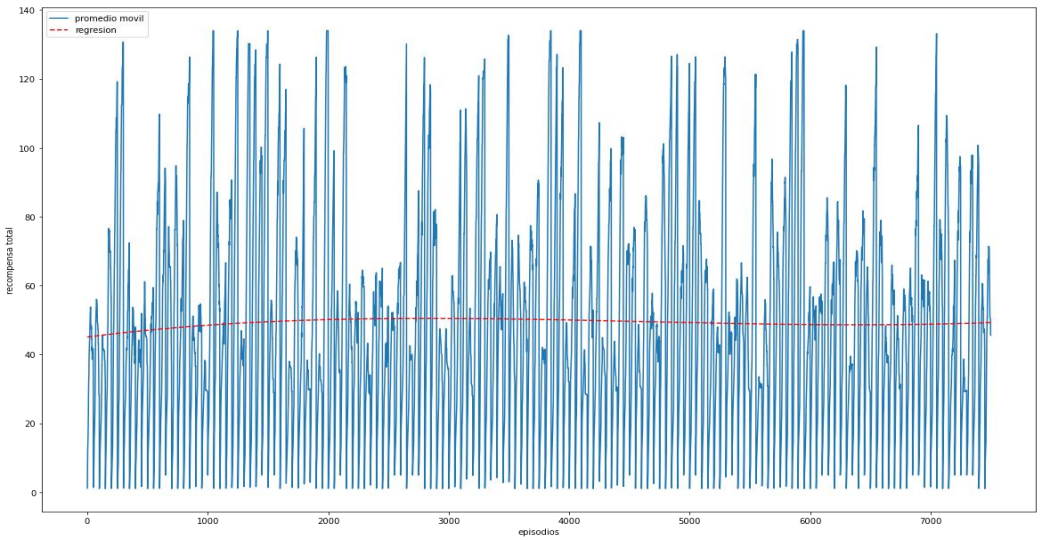
Para ejecutar el entrenamiento del modelo planteado en los dos puntos anteriores se empleó una GPU GTX 1070. Para el desarrollo de las redes y de los agentes, se codificó con Python 3.7 y la librería Tensorflow versión 2.3.

Cada serie se corre 50 veces, y cada corrida de una acción se entrena con 10,000 iteraciones. Donde se van a evaluar 150 series sintéticas, cada una de éstas tiene una longitud de 216 registros, que se parte en 3 partes: 135 para el entrenamiento, 27 para la validación y 54 para las pruebas. Analizando los resultados después de generar las series, se encontró que en las pruebas era mejor utilizar datos reales

que no hubieran sido utilizadas para entrenar los modelos GAN en lugar de los datos sintéticos. Para este estudio, la muestra de pruebas corresponde a registros de cuarto de hora de precios de la acción que van desde el 4 de enero del 2021 a 31 de mayo del mismo año.

De esta serie se tomaron 540 registros para la evaluación del desempeño del algoritmo por fuera de entrenamiento. La medida de desempeño es la riqueza acumulada a partir de una inversión inicial de USD$ 10,000. No se consideran costos de transacción, por tanto, el modelo no tiene fricciones de ningún tipo. El resultado del comportamiento del agente se compara con una estrategia pasiva de comprar y retener (buy and hold).

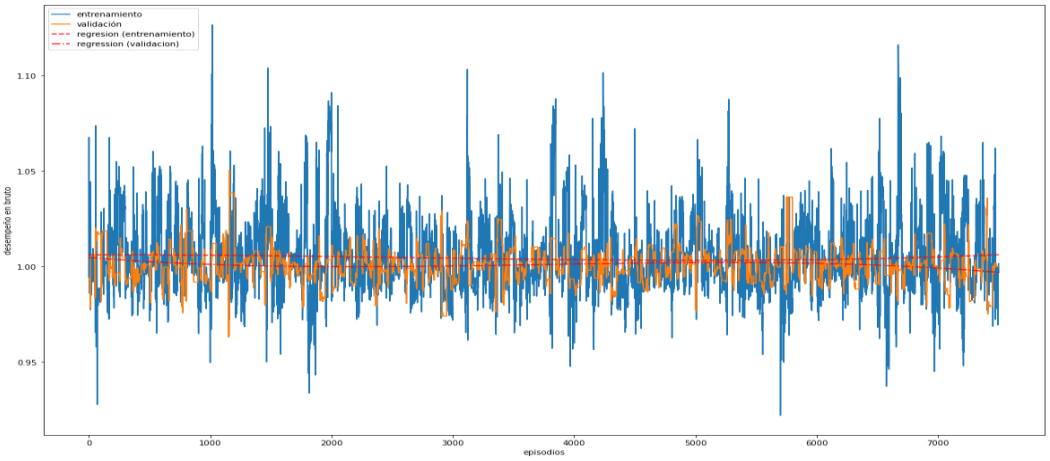
En el Gráfico 22 se muestra el comportamiento del promedio de las recompensas durante el entrenamiento para todas las series sintéticas. Estes se fijó en 27 períodos, que equivalen a los registros de un día de transacciones, una medida utilizada por los participantes de mercado como referencia de transacciones. Como puede observarse el promedio de las recompensas se mantiene estable a través del tiempo, lo que evidencia que el agente no mejora su desempeño a medida que se aumentan los episodios de entrenamiento.



Fuente: elaboración propia.

Gráfico 22. Esquema del costo del dilema de explotación de ε.

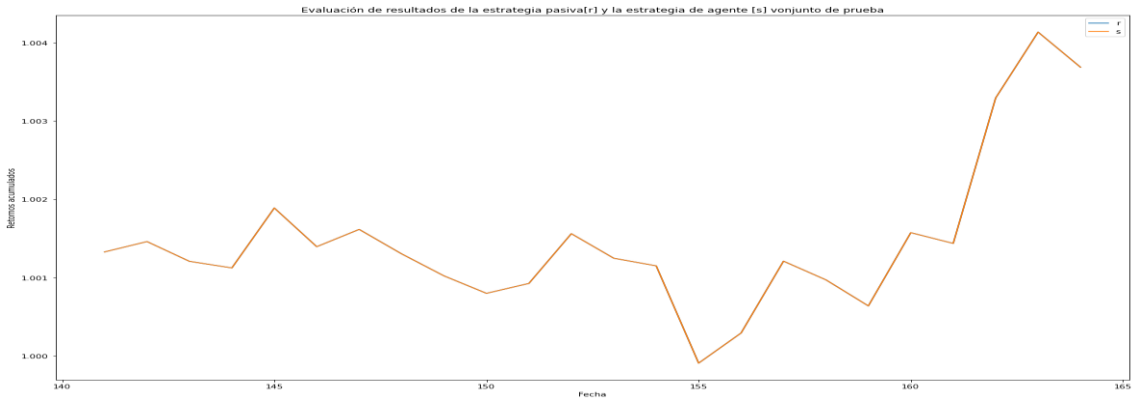
En el Gráfico 23 se muestra la evolución del desempeño bruto, es decir, el comportamiento de la recompensa dada por la selección de la acción (1 o 0) más la recompensa financiera (retornos). Como puede observarse, tanto la recompensa de entrenamiento como la de validación tienden a tener comportamientos estables a través del tiempo, no hay mejorías por entrenar con mayor número de series.



