

Framework para la evaluación y análisis comparativo de modelos de pronóstico en series de tiempo financieras



Ing. Jesús Rodrigo Navia Rodríguez

Director: PhD. Carlos Alberto Cobos Lozada

**Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Sistemas
Grupo de I+D en Tecnologías de la Información (GTI)
Área de interés en Gestión de la Información y Sistemas Inteligentes
Popayán, octubre de 2023**

Framework para la evaluación y análisis comparativo de modelos de pronóstico en series de tiempo financieras

Ing. Jesús Rodrigo Navia Rodríguez

Tesis presentada a la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca para obtener el título de

Magíster en Computación

Director: PhD. Carlos Alberto Cobos Lozada

**Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Sistemas
Grupo de I+D en Tecnologías de la Información (GTI)
Área de interés en Gestión de la Información y Sistemas Inteligentes
Popayán, octubre de 2023**

Resumen

La compra y venta de activos financieros ha experimentado un crecimiento significativo en los últimos años, gracias al avance tecnológico en el sector Fintech. Esto ha permitido captar a una amplia variedad de inversores, tanto profesionales como particulares, que pueden operar a través de intermediarios conocidos como brokers. Estos, a cambio de una comisión, brindan servicios de compra y venta de activos financieros. El trading algorítmico es un enfoque de inversión donde se realizan operaciones automatizadas en el mercado financiero, usando modelos de aprendizaje de máquina (ML), metaheurísticas (MH), redes neuronales artificiales (ANN) y lógica difusa (FL), entre otras. La investigación en este campo ha demostrado que estas técnicas son capaces de obtener resultados óptimos. Sin embargo, aún no se ha encontrado un framework que pueda evidenciar cuál de estos modelos es el mejor bajo los mismos parámetros de evaluación. Por esta razón, investigadores de diferentes áreas están proponiendo nuevos modelos o combinaciones de técnicas existentes para mejorar la predicción de precios futuros de activos financieros y así tomar decisiones más acertadas. De acuerdo con lo expuesto previamente, en este estudio de investigación se desarrolló un framework que integró tres algoritmos de trading, utilizando los mismos indicadores de rendimiento. A través de esta integración, se logró determinar cuál de estos algoritmos era el más eficiente. Este framework fue programado en Python, y cuenta con una interfaz gráfica de usuario que permite seleccionar el rango temporal de los datos históricos obtenidos de Yahoo Finance para la entrenamiento y validación de los algoritmos. El primer algoritmo utilizó la técnica de redes neuronales convolucionales profundas para reconocer imágenes en escala de grises generadas a partir de indicadores técnicos financieros y determinar si se debía comprar, vender o retener. El segundo algoritmo utilizó aprendizaje profundo por refuerzo con los mismos datos históricos, con el objetivo de mejorar la estrategia mediante la relación Sharpe y generar trayectorias artificiales a partir de un número limitado de datos. El tercer algoritmo implementado se basó en una red neuronal recurrente de tipo LSTM (Long Short-Term Memory). En este algoritmo, se tomaron los precios de cierre históricos como entrada y se analizaron mediante una Transformada de Fourier Discreta (DFT) para descomponerlos en señales de diferentes frecuencias con el objetivo de predecir los precios del stock. Por último, se utilizó la prueba de Friedman para evaluar el desempeño de los tres algoritmos de trading en términos de ganancias durante un periodo de tiempo determinado por el usuario. La prueba mostró diferencias significativas en el desempeño de los algoritmos. Luego, se aplicó el post hoc de Holm para identificar qué pares de algoritmos presentaban estas diferencias significativas. De esta manera, se determinaron cuáles algoritmos tuvieron el mejor desempeño y se tomó una decisión informada sobre cuál algoritmo utilizar. En conclusión, la combinación de la prueba de Friedman y el post hoc de Holm son herramientas valiosas para evaluar algoritmos de trading y tomar decisiones más precisas e informadas.

Dedicatoria

A mis padres, Rodrigo y Mirian, quienes me brindaron mi primera formación, y a mi esposa, Elizabeth, por su inquebrantable apoyo a lo largo de este proceso. También, dedico un especial reconocimiento a mi querida abuelita Herlinda (Q.E.P.D).

Agradecimientos

Al Dr. Carlos Alberto Cobos, quiero expresar mi profundo agradecimiento por su generosidad al brindarme su valioso conocimiento, tiempo y dedicación, los cuales resultaron fundamentales para la culminación de este proyecto de investigación. Además, no puedo dejar de mencionar lo valiosos que fueron sus consejos, los cuales se convirtieron en un faro de luz en medio de los obstáculos y desafíos que surgieron a lo largo de este arduo camino.

Deseo expresar mi más sincero agradecimiento a la Dra. Martha Eliana Mendoza por su invaluable apoyo a lo largo de todo el proceso de investigación, lo que culminó en la creación de un valioso artículo de investigación.

Quiero expresar mi gratitud al Dr. Gabriel Elías Chanchi, quien estuvo siempre atento al proceso de investigación y brindó un valioso respaldo para la culminación de este trabajo de investigación.

Asimismo, deseo expresar mi agradecimiento a todo el grupo de trabajo académico y administrativo de la maestría en computación por proporcionarme el apoyo indispensable para concluir satisfactoriamente este proceso.

Tabla de Contenido

1	Introducción.....	1
1.1	Planteamiento del problema.....	1
1.2	Aportes del proyecto	4
1.3	Objetivos	5
1.3.1	Objetivo general.....	5
1.3.2	Objetivos específicos.....	5
1.4	Resultados obtenidos.....	6
1.5	Estructura de la monografía	6
2	Estado del Arte.....	9
2.1	Introducción.....	9
2.2	Protocolo de Investigación	10
2.3	Resultados obtenidos (Ejecución y Discusión).....	13
2.3.1	Contribuciones basadas en aprendizaje de máquina	13
2.3.2	Contribuciones basadas en metaheurísticas	17
2.3.3	Contribuciones basadas en redes neuronales artificiales.....	20
2.3.4	Contribuciones basadas en lógica difusa	23
2.4	Conclusiones recomendaciones y trabajo futuro.....	24
3	Framework propuesto	27
3.1	Interfaz Gráfica del framework	28
3.2	Datos suministrados al framework	31
3.3	Cálculo de indicadores financieros.....	31
3.3.1	Indicador RSI.....	32
3.3.2	Indicador Williams.....	33
3.3.3	indicador SMA	33
3.3.4	Indicador EMA	33
3.3.5	Indicador WMA	34
3.3.6	Indicador HMA	34
3.3.7	Indicador TEMA	34
3.3.8	Indicador CCI.....	34
3.3.9	Indicador CMO.....	35

3.3.10	Indicador MACD	35
3.3.11	Indicador CMF	36
3.3.12	Indicador DMI	36
3.3.13	Indicador PPO	37
3.3.14	Indicador ROC.....	37
3.3.15	Indicador SAR parabólico.....	37
3.4	Etiquetado del dataset.....	37
3.5	Técnicas de trading algorítmico implementadas al framework.....	38
3.5.1	Redes neuronales convolucionales profundas	40
3.5.2	Aprendizaje por refuerzo profundo	45
3.5.3	LSTM (Long Short-Term Memory).....	54
4	Evaluación y análisis comparativo de los modelos de trading algorítmico	59
4.1	Resultados obtenidos de profit_loss de los algoritmos implementados ...	59
4.1.1	Resultados de la matriz de confusión	62
4.1.2	Test de Friedman.....	64
4.1.3	Pruebas Post Hoc.....	66
5	Conclusiones y trabajo futuro.....	71
6	Bibliografía	73

Índice de Tablas

Tabla 1. Preguntas de investigación	11
Tabla 2. Cadena de búsqueda.....	12
Tabla 3. Clasificación de evaluación en los documentos.....	12
Tabla 4. Estudios basados en aprendizaje de maquina.....	15
Tabla 5. Estudios basados en metaheurísticas.....	19
Tabla 6. Estudios basados en redes neuronales artificiales	22
Tabla 7. Estudios basados en lógica difusa	24
Tabla 8. Datos contenidos en cada activo financiero (fecha, precio de apertura, precio más alto, precio más bajo, precio de cierre, precio de cierre ajustado y número de acciones negociadas por día).....	32
Tabla 9. Indicadores financieros calculados con periodos de 6 a 20 días	32
Tabla 10. Datos contenidos en cada activo financiero (fecha, precio de apertura, precio más alto, precio más bajo, precio de cierre, precio de cierre ajustado, volumen y etiqueta.....	39
Tabla 11. DataFrame de los activos financieros	40
Tabla 12. Resultado de cada algoritmo en su respectiva evaluación	40
Tabla 13. Media aritmética y desviación estándar de cada activo financiero y dataset	40
Tabla 14. Datos en cada archivo CSV con su respectiva evaluación para el algoritmo 1	45
Tabla 15. Datos en cada archivo CSV en su respectiva evaluación para el algoritmo 2	54
Tabla 16. Los nombres de cada activo financiero se encuentran ordenados, desde el dataset 1 hasta el dataset 5.....	55
Tabla 17. Medias móviles de 5,10,15 y 30 días de cada activo financiero con la columna del precio de cierre normalizado.....	55

Tabla 18. Símbolos y definiciones utilizados en la técnica (tomado de [58])	56
Tabla 19. Resultados obtenidos de ganancias o pérdidas porcentuales para el algoritmo 1 (redes neuronales convolucionales) en cada uno de los activos financieros según su dataset.....	61
Tabla 20. Resultados obtenidos de ganancias o pérdidas porcentuales para el algoritmo 2 (aprendizaje por refuerzo profundo) en cada uno de los activos financieros según su dataset.....	61
Tabla 21. Resultados obtenidos de ganancias o pérdidas porcentuales para el algoritmo 3 (red LSTM) en cada uno de los activos financieros según su dataset	62
Tabla 22. Matriz de confusión del algoritmo_1 para el activo financiero “aapl” y también para el dataset_1 que contiene a los activos “aapl” con “^gspc”	63
Tabla 23. Métricas obtenidas a partir de las matrices de confusión del activo financiero “aapl” y también para el dataset_1 que contiene a los activos “aapl” con “^gspc”.....	63
Tabla 24. Métrica F1_score de los 5 dataset para cada uno de los 3 algoritmos (en filas).....	63
Tabla 25. Métrica de precisión de los 5 dataset para cada uno de los 3 algoritmos (en filas).....	63
Tabla 26. Métrica de recall de los 5 dataset para cada uno de los 3 algoritmos (en filas).....	64
Tabla 27. Resultados obtenidos de la prueba de Friedman para cada una de las métricas previamente calculadas	65
Tabla 28. Resultados obtenidos de Ranking Friedman para cada una de las métricas previamente calculadas	65
Tabla 29. Resultados obtenidos de la prueba de Holm para la métrica “profit_loss”	66
Tabla 30. Resultados obtenidos de la prueba de Holm para la métrica “f1_score”	67
Tabla 31. Resultados obtenidos de la prueba de Wilcoxon para la métrica “profit”	68
Tabla 32. Resultados obtenidos de la prueba de Wilcoxon para la métrica “f1_score”	69

Índice de Figuras

Figura 1. Interfaz gráfica del framework para seleccionar el algoritmo a evaluar .	28
Figura 2. Interfaz gráfica del framework para seleccionar las fechas de entrenamiento y validación.....	28
Figura 3. Interfaz gráfica del framework para asignar valores de evaluación y dinero para invertir	29
Figura 4. Interfaz gráfica del framework para asignar parámetros de dos algoritmos y botón de 'Start' para dar inicio	29
Figura 5. Interfaz gráfica del framework para visualizar los resultados obtenidos	30
Figura 6. Interfaz gráfica del framework para evaluar y comparar modelos de pronóstico en series de tiempo financieras	30
Figura 7. Ubicación de activos financieros en formato CSV para entrenamiento y validación	31
Figura 8. Ubicación de los archivos CSV de cada activo financiero debidamente etiquetados en cada día como 'comprar', 'vender' o retener	39
Figura 9. Organización general de cada uno de los algoritmos.....	41
Figura 10. Imágenes generadas utilizando los datos de indicadores financieros previamente calculados con diferentes periodos.....	42
Figura 11. Imágenes de 15x15 pixeles debidamente etiquetadas.....	42
Figura 12. Ubicación de cada imagen en su carpeta correspondiente de comprar, retener o vender	42
Figura 13. Estructura de red neuronal convolucional implementada de la siguiente referencia (Tomada de [44])	43
Figura 14. Ubicación de los modelos generados para cada activo financiero utilizando la API keras.....	44
Figura 15. Ubicación de archivos csv generados por cada iteración de inversión en la línea de tiempo de evaluación para el algoritmo 1	44
Figura 16. Ilustración de la ejecución de una estrategia de trading (Tomada de [54])	45

Figura 17. <i>Bloques básicos fundamentales del aprendizaje por refuerzo (Tomado de [54])</i>	46
Figura 18. <i>Ilustración del algoritmo DQN (Tomada de [54])</i>	50
Figura 19. Ubicación de archivos csv generados por cada iteración de inversión en la línea de tiempo de evaluación para el algoritmo 2	53
Figura 20. Ubicación del archivo csv para las medias móviles calculadas de cada activo financiero	55
Figura 21. <i>Ilustración general de las capas utilizadas en esta última técnica (tomado de [58])</i>	56
Figura 22. Evolución del dinero invertido para el activo financiero “pg” en cada uno de los tres algoritmos implementados	60

Índice de Algoritmos

Algoritmo 1. Cálculo de DMI	36
Algoritmo 2. Etiquetado diario de cada activo financiero (comprar, vender o retener) (Tomado de [44])	38
Algoritmo 3. Estrategia de trading TDQN (Tomado de [54])	52

Índice de Anexos

Anexo 1: Prototipo Software y código fuente de la herramienta desarrollada como prueba de concepto del modelo propuesto disponible en línea en <https://github.com/JesusNavia/financial>.

Anexo 2: Artículo “Trading Algorítmico para la Predicción de Series de Tiempo Financieras: Una Revisión Sistemática” disponible en línea vía Dialnet en <https://dialnet.unirioja.es/servlet/articulo?codigo=8576449> (todo el volumen de la revista) o vía ProQuest en <https://www.proquest.com/openview/a76edd6838c1c4e6e28d63c163a0f301/1?pq-origsite=gscholar&cbl=1006393> (artículo en forma individual).

Esta página se ha dejado intencionalmente en blanco.

CAPÍTULO 1

1 Introducción

1.1 Planteamiento del problema

El trading se conoce como la compraventa de activos financieros (divisas, índices, acciones, bonos, entre otros) en un mercado bursátil, realizado por traders. Los mercados bursátiles han avanzado tecnológicamente a tal punto, que han logrado automatizar los procesos en la toma de decisiones para la compra o venta de estos activos financieros, mediante trading automático o también llamado trading algorítmico [1]. El trading algorítmico tiene ventajas respecto al trading manual ya que no se afecta por las emociones al momento de tomar las decisiones, los algoritmos funcionan como inyectores de liquidez, reducen los spreads (en el mercado de valores, un spread es la diferencia del valor de compra con el valor de venta) y los costos de las transacciones, disminuyendo así el riesgo. Los avances en el trading algoritmo le dan ventajas competitivas a las empresas que lo realizan, tal es el caso de la empresa Renaissance Technologies, la cual obtuvo utilidades por un valor de \$2,5 billones de dólares en un año usando esta tecnología. Esta tecnología es hoy día un tema muy importante de investigación y la transferencia rápida de los mejores algoritmos al mercado es clave para los inversores[2].

Un tema de interés en el mercado bursátil consiste en la predicción de series de tiempo financieras, por esta razón diversos investigadores de las ciencias de la computación han contribuido aplicando técnicas basadas en aprendizaje de máquina, metaheurísticas, redes neuronales artificiales y lógica difusa, entre otras [3]. Por ejemplo, en 2015 [4] se utilizaron algoritmos de bi-agrupamiento y el algoritmo de clasificación K-NN para encontrar patrones en series financieras de índices y acciones de empresas, utilizando datos históricos de Yahoo! Finance (finance.yahoo.com). Luego, en 2016 [5], se utilizaron redes neuronales artificiales para pronosticar el movimiento de los precios del índice S&P500 tomando los datos históricos de Yahoo! Finance e indicadores técnicos financieros. En sus resultados observaron que obtener una mayor precisión en la predicción no garantiza un mayor rendimiento de los retornos. Después, en 2017 [6], proponen una herramienta de pronóstico y negociación de activos financieros con programación genética (genetic expression programming, GEP), obteniendo como resultado una mejora en la precisión del pronóstico y de la utilidad frente a otros modelos del estado del arte, incluso cuando se considera el costo de la transacción. Ya en 2018 [7], se implementó un modelo de red de Petri difusa de alto nivel con ayuda de máquinas de soporte vectorial para tomar decisiones de compra o venta de acciones, con

resultados de precisión en la predicción mayor al 50%. Y más recientemente, en 2019 [8], se integró una representación lineal por partes y una máquina de soporte vectorial ponderada para pronosticar los puntos de inflexión de 20 acciones aleatorias de la Bolsa de Shanghái en China. El modelo propuesto mostró la capacidad de descubrir los puntos de inflexión y obtener ingresos altos y estables en el stock de los mercados. Los resultados experimentales mostraron que los ingresos obtenidos por la propuesta son significativamente más altos que los obtenidos por otros modelos. En 2020 se aplicaron redes neuronales LSTM y árboles de decisión [9] para desarrollar un modelo de predicción basado en una ventana móvil de datos de precios históricos. Este modelo se utilizó para alimentar una estrategia comercial basada en eventos, que considera la distribución de rendimientos previstos en lugar de confiar únicamente en una única predicción de precios; el mejor modelo logró rendimientos acumulados del 350%, 403%, 497% y 333% respectivamente en diferentes índices. En 2021 se empleó el algoritmo Q-Learning [10] para encontrar una estrategia comercial óptima. Se propusieron dos modelos (uno basado en k-means y el otro en velas japonesas) basados en la representación de los estados del entorno. Ambos modelos superaron estrategias basadas en Decision Tree y Buy-and-Hold en términos de retorno de inversión. En este trabajo se consideró el costo de transacción y se limitó el número de operaciones utilizando indicadores de impulso. En 2023 [11] se presenta el modelo BGSA-SVM (Binary Gravity Search Algorithm - Support Vector Machine), para pronosticar series de tiempo financieras con alta dimensionalidad y reducir la pérdida de información. Los datos se preprocesan mediante PCA (Principal Component Analysis) para reducir la dimensionalidad, las entradas y parámetros de los clasificadores SVM se actualizan basándose en el rendimiento y las reglas de iteración de BGSA. La iteración continúa hasta alcanzar un tiempo máximo y el mejor clasificador SVM se utiliza para predecir un conjunto externo con datos recopilados desde 1990 hasta 2016. En comparación con los modelos SVM híbridos, BGSA-SVM destaca por pronosticar series de tiempo financieras con alta dimensionalidad y menos pérdida de información, lo que sugiere un mayor rendimiento. También en 2023 [12], se introduce el algoritmo Growing Neural Gas (GNG) para agrupar datos históricos de negociación de acciones y formar el estado del entorno comercial del mercado de valores utilizando un número limitado de estados, luego rediseñan la función de recompensa para evaluar el comportamiento comercial de la salida del modelo obteniendo un nuevo algoritmo del agente comercial llamado Triple Q-Learning para construir el agente comercial del modelo y representar el comportamiento del entorno del mercado de valores obteniendo excelentes resultados de rentabilidad acumulada; el modelo mostró ser competitivo con las otras estrategias (Decision Tree, K-mean y Q, GNG y Q, Buy and Hold) con diferentes tipos de datos bursátiles, especialmente en términos de rendimiento acumulado.

En la literatura se reportan otros algoritmos que buscan mejorar las estrategias de operación ya existentes, pero actualmente es difícil establecer el o los algoritmos que son el estado del arte en el área de investigación, por diversos motivos, entre ellos:

- 1) El uso de datos privados dificulta la comparación entre los diferentes algoritmos (estrategias/métodos/técnicas) reportados debido a la dificultad de acceso a dichos datos;
- 2) El uso discrecional de partes de los datos de acceso libre y la inexistencia de conjuntos de evaluación y prueba estándar, ya que los investigadores toman de dichas fuentes los subconjuntos que consideran apropiados, los procesan y usan a su manera, y de esta forma no se pueden hacer comparaciones justas con los otros algoritmos previamente reportados;
- 3) Las medidas usadas para la comparación de las propuestas y los trabajos previos no son las mismas y en muchos casos no se usan pruebas estadísticas para establecer rankings entre los resultados reportados por los algoritmos;
- 4) La falta del código fuente de las propuestas presentadas en los artículos, lo cual implica que los autores de nuevas propuestas tengan que implementarlas con lo que se entiende del artículo, hecho que en la mayoría de los casos no garantiza la repetibilidad de los experimentos realizados por los autores originales de las propuestas y además genera un desgaste de tiempo en implementación, desarrollo y experimentación;
- 5) La selección discrecional de los algoritmos con los cuales cada propuesta se compara, de esta forma cada propuesta nueva se compara con trabajos previos que son distintos de los usados por otras propuestas también nuevas, o que comparten solo algunos pocos trabajos previos, y eso agravado en que cada una usa conjuntos de datos diferentes y/o métricas diferentes, lo que dificulta establecer cuáles de estas nuevas propuestas son las mejores.

En este sentido, en este trabajo de grado se planteó como pregunta que guía la investigación: ¿cuál o cuáles de las propuestas reportadas en el estado del arte son las mejores en precisión de pronóstico y rentabilidad para el trading algorítmico? Esta pregunta no se va a resolver con el presente trabajo, pero si se espera dar un paso en la solución de esta, atacando de raíz cuatro (1,3,4 y 5) de los cinco puntos previamente identificados como problemáticos para poder resolver la pregunta en el futuro cercano.

Es así como en el presente trabajo de grado (tesis de maestría) se buscó modelar e implementar en Python un marco de trabajo que facilitará el Backtesting y el análisis comparativo de modelos de trading algorítmico enfocados en la predicción de series de tiempo financieras que está compuesto por cinco (5) conjuntos de evaluación, las medidas de calidad que se pueden extraer de la matriz de confusión, una (1) medida de rentabilidad comúnmente usada en el estado del arte, la prueba estadística no paramétrica de Friedman con el post hoc de Holm para establecer rankings estadísticamente significativos con 90% y 95% de confianza, y tres (3) algoritmos recientemente reportados en el estado del arte (uno basado en redes neuronales convolucionales, otro en aprendizaje por refuerzo profundo, y otro basado en una red LSTM). Finalmente, el marco de evaluación y comparación se usó para realizar el análisis comparativo de los algoritmos implementados y proveyó un ranking de estos, como un aporte de nuevo conocimiento para la comunidad académica y científica del área.

Acotado el alcance de la tesis de maestría, la pregunta específica de investigación que se buscó resolver fue la siguiente: ¿Cuál algoritmo de trading (basado en redes neuronales convolucionales, aprendizaje por refuerzo profundo, basado en una red LSTM) recientemente reportado en el estado del arte e implementado en un marco de trabajo de evaluación y comparación es el mejor en precisión y rentabilidad?

Con la anterior pregunta, la hipótesis principal del trabajo de grado se basa en los reportes actuales de los artículos revisados en el estado del arte y corresponde a:

H1: El algoritmo del estado del arte basado en Q-Learning (aprendizaje por refuerzo) es estadísticamente mejor en rentabilidad que los otros dos algoritmos seleccionados del estado del arte con una significancia del 95%.

Por otro lado, la hipótesis alternativa es:

H0: El algoritmo del estado del arte basado en Q-Learning (aprendizaje por refuerzo) NO es estadísticamente mejor en rentabilidad que los otros dos algoritmos seleccionados del estado del arte con una significancia del 95%.

Se espera que investigaciones futuras usen el marco de trabajo propuesto para evaluar y comparar sus trabajos, enriqueciéndolo con el código fuente de nuevas propuestas y ahorrando tiempo en la evaluación y comparación de sus algoritmos frente al estado del arte.

1.2 Aportes del proyecto

El framework de comparación de algoritmos de trading propuesto aporta un valor significativo para los investigadores en varias tareas, por ejemplo, para evaluar y comparar diferentes estrategias de trading en función de su rendimiento histórico, o para facilitar la optimización de los algoritmos de trading al permitir la calibración de parámetros y la selección de indicadores técnicos de manera más efectiva. En la evaluación de riesgos en el trading, el framework se puede utilizar para analizar y comparar la forma como diferentes algoritmos manejan los riesgos en condiciones cambiantes del mercado, lo que puede ayudar a identificar estrategias más resistentes y seguras. Al permitir la comparación de algoritmos de trading esto permite estudiar y comparar el desempeño de los algoritmos en diferentes mercados financieros, teniendo en cuenta las diferencias en instrumentos comerciales, volatilidad y estructura de mercado. En el ámbito de la inteligencia artificial y el aprendizaje profundo, este framework puede facilitar la comparación y evaluación de algoritmos de trading basados en estos enfoques, lo que resulta relevante en la investigación de inteligencia artificial en finanzas.

Desde la perspectiva de innovación y basado en el conocimiento de los autores, este framework es un primer aporte que estará disponible a la comunidad académica y científica que servirá para que se hagan comparaciones más justas en las nuevas propuestas de algoritmos de trading buscando saber cuál es

el mejor en precisión y rentabilidad frente a los ya incluidos en el framework o los que se vayan agregando a lo largo del tiempo.

1.3 Objetivos

A continuación se presentan los objetivos de la tesis conforme fueron aprobados por parte del Consejo de Facultad de la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca.

1.3.1 Objetivo general

Proponer un framework para la evaluación y el análisis comparativo justo de modelos de pronóstico en series de tiempo financieras usando un enfoque Backtesting.

