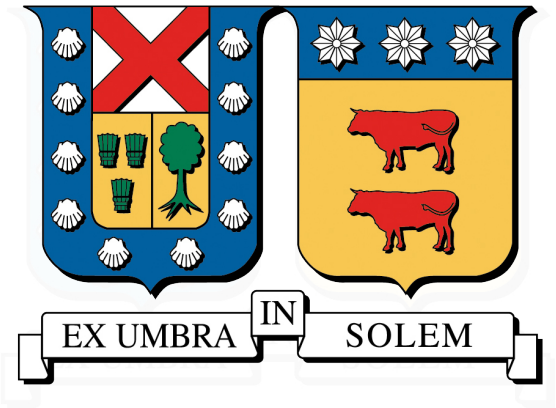
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INDUSTRIAS
VALPARAÍSO - CHILE



DEEP TRANSFORMER Q-LEARNING BASADO EN APRENDIZAJE REFORZADO PARA OPTIMIZACIÓN DE PORTAFOLIO EN CRIPTOMONEDAS

MATIAS ANDRÉS DAVID ZEPEDA VEGA

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL INDUSTRIAL Y MAGISTER EN CIENCIA DE LA INGENIERA

PROFESOR GUÍA	:	Dr. Ing. WERNER KRISTJANPOLLER
PROFESORA CORREFERENTE	:	Dr. Ing. FELIPE ESCUDERO

ABRIL, 2025

CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

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

Título del trabajo: DEEP TRANSFORMER Q-LEARNING BASADO EN APRENDIZAJE REFORZADO PARA OPTIMIZACIÓN DE PORTAFOLIO EN CRIPTOMONEDAS

Nombre del candidato(a): Matias Andres David Zepeda Vega

Carrera / Grado: Magister en Ciencias de la Ingenieria

Campus: Casa Central Valparaiso **;** **Departamento**: Industrias

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Werner Kristjanpoller, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

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

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

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

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

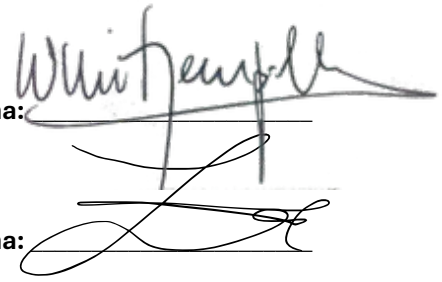
☐ 6 meses; ☐ 12 meses; ☐ 2 años; ☐ 3 años; ☐ 5 años; ☐ 10 años

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

4.- FIRMAS

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

Fecha: 05-08-2025 ; **Firma:**



Estudiante o Candidato(a):

Fecha: 05-08-2025 ; **Firma:**

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

En especial a mi madre, Maria Rosa Vega, por su cariño, atención y apoyo en cada momento. Siendo un ejemplo de superación y fortaleza frente a momentos difíciles.

En especial a mi padre, Raúl Zepeda, quien con su sabiduría y cariño, supo guiarme en mi aprendizaje. Siendo un ejemplo de como se pueden lograr cosas importantes con esfuerzo.

A mi hermano, hermana y mis amigos, quienes se mostraron siempre atentos y disponibles para apoyarnos en momentos complejos.

A mis tíos, primos y mi padrino, Pedro Valenzuela, por haber estado presente en mi formación profesional y personal, siendo un apoyo incondicional. Que al yo ser de otra región, me hicieron sentir como en casa.

Muchas gracias a cada uno de ustedes, por haber participado de una u otra forma con su cariño. A todos ustedes, dedico con gratitud y amor este logro.

Índice de Contenidos

Índice de Tablas

Índice de Figuras

Resumen Ejecutivo

La predicción de un activo financiero para tomar la mejor decisión de inversión que optimice el rendimiento de un portafolio, ha sido ampliamente estudiada en el campo de las finanzas, teniendo un importante avance mediante el uso de nuevas tecnologías como el machine learning, una de las mas importantes en el ultimo tiempo. Ademas, uno de los activos mas en auge y novedosos de los últimos años han sido las criptomonedas, las cuales se mueven en un mercado complejo con mucho dinamismo y varios factores externos que influyen sobre su precio. Por lo que, ambos aspecto han generado que el campo sobre el estudio del uso del *machine learnig* para predecir el comportamiento de las Cryptocurrencies se haga muy relevante y desafiante. Esta investigacion contribuye, proponiendo un *deep Transformer Q learning model (DTQL)* para la optimización de un portafolio compuesto por quince criptomonedas, ademas es el primer paper en comparar para un mismo set de datos los tres modelos de *deep learing* mas usados para predicción en los últimos años: *Transformer, Long Short Term Memory (LSTM)* y *multi-ouput ANN*. El estudio, utiliza los precios históricos de las quince criptomonedas mas liquidas y con mayor capitalización de mercado, para crear y validar los modelos, aplicando para su entrenamiento *reinfocement learing (RL)* enmarcado en un *markov decision process (MDP)*, una de las técnicas del campo de RL ampliamente utilizadas. Esto permite al modelo aprender políticas optimas para la asignación del portafolio, adaptándose mejor que otras técnicas al dinamismo y complejidad del mercado. Los resultados muestran que el modelo de d*eep Transformer Q-learning* se desempeña mejor que los modelos de *deep LSTM Q-learning* y *deep multi-output ANN Q-learning*, logrando mejor respuesta en términos del Indice de Sharpe. La inclusión de un mecanismo de atención y una arquitectura de *encoder-decoder*, permite capturar dependencias de largo plazo junto con información

importante sin depender de una estructura recurrente, logrando generar que el modelo basado en Transformer tenga mejor desempeño. Por lo que, los resultados obtenidos en esta investigacion resaltan el uso de *Transformer* basado en *reinforcement learing* para la optimización de un portafolio compuesto de Criptomonedas, mejorando el retorno del administrador del portafolio mediante una mejor toma de decisiones y control del riesgo. Así, la exploración de mejoras al modelo de *Transformer* en su arquitectura y la inclusión de nuevas variables al *input*, pueden contribuir al desarrollo de modelos mas sofisticados de predicción en el futuro.

1 | Introducción

La Optimización de portafolio [1–3] es un proceso que busca maximizar los retornos mientras se minimiza el riesgo diversificando la colocación de los activos. Dada la enorme cantidad de información a ser procesada, es que su análisis resulta muy complejo, por lo que ha sido un campo ampliamente estudiado, buscando nuevas técnicas que ayuden la toma de decisiones. Las mas tradicionales [4–8] enmarcadas en la optimización según media-varianza, se enfocan en balancear el retorno esperado con el riesgo, normalmente medido por la varianza. El desafió de estas técnicas radica en estimar los retornos y las covarianzas, altamente complejo debido al modelamiento del riesgo y la estimación del error. Una de las técnicas tradicionales mas conocidas es la desarrollada por Markowitz (1952) [9].

Los estudios mas recientes, están enfocados en modelos predictivos basados en *machine learning*. Campo que se ha desarrollado principalmente en la predicción de acciones [10–17], con mas 138 artículos de investigacion sobre técnicas de *machine learning* o aprendizaje automático para la predicción de acciones del mercado de valores al 2022 [18]. Surgiendo a partir de ello nuevas metodológicas como los modelos *deep learning* o aprendizaje profundo [19–21] y análisis de sentimientos [22, 23], los cuales han contribuido importantemente a la mejora en la precisión de las predicciones de los mercados financieros. Entre los modelos de *deep leaning* para la predicción de acciones, destacan lo modelos basados en redes neuronales convolucionales o *Convolutional Neural Networks* (CNNs) [24–26] y Redes neuronales recurrente o *Recurrent Neural Networks* (RNNs) [27], mostrando un rendimiento superior aquellos que combinan ambos modelos. Mientras que otro de los modelos que ha mostrado positivos resultados es el Memoria a Largo-corto plazo o *Long Short-Term Memory* (LSTM) [28–30], utilizado para estimar largas series de tiempo, y

que combinado con otros modelos logra ser bastante eficiente, principalmente al estimar el precio de las acciones. Sin embargo dado que estos modelos sufren de problemas de gradientes que desaparecen y explotan o *vanishing and exploding gradient problems*, se ha desarrollado un modelo que no depende de una estructura recurrente, los Transformers [31], los cuales han venido a revolucionar las técnicas de predicción, siendo de gran interés su estudio e implementación en los últimos años [32, 33]. En complemento, otro de los campos que ha estado siendo ampliamente utilizado ha sido el Aprendizaje Reforzado o *Reinforcement Learning* (RL) [34, 35], una técnica que logra adaptarse a los cambios dinámicos de los mercados financieros. Entre sus algoritmos esta el *Q-learning*, el cual mediante la prueba y error, va explorando soluciones optimas que maximicen la función de recompensa o *Rewards*.

Así, dado los avances en los campos del *machine learing, deep learning y Reinforcement Learning*, es que se han desarrollado técnicas avanzadas para la optimización de portafolios [36–38]. Chen et al.(2021) [39] ha sido uno de los pioneros en la investigacion sobre el desarrollo de un modelo de machine learning con una relación de media-varianza para la construcción de un portafolio de acciones. Al aplicar herramientas del deep learning como LSTM o Transformer, se han logrado mejoras en la optimización de portafolios [40, 41] que al ser combinadas con técnicas de Reinforcement Learning como el Q-Learning, los investigadores han logrado desarrollar nuevos modelos llamados deep Reinforcement Learning [42, 43], que han demostrado nuevas oportunidades para mejorar los resultados y las limitaciones presentadas por modelos previos.

Por otro lado, en los últimos años uno de los activos financieros que ha ganado mucha popularidad han sido las Criptomonedas o también llamadas *Cryptocurrencies* en ingles [44, 45]. Un activo financiero que es digital basado en una tecnología de *blockchain* que asegura sus transacciones. Impulsadas por diversos factores como los avances tecnológicos, dinámicas de mercado y el comportamiento de los inversionistas. Lo cual se ha visto reflejado en el incremento de la capitalización de mercado, principalmente del Ethereum y Bitcoin. Solo el Bitcoin ha alcanzado una capitalización de 1.23 USD trillones para el 2023 [46]. Así, según la investigacion realizada por Murugappan et al. (2023) [46], entre los factores que han influenciado este crecimiento han sido la adopción de la especulación y desarrollo

de nuevos algoritmos y modelos mas sofisticados, acompañado de una mayor confianza en el activo debido a mejoras en su tecnología de Blockchain, mas segura y trasparente. Sin embargo, este instrumento sigue enfrentando importantes desafíos relacionados a su regulación, volatilidad de mercado y seguridad. Por lo que ofrece un potencial económico importante con interesantes oportunidades de investigacion, las que han demostrado que mejoras en la predicción de los precios contribuyen importantemente al desarrollo de estrategias con una mejor administración del riesgo, lo cual es crucial para el desarrollo de un mercado con tan alta volatilidad [47].

1.1. Problema de Investigación

El problema de la investigacion desarrollada esta relacionado en mejorar la eficiencia con la cual se asigna la distribución de la inversión de un portafolio compuesto por criptomonedas. Así, hoy en día existen algunos modelos que permiten optimizar este proceso, sin embargo aun se esta lejos de una eficiencia absoluta, por lo que investigar nuevos modelos que mejoren el proceso resulta de suma importancia en un mercado con alto nivel de dinamismo.

1.2. Objetivos

1.2.1. *Objetivo General*

El objetivo general de la investigacion es presentar un nuevo modelo que optimice la asignación de inversión de un portafolio compuesto por Criptomonedas. Asi, esto busca mejorar la eficiencia con la cual se realiza este proceso actualmente, siendo de gran relevancia en un mercado que se caracteriza por una gran volatilidad y dinamismo, el cual ademas se ha visto ampliamente masificado afectando su comportamiento cada vez a un mayor numero de personas en el mundo.

1.2.2. Objetivos Específicos

- Investigar sobre la eficiencia de modelos de redes neuronales como el Transformer, para la optimización de portafolio
- Comparar distintos tipos de modelos del campo del machine learning para la optimización de portafolio compuesto por criptomonedas.
- Investigar sobre la eficiencia del uso del Reinforcement Learning para la optimización de portafolios de criptomonedas.
- Crear y poner a prueba un nuevo modelo que combine las técnicas del machine learning, por un lado la utilización del Reinforcement Learning para el proceso de aprendizaje y toma de decisiones de inversión, mientras que por otro lado el uso de Transformer para mejorar la eficiencia del proceso de aprendizaje.

1.3. Contribución de la investigación

En el campo del deep Reinforcement Learning, las Cryptocurrencies también han visto su espacio con grandes avances [48–53]. Por lo que esta investigacion contribuye a este campo con los siguientes aspectos:

- Metodología: Hemos introducido el uso de un nuevo modelo, deep Transformer Q-learning basado en Reinforcement Learning, con el cual se ha configurado un agente capaz de elegir entre comprar, vender o mantener, cada uno de los activos, entregando una colocación de ellos. Así, este agente es capaz de evaluar el desempeño pasado de los activos, para generar una proyección futura que le permita tomar una decisión respecto al activo, de tal manera de optimizar el desempeño total del portafolio. Repitiendo este proceso en cada re-balanceo del portafolio, permitiéndole aprender de sus decisiones pasadas, para optimizar sus decisiones futuras y así entregarle mayor dinamismo a la construcción del portafolio.
- Teoría: Utilizamos Transformer en la aproximación del Q-valor, con la idea de capturar dependencias de largo plazo a través de un mecanismo de atención y una

arquitectura de encoder-decoder. Ademas incorporamos este mecanismo a un sistema de entrenamiento basado en Reinforcement Learning, permitiendo que el modelo pueda ir aprendiendo enmarcado en un Markov Decision Process (MDP). Mostramos que el uso de Transformer con RL, logra mejores resultados en la optimización de portafolio, con un menor tiempo de ejecución.

- Experimentos: Para un portafolio que tiene un set de datos que incluye las 15 Cryptocurrencies con mayor capitalización de mercado. Realizamos 4 experimentos para evaluar el desempeño del State-Of-The-Art de tres modelos altamente utilizados en los últimos años: multi-output ANN, LSTM y Transformer. Aplicados todos sobre una base de Reinforcement Learning y MDP. Hemos contribuido con la realización de una investigacion donde por primera vez, se compara un modelo Transformer con otros modelos de deep RL para un mismo set de datos y considerando un periodo de crisis económica, generada por pandemia COVID-19.

2 | Marco Teórico

2.1. Definición de portafolio financiero

Un portafolio financiero (o cartera de inversión) es un conjunto de activos financieros, como acciones, bonos, fondos mutuos y otros instrumentos, que un inversionista posee. El objetivo principal de construir un portafolio es diversificar el riesgo, es decir, no colocar todo el capital en un solo activo. La selección de estos activos se realiza con el fin de maximizar el rendimiento esperado para un nivel de riesgo determinado o, de manera equivalente, minimizar el riesgo para un rendimiento esperado. [9]

2.2. Optimización de portafolios

La optimización de portafolio es un recurso utilizado ampliamente en el mundo financiero, este consta de un proceso en el cual se gestionan las inversiones buscando maximizar el rendimiento para un nivel de riesgo determinado, o bien minimizar el riesgo dado un nivel de rendimiento esperado. Así se va equilibrando el riesgo con el retorno del portafolio mediante la búsqueda de una estrategia de inversión que combine distintos activos como bonos, acciones y otros instrumentos financieros, de tal manera de llegar al nivel optimo deseado.

En el proceso esto implica un alto grado de análisis, el cual es apoyado con modelos matemáticos y herramientas computacionales que permitan determinar el valor optimo del portafolio, junto con la configuración de activos necesaria. Esto determinado por un análisis histórico de los activos sobre su rendimiento, volatilidad y correlación.

2.2.1. Teoría de portafolios

La teoría de portafolios o teoría de portafolio moderno *(Modern Portfolio Theory, MPT)*, es un concepto desarrollado por Harry Markowitz en 1952 a través de su investigacion *"Portfolio Selection"* [9]. En ella establece un marco conceptual que busca definir una metodología de construcción de portafolios de inversión, ofreciendo un mayor rendimiento esperado dado un nivel de riesgo, o bien obtener un menor nivel de riesgo dado un nivel de rendimiento esperado. Esta teoría se basa en la premisa de que los inversionistas son racionales y aversos al riesgo, utilizando la diversificación como herramienta principal para gestionar la relación entre riesgo y retorno. Para lograr un portafolio óptimo, la teoría desarrollada por Markowitz considera tres variables clave: el retorno esperado de cada activo, su volatilidad (medida por la desviación estándar) y la correlación entre los activos del portafolio. Así, la teoría postula que el riesgo del portafolio no va a ser solo la suma de los riesgos de sus activos individuales, sino que también dependen de la correlación entre ellos, por lo que al combinar activos que no se mueven de manera sincronizada, es posible reducir la volatilidad y riesgo general del portafolio, logrando una mayor eficiencia en la administración del riesgo.

En la practica, la teoría de portafolio va a implicar el uso de modelos matemáticos y análisis sofisticados para estudiar una amplia gama da activos, sus retornos, volatilidad y correlación. Determinando la asignación de capital de cada activo que mejor se alinee con los objetivos del inversionista. El resultado del análisis es un portafolio .^{ef}iciente", que se va a situar en la llamada "Frontera eficiente", la cual es explicada por Markowitz como el conjunto de portafolios óptimos dado un retorno y riesgo.