Fuente: elaboración propia.

Gráfico 23. Esquema del costo del dilema de explotación de ε.

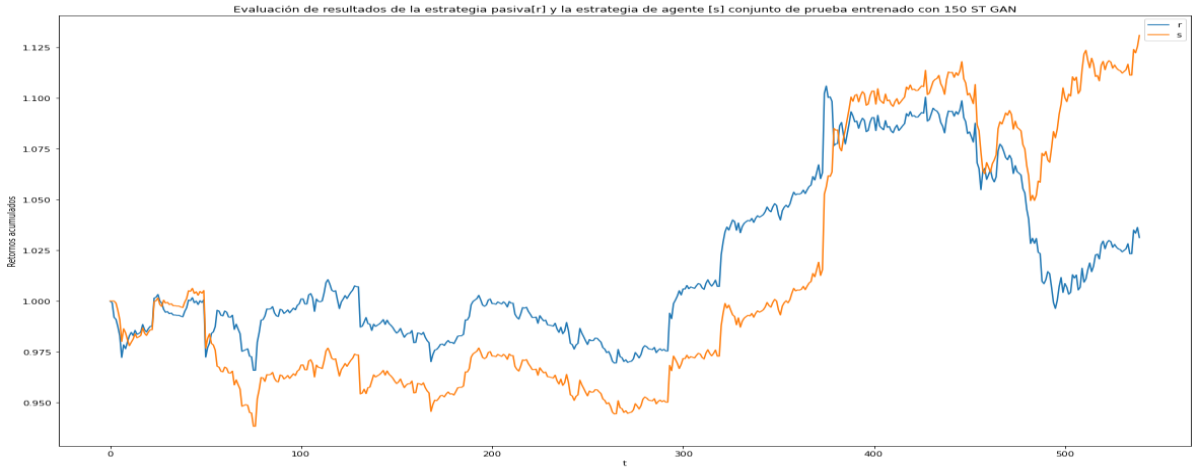
Para evidenciar el comportamiento del algoritmo se graficó la evolución de los retornos acumulados a través del tiempo de la estrategia pasiva y activa (la del agente). En el Gráfico 24 se puede observar que ambas estrategias se superponen (se espera que la estrategia del agente debe producir una línea por encima de la estrategia pasiva), como si no existiera una diferencia entre ambas, esto parece indicar que el agente tiende a sobre ajustar su comportamiento, para hacerlo idéntico al comportamiento de la serie original y no busca generar riqueza. Este comportamiento del agente muestra un exceso de ajuste en su desempeño, y no de maximización de la utilidad, que es el objetivo mismo del algoritmo.



Fuente: elaboración propia.

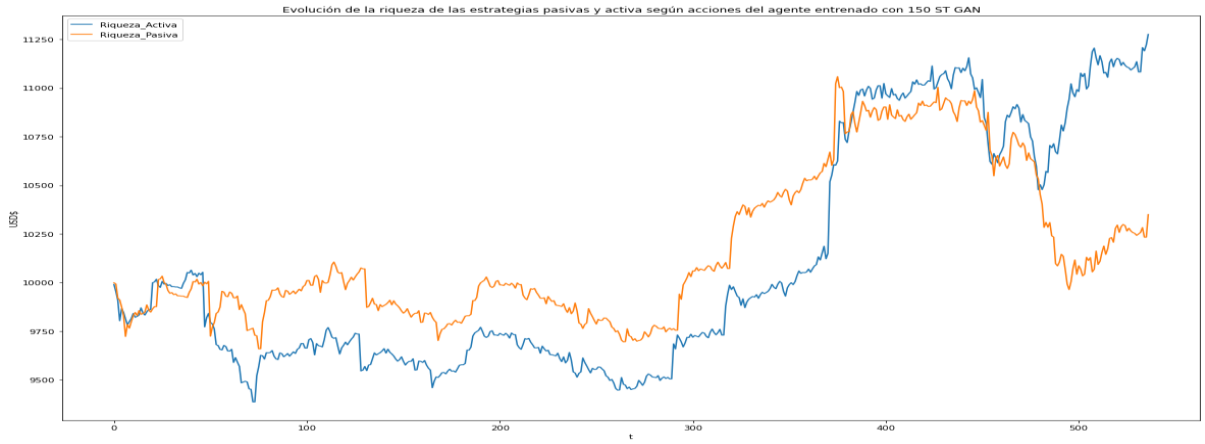
Gráfico 24. Evolución de los retornos acumulados estrategias pasivas y activa última serie sintética analizada.

Lo anterior supone un problema, pero como puede verse en las pruebas que se hicieron con datos nuevos, este sobre ajuste no es relevante. En los gráficos 25 y 26 se muestra la evolución de los retornos de las estrategias (pasiva y activa) y la evolución de la riqueza generada por ambas estrategias. Ambos gráficos se complementan, la diferencia entre ellos radica en las unidades de valoración. El primero está en porcentaje y el segundo en dólares que se calcularon asumiendo que hay una inversión inicial de USD$ 10,000.00 al inicio de la evaluación, lo que permite visualizar los resultados en términos de rentabilidad y de riqueza. La razón para mostrarlos de esta manera es dar una mayor claridad en los efectos de las decisiones del agente a través de un marco temporal y en métricas que utilizan normalmente los participantes en los mercados. De estos gráficos se evidencia que el resultado final de la estrategia activa es de USD$ 11,275.79 en 540 períodos, o un mes de transacciones, en otras palabras, en dos meses el agente logró 12.76% de rentabilidad. Si bien el resultado es alentador, debe analizarse la evolución de los resultados de la estrategia a través del tiempo. En realidad, la estrategia activa tuvo desempeños inferiores a través del tiempo, empezó a cambiar al final de la evaluación el 21 de enero del 2021, dato 350 aproximadamente en esta serie.



Fuente: elaboración propia.

Gráfico 25. Evolución de los retornos acumulados estrategias pasivas y activa obtenidas de analizar datos reales de prueba.



Fuente: elaboración propia.

Gráfico 26. Evolución de la riqueza acumulada estrategias pasivas y activa obtenidas de analizar datos reales de prueba.

Estos resultados, muestran debilidades en el sistema de transacciones planteado por el algoritmo, por lo que se hizo necesario evaluar otras alternativas de estimación modificando parámetros en el entrenamiento. En la literatura no se encontró ningún derrotero, guía o recomendaciones que permitieran definir un procedimiento de calificación de cuáles características eran las más adecuadas de optimizar y lograr mejores resultados. Por tanto, se plantearon los siguientes escenarios para entender la sensibilidad en la generación de utilidades o riqueza

del modelo: cambios en el número de episodios de entrenamiento; cambios en el valor γ; cambios en los tamaños de la muestra de prueba; cambios en el número de neuronas de la red; y cambios en el número de series sintéticas empleadas para el entrenamiento.