1.3.2 Objetivos específicos

- Establecer una revisión sistemática de métodos para los modelos de pronóstico en series de tiempo financieras usando las guías propuestas por Barbara Kitchenham y otros en [13] buscando identificar los enfoques más recientes y exitosos de solución en los mercados bursátiles.
- Definir un framework implementado en Python usando como metodología de desarrollo SCRUM para evaluar y comparar modelos de pronóstico compuesto por cuatro conjuntos de evaluación basados en Yahoo! Finance, Thomson Reuters o investing, tres algoritmos del estado del arte con diferentes enfoques (Redes neuronales convolucionales, algoritmos metaheurísticos¹ y aprendizaje por refuerzo) o combinaciones de ellos, las medidas de calidad de la clasificación que se pueden extraer de la matriz de confusión y una medida de la rentabilidad de los algoritmos en la toma de decisiones, y el soporte de la prueba estadística no paramétrica de Friedman y su post hoc de Holm para establecer un ranking entre los algoritmos evaluados.
- Realizar el análisis comparativo de los algoritmos del estado del arte implementados usando los conjuntos de evaluación, las medidas de calidad del framework previamente construido y las pruebas estadísticas no paramétricas para establecer un ranking de los resultados obtenidos por los algoritmos y la dominancia entre estos.

¹ La opción metaheurística se cambio por una basada en una red LSTM teniendo en cuenta los buenos resultados reportado y el detalle de la publicación, lo que permitió su repetibilidad en su implementación en el framework.

1.4 Resultados obtenidos

A continuación, se presentan los principales resultados del trabajo realizado:

- **Monografía de la investigación**, que corresponde al presente documento donde se resume la investigación llevada a cabo, abordando de manera concisa el problema estudiado, los objetivos planteados para su resolución, los productos obtenidos, como el framework para la evaluación de algoritmos de trading, así como la evaluación realizada sobre el trabajo desarrollado. Además, se presentan las conclusiones obtenidas y se proponen ideas de trabajo futuro que resultan de particular interés para desarrollar en el área de investigación.
- **Framework** propuesto que incluye la implementación de tres algoritmos de trading, junto con su documentación y código fuente.
- **Un artículo** que corresponde a una revisión de la literatura, el cual fue publicado en una revista categoría B del PUBLINDEX de Minciencias con la siguiente referencia: Jesús-Rodrigo Navia-Rodríguez, Carlos-Alberto Cobos-Lozada, Martha-Eliana Mendoza-Becerra. Trading Algorítmico para la Predicción de Series de Tiempo Financieras: Una Revisión Sistemática. RISTI - Revista Ibérica de Sistemas e Tecnologías de Informação, E38, 337-357. Diciembre 2020, ISSN: 1646-9895.

1.5 Estructura de la monografía

A continuación, se describe de manera general el contenido y organización de la presente monografía:

CAPITULO 1: INTRODUCCIÓN: Hace referencia al presente capítulo que introduce el tema de investigación, presenta la pregunta de investigación que originó el trabajo, los aportes al problema, también los objetivos (general y específicos) definidos para el proyecto, un breve resumen de los resultados obtenidos y finalmente la organización de la monografía.

CAPITULO 2: ESTADO DEL ARTE: Este capítulo presenta el estado del arte de los trabajos relacionadas con los algoritmos de trading para la predicción de series de tiempo financieras entre 2015 y 2019. De esta revisión se muestran los pasos seguidos y el resultado obtenido, junto con las conclusiones de esta.

CAPITULO 3: FRAMEWORK PROPUESTO: En este capítulo se presenta el framework propuesto en esta investigación, el diseño general y las interfaces más importantes junto con la descripción de los tres algoritmos de trading implementados en el framework y la explicación detallada de cada uno de estos.

CAPITULO 4: EVALUACIÓN Y ANÁLISIS COMPARATIVO DE LOS MODELOS DE TRAINING ALGORÍTMICO: En este capítulo se muestran los resultados de la evaluación de los algoritmos de trading, utilizando medidas de calidad de clasificación obtenidas de la matriz de confusión. Además, se analiza la rentabilidad

de los algoritmos que es de especial interés para la toma de decisiones. Usando la prueba estadística no paramétrica de Friedman y su post hoc de Holm se muestra un ranking entre los algoritmos.

CAPITULO 6: CONCLUSIONES Y TRABAJO FUTURO: En este capítulo se presentan las conclusiones obtenidas al finalizar el trabajo de grado e ideas que el grupo de investigación y terceros pueden realizar como trabajo futuro en la temática de interés de esta investigación.

CAPITULO 7: BIBLIOGRAFIA: Este último capítulo lista las referencias bibliográficas de los artículos consultados para la realización de la tesis.

Esta página se ha dejado intencionalmente en blanco.

CAPÍTULO 2

2 Estado del Arte

Para definir el estado del arte se realizó una Revisión Sistemática de la Literatura (RSL) del tema en cuestión siguiendo la guía propuesta en [13]. Esta revisión se postuló y después de su evaluación por pares internacionales fue aceptada como ponencia en el evento denominado JIISIC 2020 – XV Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento, que se realizó virtualmente del 19 al 21 de agosto de 2020. Finalmente, los organizadores del evento definieron que la publicación del artículo se realizará en una revista categoría B (SJR Q3) del PUBLINDEX de Minciencias, con la siguiente referencia: Jesús-Rodrigo Navia-Rodríguez, Carlos-Alberto Cobos-Lozada, Martha-Eliana Mendoza-Becerra. Trading Algorítmico para la Predicción de Series de Tiempo Financieras: Una Revisión Sistemática. RISTI - Revista Ibérica de Sistemas e Tecnologías de Informação, E38, 337-357. Diciembre 2020, ISSN: 1646-9895. A continuación, se presenta el resultado de este trabajo. Luego de realizada esta RSL se han realizado nuevos hallazgos relacionados con el modelo propuesto y cada uno de sus servicios, estos hallazgos se indican al inicio de la explicación de cada servicio en el siguiente capítulo.

2.1 Introducción

Trading es la palabra en inglés usada como la acción de comprar y vender activos financieros (divisas, índices, acciones, bonos, entre otros) en un mercado bursátil. Es realizado por traders, personas que compran y venden dichos activos financieros. Los mercados bursátiles han avanzado tecnológicamente a tal punto, que han logrado automatizar la toma de decisiones para la compra o venta de estos activos financieros, mediante el trading automático o también llamado trading algorítmico [1]. El trading algorítmico tiene ventajas respecto al trading manual realizado por las personas, ya que no se afecta por las emociones al momento de tomar las decisiones. Los algoritmos funcionan como inyectores de liquidez, reducen los spreads (en el mercado de valores, un spread es la diferencia del valor de compra con el de venta) y los costos de las transacciones, disminuyendo así el riesgo. Los avances en el trading algoritmo le dan ventajas competitivas a las empresas que lo realizan, tal es el caso de la empresa Renaissance Technologies, la cual obtuvo utilidades por un valor de \$2,5 billones de dólares en un año usando esta tecnología. Esta tecnología es hoy día un tema muy importante de investigación y la transferencia y adopción rápida de los mejores algoritmos es clave para los inversores [2].

En el mercado bursátil es muy importante predecir los valores de los activos financieros en el tiempo (predicción de series de tiempo financieras), por esta razón, diversos investigadores de las ciencias de la computación y otros campos han contribuido aplicando técnicas basadas en aprendizaje de máquina, metaheurísticas, redes neuronales artificiales, lógica difusa, entre otras [3].

El presente documento muestra una revisión de 43 artículos primarios reportados en Web of Science relacionados con el trading algorítmico en la predicción de activos financieros. Para el análisis de estos documentos se realiza una agrupación con base en la técnica que en cada artículo reporta los mejores resultados, siendo estos: aprendizaje de máquina, metaheurísticas, redes neuronales y lógica difusa. Para cada artículo se presenta además de una corta descripción, los datos, las fuentes de esos datos, los algoritmos con los que se hace la comparación de la técnica que se propone y las métricas usadas en la comparación.

El resto de este documento está organizado así: en la Sección 2 se describe el protocolo de investigación llevado a cabo para realizar el estado del arte. La Sección 3 muestra los resultados obtenidos de dicha revisión y la discusión de las observaciones principales de estos resultados, así como las limitaciones e implicaciones de la revisión en el campo de investigación. Y, finalmente, en la Sección 4 se presentan las conclusiones y el trabajo futuro que espera realizar el grupo de investigación en el área.

2.2 Protocolo de Investigación

El presente estado del arte se ha llevado a cabo siguiendo las guías provistas en [14], realizando tres etapas principales: Planificación, Ejecución y Discusión.

En la **Etapas de Planificación** se llevaron a cabo las siguientes actividades: 1) Definición de la pregunta de investigación, 2) Definición de la estrategia de búsqueda, 3) Establecimiento de los criterios de selección de los estudios primarios, 4) Establecimiento de los criterios de evaluación de calidad, 5) Definición de la estrategia de extracción de datos, y 6) Selección de los métodos de síntesis.

Preguntas de Investigación: Esta investigación tiene como objetivo principal identificar las técnicas, métodos, modelos o similar, utilizadas en el trading algorítmico para la predicción de series de tiempo financieras. Para ello se han establecido las preguntas de investigación que se presentan en la **Tabla 1**. Con estas preguntas se recopila y categoriza la información existente, con el fin de realizar una revisión sistemática del tema, identificando los caminos prometedores y las mejores propuestas existentes a la fecha.

Para realizar la búsqueda de los artículos se utilizó la base de datos de Web of Science. La **Tabla 2** muestra la cadena de búsqueda final, la cual se obtuvo de un proceso iterativo de mejora con el análisis de los resultados reportados por cada versión de esta. La metodología utilizada para la construcción de la cadena de búsqueda se basó en PICOT (población-intervención-comparación-logro-tiempo)

[15], estructurando los términos por población, intervención, comparación, resultados y tiempo (PICOT en inglés). La cadena se presenta dividida según PICOT y se unen con el operador AND. La búsqueda se realizó sobre el título, el resumen y las palabras claves de los artículos en la base de datos.

Tabla 1. Preguntas de investigación

Preguntas de Investigación	Motivación
Q1. ¿Cómo se ha ampliado el conocimiento o la comprensión del tema con la propuesta?	Busca determinar si un modelo de trading algorítmico es nuevo o combina de una forma novedosa elementos existentes que permiten identificar hacia donde se desarrolla la investigación en el tema.
Q2. ¿Qué tan claro y consistente es el documento?	Identificar si la solución presentada es clara, fácil de comprender y muestra consistencia en cada uno de los componentes propuestos.
Q3. ¿Qué nivel de correlación se encuentra entre los resultados experimentales, la interpretación de estos y las conclusiones, es decir, qué nivel de soporte tiene cada conclusión?	Determinar si las conclusiones tienen su soporte en los resultados del experimento y que la interpretación de estos resultados no está sesgada.
Q4. ¿Qué tan bien apoyada es la evaluación de la propuesta y que tan creíbles son los resultados?	Determinar las métricas de evaluación de mayor uso en el trading algorítmico e identificar qué tan creíbles son los resultados.
Q5. ¿Cuán explícita y repetible es la implementación de la propuesta de acuerdo con sus objetivos y requisitos?	Identificar las herramientas de implementación en los modelos de trading algorítmico y definir si la descripción es lo suficientemente detallada para facilitar la repetición de la propuesta y experimentos.
Q6. ¿Cuán detallado es el proceso de recopilación de datos, sus características y disponibilidad?	Identificar los activos financieros analizados con sus respectivas fuentes y características de análisis.
Q7. ¿Cuán completo y bien documentado ha sido el proceso de investigación?	Determinar si la propuesta ha sido bien realizada desde el punto de vista metodológico.

Criterios de Selección de los Estudios Primarios: Con el objetivo de definir cuáles de los resultados entregados por el motor de búsqueda se incluyen en la revisión, se realizaron y aplicaron unos criterios de aceptación basado en la información en el título, resumen y palabras clave. Los criterios de inclusión fueron los siguientes: 1) Artículos escritos en español o inglés, 2) Estudios que muestren la toma de decisiones en mercados financieros y 3) Propuestas de trading algorítmico.

Luego, se eliminaron los documentos que cumplieron con alguno de los siguientes criterios de exclusión: 1) El estudio utiliza portafolios de inversión, 2) El estudio utiliza técnicas de análisis de sentimientos u otras fuentes de datos relacionadas con noticias y similares, y 3) El estudio aplica procedimientos de trading algorítmico de alta frecuencia. Luego de aplicar los criterios de selección y dejar los documentos base para el estudio, se procedió a su evaluación.

Criterios de Evaluación de Calidad: como es común en las revisiones sistemáticas [10] se aplicaron unos criterios para evaluar la calidad de los estudios primarios recolectados. Esta evaluación es necesaria para comprender mejor las posibles comparaciones entre los estudios y para guiar la interpretación de los hallazgos.

Para esto se desarrolló una prueba con cada una de las preguntas de investigación presentadas en la **Tabla 2** y un sistema de puntuación de tres valores (0, 0.5, 1). En general, el valor de cero (0) significa que el artículo no aporta nada en relación con la pregunta de investigación o no cumple con lo que se pregunta; el valor de cero cinco (0.5) significa que el artículo realiza un aporte parcial a resolver la pregunta de investigación o cumple parcialmente con lo que pregunta; y finalmente el valor de uno (1) significa que hace un aporte muy completo a resolver la pregunta de investigación o cumple totalmente con lo que se pregunta. Como son 7 preguntas, los trabajos primarios tendrán un valor mínimo de 0 y máximo de 7.

Tabla 2. Cadena de búsqueda

Estructura	Términos Alternativos y Sinónimos	Descripción
Población	("financiera" AND "trading" AND "stock")	Define la temática general que se está buscando: financiera, negociación y stock.
Población	NOT ("high frequency" AND portfolio)	Se excluyen las temáticas: alta frecuencia y portafolios.
Intervención	("time series" OR "alpha" OR "buy" OR "sell" OR "no trade")	Se define qué actividad abordara: Series de tiempo, tipos de negociación, compra, venta y comercio
Tiempo	PUBYEAR > 2014 AND DOCTYPE ("ar" OR "ip")	Se delimita el año y tipo de documento: mayores a 2014, artículo o artículo "in press".

Extracción de Datos: Para la clasificación de los documentos se utilizaron los datos extraídos del texto completo de estos junto con las respuestas encontradas a cada una de las preguntas de investigación previamente presentadas. Las preguntas de investigación, como sus posibles respuestas se muestran en la **Tabla 3**.

Tabla 3. Clasificación de evaluación en los documentos

Preguntas	Respuestas
Q1	Si , presenta una nueva propuesta que transforma o modifica los modelos existentes de trading algorítmico por lo que genera nuevo conocimiento. Parcialmente , presenta modelos de trading ya existentes con un aporte al nuevo conocimiento de forma limitada. No , utiliza modelos o datos de trading diferentes a los usados en otros reportes.
Q2	La información del artículo Si es relevante en los modelos de trading algorítmico. La información del artículo es Parcialmente relevante en los modelos de trading algorítmico. La información del artículo No es relevante en los modelos de trading algorítmico.
Q3	Las conclusiones del modelo propuesto Si tiene un soporte cuantitativo. Las conclusiones del modelo propuesto exponen Parcialmente el soporte cuantitativo. Las conclusiones del modelo No tienen un soporte cuantitativo.
Q4	Si presenta las métricas de evaluación en el trading algorítmico para identificar qué tan creíbles son los resultados. Parcialmente presenta las métricas de evaluación en el trading algorítmico ... No presenta las métricas de evaluación en el trading algorítmico ...
Q5	Si presentan las herramientas de implementación de los modelos de trading algorítmico lo suficientemente detallada para facilitar la repetición de la propuesta y experimentos. Parcialmente presentan las herramientas de implementación ... No presentan las herramientas de implementación ...
Q6	Si presenta los activos financieros analizados con sus respectivas fuentes y características de análisis. Parcialmente presenta los activos financieros ... No presenta los activos financieros ...
Q7	Si tiene estructura consistente para que transmita el conocimiento del modelo de trading algorítmico investigado. Parcialmente tiene una estructura consistente ... No tiene la estructura consistente ...

Métodos de Síntesis: En la presente revisión se aplicó una síntesis de datos cuantitativos y cualitativos basada en una tabla que resumen los estudios seleccionados ordenados según el año, enfoque, técnicas, referencia, criterio de calidad y datos con su respectiva fuente del activo financiero investigado. Además, un párrafo que resume cada artículo organizado por los tipos de técnicas usadas.

Calendario de la revisión sistemática: La presente revisión sistemática inició la recolección de datos en agosto de 2019 y finalizó en septiembre del mismo año.

2.3 Resultados obtenidos (Ejecución y Discusión)

Una vez definido el protocolo de la revisión de literatura, se procedió con la etapa de ejecución, esta misma fue guiada mediante las siguientes fases principales: 1) Se aplicó la cadena de búsqueda (**Tabla 2**) en la base de datos científica de Web of Science y se obtuvieron 206 documentos diferentes, 2) Con estos documentos se procedió a aplicar los criterios de selección (inclusión/exclusión) sobre el título, resumen y palabras claves de cada uno de ellos, una vez finalizada esta segunda fase, se obtuvo un total de 43 estudios relevantes. 3) Durante la tercera fase, se llevó a cabo la aplicación de los criterios de evaluación de calidad sobre cada uno de los 43 estudios relevantes, pero en este caso leyendo el documento completo. Las siguientes subsecciones presentan los resultados de la evaluación de los estudios de acuerdo con los criterios de evaluación.

Los 43 estudios relevantes se organizaron de acuerdo con la técnica utilizada, a saber: aprendizaje de máquina (Machine Learning, ML) (ver **Tabla 4**), metaheurísticos (Metaheuristics, MH) (ver **Tabla 5**), redes neuronales artificiales (Artificial Neural Networks, ANN) (ver **Tabla 6**) y lógica difusa (Fuzzy Logic, FL) (ver **Tabla 7**). Es preciso comentar que los estudios relacionados con redes neuronales artificiales se separan de los estudios de aprendizaje de máquina, dada la evolución e importancia que esta área de investigación ha tenido en los últimos años en el trading algorítmico. A continuación, se describen cada uno.

2.3.1 Contribuciones basadas en aprendizaje de máquina

En 2015 se utilizó biclustering y clasificación con los k vecinos más cercanos (Biclustering and k-Nearest Neighbor, BIC-KNN) para encontrar patrones en series financieras de índices y acciones de empresas, obtenidas de Yahoo! Finance. Los algoritmos se implementaron en MATLAB obteniendo ganancias mayores al 39%. Este es el primer reporte del uso de biclustering para obtener patrones con datos financieros [4].

En 2016 se utilizó K-means con descomposición por valores singulares (Singular Value Decomposition, SVD) en una propuesta denominada K-SVD, para realizar predicciones en series de tiempo financieras con datos de Yahoo! Finance del mercado de valores de EE.UU y de esta forma se obtuvieron utilidades mayores a las que ofrece el mercado [16]. También en 2016 se planteó que el uso de la hipótesis de mercado eficiente (Efficient Market Hypothesis, EMH) no permite

predecir los retornos de sus activos. Para lograr demostrar esto, se evaluó el uso de máquinas de soporte vectorial vs redes neuronales artificiales (SVM vs ANN) y las máquinas de soporte vectorial vs el análisis de regresión (SVM vs AR), encontrando que el modelo SVM obtuvo el mejor desempeño [17].

En 2017 propusieron un modelo que utiliza una máquina de aprendizaje extrema sin supervisión (UnSupervised-Extreme Learning Machine, US-ELM) y la Regresión por Soporte Vectorial (Support Vector Regression, SVR), para predecir el índice de futuros de materia primas ENERGY con datos históricos obtenidos de <http://www.mcxindia.com>. Los autores afirman que la precisión de la predicción de la propuesta es mejor que los modelos de referencia [18]. También en 2017 se propuso un método para predecir las tendencias alcistas o bajista de series de tiempo financieras con el algoritmo MKK (modelo construido de los k valores alcistas o bajistas del histórico conocido) en activos financieros, utilizando los datos extraídos de Yahoo! Finance. La propuesta comparó el valor observado y encontró patrones de tendencia, donde en promedio las tasas de predicción del algoritmo MKK fue de 89.8%, y utilizando el indicador técnico bursátil MACD (Moving Average Convergence Divergence) fue tan sólo del 33.4% [19]. Este mismo año se utilizaron mapas auto organizados (Self-Organizing Maps, SOM) para generar opciones de compra o venta de activos financieros, de datos previamente preprocesados de Yahoo! Finance utilizando un filtro Hodrick y Prescott. Como resultado se obtuvieron ganancias mayores respecto a propuestas similares [20]. Siguiendo con el 2017, se realizó una propuesta para pronosticar precios europeos de emisiones de carbono con SVM, K-NN y árboles de decisión (Decision Trees, DT), usando Matlab R2015 con Statistic y Machine Learning Toolbox v10.0. Como resultado de la investigación se logró una tasa de precisión para todos los clasificadores superior al 48% [21]. Terminando el 2017, se presentó un método para predecir series de tiempo de los índices bursátiles S&P500, FTSE 100 y Nikkei 225, utilizando el modelo SVR desarrollado en Matlab con ayuda de LibSVM versión 2.91. El método inicializa los parámetros obtenidos de Yahoo! Finance optimizados por enjambre de partículas (Particle Swarm Optimization, PSO), luego entrena el SVR con los parámetros generados para realizar la predicción con modelos estadísticos guiados por error (MSE, MAPE, DS) y modelos guiados por ganancias (Naive, Filter, DMA), obteniendo como mejor resultado a SVR_{MSE} según la prueba de Friedman. [22].

En 2018, proponen predecir índices del mercado financiero con un modelo híbrido que utiliza multi-categoría ponderada (WM) usando máquinas de soporte vectorial (SVM), programación genética (GEP), Random Forest, (RF) e indicadores técnicos para formar la propuesta llamada RF-WMGEP-SVM. Utiliza como entrada los datos históricos de los índices recogidos de Quandl.com. Como conclusión se observó que los modelos híbridos funcionan mejor que sus versiones que no lo son [23]. Este mismo año (2018), proponen diferentes árboles de decisión: arboles de clasificación (CT), arboles de regresión (RT), árboles de clasificación fija (FCT) y árboles de regresión fija (FRT), para predecir el índice bursátil S&P500 utilizando datos obtenidos de Thomson-Reuters y Eikon. Utilizan procesos de Markov binarios que ayudan a ser flexible la captura de patrones arbitrarios y como resultado el modelo FCT tiene mejor relación el indicador técnico bursátil Sharpe (que mide la

relación del riesgo con la rentabilidad), igual a 0,96 con un rendimiento anual compuesto del 7,87%, y un nivel de confianza en las estrategias mayor al 95%, que fue el mejor resultado obtenido por las cuatro técnicas compradas [24]. También en 2018, se planteó un método que combina el análisis de componentes principales (Principal Component Analysis, PCA) y máquinas de soporte vectorial ponderado (Weighted Support Vector Machines, WSVM) para predecir puntos de negociación en acciones, utilizando indicadores técnicos bursátiles como variables de entrada según los datos históricos de las acciones obtenidos en Yahoo! Finance. Como resultado, la propuesta PCA-WSVM obtuvo una precisión máxima del 57% y una rentabilidad del 222,80%, resultados superiores a los obtenidos con PCA y WSVM por separado [25].

En 2019 utilizan la transformada wavelet discreta para crear una nueva base de datos extendida mediante filtros pasa baja y pasa alta. Así estos nuevos datos se utilizan por una regresión por soporte vectorial (SVR), para predecir el precio del oro. La combinación de estas técnicas utiliza los datos mensuales desde mayo de 1972 a diciembre de 2016 disponibles en <https://fred.stlouisfed.org> y para generar la serie de tiempo original con un resultado que aumenta la riqueza de dos tercios en el periodo de predicción [26]. También en 2019 utilizan el modelo de clasificación Random Forest con un filtro de similitud difusa y entropía (FSAE) para predecir el índice bursátil S&P500 basado en cuatro configuraciones de FSAE con datos obtenidos de Yahoo! Finance. La mejor configuración del filtro obtuvo una presión del 44.78% seguida por las configuraciones restantes con 44,38%, 43,06% y 42,91%, además el modelo Random Forest obtiene una precisión del 44,72% superando a los modelos KNN, Naive Bayes y árbol de decisión [27]. En el mismo año, 2019, proponen un bosque aleatorio adaptativo de conceptos recurrentes (Recurring Concepts Adaptive Random Forest, RCARF), para predecir el índice bursátil S&P500 con 10 indicadores técnicos bursátiles (Simple n-second moving average, Weighted n-second moving Average, Momentum, Stochastic K%, Stochastic D%, RSI, MACD, Larry William's R%, A/D y CCI), calculados con ayuda de la biblioteca para análisis técnico TA-Lib con datos de quantquote. La implementación RCARF y los modelo de comparación RCD, DWM, AHOEFT y ARF se realizaron en el framework MOA con una evaluación Interleaved-Test-Then-Train, obteniendo como resultado un error máximo en precisión de 34,7791% para RCARF, superando a los demás [28]. Finalmente, en 2019, utilizan el modelo híbrido de representación lineal por partes con máquina de vectores de soporte ponderado (PLR-WSVM), apoyado en el indicador técnico bursátil RSI para determinar si los puntos de inflexión en la serie tiempo son de compra o venta en 20 activos financieros extraídos de english.sse.com.cn. El modelo propuesto obtuvo un 43,74% de rentabilidad superando a los demás modelos comparados (PLR-WSVM, IPLR-WSVM, PLR-ANN, BHS) [8].