2.2.2. Importancia de la optimización de las asignaciones de capital de un portafolio

El proceso de optimización de un portafolio requiere una asignación de capital optima dado un objetivo determinado. Esto va a ser de gran importancia para los inversionistas puesto que permite que puedan tomar decisiones de inversión basadas en una metodología sistemática y racional, en lugar de basarse en la intuición sin considerar al portafolio como

un conjunto. Así el inversionista va a ser capaz de maximizar el retorno, minimizando el riesgo, siendo un punto crucial ya que ayuda a los inversores a encontrar el punto de equilibrio que se ajuste a su tolerancia al riesgo dada una meta financiera. Ademas, el proceso de optimización tiene una importancia desde el punto de la diversificación efectiva, ya que al analizar la correlación de los activos permite reducir la volatilidad general de la cartera, minimizando el riesgo no sistemático.

Complementariamente, el proceso de optimización se basa en datos con modelos matemáticos y estadísticas, lo que permite la toma de decisiones de inversión mas informadas, dejando de lado las emociones y sesgos que normalmente afectan las decisión de inversión.

De esta forma es que el proceso de optimización va a ayudar al administrador del portafolio a adaptarse de mejor forma a los objetivos del inversor, personalizando para cada tipo de inversión un tipo de cartera, la cual puede ser conservadora, moderada o mas agresiva, considerando distintos horizontes de inversión. Lo cual se va a ver acompañado de una gestión proactiva que permite la metodología, adaptando la composición del portafolio a medida que van cambiando las condiciones del mercado o los objetivos personales del inversor, asegurando la eficiencia de la cartera a lo largo del tiempo.

2.3. Criptomonedas

2.3.1. Definición de las criptomonedas

Según el articulo publicado por Brookings Institution, una de las instituciones de investigación de políticas públicas más influyentes, una Criptomoneda (o Cryptocurrency en ingles) esta definida como un activo digital o virtual que utiliza la criptografía para asegurar las transacciones y controlar la creación de nuevas unidades. Estas monedas operan en una red descentralizada, lo que significa que no están sujetas al control de una autoridad central, como un gobierno o un banco. La mayoría de las criptomonedas se basan en la tecnología de cadena de bloques (blockchain), que es un libro de contabilidad distribuido y público donde se registran todas las transacciones. [54]

2.3.2. Como funcionan las criptomonedas

Las criptomonedas basan su funcionamiento en una tecnología llamada çadena de bloques"(o blockchain en ingles), la cual actúa como un libro contable digital que se encuentra disponible públicamente, el cual va registrando todas las transferencias de forma segura y cronológicamente. Este proceso a diferencia de las monedas tradicionales que son administradas por un banco centralizado, va a estar compuesto por una red distribuida (bloques) en miles de computadoras en distintas partes del mundo que actúan como nodos, los que hacen que sea un proceso muy resistente a la censura y manipulación. De esta forma el nombre de cadena de bloques hace referencia a que cada nueva transacción va a estar agrupada en un bloque enlazado de forma criptográfica al bloque anterior, creando una cadena inmutable de datos y registros consecutivos, ya que cada transacción va encima de la anterior, siendo imborrable e identificable el registro.

El proceso de transacciones y almacenamiento de las criptomonedas esta compuesto de diferentes partes y complejidades, pero en resumen, el sistema depende de una criptografía que esta asociada a claves privadas y publicas que hacen posible una transacción. La que a su vez utiliza mecanismos de consensos y validación que son llevados a cabo por "mineros"que validan las transacciones de manera segura y descentralizada, de tal manera de garantizar la integridad de la red sin necesidad de intermediarios.

2.3.3. Importancia de las criptomonedas

La importancia de las criptomonedas en el mundo radica en que son un tipo de moneda que no se encuentra centralizada, lo cual ofrece una alternativa a sistemas financiero tradicionales que normalmente tiene un estricto control por gobiernos y bancos centrales. Ademas, esta tecnología al estar gestionada por una red de computadoras, elimina la necesidad de instituciones intermediarias, reduciendo costos y aumentando la velocidad de transacciones, sobre todo a nivel internacional.

Complementariamente, este tipo de herramienta ofrece acceso a servicios financieros a personas no bancarizadas o con acceso limitado, por lo que democratiza el acceso a financiamiento y otras formas de inversión, logrando que cualquier persona con acceso a

internet, pueda enviar, recibir y almacenar un activo financiero con valor económico.

Otro de los usos que ha tenido últimamente las criptomonedas y que han aumentado su relevancia, es que se han transformado en un activo de refugio frente a la inflación, permitiéndole a los accionistas diversificar su portafolio y disminuir su riesgo, ya que hay criptomonedas no ligadas a una moneda fiduciaria especifica.

Las principales criptomonedas que existen hoy en día y que son las que han adquirido mayor popularidad dada su capitalización de mercado y su impacto tecnológico: el Bitcoin (BTC), Ethereum (ETH), Tether (USDT) y Binance Coin (BNB). Así el BTC creada en 2009 es la que tiene mayor capitalización de mercado, siendo una de las mas utilizadas llegando a tener incluso la denominación de .ºro digital", limitando su oferta a 21 millones de unidades. Mientras que la segunda criptomoneda con mayor capitalización de mercado es el ETH, la cual se diferencia por ser más que solo una moneda, ya que también es una plataforma de software descentralizada que permite la creación de contratos inteligentes y aplicaciones, realizando todas las transacciones en su red propia. Por otro lado esta la Thether o USDT, la cual se caracteriza por ser la stablecoin más grande del mercado, y que se encuentra asociada al dólar estadounidense. Su estabilidad hace que sea un instrumento financiero ideal para transacciones que resguarden su valor en un ecosistema de criptos. Por ultimo, otra de las mas populares es la Binance Coin, la cual es una moneda nativa de la plataforma Binance, utilizada para pagar comisiones de trading y ser la base de una red desarrollada como BNB Chain.

2.4. Reinforcement Learning

2.4.1. Definición del Reinforcement Learning

El Aprendizaje por Refuerzo o Reinforcement Learning (RL) en ingles, es una de las áreas del machine learning en la cual el aprendizaje es automático, por lo que existe un agente que va tomando decisiones optimas en un entorno para maximizar una recompensa acumulada a lo largo de las iteraciones, las que normalmente representan el tiempo transcurrido. A diferencia de otros tipos de aprendizaje, el RL no se basa en datos supervisados

o identificados previamente, sino que el agente va aprendiendo a través de la experiencia directa e iterativa respecto a cuales son las mejores decisiones dado un entorno y una recompensa por sus acciones. Así, este proceso va a estar compuesto por un Agente, Entorno, Estados, Acciones y una Recompensa. Esto con el objetivo de determinar una "política"que indique que acciones tomar en cada estado, de tal manera de obtener la mayor recompensa, por lo que esta política va a ser aprendida mediante la prueba y error, explorando diferentes acciones y evaluando sus resultados a través de la recompensa obtenida a partir de ellas, logrando al final del periodo una política optima que maximice la recompensa.

2.4.2. Agente

En RL el agente es uno de los componentes mas importantes, puesto que es el algoritmo que va determinando que acciones tomar dado un entorno y estado. Así en el contexto financiero, este agente es la representación de lo que seria un administrador de portafolio o inversionista, siendo el encargado de tomar decisiones inteligentes y autónomas, de tal manera de lograr un objetivo especifico. El que a su vez se diferencia de simples programas que necesitan de etiquetas en los datos o retroalimentación humana, en que puede ir aprendiendo autónomamente a partir de la interacción que va teniendo con el entorno.

El agente va a tener la función de identificar el entorno observando su estado actual, para luego tomar una decisión basado en su conocimiento y experiencia respecto al estado. Acción que a su vez va alineada con su política, definida como la estrategia que el agente ha ido aprendiendo a través del tiempo hasta ese momento. Finalmente el agente tiene que evaluar la reacción del entorno dada la acción tomada, sobre la cual hay una recompensa que puede ser positiva o negativa, siendo una señal de aprendizaje para el agente, puesto que basado en ella va a aprender si la acción tomada fue correcta o incorrecta. De esta forma, luego de miles de iteraciones el agente va afinando su política, con el objetivo final de aprender una "política optima"que le permita maximizar la recompensa en el largo plazo. Por lo que el agente va a ser el experto financiero que con su experiencia va adquiriendo la habilidad de tomar las mejores decisiones a través de la practica y la retroalimentación.

2.4.3. Entorno del Reinforcement Learning

En el campo del RL el entorno es el universo de estados con que el agente interactúa, siendo todo lo que existe ademas del agente, por lo que es todo lo que el agente puede interactuar, percibir y efectuar a través de las acciones. El entorno tiene la opción de ser real o simulado.

Dado que el entorno es una representación de la realidad, este no es estático, sino que va cambiando y reaccionando conforme a las acciones tomadas por el agente. Así el entorno va a tener un conjunto de estados, que van describiendo la situación actual del entorno. A su vez a medida que el agente va interactuando con el entorno, van existiendo transiciones de estado, los que a su vez definen una recompensa numérica, indicándole al agente que tan buena o mala fue su decisión.

Un ejemplo realista se da en el juego de ajedrez, en donde el entorno es el tablero y todas las piezas, siendo el estado la disposición actual de las piezas. En el ámbito financiero el entorno son los datos históricos y estadísticos de los activos, en donde a medida que va pasando el tiempo este va cambiando.

2.4.4. Estados

Los estados van a representar las diferentes situaciones en la cual se configura el entorno. Por lo que el agente va a utilizar la información proveniente de los estados para tomar una decisión respecto a la acción. De esta forma la efectividad con que el agente tome buenas decisiones va a estar condicionada con la calidad con que se representan los estados y si estos realmente definen bien la situación actual del entorno. Así un buen estado va a contener toda la información necesaria para que el agente tome la mejor decisión, la que a su vez es percibida por el agente a través de sensores o algoritmos. Si el estado queda incompletamente definido, entonces el agente podría tomar decisiones erróneas, mientras que si el estado se define muy complejo puede significar que el aprendizaje sea computacionalmente inviable.

Así, en cada ciclo que el agente se enfrente a un estado especifico, tiene que elegir una acción, por lo que los estados dentro de un árbol de decisiones corresponden a lo diferentes

"nudos.º çaminos.ª elegir por el agente, por lo que a medida que el agente avanza por estos estados va aprendiendo sobre la mejor decisión que construya su estrategia o política mas eficiente.

Para el caso del contexto financiero, los diferentes estados corresponden a la situación de los distintos activos dado un tiempo t. Esta situación puede ser representada mediante su retorno, volatilidad, correlación, precio, volumen, etc.

2.4.5. Espacio de acciones

En el RL las acciones corresponden al conjunto de decisiones que el agente puede tomar en un estado particular de un entorno. Por lo que cada acción ejecutada por el agente va a cambiar el estado del entorno, generando una recompensa positiva o negativa.

El conjunto predefinido de acciones puede ser discreto como los movimientos de un robot (arriba, abajo, izquierda o derecha), o bien puede ser continuo en un numero infinito de opciones como la cantidad de capital a invertir dentro de un portafolio, limitado por cierto rango. Las consecuencias de las acciones pueden tener un efecto inmediato o bien al mediano o largo plazo, por lo que la "memoria"del agente va a ser parte importante de su aprendizaje, logrando este por medio de la exploración probar diferentes estados comparando las recompensas resultantes, aprendiendo el agente a asociar las acciones que permitan los mejores estados para una política en base a la maximización de la recompensa.

Por tanto, el éxito del proceso de aprendizaje va a estar dado por la capacidad que tenga el agente de seleccionar la mejor acción dado un estado determinado.

2.4.6. Función de recompensa

La función de recompensa o *Reward* en ingles, corresponde a la señal numérica que recibe el agente después de haber interactuado con el entorno mediante una acción especifica. De esta forma el agente va aprendiendo que tan mala o buena fue su acción, obteniendo así una retroalimentación respecto a la calidad de sus acciones y el como configurar mejor su política o estrategia.

El objetivo del agente va a ser maximizar la recompensa acumulado a lo largo del

tiempo, por lo que el diseño de la función de recompensa tiene que ir alineado con el comportamiento que se desea que el agente aprenda. Para el caso del ajedrez, si se requiere que el agente gane una partida, la recompensa final va a ser la victoria, que puede ser representada con un 1 si gana, 0 si empata y -1 si pierde. Asimismo, la recompensa puede ser inmediata, para cada una de las jugadas, ganando 1 punto cada vez que se ejecuta un movimiento favorable, o bien puede ser una recompensa tardía como ganar la partida.

El desafió en el diseño de la función de recompensa radica en que el agente pueda prestar atención tanto a recompensas inmediatas como tardías, logrando a partir de sus acciones del corto plazo lograr una recompensa acumulada mayor al largo plazo.

2.4.7. Estructura Markov decision process (MDP)

Una estructura de Proceso de Decisión de Markov o *Markov Decision Process (MDP) framework* en ingles, es un marco matemático utilizado para modelar la toma de decisiones en donde los resultados son a la misma vez controlados y aleatorios. Así en el área del RL, el MDP viene siendo la base teórica que el agente esta tratando de resolver en cada una de sus acciones, es decir el agente se ve enfrentado a un estado y debe tomar una decisión que no sabe cual sera su resultado pero que tiene ciertas probabilidades de obtener una recompensa positiva frente a determinadas acciones.

Este proceso se va a ver marcado por una función de transición de probabilidad, en donde dado un entorno, un estado actual s, un estado futuro siguiente s' y una acción a, se va a definir una probabilidad $P[s'|a, s]$, es decir la probabilidad que el agente pase al estado s' dada la acción a realizada en el estado s. Con lo cual el agente va a obtener una recompensa definida como $R[s, a, s']$, la cual indica el valor de una recompensa obtenida a partir de una acción a que se produjo en el estado s y que genero el estado s'

Así, la principal característica del MDP es su propiedad de Markov, la cual establece esta probabilidad de transición a un estado futuro s', dependiendo unicamente de la acción a tomada en el estado s, no siendo influenciada por la secuencia de estados o acciones que llevaron a aquel estado actual s. Explicado de forma mas simple, que el futuro sea independiente del pasado y que solo afecte en la decisión el presente.

De esta forma es que el agente va a estar inmerso en u estado que es modelado como un

MDP, aprendiendo la mejor política o estrategia que maximice la recompensa total esperada a largo plazo, sin conocer de antemano las funciones de transición y recompensa.

Un ejemplo de ilustración sencillo es representado mediante un jugador de ajedrez (agente), el cual durante su vida estuvo estudiando diferentes jugadas y partidas (fase de aprendizaje). Todo aquel tiempo le sirvió para aprender sobre los mejores movimientos y cuales le traen buenos y malos resultados. Sin embargo, al momento de verse enfrentado a una jugada en medio de una partida, el jugador de ajedrez no necesita recordar todos los movimientos y partidas jugadas con anterioridad, sino que basa su decisión mirando la posición actual del tablero (estado), acompañándola de su intuición y conocimiento (política aprendida), sabe que decisión tomar para tener una mayor probabilidad de ganar el juego. Así la información del tablero le va a proporcionar al jugador la información necesaria para tomar la siguiente jugada, sin necesidad de ver todas las jugadas pasadas relacionadas a la partida.

2.4.8. Ecuación de Bellman y el Q-valor

La Ecuación de Bellman es una relación matemática fundamental en la programación dinámica y el Aprendizaje por Refuerzo. No es un algoritmo, sino un principio de optimización que establece una relación de recurrencia para la función de valor de un estado. Su propósito es descomponer un problema complejo de toma de decisiones a largo plazo en pasos más pequeños y manejables. [55]

De esta forma la ecuación nos va a indicar que el valor optimo de un estado va a ser igual a la recompensa inmediata mas el valor esperado de la mejor recompensa futura dado un estado siguiente. Idea que se ve expresada en la siguiente formula:

$$Q^*(s,a) = \mathbb{E}[r + \gamma \underset{a'}{max}\, Q^*(s',a')] \tag{2.1}$$

La ecuación expresada en su forma de optimización, va a establecer que para encontrar el valor máximo dada una acción en un estado, el agente debe considerar la recompensa instantánea r y el valor de la mejor acción futura a tomar en el estado s', descontado por un factor γ.

Complementariamente el Q-valor o *Q-value* en ingles, es también conocida como la función valor del par acción-estado, la cual corresponde a una medida de la çalidad"(de ahí el uso de la Q por *Quality*) o el valor a largo plazo de tomar una acción especifica en un determinado estado. Por lo que el valor $Q(s, a)$ va a representar la recompensa total futura que el agente espera recibir dado una acción a en el estado s, siguiendo una política optima para el resto de las iteraciones. De esta forma el objetivo del RL es encontrar los valores Q que representen la mejor política posible.

Por tanto la relación entre la Ecuación de Bellman y el Q-valor, es que la primera propone el marco teórico para calcular y actualizar los valores Q, mientras que la segunda va a corresponder a una aplicación de la ecuación siendo un valor que demuestra calidad del par acción-estado, la cual puede ser calculada mediante algoritmos de aprendizaje.

2.4.9. Q-learning

El Q-learning es una implementación practica de la ecuación de Bellman. Corresponde a un algoritmo fundamental del RL, permitiéndole al agente aprender la mejor política de acción en un entorno determinado, sin conocer las reglas del entorno, esto implica que es un algoritmo que no depende de un modelo matemático. El objetivo principal del algoritmo es encontrar la función Q-valor optima que permita al agente tomar las mejores decisiones.

De esta forma el objetivo del Q-learning es lograr determinar el valor Q para cada par posible (estado, acción). Resultados que normalmente son expresados en una tabla, en donde el agente evalúa según los valores de ella cual es la acción que le otorga el mayor Q-valor, que por tanto le va a entregar una mejor recompensa futura.