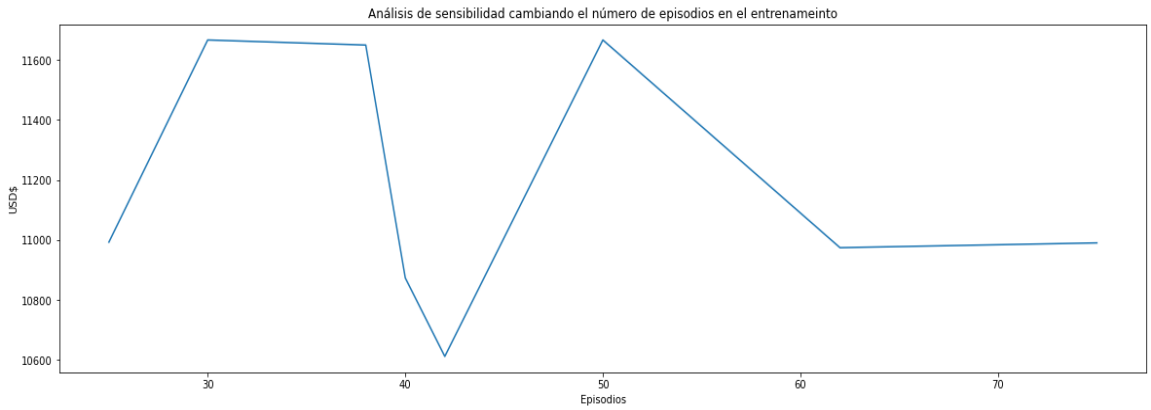
En la Tabla 9 se presentan los modelos y resultados consolidados obtenidos en diversos experimentos. En cada caso, se modificaron una o más variables para entender la sensibilidad del modelo ante cambios de estas. Se modificaron el número del tamaño de la muestra (N_pruebas), la cantidad de neuronas o unidades ocultas de la red Q, el γ de la ecuación de Bellman, el número de episodios y el número de muestras que alimentan el entorno del agente. Como variables de evaluación se calcularon la riqueza generada según el horizonte de tiempo planteado por el tamaño de la muestra, el retorno bruto de la estrategia activa y pasiva para cada modelo, y su equivalente efectivo anual.

	Modelo	N_pruebas	Neuronas	γ	Episodios	N_muestras	Riqueza	Retorno_estrategia	Retorno_pasivo	Est._activa	Est._Pasiva
0	1	540	24	0.3500	50	150	11275.79	12.76%	3.09%	354.00%	46.70%
1	2	540	24	0.3500	25	5	10992.86	9.93%	3.09%	229.61%	46.70%
2	3	540	24	0.3500	30	5	11666.53	16.67%	3.09%	597.37%	46.70%
3	4	540	24	0.3500	38	5	11649.65	16.50%	3.09%	584.77%	46.70%
4	5	540	24	0.3500	50	5	11666.53	16.67%	3.09%	597.37%	46.70%
5	6	540	24	0.3500	62	5	10974.44	9.74%	3.09%	222.71%	46.70%
6	7	540	24	0.3500	75	5	10990.34	9.90%	3.09%	228.65%	46.70%
7	8	540	24	0.3500	50	25	11092.00	10.92%	3.09%	269.08%	46.70%
8	9	540	24	0.3500	42	5	10612.06	6.12%	3.09%	111.39%	46.70%
9	9'	540	24	0.3500	40	5	10873.59	8.74%	3.09%	187.28%	46.70%
10	10	540	24	0.7000	50	5	11572.14	15.72%	3.09%	529.52%	46.70%
11	11	540	24	0.1750	50	5	11068.58	10.69%	3.09%	259.38%	46.70%
12	12	540	24	0.5250	50	5	11730.00	17.30%	3.09%	646.72%	46.70%
13	13	540	24	0.4375	50	5	11666.53	16.67%	3.09%	597.37%	46.70%
14	14	540	24	0.6125	50	5	11577.97	15.78%	3.09%	533.53%	46.70%
15	15	540	24	0.5250	38	5	11696.67	16.97%	3.09%	620.42%	46.70%
16	16	540	24	0.5250	50	10	10987.69	9.88%	3.09%	227.66%	46.70%
17	17	135	24	0.5250	50	5	9999.24	-0.01%	-0.80%	-0.38%	-33.29%
18	18	1080	24	0.5250	50	5	10681.83	6.82%	-5.74%	51.52%	-31.09%
19	19	2754	24	0.5250	50	5	10828.66	8.29%	-5.00%	21.74%	-11.90%
20	20	540	16	0.5250	50	5	10097.89	0.98%	3.09%	13.06%	46.70%
21	21	540	32	0.5250	50	5	10390.64	3.88%	3.09%	61.59%	46.70%
22	22	135	24	0.5250	38	5	9673.17	-3.27%	-0.80%	-81.26%	-33.29%
23	23	135	24	0.5250	75	5	10034.54	0.35%	-0.80%	18.98%	-33.29%
24	24	540	24	0.5250	50	1	10952.55	9.53%	3.09%	214.70%	46.70%
25	25	540	24	0.5250	30	1	10633.23	6.33%	3.09%	116.76%	46.70%
26	26	540	24	0.5250	75	1	11576.15	15.76%	3.09%	532.28%	46.70%
27	27	135	24	0.5250	30	1	10164.22	1.64%	-0.80%	127.26%	-33.29%
28	28	135	24	0.5250	75	1	9599.17	-4.01%	-0.80%	-87.28%	-33.29%
29	29	540	24	0.5250	70	5	10959.56	9.60%	3.09%	217.24%	46.70%
30	30	135	24	0.5250	50	1	9991.15	-8.85%	-0.80%	-99.06%	-33.29%
31	31	540	24	0.5250	50	5	10393.98	3.94%	3.09%	62.72%	46.70%
32	32	540	24	0.5250	50	5	10443.50	4.44%	3.09%	72.77%	46.70%

Tabla 9. Resumen de resultados de diversos experimentos.