Tabla 4. Estudios basados en aprendizaje de máquina

Framework para la evaluación y análisis comparativo de modelos de pronóstico en series de tiempo financieras

Año	Referencia	Técnica	Técnicas	Datos fuente	Criterio de Calidad							Puntaje
					C1	C2	C3	C4	C5	C6	C7	
2015	[4]	ML	Biclustering, k-nn	S&P500 finance.yahoo.com	1	1	1	1	1	0,50	5	6,0
2016	[16]	ML	k-means, SVD	AA, AMAT finance.yahoo.com	1	1	0,5	1	0,50	0,51	5	5,5
2016	[17]	ML	SVM, ANN	S&P 500, SP/TSX, Nikkei 225, KOSPI 200, Hang Seng, traits Times Index, Shanghai SE, FTSE Bursa Malaysia KLCI, MAI security, JKSE, CAC 40, FTSE 100, FTSE MIB, DAX, BUX, FTSE/JSE top 40, ISE-100, Swiss Market Index, IBEX 35, AEX, BEL20, PSI-20, OMX ALLSHARE Stockholm, OSE All Share Index, OMX Copenhagen, OMXH25, ATX, prague stock exchange index, OMX Vilnius Index, OMX Tallinn Index, OMX Riga Index, INDEXDJX, NASDAQ-100, Brazilian Bovespa Futures tickdata.com	0,5	1	1	0,50	0,51	5	5,5	5,5
2017	[18]	ML	US-ELM, SVR	MCX mcxindia.com	1	1	1	1	0,50	0,51	6	6,0
2017	[19]	ML	MKK, MKK prior MKK-MACD	S&P 500, Amazon, Apple, Bank of America, Citi Group, Coca Cola, Disney, Ford, GE, Microsoft, Nike finance.yahoo.com	1	1	1	1	0,50	0,51	6	6,0
2017	[20]	ML	SOM Clustering	ACC, GAIL, ICICI, NTPC, TE, BoA, BA, GE, M3, PEP, BT, CCL, GSK, RDS, SBRY, DTEX, EMBR, ITSA4, PETR, TFA finance.yahoo.com	0,5	1	0,5	1	0,55	5	5,5	5,5
2017	[21]	ML	SVM, k-nn, DT	Brent crude, GN, FTSE100, DAX30, CAC40, S&P500 Investing.com, finance.yahoo.com, bloomberg.com	0,5	1	1	0,50	0,50	0,55	5	5,0
2017	[22]	ML	SVR, PSO	S&P 500, FTSE 100, Nikkei 225 finance.yahoo.com	0,5	1	1	1	1	1	6	6,5
2018	[23]	ML, MH	WM, SVM, GEP,RS, BUY-HOLD, RF-PSVM, RF-BMP SVM, RF-LST SVM, RF-WMGEPSVM	NASDAQ, DOWJONES, S&P 500, NIFTY 50 y NIFTY BANK quandl.com	1	1	1	0,50	0,51	0,55	5	5,5
2018	[24]	ML	B&H, FCT, FRT, AR, RTMSE, CTGini	S&P 500 thomsonreuters.com	0,5	1	0,50	0,50	0,50	0,50	5	4,0
2018	[25]	ML	PCA-WSVM, WSVM, PCA-ANN, BHS	002617.SZ, 002314.SZ, 002149.SZ finance.yahoo.com	0,5	1	1	0,50	0,51	5	5,5	5,5
2019	[26]	ML	OLS, OLS EW, LASSO, BOOST, FOREST, SVR, DWT-SVR, Buy & hold	Gold, cpi, twex, tbod, tms, sp500.gsci, crb, IMF-IFS foreign exchange reserves Adjusted realized volatility of gold fred.stlouisfed.org	0,50	0,51	1	1	0,50	0,55	5	5,0
2019	[27]	ML	RF	S&P500 finance.yahoo.com	0,50	0,51	1	0,51	0,55	5	5,0	5,0
2019	[28]	ML	RCARF, ARF, RCD, DWM, AHOEFT	SPDR, S&P 500 quantquote.com	1	1	0,50	0,50	0,50	0,51	5	5,0

2019[8]	ML	PLR- WSVM, IPLR-WSVM, PLR-ANN, BHS	600736, 600197, 600211, 600694, 600351, 600488, 600054, 600019, 600058, 600682, 600687, 600066, 600881, 600228, 600697, 600107, 600053, 600051, 600163, 600167 english.sse.com.cn	0,51	1	0,50	50,50	50,50	54,5
---------	----	------------------------------------	---	------	---	------	-------	-------	------

2.3.2 Contribuciones basadas en metaheurísticas

En 2015, proponen una estrategia universal de negociación utilizando espacios de Hilbert con kernel reproducible (RKHSs) para predecir datos históricos (de los activos financieros: SIBN, GAZP, LKOH, MTSI, ROSN, SBER) en la bolsa de valores de Rusia y la bolsa de valores electrónica BATS (Bats Global Markets) de US, obtenidos de www.finam.ru. La técnica fue afinada con los algoritmos de Foster-Vohra y Kakade-Foster. La propuesta obtuvo una rentabilidad considerable ante otras técnicas de baja complejidad sin mencionarlas [29]. Este mismo año 2015, se identifican estudios en computación inteligente aplicados a la construcción de operaciones comerciales con pares de activos financieros y como propuesta de investigación utilizan un enfoque general para más de dos acciones como grupo de inversión, empleando algoritmos genéticos (Genetic Algorithms, GA) que optimizan la media móvil, Bandas de Bollinger y un coeficiente de ponderación del stock. La anterior propuesta usa datos de la bolsa de Taiwán obteniendo una precisión en las predicciones superiores al 0,7 con las mayores capitalizaciones del mercado, superando al modelo de regresión lineal [30].

En 2016, utilizan máquinas de soporte vectorial evolutivo (ESVM) para predecir la dirección de los índices bursátiles ASE20, Dax30, Nikkei225 y S&P 500 extrayendo históricos de bloomberg.com donde se comparan con estrategias comerciales Buy&Hold, Naive, el indicador técnico MACD, modelo auto regresivo ARMA y un perceptrón multicapa (MLP). Como resultado el modelo híbrido ESVM obtuvo un 70,29% de rentabilidad anual excluida del costo, seguida de la estrategia MPL que fue de 42,19% [31]. En el mismo año (2016), se propone un modelo híbrido basado en series de tiempo difusas de múltiples órdenes y un análisis técnico con un algoritmo genético, todo desarrollado en C++ para predecir series de tiempo de índices bursátiles (TAIEX, Dow Jones, NASDAQ, HSI, S&P500) y el par de divisas NTD/USD (dólar taiwanés/dólar) con datos históricos extraídos de twse.com.tw/zh/, cbc.gov.tw. La propuesta es superior en la tasa de precisión direccional entre un 1% al 3% respecto a las series de tiempo difusas con el algoritmo genético (FTSGA) y con un algoritmo de colonia de hormigas con regresión automática (ACO-AR) [32].

En el año 2017, se estudió la fluctuación dinámica del movimiento (la correlación entre volatilidades pasadas y rendimientos futuros) de precios en 20 índices bursátiles (MERV, S&P500, AXJO, BFX, BVSP, GSPTSE, IPSA, SCI, FTSE, FCHI, DAX, HIS, BSESN, JKSE, N225, KOSPI, KLSE, MXX, NZ50, TWII) con datos de Yahoo! Finance para ser procesados en MatlabR2012a y obtener una estrategia de efecto impulsado por la fluctuación (FDE). La propuesta FDE obtuvo un retorno acumulado de un 70% mientras tanto las propuestas basadas en indicadores técnicos bursátiles (RSI, MACD, Momentum) y la estrategia B&H obtuvieron como

máximo un 40% [33]. Continuando con el año 2017, proponen un modelo PAA-MS-IDPSO-V (aproximación agregada por partes usando optimización auto adaptativa por Multi enjambre mejorada de partículas con validación) para obtener patrones en las series de tiempo del índice bursátil S&P100 extraídas de Yahoo! Finance. La propuesta PAA-MS-IDPSO-V obtiene mejores resultados que las variaciones de la propuesta PAA-IDPSO, PAA-IDPSO-V y PAA-MS-IDPSO en los retornos de inversión según la prueba no paramétrica de Friedman y la prueba post-hoc de Nemenyi con un nivel de confianza del 95% [34]. Finalizando el 2017 proponen una herramienta en java para el pronóstico y la negociación de activos con programación genética usando los datos extraídos de la bolsa de valores de Londres, Nueva York y Frankfurt. La propuesta supera en precisión de pronóstico y utilidad a los modelos basados en una caminata aleatoria (Random Walks o Random Search, RS), una comparación poco informativa de la calidad de la propuesta [6].

En el 2018, proponen negociar futuros del índice KOSPI200 con un sistema de comercio de coincidencias de patrones (PMTS) que utiliza el algoritmo envolvente de tiempo dinámico (DTW) en las series de tiempo extraídas de data.koscom.co.kr. La propuesta que obtuvo la mejor relación del indicador técnico bursátil Sharpe con 0,94 según los parámetros de costo de deslizamiento de 0.02 pt, una relación de stop-loss del 0.5%, un período de entrenamiento de 18 meses, un período de prueba de 3 meses, 20 criterios de filtrado ea y 65% de frecuencia U/D [35]. En ese mismo año, 2018, se utilizó programación evolutiva para predecir series de tiempo en activos financieros (FTSE MIB, PSI-20, ISEQ 20, FTSE/Athens Large Cap, IBEX 35) obtenidos de borsaitaliana.it, bolsamadrid.es, bolsadelisboa.com, ise.ie y athexgroup.gr. Para el algoritmo evolutivo se estima y ajusta los parámetros con modelos STAR (ESTAR, LSTAR, TSTAR, GBELL-STAR) y la métrica Diebold-Mariano. Como resultado quien obtuvo la mejor relación del indicador técnico bursátil sharpe ratio fue LSTAR con un valor de 0.72410 dado en el índice PSI-20 superando a los demás modelos [36]. Continuando con el 2018, proponen un sistema que utiliza programación genética, optimización multiobjetivo, análisis técnico y una sección donde determina cuando comprar o vender el activo financiero, para series temporales de BBAS3, BOVA11, CMIG4, EMBR3, GBR4 y VALE5 obtenidas de bb.com.br, bmfbovespa.com.br, cemig.com.br, embraer.com, gerdau.com y vale.com. La propuesta obtuvo hasta un 77,61% en retornos en comparación a la segunda mejor estrategia B&H con un 46,57% en retornos, seguidas de las estrategias TA-10 y TA-36 [37]. Siguiendo en 2018, utilizan el algoritmo de biclustering para encontrar patrones de negociación con los indicadores técnicos bursátiles (SMA, RSI, %R, ROC, CCI, EMV, UO, BRAR, KJD y MTM) de activos financieros (Apple, BA, CAT, JNJ, XOM, VZ y S&P500) obtenidos de la bolsa de valores de EE. UU, posteriormente aplican el algoritmo Naïve Bayes para determinan la tendencia de las series de tiempo y por último se optimiza la precisión de la tendencia con el algoritmo Adaboost. La propuesta fue desarrollada en Qt Creator obteniendo una tasa de retorno futuro del 51,08% como máximo, seguido del modelo TS-Fuzzy con un 45,16% y los demás algoritmos comparados obtienen porcentajes inferiores PLR-SVR, PLR-BPN, BIC-K-NN, BIC-NV-AD [38]. Además, en 2018, se utilizó programación evolutiva para generar estrategias de negociación cambiando la ventana de tiempo, pero conservando la población o

inicializando la población de los genotipos, para el par divisa EUR/USD (euro/dólar) del mercado Forex. La propuesta fue implementada en Java SE6 y ser comparada con la estrategia B&H [39]. Para terminar el 2018, se proponen el algoritmo de campeonato de liga (LCA) con la estructura de red equipada con backward Q-Learning (BQL), para extraer y guardar reglas de negociación óptimas para 20 activos financieros (MKBT, BDAN, ABDI, KSIM, MARK, ARDK, KSHJ, ROOI, BARZ, IKCO, INFO, MAPN, NSTH, STEH, NAFT, PARK, IRNZ, OIMC, TSHE, BTEJ) de la bolsa de valores de Teherán (tse.ir). La propuesta obtuvo una ganancia máxima en un activo financiero de 460,04% seguida de la estrategia B&H con un 441,28% y terminado en pérdidas en todos los activos con la programación de redes genéticas con acumulación de reglas (GBP-RA) [40].

Tabla 5. Estudios basados en metaheurísticas

Año	Referencia	Técnica	Técnicas	Datos fuente	Criterio de Calidad							Puntaje
					C1	C2	C3	C4	C5	C6	C7	
2015	[29]	MH	RKHS	SIBN, GAZP, LKOH, MTSI, ROSN, SBER finam.ru	1	0,5	0,5	0,5	0,5	1	1	5,0
2015	[30]	MH	GA	TPE: 2317, 2454, 1301, 1303, 1326, 2412, 2882, 2308 y 2008 twse.com.tw/zh/	1	1	1	1	0,5	0,5	0,5	5,5
2016	[31]	MH	NAIVE, MACD, ARMA, MLP, ESVM	ASE20, DAX30, NIKKEI225, FTSE100, S&P500 bloomberg.com	0,5	1	0,5	1	1	1	1	6,0
2016	[32]	MH	FTSGA, ACO-AR	TAIEX, Dow Jones, NASDAQ, HSI, SP500, NTD / USD twse.com.tw/zh/, cbc.gov.tw	1	1	1	0,5	0,5	0,5	0,5	5,0
2017	[33]	MH	FDE, RSI, MACD	MERV, S&P500, AXJO, BFX, BVSP, GSPTSE, IPSA, SCI, FTSE, FCHI, DAX, HIS, BSESN, JKSE, N225, KOSPI, KLSE, MXX, NZ50, TWII finance.yahoo.com	1	1	0,5	0,5	0,5	0,5	0,5	5,0
2017	[34]	MH	PAA-IDPSO, PAA-IDPSO-V, PAA-MS-IDPSO, PAA-MS-IDPSO-V	S&P100 finance.yahoo.com	0,5	1	1	0,5	1	1	1	6,0
2017	[6]	MH	GEP, RS, RW, MACD, ARMA, MLP, RNN, HONN, GP, GEP Trader Tool	FTSE100, DAX30, S&P500 londonstockexchange.com, nyse.com, deutsche-boerse.com	1	1	0,5	0,5	0,5	0,5	0,5	4,5
2018	[35]	MH	WTS	KOSPI 200 data.koscom.co.kr	0,5	1	0,5	0,5	0,5	0,5	1	5,0
2018	[36]	MH	ESTAR, LSTAR, TSTAR,	FTSE MIB, PSI-20, ISEQ 20, FTSE/Athens Large Cap, IBEX 35 borsaitaliana.it, bolsamadrid.es,	0,5	0,5	0,5	0,5	0,5	0	0,5	2,5

			GBELL-STAR	bolsadelisboa.com, ise.ie, athexgroup.gr										
2018	[37]	MH	GA-TA-B&H, TA-10, TA-36	BBAS3, BOVA11, CMIG4, EMBR3, GBR4, VALE5, bb.com.br, bmfbovespa.com.br, cemig.com.br, embraer.com, gerdau.com y vale.com	1	1	0,5	0,5	0,5	0,5	0,5	0,5	0,5	4,0
2018	[38]	MH	TS fuzzy, PLR-SVR, PLR-BPN, BIC-K-NN, BIC-NV-AD	Apple, BA, CAT, JNJ, XOM, VZ, S&P500 bolsa de valores de EE.UU	1	1	0,5	1	0,5	0,5	0,5	1	5,5	
2018	[39]	MH	AE, Buy & Hold	Forex No reporta	0,5	1	1	0,5	0,5	1	5,5			
2018	[40]	MH	LCA-NBQL, GNP-RA, Buy & Hold	MKBT, BDAN, ABDI, KSIM, MARK, ARDK, KSHJ, ROOI, BARZ, IKCO, INFO, MAPN, NSTH, STEH, NAFTA, PARK, IRNZ, OIMC, TSHE, BTEJ tse.ir	1	1	0,5	0,5	1	0,5	5,5			

2.3.3 Contribuciones basadas en redes neuronales artificiales

En 2015 se presentó un análisis de 113 artículos que datan de 1933 a 2013, donde se identificaron las técnicas más comúnmente utilizadas en el ámbito financiero y la inteligencia artificial (IA) para el pronóstico de los mercados. Como resultado del análisis, se observó un creciente uso de las técnicas de IA en los últimos años. Con base en esta revisión, se propuso una nueva técnica que usa filtrado Hodrick-Presscott, una transformada discreta de Fourier y un Perceptrón multicapa (Multi-Layer Perceptron, MLP) con una capa oculta entrenada con el algoritmo de Levenberg-Marquardt. En los resultados de la comparación, se definió una propuesta combinada entre MLP y árboles de decisión que supera la predicción de las series temporales basado en exactitud a MLP con ANFIS usando los datos disponibles en Yahoo! Finance para los índices bursátiles S&P500, FTSE-100 y BSE-Sensex [3].

En 2016, se usó PSO para definir los pesos iniciales de una red MLP implementada en lenguaje C, para pronosticar el movimiento de los precios de índices financieros tomando los datos históricos de Yahoo! Finance e indicadores técnicos bursátiles. La propuesta obtuvo el 154.80% de rendimiento promedio. Los autores observaron que una mayor precisión de predicción no garantiza un mayor rendimiento de los retornos [5]. También en 2016, utilizan un modelo híbrido de PCA con transformadas de wavelet y redes neuronales artificiales (WPCA-NN), para predecir los precios de los índices bursátiles NIKKEI-225, SiMSCI, KOSPI-200 y TAIEX de acuerdo con los indicadores técnicos bursátiles RSI, MACD, MACDsig, MACDHis, Stochastics (fast %K, slow %K y %D), y Ultimate Oscillator (UO) extraídos de Bloomberg. El modelo WPCA-NN con red neuronal auto regresiva no lineal con entradas exógenas (NARX-NN) entrenada con el algoritmo Levenberg-Marquardt obtiene mayor rentabilidad que los algoritmos W-NARX-NN o NARX-NN. Todos los algoritmos se evaluaron en las mismas condiciones y fueron implementados en Matlab [41].

En 2017, se estudió la predicción de la tendencia del índice bursátil KOSPI200 con una red MLP entrenada con el algoritmo back-propagation, un modelo SVM y una red neuronal con función de base radial (RBF). Estos modelos se implementaron con Matlab utilizando los datos históricos del índice obtenidos del portal web <https://global.krx.co.kr/contents/GLB> con 10 indicadores financieros adicionales. Este trabajo logró la evaluación y comparación de los modelos MLP, SVM y RBF en un framework significativamente realista que obtuvo una precisión de las predicciones del 75.74%, 71.52% y 62.23%, respectivamente [42].

En 2018 propusieron replicar una estrategia de trading utilizando máquinas de memoria a corto plazo (LSTM). Donde implementan una CNN para mapear datos sin procesar y eventualmente incrustando correlaciones parciales que después ingresan estos datos con indicadores técnicos a la LSTM. Esta técnica fue desarrollada en Python con la API Keras y Tensorflow. Se proporcionó un conjunto de datos de 30 empresas que conforman el índice bursátil Dow Jones. La red implementada fue entrenada 3.000 épocas en el caso de elección aleatoria (RC), y 15.000 épocas para el Crossover y Media móvil Convergencia/Divergencia MACD y también una tasa de abandono de 0.25 para un posible sobreajuste. Los resultados indicaron que este enfoque es viable, pero se observaron que hubo un deterioro en el rendimiento al aumentar datos de entrada como los índices técnicos. Además, esta arquitectura de LSTM al dedicarle subconjuntos de indicadores combinado con capas adicionales aumenta el rendimiento de la red [43]. Ese mismo año (2018), propone una novedosa técnica para el trading algorítmico basado en redes neuronales convolucionales profundas (Deep convolutional neural networks, DCNN). El método propuesto, inicia convirtiendo los datos de series temporales unidimensionales a imágenes bidimensionales de 15x15 que fueron etiquetados como vender, comprar y retener, según los valles o colinas de las series de tiempo. La DCNN utiliza nueve capas para su entrenamiento incluyendo convolucionales, maxpooling, drop-out y una capa densa (completamente conectada) al final. Los datos usados para el entrenamiento de la DCNN fueron tomados del índice bursátil Dow-30 y fondos de inversión cotizados (ETFs) disponibles en Yahoo! Finance, los cuales usan 15 indicadores técnicos bursátiles. Como resultado de este trabajo, los valores de precisión total para el índice bursátil Dow30 fueron de 0,58 y para ETF de 0,62. Por último, la evaluación financiera de la propuesta fue comparada contra los métodos "Buy&Hold", (BaH), RSI (14 días, 70–30), SMA (50 días), LSTM y MLP, obteniendo mejores resultados en los dos data sets, que incrementan desde un 1,5% hasta un 9% de acuerdo al método con el que se compara [44].

En 2019 proponen un modelo denominado red neuronal de filtros múltiples (Multi-Filters Neural Network, MFNN) para predecir movimientos del índice bursátil CSI 300. Toma como entrada 120 indicadores técnicos bursátiles, los procesa a través de un módulo de neuronas convolucionales y recurrentes para obtener un mapa de características. La propuesta obtuvo una precisión de predicción promedio del 49,53%, superando a otras 7 técnicas (RNN, CNN, LSTM, SVM, Logistic Regression, Random Forest y Linear Regression) más y en la prueba de rangos con signos de Wilcoxon supera a los demás en el rendimiento. Como resultado la técnica más estable debido al riesgo y rentabilidad fue una red neuronal recurrente

(Recurrent Neural Network, RNN) con un indicador bursátil Sharpe de 6.42 y con una ganancia del 67.43%, y MFNN obtuvo el segundo mejor resultado [45]. Ese mismo año (2019), proponen un algoritmo en Python de memoria de corto plazo (Long-Short Term Memory, LSTM) que utiliza Keras y TensorFlow para predecir el precio de cierre de acciones en un determinado tiempo, con datos obtenidos de Yahoo! Finance desde octubre del año 2000 a octubre de 2018. La primera red pronostica la tendencia del stock con una capa de entrada que utiliza datos del cierre de acciones, una capa oculta de 400 neuronas y con una neurona en la salida, la segunda red realiza la regresión de series temporales para predecir el precio de cierre apoyado con un modelo estadístico de Markov. La precisión del método propuesto tiene un rango del 50% al 99% [46]. También en 2019, identificaron la existencia de pocos estudios para pronosticar retornos diarios del mercado financiero con los índices bursátiles S&P500, SPDR y ETF, utilizando redes neuronales profundas (Deep Neural Network, DNN) con backpropagation. El modelo propuesto combina la DNN con el algoritmo de análisis de componentes principales que toma hasta 60 parámetros financieros obtenidos de Yahoo! Finance, investing.com y fred.stlouisfed.org, alcanzando una precisión de pronóstico del 60,4%, respecto al 57,5% de una red neuronal feed-forward tradicional (ANN) y una relación ANN/DNN del Sharpe ratio de 0,0778 [47]. Además, en 2019, se propone predecir la cantidad de acciones a negociar combinando aprendizaje por refuerzo con tres modelos de DNN continuos (regresores DNN) que luego se usan para tomar la decisión de comprar, vender o retener activos financieros (S&P500, KOSPI, HSI y EuroStoxx50) que utiliza datos de Thomsonreuters y Yahoo! Finance. Como resultado de la investigación el modelo aumento el beneficio comercial total desde cuatro veces en el índice bursátil S&P500 hasta doce veces en el índice bursátil HSI con respecto a sistemas de comercio fijo y también mejoran el modelo a la adaptabilidad del mercado real [48]. Finalmente, en 2019, se propone un modelo híbrido con la red neuronal basada en máquina de descomposición y factorización empírica (EMD2FNN) para pronosticar lo precios de cierre en los activos financieros (SSEC, NASDAQ y S&P 500) con datos extraídos de Yahoo! Finance. Los modelos EMD2FNN simple y EMD2FNN múltiple obtuvieron un rendimiento promedio anual del 13% y el 25%, respectivamente [49].

Tabla 6. Estudios basados en redes neuronales artificiales

Año	Referencia	Técnica	Técnicas	Datos fuente	Criterio de Calidad							Puntaje	
					C1	C2	C3	C4	C5	C6	C7		
2015	[3]	ANN	MLP, Fourier, Hodrick-Presscott	BSE-Sensex, FTSE-100, S&P 500 finance.yahoo.com	1	1	1	1	1	1	1	1	7,0
2016	[5]	ANN	PSO, RNA	IXIC, SPY, QQQ, MDY, IWM, FXI, EFA, EEM, HSI, TWII, BSESN, XLF, XLY, XLE, XLV, XLK, SMH, OIH, FTSE, ATX, FCHI, AEX finance.yahoo.com	1	1	1	0,5	0,5	1	1	6,0	
2016	[41]	MH ANN	WPCA-NN, WNN, NN, B&H	Hang Seng, NIKKEI 225, MSCI, KOSPI 200, TAIEX bloomberg.com	1	1	1	1	0,5	0,5	1	6,0	

2017	[42]	MH, ML ANN	ANN, SVM	RBF,	Kospi200 global.krx.co.kr/contents/GLB	0,5	1	1	1	1	1	1	6,5
2018	[43]	ANN	LSTM, CNN		AAPL, AXP, BA, CAT, CSCO, CVX, DD, DIS, GE, GS, HD, IBM, INTC, JNJ, JPM, KO, MCD, MMM, MRK, MSFT, NKE, PFE, PG, TRV, UNH, UTX, V, VZ, WMT, XOM No reporta	1	1	1	0,5	0,5	0,5	0,5	4,5
2018	[44]	ANN	DCNN, CNNTAr, RSI, LSTM, MLP	BaH, SMA,	Dow-30, ETFs: XLF, XLU, QQQ, SPY, XLP, EWZ, EWH, XLY, XLE finance.yahoo.com	1	1	1	1	1	0,5	1	6,5
2019	[45]	ML ANN	MFNN, CNN, SVM, Regression, Random Forest, Linear Regression	RNN, LSTM, Logistic	CSI300 no reporta	0,5	1	0,5	0,5	0,5	0,5	1	5,0
2019	[46]	ML ANN	LSTM Markov,		ENEL.MI, UCG.MI, BMPS.MI, CVAL.MI, ISP.MI finance.yahoo.com	1	0,5	1	0,5	1	0,5	0,5	5,5
2019	[47]	ANN	ANN(MLP), DNN, PCA		SPDR, S&P500, ETF finance.yahoo.com, investing.com, fred.stlouisfed.org	1	1	1	1	0,5	0,5	1	6,0
2019	[48]	ANN	DNN, DQN		S&P500, KOSPI, HSI, EuroStoxx50 thomsonreuters.com, finance.yahoo.com	1	1	0,5	0,5	0,5	0,5	1	5,0
2019	[49]	ANN	EMD2FNN, FNN, NN(MLP)		SSEC, NASDAQ, S&P 500 finance.yahoo.com	1	0,5	0,5	0,5	0,5	0,5	1	4,5

2.3.4 Contribuciones basadas en lógica difusa

En 2015, se recuperó información de profesionales expertos en análisis técnico con más de 15 años de experiencia, mediante cuestionarios para el sistema de inferencia difuso tipo Mandani, obteniendo como resultado que indicadores utilizar, y con la información se extraen reglas de negociación con el enfoque de conjunto aproximado basado en el dominio de la consistencia variable (Variable Consistency Dominance-based Rough Set Approach, VC-DRSA). El VC-DRSA utiliza el software jMAF y alcanzó un 76% de precisión en clasificación, superando al DRSA que también utiliza jMAF y al análisis discriminatorio que emplea una herramienta estadística de Matlab con los datos extraídos de sysjust.com.tw [50].