Este algoritmo funciona bajo un proceso iterativo en donde el agente va a interactuar con el entorno, y en cada uno de los estados que va recorriendo, va actualizando los valores de la tabla Q, esto segun la formula:

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)] \tag{2.2}$$

En donde $Q(s, a)$ corresponde al valor Q actual para el par (estado, acción), el valor α es la tasa de aprendizaje que va a determinar cuánto aprende el agente de la nueva

información, por lo que un valor alto de α hace que el aprendizaje sea más rápido, pero puede ser inestable. r va a ser la recompensa inmediata que el agente recibe al tomar la acción a en el entorno dado. Ademas, el factor de descuento γ, va a ponderar la importancia de la recompensa futura, teniendo un valor entre 0 y 1, un valor cercano a 0 hace que el agente priorice decisiones con recompensas inmediatas, mientras que un valor cercano a 1 va a priorizar decisiones que tengan una mayor relevancia al largo plazo.

2.4.10. Como se relaciona el MDP, Ecuación de Bellman, Q-valor y el Q-learning

Los cuatro conceptos descritos previamente se relacionan en que el MDP va a definir la estructura teórica del problema que se enfrenta el agente en el RL, mientras que la ecuación de Bellman va a ser la herramienta matemática que permita resolver la problemática planteada por el MDP y el Q-learning va a ser el algoritmo practico sobre el cual se resuelve el problema implementando la herramienta matemática sobre el valor Q, representante de los pares estado-acción.

Al describir esto en mayor detalle, el MDP define la estructura del RL, formalizando los elementos: estados, acciones, transiciones de probabilidad y recompensas. Representando de esta forma el universo sobre el cual el agente toma decisiones, sin conocer las reglas (transiciones y recompensa). Por lo que el resolver el problema del MDP va a permitir al agente encontrar aquella política optima que maximiza las recompensas a largo plazo.

Complementariamente, la ecuación de Bellman va a ser la solución teórica sobre la cual se plantea resolver el problema del MDP, que al ser contextualizada en el RL, esta ecuación va a ser resuelta para los Q-valores representantes de los pares estado-acción, permitiendo mediante la ecuación resolver la solución optima de estos valores.

Finalmente el Q-learning permite resolver de forma practica la ecuación de Bellman expresada para los Q-valores, resolviendo y aprendiendo de manera iterativa los valores de Q. Proceso que resulta fundamental ya que como el agente no conoce las probabilidades de transición ni las recompensas (el modelo del entorno), no es posible calcular la esperanza $\mathbb{E}$ directamente de la ecuación de Bellman, siendo fundamenta la utilización del algoritmo

Q-learning para ir actualizando dichos valores basados en la experiencia.

En conclusión, en el área del RL el MDP define el problema a resolver, mientras que la ecuación de Bellman propone el principio teórico que da la relación entre los estados, las acciones y sus recompensas futuras, utilizando el algoritmo del Q-learning para encontrar la solución optima del MDP, a través de un proceso iterativo y de aprendizaje.

2.4.11. Importancia del Reinforcement Learning

El *Reinforcement Learning* se ha transformando en una de las áreas fundamentales del machine learning, esto debido a que a medida que han evolucionado las tecnologías relacionadas al procesamiento de datos y algoritmos que puedan ir absorbiendo un mayor costo computacional, es que esta metodología ha permitido resolver problemas complejos de toma de decisiones, sobre todo en entornos que son dinámicos e inciertos.

La importancia de esta metodología esta en que un agente tenga la capacidad de aprender de forma autónoma, es decir sin supervisión humana directa, solo a través de su experiencia. Lo cual la diferencia de otros tipos de aprendizajes automáticos, ya que estos requieren una gran cantidad de datos identificados, mientras que el RL puede aprender a partir de su misma retroalimentación mediante la prueba y error o recompensa positiva y negativa, siendo muy relevante cuando la solución no es una respuesta obvia, como en la robótica, juegos o en la gestión de recursos limitados. Simulando para esto el proceso de aprendizaje humano y animal, pero potenciado por computadoras que son capaces de realizar este proceso a una gran velocidad.

Dentro de los grandes logros del RL, es que en el campo de los videojuegos ha sido capaz de superar al humano en juegos complejos como el ajedrez (AlphaZero de DeepMind), Go (AlphaGo), Starcraft II y Dota 2. Sin tener un conocimiento explicito sobre las reglas y estrategias de los juegos. Mientras que en el ámbito de la robótica el RL ha sido capaz de aprender tareas como caminar o manipular objetos, lo cual seria muy difícil de programar manualmente. Por otro lado, plataformas como Netflix o del comercio electrónico, utilizan RL para mejorar la personalización de las recomendaciones.

En lo que respecta a las finanzas, el RL es muy valioso, puesto que las decisiones de inversión son secuenciales, inciertas y dinámicas, compatibilizando con lo que el RL

es capaz de resolver. Motivo por el cual ha sido una herramienta utilizada para poder optimizar y re-balancear portafolios, ejecutar operaciones de alta frecuencia, gestión de riesgos en entornos dinámicos, fijación de precios sobre los derivados y toma de decisiones sin modelos.

2.5. Deep learning en finanzas

2.5.1. Definición del deep learning

El Deep Learning es una sub-categoría del Machine Learning, la cual utiliza redes neuronales artificiales con múltiples capas. A diferencia de las redes neuronales tradicionales que tienen solo unas pocas capas ocultas, los modelos de Deep Learning cuentan con decenas o incluso cientos de estas capas. Cada capa se especializa en extraer y procesar características de los datos de entrada, permitiendo que el modelo aprenda representaciones complejas y abstractas de manera automática, sin necesidad de que un humano las programe explícitamente. [56]

Esta herramienta a sido disruptiva en el sector financiero, ya que ha permitido analizar grandes volúmenes de datos complejos y no estructurados, superando a metodologías tradicionales. Esta capacidad de las redes neuronales profundas, ha permitido que se puedan identificar patrones y relaciones no lineales complejas, habiendo un gran numero de posibilidades y aplicaciones. Dentro de las mas conocidas están las utilizadas en algoritmos de trading, donde los modelos de deep learning son utilizados para predecir los movimientos de los activos y el mercado, logrando identificar patrones complejos que involucran diversas variables como sentimientos de redes sociales, noticias, datos alternativos, que son demasiado sutiles para ser detectados por humanos. Otro de los usos en finanzas es en la detección de fraude, logrando identificar cuando los movimientos son "normales.º bien parte de una estafa. Ademas también han estado siendo utilizados en la evaluación del riesgo y puntuaciones crediticias, permitiendo predecir la solvencia de una persona o empresa, siendo un proceso mas preciso y automatizado. En resumen, esta capacidad del Deep Learning de poder analizar y manejar grandes cantidades de datos complejos, lo han

convertido en un motor clave para la automatización de procesos y toma de decisiones que están en un entorno cada vez mas dinámico.

2.5.2. Redes neuronales artificiales (ANN)

Las redes neuronales artificiales (ANN), son modelos computacionales que se inspiran en el cerebro, su estructura y funcionamiento. Su objetivo es poder identificar patrones a partir de los datos. [56]

La estructura básica de una red neuronal esta compuesta por una capa de entrada, que es la que recibe los datos iniciales a ser procesados por la red. Luego le siguen capas ocultas, las cuales son capas intermedias donde se procesa y extrae la mayor parte de la información. A partir de esta configuración, las capas van asignando pesos y sesgos que son ajustados durante su entrenamiento, con la idea que puedan irse adaptando a los patrones de los datos, aprendiendo su modelado. Lo que diferencia al deep learning de una red neuronal tradicional, es que tiene muchas capas ocultas. Finalmente tiene una o múltiples capas de salida que producen el resultado final del modelo.

Su funcionamiento se basa en ajustes iterativos de los pesos de las conexiones entre neuronas, procesos iterativo realizado a partir del entrenamiento, el cual busca minimizar el error entre la salida de la capa y el resultado deseado. Permitiéndole al modelo minimizar el error entre la salida y el resultado buscado.

2.5.3. Long-Short Term Memory (LSTM)

La red de Memora a Largo-Corto Plazo o mas bien conocida como *Long-Short Term Memory (LSTM)* en ingles, es una red neuronal recurrente o *Recurrent Neural Network (RNN)*, la cual fue diseñada para procesar datos secuenciales. Las redes LSTM a diferencia de las RNN, poseen una capacidad para aprender y retener información a largo plazo, resolviendo el problema del "gradiente evanescente"que sufren las RNN y que genera que a medida que avanzan las iteraciones, las RNN va perdiendo información y por tanto no son capaces de capturar relaciones de largo plazo.

Así, una red LSTM esta hecha para almacenar información relevante durante periodos

prolongados de tiempo, utilizando "puertas"que controlan el flujo de la información, permitiéndole recordar"datos relevantes dentro de una secuencia, un aspecto relevantes en áreas como la traducción, reconocimiento de voz, series de tiempo, etc.

Los elementos claves de una celda LSTM son la Puerta de Olvido (o Forget Gate en ingles), Puerta de Entrada (Input Gate) y Puerta de Salida (Output Gate). Están van a regular que información es transmitida y cual olvidada.

La Puerta de Olvido (2) va a ser la encargada de decidir que datos son descartados, utilizando una función sigmoide con valores entre 0 y 1, entregando un valor de 0 para aquellos datos a ser olvidados y un valor 1 para los que deben mantenerse.

La puerta de entrada (1) va ser la que selecciona la información que se utilizara en el estado de la celda actual (x_t), utilizando una función sigmoide con valores de 0 y 1 para este proceso. Complementariamente utiliza una función tangente hiperbólica (tanh) para generar un vector con valores candidatos a ser agregados al estado de la celda.

Finalmente la puerta de salida (3) controla el resultado e información a ser generada por la celda LSTM. Esta va aplicar una función sigmoide con valores de 0 y 1, al estado oculto anterior y la entrada actual, y luego multiplica el resultado por una versión de la celda de estado pasada por una función tanh. El resultado es la nueva salida de la celda y el nuevo estado oculto.

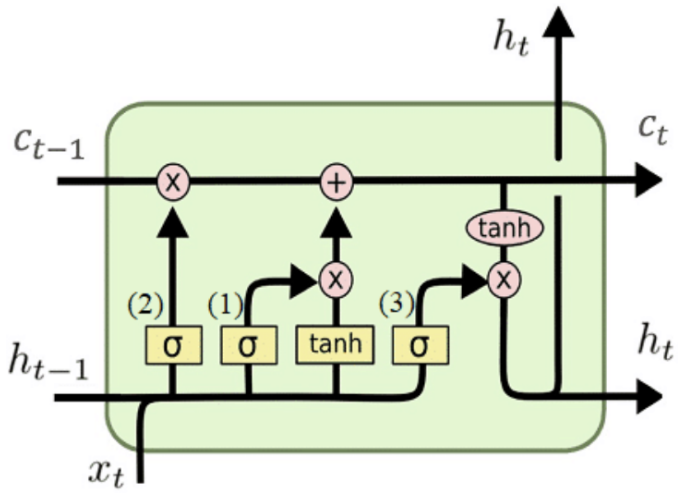


Figura 2.1: Arquitectura celda LSTM
Fuente: Memoria a largo plazo a corto plazo (LSTM): ¿Qué es?. Data Scientest

Así, h_t es el estado oculto de las RNN, pero que en celdas LSTM se le incorpora un

segundo estado denominado c_t, el cual va a ser el estado de memoria. Por lo que h_t es la representación de la memoria corto plazo, mientras que c_t es la memoria a largo plazo que pasa por la puerta de olvido. Ademas, x_t va a representar a la entrada actual de datos.

De esta forma, es que la red LSTM va a utilizar estas tres puertas para manejar eficazmente las dependencias de corto y largo plazo, manteniendo el flujo constante de la información durante la retropropagación del gradiente a través del tiempo, evitando la problemática de la gradiente dada en las RNN comunes, permitiendo que la red aprenda de manera más estable.

2.5.4. Transformers

Los Transformer fueron introducidos por primera vez en el 2017 con el articulo *"Attention Is All You Need"* [31], marcando un precedente en el campo del *deep learning*, puesto que fue capaz de superar a las RNN y LSTM, quienes habían dominado ampliamente hasta ese momento.

Los Transformer son un tipo de red neuronal que trabaja con datos secuenciales pero no los procesa de manera secuencial. Es decir, estos modelos tienen la capacidad de analizar todos los datos dentro de la secuencia al mismo tiempo, siendo clave para esto su mecanismo de auto-atención (o *self-attention* en ingles).

Los Transformer tienen una arquitectura dividida en dos fases, una *encoder* y otra *decoder*, contando como principales elementos: los tokens, codificación posicional y un mecanismo de atención.

De esta forma el Transformer va a comenzar con su encoder para procesar una secuencia de entrada, generando un contexto de ella. Por ejemplo, cuando se traducen textos, el encoder va a leer el texto en el idioma original. Luego el modelo va de descomponer los datos secuenciales en tokens, transformando la data en vectores numéricos posicionales (*encoding process* en ingles), de tal manera que la secuencia pueda ser descompuesta con valores de posición codificados para saber el orden de los datos.

A partir de lo anterior, las datos van a pasar por un mecanismo de atención que va a analizar las relaciones entre los diferentes tokens o datos secuenciales, creando una representación de la secuencia, la cual es procesada por capas de redes neuronales,

logrando un resultado de salida que se traduce en los datos de salida del Transformer. Este proceso es realizado varias veces en paralelo, lo cual se denomina como mecanismo de atención múltiple (o *Multi-Head Attention* en ingles), en donde cada uno de ellos aprende a enfocarse en diferentes aspectos de los datos, por ejemplo uno puede dedicarse a las relaciones sintácticas, mientras que la otra a las semánticas. Finalmente los resultados de estos mecanismos se juntan y forman una representación mas robusta.

Por ultimo la representación de salida generada por el *encoder*, va a ser utilizada por un *decoder* el cual va a generar una secuencia de salida, dato por dato, utilizando también mecanismos de atención para identificar y enfocarse en lo relevante de los datos de salida del encoder.

Los Transformers se han transformado en una herramienta importante ya que las redes recurrentes tradicionales solo procesaban la información de manera lineal, mientras que los Transformer lo hacen en paralelo, acelerando en gran medida el entrenamiento y por tanto reduciendo significativamente los costos computacionales, lo que permitiría a su vez utilizar set de datos mas extensos y complejos. Ejemplos de su uso son los modelos de lenguaje (LLM) como ChatGPT o BERT.

2.5.5. Importancia de los modelos predictivos del deep learning en criptomonedas

Dada las capacidades que tiene el deep learning para analizar una gran cantidad de datos, es que ha adquirido una gran relevancia en el mundo de las criptomonedas, logrando extraer patrones complejos a partir de datos sofisticados, algo que seria casi imposible a ser realizado por humanos. Sobre todo en el área de las criptomonedas, las que se caracterizan por su complejidad de los datos debido a una alta volatilidad, descentralización y dinamismo, siendo un campo ideal para aplicar esta sofisticadas técnicas modernas.

Asi los principales usos que se les han dado a los modelos de deep learning han sido en la predicción del precio y series de tiempo de las criptos, el análisis sobre el sentimiento del mercado, detección de fraudes y seguridad, y la automatización en el trading.

3 | Trabajos Relacionados

En el ultimo tiempo, ha crecido el interés por desarrollar nuevas técnicas y metodologías centradas en el uso de machine learning, tanto para la predicción de los activos financieros como para la optimización de los portafolios. Ellas han demostrado adaptarse mejor que técnicas mas tradicionales, capturando el dinamismo y complejidad de los mercados. Así, unas de las técnicas utilizadas han sido las de deep learning, siendo el uso de Transformers, una de las ultimas novedades y que ha demostrado mejores resultados que las otras, ya que es capaz de capturar dependencias de largo plazo sin depender de una estructura recurrente, por lo que evita que se pierda información y patrones a medida que se avanza en el entrenamiento. Una de las investigaciones es la realizada por C. Wang et al. [57] la cual a diferencia de los modelos que se habían estado utilizando previamente, enfocados en convolutional neural networks y recurrent neurnal networks, decide utilizar un modelo Transformer para predecir indices de mercado. Sus resultados mostraron que el modelo de Transformer se desempeño significativamente mejor que los otros modelos como RNN, CNN y LSTM.

Otra de las técnicas que ha demostrado ser prometedora en los últimos años, es el uso de Reinforcement Learning (RL), la cual mediante un proceso de entrenamiento de prueba y error, aprende a tomar decisiones que la llevan a un premio o una penalización, dado un entorno determinado. Lo interesante de esta técnica es que le permite aprender al modelo de sus mismos errores, corrigiendo sus predicciones y mejorando sus resultados a lo largo del tiempo. Estudios recientes como el realizado por D. O. Oyewolaa et al. [58], el cual propone un modelo de deep Long Short-Term Memory Q-learning y deep Long Short-Term Memory Attention Q-learning, para predecir el precio de las acciones del sector de gas y petroleo. O bien el desarrollado por Y. Baek et al. [59] quien propone un "Novel

Reinforcement learning algorithm" para la predicción de acciones usando una combinación de LSTM con RL. Han demostrando ambos que la combinación del deep learning con el Reinforcement Learning mejora las predicciones de las acciones en relación a modelos tradicionales o solamente modelos de deep learning.