En primer lugar, se corrió el modelo para un γ constante de 0.35 (el mismo del modelo inicial) y diferentes episodios (25, 30, 38, 40, 42, 50, 62, 75). En el Gráfico 27 se aprecia la riqueza generada por el modelo para las diferentes corridas. Este gráfico es muy diciente porque no muestra una línea suavizada y con variaciones poco pronunciadas, en especial para 42 episodios en donde el valor del portafolio

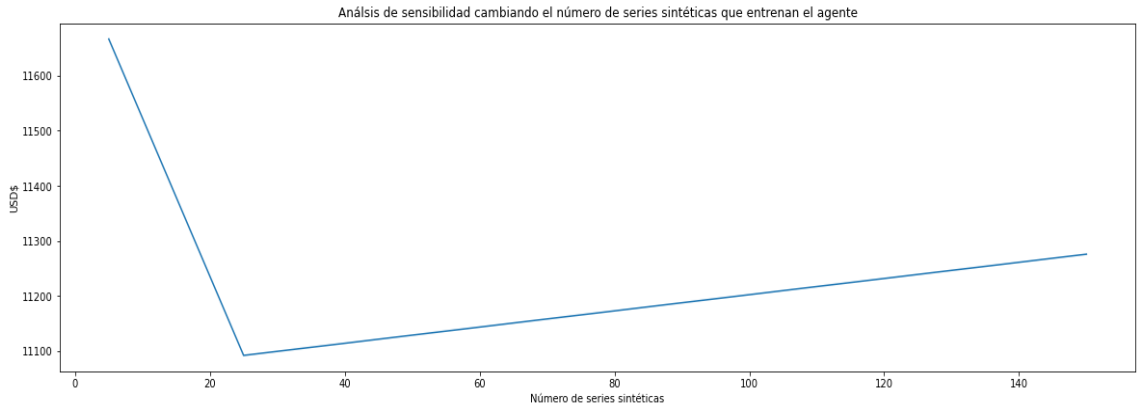
baja a USD$ 10,612.06 y luego vuelve a subir. En principio parecería un error, pero no es el caso, es probable que en ese número de corridas el modelo encontrara un mínimo local subóptimo que ocasionara dicho comportamiento.



Fuente: elaboración propia.

Gráfico 27. Sensibilidad del modelo a cambios en el número de episodios.

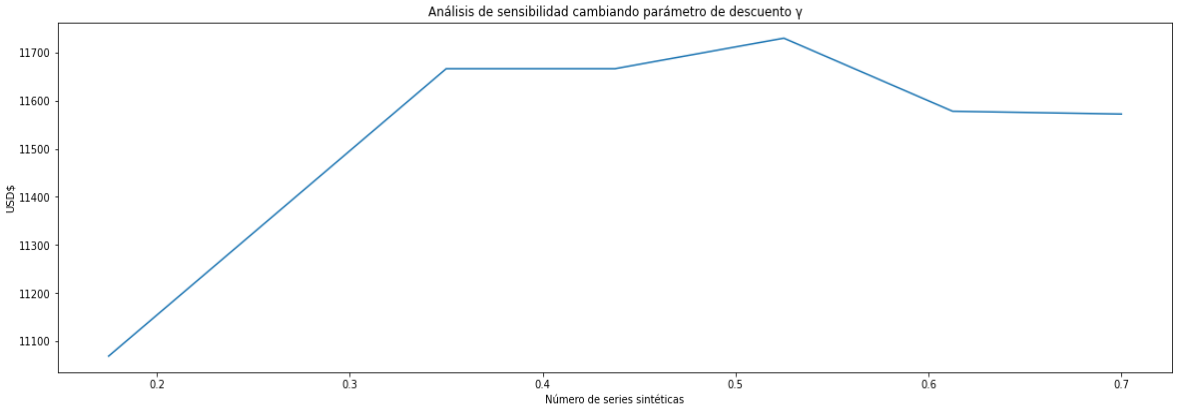
Para el siguiente análisis, se explora el efecto de la cantidad de series sintéticas en el entrenamiento, definiendo tres tamaños de muestras: 5, 25 y 150. Al igual que con los cambios en el número de episodios, no hay evidencia de que existiera un patrón en el cual el diseñador del algoritmo pudiera establecer un tamaño de muestra (series sintéticas) óptimo para entrenar el agente.



Fuente: elaboración propia.

Gráfico 28. Sensibilidad del modelo a cambios en el número de series sintéticas de entrenamiento.

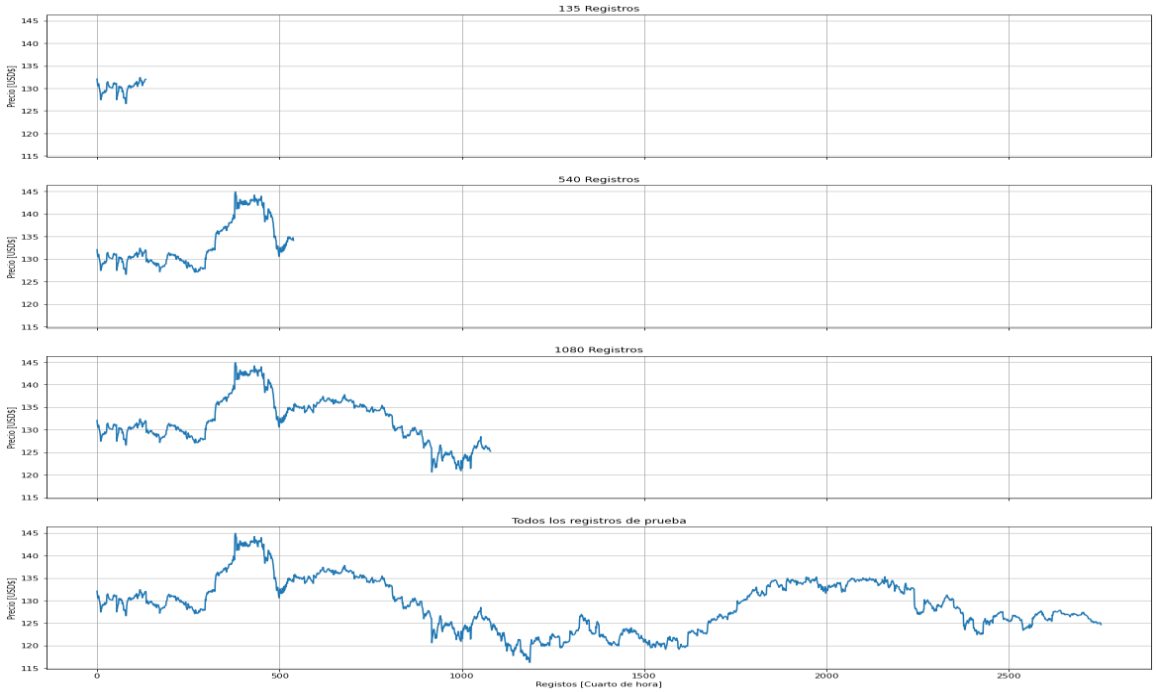
En el Gráfico 29 se presenta la sensibilidad de la riqueza al cambio en el factor de descuento γ. El mejor resultado se da con un γ de 0.525, obteniendo una rentabilidad de un mes de 17.3%, superior al primer escenario y con menos series de entrenamiento, y por lo tanto más fácil de entrenar.



Fuente: elaboración propia.

Gráfico 29. Sensibilidad del modelo a cambios en el parámetro de descuento γ.