En 2016 se implementó un sistema experto para el mercado Forex con un enfoque de razonamiento probatorio basado en reglas (Rule-Based Evidential Reasoning, RBER) para tomar decisiones de comprar o vender, basado en análisis de gráficos de velas mediante indicadores técnicos adaptados con lógica difusa. Esta implementación se adaptó a la plataforma MetaTrader4 y la evaluación mostró una rentabilidad mínima del 45.77% [51]. También en 2016, implementaron un sistema para predecir puntos de inflexión utilizando datos históricos de la bolsa de valores

de Estados Unidos con el modelo difuso Takagi-Sugeno-Kan (TSK) a partir de indicadores técnicos y señales de negociación del modelo SVR. El resultado de esta investigación indica que la propuesta supera a otros modelos como PLR-SVR y PLR-BPN hasta en un 6% en la tasa de beneficio [52].

En 2018 se implementó un modelo de red Petri difusa de alto nivel (High-Level Fuzzy Petri Net, HLFPN) en C# con ayuda de máquinas de soporte vectorial que utilizan LibSVM (Library for Support Vector Machines) para tomar decisiones de compra o venta de los activos financieros TWSE, Wei Chuan, Formosa Plastic, China Steel, UMC, TSMC, Chunghwa Telecom, Grand Formosa Regent Taipei y Mega Bank Public Co. Ltd obtenidas de twse.com.tw/en. Como resultado de la investigación obtuvieron una predicción total del 64,05% y con el retorno anual promedio del 4,80% con la propuesta HLFPN siendo mejor que SVM, Neural Fuzzy, Technical Indicator Strategies y Trend Following Method, [7].

Tabla 7. Estudios basados en lógica difusa

Año	Referencia	Técnica	Técnicas	Datos fuente	Criterio de Calidad							Puntaje
					C1	C2	C3	C4	C5	C6	C7	
2015	[50]	FL	FIS-Mandini, VC-DRSA	TAIEX sysjust.com.tw	1	1	1	1	0,5	0,5	0,5	5,5
2016	[51]	FL	RBER	Forex metatrader4.com	1	0,5	0,5	0,5	0,5	0,5	0,5	4,0
2016	[52]	FL	TSK, SVR	Bolsa US No reporta	0,5	1	1	0,5	0,5	0,5	1	5,0
2018	[7]	FL	HLFPN, SVM, Neural Fuzzy, Technical Indicator Strategies, Trend Following Method, Trend Following Method, Buy & Hold	TWSE, Wei Chuan, Formosa Plastic, China Steel, UMC, TSMC, Chunghwa Telecom, Grand Formosa Regent Taipei, and Mega Bank Public Co. Ltd twse.com.tw/en	1	1	1	1	1	0,5	1	6,5

2.4 Conclusiones recomendaciones y trabajo futuro

En este documento se identifican las técnicas, métodos, modelos o similares, utilizados en el trading algorítmico para la predicción de series de tiempo financieras. Estos trabajos centran sus contribuciones principalmente en aprendizaje de máquina, metaheurísticas, redes neuronales artificiales y lógica difusa, observándose para los últimos años, una mayor actividad investigativa en el aprendizaje de máquina seguido de las redes neuronales artificiales. Las investigaciones utilizan principalmente datos de Yahoo! Finance debido a su fácil acceso respecto a otras fuentes expuestas tomando en cuenta la ventana de tiempo de investigación que fue desde 2015 a 2019. Una de las técnicas más innovadoras en redes neuronales fue la de convertir series temporales en imágenes para ser procesadas por redes neuronales convolucionales obteniendo hasta un 62% de precisión de pronóstico. En el enfoque metaheurístico se encuentran propuestas que sobrepasan el 70% de precisión en pronóstico bajo condiciones similares, pero no iguales. Debido a esto es difícil definir cuáles son las mejores propuestas y en

que contextos obtienen esos mejores resultados, ya que entre ellas no se comparan, razón por la cual se necesita un framework de evaluación y comparación de algoritmos de trading algoritmo de activos financieros.

Esta página se ha dejado intencionalmente en blanco.

CAPÍTULO 3

3 Framework propuesto

Un marco de trabajo es una estructura que proporciona herramientas para facilitar el desarrollo de aplicaciones en diversas áreas. Estos frameworks incluyen componentes preconstruidos que pueden ser utilizados para resolver problemas comunes de las aplicaciones, lo que reduce el tiempo y esfuerzo necesarios para desarrollarlas. Los marcos de trabajo también ayudan a mejorar la calidad del software, simplificar la mantenibilidad del código y reducir la complejidad de la aplicación. Ejemplos populares de frameworks en trading algorítmico incluyen *yfinance* y *TA – Lib* que están desarrollados en Python.

El framework propuesto se ha desarrollado con el fin de facilitar la evaluación y comparación de tres algoritmos de trading (el número de algoritmos puede crecer), permitiendo a los usuarios analizar el rendimiento y rentabilidad de los algoritmos en un entorno controlado. Este proceso se realiza usando datos históricos de Yahoo Finance y con métricas comúnmente aceptadas por la comunidad académica y científica en un entorno justo de comparación.

La implementación de una interfaz gráfica fácil de usar permite a los usuarios seleccionar las líneas de tiempo para los datos de entrenamiento y de validación que usarán los algoritmos, así como el número de evaluaciones que se deben realizar. Además, los usuarios tienen la opción de ajustar los parámetros de los algoritmos según sea requerido.

Una vez que se ha completado el proceso de evaluación, el usuario puede visualizar los resultados en forma de gráficos que muestran la evolución del dinero invertido a lo largo del tiempo para cada activo financiero evaluado. Esta información es esencial para comprender el desempeño de cada algoritmo y tomar decisiones informadas sobre la inversión de los recursos.

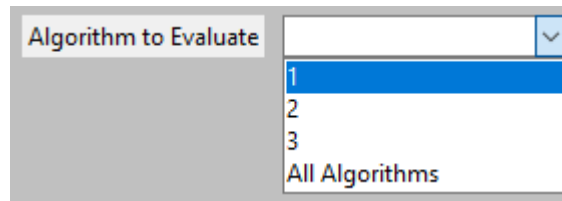
En resumen, el framework propuesto permite la evaluación y comparación justa de algoritmos de trading, lo que puede ser útil tanto para inversores individuales como para empresas financieras. La combinación de datos históricos, una interfaz gráfica fácil de usar y la posibilidad de ajustar los parámetros de los algoritmos hacen de este framework una solución útil y adaptable a diferentes necesidades y contextos.

A continuación, se describen detalles del uso y la implementación del framework propuesto para la evaluación y comparación de algoritmos de trading.

3.1 Interfaz Gráfica del framework

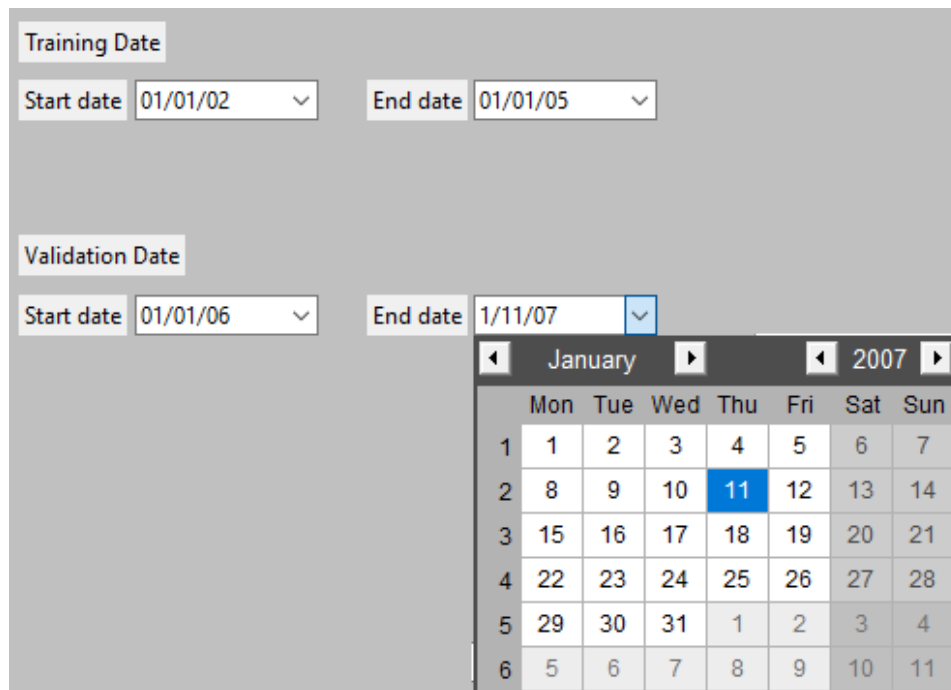
Haciendo uso de la API Tkinter se creó una interfaz gráfica de usuario que permitió evaluar cada algoritmo por separado o todos los algoritmos al mismo tiempo. Cabe destacar que cada algoritmo se identifica con un número específico: el '1' corresponde al algoritmo basado en redes neuronales convolucionales profundas, el '2' al algoritmo basado en aprendizaje por refuerzo profundo, el '3' al algoritmo basado en una red LSTM, y finalmente, la opción 'All Algorithms' ejecuta todos los algoritmos mencionados anteriormente, tal y como se puede apreciar en la **Figura 1**.

Figura 1. Interfaz gráfica del framework para seleccionar el algoritmo a evaluar



Usando un DateEntry se soporta la selección de fechas para la línea de tiempo de los datos que conformaran el conjunto de entrenamiento y otra para el conjunto de validación como se muestra en la **Figura 2**.

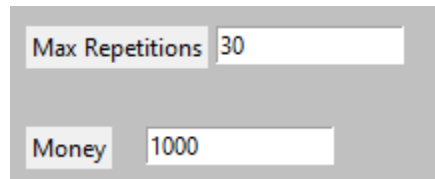
Figura 2. Interfaz gráfica del framework para seleccionar las fechas de entrenamiento y validación



Se cuenta con dos campos de entrada denominados 'Max Repetitions' y 'Money', respectivamente (ver **Figura 3**). El primero de ellos permite evaluar el número de veces que se ejecuta cada algoritmo (cada algoritmo se ejecuta este número de

repeticiones y los resultados de todas esas repeticiones se promedian para obtener el comportamiento esperado del algoritmo, esto se hace teniendo en cuenta que son algoritmos probabilísticos y no estocásticos), mientras que el segundo permite asignar una cantidad de dinero a invertir en cada activo financiero que se desee evaluar.

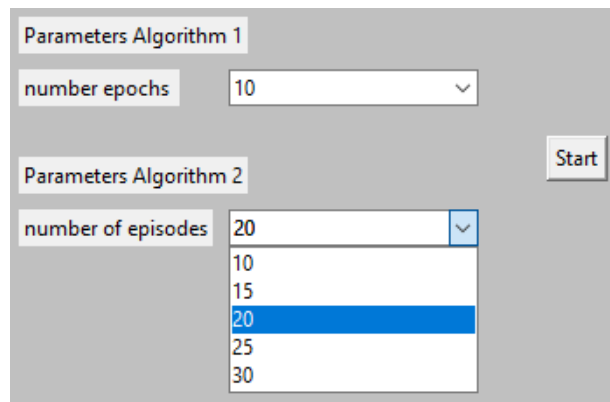
Figura 3. Interfaz gráfica del framework para asignar valores de evaluación y dinero para invertir



The image shows a graphical user interface with two input fields. The first field is labeled 'Max Repetitions' and contains the value '30'. The second field is labeled 'Money' and contains the value '1000'.

Por último, se presentan dos Combo box que permiten especificar el número de épocas para el algoritmo '1' y el número de episodios para el algoritmo '2'. Una vez que se han asignado los valores a estos parámetros, se puede iniciar el framework haciendo clic en el botón 'Start', el cual se encarga de ejecutar los procedimientos correspondientes (ver **Figura 4**). El algoritmo 3 no requiere la definición de ningún parámetro.

Figura 4. Interfaz gráfica del framework para asignar parámetros de dos algoritmos y botón de 'Start' para dar inicio



The image shows a graphical user interface with two sections. The first section is titled 'Parameters Algorithm 1' and has a dropdown menu labeled 'number epochs' with the value '10' selected. The second section is titled 'Parameters Algorithm 2' and has a dropdown menu labeled 'number of episodes' with a list of options: 10, 15, 20, 25, and 30. The option '20' is currently selected and highlighted in blue. To the right of these sections is a button labeled 'Start'.

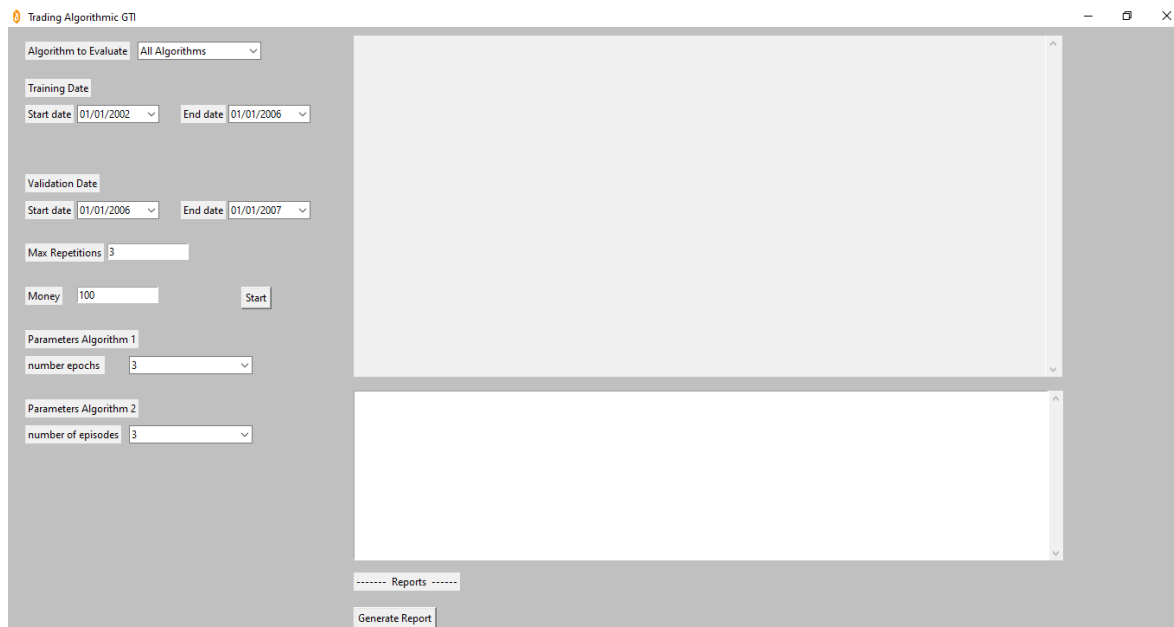
Una vez se terminan de ejecutar los algoritmos, los resultados se presentan en la misma interfaz gráfica. Esto incluye gráficos que muestren la evolución del precio del dinero en el tiempo para la evaluación de cada activo financiero, los cuales se pueden visualizar en un Scrolled Text al presionar el botón 'Generate Report', tal como se muestra a continuación en la **Figura 5**.

La **Figura 6** muestra la interfaz gráfica completa mencionada previamente por partes.

Figura 5. Interfaz gráfica del framework para visualizar los resultados obtenidos



Figura 6. Interfaz gráfica del framework para evaluar y comparar modelos de pronóstico en series de tiempo financieras

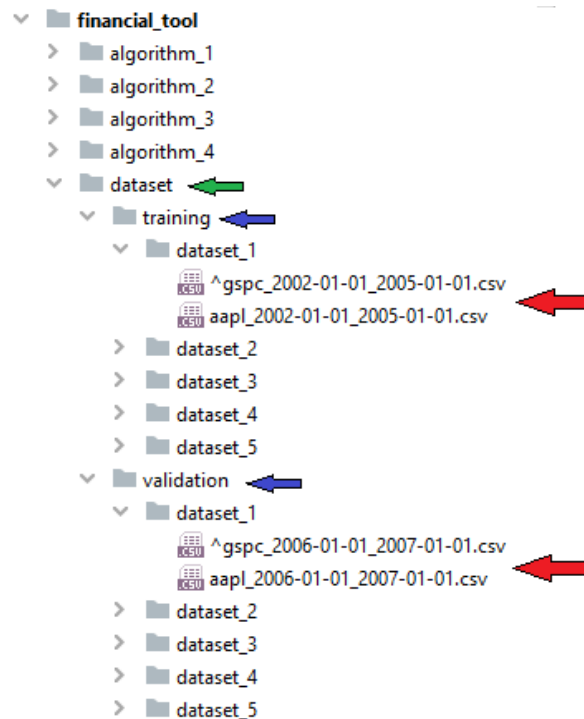


3.2 Datos suministrados al framework

Para implementar el marco de trabajo propuesto, se utilizaron datos extraídos desde <https://finance.yahoo.com/>, ya que este repositorio es ampliamente usado en la investigación del trading algorítmico, entre otras razones por tener datos reales, de muchos años y que es accesible sin costo alguno [53]. La descarga de estos datos se realizó a través de la API yfinance, que permite solicitar información de diferentes activos financieros mediante sus siglas (por ejemplo, Apple=aapl, Tesla=tsla, entre otras) y las fechas de inicio y fin correspondientes, seleccionadas a través de una interfaz gráfica en una línea de tiempo para la conformación del conjunto de datos de entrenamiento y de validación.

Una vez descargados los datos, estos se almacenan en un archivo con formato CSV (Comma Separated Values) que llevan el nombre del activo financiero y la línea de tiempo correspondiente, y se ubica en su respectiva carpeta dependiendo si es de entrenamiento (training) o de validación (testing), tal como se muestra en la **Figura 13**. Esto permite tener los data sets ordenados y organizados en el framework.

Figura 7. Ubicación de activos financieros en formato CSV para entrenamiento y validación



Los datos obtenidos sean de entrenamiento o de validación cuentan con los atributos que se muestran en la **Tabla 8**.

3.3 Cálculo de indicadores financieros

Los indicadores técnicos en trading son funciones matemáticas y estadísticas que se aplican en las series de precios y volúmenes para ayudar a tomar decisiones.

Estos indicadores son de diferentes tipos como: indicadores de impulso, indicadores de volumen, indicadores de volatilidad, indicadores de transformación de precios e indicadores de ciclo [44].

En este marco de trabajo se implementó la obtención de 15 indicadores técnicos para un intervalo de días a estudiar. Para realizar este cálculo se utilizó las APIs de *TA – Lib* y *ta* con periodos de 6 hasta 20 días con el lenguaje de programación Python, para generar un Data Frame como el que se muestra en la **Tabla 9**.

Tabla 8. Datos contenidos en cada activo financiero (fecha, precio de apertura, precio más alto, precio más bajo, precio de cierre, precio de cierre ajustado y número de acciones negociadas por día)

Date	Open	High	Low	Close	Adj Close	Volume
2006-01-03	2.58500003	2.66964292	2.58035707	2.66964292	2.27565145	807234400.0
2006-01-04	2.68321394	2.71357107	2.66071391	2.67750000	2.28234815	619603600.0
2006-01-05	2.67249989	2.67499995	2.63392901	2.65642905	2.26438760	449422400.0
2006-01-06	2.6875	2.73928594	2.66249990	2.72499990	2.32283878	704457600.0
2006-01-09	2.74035692	2.75714302	2.70499992	2.71607089	2.31522846	675040800.0
2006-01-10	2.72321391	2.92464303	2.70821404	2.88785696	2.46166086	2279869200
2006-01-11	2.99428606	3.02857089	2.94964289	2.99642896	2.55420947	1493794400

Tabla 9. Indicadores financieros calculados con periodos de 6 a 20 días

RSI 6	RSI 7	R	R	R	R	R	R	R	R	R	R	RSI 19	RSI 20	WILLIAMS 6	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	WILLIAMS 20	
														-5.8003390	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-5.80033908	
														-6.4462837	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-6.44628371	
														-11.004327	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-11.0043274	
														-1.5861587	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-1.58615870	
														-1.3987662	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-1.39876628	
														-2.4202640	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-2.42026408	
98.25														-2.6508460	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-1.46454475	
71.69	75.07													-35.934856	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-17.9820172	
73.34	76.31	7												-34.033725	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-14.8291985	
60.54	64.90	6	6											-73.480306	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-24.3489877	
49.48	54.70	5	5	6										-74.364241	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-34.5198513	
61.49	64.07	6	6	6	6									-43.207518	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-20.0569131	
31.61	35.61	3	4	4	4	4								-98.286391	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-67.9617798	
35.34	38.75	4	4	4	4	4	4							-89.657949	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-63.2222467	
40.43	42.98	4	4	4	4	4	4	4						-77.893775	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-57.0382822
37.86	40.63	4	4	4	4	4	4	4	4					-81.459250	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-61.4726397
52.90	53.17	5	5	5	5	5	5	5	5	5				-49.207209	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-42.8601580
64.15	63.01	6	6	6	6	6	5	5	5	5	5			-9.8660099	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-22.7421601
65.62	64.31	6	6	6	6	6	6	5	5	5	5	5		-9.6423909	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-19.7519858
56.06	56.28	5	5	5	5	5	5	5	5	5	5	55.151		-27.559655	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-30.1465786
59.37	59.06	5	5	5	5	5	5	5	5	5	5	56.157	56.105	-19.214586	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-35.0622049
41.20	43.37	4	4	4	4	4	4	4	4	4	4	50.338	50.575	-73.517016	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-67.8130547
33.90	36.70	3	4	4	4	4	4	4	4	4	4	47.305	47.676	-88.818653	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-87.0067985
35.88	38.31	4	4	4	4	4	4	4	4	4	4	47.788	48.131	-85.141115	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-84.2165200

A continuación se describen cada uno de los 15 indicadores técnicos implementados.