Así, dado los buenos resultados mostrados por los modelos de Trasformer y técnicas como el Reinforcement Learning, es que resulta interesante de investigar una combinación de ambos. El estudio realizado por B. Yang et al. [60], propone un modelo llamado DRL-UTrans que combina el deep reinforcement learning, transformer layers y una arquitectura U-net, para aprender una estrategia de inversión sobre una acción determinada. Sus resultados demuestran que el modelo tiene un mayor retorno comparado con otras estrategias tradicionales y de machine learning como: la Buy and Hold (B&H), PPO, SAC, A2C, GDQN, MLP- windowed y SPMPN.

Al igual que como se han aplicado estas técnicas de deep Reinforcement Learning para predecir precios y desarrollar estrategias de inversión para acciones únicas, también se han desarrollado para portafolios de inversión compuesto por múltiples activos. N. Lee [61], propone un modelo que realiza transacciones automáticas para las acciones de un portafolio y/o que predice el precio futuro ellas, utilizando Reinforcement learning con LSTM y Transformers. Ademas innova al incorporar al Transformer una técnica "Actor-Critic" con regularización (TACR) junto con una red de atención, que le permite entrenar al modelo con la correlación de pasados MDP (Markov decision process) y así que el RL no solo infiera a partir de estados actuales. Sus resultados muestran para diferentes datasets de acciones, el modelo se desempeña mejor que otros state-of-the-art, en términos del Indice de Sharpe y retornos.

Otro de los estudios que muestra la superioridad de estos modelos fue el realizado por V. M. Ngo et al. [62], quien demuestra que para la optimización de portafolios, el modelo de Reinforcement Learning obtiene mejores resultados en términos de Indice de Sharpe, que aquellos modelos tradicionales y de deep learning. Para esto, realiza dos experimentos utilizando datos de acciones de los mercados Vietnamitas y de Estados Unidos, poniendo a prueba los modelos en tiempo de pandemia por COVID 19 y tiempos post-pandemia, en los cuales el modelo de RL obtiene mejores resultados que sus pares, confirmando la

habilidad de este tipo de modelos para adaptarse y responder mejor al dinamismo en las condiciones de mercado.

Dada la relevancia que ha tomado el Reinforcement Learning (RL), su uso también se ha expandido hacia el mercado de las Cryptocurrencies. La investigacion realizada por Z. Jiang [63] propone un modelo que consiste en el uso de RL con ”Ensemble of Identical Independen Evaluators (EIIE)”, un ”Portfolio-Vector Memory (PVM)”, un esquema ”Online Stochastic Batch Learning (OSBL)” y una función de reward "fully exploiting and explicit". Realizando ademas tres instancias para el calculo de la función policy: una con *Convolutional Neural Network (CNN)*, una con una básica *Recurrent Neural Network (RNN)*, y una con *Long Short-Term Memory (LSTM)*. Para evaluar su desempeño se utilizaron las 12 Cryptocurrencies con mayor volumen de transacción, realizando transacciones cada 30 minutos para el portafolio. Los resultados obtenidos, en términos del retornos acumulado de los portafolios, muestran que los modelos de RL se desempeñan mejor que aquellos métodos tradicionales como el "Best Stock", *Uniform Buy and Hold (UBAH)* y el ”*Uniform Constant Rebalanced Portfolios (UCRP)*”, destacando por sobre los otros modelos de deep RL aquel que utiliza CNN.

Otra de las formas de combinar estas técnicas de machine learning, es a través del uso del deep learning para aproximar la función Q del Reinforcement Learning, técnica denominada como deep Q-learning. La investigacion realizada por G. Lucarelli et al. [53], propone un modelo "Double Deep Q-Networks (D-DQNs)", el cual utiliza para el entrenamiento y aproximación de la ”optimal action value function (Q)”, una *Convolutional Neural Network (CNN)* seguida de dos capas totalmente conectadas. Modelo que es utilizado para optimizar un portafolio compuesto por cuatro Cryptocurrencies, obteniendo un desempeño superior a un portafolio *equally weigthed* y otro optimizado a partir de algoritmo genético.

4 | Metodología

4.1. Construcción del Portafolio

Existes múltiples técnicas para la construcción de un portafolio. Por lo que dada la complejidad que significa, a lo largo de los años ha sido un campo ampliamente estudiado. Así, uno de los objetivos de esta investigacion, es plantear una nueva técnica de inteligencia artificial, que permita optimizar la decisión de inversión de un portafolio de Cryptocurrencies.

Si bien algunas investigaciones han utilizado deep Reinforcement Learning con LSTM, CNN y ANN, nosotros proponemos utilizar Transformer para la estimación de la función del valor Q. Así, el modelo propuesto es un deep Transformer Q-learning basado en Reinforcement Learning, el cual va a interactuar en un entorno modelado como un *Markov Decision Process (MDP)*. De esta forma se va a crear un agente que va a actuar como el administrador de un portafolio de Cryptocurrencies, el cual va a recibir la información sobre el estado ellas, para luego decidir si va a comprar, vender o mantener, según la decisión que maximice el Indice de Sharpe del portafolio. Para medir la eficiencia de la técnica propuesta, se utilizan como *Benchmark* un portafolio *equally weighted*, uno modelado según deep LSTM Q-learning y otro según multi-output ANN Q-learning.

4.2. Definición del Problema

Las Criptomonedas han sido en los últimos años uno de los activos que ha adquirido mayor popularidad, con significativa volatilidad producto de importantes ciclos. Por lo que se ha transformado en un activo muy interesante de investigar, dado el crecimiento,

dinamismo y complejidad de su mercado. Esto producto que se trata de un activo muy novedoso, sobre el cual existe aun mucha desconfianza, lo cual ha generado que sus precios tengan cambios abruptos. Sumado a la amplia información relacionada a ellas, las transforman un activo mas sensible a múltiples variables, resultando complejo tomar decisiones de inversión acertadas.

Debido a que un portafolio se compone de un capital que es utilizado para ser invertido en múltiples activos, los cuales a su vez van evolucionando respecto a su precio (aquí diario), es que se genera el principal problema que es la optimización del portafolio. Para ello se debe saber sobre que, cuanto y cuando se debe comprar, vender o mantener un activo en particular. Motivo por el cual, esta investigacion busca aportar al conocimiento de los inversionistas y administradores de portafolios, con nueva información y una metodología que les permitan tomar mejores decisiones en torno a sus portafolios de Cryptocurrencies.

4.3. Reinforcement Learning

El Reinforcement Learning (RL) es un técnica del machine learning que ha ganado popularidad en el mundo de las finanzas en los últimos años. Esta metodología consiste en que se configura un agente el cual interactúa con un determinado entorno reconociendo estados, para luego tomar una decisión desde la cual obtiene una recompensa o una penalización. Lo interesante de esta técnica es que el RL no resuelve un problema de optimización siguiendo una regla en especifico, sino que mediante un proceso de entrenamiento de prueba y error, desde el cual va aprendiendo de acuerdo a las acciones tomadas por el agente y la recompensa recibida, permitiendo que el modelo sea capaz de adaptarse mejor a dinamismos del mercado y nuevos estados, mejorando con el tiempo a partir de sus mismos errores y optimizando el resultado final.

El Q-learning es la base del RL, puesto que es el método de como se actualiza la Q-function, la cual a su vez es aquella que relaciona la recompensa obtenida por el agente con la acción tomada para un determinado estado. Así, el algoritmo de RL que adopta una red neuronal para aproximar el valor de la Q-function es llamado deep Q-Network (DQN). En esta investigacion, resolvemos el problema de optimización de portafolio, mediante el

uso del RL combinado con una red neuronal Transformer para el Q-learning. Operando el portafolio con 15 Cryptocurrencies, las cuales son ajustadas cambiando su porcentaje de inversión (weight) para cada periodo de re-balanceo (para este estudio 30 días).

4.3.1. Entorno: Mercado de Criptomonedas

Se define un entorno para el agente, el cual esta compuesto por los precios de cierre de las 15 Cryptocurrencies con mayor capitalización de mercado[1], tomando como principal información para el entorno los precios de cierre, obtenidos a partir de *Yahoo Finance*, plataforma de libre acceso.

Así, a partir de los precios, se logra calcular el Retorno, la Volatilidad, Co-varianzas e Indice de Sharpe del portafolio. Información con la cual el agente interpreta estados, acciones y recompensa. Análisis que le permite al modelo identificar tendencias de mercado, patrones y potenciales oportunidades.

4.3.2. Estados

El agente dispone de los diferentes precios de las Cryptocurrencies, con los cuales calcula una matriz de covarianzas, que le permiten evaluar si existe algún tipo de cobertura o amplificación del riesgo, ademas de reducir la dimensionalidad sin perder información importante. Esto le permite al agente evaluar la mejor decisión para poder balancear el riesgo-retorno del portafolio, el cual va a definir la mejor decisión según el Indice de Sharpe obtenido. Motivo por el cual, utilizar solo los retornos para los estados, se considera que entrega menos información que utilizar las covarianzas.

Los estados están representados por la matriz de covarianzas de las Crypto, las cuales se obtuvieron a partir de los retornos para un horizonte de observación determinado. Por lo que los estados estarían representados según:

$$S_t = [S_t^1, S_t^2, S_t^3, ..., S_t^N] \tag{4.1}$$

Donde S_t^n representa un vector que contiene las covarianzas de una n Crypto con

[1] De acuerdo con *Yahoo Finance* para el 27 de Agosto de 2024

respecto a las demás para el tiempo t.

4.3.3. Espacio de Acciones

El espacio de acciones posibles por parte del agente, se define pensando en una situación realista en donde un inversor requiere decidir si se debe comprar, vender o mantener una determinada Criptomoneda. Decidiendo ademas la colocación que va a tener dentro del portafolio.

El espacio de acciones esta definido como:

$$A^i = \{1, -1, 0\} \; ; \; a^i \, \epsilon \, \{Buy, Sell, Hold\} \tag{4.2}$$

Por lo que, las acciones para N Crypto van a estar definidas por la matriz:

$$A_t = [a_t^1, a_t^2, ..., a_t^N] \tag{4.3}$$

donde a_t^n representa la acción para una n Crypto en el tiempo t.

4.3.4. Agente

Se define un agente que representa a un inversor, el que tiene como objetivo determinar la colocación de las diferentes criptomonedas, por lo que debe interactuar con la información proveniente del entorno, para así determinar si se debe comprar, vender o mantener. Para lograr tomar la mejor decisión, el agente pasa por una fase de entrenamiento, en donde iterativamente va aprendiendo sobre la mejor decisión basado en una recompensa. Así, el agente va a comenzar con una fase de iniciación, donde se configuran importantes parámetros como la cantidad de información considerada para el estado, el espacio de acciones, el tamaño del portafolio, la capacidad de la memoria de repetición y los híperparametros. Luego, siguiendo una estrategia de *epsilon-greedy*, el agente primero va a decidir entre explorar una nueva solución aleatoria o bien explotar una determinada solución aprendida. En caso de decidir explorar una nueva solución, genera una colocación aleatoria de las Criptomonedas, evaluando su desempeño durante un periodo determinado y agregando esta

experiencia al búfer de memoria, almacenando su estado, acción y recompensa. Por otro lado, en el caso de elegir explotar una solución, utiliza el modelo desarrollado a partir de redes neuronales, para poder predecir las acciones a tomar para cada Crypto del portafolio. Complementariamente, un vez llenado el lote de la memoria a repetir, el modelo es entrenado utilizando la función de *Exp Replay*, según los estados, las acciones y la recompensa obtenida en los diferentes rebalanceos del portafolio. Por lo que este proceso, le permite dar estabilidad, ademas de quebrar la correlación temporal entre procesos consecutivos.

De esta forma es que el agente, utiliza un proceso en el cual experimentando con el entorno, ya sea con acciones aleatorias (exploración) o bien según el modelo (explotación), realiza un proceso de entrenamiento consecutivo para lograr determinar cual es la acción mas conveniente dada una combinación de estados, utilizando redes neuronales para aproximar la función Q que es la encargada de determinar la relación par estado-acción con la recompensa futura que se obtendrá a partir de la acción.

En este paper se propone configurar un agente basado en Transformer, uno de los modelos mas recientes del deep learning. Así, se busca que el modelo sea mas eficiente que otros de deep learning y tradicionales, debido a que el Transformer es capaz de capturar dependencias de largo plazo sin tener una estructura recurrente que genere problemas de perdida de información. Este agente es comparado con las otras 2 principales redes neuronales utilizadas en el ultimo tiempo, de tal manera de tener una perspectiva completa de cual se adapta mejor para las Criptos. Por lo que primero se utiliza el modelo basado en Transformer. Para luego ser comparado con un modelo *multi-output* de redes neuronales conectadas entre si (ANN). Adicionalmente se decide realizar el proceso de RL utilizando una red neuronal mas compleja que logre captar mejor las relaciones de corto y largo plazo, eligiendo una red neuronal recurrente del tipo LSTM.

4.3.5. Función de Recompensa

La función de recompensa es una parte importante del funcionamiento del modelo, debido a que es la función que indica si la decisión tomada ha sido buena o mala, permitiéndole al agente poder mejorar sus futuras decisiones. En el caso de este estudio, se debe definir una función de recompensa que permita simular la situación de un inversionista

que debe colocar las diferentes Cryptocurrencies dentro de un portafolio. Sin embargo, a diferencia de la mayoría de los estudios desarrollados, quisimos generar una función de recompensa que ademas de considerar el retorno generado por la venta del activo, también tomara en consideración el riesgo, representado por la volatilidad del precio de la Cryptos. Motivo por el cual la función de recompensa queda definida según el Indice de Sharpe del portafolio, obtenido a partir del retorno ponderado (*weighted profit*) generado por el porfolio y la volatilidad del mismo. Ademas, se define un periodo de evaluación para el portafolio, correspondiente al tiempo utilizado para el re-balanceo (t_r), el cual va desde t_r hasta t_f. Según la siguiente formula

$$recompensa(R) = Sharpe\ Index(SI) = \frac{W_{t_f} \cdot \bar{P}_{t_f}}{\sqrt{W_{t_f}^T \cdot \sum \cdot W_{t_f}}} \tag{4.4}$$

$$\bar{P}_{t_f} = \frac{\sum_{t=t_r}^{t_f} p_t - p_{t-1}}{t_f - t_r} \tag{4.5}$$

$$W_{t_f} = \begin{cases} W_{buy},\ if\ a_t^i > 0 \\ 0,\ if\ a_t^i = 0 \\ W_{sell},\ if\ a_t^i < 0 \end{cases} \tag{4.6}$$

Donde W_{t_f} corresponde al porcentaje de colocación de la Crypto en el portafolio para el periodo t_f. Mientras que, $\bar{P}_{t_f}$ es la matriz de retornos promedios de las Crypto en el periodo de evaluación definido entre t_f y t_r, donde $t_f > t_r$. Y $\sum$ corresponde a la matriz de covarianza de los retornos para el mismo periodo. Ademas, W_{t_f} corresponde a la colocación del activo, donde va a ser un valor positivo en el caso en que el agente decida comprar la Crypto, negativo si decide vender y 0 si decide mantener la posición.

Al definir el retorno promedio $\bar{P}_{t_f}$, se esta permitiendo la comparabilidad de la recompensa entre los diferentes periodos en los que va interactuando el agente, haciendo mas justa la comparación entre diferentes periodos transcurridos.

4.4. Deep Q-Learning basado en Reinforcement Learning

Se utiliza la técnica del Reinforcement Learning, buscando mejorar la eficiencia con la que se toman las decisiones de compra, venta o mantener. Ademas, para la investigacion se utilizan dos importantes consideraciones: no impacto sobre el mercado, es decir el modelo de transacciones no va a influir sobre las otras transacciones en el mercado ni en sus precios; y alta liquidez del mercado, donde el modelo siempre va a ser posible de ejecutar una compra o venta a precio de cierre.

Siguiendo la metodología del Q-learning, se va a necesitar calcular el valor de la función Q según el par estado-acción. Sin embargo dada la complejidad de la información, con una alta dimensionalidad del espacio par estado-acción, es que se utilizan modelos de redes neuronales del deep learning para aproximar el valor de la función optima del par estado-acción $Q^*(s, a)$. En este paper se propone un modelo basado en Transformer, que es comparado con otros 2: red neuronal clásica con Multi-output, y otro modelo basado en redes neuronales recurrentes con LSTM. La función Q, es la encargada de estimar la recompensa acumulada esperada, al tomar una acción A_t en el estado S_t, siguiendo una política π, descrita en la ecuación 4.7.

$$Q^*(S_t, A_t) = max\ \mathbb{E}_\pi[R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + ...] \tag{4.7}$$

Donde γ es el factor de descuento, R_t la recompensa en el tiempo t, π la acción tomada siguiendo la política o estrategia, A_t la acción tomada y S_t el estado en el tiempo t. De tal manera, que según la ecuación de Bellman, la función optima de estado-acción, estaría definida como:

$$Q^*(S_t, A_t) = \mathbb{E}_\pi[R + \max_{A'} Q^*(S', A')] \tag{4.8}$$

En este proceso de optimización, para el algoritmo de RL se define en el agente una clase, la cual va a ser la encargada junto con el modelo de deep learning, de predecir y aproximar los Q-valores para cada posible acción (Comprar, vender y mantener) para cada Crypto del portafolio. De esta forma, es que el agente va a seleccionar la mejor opción

siguiendo la política, que a su vez selecciona la acción con el mayor valor de Q para cada Crypto, según la ecuación 4.9.

$$\pi(S_t) = arg\ \max_{A_t} Q(S_t, A_t) \tag{4.9}$$

donde π corresponde a la política, S_t al estado en el tiempo t y A_t a las acciones tomadas para cada una de las Crypto según la ecuación 4.10.

$$a_t^i = arg\ \max_{a^i \epsilon A^i} Q(s_t^i, a^i) \tag{4.10}$$

Donde i corresponde a cada una de las N Crypto del portafolio, siguiendo un estado s_t^i.