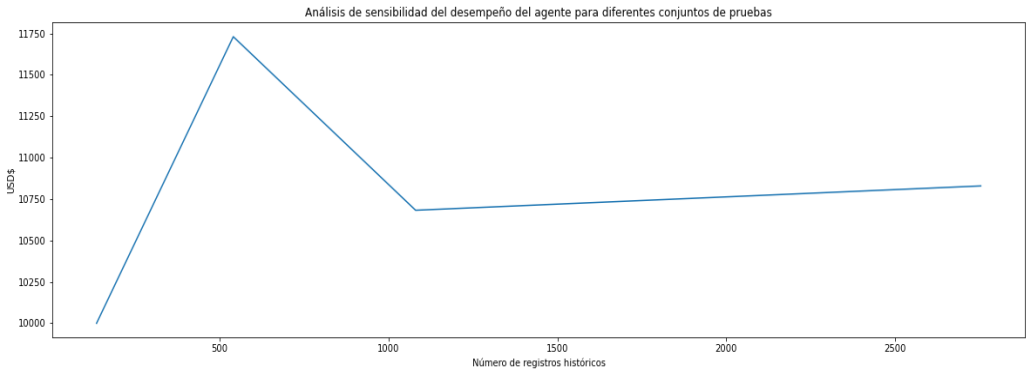
Para entender la capacidad del agente de generar riqueza en el tiempo, se prueba el algoritmo con diferentes conjuntos de prueba, modificando la longitud de las series evaluadas. En el Gráfico 30 se aprecia en escala la diferencia entre las distintas series evaluadas. La primera serie corresponde a un rango de datos de una semana (135). La segunda serie corresponde a un mes de registros (540). La tercera serie equivale a dos meses de datos (1080), y por último el conjunto total de datos registrados en el período seleccionado para las pruebas (2754).



Fuente: elaboración propia.

Gráfico 30. Sensibilidad del modelo a cambios en el parámetro de descuento γ.

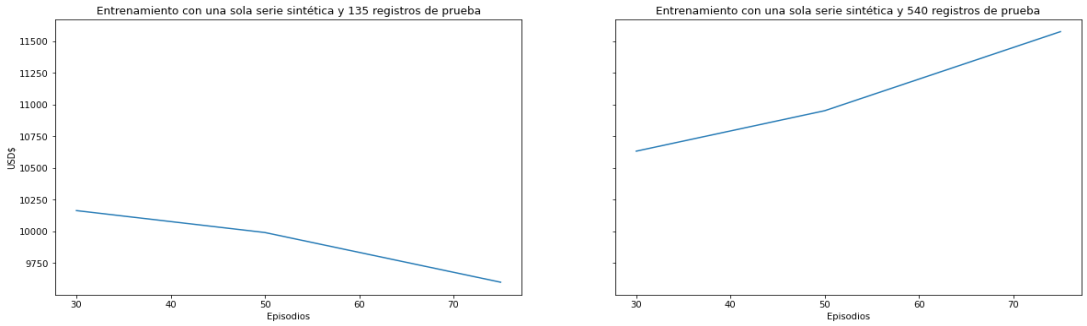
En el Gráfico 31 se aprecia de forma más clara el impacto de la longitud del conjunto de prueba en el desempeño del agente. De la Tabla 9 se desprende que el agente se comporta mejor con datos de un mes (540 registros), pues es con ese número de registros en donde alcanza la mayor riqueza.



Fuente: elaboración propia.

Gráfico 31. Sensibilidad del modelo a cambios en el tamaño del conjunto de pruebas.

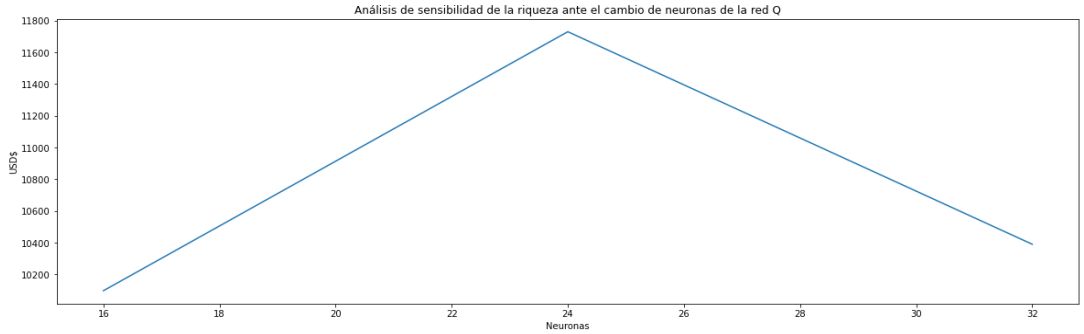
En el Gráfico 32 se comparan dos escenarios en donde los modelos se prueban con 135 y 540 registros históricos respectivamente. Estos fueron entrenados con diferentes episodios y con una sola serie sintética como datos de entrada para el agente. Al observar amabas gráficas, el resultado es contradictorio, con pocos datos de prueba el comportamiento es inferior, que si se validarán más datos, y no es posible concluir nada al respecto.



Fuente: elaboración propia.

Gráfico 32. Sensibilidad del modelo a cambios en el tamaño del conjunto de pruebas y entrenado con una sola serie sintética.

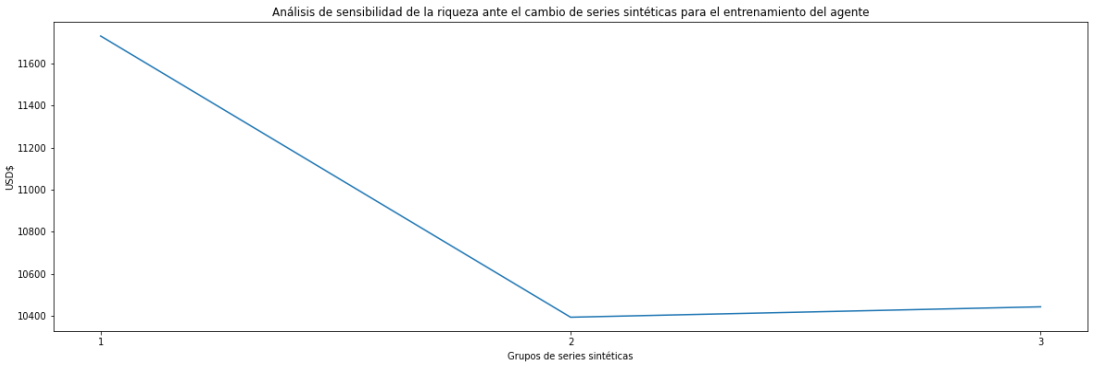
Otro parámetro que se cambió fue el de número de neuronas en la red Q, que hasta el momento ha sido de 24 unidades ocultas. Para el análisis de sensibilidad se probaron arquitecturas con 16 y 32 neuronas. Con base en el Gráfico 33 se evidencia que el valor óptimo es 24, en donde se genera la mayor riqueza.



Fuente: elaboración propia.

Gráfico 33. Sensibilidad del modelo a cambios en el número de filtros de la red neuronal Q.