3.3.1 Indicador RSI

El índice de fuerza relativa (Relative Strength Index, RSI) mide la fortaleza y debilidad de los precios de las acciones en una escala de 0 a 100. Valores por encima de 70 indican "sobrecompra" y por debajo de 30 "sobreventa" [44]. La fórmula para calcular el RSI se muestra en la Ecuación (1).

$$RSI = 100 - \frac{100}{1 + \left(\frac{\bar{x}_g}{\bar{x}_p}\right)} \quad (1)$$

Donde:

$$\begin{aligned} \bar{x}_g &= \text{ganancia promedio} \\ \bar{x}_p &= \text{perdida promedio} \end{aligned}$$

3.3.2 Indicador Williams

El indicador técnico Williams% R se basa en el impulso y detecta las condiciones de sobrecompra y sobreventa de los precios de las acciones. Oscila entre -100 y 0, y se considera que los precios de las acciones están en la región de "sobreventa" si el valor está por debajo de -80, mientras que los valores superiores a -20 se consideran "sobrecompra"[44]. La Ecuación (2) se utiliza para calcular el valor de Williams% R.

$$R = \frac{M_H - P_f}{M_H - M_L} \quad (2)$$

Donde:

$$\begin{aligned} M_H &= \text{Precio mas alto alcanzado durante el periodo analizado} \\ M_L &= \text{Precio mas bajo alcanzado durante el periodo analizado} \\ P_f &= \text{Precio de cierre final para el periodo analizado} \end{aligned}$$

3.3.3 indicador SMA

El promedio móvil simple (Simple Moving Average, SMA) muestra el promedio móvil de los precios para un período determinado. En su interpretación más ampliamente aceptada, la intersección de los valores de SMA con diferentes valores de intervalo se utilizan para determinar la dirección de la tendencia. Como resultado, se pueden combinar varios SMA para usarlos juntos, o se puede usar un solo valor de SMA junto con el stock subyacente, es decir, si el precio del stock es más alto que el SMA (por ejemplo, 50 días), se supone que el stock está en tendencia alcista, que indica que el precio de la acción continuará aumentando (disparador de compra), mientras que si el precio de la acción es más bajo que el SMA, se supone que la acción está en tendencia bajista, lo que indica que el precio de la acción disminuirá (disparador de venta) [44]. El cálculo de SMA se resume en la Ecuación (3).

$$SMA(M, n) = \sum_{k=a+1}^{a+n} \frac{M(k)}{n} \quad (3)$$

3.3.4 Indicador EMA

El promedio móvil exponencial (Exponential Moving Average, EMA) es un tipo de indicador de promedio móvil que muestra el promedio móvil de los precios haciendo mayor énfasis para los últimos días. Los últimos datos tienen más peso al calcular

el promedio móvil. La importancia de los últimos datos más recientes se aumenta exponencialmente en los cálculos de EMA. La ecuación (4) ilustra el cálculo de EMA de los precios de las acciones [44].

$$(M(t) - EMA(M, t - 1, \tau)) \cdot \frac{2}{\tau + 1} + EMA(M, t - 1, \tau) \quad (4)$$

3.3.5 Indicador WMA

El promedio móvil ponderado (Weighted Moving Average, WMA) es un indicador similar al promedio móvil exponencial. La única diferencia es que la importancia del precio de cierre está disminuyendo linealmente con datos más antiguos. Por otro lado, la importancia del precio de cierre de las acciones disminuye exponencialmente en EMA [44] para esos datos antiguos. La ecuación (5) muestra cómo se calcula WMA.

$$WMA(M, n) = \frac{\text{Suma de promedios ponderados}}{\text{Suma de los pesos}} \quad (5)$$

3.3.6 Indicador HMA

El promedio móvil del casco (Hull Moving Average, HMA) es un tipo de indicador de promedio móvil que reduce el retraso asociado con SMA. EMA y WMA prueban la reducción del retraso utilizando más énfasis en los últimos datos. HMA mejora esta reducción del retraso y obtiene mejores resultados en comparación con EMA y WMA [44]. La ecuación (6) exhibe el cálculo de HMA.

$$WMA(M, n) = WMA\left(2 \cdot WMA\left(\frac{n}{2}\right) - WMA(n)\right), \text{sqrt}(n) \quad (6)$$

3.3.7 Indicador TEMA

El promedio móvil exponencial triple (Triple Exponential Moving Average, TEMA) es un tipo de indicador EMA que proporciona la reducción de las fluctuaciones de los precios menores y filtra la volatilidad [44]. Se puede calcular con base en la Ecuación (7).

$$TEMA = (3 \cdot EMA - 3 \cdot EMA(EMA)) + EMA(EMA(EMA)) \quad (7)$$

3.3.8 Indicador CCI

El índice de canal de productos básicos (Consumer Confidence Indicator, CCI) es un indicador que compara los precios actuales y el precio promedio durante un período de tiempo. Es importante tener en cuenta que el CCI normalmente oscila alrededor del 75% del tiempo dentro de un rango que va desde -100 hasta 100, reflejando las condiciones habituales del mercado. Sin embargo, durante el 25%

restante del período de tiempo, el indicador puede extenderse más allá de estos límites y mostrar valores que se encuentran fuera del rango que indica señal de compra por encima de +100 y venta por debajo de -100 [44]. Las Ecuaciones (8) y (9) muestran los cálculos para CCI.

$$CCI = \frac{P_M - SMA \text{ de } P_M}{0,015 \times D_M}$$

Donde:

$$D_M = \text{Desviación Media} \quad (8)$$

$SMA \text{ de } P_M = \text{Promedio de los precios medios durante un periodo elegido}$

$P_M = \text{Precio Medio}$

$$P_M = \frac{\text{Precio Alto} + \text{Precio Bajo} + \text{Precio de cierre}}{3} \quad (9)$$

3.3.9 Indicador CMO

El oscilador de momento de Chande (Chande Momentum Oscillator, CMO) es un tipo de indicador de momento similar al RSI y al oscilador estocástico. Oscila entre -100 y 100. Si el valor del indicador es superior a 50, se interpreta que los precios de las acciones están en la región de "sobrecompra". Si el valor está por debajo de -50, se considera comúnmente que los precios de las acciones están en la región de "sobreventa" [44]. La fórmula del indicador se ilustra en la Ecuación (10).

$$CMO = \frac{S_u - S_d}{S_u + S_d} \quad (10)$$

Donde:

$S_u = \text{suma del impulso de los días de subida}$

$S_d = \text{suma del impulso de los días de bajada}$

3.3.10 Indicador MACD

La convergencia y divergencia de la media móvil (Moving Average Convergence/Divergence, MACD) es un indicador técnico que muestra la tendencia de los precios de las acciones. Si la línea MACD cruza las líneas de señal en dirección ascendente, se predice que los precios de las acciones aumentarán. Por el contrario, si la línea MACD cruza las líneas de señal en dirección descendente, se interpreta que los precios de las acciones disminuirán [44]. Las Ecuaciones (11) y (12) muestran los cálculos de MACD y las líneas de señal.

$$MACD = 12 \text{ Day EMA} - 26 \text{ Day EMA} \quad (11)$$

$$\text{Línea de señal: EMA de 9 días de la línea MACD} \quad (12)$$

3.3.11 Indicador CMF

El flujo de dinero de Chaikin (Chaikin Money Flow, CMF) es un indicador técnico que se utiliza para medir el volumen de flujo de dinero durante un período de tiempo. El valor del indicador fluctúa entre 1 y -1. Si el valor está más cerca de 1, se interpreta que la presión de compra es mayor. Por el contrario, si el valor está más cerca de -1, se interpreta que la presión de venta es mayor [44]. Las Ecuaciones (13), (14) y (15) ilustran el cálculo de CMF.

$$\begin{aligned} & \text{Multiplicador} \\ & = \frac{(\text{Precio cierre} - \text{Precio Bajo}) - (\text{Precio Alto} - \text{Precio cierre})}{\text{Precio Alto} - \text{Precio Bajo}} \end{aligned} \quad (13)$$

$$\text{volumen de flujo de dinero (MFV)} = \text{Volumen} * \text{Multiplicador} \quad (14)$$

$$\text{CMF de 21 Periodos} = \frac{\text{Suma de 21 períodos de MFV}}{\text{Suma de 21 períodos del volumen}} \quad (15)$$

3.3.12 Indicador DMI

El indicador de movimiento direccional (Directional Movement Index, DMI) es un indicador técnico que muestra la fuerza y la dirección de la tendencia. Consta de tres indicadores separados: índice direccional promedio (ADX), más el indicador direccional (+DI) y menos el indicador direccional (-DI). DMI oscila entre 0 y 100 [44]. El **Algoritmo 1**, y las Ecuaciones (16), (17) y (18) muestran el cálculo de DMI.

Algoritmo 1. Cálculo de DMI

Procedure DMI():

1. *UpMove = Current High - Previous High*
2. *DownMove = Current Low - Previous Low*
3. *If (UpMove > DownMove and UpMove > 0)*
4. *then return(+DMI) = UpMove,*
5. *else return(+DMI) = 0*
6. *If (DownMove > Upmove and DownMove > 0)*
7. *the return(-DMI) = DownMove,*
8. *else return(-DMI) = 0*

$$+DI = 100 * EMA \left(\frac{+DMI}{\text{Average True Range}} \right) \quad (16)$$

$$-DI = 100 * EMA \left(\frac{-DMI}{\text{Average True Range}} \right) \quad (17)$$

$$ADX = 100 * EMA \left(\text{absolute value of} \left(\frac{+DI - (-DI)}{+DI + (-DI)} \right) \right) \quad (18)$$

3.3.13 Indicador PPO

El oscilador del precio porcentual (Percentage Price Oscillator, PPO) es similar al MACD. El cálculo del PPO y la línea de señal de PPO se ilustran en las Ecuaciones (19) y (20) [44].

$$PPO = \frac{(12 \text{ Day EMA} - 26 \text{ Day EMA})}{26 \text{ Day EMA}} \cdot 100 \quad (19)$$

$$\text{Signal Line} : 9 \text{ Day EMA of PPO} \quad (20)$$

3.3.14 Indicador ROC

La tasa de cambio (Rate Of Change, ROC) es un indicador técnico que ilustra la velocidad del cambio de precio durante un período de tiempo [44]. La Ecuación (21) muestra el cálculo del ROC.

$$ROC = \frac{(\text{Latest Close} - \text{Previous Close})}{\text{Previous Close}} \cdot 100 \quad (21)$$

3.3.15 Indicador SAR parabólico

El SAR parabólico (SAR) es un indicador de análisis técnico que se utiliza para determinar puntos de posibles paradas y retrocesos. El SAR actual se calcula con tres elementos; SAR previo (PSAR), punto extremo (EP) y factor de aceleración (AF). El SAR anterior es un valor de SAR del período anterior. EP es el punto extremo (máximo o mínimo) del precio del activo durante la tendencia actual. AF explica la sensibilidad del SAR. El AF comienza en 0.02 y aumenta en 0.02 cada vez que el EP aumenta en un SAR. El AF disminuye en 0.02 cada vez que el EP cae en un SAR descendente [44]. La Ecuación (22) muestra el cálculo del SAR parabólico ascendente. El SAR parabólico descendente se calcula con la Ecuación (23).

$$PSAR + AF \text{ anterior}(EP \text{ anterior} + SAR) \quad (22)$$

$$PSAR - AF \text{ anterior}(SAR - EP \text{ anterior}) \quad (23)$$

3.4 Etiquetado del dataset

Luego de obtener los datos financieros desde <https://finance.yahoo.com/>, se procede a la tarea de etiquetado para determinar si un activo debe ser comprado, vendido o mantenido para cada día en la línea de tiempo de entrenamiento y validación. Para llevar a cabo esta tarea, se utiliza una ventana deslizante que identifica los puntos superiores e inferiores como "vender" y "comprar", respectivamente, mientras que los demás días son etiquetados como "mantener".

Cabe destacar que solamente el algoritmo de trading basado en redes neuronales convolucionales profundas hace uso de los datos etiquetados de entrenamiento y validación, mientras que los otros dos algoritmos solo hacen uso de los datos de validación. El etiquetado se realiza mediante la aplicación del **Algoritmo 2**.

Algoritmo 2. *Etiquetado diario de cada activo financiero (comprar, vender o retener) (Tomado de [44])*

```
1:  procedimiento_etiquetado()
2:  tamaño_ventana = 11 dias
3:  while(contador_filas < numero_dias_archivo)
4:      contador_filas ++
5:      if(contador_filas > tamaño_ventana)
6:          indice_inicial_ventana = contador_filas - tamaño_ventana
7:          indice_final_ventana = indice_inicial_venta + tamaño_ventana - 1
8:          indice_medio_ventana = (indice_inicial_ventana + indice_final_ventana)/2
9:          for (i = indice_inicial_ventana; i <= indice_final_ventana; i ++ )
10:             numero = lista_precio_cierre.obtener(i)
11:             if(número < minimo)
12:                 minimo = número
13:                 indice_minimo = lista_precio_cierre.obtener_indice(minimo)
14:             if(número > maximo)
15:                 maximo = número
16:                 indice_maximo = lista_precio_cierre.obtener_indice(maximo)
17:             if(indice_maximo == indice_medio_ventana)
18:                 resultado = "vender"
19:             elif(indice_minimo == indice_medio_ventana)
20:                 resultado = "comprar"
21:             else
22:                 resultado = "retener"
```

El etiquetado creado por el algoritmo para los datos de entrenamiento y validación se guarda en la carpeta '*strategy*' como un archivo CSV, junto con su línea de tiempo correspondiente, tal como se muestra en la **Figura 8**. Los datos contenidos en estos archivos corresponden a los presentados en la **Tabla 10**.

3.5 Técnicas de trading algorítmico implementadas al framework

En esta sección se describen tres algoritmos de trading: *algorithm_1*, *algorithm_2* y *algorithm_3*, que utilizan diferentes técnicas para su implementación. El *algorithm_1* utiliza redes neuronales convolucionales, el *algorithm_2* se basa en aprendizaje por refuerzo profundo y el *algorithm_3* utiliza una red LSTM (Long Short-Term Memory).

Para ejecutar cada algoritmo, se utilizó un Data Frame que contenía los nombres de los activos financieros (ver **Tabla 11**), así como las líneas de tiempo para el entrenamiento y la validación. También se especificaron el número de veces que se debía ejecutar cada algoritmo (número de repeticiones), el capital de dinero a invertir, el número de épocas para el *algorithm_1*, y el número de episodios para *algorithm_2*.

Como resultado, cada algoritmo exportó un archivo CSV por cada evaluación obteniendo los siguientes datos: acción realizada diaria (*action* = "comprar",

“vender” o “retener”), dinero invertido en el activo financiero (*holding*), dinero disponible en efectivo para invertir (*cash*), dinero total ($money = holding + cash$) y retornos diarios porcentuales (Return) como se muestra en la **Tabla 12**.

Figura 8. Ubicación de los archivos CSV de cada activo financiero debidamente etiquetados en cada día como ‘comprar’, ‘vender’ o retener

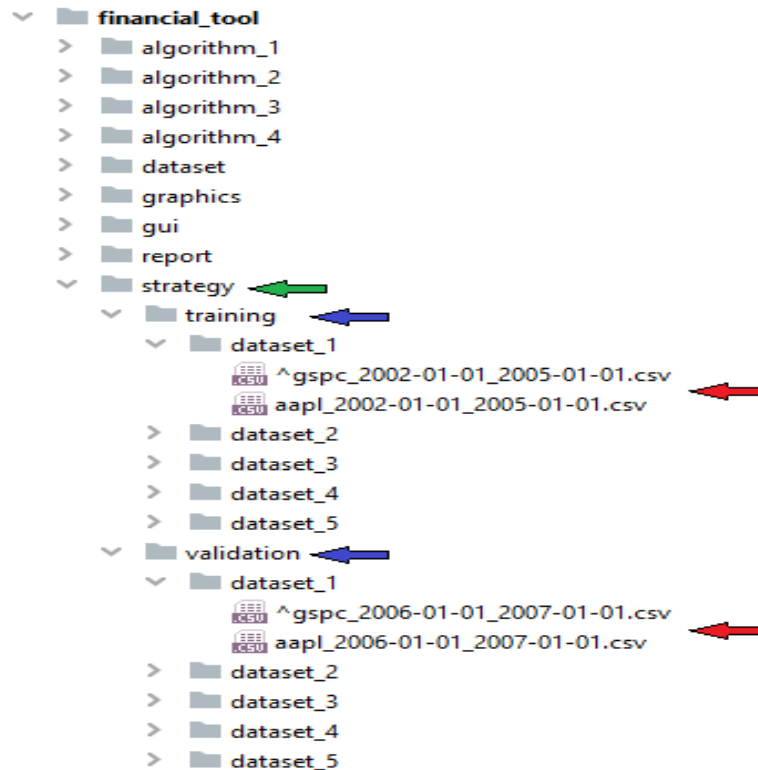


Tabla 10. Datos contenidos en cada activo financiero (fecha, precio de apertura, precio más alto, precio más bajo, precio de cierre, precio de cierre ajustado, volumen y etiqueta)

Date	Open	High	Low	Close	Adj Close	Volume	Etiqueta
2002-02-14	0.44732099	0.45053601	0.43535700	0.43928599	0.37388440	260170400	retener
2002-02-15	0.43803599	0.44607099	0.42589300	0.42678600	0.36324533	260187200	retener
2002-02-19	0.42428600	0.42625001	0.40142899	0.40392899	0.34379127	390258400	retener
2002-02-20	0.40660700	0.41428598	0.39910700	0.41303598	0.35154244	285443200	retener
2002-02-21	0.40928599	0.41071400	0.38303598	0.38392901	0.32676902	446751200	retener
2002-02-22	0.38678601	0.40982100	0.38392901	0.40607100	0.34561443	406476000	vender
2002-02-25	0.40803599	0.44142898	0.39928600	0.42517900	0.36187770	426848800	retener
2002-02-26	0.42696401	0.43517899	0.41517901	0.42267900	0.35974991	260131200	retener
2002-02-27	0.42750000	0.43303599	0.37392899	0.39214301	0.33376014	1030159200	retener
2002-02-28	0.39553600	0.40339300	0.38124999	0.38749998	0.32980823	456937600	retener
2002-03-01	0.39160698	0.41964301	0.38964301	0.41874998	0.35640573	348992000	comprar
2002-03-04	0.41535699	0.43892899	0.40642899	0.43375000	0.36917260	348258400	retener

Tabla 11. DataFrame de los activos financieros

dataset	stock_1	stock_2
dataset_1	aapl	^gspc
dataset_2	mcd	pg
dataset_3	xle	ewh
dataset_4	xlfi	qqq
dataset_5	dia	spy

Tabla 12. Resultado de cada algoritmo en su respectiva evaluación

Date	Open	High	Low	Close	Adj Close	Volume	Etiqueta	position	action	holding	cash	Money	Return
2006-04-05	2.311	2.400	2.291	2.400	2.0495009	2233408800.	comprar		retener	0.0	100	100.0	0.0
2006-04-06	2.439	2.573	2.435	2.544	2.1723928	2663768800.	retener		retener	0.0	100	100.0	0.0
2006-04-07	2.533	2.543	2.445	2.492	2.1281762	1545238800.	retener		retener	0.0	100	100.0	0.0
2006-04-10	2.510	2.533	2.444	2.452	2.0940229	903515200.0	retener		retener	0.0	100	100.0	0.0

Estas evaluaciones se utilizaron para obtener la media aritmética y la desviación estándar de la columna money para formar un solo CSV por cada activo financiero (*aapl*, *^gspc*, *mcd*, *pg*, *xle*, *ewh*, *xlfi*, *qqq*, *dia* y *spy*) y dataset (*dataset_1*, *dataset_2*, *dataset_3*, *dataset_4* y *dataset_5*) como se muestra en la **Tabla 13**.

Tabla 13. Media aritmética y desviación estándar de cada activo financiero y dataset

Date	mean	mean + std	mean - std
2006-03-28	100.0	102.28828855985167	97.71171144014833
2006-03-29	100.0	102.28828855985167	97.71171144014833
2006-03-30	99.90139287662502	102.18968143647669	97.61310431677336
2006-03-31	99.86853954315185	102.15682810300352	97.58025098330019

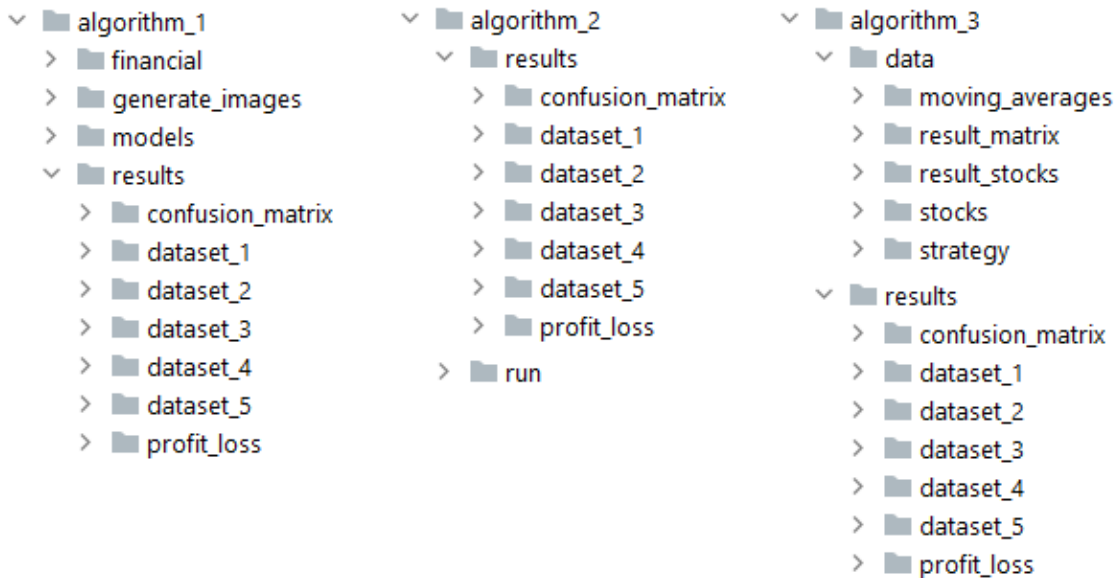
También se creó un archivo CSV para la matriz de confusión de cada activo financiero y conjunto de datos, lo que proporciona métricas adicionales para evaluar el rendimiento de los algoritmos. A continuación, se describen los algoritmos anteriormente mencionados e implementados en el framework, los cuales están organizados en su paquete (carpeta) y sub-paquetes conforme se muestra en la **Figura 9**.

3.5.1 Redes neuronales convolucionales profundas

Este algoritmo corresponde a la referencia [44] de 2018 publicada en la revista Applied Soft Computing de Elsevier, la cual contó con un factor de impacto de 4.873 y la categoría Q1 del JCR ese año, siendo la revista número 20 de 134 en el tema de Inteligencia Artificial (Ciencias de la Computación).

Para ejecutar esta técnica, se requieren los parámetros previamente mencionados, así como un archivo CSV que contenga los 15 indicadores financieros previamente calculados durante 15 periodos que oscilan entre los 6 y 20 días. También se necesita el archivo CSV previamente obtenido que contiene las etiquetas de entrenamiento y validación.

Figura 9. Organización general de cada uno de los algoritmos



Se recorre el DataFrame presentado en la **Tabla 10** para obtener el nombre del dataset y el stock que se utilizarán en el proceso de generación de las imágenes del conjunto de datos de entrenamiento y de validación. Posteriormente se usa la matriz de los 15 indicadores financieros con 15 periodos, lo que resulta en 225 valores por día. Estos valores se utilizan para crear una nueva matriz que, al ser representada como una imagen, tiene una resolución de 15x15 píxeles en formato PNG (Portable Network Graphics) tal como se muestra en la **Figura 10**.

Las imágenes creadas anteriormente se etiquetan adecuadamente como 'comprar', 'vender' o 'retener', según la fecha correspondiente a los indicadores técnicos calculados, tal como se muestra en la **Figura 11**.

Cada imagen se almacena en la carpeta correspondiente de comprar, vender o retener según el activo de cada dataset, tal como se muestra en la **Figura 12**.

Después de crear las imágenes del conjunto de datos de entrenamiento y validación, se procede a entrenar la red neuronal convolucional que incluye las siguientes capas:

- La capa convolucional, que realiza una operación de convolución según se muestra en la Ecuación (24). Esta operación se implementa en imágenes bidimensionales de acuerdo con la Ecuación (25).

- Las capas de convolución y Max Pooling consecutivas forman la estructura de la red neuronal profunda, que se denota en la ecuación (26). Al final de la red, se utiliza la función *softmax* para obtener la salida que se muestra en la ecuación (27) [44].

Figura 10. Imágenes generadas utilizando los datos de indicadores financieros previamente calculados con diferentes periodos

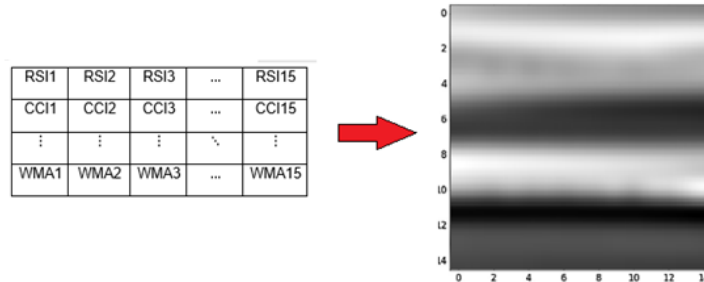


Figura 11. Imágenes de 15x15 pixeles debidamente etiquetadas

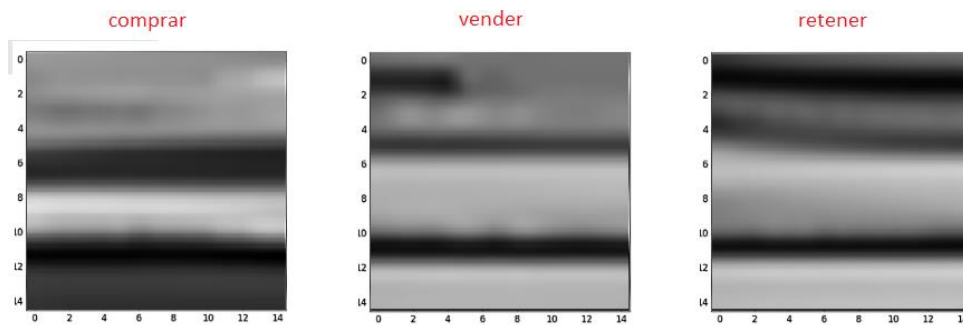
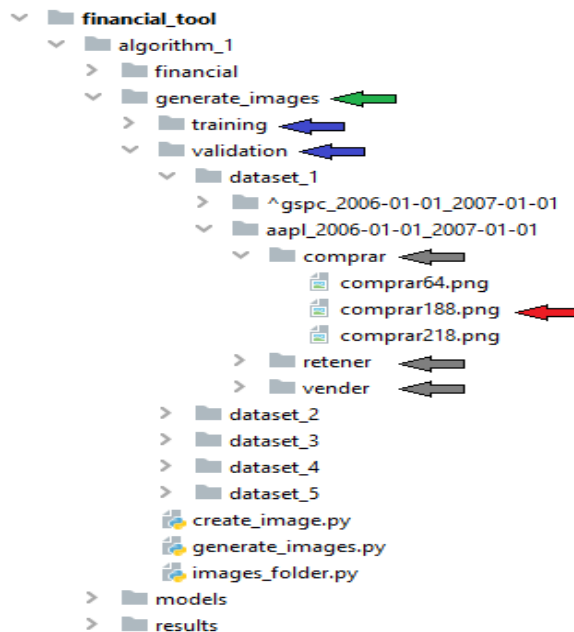


Figura 12. Ubicación de cada imagen en su carpeta correspondiente de comprar, retener o vender



$$s(t) = (x \cdot w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \quad (24)$$

Donde t denota tiempo

$$S(i,j) = (I \cdot K)(i,j) = \sum_m \sum_n I(m,n)K(i-m,j-n) \quad (25)$$

Donde K denota el núcleo

$$e_i = \sum_j w_{i+j} x_j + b_i \quad (26)$$

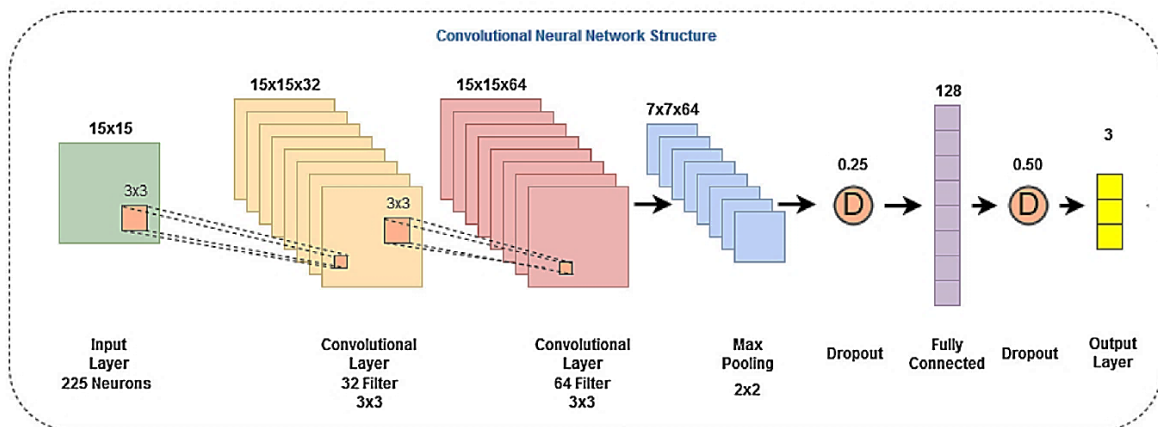
Donde w denota pesos, x denota entradas y b denota sesgos

$$y = \text{softmax}(e) \quad (27)$$

Donde y denota la salida

La estructura de esta red neuronal convolución utiliza ocho capas (ver **Figura 13**), a saber: Una capa de entrada (15×15), una capa convolucional ($15 \times 15 \times 32$), una capa convolucional ($15 \times 15 \times 64$), una capa Max Pooling ($7 \times 7 \times 64$), una capa de Dropout (0.25), una capa de conexión completa (128), una capa de Dropout (0.5) y una capa de salida (3).