4.4.1. Modelo propuesto: deep Transformer Q-learning

El modelo propuesto por la investigacion, es un modelo basado en una arquitectura Transformer, diseñado para lograr capturar de mejor forma relaciones complejas entre las diferentes Cryptos, utilizando un mecanismo de atención, permitiendo analizar portafolios mas complejos y con un set de datos mas grande.

Este modelo, tiene como entrada, una matriz de covarianzas con la forma *(portfolio_size, porfolio_size)*, siendo *portfolio_size* el numero de N Cryptos que componen el portafolio. Por tanto, al utilizar data secuencial permite al modelo aprender sobre tendencias e ínter-dependencias, apoyándose con la matriz de covarianzas que ayuda a capturar la relación entre los retornos de las diferentes Cryptos en un horizonte de observación determinado de 180 días.

La información contenida en la entrada es procesada por un modelo que se compone de una arquitectura de 5 secciones. La primera de ellas, es una capa de entrada, que define un tensor de 2D ($N \times N$).

$$Input:\ X_t \tag{4.11}$$

Seguido se utiliza un bloque de *Transformer Enconder*, el que se compone de una primera capa de normalización, para luego utilizar un mecanismo de *multi-head attention* (4

çabezasz una llave de dimensión igual a 64), esto con la idea de lograr captar dependencias entre las diferentes Cryptos. Seguido, se utiliza un mecanismo de descarte para prevenir el efecto de *overfitting*, junto con una capa de concatenación, la cual se conecta a una red de avance. Esto, de tal manera que el Transformer encoder pueda capturar relaciones no lineales mas complejas que no pudieron ser captadas por la sección de *self-attention*, logrando combinar efectos locales y globales. Esta ultima parte del *econder* esta compuesta de una capa de normalización, una capa densa con función de activación ReLU, un mecanismo de descarte, otra capa densa, para finalizar con una función de concatenación.

Por tanto el bloque de *Transformer encoder* va a ser: capa de normalización, multi-head attention, conexion residual y descarte, seguido una red de avance, otra conexión residual y una concatenación de salida del *Transformer encoder*. Esto con la idea de capturar diferentes efectos, lineales, no lineales, locales y globales, manteniendo la información crucial sin ser degradada a medida que la red se extiende. Proceso que se rige bajo las ecuaciones:

$$capa\ Normalization:\ z_t = capaNorm(s_t) \tag{4.12}$$

$$Multi-Head\ Attention:\ Attention_i(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4.13}$$

$$h_{attn} = Concat(Attention_1, ..., Attention_h)W_o \tag{4.14}$$

$$Residual\ Connection\ and\ Dropout:\ h_{res1} = h_{attn} + s_t \tag{4.15}$$

$$Feed-Forward\ Network:\ h_{ff1} = ReLU(W_1 h_{res1} + b_1) \tag{4.16}$$

$$h_{ff2} = W_2 h_{ff1} + b_2 \tag{4.17}$$

$$Residual\ Connection:\ h_{res2} = h_{ff2} + h_{res1} \tag{4.18}$$

$$Output\ of\ Transformer\ Encoder:\ h_{encoder} = h_{res2} \tag{4.19}$$

Una vez la información pasa a través del bloque Transformer encoder, se utiliza una capa de *Global Average Pooling* (GAP), la cual reduce la dimensionalidad de salida generada por el bloque Transformer encoder, agregando la información temporal en un vector unidimensional, manteniendo la información global esencial y permitiendo al modelo enfocarse en patrones globales. Con ello, el modelo logra adaptarse para pasar esta información global, acompañada de complementos individuales, a través de una capa densa con 50 unidades y función de activación ELU, y una capa de descarte con probabilidad de un 0.5, incorporando nuevamente capturar efectos lineales, pero esta vez a datos procesados. Proceso matemático representado por las ecuaciones:

$$Global\ Average\ Pooling:\ h_{pool} = \frac{1}{n}\sum_{t=1}^{n} h_{encoder}[t] \tag{4.20}$$

$$Dense\ capa:\ h_{dense} = ELU(W_d h_{pool} + b_d) \tag{4.21}$$

$$Dropout:\ h_{drop} = Dropout(h_{dense}, rate = 0{,}5) \tag{4.22}$$

Finalmente para el resultado del modelo, en cada una de las Cryptos se define una capa densa con 3 unidades, correspondientes al Q-valor de cada acción (comprar, vender y mantener), utilizando una función de activación linear, ya que los Q-valores pueden tomar valores positivos y negativos. Si bien, se crea una salida para cada Crypto, todas comparten las mismas capas ocultas.

$$Output:\ Q_t = W_i h_{drop} + b_i \tag{4.23}$$

$$Weights\ Mapping:\ w_t^i = \frac{exp(Q_t^i)}{\sum_{j=1}^{N} exp(Q_j^i)} \tag{4.24}$$

Por tanto, siguiendo esta arquitectura el Transformer genera un output, el cual es procesado con una dense capas, prediciendo el Q-valor para cada acción (comprar, vender o mantener) de cada una de las Cryptocurrencies. Para luego con aquel Q-valor, estimar el weight correspondiente, según un proceso de normalización de los Q-valores, de tal manera que todo sume 1, manteniendo la consistencia del portafolio.

A partir de lo anterior, es compilado el modelo utilizando una función de perdida (*loss function*) de Error Cuadrático Medio o *Mean Square Error* (MSE) 4.44, junto con el optimizador *adam*, el cual se ajusta mejor a datos financieros debido a su capacidad de adaptarse a diferentes escalas de la gradiente. De esta forma, es que mediante el entrenamiento de la red neuronal se van a actualizar los parámetros θ de la red, minimizando la diferencia temporal o *temporal difference* (TD), que a su vez va a actuar como la gradiente de la función de perdida MSE. Los Q-valores (Q_θ) objetivos, van a ser derivados utilizando la ecuacion de Bellman descrita en la ecuación 4.8.

$$L(\theta) = \mathbb{E}\left[(\hat{Q}_\theta - Q_\theta)^2\right]$$

$$L(\theta) = \mathbb{E}\left[(R_t + \gamma \underset{A_{t+1}}{max}\, Q_\theta(S_{t+1}, A_{t+1}) - Q_\theta(S_t, A_t))^2\right] \tag{4.25}$$

donde θ corresponde a los parámetros de la red neuronal, definidos como $\theta = \left\{W^{(l)}, U^{(l)}, b^{(l)}\right\}_{l=1}^{L+1}$, R_t la recompensa en el tiempo t, γ al factor de descuento y $Q_\theta(S_t, A_t)$ a la predicción del Q-valor para el par estado-acción (S_t, A_t).

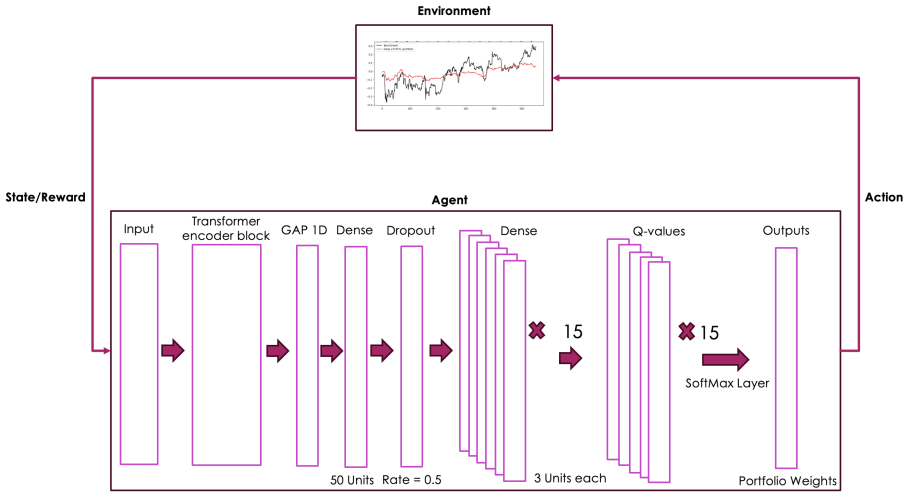


Figura 4.1: RL con arquitectura deep Transformer Q-learning
Fuente: Elaboración propia

4.4.2. deep multi-output ANN Q-learning

El primer modelo utilizado como *Benchmark*, es un modelo secuencial clásico de redes neuronales con múltiples salidas. Este modelo, tiene como datos de entrada, una matriz de covarianzas con la forma *(portfolio_size, porfolio_size)*, siendo *portfolio_size* el numero de N criptomonedas que componen el portafolio. Por tanto, esta matriz estaría representando la relación que existe entre los retornos de las Cryptos en un horizonte de observación determinado de 180 días.

Así, la información contenida en la entrada ingresa a un modelo que se compone de una arquitectura de 4 secciones. La primera de ellas, es una capa de entrada, que define un tensor de 2D. Seguido se utiliza una capa de aplanamiento o *flatten layer*, la cual reduce la matriz de covarianzas desde 2D a 1D, esto con el objetivo de preparar los datos para que sean procesados por una capa densa manteniendo la relación entre las Cryptos. Luego, con el objetivo de lograr captar complejas dependencias temporales y patrones en los datos, dos capas completamente conectadas procesan la los datos de entrada .ªplanados", con una primera capa con 100 unidades y una función de activación ELU (*Exponential Linear Unit)*, y una segunda capa con 50 unidades con función de activación ELU, utilizada debido que, a diferencia de la función ReLU, permite manejar de mejor forma los valores negativos. Finalmente para el resultado, para cada una de las Cryptocurrencies se define una

capa densa con 3 unidades, correspondientes al Q-valor de cada acción (comprar, vender y mantener), utilizando una función de activación linear, ya que los Q-valores pueden tomar valores positivos y negativos. Si bien, se crea un resultado para cada Crypto, todas comparten las mismas capas ocultas.

Por tanto, de esta forma el modelo va a recibir datos de entrada, que van a ser procesados por una arquitectura de redes neuronales la cual va a predecir el Q-valor para cada acción (comprar, vender o mantener) de cada una de las Cryptos. Para luego con aquel Q-valor, estimar el peso correspondiente, según un proceso de normalización de los Q-valores, de tal manera que todo sume 1 y manteniendo la consistencia del portafolio. Proceso descrito matemáticamente en las siguientes ecuaciones:

$$Input: \ X_t \tag{4.26}$$

$$Hidden\ capa\ 1: \ H^{(1)} = f(W^{(1)}X_t + b^{(1)}) \tag{4.27}$$

$$Hidden\ capa\ 2: \ H^{(2)} = f(W^{(2)}H^{(1)} + b^{(2)}) \tag{4.28}$$

$$Output: \ Q_t = W^{(L+1)}H^{(L)} + b^{(L+1)} \tag{4.29}$$

$$Weights\ Mapping: \ w_t^i = \frac{exp(Q_t^i)}{\sum_{j=1}^{N} exp(Q_j^i)} \tag{4.30}$$

Donde los datos de entrada X_t van a corresponder a una matriz compuesta por vectores de estados $S_t = \left[s_t^1, \ s_t^2, ..., \ s_t^N\right]$ con N numero de Cryptos. Ademas, $H^{(l)}$ representando a las L capas ocultas, con $H^{(0)} = X_t$. $W^{(l)}$ a la matriz de pesos de las Cryptos en el portafolio, con $W_t = \left[w_t^1, \ w_t^2, ..., \ w_t^N\right]$ y $\sum_{i=1}^{N} w_t^i = 1$. Y $b(l)$ representando al vector de sesgo de cada Crypto para las L capas ocultas. Ademas se definen funciones de activación ELU representadas por $f(\cdot)$. Finalmente Q_t que representa los Q-valores, que va a tener $N \times k$ dimensiones, con k siendo el numero de posibles acciones por activo ($k = 3, \ \{Buy, \ Hold, \ Sell\}$)

A partir de lo anterior, es compilado el modelo utilizando una función de perdida (*loss*

function) de *Mean Square Error* (MSE) 4.44, junto con el optimizador adam, el cual se ajusta mejor a datos financieros debido a su capacidad de adaptarse a diferentes escalas de la gradiente. De esta forma, es que mediante el entrenamiento de la red neuronal se van a actualizar los parámetros θ de la red, minimizando las diferencias temporales (TD), que a su vez van a actuar como la gradiente de la función de perdida MSE. Los Q-valores (Q_θ) objetivos, van a ser derivados utilizando la ecuación de Bellman descrita en la ecuación 4.8.

$$L(\theta) = \mathbb{E}\left[(\hat{Q}_\theta - Q_\theta)^2\right]$$

$$L(\theta) = \mathbb{E}\left[(R_t + \gamma \underset{A_{t+1}}{max}\, Q_\theta(S_{t+1}, A_{t+1}) - Q_\theta(S_t, A_t))^2\right] \tag{4.31}$$

donde θ corresponde a los parámetros de la red neural, definidos como $\theta = \left\{W^{(l)}, b^{(l)}\right\}_{l=1}^{L+1}$, R_t a la recompensa en el tiempo t, γ al factor de descuento y $Q_\theta(S_t, A_t)$ a la predicción del Q-valor para el par estado-acción (S_t, A_t).

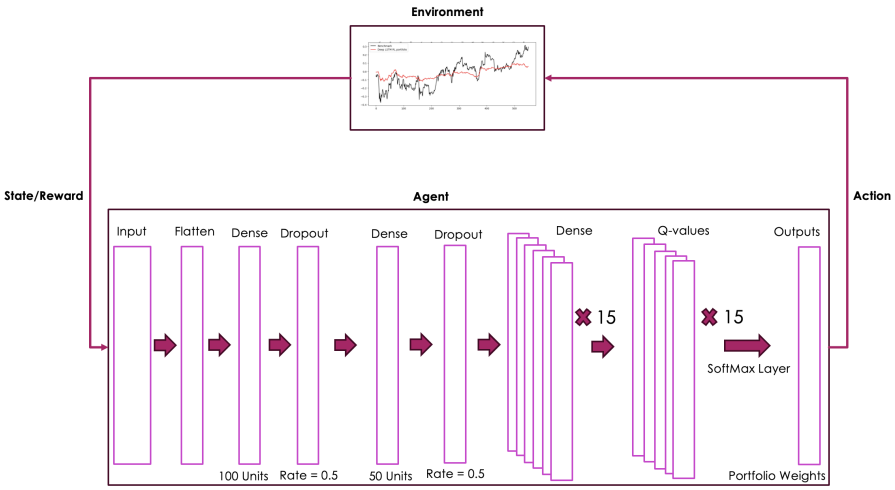


Figura 4.2: RL con arquitectura deep neural network Q-learning
Fuente: Elaboración propia

4.4.3. deep LSTM Q-learning

El segundo modelo utilizado como Benchmark, es un modelo secuencial de redes neuronales recurrentes apalancado en capas LSTM (Long Short-Term Memory), el cual genera múltiples resultados. Este modelo, tiene como datos de entrada, una matriz de

covarianzas con la forma *(portfolio_size, porfolio_size)*, siendo *portfolio_size* el numero de N Cryptos que componen el portafolio. Por tanto, al utilizar datos secuenciales permiten al modelo aprender sobre tendencias e ínter-dependencias, apoyándose con la matriz de covarianzas que ayuda a capturar la relación entre los retornos de las diferentes Cryptos en un horizonte de observación determinado.

La información contenida en los datos de entrada, ingresa al modelo que se compone de una arquitectura de 5 secciones, todas totalmente conectadas, tal como se aprecia en la figura 4.3. La primera de ellas, es una capa de entrada, que define un tensor de 2D ($N \times N$).

Seguido se utiliza una capa LSTM con 50 unidades, que procesa los datos secuenciales fila por fila, generando como resultado una secuencia de estados ocultos ($H^{(1)}$). Luego, este resultado es procesado por una segunda capa LSTM, la cual tiene 50 unidades, con las mismas ecuaciones que la capa anterior pero con parámetros específicos a ella, generando como resultado la secuencia de estados ocultos $H^{(2)}$. El proceso utilizado por cada celda de LSTM es descrito en las ecuaciones a continuación:

$$Forget\ Gate: \ f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{4.32}$$

$$Input\ Gate: \ i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{4.33}$$

$$Candidate\ State: \ \tilde{c}_t = tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{4.34}$$

$$Cell\ State\ Update: \ c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{4.35}$$

$$Output\ Gate: \ o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{4.36}$$

$$Hidden\ State\ Update: \ h_t = o_t \odot tanh(c_t) \tag{4.37}$$

Donde x_t es el vector de entrada, i_t es la puerta de entrada en el tiempo t, f_t es la puerta

de olvido en el tiempo t, $\tilde{c}$ es el estado candidato para el tiempo t, c_t es el estado de la celda en el tiempo t, h_t es el estado oculto de la LSTM en el tiempo t, σ la función de activación sigmoid, $W_f,\ W_i,\ W_c,\ W_o, U_f,\ U_i,\ U_c,\ U_o$ correspondiente a las matrices de pesos y $b_f,\ b_i,\ b_c,\ b_o$ son los vectores de sesgo.