Adicional a los análisis anteriores, se validó que tanto puede cambiar el resultado de la riqueza si se intercambian de forma aleatoria las series sintéticas de entrenamiento, pues hasta acá siempre se han empleado las primeras cinco series sintéticas de la base de datos. Para poder evaluar esta sensibilidad, se seleccionaron de manera aleatoria diez de las 13,180 series sintéticas almacenadas en la base de datos, que luego se separaron en dos grupos de cinco series cada uno. En el Gráfico 34 se presenta la sensibilidad de los resultados a cambios en los datos de entrada. Es evidente que los mejores valores dependen de las series de entrenamiento iniciales. Al parecer, el agente entrenado con las cinco primeras series ficticias presenta la mayor riqueza, cambios en las mismas disminuyeron considerablemente el desempeño del mismo.



Fuente: elaboración propia.

Gráfico 34. Sensibilidad al cambio de las series sintéticas para entrenamiento del agente.

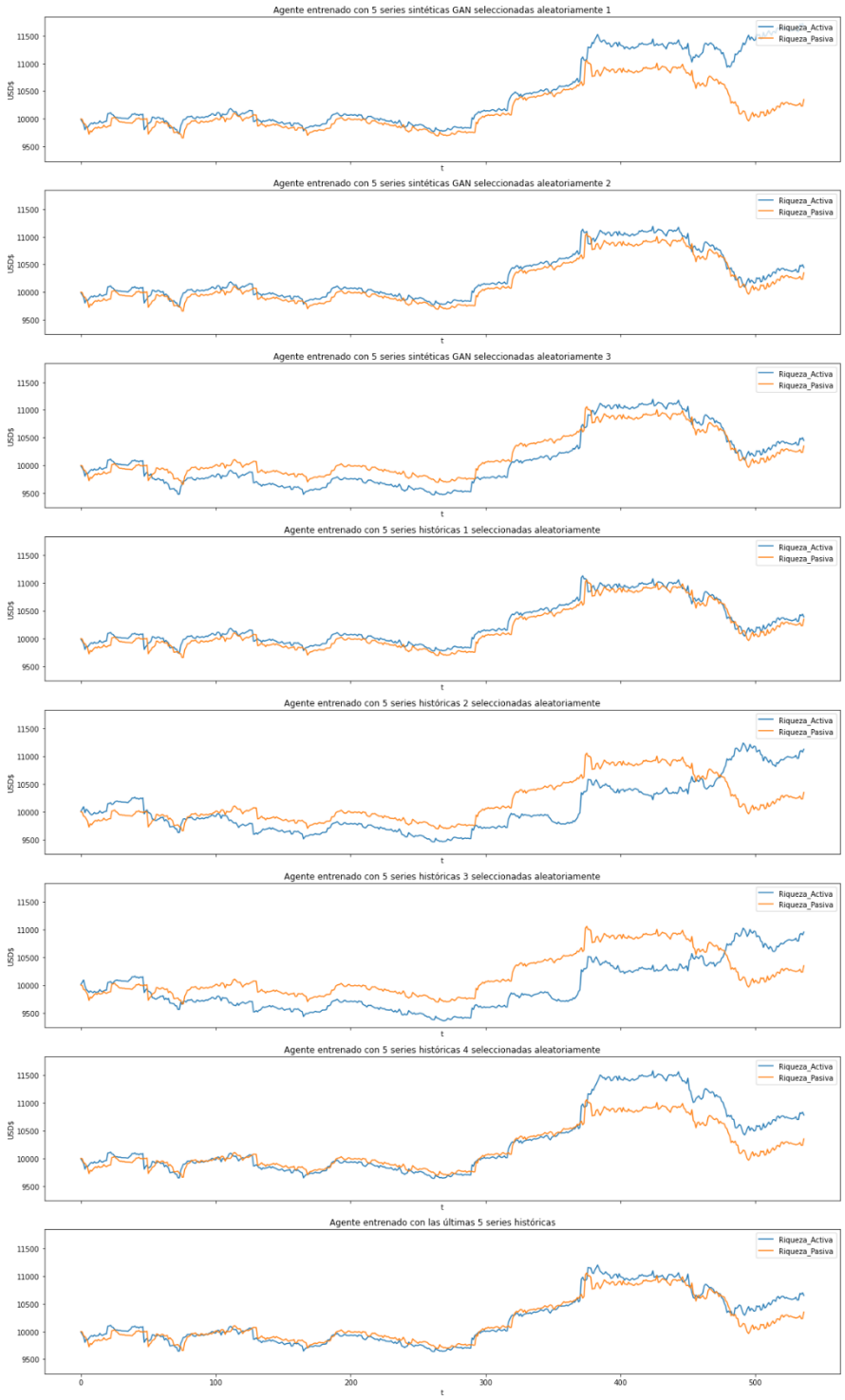
La sensibilidad en los resultados ante cambios en los datos de entrada lleva a preguntarse si en lugar de utilizar datos sintéticos era preferible entrenar el agente con muestras de la serie histórica con longitud igual a las de las generadas por los modelos GAN. Para responder esta pregunta, se segmentó la serie histórica en 188 series todas de igual tamaño, cada una de 216 registros. Luego se generaron cuatro grupos de cinco acciones cada uno de forma aleatoria y, un último grupo con las últimas cinco series de los datos históricos. En la Tabla 10, se resumen los resultados para el mejor modelo del entrenamiento con series sintéticas y entrenamiento con series históricas. Las métricas de evaluación fueron la riqueza, el rendimiento de la estrategia activa (r% act.) y el rendimiento de la estrategia pasiva (r% pas.) y sus respectivas conversiones a interés efectivo anual. En

promedio, las series reales obtienen mejores resultados, pero ninguna supera al mejor resultado entrenado con series sintéticas.

Fuente	Riqueza	r % act.	r % pas.	r % act. EA	r % pas. EA
Series Sintética Aleatorias 1	**11,730.00**	**17.30%**	**3.09%**	**646.72%**	**46.70%**
Series Sintética Aleatorias 2	10,393.98	3.94%	3.09%	62.72%	46.70%
Series Sintética Aleatorias 3	10,443.5	4.44%	3.09%	72.77%	46.70%
Series Reales Aleatorias 1	10,390.64	3.91%	3.09%	62.07%	46.70%
Series Reales Aleatorias 2	11,128.59	11.29%	3.09%	284.72%	46.70%
Series Reales Aleatorias 3	10,954.71	9.55%	3.09%	215.48%	46.70%
Series Reales Aleatorias 4	10,746.99	7.47%	3.09%	120.69%	46.70%
Últimas 5 Series Reales	10,648.41	6.48%	3.09%	120.69%	46.70%

Tabla 10. Comparativo entre resultados de entrenamiento con diferentes series sintéticas y reales.

En el Gráfico 35, se muestra la evolución de la riqueza generada por el agente según su fuente de datos de entrenamiento. Como puede verse, los comportamientos a través del tiempo del agente son muy diferentes, no se puede identificar un patrón que permita inferir una regla específica para su puesta en productivo. De todas formas, de los resultados obtenidos se puede deducir que las series sintéticas pueden servir para obtener mejores rendimientos.