Figura 13. Estructura de red neuronal convolucional implementada de la siguiente referencia (Tomada de [44])



La estructura de esta red neuronal convolucional se entrena según el número de iteraciones y épocas que el usuario establece para generar un modelo por cada activo financiero, el cual se almacena en la carpeta *'models'* para su posterior evaluación. Para el desarrollo de este algoritmo se usó la API keras. La organización del almacenamiento de los modelos resultantes de este algoritmo se muestra en la **Figura 14**.

El modelo generado se utiliza con cada imagen diaria de la línea de tiempo del conjunto de datos de evaluación, teniendo en cuenta la estrategia de inversión correspondiente. De esta manera, se obtiene el valor invertido por cada día en esta línea de tiempo, junto con su activo financiero correspondiente. Posteriormente, esta información se almacena en un archivo CSV para cada iteración, tal como se muestra en la **Figura 15**.

Figura 14. Ubicación de los modelos generados para cada activo financiero utilizando la API keras

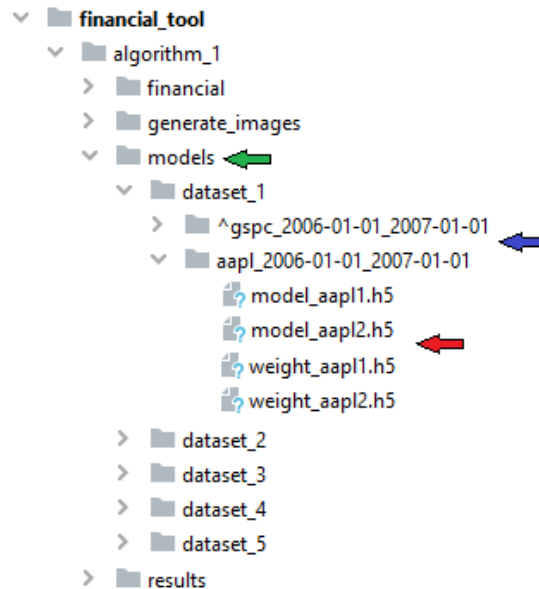
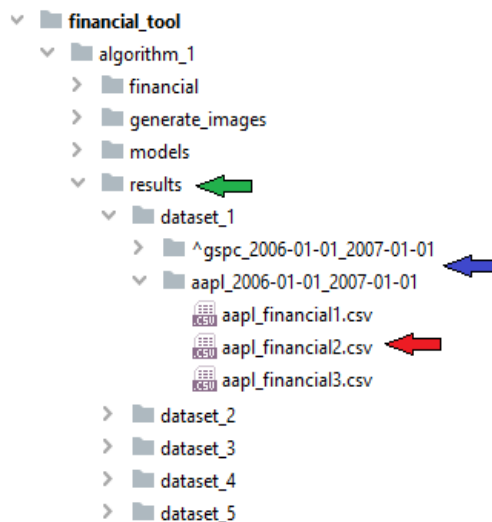


Figura 15. Ubicación de archivos csv generados por cada iteración de inversión en la línea de tiempo de evaluación para el algoritmo 1



A continuación, la **Tabla 14** muestra los datos contenidos en el archivo de salida por cada iteración mencionada anteriormente.

Tabla 14. Datos en cada archivo CSV con su respectiva evaluación para el algoritmo 1

action	holding	cash	Money	Returns
retener	0.0	100	100.0	0.0
retener	0.0	100	100.0	0.0
retener	0.0	100	100.0	0.0
retener	0.0	100	100.0	0.0

3.5.2 Aprendizaje por refuerzo profundo

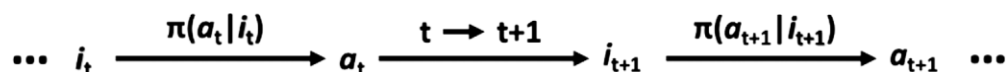
Este algoritmo corresponde a la referencia [54] de 2021 publicada en la revista Expert Systems With Applications de Elsevier, la cual contó con un factor de impacto de 8.665 y la categoría Q1 del JCR ese año, siendo la revista número 21 de 145 en el tema de Inteligencia Artificial (Ciencias de la Computación).

Este enfoque se centra en el diseño de una nueva política de trading basada en técnicas de inteligencia artificial con el objetivo de competir con estrategias de trading algorítmico ampliamente utilizadas. En este estudio se presenta y analiza una solución de DRL (Aprendizaje Profundo por Reforzamiento) que aborda el desafío del trading algorítmico, específicamente la determinación de la posición óptima de trading en el mercado de valores. La solución propuesta se basa en el algoritmo DQN (Deep Q-Network), adaptado para abordar el problema secuencial de toma de decisiones en el contexto del mercado de valores, utilizando algún data frame de la **Tabla 11** que contiene los activos financieros para operar. Las operaciones involucran intercambios de efectivo y acciones, y el agente interactúa con el mercado a través de un libro de órdenes. Para que se produzca un intercambio, se requiere una coincidencia entre las órdenes de oferta y demanda, y el agente debe tomar decisiones complejas para generar ganancias.

3.5.2.1 Discretización de la línea de tiempo

En este estudio, se analiza el problema del comercio algorítmico en pasos de tiempo discretos de duración constante Δt . Para modelar la transición entre los pasos de tiempo, se utilizan operaciones de incremento o decremento. La frecuencia de comercio objetivo es diaria, por lo que el agente de comercio toma una decisión una vez al día. La estrategia de comercio utilizada es una política programada $\pi(a_t|i_t)$, que genera una acción de comercio a_t basada en la información disponible i_t en el paso de tiempo t . Su naturaleza secuencial se ilustra en la **Figura 16**.

Figura 16. Ilustración de la ejecución de una estrategia de trading (Tomada de [54])



Un agente que ejecuta su estrategia de comercio aplica secuencialmente los siguientes pasos:

- Actualización de la información de mercado disponible i_t .

- Ejecución de la política $\pi(\mathbf{a}_t|\mathbf{i}_t)$ para obtener la acción de comercio \mathbf{a}_t .
- Aplicación de la acción de comercio designada \mathbf{a}_t .
- En el siguiente paso de tiempo $t \rightarrow t + 1$, volver al paso 1.

3.5.2.2 Formalización del problema de aprendizaje por refuerzo

En la **Figura 17** se muestra cómo el aprendizaje por refuerzo aborda la interacción secuencial entre un agente y su entorno. En cada paso de tiempo t , el agente RL (basado en aprendizaje por refuerzo o Reinforcement Learning) observa el estado interno del entorno, \mathbf{S}_t , y recupera una observación \mathbf{o}_t . Luego ejecuta una acción \mathbf{a}_t a través de su política $\pi(\mathbf{a}_t|\mathbf{h}_t)$, donde $\mathbf{h}_t = \{(\mathbf{o}_\tau, \mathbf{a}_\tau, \mathbf{r}_\tau) | \tau = 0, 1, \dots, t\}$ es el historial del agente RL y recibe una recompensa \mathbf{r}_t como consecuencia de su acción. Las técnicas de aprendizaje por refuerzo buscan diseñar políticas π que maximicen un criterio de optimización, que depende directamente de las recompensas inmediatas \mathbf{r}_t observadas en un horizonte temporal determinado. La política óptima π^* se expresa matemáticamente como resultado de este proceso (ver las Ecuaciones (28) y (29)).

$$\pi^* = \operatorname{argmax} \mathbb{E}[R|\pi] \quad (28)$$

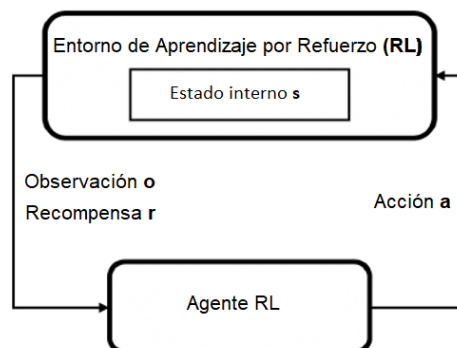
$$R = \sum_{t=0}^{\infty} \gamma^t r_t \quad (29)$$

El factor de descuento ($\gamma \in [0, 1]$) determina la importancia de las recompensas futuras en el aprendizaje por refuerzo. Un valor bajo de γ ($\gamma = 0$) indica que el agente de RL es miope y solo considera la recompensa actual, mientras que un valor alto de γ ($\gamma = 1$) indica un enfoque más orientado a largo plazo. Este parámetro se ajusta según el comportamiento deseado.

3.5.2.3 Observaciones de RL

El problema de negociación algorítmica presenta un desafío importante debido a la limitada observabilidad del entorno, ya que se oculta información relevante para el agente de negociación.

Figura 17. Bloques básicos fundamentales del aprendizaje por refuerzo (Tomado de [54])



En este sentido, el agente RL enfrenta un desafío importante debido a esta limitada información disponible en el entorno de negociación. La información recopilada puede ser cuantitativa o cualitativa y procesarla adecuadamente es crucial para minimizar el sesgo subjetivo. En cada paso de tiempo, el agente observa el mercado cuyo estado interno es $s_t \in \mathcal{S}$, y la información limitada recopilada se denota como $o_t \in \mathcal{O}$. Una observación o_t se considera una secuencia de τ pasos de tiempo tanto de información previa como de información recién disponible. Las observaciones pueden expresarse matemáticamente conforme la Ecuación (30).

$$o_t = \{S(t'), D(t'), T(t'), I(t'), M(t'), N(t'), E(t')\}_{t'=t-\tau}^t$$

Donde:

$S(t)$ representa la información del estado del agente RL en el paso de tiempo t (posición actual de negociación, número de acciones en propiedad del agente y efectivo disponible).

$D(t)$ es la información recopilada por el agente en el paso de tiempo t con respecto a los datos **OHLCV** (**O**pen – **H**igh – **L**ow – **C**lose – **V**olume) que caracterizan el mercado de valores. Más precisamente, $D(t)$ puede expresarse según la ecuación (31).

$T(t)$ es la información del agente sobre el paso de tiempo de negociación t (fecha, día de la semana, hora).

$I(t)$ es la información del agente sobre múltiples indicadores técnicos del mercado de valores que se enfocan en el paso de tiempo t . (30)

$M(t)$ recopila la información macroeconómica a disposición del agente en el paso de tiempo t .

$N(t)$ representa la información de noticias recopilada por el agente en el paso de tiempo t . Con estos datos de noticias se podrían construir modelos complejos de análisis de sentimientos para extraer cifras cuantitativas significativas de las noticias.

$E(t)$ es cualquier información extra útil a disposición del agente de trading en el paso de tiempo t , como las estrategias de negociación de otros participantes del mercado, información confidencial de empresas, comportamientos similares en el mercado de valores, rumores, consejos de expertos, etc.

$$D(t) = \{p_t^o, p_t^H, p_t^L, p_t^C, V_t\}$$

Donde:

p_t^o es el precio de mercado de valores en la apertura del período de tiempo.

p_t^H es el precio de mercado de valores más alto en el período de tiempo. (31)

p_t^L es el precio de mercado de valores más bajo en el período de tiempo.

p_t^C es el precio de mercado de valores en el cierre del período de tiempo.

V_t es el volumen total de acciones intercambiadas durante el período de tiempo.

3.5.2.4 Reducción del espacio de observación

En esta investigación, el agente RL solo considera la información **OHLCV** de $D(t)$ y la información de estado $S(t)$. El espacio de observación reducido O abarca la posición de negociación actual y una serie de precios ($p_t^o, p_t^H, p_t^L, p_t^C$) y volumen negociado diario anterior. La observación reducida o_t puede expresarse como la Ecuación (32).

$$o_t = \{ \{ p_t^o, p_t^H, p_t^L, p_t^C, V_t \}_{t'=t-\tau}^t, P_t \} \quad (32)$$

Donde P_t es la posición de trading del agente RL en el paso de tiempo t .

3.5.2.5 Acciones de RL

En cada paso de tiempo t , el agente RL ejecuta una acción de trading en $a_t \in A$, como resultado de su política $\pi(a_t|h_t)$. Las decisiones pueden ser modeladas por la cantidad de acciones compradas por el agente de trading en el tiempo t , representada por $Q_t \in Z$. Por lo tanto, las acciones del agente RL se pueden expresar conforme la Ecuación (33).

$$a_t = Q_t$$

Pueden ocurrir tres casos según el valor de Q_t :

- $Q_t > 0$: El agente RL compra acciones en el mercado de valores. (33)
- $Q_t < 0$: El agente RL vende acciones en el mercado de valores.
- $Q_t = 0$: El agente RL se retiene, es decir, no compra ni vende ninguna acción en el mercado de valores.

3.5.2.6 Reducción del espacio de acción

Con el objetivo de simplificar el problema de negociación algorítmica, el espacio de acción original A se limita a solo dos opciones de RL. Estas acciones se representan matemáticamente con la Ecuación (34).

$$a_t = Q_t \in \{ Q_t^{Long}, Q_t^{Short} \} \quad (34)$$

La primera acción de RL designada como Q_t^{Long} , maximiza la cantidad de acciones que posee el agente de negociación, convirtiendo tanto valor efectivo v_t^f como sea posible en valor de acciones v_t^s . Se puede expresar matemáticamente conforme la ecuación (35). La acción Q_t^{Long} siempre es válida ya que está incluida obviamente en el espacio de acción original A . Como resultado de esta acción, el agente de negociación posee un número de acciones $N_t^{Long} = n_t + Q_t^{Long}$.

$$Q_t^{Long} = \begin{cases} \left| \frac{v_t^c}{p_t(1+C)} \right| & \text{si } a_{t-1} \neq Q_{t-1}^{Long} \\ 0 & \text{de lo contrario} \end{cases} \quad (35)$$

$$\widehat{Q}_t^{Short} = \begin{cases} -2n_t - \left| \frac{v_t^c}{p_t(1+C)} \right| & \text{si } a_{t-1} \neq Q_{t-1}^{Short} \\ 0 & \text{de lo contrario} \end{cases} \quad (36)$$

Por el contrario, la segunda acción de RL, designada por Q_t^{Short} , convierte el valor de las acciones v_t^s en valor en efectivo v_t^c , de modo que el agente RL posea un número de acciones igual a $-N_t^{Long}$. Esta operación se puede expresar matemáticamente conforme la Ecuación (36). Sin embargo, la acción \widehat{Q}_t^{Short} puede violar el límite inferior Q_t del espacio de acción original A cuando el precio aumenta significativamente con el tiempo. La acción de RL Q_t^{Short} se expresa conforme la Ecuación (37).

$$Q_t^{Short} = \max \{ \widehat{Q}_t^{Short}, Q_t \} \quad (37)$$

Para concluir esta subsección, se debe mencionar que las dos acciones reducidas de RL están relacionadas con la próxima posición de negociación del agente RL, designada como P_{t+1} . De hecho, la primera acción Q_t^{Long} induce una posición de negociación larga porque el número de acciones poseídas es positivo. Por el contrario, la segunda acción Q_t^{Short} siempre resulta en un número de acciones que es negativo, lo que generalmente se conoce en finanzas como una posición de negociación corta.

3.5.2.7 Recompensas de RL

Para este problema de trading algorítmico, una elección natural para las recompensas de RL son los rendimientos diarios de la estrategia. Intuitivamente, tiene sentido favorecer los rendimientos positivos que son evidencia de una estrategia rentable. Además, esta cantidad tiene la ventaja de ser independiente del número de acciones n_t actualmente poseídas por el agente. Esta elección también está motivada por el hecho de que permite evitar una configuración de recompensa dispersa, que es más compleja de manejar. Las recompensas de RL se pueden expresar matemáticamente conforme la Ecuación (38).

$$r_t = \frac{v_{t+1} - v_t}{v_t} \quad (38)$$

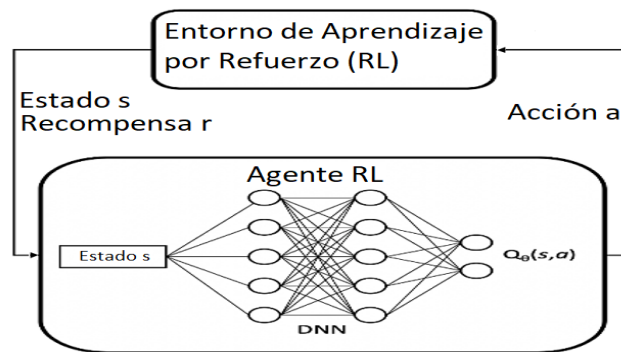
Donde v_t es el volumen total de acciones intercambiadas durante el período de tiempo t .

3.5.2.8 Algoritmo de Redes Neuronales Profundas-Q (Deep Q-Network)

El algoritmo DQN (Deep Q-Network) es un tipo de Deep Reinforcement Learning (DRL) que aprende políticas de control a partir de entradas sensoriales de alta dimensionalidad. No requiere un modelo completo del entorno, y se basa en el aprendizaje de una aproximación de la función de valor de estado-acción mediante una Deep Neural Network (DNN). Aprender la función Q equivale a aprender los parámetros θ de esta DNN, y el algoritmo DQN es fuera de política, ya que explota en modo de lote, experiencias previas $e_t = (s_t, a_t, r_t, s_{t+1})$ recopiladas en cualquier momento durante el entrenamiento [54]. A partir del anterior algoritmo la estrategia de negociación resultante se denomina algoritmo **TDQN** (Trading Deep Q-Network).

A continuación, en la **Figura 18** se ilustra el algoritmo DQN, pero no se presenta de manera exhaustiva su explicación en este documento (ver detalles en referencia [54]).

Figura 18. Ilustración del algoritmo DQN (Tomada de [54])



3.5.2.9 Generación de trayectorias artificiales

En el ámbito del problema de trading algorítmico, no se dispone de un modelo completo del entorno E . El entrenamiento del algoritmo **TDQN** se basa por completo en la generación de trayectorias artificiales a partir de un conjunto limitado de datos históricos **OHLCV** de los mercados de valores. Una trayectoria τ se define como una secuencia de observaciones $o_t \in \mathcal{O}$, acciones $a_t \in \mathcal{A}$ y recompensas r_t de un agente RL para un cierto número T de pasos de tiempo de negociación t : conforme la Ecuación (39).

$$\tau = (\{o_0, a_0, r_0\}, \{o_1, a_1, r_1\}, \dots, \{o_T, a_T, r_T\}) \quad (39)$$

Inicialmente, se tiene una sola trayectoria real del comportamiento histórico del mercado de valores para representar la inactividad del agente RL, ya que el ambiente E es desconocido. Se generan nuevas trayectorias ficticias a partir de esta única trayectoria verdadera para simular interacciones con el ambiente E . Estas nuevas trayectorias están compuestas por la secuencia de observaciones históricas reales asociadas con diversas secuencias de acciones de negociación del agente

RL. El agente de negociación no debe influir en el comportamiento del mercado de valores para que la simulación sea realista, lo que generalmente se cumple cuando el número de acciones negociadas por el agente es bajo en comparación con la liquidez del mercado.

Además de la generación de las trayectorias artificiales descritas, se utiliza un truco para mejorar ligeramente la exploración del agente RL. Se basa en el hecho de que el espacio de acción reducido A está compuesto solo por dos acciones: larga (Q_t^{Long}) y corta (Q_t^{Short}). En cada paso de tiempo de negociación t , se ejecuta la acción elegida a_t en el ambiente de negociación E y la acción opuesta a_t^- se ejecuta en una copia de este ambiente E^- . Aunque este truco no resuelve completamente el desafío de la compensación entre exploración/explotación, permite al agente RL explorar continuamente con un pequeño costo computacional adicional.

3.5.2.10 *Diversas modificaciones y mejoras al algoritmo DQN*

El algoritmo DQN fue elegido para la nueva estrategia de trading DRL desarrollada, pero fue adaptado al problema específico de toma de decisiones de trading algorítmico. Las diversas modificaciones y mejoras, que se basan principalmente en las numerosas simulaciones realizadas, se resumen a continuación:

- **Arquitectura de red neuronal profunda:** la primera diferencia con respecto al algoritmo DQN clásico es la arquitectura de la red neuronal profunda que aproxima la función de valor de acción $Q(s, a)$. Debido a la naturaleza diferente de la entrada (series de tiempo en lugar de imágenes sin procesar). La red neuronal convolucional (CNN) se reemplazó por una red neuronal clásica de alimentación directa con algunas funciones de activación de unidades rectificadas con fuga (Leaky, ReLU).
- **Doble DQN:** El algoritmo DQN tiene sobreestimaciones importantes que afectan su rendimiento. Para solucionar este problema, se implementó el algoritmo de doble DQN, basado en la descomposición de la operación máxima de destino en la selección y la evaluación de las acciones. Esta solución se presenta en el artículo de van Hasselt [55].
- **Optimizador ADAM:** El algoritmo DQN clásico usa el optimizador RMSProp. Sin embargo, el optimizador ADAM presentado por Kingma [56] demuestra experimentalmente mejorar tanto la estabilidad del entrenamiento como la velocidad de convergencia del algoritmo DRL.
- **Recorte de gradiente:** La técnica de recorte de gradiente se implementa en el algoritmo TDQN para resolver el problema de explosión de gradientes, que induce importantes inestabilidades durante el entrenamiento de la DNN.
- **Inicialización Xavier:** Mientras que el algoritmo DQN clásico inicializa los pesos de la DNN al azar, en este trabajo se implementa la inicialización Xavier para mejorar la convergencia del algoritmo. La idea es establecer los pesos iniciales de tal manera que la varianza de los gradientes permanezca constante en todas las capas de la DNN.

- **Capas de normalización por lotes:** Esta técnica de DL, introducida por Loffe y Szegedy [57], consiste en normalizar la capa de entrada ajustando y escalando las funciones de activación. Aporta muchos beneficios, incluyendo una fase de entrenamiento más rápida y robusta, así como una mejora en la generalización.
- **Técnicas de regularización:** Debido a una fuerte tendencia al sobreajuste observada durante los primeros experimentos con la estrategia de trading DRL, se implementaron tres técnicas de regularización: Dropout, regularización L2 y Early Stopping.
- **Preprocesamiento y normalización:** Para mejorar el rendimiento del algoritmo TDQN, se realizó una operación de preprocesamiento y normalización de las observaciones de RL \mathbf{o}_t . Primero realiza una operación de filtrado de paso bajo para reducir el ruido de alta frecuencia presente en los datos de trading, aunque esto puede tener un costo al modificar o incluso destruir algunos patrones de trading. Segundo, los datos resultantes se transforman para transmitir información más significativa sobre los movimientos del mercado, considerando la evolución diaria de los precios en lugar de los precios en bruto. Tercero, los datos restantes se normalizan para obtener una escala común y reducir el impacto de los valores atípicos.
- **Técnicas de aumento de datos:** Un desafío clave de este problema de trading algorítmico es la cantidad limitada de datos disponibles, que además son generalmente de baja calidad. Como contraparte a este problema importante, se implementan varias técnicas de aumento de datos: desplazamiento de señal, filtrado de señal y adición de ruido artificial. La aplicación de tales técnicas de aumento de datos generará artificialmente nuevos datos de trading que son ligeramente diferentes pero que dan lugar a los mismos fenómenos financieros.

El Algoritmo 3 se describe detalladamente el algoritmo subyacente a la estrategia de trading TDQN.