Una vez procesados los datos por las dos capas LSTM, una capa de descarte es aplicada al resultado de la segunda capa LSTM para así prevenir el *overfitting*, con p como la probabilidad de descarte. Finalmente para el resultado del modelo, para cada una de las Cryptocurrencies se define una capa densa con 3 unidades, correspondientes al Q-valor de cada acción (comprar, vender y mantener), utilizando una función de activación linear, ya que los Q-valores pueden tomar valores positivos y negativos. Si bien, se crea un resultado para cada Crypto, todas comparten las mismas capas ocultas.

Por tanto, de esta forma el modelo va a recibir datos de entrada, los cuales van a ser procesados por una arquitectura de red neuronal adaptada con LSTM, la cual va a predecir el Q-valor para cada acción (comprar, vender o mantener) de cada una de las Cryptos. Para luego con aquel Q-valor, estimar el peso correspondiente, según un proceso de normalización de los Q-valores, de tal manera que todo sume 1 y manteniendo la consistencia del portafolio. Proceso descrito matemáticamente en las siguientes ecuaciones:

$$Input:\ X_t \tag{4.38}$$

$$First\ LSTM\ capa\ Outputs:\ H^{(1)} = \left[h_1^{(1)}, h_2^{(1)}, ..., h_T^{(1)}\right] \tag{4.39}$$

$$Second\ LSTM\ capa\ Outputs:\ H^{(2)} = \left[h_1^{(2)}, h_2^{(2)}, ..., h_T^{(2)}\right] \tag{4.40}$$

$$Dropout\ capa:\ H^{(3)} = Dropout(H^{(2)}, p) \tag{4.41}$$

$$Output:\ Q_t = W^{(L+1)}H^{(L+1)} + b^{(L+1)} \tag{4.42}$$

$$Weights\ Mapping:\ w_t^i = \frac{exp(Q_t^i)}{\sum_{j=1}^{N} exp(Q_j^i)} \tag{4.43}$$

Donde T corresponde al numero de columnas (time steps) de la matriz de datos de entrada, X_t, que va a corresponder a una matriz compuesta por vectores de estados $S_t = \left[s_t^1,\ s_t^2, ...,\ s_t^N\right]$ con N numero de Cryptos. Ademas, $H^{(l)}$ representando a las L capas ocultas LSTM, con $H^{(0)} = X_t$. $W^{(l)}$ a la matriz de pesos de las Cryptos en el portafolio, con $W_t = \left[w_t^1,\ w_t^2, ...,\ w_t^N\right]$ y $\sum_{i=1}^{N} w_t^i = 1$. Y $b(l)$ representando al vector de sesgo de cada Crypto en la capa de resultado. Finalmente Q_t que representa los Q-valores, que va a tener $N \times k$ dimensiones, con k siendo el numero de posibles acciones por activo ($k = 3,\ \{Buy,\ Hold,\ Sell\}$)

A partir de lo anterior, es compilado el modelo utilizando una función de perdida (loss function) de Mean Square Error (MSE) 4.44, junto con el optimizador *adam*, el cual se ajusta mejor a datos financieros debido a su capacidad de adaptarse a diferentes escalas de la gradiente. De esta forma, es que mediante el entrenamiento de la red neuronal se van a actualizar los parámetros θ de la red, minimizando la diferencia temporal (TD), que a su vez va a actuar como la gradiente de la función de perdida MSE. Los Q-valores (Q_θ) objetivos, van a ser derivados utilizando la ecuación de Bellman descrita en la ecuación 4.8.

$$L(\theta) = \mathbb{E}\left[(\hat{Q}_\theta - Q_\theta)^2\right]$$

$$L(\theta) = \mathbb{E}\left[(R_t + \gamma \underset{A_{t+1}}{max}\ Q_\theta(S_{t+1}, A_{t+1}) - Q_\theta(S_t, A_t))^2\right] \tag{4.44}$$

donde θ corresponde a los parámetros de la red neuronal, definidos como $\theta = \left\{W^{(l)}, U^{(l)}, b^{(l)}\right\}_{l=1}^{L+1}$, R_t a la recompensa en el tiempo t, γ al factor de descuento y $Q_\theta(S_t, A_t)$ a la predicción del Q-valor para el par estado-acción (S_t, A_t).

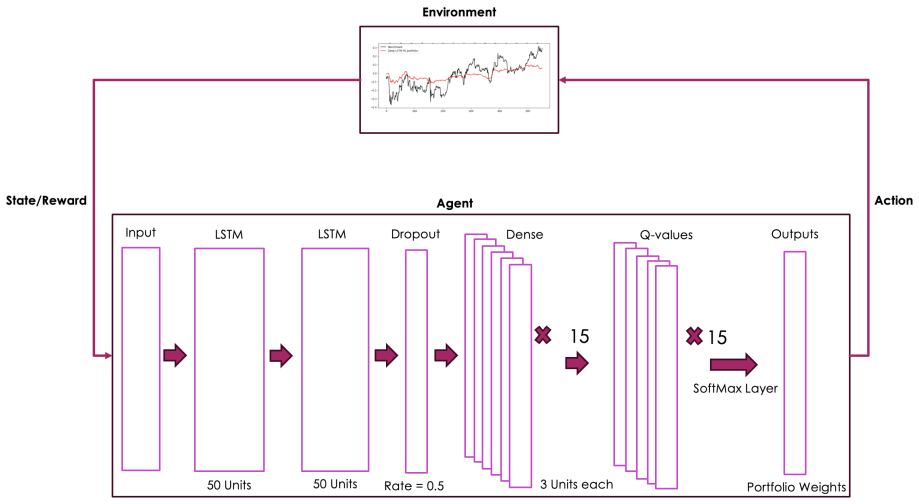


Figura 4.3: RL con arquitectura deep LSTM Q-learning
Fuente: Elaboración propia

4.5. Definición del Set de datos

Dada la popularidad que han tenido las Cryptocurrencies en el ultimo tiempo, y con el objetivo de poder probar cual de los modelos es mas eficiente en la colocación del portafolio, es que se utilizaron como datos de entrada un set de Cryptocurrencies, acompañadas de sus precios de cierre. Permitiendo tener una mejor comparación de sus retornos ya que se tiene un mayor control sobre la calidad y especificaciones de la data.

Complementariamente, los años definidos para el estudio, han sido condicionados por la disponibilidad de datos de Cryptos en los últimos 10 años, donde al tratarse de un instrumento financiero nuevo, no se tiene una basta historia, sino que todo el análisis se remonta mas en los últimos 10 años. Asimismo, buscando tener las Crypto que mejor representen el comportamiento del mercado, es que se seleccionaron las 30 Crypto con mayor capitalización de mercado, sin embargo, dado que muchas de ellas solo tienen historia muy reciente (no mas de 3 años), es que finalmente se consideraron 15 Cryptos, las cuales cumplían con tener información suficiente desde 2018 a 2024.

4.6. Métricas de evaluación

4.6.1. Índice de Sharpe

Diferentes métricas son utilizadas en el campo para evaluar el desempeño de los modelos y portafolios. Uno de las mas directas y usadas es el retorno anualizado, sin embargo este no considera el riesgo o variabilidad que ha implicado el activo. Por tanto, una de las métricas de evaluación utilizadas para ello es la introducida por Sharpe [**?**], la cual incorpora la relación riesgo-retorno para evaluar el desempeño del portafolio.

$$SR = \frac{R_p - R_f}{\sigma_p} \tag{4.45}$$

Donde R_p corresponde al retorno esperado del portafolio, normalmente el promedio. R_f es la tasa libre de riesgo, donde al tratarse de Cryptocurrencies, se considera que va a ser 0. Y σ_p va a ser la volatilidad de los retornos del portafolio, la que va a representar el riesgo del mismo. EL Indice de Sharpe es generado sobre retornos diarios.

4.6.2. Intervalos de confidencialidad para el Índice de Sharpe

Dado que el Sharpe Ratio es un instrumento donde estadísticamente se miden sus resultados para ser comparados, es que consideramos importante obtener su significancia estadística para afirmar que dos portafolios obtuvieron resultados diferentes. Así, como la distribución del Indice de Sharpe puede ser descrita explícitamente como *T-student* [64][65], se utiliza el intervalo de confianza con 95 %, para determinar si el Indice de Sharpe promedio del portafolio, obtenido con los modelos propuestos, son estadísticamente diferentes del Benchmark. $\widehat{SR}$, representado en la ecuación 4.46.

$$Confidence\ Interval\ (SR):\ \widehat{SR} \pm 1{,}96 \times \sqrt{\left(1 + \frac{1}{2}\widehat{SR}^2\right)/T} \tag{4.46}$$

4.6.3. Índice Sortino

El Índice de Sortino o Sortino Ratio (SoR) es una variación del Índice de Sharpe, el cual solo considera desviaciones negativas, por lo que la volatilidad es calculada en base a las bajas. Una métrica apropiada ya que alzas abruptas no debiesen ser una preocupación.

$$SoR == \frac{R_p - R_f}{\sigma_d} \tag{4.47}$$

donde R_p corresponde al retorno esperado del portafolio, normalmente su valor promedio. R_f es la tasa libre de riesgo, la cual es definida con valor 0 para las criptomonedas. $sigma_d$ es la volatilidad negativa de los retornos del portafolio, la cual representa el riesgo de caídas. Este indice es calculado en base diaria.

4.6.4. Indicador Maximum Drawdown (MDD)

El indicador de Máxima Caída o *Maximum Drawdown (MDD)*, va a indicar la mayor la mayor perdida observada a partir de un peak. Eso implica que el MDD va a medir el riesgo a la baja del portafolio, por lo que un menor valor de MDD implica que la estrategia evita perdidas profundas, siendo un aspecto critico en entornos muy volátiles como las criptomonedas. Por lo que un menor valor de MDD va a significar una estrategias mas robusta en condiciones de incertidumbre y volatilidad.

$$MDD = \underset{\gamma>t}{max}\frac{\rho_t - \rho_\gamma}{\rho_t} \tag{4.48}$$

4.6.5. Índice de Calmar

Cambien es utilizado para el análisis el indice de Calmar o *Calmar Ratio (CR)*, el cual es una medida de riesgo ajustado que compara el retorno anual con su máxima caída durante el periodo (o Maximum drawdown).

$$CR = \frac{R_p}{MDD} \tag{4.49}$$

Donde R_p corresponde al retorno anualizado del portafolio. Un mayor valor del indice

de Calmar, va a indicar que tan bien esta desempeñándose el portafolio en relación a su mayor riesgo expuesto. Por lo que es deseable tener un mayor valor de CR ya que va a indicar que el portafolio es mas eficiente al ser capaz de obtener grandes retornos evitando grandes caídas. Este es un aspecto muy relevante en un entorno de criptomonedas, ya que es un mercado que tiende a tener dramáticas caídas de precios de forma repentina.

4.6.6. Benchmarks

Para evaluar los resultados del modelo propuesto, necesitamos compararlos con aquellos métodos financieros populares y ampliamente utilizados. El primero de ellos es el modelo clásico basado en el promedio aritmético, donde se invierte de forma equivalente en todos los activos del portafolio sin importar su volatilidad, ni retornos previos, *equally weigthed*. El segundo de ellos, es un modelo basado en una red neuronal simple, *deep multi-output ANN Q-Learning*. Mientras que el tercero, es un modelo basado en una red neuronal recurrente muy utilizada en el campo de las finanzas en los últimos años, *deep LSTM Q-learning*.

5 | Resultados Experimentales

5.1. Configuración de Métricas y parámetros

Como se definió en la sub sección 4.6, la métrica de evaluación de los modelos va a ser el Indice de Sharpe, el cual a su vez tendrá que ser significativamente superior o inferior, para demostrar que es diferente a los otros modelos propuestos.

La tabla 5.1 presenta los híperparametros utilizados para el entrenamiento y test de los modelos. Los datos utilizados son basado en días, por lo que se define una ventana de 180 días, un lote de 32 días y un periodo de re-balanceo de 30 días. Ademas se realizan experimentos considerando 50 ciclos de iteración y 200 ciclos de iteración, de tal manera de tener diferentes largos de entrenamiento de los modelos. Respecto a las acciones a realizar por el agente, van a ser 3: comprar, vender o mantener, permitiendo la existencia de venta corta. No se limitan ni máximos, ni mínimos respecto al numero de acciones a comprar.

Parámetros	Valor
Tamaño ventana	180
Tamaño del lote	32
Tasa de aprendizaje	0.001
Numero de ciclos	{50, 200}
Periodo de Re-balanceo	30
Optimizador	Adam
Tamaño del búfer de repetición	32
Alpha	0.5
Gamma	0.95
Epsilon	1
Epsilon min	0.01
Epsilon decaimiento	0.99

Tabla 5.1: Híperparametros del modelo
Fuente: Elaboración propia

5.2. Set de datos

El estudio realiza una comparación de diferentes modelos basados en Reinforcement Learning para la colocación de los diferentes activos de un Portafolio. Para ello, en un comienzo se seleccionan los precios de cierre diarios de las 30 Cryptocurrencies con mayor capitalización de mercado para el 2024 [2]. Sin embargo, dado que mucha de ellas no tienen datos mas allá de 3 años de antigüedad, es que por consistencia se seleccionan las 15 Cryptos que cumplen con la data para los años desde 2018 a 2024. Los datos fueron extraídos desde Yahoo Finance, plataforma gratuita y de libre acceso[3]. La tabla 5.2 resume los retornos diarios y las desviaciones de las Cryptos utilizadas para conformar el portafolio.

Los experimentos son realizados considerando diferentes conjunto de datos, primero se considera el total de datos desde 2018 a 2024, luego los datos se separan en tres conjuntos

[2]De acuerdo con Yahoo Finance para el 27 de Agosto de 2024
[3]https://finance.yahoo.com/markets/crypto/all/

considerando la crisis generada por el Covid-19, en donde existió una gran volatilidad en los mercados, que puede afectar los resultados de los modelos. Así, estos conjuntos se definieron pre pandemia desde 2018/01/01 a 2019/12/31, durante pandemia 2020/01/01 a 2021/12/31, y post pandemia desde 2022/01/01 a 2024/01/01.

La tabla 5.2, muestra que antes de la pandemia la mayoría de las Cryptos analizadas tenían retornos promedios diarios negativos, situación que cambio abruptamente durante la pandemia, donde todas las Cryptos se vieron favorecidas, teniendo todas retornos medios positivos. Resultados que se vieron revertido en el periodo post-pandemia, donde todas las Cryptos mostraron resultados negativos. Respecto a las desviaciones estándar, el periodo con valores mas elevados fue durante la pandemia, lo que demuestra la alta volatilidad que tuvieron las Criptomonedas durante aquel periodo.

Ticker	Periodo Total	Pre-Pandemia	Pandemia	Post-Pandemia
ADA-USD				
Retornos (%)	-0.01	-0.42	0.51	-0.12
Desviación estándar (%)	5.56	5.91	6.32	4.19
BCH-USD				
Retornos	-0.10	-0.34	0.10	-0.07
Desviación estándar	5.77	6.50	6.32	4.21
BNB-USD				
Retornos	0.17	0.07	0.50	-0.07
Desviación estándar	5.29	5.84	6.31	3.19
BTC-USD				
Retornos	0.05	-0.09	0.26	-0.02
Desviación estándar	3.68	3.94	4.11	2.88
DASH-USD				
Retornos	-0.16	-0.44	0.16	-0.20
Desviación estándar	5.50	5.16	6.77	4.27
DOGE-USD				
Retornos	0.11	-0.20	0.61	-0.09
Desviación estándar	7.14	5.58	10.08	4.53
EOS-USD				
Retornos	-0.11	-0.17	0.02	-0.18
Desviación estándar	5.92	6.55	6.70	4.20
ETH-USD				
Retornos	0.05	-0.24	0.46	-0.07
Desviación estándar	4.75	4.96	5.43	3.66
LINK-USD				
Retornos	0.14	0.13	0.33	-0.04
Desviación estándar	6.51	7.44	7.16	4.58
LTC-USD				
Retornos	-0.05	-0.23	0.17	-0.10
Desviación estándar	5.04	5.19	5.79	3.99
TRX-USD				
Retornos	0.03	-0.19	0.24	0.05
Desviación estándar	5.77	7.42	6.02	2.97
USDT-USD				
Retornos	0.00	0.00	0.00	0.00
Desviación estándar	0.36	0.47	0.40	0.04
XLM-USD				
Retornos	-0.06	-0.32	0.24	-0.10
Desviación estándar	5.50	5.76	6.49	3.95
XMR-USD				
Retornos	-0.04	-0.28	0.22	-0.06
Desviación estándar	4.96	5.36	5.64	3.63
XRP-USD				
Retornos	-0.06	-0.34	0.20	-0.04
Desviación estándar	5.69	5.45	6.99	4.31

Tabla 5.2: Lista de Criptomonedas con sus respectivos Retornos y desviaciones estándar en base diaria
Fuente: Elaboración propia

5.3. Experimento 1 - Periodo Total

La primera de las pruebas, ha sido enfocada en analizar el rendimiento de los diferentes modelos considerando todo el set de datos. Frente a esto, el modelo que mejor logra responder frente a los datos es el modelo multi-output ANN, con 50 iteraciones, logrando un Indice de Sharpe 1.0621 %, significativamente superior al portafolio equal weighted de 0.6068 % y al modelo deep LSTM Q-learning para 50 iteraciones (0.6744 %) y 200 iteraciones (0.9347 %). Ademas todos los modelos, ANN, LSTM y Transformer, para todas sus iteraciones, logran ser significativamente superiores al portafolio Equal Weighted. Esto explicado en el sentido que los modelos de deep Q-learning son mas dinámicos al momento de realizar las decisiones de compra y venta.