Fuente: elaboración propia.

Gráfico 35. Comparativo entre resultados de entrenamiento con diferentes series sintéticas y reales.

6. CONCLUSIONES

6.1 CUMPLIMIENTO DE OBJETIVOS

6.1.1 OBJETIVO ESPECÍFICO 1.

Caracterizar las metodologías de Redes Generativas Adversarias y de Aprendizaje por Refuerzo empleados para la negociación algorítmica.

Al caracterizar las metodologías de Redes Generativas Adversarias y de Aprendizaje por Refuerzo, se logró entender los fundamentos necesarios para poder enlazar las dos fuentes de conocimiento y así poder proponer un método para implementar un agente de negociaciones en mercados electrónicos financieros.

6.1.2 OBJETIVO ESPECÍFICO 2.

Desarrollar un modelo de Aprendizaje por Refuerzo Profundo para negociación algorítmica que incluya una arquitectura de Redes Generativas Adversarias para el entrenamiento del agente.

Teniendo las bases teóricas estudiadas en la caracterización de las metodologías propuestas en este trabajo, se desarrolló un modelo de Aprendizaje por Refuerzo cuyo entorno es alimentado por series sintéticas generadas por Redes Generativas Adversarias. Este modelo se implementó con las herramientas de Python y Tensorflow.

6.1.3 OBJETIVO ESPECÍFICO 3.

Validar la modelación propuesta con datos de una acción de un mercado de acciones electrónico.

El modelo que se desarrolló fue contrastado con datos con la acción de Apple en una frecuencia de 15 minutos para un período comprendido entre el primero de

enero de 2015 a 28 de mayo de 2021. Ésta se seleccionó de un grupo de 98 acciones listadas en *Standard & Poor's,* con base en criterios de calidad de la información disponible. El conjunto de datos se dividió en dos secciones, la primera y más extensa comprendida entre el primero de enero del 2015 a 31 de diciembre de 2020 sirvió para entrenar los modelos GAN que generaron los datos sintéticos. El segundo conjunto de datos comprendido entre el primero de enero de 2021 y 28 de mayo de 2021 sirvió para probar el agente del algoritmo de Aprendizaje por Refuerzo entrenado con los datos sintéticos GAN.

Con base en estos conjuntos de datos se hicieron dos tipos de validaciones. La primera está relacionada con la definición y calidad de los modelos GAN que generan los datos sintéticos para entrenar el agente del algoritmo de Aprendizaje por Refuerzo. La segunda validación se hizo con un conjunto de pruebas, permitiendo explorar diferentes análisis de sensibilidad y contrastar con datos reales la viabilidad de los modelos implementados como alternativas para la automatización de agentes negociadores de activos financieros en mercados electrónicos.

6.1.4 OBJETIVO GENERAL

Desarrollar un modelo de Redes Generativas Adversarias como entorno de entrenamiento para un algoritmo de Aprendizaje por Refuerzo aplicado en el contexto de la negociación algorítmica.

Al lograr los tres objetivos específicos se alcanza el objetivo general del trabajo. Se implementó un modelo de Aprendizaje por Refuerzo Profundo para negociaciones algorítmicas, cuyo agente fue entrenado con múltiples series sintéticas generadas por un modelo de Redes Generativas Adversarias, respondiendo la pregunta planteada para este estudio, al simular series sintéticas que preservan las dinámicas propias de las series de tiempo financieras, y que, sirvieron para entrenar un agente de un algoritmo de Aprendizaje por Refuerzo. Evidenciando el potencial del Aprendizaje por Refuerzo es relevante para la aplicación en las finanzas.

Como trabajos futuros, se pueden hacer este mismo tipo de análisis, para más series de tiempo y de diversos tipos de activos financieros. También, se pueden

explorar otros tipos de arquitectura diferentes a la empleada en este trabajo y validar si modelos más complejos de agentes de Aprendizaje Profundo Q logran mejores resultados.

6.2 POSIBLES USOS DEL APRENDIZAJE POR REFUERZO PROFUNDO APLICADO A NEGOCIACIONES ELECTRÓNICAS

El algoritmo empleado para este trabajo es una muestra alternativa a los algoritmos desarrollados por reglas. La ventaja de un algoritmo de Aprendizaje pro Refuerzo Profundo es que éste está en capacidad de aprender en sincronía los patrones de los mercados según las dinámicas propias del mismo (tanto en el que opera como en otros). En este orden de ideas, la implementación de un programa de esta categoría en un ambiente productivo representa una ventaja frente a otras alternativas, pues este es capaz de recoger relaciones no lineales y temporales, que dadas las dimensiones de la información disponible en los mercados financieros [7] no son posibles de automatizar por códigos convencionales o de seguir por una persona al mismo tiempo. Tienen la ventaja también, que pueden ser recalibrados según se vayan dando los eventos del mercado, lo que permite establecer mecanismos de control de riesgos y efectividad de sus acciones de manera automática. Estas capacidades dejan entrever posibles usos prácticos de dichas metodologías en la practica como: hacedores de mercado, negociaciones de alta frecuencia, clasificación ejecución de acciones, y robots especializados en mercados entre otros posibles usos.

7. BIBLIOGRAFÍA Y FUENTES DE INFORMACIÓN

[1] C. Lattemann, P. Loos, j. Gomolka, H.P. Burghof, A. Breuer A, Gomber P, M. Krogmann, J. Nagel, R. Riess, R. Riordan, R.Zajonz (2012) High Frequency Trading. Kosten und Nutzen im Wertpapierhandel und Notwendigkeit der Marktregulierung. WIRTSCHAFTSINFORMATIK. Gabler Verlag. Doi: 10.1007/s11576-012-0311-9.

[2] G.N. Gregoriou. "The Handbook of HIGH FREQUENCY TRADING". Elsevier Inc. 2015.

[3] V. Bacoyannis, V. Glukhov, T. Jin, J. Kochems, and D. R. Song. “Idiosyncrasies and challenges of data driven learning in electronic trading”. 2018. arXiv:1811.09549v2.

[4] H. Dong, Z. Ding and S. Zhang. "Deep Reinforcement Learning, An introduction". Springer Nature Singapore Pte Ltd. 2020. doi:10.1007/978-981-15-4095-0.

[5] M. López de Prado, Marcos. "The Future of Empirical Finance". Journal of Portfolio Management, 41(4), 2015. Doi: 10.2139/ssrn.2609734.

[6] M. López de Prado. "The 10 Reasons Most Machine Learning Funds Fail". JPM 2018, 44 (6) 120-133 doi: 10.3905/jpm.2018.44.6.120.

[7] M. López de Prado. “Advances in Financial Machine Learning”. Wiley. 2018.

[8] P. Treleaven, M. Galas, and V. Lalchand. "Algorithmic Trading Review". Communications of the ACM. november 2013. Vol. 56, no. 11. Doi: 10.1145/2500117.