Algoritmo 3. Estrategia de trading TDQN (Tomado de [54])

- 1: Inicializar la memoria de repetición de experiencia M con capacidad C .
- 2: Inicializar los pesos θ principales de la DNN (inicialización Xavier).
- 3: Inicializar el objeto DNN con pesos $\theta^- = \theta$.
- 4: **for** episodio = 1 to N **do**
- 5: Obtener la observación inicial \mathbf{o}_1 del entorno \mathbf{E} y procesarla.
- 6: **for** $t = 1$ to T **do**
- 7: Con probabilidad ϵ , seleccionar una acción aleatoria \mathbf{a}_t de A .
- 8: De lo contrario, seleccionar $\mathbf{a}_t = \mathbf{argmax}_{\mathbf{a} \in A} Q(\mathbf{o}_t, \mathbf{a}; \theta)$
- 9: Copiar el entorno $\mathbf{E}^- = \mathbf{E}$
- 10: Interactuar con el entorno \mathbf{E} (acción \mathbf{a}_t) y obtener la nueva observación \mathbf{o}_{t+1} y la recompensa \mathbf{r}_t .
- 11: Realizar la misma operación en \mathbf{E}^- con la acción opuesta \mathbf{a}_t^- , para obtener \mathbf{o}_{t+1}^- y \mathbf{r}_t^- .
- 12: Preprocesar ambas nuevas observaciones \mathbf{o}_{t+1} y \mathbf{o}_{t+1}^- .

```

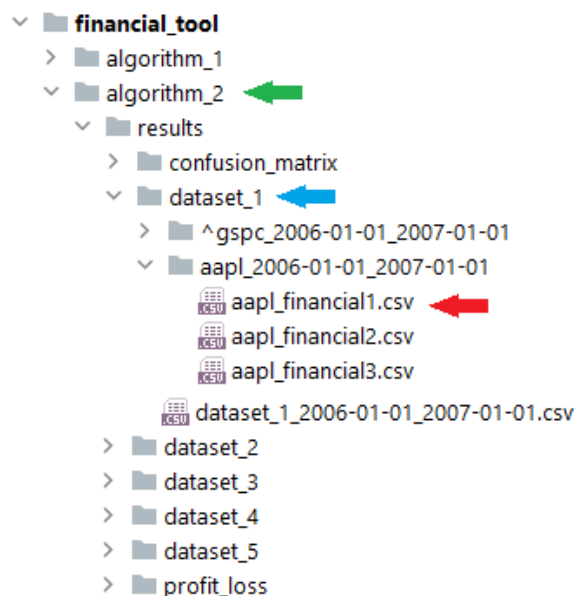
13:     Almacenar ambas experiencias  $e_t = (o_t, a_t, r_t, o_{t+1})$  y  $e_t^- = (o_t, a_t^-, r_t^-, o_{t+1}^-)$ 
    en  $M$ 
14:     if  $t \% T' = 0$  then
15:         Seleccionar aleatoriamente de  $M$  un miniconjunto de
            experiencias  $N_e$  denotado como  $e_t = (o_t, a_t, r_t, o_{t+1})$ 
16:         Establecer  $y_i =$ 
            
$$\begin{cases} r_i & \text{si el estado } s_{i+1} \text{ es terminal.} \\ r_i + +\gamma Q(o_{i+1}, \text{argmax}_{a \in A} Q(o_{i+1}, a; \theta^-)) & \text{De lo contrario} \end{cases}$$

16:         Calcular y recortar los gradientes basados en la pérdida de
            Huber  $H(y_i, Q(o_i, a_i; \theta))$ .
17:         Optimizar los parámetros principales  $\theta$  de la DNN basados en
            gradientes en estos gradientes recortados.
18:         Actualizar los parámetros  $\theta^- = \theta$  del objeto DNN
            a partir de cada  $N^-$  pasos.
19:     end if
20:     Disminuir gradualmente el parámetro de exploración  $\epsilon$  – Greedy
21: end for
22: end for

```

Efectuadas las modificaciones al algoritmo DQN se obtiene la estrategia TDQN para ser debidamente entrenado y evaluado hasta obtener su estrategia óptima de inversión con resultados de inversión diarios para cada activo financiero; esta información queda debidamente consignada en un archivo csv para cada iteración como se muestra en la **Figura 19**.

Figura 19. Ubicación de archivos csv generados por cada iteración de inversión en la línea de tiempo de evaluación para el algoritmo 2



A continuación se muestra la información contenida por cada iteración del algoritmo mencionado anteriormente en la **Tabla 15**.

Tabla 15. Datos en cada archivo CSV en su respectiva evaluación para el algoritmo 2

Action	Holdings	Cash	Money	Returns
0	101.372879	1.29426950	102.667149	0.02078012
0	99.5971250	1.29426950	100.891394	-0.0172962
0	99.2357091	1.29426950	100.529978	-0.0035822
0	99.5342874	1.29426950	100.828556	0.00297004
-1	-101.08999	203.272080	102.182084	0.01342405
0	-99.864295	203.272080	103.407785	0.01199525
0	-96.202877	203.272080	107.069203	0.03540756
1	108.866796	0.27819679	109.144993	0.01938737

3.5.3 LSTM (Long Short-Term Memory)

Este algoritmo corresponde a la referencia [58] de 2019 publicada en la revista ACM Transactions on Information Systems de ACM, la cual contó con un factor de impacto de 2.889 y la categoría Q2 del JCR ese año, siendo la revista número 61 de 156 en el tema de Sistemas de Información (Ciencias de la Computación).

Para llevar a cabo las predicciones de los activos financieros se utilizó la técnica de Redes Neuronales de Memoria a Corto y Largo Plazo (LSTM), se implementaron los siguientes pasos con el fin de obtener resultados precisos y confiables.

En primer lugar, se recopilaron y prepararon los datos necesarios para el entrenamiento y la evaluación del modelo. Se adquirió el conjunto de datos históricos de acciones, específicamente del periodo de entrenamiento, el cual se encontraba almacenado en un archivo llamado "trading.csv" que contiene lo que se muestra en la **Tabla 8**. Luego, se procedió a concatenar las fechas correspondientes al periodo de entrenamiento y validación en un archivo denominado "date.csv". Además, se extrajeron los nombres de cada activo financiero y se guardaron en un archivo llamado "name_stock.csv", tal como se muestra en la **Tabla 16**.

A continuación, se realizó la fase de acondicionamiento de los datos de entrada, en la cual se calcularon las medias móviles de 5, 10, 15 y 30 días. Estas medias móviles fueron normalizadas y almacenadas en un archivo CSV para cada activo financiero. El archivo contiene la concatenación de los datos de entrenamiento y validación, como se muestra en la **Tabla 17**. La ubicación de esta se muestra en la **Figura 20** para cada activo financiero.

Utilizando los datos anteriores para la predicción de acciones, comúnmente se plantea un escenario en el que se busca clasificar el movimiento de precios y realizar una regresión sobre estos. El objetivo consiste en aprender una función de predicción, representada por $\hat{y}^{t+1} = f(X^t)$ que mapee una acción desde el espacio de características al espacio de etiquetas objetivo en un momento específico, t.

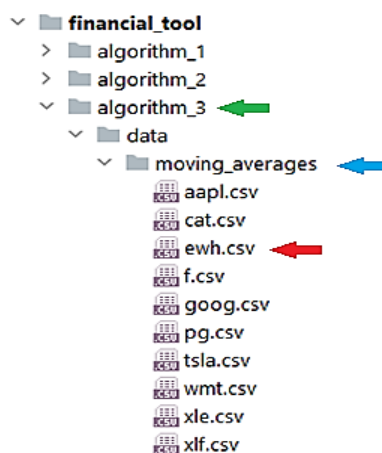
Tabla 16. Los nombres de cada activo financiero se encuentran ordenados, desde el dataset 1 hasta el dataset 5

stock	dataset
aapl	dataset_1
^gspc	dataset_1
mcd	dataset_2
pg	dataset_2
xle	dataset_3
ewh	dataset_3
xlif	dataset_4
qqq	dataset_4
dia	dataset_5
spy	dataset_5

Tabla 17. Medias móviles de 5, 10, 15 y 30 días de cada activo financiero con la columna del precio de cierre normalizado

0.287119	0.282135	0.283774	0.289080	0.289033
0.287973	0.283505	0.283318	0.288638	0.287238
0.287985	0.284212	0.283508	0.288156	0.285458
0.286179	0.284630	0.283080	0.287448	0.280672
0.285640	0.285267	0.282704	0.286999	0.285800

Figura 20. Ubicación del archivo csv para las medias móviles calculadas de cada activo financiero



En este escenario, la matriz $\mathbf{X}^t = [\mathbf{x}^{t+S+1}, \dots, \mathbf{x}^t]^T \in \mathbb{R}^{S \times D}$ representa las características de entrada secuenciales. Aquí, \mathbf{D} indica la dimensión de las

características en cada paso de tiempo, mientras que \mathbf{S} se refiere a la longitud de la secuencia.

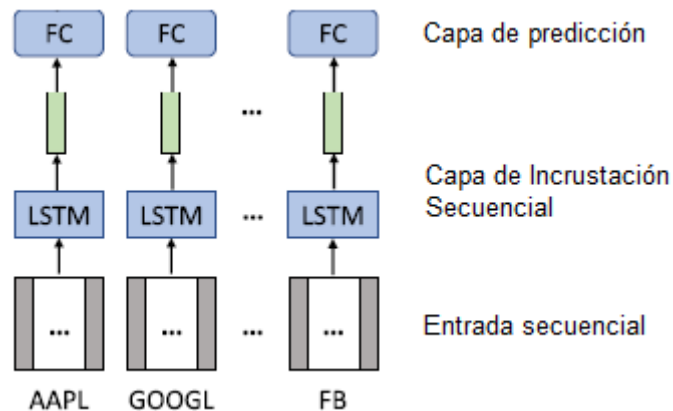
Para una mejor comprensión, se han resumido algunos de los términos y notaciones utilizados en la **Tabla 18**.

Tabla 18. Símbolos y definiciones utilizados en la técnica (tomado de [58])

Símbolos	Definición
$\mathbf{X}^t \in \mathbb{R}^{N \times S \times D} = [\mathbf{X}_1^t, \dots, \mathbf{X}_N^t]^T$	Precios históricos de N acciones en el día de negociación t .
$\mathbf{E}^t = [e_1^t, \dots, e_N^t]^T \in \mathbb{R}^{N \times U}$	Incrustación secuencial de N acciones aprendidas a partir de precios históricos.
$r^{t+1}, \hat{r}^{t+1} \in \mathbb{R}^N$	Puntuaciones de clasificación verdaderas y predichas de N acciones.
w, b	Pesos y sesgo para aprender.

En la técnica que se describe a continuación, se utilizan dos capas: la capa de incrustación secuencial y la capa de predicción. Estas capas se pueden visualizar en la **Figura 21**.

Figura 21. Ilustración general de las capas utilizadas en esta última técnica (tomado de [58])



3.5.3.1 Capa de incrustación secuencial

Con el fin de predecir la tendencia futura de una acción, se considera que su historial es el factor más influyente debido a la naturaleza temporal del mercado de valores. Para capturar las dependencias secuenciales presentes en los datos históricos, se aplica una capa de incrustación secuencial utilizando una LSTM, la cual ha demostrado un rendimiento significativo en la predicción de acciones [59]. La elección de LSTM se basa en su capacidad para capturar dependencias a largo plazo, lo cual es importante debido a que muchos factores tienen efectos prolongados en una acción.

En esta metodología, los datos de series temporales históricas de la acción i en el paso de tiempo t (\mathbf{X}_i^t) se introducen en una red LSTM. Luego, se toma el último

estado oculto (h_i^t) como la incrustación secuencial (e_i^t) de dicha acción (cabe destacar que $e_i^t = h_i^t$). En resumen, se obtiene la ecuación (40).

$$E^t = LSTM(X^t)$$

Donde $E^t = [e_1^t, \dots, e_N^t] \in \mathbb{R}^{N \times U}$ representa las incrustaciones secuenciales de (40) todas las acciones, y U representa el tamaño de la incrustación (es decir, U es el número de unidades ocultas en LSTM).

3.5.3.2 Capa de Predicción

Por último, se hace la predicción del precio de cierre de cada activo financiero con 1 día de antelación utilizando precios históricos como entrada. Los autores consideraron el precio histórico como una señal y desarrollaron una extensión de LSTM que descompone los precios históricos en el dominio de frecuencia mediante una Transformada Discreta de Fourier (DFT) y asigna a cada frecuencia un estado de memoria para capturar los patrones en diferentes frecuencias [59].

Esta página se ha dejado intencionalmente en blanco.

CAPÍTULO 4

4 Evaluación y análisis comparativo de los modelos de trading algorítmico

En este capítulo, se resume la evaluación y el análisis comparativo de los tres algoritmos de trading que se han implementado en el framework desarrollado en el marco de esta investigación. Este framework cuenta con una interfaz gráfica, la cual se muestra en la **Figura 6**. Para llevar a cabo la evaluación y comparación se han tomado los siguientes valores dentro del framework:

- Algorithm to Evaluate: All Algorithms
- Training Date: 01/01/2002 a 01/01/2006
- Validation Date: 01/01/2006 a 01/01/2007
- Max Repetition: 30
- Money: 100000
- Parameters Algorithm 1: 1
- Parameters Algorithm 2: 1

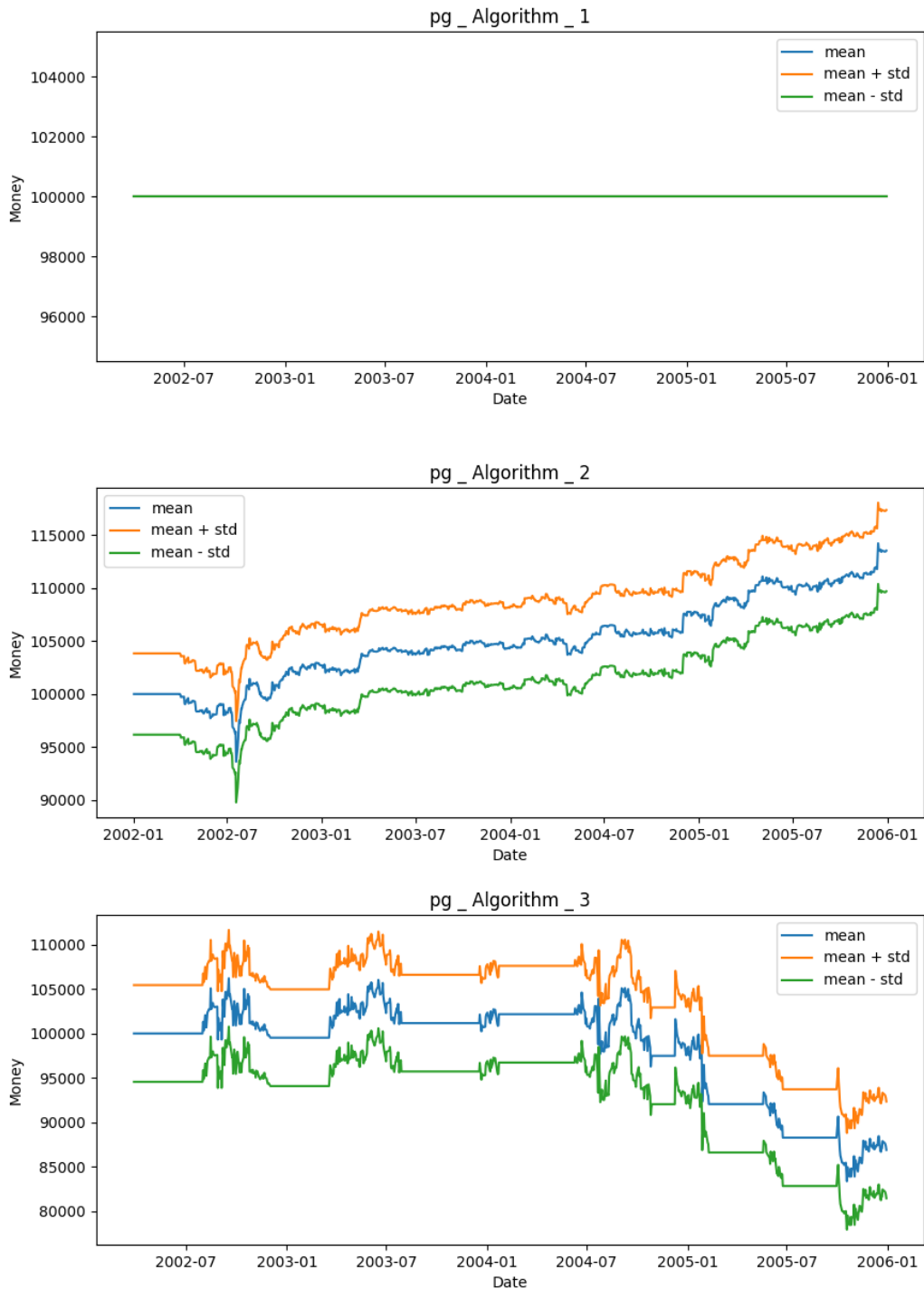
A continuación, se presentarán los resultados obtenidos con base en los parámetros seleccionados en el framework. Estos resultados proporcionarán una visión comparativa y detallada del desempeño de los algoritmos de trading bajo análisis. Mediante el análisis de estos resultados, se puede determinar cuál de los algoritmos presenta un mejor rendimiento en el contexto de este estudio.

4.1 Resultados obtenidos de profit_loss de los algoritmos implementados

Al ejecutar la evaluación de los algoritmos, se obtienen los resultados de las pérdidas o ganancias (profit_loss), los cuales se registran en el archivo correspondiente a la inversión financiera a lo largo de la línea de tiempo de evaluación. Esto se puede apreciar visualmente en la **Figura 22**, que muestra de manera gráfica la evolución del dinero invertido. Es importante tener en cuenta estos registros para analizar el rendimiento de los algoritmos y tomar decisiones informadas en función de los resultados obtenidos.

En los resultados obtenidos para el primer algoritmo, se observa que ninguno de los activos financieros presentó pérdidas o ganancias, según se muestra en la **Tabla 19**. Este hecho se atribuye a la naturaleza conservadora del algoritmo evaluado, ya que la mayoría de las imágenes utilizadas en el análisis correspondían a situaciones de retención.

Figura 22. Evolución del dinero invertido para el activo financiero “pg” en cada uno de los tres algoritmos implementados



La identificación de la ausencia de pérdidas o ganancias en el primer algoritmo abre la oportunidad de mejorar estrategias o explorar alternativas que puedan generar resultados más favorables en futuras investigaciones. Los hallazgos obtenidos

brindan una base sólida para la toma de decisiones informadas y la optimización del desempeño de los algoritmos en el contexto financiero.

Tabla 19. Resultados obtenidos de ganancias o pérdidas porcentuales para el algoritmo 1 (redes neuronales convolucionales) en cada uno de los activos financieros según su dataset

stock	profit_loss	dataset
aapl	0.0	dataset_1
^gspc	0.0	dataset_1
mcd	0.0	dataset_2
pg	0.0	dataset_2
xle	0.0	dataset_3
ewh	0.0	dataset_3
xlf	0.0	dataset_4
qqq	0.0	dataset_4
dia	0.0	dataset_5
spy	0.0	dataset_5

Como resultado del algoritmo 2, se puede apreciar que el activo "pg" obtuvo la mayor ganancia porcentual, 13.52%, mientras que el activo "aapl" registró una pérdida de -15.74%, tal como se muestra en la **Tabla 20**. Estos resultados indican la capacidad del algoritmo 2 para identificar oportunidades de inversión favorables y riesgos asociados en diferentes activos financieros.

Tabla 20. Resultados obtenidos de ganancias o pérdidas porcentuales para el algoritmo 2 (aprendizaje por refuerzo profundo) en cada uno de los activos financieros según su dataset

stock	profit_loss	dataset
aapl	-15.742929	dataset_1
^gspc	2.20173753	dataset_1
mcd	-6.5091451	dataset_2
pg	13.5211140	dataset_2
xle	8.84300518	dataset_3
ewh	-7.0241381	dataset_3
xlf	-10.841670	dataset_4
qqq	-5.7362416	dataset_4
dia	9.44595065	dataset_5
spy	-2.2600042	dataset_5

Además, al analizar los diferentes conjuntos de datos utilizados, se concluye que el algoritmo 2 logró ganancias en el dataset 2 con un incremento del 7.01%, en el dataset 3 con un aumento del 1.81% y en el dataset 5 con un crecimiento del 7.18%. Estos hallazgos demuestran la capacidad del algoritmo para adaptarse a diversas condiciones del mercado y generar rendimientos positivos en múltiples contextos.

Estos resultados resaltan la importancia de evaluar y comparar el desempeño de distintos algoritmos en función de su capacidad para identificar oportunidades de ganancia y mitigar riesgos. Asimismo, brindan información valiosa para tomar

decisiones informadas en el ámbito financiero y optimizar las estrategias de inversión.

Finalmente, al analizar el desempeño del algoritmo 3, los resultados de su evaluación reflejan únicamente pérdidas porcentuales, que varían desde -10.00% para el activo "xle" hasta un preocupante -55.43% para "aapl", tal como se detalla en la **Tabla 21**.

Estos hallazgos resaltan la importancia de comprender las limitaciones y riesgos asociados con el algoritmo 3 en el ámbito financiero, lo que permite tomar decisiones más cautelosas y considerar cuidadosamente su implementación. Es necesario profundizar en el análisis de las causas detrás de estas pérdidas significativas y, en futuras investigaciones, se sugiere explorar posibles ajustes para mejorar el rendimiento del algoritmo o considerar alternativas más eficaces en busca de resultados más favorables.

Tabla 21. Resultados obtenidos de ganancias o pérdidas porcentuales para el algoritmo 3 (red LSTM) en cada uno de los activos financieros según su dataset

stock	profit_loss	dataset
aapl	-55.430426	dataset_1
^gspc	-17.330359	dataset_1
mcd	-36.771655	dataset_2
pg	-13.105618	dataset_2
xle	-10.000402	dataset_3
ewh	-39.038996	dataset_3
xlfi	-14.057350	dataset_4
qqq	-26.428848	dataset_4
dia	-11.703141	dataset_5
spy	-20.154221	dataset_5

En resumen, los resultados obtenidos de los tres algoritmos evaluados revelan escenarios diferentes en términos de ganancias y pérdidas. Mientras que el primer algoritmo no mostró pérdidas ni ganancias, lo que sugiere una naturaleza conservadora, el segundo algoritmo destacó con una ganancia del 13.52% y la capacidad de identificar oportunidades de inversión favorables en activos financieros específicos. Por otro lado, el tercer algoritmo registró únicamente pérdidas, con rangos que van desde -10.00% hasta un preocupante -55.43%. Estos resultados subrayan la importancia de considerar ajustes y alternativas más efectivas para mejorar su desempeño y mitigar las pérdidas.

4.1.1 Resultados de la matriz de confusión

Después de realizar la evaluación de pérdidas o ganancias, se generó una matriz de confusión para cada activo financiero en cada conjunto de datos que incluye dos activos financieros, junto con sus respectivas métricas, como se detalla a continuación en la **Tabla 22** y en la

Tabla 23.

Tabla 22. Matriz de confusión del algoritmo_1 para el activo financiero "aapl" y también para el dataset_1 que contiene a los activos "aapl" con "gspc"

```

- - - Matrix confusion - - -> aapl      - - Matrix confusion - - -> dataset_1

algorithm_1  comprar  vender  retener  algorithm_1  comprar  vender  retener
comprar      0         0       1680    comprar      0         0       3150
vender       0         0       1770    vender       0         0       3390
retener      0         0       24990   retener      0         0       50340
    
```

Tabla 23. Métricas obtenidas a partir de las matrices de confusión del activo financiero "aapl" y también para el dataset_1 que contiene a los activos "aapl" con "gspc"

```

- - -      Metric      - - -> aapl

algorithm_1  comprar  vender      retener      macro avg  weighted avg  accuracy
precision    0.0     0.0     0.878692    0.292897    0.772100    0.878692
recall       0.0     0.0     1.000000    0.333333    0.878692
f1-score     0.0     0.0     0.935430    0.311810    0.821954
support     1680.0  1770.0  24990.000000  28440.000000  28440.000000

- - -      Metric      - - -> dataset_1

algorithm_1  comprar  vender      retener      macro avg  weighted avg  accuracy
precision    0.0     0.0     0.885021    0.295007    0.783262    0.885021
recall       0.0     0.0     1.000000    0.333333    0.885021
f1-score     0.0     0.0     0.939004    0.313001    0.831038
support     3150.0  3390.0  50340.000000  56880.000000  56880.000000
    
```

Después de calcular las métricas mencionadas anteriormente (precisión, recall y F1-score) para cada conjunto de datos, se procedió a organizarlos en una matriz. Cada fila de esta matriz representa los resultados de cada algoritmo, comenzando con el algoritmo 1 y finalizando con el algoritmo 3. A continuación, se presentan los resultados de estas métricas en la **Tabla 24**, **Tabla 25** y **Tabla 26**.

Tabla 24. Métrica F1_score de los 5 dataset para cada uno de los 3 algoritmos (en filas)

dataset_1	dataset_2	dataset_3	dataset_4	dataset_5
0.31300130	0.31250583	0.31200897	0.31250583	0.31388966
0.31622350	0.30731989	0.31483852	0.31450470	0.30776751
0.32024687	0.31149767	0.32155298	0.32249085	0.32682157

Tabla 25. Métrica de precisión de los 5 dataset para cada uno de los 3 algoritmos (en filas)

dataset_1	dataset_2	dataset_3	dataset_4	dataset_5
0.29500703	0.29412798	0.29324894	0.29412798	0.29658931
0.33596468	0.33395273	0.33600233	0.33435809	0.33340115
0.33186060	0.30271804	0.33978382	0.34015594	0.35201097

Tabla 26. Métrica de recall de los 5 dataset para cada uno de los 3 algoritmos (en filas)

dataset_1	dataset_2	dataset_3	dataset_4	dataset_5
0.33333333	0.33333333	0.33333333	0.33333333	0.33333333
0.34039897	0.33432371	0.33976857	0.33633989	0.33286549
0.33287626	0.32770948	0.33411055	0.33456469	0.33638237

En cuanto al análisis de resultados, se observa lo siguiente:

El Algoritmo_3 presenta un F1-score moderado en Dataset_5, a pesar de esto experimenta pérdidas financieras. Esto indica que, aunque tiene la capacidad de identificar oportunidades comerciales con cierta precisión, no logra mantener una rentabilidad constante.