Los modelos ANN, logran mejores resultados que los otros dos modelos que utilizan redes neuronales, a pesar de ser un modelo simple del deep learning, logrando adaptarse mejor a los datos de las Crypto. Sin embargo, el tiempo de ejecución para estos modelos simples con ANN, es hasta 3 veces superior que los otros dos modelos de deep learning, marcando mayor diferencia a medida que se aumentan las iteraciones.

En términos del Sortino Ratio, el modelo basado en LSTM logra tener el mejor desempeño, evidenciando una estrategia mas eficiente al evitar grandes caídas en los retornos. Aspecto que queda demostrado con su mejor desempeño según MDD y obteniendo un considerable mayor Calmar Ratio. Así si bien la estrategia basada en ANN logra mejor SR, también tiende exponerse a mayores riesgos, siendo menos eficientes en su administración.

Modelo	N° Ciclos	P. Retornos* (%)	Volatilidad	Sharpe**	Sortino**	MDD	Calmar**	Alpha	Beta	Ex. Time (s)	IC Min***	IC Max***
Equal Weighted	-	0.1516	0.0396	0.6068	0.8511	-0.8134	0.5713	0.0000	1.0000	-	0.5612	0.6524
ANN	50	0.0680	0.0102	1.0621	1.5864	-0.2197	0.8503	0.0005	0.1245	2128	1.0097	1.1145
ANN	200	0.0819	0.0135	0.9609	1.5352	-0.3279	0.6987	0.0004	0.2695	28834	0.9103	1.0115
LSTM	50	0.0522	0.0123	0.6744	1.0595	-0.2658	0.5284	0.0007	-0.1488	2026	0.6280	0.7208
LSTM	200	0.0585	0.0100	0.9347	1.7080	-0.1365	1.1626	0.0005	0.0272	9960	0.8845	0.9849
Transformer	50	0.0861	0.0161	0.8516	1.3412	-0.4475	0.5415	0.0004	0.3055	1692	0.8027	0.9005
Transformer	200	0.0769	0.0142	0.8610	1.4213	-0.3874	0.5516	0.0004	0.2530	10876	0.8120	0.9100

* Los retornos del portafolio y la volatilidad están expresados en base diaria

** El desempeño del Indice de Sharpe, Indice Sortino e Indice Calmar están anualizados

*** IC Min and Max representan el intervalo de confianza con 95 % para el Índice de Sharpe

Tabla 5.3: Experimento 1 - Desempeño de las métricas de evaluación para los portafolios
Fuente: Elaboración propia

Los modelos donde se han utilizado 200 iteraciones logran mejores desempeños que aquellos con 50 (a excepción del multi-output ANN). Sin embargo, una de sus limitantes es el tiempo de ejecución de su fase de entrenamiento, donde los modelos con 200 iteraciones utilizan al menos 5 veces mas tiempo de ejecución que aquellos con solo 50 iteraciones, demostrando el alto costo que significa el entrenamiento.

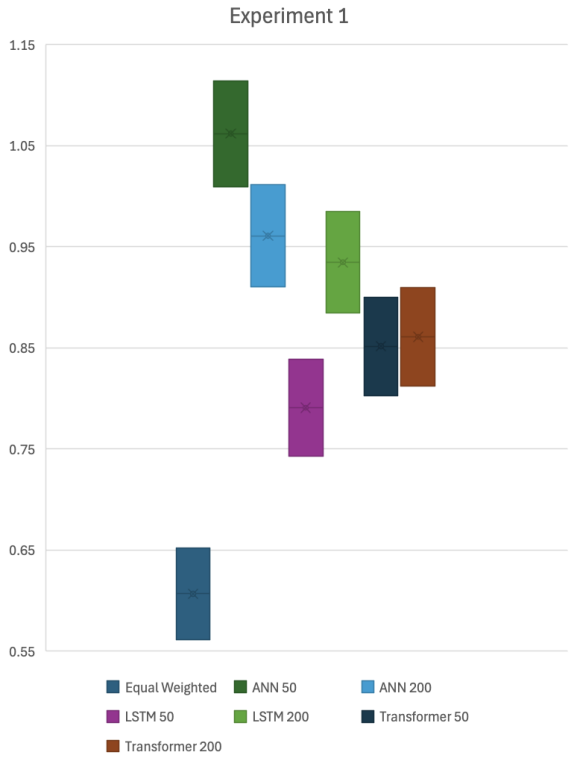


Figura 5.1: Experimento 1 - Gráfico de Caja de los Intervalos de confianza para el Indice de Sharpe
Fuente: Elaboración propia

5.4. Experimento 2 - Pre Pandemia

Respecto al periodo pre-pandemia, el modelo con mejores resultados es el propuesto por esta investigacion, el cual utiliza Transformer con 200 iteraciones, obteniendo un Indice de Sharpe de 1.459 %, seguido del modelo simple de ANN con 200 iteraciones. Ademas, todos los modelos logran tener un rendimiento superior al portafolio Equal Weighted, el cual tuvo resultados negativos para el periodo, mostrando la superioridad de los modelo con machine learning sobre modelos tradicionales, para el periodo antes de la pandemia. Los modelos con LSTM si bien le ganan al portafolio Equal Weighted, no logran tener mejores resultados que sus otros pares que utilizan redes neuronales.

En relación a que tan eficientes son para administrar aquellos riesgos sobre las caídas, vuelve a destacar el modelo propuesto en la investigacion basado en Transformer. Obteniendo un mayor Sortino Ratio, el menor MDD y un considerable mayor Calmar Ratio. Ademas, todos los modelos basados en machine learning lograr ser superiores al equal

weighted, evidenciando la superioridad en la administración del riesgo.

Modelo	N° Ciclos	P. Retornos* (%)	Volatilidad	Sharpe**	Sortino**	MDD	Calmar**	Alpha	Beta	Ex. Time (s)	CI Min***	CI Max***
Equal Weighted	-	-0.0330	0.0388	-0.1609	-0.1903	-0.7151	-0.1117	0.0000	1.0000	-	-0.2340	-0.0878
ANN	50	0.0671	0.0342	0.3220	0.4436	-0.6947	0.2651	0.0004	-0.8504	516	0.2475	0.3965
ANN	200	0.0679	0.0078	1.3836	2.0860	-0.2024	0.9215	0.0007	0.0094	2519	1.2820	1.4852
LSTM	50	0.0647	0.0084	1.2269	1.802	-0.1860	0.9520	0.0007	0.0214	567	1.1308	1.3230
LSTM	200	0.0549	0.0091	0.9562	1.3432	-0.165	0.8983	0.0005	-0.125	2710	0.8686	1.0438
Transformer	50	0.1270	0.0236	0.8657	1.2124	-0.3681	1.0235	0.0011	-0.4693	455	0.7806	0.9508
Transformer	200	0.1158	0.0128	1.4590	2.3392	-0.1125	3.0104	0.0012	0.0309	2282	1.3547	1.5633

* Los retornos del portafolio y la volatilidad están expresados en base diaria

** El desempeño del Indice de Sharpe, Indice Sortino e Indice Calmar están anualizados

*** IC Min and Max representan el intervalo de confianza con 95 % para el Índice de Sharpe

Tabla 5.4: Experimento 2 - Desempeño de las métricas de evaluación para los portafolios
Fuente: Elaboración propia

Respecto a los tiempos de ejecución, los modelos con 200 iteraciones, tardan casi 5 veces mas que aquellos con 50 iteraciones. Sin embargo, a excepción de los modelos con LSTM, los resultados con 200 iteraciones fueron superiores a los con 50.

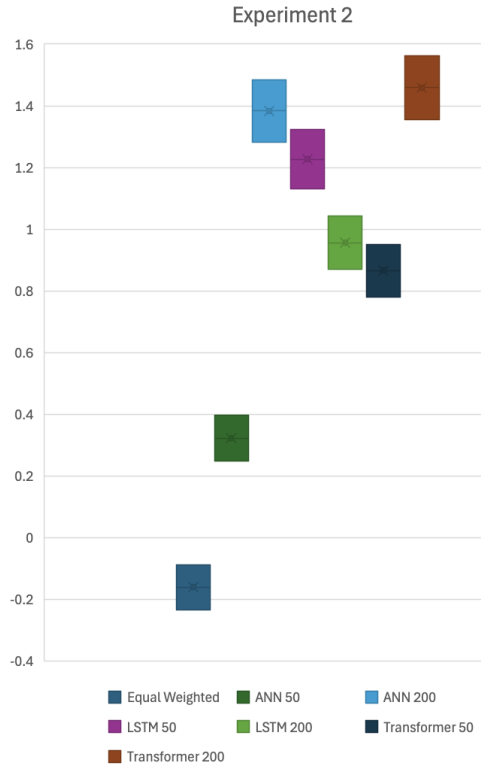


Figura 5.2: Experimento 2 - Gráfico de Caja de los Intervalos de confianza para el Indice de Sharpe
Fuente: Elaboración propia

5.5. Experimento 3 - Pandemia

Durante la Pandemia, el periodo se caracterizo por tener altos niveles de retornos, pero con una alta volatilidad e incertidumbre, viéndose reflejado como las Cryptocurrencies tuvieron un importante subida de valor dado que fueron utilizadas como refugio para muchos inversionistas.

Nuestro modelo propuesto, Transformer, es el único modelo que logra superar al portafolio Equal Weighted, utilizando 200 iteraciones. Así, los demás modelos en general no lograron superar el desempeño del mercado, representado por el portafolio equal weighted, principalmente porque no lograron adaptarse a la volatilidad que presento el periodo, intentando mantenerse lo mas estables posibles, pero sin lograr compensar con mejores retornos.

Los buenos resultados del modelo propuesto por la investigacion, también se ven destacados por su sobresaliente Sortino Ratio y Calmar Ratio, junto con el mas bajo MDD. Lo cual demostraría que el modelo también es el mejor en administrar los riesgos negativos y las caídas.

Los modelos que utilizan 200 iteraciones, superan importantemente a aquellos con solo 50 iteraciones, por lo que se evidencia que para tener mejores resultados, dada la volatilidad de los datos, se van a necesitar un mayor numero de iteraciones, con un mayor procesamiento de los datos que logren captar de mejor forma el mayor dinamismo.

Modelo	N° Ciclos	P. Retornos* (%)	Volatilidad	Sharpe**	Sortino**	MDD	Calmar**	Alpha	Beta	Ex. Time (s)	IC Min***	IC Max***
Equal Weighted	-	0.4737	0.0478	1.5945	2.3305	-0.6246	3.6664	0.0000	1.0000	-	1.4852	1.7038
ANN	50	0.1109	0.0156	1.1229	1.9386	-0.1768	1.8219	0.0012	-0.0170	551	1.0303	1.2155
ANN	200	0.3410	0.0362	1.5104	2.2971	-0.5218	2.6028	-0.0001	0.7421	2415	1.4043	1.6165
LSTM	50	0.0626	0.0186	0.5555	0.7477	-0.2827	0.6044	-0.0007	0.2727	566	0.4776	0.6334
LSTM	200	0.1996	0.0212	1.5060	2.5832	-0.2893	2.2564	0.0002	0.3846	2721	1.4000	1.6120
Transformer	50	0.1053	0.0297	0.5638	1.3929	-0.3013	1.0079	-0.0004	0.2965	472	0.4857	0.6419
Transformer	200	0.1632	0.0134	1.9435	4.8707	-0.1087	4.6787	0.0011	0.1079	2355	1.8202	2.0668

* Los retornos del portafolio y la volatilidad están expresados en base diaria

** El desempeño del Indice de Sharpe, Indice Sortino e Indice Calmar están anualizados

*** IC Min and Max representan el intervalo de confianza con 95 % para el Índice de Sharpe

Tabla 5.5: Experimento 3 - Desempeño de las métricas de evaluación para los portafolios
Fuente: Elaboración propia

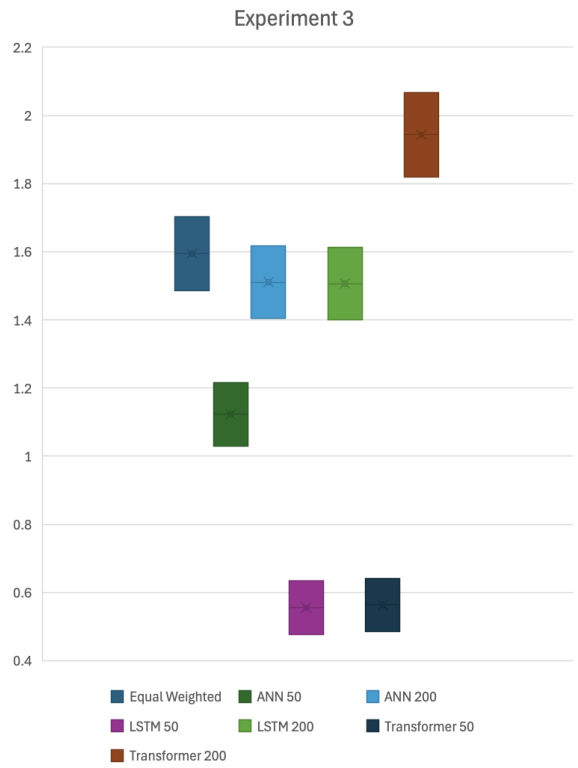


Figura 5.3: Experimento 3 - Gráfico de Caja de los Intervalos de confianza para el Indice de Sharpe
Fuente: Elaboración propia

5.6. Experimento 4 - Post-Pandemia

Para el periodo post-pandemia, los modelos que significativamente presentan los mejores resultados son el propuesto por la investigacion, el deep Transformer Q-Learning de 50 iteraciones, obteniendo un Indice de Sharpe de 0.7839 %, y el modelo con ANN de 200 iteraciones y un Indice de Sharpe de 0.8558 %. Ademas los modelos de deep learning Transformer y ANN, logran superar al portafolio tradicional Equal Weighted. Sin embargo, el modelo con LSTM, no logra superar al portafolio tradicional Equal Weighted, ni utilizando 50 ni 200 iteraciones.

En lo que respecta a la gestión del riesgo negativo y los drowdawns, los modelos Transformer y ANN logran ser los mas destacados considerando sus Sortino, Calmar y MDD. Obteniendo ambos resultados muy similares, lo cual evidenciaría que no existieron grandes diferencias en su desempeño.

El periodo se vio marcado por una corrección en los precios, junto con una menor

volatilidad que el periodo anterior. Ademas lo modelos con 200 iteraciones presentan mejores resultados que aquellos con 50, sin embargo para el modelo con Transformer, el con 50 iteraciones es mejor.

Modelo	N° Ciclos	P. Retornos* (%)	Volatilidad	Sharpe**	Sortino**	MDD	Calmar**	Alpha	Beta	Ex. Time (s)	IC Min***	IC Max***
Equal Weighted	-	0.0536	0.0275	0.3761	0.4276	-0.3292	0.4389	0.0000	1.0000	-	0.3010	0.4512
ANN	50	0.0218	0.0055	0.5771	0.9285	-0.1175	0.4796	0.0003	-0.0663	531	0.4987	0.6555
ANN	200	0.0442	0.0092	0.8558	1.2923	-0.1162	1.0148	0.0003	0.2437	2555	0.7710	0.9406
LSTM	50	0.0066	0.0066	0.1316	0.1994	-0.1565	0.1070	0.0001	0.0007	545	0.0587	0.2045
LSTM	200	0.0110	0.0066	0.3062	0.3767	-0.1340	0.2092	0.0000	0.1413	2442	0.2320	0.3804
Transformer	50	0.0459	0.0092	0.7839	1.1312	-0.1195	1.0254	0.0003	0.2103	526	0.7010	0.8668
Transformer	200	0.0288	0.0087	0.5492	0.7539	-0.1144	0.6571	0.0002	0.2557	2690	0.4714	0.6270

* Los retornos del portafolio y la volatilidad están expresados en base diaria

** El desempeño del Indice de Sharpe, Indice Sortino e Indice Calmar están anualizados

*** IC Min and Max representan el intervalo de confianza con 95 % para el Índice de Sharpe

Tabla 5.6: Experimento 4 - Desempeño de las métricas de evaluación para los portafolios
Fuente: Elaboración propia

Respecto al tiempo de ejecución, no se observa una gran diferencia entre los modelos de deep learning. Sin embargo, al aumentar el numero de iteraciones se observa un importante aumento del costo computacional, con 5 veces mas tiempo de ejecución para las 200 iteraciones.

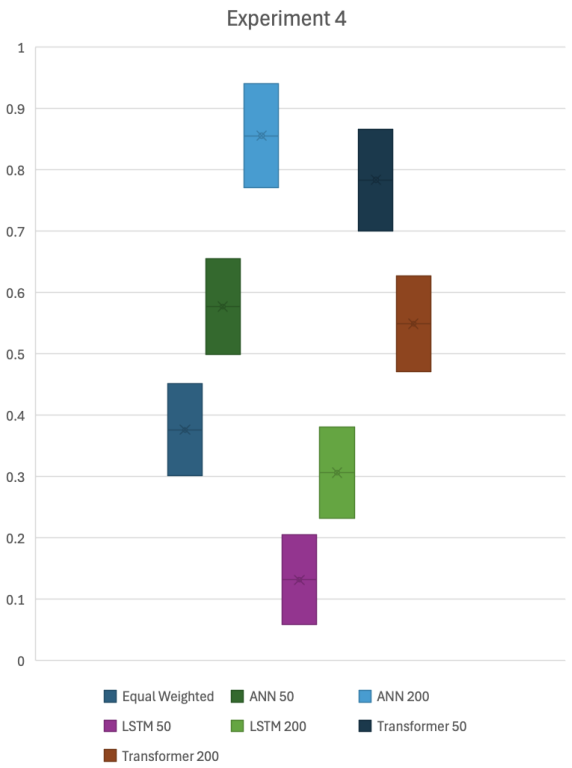


Figura 5.4: Experimento 4 - Gráfico de Caja de los Intervalos de confianza para el Indice de Sharpe
Fuente: Elaboración propia

6 | Principales descubrimientos y resultados

Los resultados indican que, en relación con los diferentes periodos de tiempo, los modelos basados en deep learning obtienen mejores resultados antes y después de la pandemia. Se observa que los modelos de Deep Learning, LSTM y ANN de Múltiples Salidas para el conjunto de datos anterior a la pandemia presentan mejores resultados en términos de Indice de Sharpe, Sortino y Calmar, logrando adaptarse mejor al dinamismo del mercado, lo que se ve confirmado por investigaciones previas.