[9] M. Karpe. "An overall view of key problems in algorithmic trading and recent progress". 2020. arXiv:2006.05515v1.

[10] T. G. Fischer, “Reinforcement learning in financial markets - a survey”. FAU Discussion Papers in Economics, No. 12/2018, Friedrich-Alexander-Universität Erlangen-Nürnberg, Institute for Economics, Nürnberg. 2018.

[11] P. Kolm and G. Ritter, “Modern Perspectives on Reinforcement Learning in Finance”. The Journal of Machine Learning in Finance, Vol. 1, No. 1, 2020. Doi: 10.2139/ssrn.3449401.

[12] R. S. Sutton and A. G. Barto. "Reinforcement Learning: An Introduction" second edition The MIT Press. 2018.

[13] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare and J. Pineau, “An Introduction to Deep Reinforcement Learning”, Foundations and Trends in Machine Learning: Vol. 11, No. 3-4. Doi: 10.1561/2200000071.

[14] M. López de Prado. “Tactical Investment Algorithms”. 2019. Available at SSRN: https://ssrn.com/abstract=3459866 or http://dx.doi.org/10.2139/ssrn.3459866.

[15] S. Assefa, D. Dervovic, M. Mahfouz, T. Balch, P. Reddy, and M. Veloso. "Generating Synthetic Data in Finance: Opportunities, Challenges and Pitfalls". In NeurIPS'19 Workshop on Robust AI in Financial Services, Vancouver, Canada, December 2019.

[16] K. B. Hansen. “The virtue of simplicity: On machine learning models in algorithmic trading”. Big Data & Society. 2020. Doi: 10.1177/2053951720926558.

[17] M. Wiese, R. Knobloch, R. Korn, and P. Kretschmer. "Quant GANs: Deep Generation of Financial Time Series". 2019. arXiv:1907.06673v2.

[18] F. Rundo, F. Trenta, A. Luigi di Stallo and S. Battiato. "Machine Learning for Quantitative Finance Applications: A Survey". Applied Sciences. 2019, 9, 5574. Doi: 10.3390/app9245574.

[19] D. Lv, S. Yuan, M. Li and Y. Xiang. “An Empirical Study of Machine Learning Algorithms for Stock Daily Trading Strategy”. Mathematical Problems in Engineering. Volume 2019, Article ID 7816154, 30 pages. Doi: 10.1155/2019/7816154.

[20] B.M. Henrique, V.A. Sobreiro and H. Kimura. "Literature review: Machine learning techniques applied to financial market prediction". Expert Systems With Applications 124 (2019) 226–251. Doi: 10.1016/j.eswa.2019.01.012.

[21] L. Ryll and S. Seidens. "Evaluating the Performance of Machine Learning Algorithms in Financial Market Forecasting: A Comprehensive Survey". 2019. arXiv:1906.07786v2.

[22] S. Nosratabadi, A. Mosavi, P. Duan, P. Ghamisi, F. Filip, S. S. Band, U. Reuter, J. Gama and A. H. Gandomi. "Data Science in Economics: Comprehensive Review of Advanced Machine Learning and Deep Learning Methods". Mathematics 2020, 8, 1799; Doi: 10.3390/math8101799.

[23] A. Ozbayoglu, M. Gudelek, and O. Sezer. "Deep learning for financial applications: A survey". Applied Soft Computing Journal 93 (2020) 106384. Doi: 10.1016/j.asoc.2020.106384.

[24] Y. Sato. "Model-Free Reinforcement Learning for Financial Portfolios: A Brief Survey". 2019. arXiv:1904.04973v2.

[25] K. Lei, B. Zhang, Y. Li, M. Yang, and Y. Shen. "Time-driven feature-aware jointly deep reinforcement learning for financial signal representation and algorithmic trading". Expert Systems With Applications 140 (2020) 112872. DOI: 10.1016/j.eswa.2019.112872.

[26] A. Mosavi, Y. Faghan, P. Ghamisi, P. Duan, S. F. Ardabili, E. Salwana and S. S. Band. "Comprehensive Review of Deep Reinforcement Learning Methods and Applications in Economics". Mathematics 2020, 8, 1640. DOI: 10.3390/math8101640.

[27] A. Charpentier, R. Elie and C. Remlinger. "Reinforcement Learning in Economics and Finance". 2020. arXiv:2003.10014v1.

[28] E. Benhamou, D. Saltiel, S. Ungari, A. Mukhopadhyay and J. Atif. "AAMDRL: Augmented Asset Management with Deep Reinforcement Learning". 2020. arXiv:2010.08497v1.

[29] S. Takahashi, Y. Chen and K. Tanaka-Ishii. "Modeling financial time-series with generative adversarial networks". Physica A 527 (2019) 121261. DOI: 10.1016/j.physa.2019.121261.

[30] R. Fu, J. Chen, S. Zeng, Y. Zhuang and A. Sudjianto. “Time Series Simulation by Conditional Generative Adversarial Net”. 2019. arXiv:1904.11419.

[31] E. Villarraga. “Generación de series de tiempo financieras sintéticas para “data augmentation” usando Redes Neuronales Generativas Adversarias (GAN)”. Trabajo Final. Universidad Nacional de Colombia. 2021.

[32] J. Schmidhuber. “Generative Adversarial Networks are Special Cases of Artificial Curiosity (1990) and also Closely Related to Predictability Minimization (1991)”. 2020. ArXiv:1906.04493v3.

[33] I. Goodfellow, Y. Bengio, and A. Courville. “Deep Learning”. The MIT Press. 2016.

[34] J. Langr and V. Bok. “GANs in Action”. Manning. 2019.

[35] J. Brownlee. “Generative Adversarial Networks with Python”. Jason Brownlee. 2019.

[36] R. Tsay. "Analysis of financial time series". 3rd ed. Wiley. 2010.

[37] P. Brandimarte. "Handbook in Monte Carlo simulation: applications in financial engineering, risk management, and economics". Wiley. 2014.

[38] G. W. Corder and D. I. Foreman. "Nonparametric statistics: a step-by-step approach". 2nd ed. Wiley. 2014.

[39] H. Tatsat, S. Puri, and B. Lookabaugh. "Machine Learning and Data Science Blueprints for Finance - From Building Trading Strategies to Robo-Advisors Using Python". O’Reilly. 2021.

[40] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath. "Deep Reinforcement Learning - A brief survey". IEEE Signal Processing Magazine. November 2017. DOI: 10.1109/MSP.2017.2743240.

[41] Y. Hilpisch. “Artificial Intelligence in Finance - A Python-Based Guide”. O’Reilly Media, Inc. 2021.

[42] Y. Hilpisch. “Python for Algorithmic Trading - From Idea to Cloud Deployment”. O’Reilly Media, Inc. 2020.

[43] R. A. Brealey, S. C. Myers, and R. C. Merton. "Principles of Corporate Finance". 12 Edition. McGraw-Hill. 2017.

[44] Z. Kakushadze and J. Serur. “151 Trading Strategies”. Palgrave Macmillan. 2018.