Sin embargo, durante la pandemia, una vez que aumenta la volatilidad y los precios suben de forma significativa, a los modelos LSTM y ANN de Múltiples Salidas les cuesta seguir el ritmo de los rápidos aumentos de precios, mostrando peores resultados que el mercado. Por tanto, una de las formas de mejorar el rendimiento de estos modelos sería aumentar el número de iteraciones y de entradas para poder captar el dinamismo de los datos. No obstante, esto podría implicar un coste computacional muy elevado. Por ello, para este periodo, el modelo propuesto por la presente investigación, *Deep Transformer Q-learning*, destaca como el único que supera significativamente al mercado, representado por el porfolio Equal-Weighted, demostrando su mejor adaptabilidad a los cambios bruscos y al dinamismo del mercado.

Con respecto al periodo post-pandemia, es evidente que los precios de las criptomonedas tienden a corregir a la baja, y los modelos LSTM no consiguen generar un buen rendimiento. Sin embargo, el modelo propuesto por la presente investigación, *Deep Transformer Q-learning*, y el ANNA de Múltiples Salidas, destacan para este periodo, con un mejor rendimiento que el porfolio Equal-Weighted y el modelo de deep learning LSTM.

En cuanto al tiempo de ejecución y el coste computacional, este es mayor para los modelos de Deep Learning que utilizan redes ANA tradicionales, lo que representa al menos 3 veces más tiempo de ejecución que otros modelos de Deep Learning. Además, este tiempo aumentará significativamente a medida que lo hagan el conjunto de datos y el número de iteraciones. Sin embargo, los resultados experimentales han demostrado que un mayor número de iteraciones no implica necesariamente mejores resultados. En los experimentos 1 y 4, los mejores resultados se dan en aquellos modelos con tan solo 50 iteraciones.

7 | Conclusiones y Recomendaciones Futuras

Esta investigación propuso un modelo *Deep Transformer Q-learning* basado en el marco del Reinforcement Learning, para aprender una estrategia de trading efectiva con el fin de resolver el problema de optimización de portafolios. Este modelo aprende, a partir de los precios históricos de cierre de criptomonedas, a generar dos resultados: la acción de trading (comprar, vender o mantener) y el porcentaje de la acción tomada. Como resultado, el modelo es capaz de controlar las transacciones del portafolio de tal manera que reduce el riesgo sin perder su eficiencia con respecto a los rendimientos. Además, la investigación evalúa el rendimiento de los otros modelos principales del deep Reinforcement Learning, comparandolos con datos de antes, durante y después de la pandemia, identificando su evolución en entornos muy diferentes. La motivación de la investigación es contribuir con un nuevo modelo de deep Reinforcement Learning a la resolución del problema de optimización de portafolios, utilizando otros modelos modernos de machine learning para evaluar el nuevo modelo y su efectividad en diferentes entornos.

Los principales hallazgos del artículo son que el uso de Transformer en el Reinforcement Learning mejora el rendimiento del agente en términos de los indices de Sharpe, Sortino y Calmar, además de reducir el tiempo de entrenamiento del modelo. Por otro lado, otros modelos, como LSTM, no superan necesariamente a los modelos clásicos de redes neuronales artificiales (ANN). Depende mucho de los datos y de la configuración de las arquitecturas. Sin embargo, sí van presentar un tiempo de entrenamiento más corto, siendo el tiempo de ejecución mucho mayor al usar una ANN de múltiples salidas, lo que resulta una limitación al manejar conjuntos de datos más grandes y con una mayor demanda

computacional. Otro hallazgo fue que un mayor número de iteraciones tiende a presentar mejores resultados en general, sin embargo esto no siempre se va a cumplir para todos los casos. Además, la mayoría de los estudios realizados no habían considerado un conjunto de datos en un contexto de crisis, como el de la pandemia, con altos niveles de volatilidad. Por esta razón, se ha demostrado que los modelos de Deep Learning de vanguardia, como LSTM y ANN de Múltiples Salidas, no logran mejores resultados en comparación con el mercado, representado por el portafolio Equal-Weighted, sobre todo cuando se enfrentan a una mayor volatilidad.

La conclusión de esta investigación destaca que el uso de Transformer para desarrollar modelos Deep Q-learning basados en el Reinforcement Learning, es una técnica y una herramienta valiosa para tomar mejores decisiones con respecto a los portafolios de criptomonedas. El uso de un mecanismo de atención y una estructura no recurrente que permite capturar dependencias a largo plazo, hace que los Transformers sean capaces de identificar patrones y tendencias complejas en los datos. No obstante, esto conlleva un alto coste computacional. Las futuras investigaciones en el área del RL, la optimización de portafolios de criptomonedas y el deep learning, podrían estar relacionadas con la mejora de modelos que utilizan Transformer. Incorporando complementos adicionales, nuevas arquitecturas, añadiendo más información de entrada al modelo, utilizando otros tipos de datos o integrando otras técnicas de RL con los Transformers. Además, futuras investigaciones podrían centrarse en integrar Transformers con otros modelos de deep learning como LSTM o CNN, con el fin de capturar otro tipo de dinámicas del mercado que los Transformers por sí solos no pueden capturar. Adicionalmente, se podría usar este modelo Deep Transformer Q-learning en otros tipos de activos. El uso de estas técnicas de deep learning requiere un alto coste computacional, por lo que también es un desafío optimizar los algoritmos de ejecución del modelo.

Bibliografía

[1] F. J. Fabozzi P. N. Kolm, R. Tütüncü. 60 years of portfolio optimization: Practical challenges and current trends. *Eur J Oper Res.*, 234(2):356–371, 2014.

[2] A. Levine L. Pedersen, A. Babu. Enhanced portfolio optimization. *Financial Analysts Journal*, 77(2):124–151, 2020.

[3] R. Uppal V. DeMiguel, L. Garlappi. Optimal versus naive diversification: How inefficient is the 1/n portfolio strategy? *Review of Financial Studies*, 22(5):1915–1953, 2009.

[4] R. O. Michaud. The markowitz optimization enigma: Is 'optimized' optimal? *Financial Analysts Journal*, 45(1):31–42, 1989.

[5] B. Korkie J. D. Jobson. Estimation for markowitz efficient portfolios, journal of the american statistical association. *Journal of the American Statistical Association*, 75 (371):544–554, 1980.

[6] H. M. Markowitz. *Mean-Variance Analysis in Portfolio Choice and Capital Markets*. Frank J. Fabozzi Associates, New Hope, Pennsylvania, 1987. ISBN 1-883249-75-9.

[7] R. Litterman B. Fischer. Global portfolio optimization. *Financial Analysts Journal*, 48(5):28–43, 1992.

[8] S. Shakya S. Singh P. C. Prasad, A. Jaiswal. Portfolio optimization: A study of nepal stock exchange. *Proceedings of International Conference on Sustainable Expert Systems*, 176(1), 2021.

[9] H. Markowitz. Portfolio selection. *J. Finance*, 7(1):77–91, 1952.

[10] H. Liang H. Zhang X. Chao C. Li Y. Dong H. Gao, G. Kou. Machine learning in business and finance: a literature review and research opportunities. *Financial Innovation*, 10(86):35, 2024.

[11] M. Shah J. Shah, D. Vaidya. A comprehensive review on multiple hybrid deep learning approaches for stock prediction. *Intelligent systems with applications*, 16, 2022.

[12] E. Kılıç Z. D. Akşehir. Analyzing the critical steps in deep learning-based stock forecasting: a literature review. *PeerJ Computer Science*, 2024.

[13] O. Iskenderoglu N. Ayyıldız. How effective is machine learning in stock market predictions? *Heliyon*, 10(2), 2024.

[14] D. N. Nagapoojitha. Stock market predictions using machine learning techniques. *Indian Scientific Journal Of Research In Engineering And Management*, 8(7):1–13, 2024.

[15] A. T. Dosdoğru M. Konur, M. Göçken. Stock price prediction using deep learning algorithms based on technical indicators. *Journal of Operations Intelligence*, 2(1): 300–320, 2024.

[16] R. A. Gangthade. Stock price prediction using machine learning. *International Journal for Research in Applied Science and Engineering Technology*, 12(4):3472–3477, 2024,.

[17] K. Gupta A. Singh, B. I. Mazhari. Stock market prediction and visualisation. *International Journal For Science Technology And Engineering*, 12(4):4460–4467, 2024.

[18] P. Luukka J. Porras M. M. Kumbure, C. Lohrmann. Machine learning techniques and data for stock market forecasting: A literature review. *Expert systems with applications*, 197, 2022.

[19] S. Srivastava R. Singh. Stock prediction using deep learning. *Multimedia Tools and Applications*, 76:18569–18584, 2016.

[20] G. Mansourfar H. Rezaei, H. Faaljou. Stock price prediction using deep learning and frequency decomposition. *Expert Syst. Appl.*, 169, 2021.

[21] V. Vakharia Y. Huang. Deep learning-based stock market prediction and investment model for financial management. *Journal of Organizational and End User Computing*, 36(1):1–22, 2024.

[22] P. Xiao. Stock market prediction based on financial news, text data mining, and investor sentiment analysis. *International Journal of Information System Modeling and Design*, 15(1):1–13, 2024.

[23] Z. Li. Review of machine learning with sentimental analysis method for cross-model stock price prediction. *Applied and Computational Engineering*, 71(1):168–173, 2024.

[24] S. Haratizadeh E. Hoseinzade. Cnnpred: Cnn-based stock market prediction using a diverse set of variables. *Expert Syst. Appl.*, 129:273–285, 2019.

[25] J. Wang L. Qin W. Lu, J. Li. A cnn-bilstm-am method for stock price prediction. *Neural Computing and Applications*, 33(1):4741 – 4753, 2020.

[26] N. Chen. Visual recognition and prediction analysis of china's real estate index and stock trend based on cnn-lstm algorithm optimized by neural networks. *PLOS ONE*, 18(2), 2023.

[27] H. Jabani S. S. A. Mosavi M. Nabipour, P. Nayyeri. Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis. *IEEE Access*, 8:150199–150212, 2020.

[28] D. Arganese L. Vollero M. Papi M. Merone L. Bacco, L. Petrosino. Investigating stock prediction using lstm networks and sentiment analysis of tweets under high uncertainty: A case study of north american and european banks. *IEEE Access*, 12: 122239 – 122248, 2023.

[29] Y. Liu Z. Jin, Y. Yang. Stock closing price prediction based on sentiment analysis and lstm. *Neural Computing and Applications*, 32:9713 – 9729, 2019.

[30] L. Ge S. Chen. Exploring the attention mechanism in lstm-based hong kong stock price movement prediction. *Quantitative Finance*, 19:1507 – 1515, 2019.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL `https://arxiv.org/abs/1706.03762`.

[32] F. Wang X. Jia G. Rui M. Li, W. Li. Applying bert to analyze investor sentiment in stock market. *Neural Computing and Applications*, 33:4663 – 4676, 2020.

[33] E. Sefer A. H. B. Gezici. Deep transformer-based asset price and direction prediction. *IEEE Access*, 12:24164–24178, 2024.

[34] X. Zhou H. Wang. Continuous-time mean–variance portfolio selection: A reinforcement learning framework. *Mathematical Finance*, 30(4):1273 – 1308, 2019.

[35] J. Clempner M. García-Galicia, A. Carsteanu. Continuous-time reinforcement learning approach for portfolio management with time penalization. *Expert Syst. Appl.*, 129: 27–36, 2019.

[36] A. E. B. Lim G. Ban, N. Karoui. Machine learning and portfolio optimization. *Manag. Sci.*, 64:1136–1154, 2018.

[37] Y. S. Asawa. Modern machine learning solutions for portfolio selection. *IEEE Engineering Management Review*, 50(1):94–112, 2022.

[38] X. Lyu. Portfolio optimization strategies: New approaches based on machine learning forecasting. *Highlights in Business, Economics and Management*, 40:1077–1082, 2024.

[39] M. K. Mehlawat L. Jia W. Chen, H. Zhang. Mean–variance portfolio optimization using machine learning-based stock price prediction. *Applied Soft Computing*, 100, 2021.

[40] W. Wang Y. Ma, R. Han. Portfolio optimization with return prediction using deep learning and machine learning. *Expert Syst. Appl.*, 165, 2021.

[41] R. Sun T. Ma S. Huang, L. Cao and S. Liu. Enhancing portfolio optimization: A two-stage approach with deep learning and portfolio optimization. *Mathematics*, 12 (21):1–21, 2024.

[42] Sukriti A. Tamuly, G. Bhutani. Portfolio optimization using deep reinforcement learning. *IEEE 5th India Council International Subsections Conference (INDISCON), Chandigarh, India*, pages 1–6, 2024.

[43] E. Paquet F. Soleymani. Deep graph convolutional reinforcement learning for financial portfolio management - deeppocket. *Expert Systems with Applications*, 182, 2021.

[44] R. C. G. Reule W. Härdle, C. R. Harvey. Understanding cryptocurrencies. *Capital Markets: Market Microstructure eJournal*, pages 1–39, 2019.

[45] A. Urquhart L. Yarovaya S. Corbet, B. Lucey. Cryptocurrencies as a financial asset: A systematic analysis. *International Review of Financial Analysis*, 62:182–199, 2019.

[46] S. Krishnan M. Murugappan, R. Nair. Global market perceptions of cryptocurrency and the use of cryptocurrency by consumers: A pilot study. *Journal of Theoretical and Applied Electronic Commerce Research*, 18(4):1955–1970, 2023.

[47] J. Choi S. Otabek. From prediction to profit: A comprehensive review of cryptocurrency trading strategies and price forecasting techniques. *IEEE Access*, 12:87039–87064, 2024.

[48] J. Liang Z. Jiang. Cryptocurrency portfolio management with deep reinforcement learning. *2017 Intelligent Systems Conference*, pages 905–913, 2016.

[49] M. Schnaubelt. Deep reinforcement learning for the optimal placement of cryptocurrency limit orders. *Eur. J. Oper. Res.*, 296(3):993–1006, 2021.

[50] B. Li J. Li H. Xie F. Liu, Y. Li. Bitcoin transaction strategy construction based on deep reinforcement learning. *Applied Soft Computing*, 113(Part B), 2021.

[51] P. Vateekul K. Kumlungmak. Multi-agent deep reinforcement learning with progressive negative reward for cryptocurrency trading. *IEEE Access*, 11:66440–66455, 2023.

[52] W. Chen C. Baca. Deep reinforcement learning for portfolio management of markets with a dynamic number of assets. *Expert Syst. Appl.*, 164, 2021.

[53] M. Borrotti G. Lucarelli. A deep q-learning portfolio management framework for the cryptocurrency market. *Neural Computing and Applications*, 32:17229 – 17244, 2020.

[54] Ben S. Bernanke Mervyn A. King Alan F. Blinder, Robert M. Solow. Digital currencies, decentralized ledgers, and the future of central banking. *Brookings Papers on Economic Activity*, 2017.

[55] R. E. Bellman. Dynamic programming. *Princeton University Press*, 1957.

[56] Bengio Y. y Courville A. Goodfellow, I. Deep learning). *The MIT Press*, 2016.

[57] S. Zhang Q. Zhang C. Wang, Y. Chen. Stock market index prediction using deep transformer model. *Expert Systems with Applications*, 208, 2022.

[58] T. O. Omotehinwa D. O. Oyewolaa, S. A. Akinwunmi. Deep lstm and lstm-attention q-learning based reinforcement learning in oil and gas sector prediction. *Knowledge-Based Systems*, 284, 2024.

[59] H. Y. Kim Y. Baek. Modaugnet: A new forecasting framework for stock market index value with an overfitting prevention lstm module and a prediction lstm module. *Expert Systems with Applications*, 113:457–480, 2018.

[60] J. Xiong C. Zhong B. Yang, T. Liang. Deep reinforcement learning based on transformer and u-net framework for stock trading. *Knowledge-Based Systems*, 262, 2023.

[61] J. Moon N. Lee. Offline reinforcement learning for automated stock trading. *IEEE Access*, 11:112577–112589, 2023.

[62] D. Xu Z. Jiang and J. Liang. A deep reinforcement learning framework for the financial portfolio management problem, 2017. URL `https://arxiv.org/abs/1706.10059`.

[63] B. Gueza E. Benhamou, D. Saltiel and N. Paris. Testing sharpe ratio: luck or skill? *HAL Open Science*, hal-02886500, 2020.

[64] A. W. Lo. The statistics of sharpe ratios. *Financial Analysts Journal*, 58(4):36–52, 2002.

[65] W. F. Sharpe. Capital asset prices: A theory of market equilibrium under conditions of risk. *The Journal of Finance*, 19(3):425–442, 1964.