- En el caso del Algoritmo_2, a pesar de que su F1-score es moderado en el Dataset_2, este algoritmo genera ganancias positivas. Esto sugiere que, aunque no sea el algoritmo más preciso, está tomando decisiones comerciales efectivas que se traducen en beneficios financieros. Sin embargo, es fundamental evaluar si estas ganancias son suficientes para compensar los costos de transacción y los riesgos asociados.
- El Algoritmo_1 se encuentra en una posición intermedia en términos de F1-score en relación con los otros dos algoritmos. Esto sugiere que Algoritmo_1 tiene un rendimiento promedio tanto en precisión como en términos de resultados financieros.
- El Algoritmo_3 demuestra la mayor precisión promedio en diversos conjuntos de datos, lo que sugiere que, en términos generales, es el más preciso de los tres algoritmos al predecir o analizar series de tiempo financieras en estos conjuntos de datos particulares.
- Se observa una notable consistencia en el desempeño del recall en todos los conjuntos de datos y para cada uno de los tres algoritmos. Esto sugiere que, en términos de recall, los algoritmos exhiben una uniformidad en su efectividad independientemente del conjunto de datos específico. Dado que no hay diferencias discernibles en el recall entre los algoritmos, al tomar decisiones sobre qué algoritmo utilizar en aplicaciones financieras, se recomienda enfocarse en otras métricas como la precisión o consideraciones específicas del dominio financiero.

4.1.2 Test de Friedman

Con base en los resultados de la prueba de Friedman como se detalla en la **Tabla 27** con un nivel de significancia alfa de 0.05, se pueden obtener las siguientes conclusiones:

- **Precision y Profit:** Los resultados indican que existen diferencias estadísticamente significativas entre las muestras en lo que respecta a las métricas de precisión y profit. El valor p (p-value) para ambas métricas es menor que el nivel de significancia alfa, lo que sugiere que al menos dos de las muestras tienen diferencias significativas en estas métricas.
- **F1 Score y Recall:** No se encontraron diferencias estadísticamente significativas en las métricas de F1 Score y Recall entre las muestras. Los valores p para estas métricas son mayores que alfa, lo que indica que las diferencias observadas podrían deberse al azar y no son estadísticamente significativas.

Tabla 27. Resultados obtenidos de la prueba de Friedman para cada una de las métricas previamente calculadas

Friedman - alfa=0.05	f1_score	precision	profit	recall
statistic	4.799999999999999	7.600000000	7.600000000	2.800000000
p-value	0.090717953289	0.02237077	0.02237077	0.24659696

En resumen, los resultados del test de Friedman indican que las métricas de precisión y profit son las que muestran las diferencias más significativas entre las muestras, mientras que las métricas de F1 Score y Recall no muestran diferencias significativas. No se puede concluir de manera definitiva que todas las métricas son diferentes en conjunto entre las muestras, ya que no hay suficiente evidencia estadística para respaldar esta afirmación.

Basándose en los resultados del Ranking de Friedman de la **Tabla 28**, se puede decir que el algoritmo 3 se posiciona como el mejor en su clasificación en la métrica "f1_score," lo que sugiere un rendimiento superior en comparación con los otros dos algoritmos en esta métrica. Sin embargo, es importante señalar que las diferencias entre las clasificaciones de los algoritmos son relativamente pequeñas. Además que el algoritmo 2 se destaca como el mejor en las métricas de "profit" y "recall," ya que obtiene las clasificaciones más altas en estas métricas. Esto indica que el algoritmo 2 es preferible cuando el "recall" y la maximización del beneficio son consideraciones clave para una estrategia de training.

Tabla 28. Resultados obtenidos de Ranking Friedman para cada una de las métricas previamente calculadas

Ranking - Friedman	algorithm 1	algorithm 2	algorithm 3
f1_score	1.6	1.6	2.8
precision	1.0	2.4	2.6
profit	2.4	2.6	1.0
recall	1.6	2.6	1.8

4.1.3 Pruebas Post Hoc

4.1.3.1 Pruebas de Holm

Los resultados presentados corresponden a una prueba de comparación múltiple aplicando el procedimiento de corrección de Holm para analizar la métrica "profit" en los tres algoritmos: Algorithm_1, Algorithm_2 y Algorithm_3. A continuación, se proporciona la interpretación de los resultados extraídos de la **Tabla 29**.

Tabla 29. Resultados obtenidos de la prueba de Holm para la métrica "profit_loss"

Holm - profit	Statistic	Adjusted p	Result
algorithm 2 vs. algorithm 3	-1.0	0.02499999	H0 es aceptado
algorithm 1 vs. algorithm 3	7.0	0.04999999	H0 es rechazado
algorithm 1 vs. algorithm 2	8.0	500000000	H0 es aceptado

- **Algorithm_2 vs. Algorithm_3:**

Estadístico: -1.0

Valor p ajustado: 0.025 (aproximadamente)

Resultado: H0 (hipótesis nula) es aceptada

En esta comparación entre Algorithm_2 y Algorithm_3, el valor del estadístico es negativo, indicando que Algorithm_3 tiene un mejor desempeño en "profit" en comparación con Algorithm_2. El valor p ajustado es menor que el nivel de significancia (alfa), lo que lleva a la aceptación de la hipótesis nula (H0) de que no hay una diferencia significativa en "profit_loss" entre Algorithm_2 y Algorithm_3.

- **Algorithm_1 vs. Algorithm_3:**

Estadístico: 7.0

Valor p ajustado: 0.05 (aproximadamente)

Resultado: H0 es rechazado

En esta comparación entre Algorithm_1 y Algorithm_3, el valor del estadístico es positivo, indicando que Algorithm_1 tiene un mejor desempeño en "profit" en comparación con Algorithm_3. El valor p ajustado es menor que el nivel de significancia (alfa), lo que conduce al rechazo de la hipótesis nula (H0) de que no hay una diferencia significativa en "profit_loss" por lo que el Algorithm_1 obtiene mejores resultados que el Algorithm_3.

- **Algorithm_1 vs. Algorithm_2:**

Estadístico: 8.0

Valor p ajustado: 50,000,000,000.0 (un valor extremadamente alto)

Resultado: H0 es aceptado

En esta comparación entre Algorithm_1 y Algorithm_2, el valor del estadístico es positivo, indicando que Algorithm_1 tiene un mejor desempeño en "profit_loss" en comparación con Algorithm_2. Sin embargo, el valor p ajustado es extremadamente alto, lo que sugiere que no hay evidencia suficiente para rechazar la hipótesis nula (H0) de que no hay una diferencia significativa en "profit" entre Algorithm_1 y Algorithm_2.

En función de estos resultados, se puede concluir que, desde una perspectiva estadística, el Algorithm_1 tiene un mejor rendimiento en "profit_loss" en comparación con Algorithm_3 y Algorithm_2 en las comparaciones individuales.

Continuando con los resultados se presenta el procedimiento de corrección de Holm para analizar la métrica "f1_score" en los tres algoritmos, Algorithm_1, Algorithm_2 y Algorithm_3. A continuación, se proporciona la interpretación de los resultados extraídos de la **Tabla 30**.

Tabla 30. Resultados obtenidos de la prueba de Holm para la métrica "f1_score"

Holm - f1_score	Statistic	Adjusted p value	Result
algorithm 2 vs. algorithm 3	-5.4	0.0249999999999875	H0 es rechazado
algorithm 1 vs. algorithm 3	-4.800000000000001	0.049999999999995	H0 es rechazado
algorithm 1 vs. algorithm 2	-0.5999999999999996	5000000000.0	H0 es aceptado

Los resultados presentados indican que hubo diferencias significativas en el rendimiento de los algoritmos 2 y 1 en comparación con el algoritmo 3 en términos de f1_score. Los valores de "Adjusted p value" para los primeros dos casos fueron notablemente bajos (0.0249999999999875 y 0.049999999999995), lo que sugiere evidencia estadística para rechazar la hipótesis nula (H0) en esos escenarios. El tercer caso presenta un valor inusualmente alto de "Adjusted p value" (5000000000.0), lo que sugiere que no hubo evidencia estadística suficiente para rechazar la hipótesis nula (H0) en este contexto.

- **Algorithm_2 vs. Algorithm_3":**

Statistic: -5.4

Adjusted p value: 0.0249999999999875

Este valor negativo indica que el algoritmo 2 tiene un desempeño significativamente menor en términos de f1_score en comparación con el algoritmo 3. Cuanto más negativo sea el Statistic, mayor será la diferencia en el rendimiento. En esta comparación, el valor "Adjusted p value" es 0.0249999999999875, lo que indica que hay evidencia estadística para rechazar

la hipótesis nula. Esto sugiere que hay una diferencia significativa entre "Algorithm_2" y "Algorithm_3" en la métrica f1_score.

- **Algorithm_1 vs. Algorithm_3:**

Statistic: -4.8000000000000001

Adjusted p value: 0.049999999999995

El Statistic negativo sugiere que el algoritmo 1 tiene un desempeño significativamente inferior en términos de f1_score en comparación con el algoritmo 3. Al igual que en el primer caso, un valor negativo grande indica una diferencia sustancial. En esta comparación, el valor "Adjusted p value" es 0.049999999999995. Esto también sugiere que hay evidencia estadística para rechazar la hipótesis nula en términos de la métrica F1-score, lo que implica que hay una diferencia significativa en el F1-score entre "Algorithm_1" y "Algorithm_3".

- **Algorithm_1 vs. Algorithm_2:**

Statistic: -0.5999999999999996

Adjusted p value: 5000000000.0

En este caso, el Statistic está cerca de cero, lo que indica que no hay una diferencia sustancial en términos de f1_score entre el algoritmo 1 y el algoritmo 2. Un Statistic cercano a cero sugiere que los algoritmos tienen un rendimiento similar. El Adjusted p value presenta un valor inusualmente alto, lo que sugiere que no hubo evidencia estadística suficiente para rechazar la hipótesis nula (H0).

4.1.3.2 Prueba de Wilcoxon

El estadístico de Wilcoxon también fue usado para evaluar y comparar los tres algoritmos en términos de sus resultados de profit_loss. El estadístico de Wilcoxon es una prueba no paramétrica utilizada para comparar dos muestras relacionadas o emparejadas. En el caso en cuestión, se están comparando cada par de algoritmos entre sí según las métricas obtenidas de la **Tabla 31** y **Ahora se** presentan los resultados obtenidos para la métrica F1 basados en los datos de la **Tabla 32**.

Tabla 32.

Tabla 31. Resultados obtenidos de la prueba de Wilcoxon para la métrica "profit"

Wilcoxon - profit	R+	R-	Exact P-value
algorithm 1 vs. algorithm 2	6.0	9.0	0.8125
algorithm 1 vs. algorithm 3	0.0	15.0	0.0625
algorithm 2 vs. algorithm 3	0.0	15.0	0.0625

- **Algorithm_1 vs. Algorithm_2:**

R+ (suma de rangos positivos): 6.0

R- (suma de rangos negativos): 9.0

Valor p exacto: 0.8125

El valor p alto (0.8125) indica que no hay evidencia significativa para rechazar la hipótesis nula de que no hay diferencia significativa en los datos financieros de profit_loss entre el Algorithm_1 y el Algorithm_2. Esto sugiere que estos dos algoritmos pueden ser considerados estadísticamente similares en términos de sus resultados financieros.

- **Algorithm_1 vs. Algorithm_3:**

R+ (suma de rangos positivos): 0.0

R- (suma de rangos negativos): 15.0

Valor p exacto: 0.0625

En este caso, el valor p (0.0625) es relativamente bajo pero no inferior a alfa (alfa=0.05), lo que significa que no hay evidencia suficiente para rechazar la hipótesis nula de que no hay diferencia significativa de profit_loss entre el Algorithm_1 y el Algorithm_3.

- **Algorithm_2 vs. Algorithm_3:**

R+ (suma de rangos positivos): 0.0

R- (suma de rangos negativos): 15.0

Valor p exacto: 0.0625

La prueba de algoritmo 2 vs. algoritmo 3 produce resultados similares al caso anterior, con un valor p de 0.0625 por lo que no hay diferencia significativa de profit_loss entre el Algorithm_2 y el Algorithm_3.

Ahora se presentan los resultados obtenidos para la métrica F1 basados en los datos de la **Tabla 32**.

Tabla 32. Resultados obtenidos de la prueba de Wilcoxon para la métrica "f1_score"

Wilcoxon - f1_score	R+	R-	Exact P-value
algorithm 1 vs. algorithm 2	6.0	9.0	0.8125
algorithm 1 vs. algorithm 3	1.0	14.0	0.125
algorithm 2 vs. algorithm 3	0.0	15.0	0.0625

- **Algorithm_1 vs. Algorithm_2:**

R+ (suma de rangos positivos): 6.0

R- (suma de rangos negativos): 9.0

Valor p exacto: 0.8125

El valor p alto (0.8125) indica que no hay evidencia significativa para rechazar la hipótesis nula de que no hay diferencia significativa en el F1-score entre el algoritmo 1 y el algoritmo 2. Esto sugiere que estos dos algoritmos pueden considerarse estadísticamente similares en términos de su capacidad de clasificación.

- **Algorithm_1 vs. Algorithm_3:**

R+ (suma de rangos positivos): 1.0

R- (suma de rangos negativos): 14.0

Valor p exacto: 0.125

En este caso, el valor p (0.125) es relativamente bajo pero no inferior a alfa (alfa=0.05), lo que significa que no hay evidencia suficiente para rechazar la hipótesis nula de que no hay diferencia significativa de F1_score entre el Algorithm_1 y el Algorithm_3.

- **Algorithm_2 vs. Algorithm_3:**

R+ (suma de rangos positivos): 0.0

R- (suma de rangos negativos): 15.0

Valor p exacto: 0.0625

La prueba de algoritmo 2 vs. algoritmo 3 produce un valor p relativamente bajo (0.0625) pero no inferior a alfa (alfa=0.05), lo que indica que no hay evidencia para rechazar la hipótesis nula y concluir que no existe una diferencia en el F1-score entre el Algorithm_2 y Algorithm_3.

En resumen, en todas las comparaciones tanto para Profit como para F1-Score, los valores p no son lo suficientemente bajos como para rechazar la hipótesis nula de que no hay diferencias significativas en los valores promedios reportados por los algoritmos en estas métricas. Se precisa la definición de un mayor número de data sets para soportar el proceso de evaluación y comparación de los algoritmos.

CAPÍTULO 5

5 Conclusiones y trabajo futuro

En el transcurso de esta investigación sobre el trading algorítmico, se emprendió una revisión sistemática inicial con el objetivo de identificar no solo los modelos exitosos de pronóstico en series de tiempo financieras, sino también las tendencias emergentes en este campo. A través de este proceso, se logró no solo reconocer las bases de datos más ampliamente utilizadas, sino también los variados enfoques y metodologías que están siendo explorados y adoptados con entusiasmo por parte de los investigadores en las áreas financieras y computacionales. Estos hallazgos proporcionan una base sólida para el desarrollo futuro de estrategias y herramientas más avanzadas en el ámbito del trading algorítmico, que prometen un impacto significativo en el mundo de las inversiones y la toma de decisiones en los mercados financieros.

En la segunda fase de la investigación, se evaluó la viabilidad de desarrollar un framework destinado a la evaluación y comparación de algoritmos (y sus modelos), un recurso de gran interés debido a la inexistencia previa de esta herramienta en el ámbito del trading algorítmico. Este framework incluye tres modelos fundamentales, el primero basado en redes neuronales convolucionales profundas; el segundo basado en aprendizaje por refuerzo profundo; y el tercero basado en una red LSTM.

El framework proporciona la flexibilidad de evaluar cualquiera de los algoritmos mencionados y otros que en el futuro se adicionen, permitiendo a los usuarios seleccionar una serie de parámetros cruciales. Estos parámetros incluyen los intervalos de fechas para entrenamiento y validación, el número de evaluaciones para verificar la robustez del algoritmo/modelo, el capital a invertir y otros como la elección del número de épocas para el algoritmo 1 y el número de episodios para el algoritmo 2.

Tras completar la evaluación, la herramienta proporciona a los usuarios la capacidad de visualizar la evolución de la inversión a lo largo del tiempo para cada activo financiero, empleando gráficos específicos para los resultados obtenidos por cada algoritmo. Además, ofrece un análisis detallado a través de matrices de confusión que evalúan el desempeño de cada algoritmo para cada activo financiero y dataset, acompañadas de métricas clave como el puntaje F1, precisión y recall.

Finalmente, la herramienta incluye la realización de una prueba de Friedman, seguida por los análisis post hoc de Holm y Wilcoxon. Estos análisis permiten a los usuarios comprender la significancia estadística de las diferencias entre los algoritmos evaluados, proporcionando una visión completa de la efectividad de cada enfoque en el contexto de la inversión financiera. Con esta combinación de herramientas de visualización y análisis estadístico, los investigadores y profesionales pueden tomar decisiones más fundamentadas y precisas en el ámbito del trading algorítmico.

De acuerdo con la hipótesis de investigación H1, se sostiene que el algoritmo de vanguardia fundamentado en Q-Learning (aprendizaje por refuerzo) exhibe una superioridad estadística en términos de rentabilidad en comparación con los otros dos algoritmos seleccionados del estado del arte, con un nivel de significancia del 95%. Este resultado se basa en la evaluación de métricas, donde dicho algoritmo obtiene la calificación más alta, seguido por el algoritmo 1, caracterizado por su enfoque conservador. En última instancia, el algoritmo 3 ocupa la tercera posición, destacándose por registrar pérdidas, un aspecto de particular interés para los inversores.

Como perspectiva para futuras investigaciones, se plantea la tarea de incorporar nuevos algoritmos/modelos, junto con la introducción de datos en tiempo real. Esta mejora permitiría un análisis más sólido y dinámico, con el propósito de que la herramienta pueda brindar un apoyo más efectivo en la toma de decisiones relacionadas con los activos de interés, tanto para investigadores como para inversionistas.

CAPÍTULO 6

6 Bibliografía

- [1] Víctor Manuel Pérez Santos, "ROBOTRADER: SISTEMA ALGORÍTMICO DE TRADING EN LOS MERCADOS FINANCIEROS," Universidad de La Laguna, San Cristóbal de La Laguna, 2014. Accessed: Jan. 24, 2019. [Online]. Available: [https://riull.ull.es/xmlui/bitstream/handle/915/216/Robotrader%3Asistemas algoritmicos de trading en los mercados financieros..pdf?sequence=1&isAllowed=y](https://riull.ull.es/xmlui/bitstream/handle/915/216/Robotrader%3Asistemas%20algoritmicos%20de%20trading%20en%20los%20mercados%20financieros..pdf?sequence=1&isAllowed=y)
- [2] G. Vasquez and J. Muñoz, "SANTI: Robot de Trading con Pronósticos Basados en Inteligencia Artificial, una Evolución para el Mercado de Valores," 2015.
- [3] B. B. Nair and V. P. Mohandas, "Artificial intelligence applications in financial forecasting-a survey and some empirical results," *Intell. Decis. Technol.*, vol. 9, no. 2, pp. 99–140, 2015, doi: 10.3233/IDT-140211.
- [4] Q. Huang, T. Wang, D. Tao, and X. Li, "Biclustering learning of trading rules," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2287–2298, 2015, doi: 10.1109/TCYB.2014.2370063.
- [5] W. C. Chiang, D. Enke, T. Wu, and R. Wang, "An adaptive stock index trading decision support system," *Expert Syst. Appl.*, vol. 59, pp. 195–207, 2016, doi: 10.1016/j.eswa.2016.04.025.
- [6] A. Karathanasopoulos, "Modelling and trading the London, New York and Frankfurt stock exchanges with a new gene expression programming trader tool," *Intell. Syst. Accounting, Financ. Manag.*, vol. 24, no. 1, pp. 3–11, 2017, doi: 10.1002/isaf.1401.
- [7] R. K. Shen, C. Y. Yang, V. R. L. Shen, W. C. Li, and T. S. Chen, "A Stock Market Prediction System Based on High-Level Fuzzy Petri Nets," *Int. J. Uncertainty, Fuzziness Knowledge-Based Syst.*, vol. 26, no. 5, pp. 771–808, 2018, doi: 10.1142/S0218488518500356.
- [8] H. Tang, P. Dong, and Y. Shi, "A new approach of integrating piecewise linear representation and weighted support vector machine for forecasting stock turning points," *Appl. Soft Comput. J.*, vol. 78, pp. 685–696, 2019, doi: 10.1016/j.asoc.2019.02.039.
- [9] C. Chalvatzis and D. Hristu-Varsakelis, "High-performance stock index trading via neural networks and trees," *Appl. Soft Comput. J.*, vol. 96, p. 106567, 2020, doi: 10.1016/j.asoc.2020.106567.
- [10] J. B. Chakole, M. S. Kolhe, G. D. Mahapurush, A. Yadav, and M. P. Kurhekar, "A Q-learning agent for automated trading in equity stock markets," *Expert Syst. Appl.*, vol. 163, p. 113761, 2021, doi: 10.1016/j.eswa.2020.113761.
- [11] H. Kang, X. Zong, J. Wang, and H. Chen, "Binary gravity search algorithm and support vector machine for forecasting and trading stock indices," *Int. Rev. Econ. Financ.*, vol. 84, no. November 2022, pp. 507–526, 2023, doi: 10.1016/j.iref.2022.11.009.
- [12] Y. Wu, Z. Fu, X. Liu, and Y. Bing, "A hybrid stock market prediction model based on GNG and reinforcement learning," *Expert Syst. Appl.*, vol. 228, no. October 2022, p.

- 120474, 2023, doi: 10.1016/j.eswa.2023.120474.
- [13] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering - A systematic literature review," *Inf. Softw. Technol.*, vol. 51, no. 1, pp. 7–15, 2009, doi: 10.1016/j.infsof.2008.09.009.
- [14] B. Kitchenham, "Procedures for Performing Systematic Reviews," p. 28, 2004, Accessed: Jan. 22, 2019. [Online]. Available: <http://www.inf.ufsc.br/~aldo.vw/kitchenham.pdf>
- [15] Z. Munn, C. Stern, E. Aromataris, C. Lockwood, and Z. Jordan, "What kind of systematic review should i conduct? A proposed typology and guidance for systematic reviewers in the medical and health sciences," *BMC Medical Research Methodology*, vol. 18, no. 1. BioMed Central, p. 5, Jan. 2018. doi: 10.1186/s12874-017-0468-4.
- [16] R. Rosas-Romero, A. Díaz-Torres, and G. Etcheverry, "Forecasting of stock return prices with sparse representation of financial time series over redundant dictionaries," *Expert Syst. Appl.*, vol. 57, pp. 37–48, 2016, doi: 10.1016/j.eswa.2016.03.021.
- [17] M. W. Hsu, S. Lessmann, M. C. Sung, T. Ma, and J. E. V. Johnson, "Bridging the divide in financial market forecasting: machine learners vs. financial economists," *Expert Syst. Appl.*, vol. 61, pp. 215–234, 2016, doi: 10.1016/j.eswa.2016.05.033.
- [18] S. P. Das and S. Padhy, "Unsupervised extreme learning machine and support vector regression hybrid model for predicting energy commodity futures index," *Memetic Comput.*, vol. 9, no. 4, pp. 333–346, 2017, doi: 10.1007/s12293-016-0191-4.
- [19] H. Kim, "A Prediction Methodology for the change of the values of Financial Products," vol. 51, no. 3, pp. 197–210, 2017.
- [20] B. B. Nair, P. K. S. Kumar, N. R. Sakthivel, and U. Vipin, "Clustering stock price time series data to generate stock trading recommendations: An empirical study," *Expert Syst. Appl.*, vol. 70, pp. 20–36, 2017, doi: 10.1016/j.eswa.2016.11.002.
- [21] K. H. Hong, H. Jung, and M. Park, "Predicting European carbon emission price movements," *Carbon Manag.*, vol. 8, no. 1, pp. 33–44, 2017, doi: 10.1080/17583004.2016.1275813.
- [22] Z. Hu, Y. Bao, R. Chiong, and T. Xiong, "Profit guided or statistical error guided? a study of stock index forecasting using support vector regression," *J. Syst. Sci. Complex.*, vol. 30, no. 6, pp. 1425–1442, 2017, doi: 10.1007/s11424-017-5293-7.
- [23] M. Thakur and D. Kumar, "A hybrid financial trading support system using multi-category classifiers and random forest," *Appl. Soft Comput. J.*, vol. 67, pp. 337–349, 2018, doi: 10.1016/j.asoc.2018.03.006.
- [24] L. Fiévet and D. Sornette, "Decision trees unearth return sign predictability in the S&P 500," *Quant. Financ.*, vol. 18, no. 11, pp. 1797–1814, 2018, doi: 10.1080/14697688.2018.1441535.
- [25] Y. Chen and Y. Hao, "Integrating principle component analysis and weighted support vector machine for stock trading signals prediction," *Neurocomputing*, vol. 321, pp. 381–402, 2018, doi: 10.1016/j.neucom.2018.08.077.
- [26] M. Risse, "Combining wavelet decomposition with machine learning to forecast gold returns," *Int. J. Forecast.*, vol. 35, no. 2, pp. 601–615, 2019, doi: 10.1016/j.ijforecast.2018.11.008.
- [27] C. Lohrmann and P. Luukka, "Classification of intraday S&P500 returns with a Random Forest," *Int. J. Forecast.*, vol. 35, no. 1, pp. 390–407, 2019, doi: 10.1016/j.ijforecast.2018.08.004.
- [28] A. L. Suárez-Cetrulo, A. Cervantes, and D. Quintana, "Incremental market behavior classification in presence of recurring concepts," *Entropy*, vol. 21, no. 1, pp. 1–18, 2019, doi: 10.3390/e21010025.

- [29] V. V. V'yugin and V. G. Trunov, "Adaptive universal trading strategy," *J. Commun. Technol. Electron.*, vol. 60, no. 6, pp. 658–672, 2015, doi: 10.1134/s1064226915060169.
- [30] C. F. Huang, C. J. Hsu, C. C. Chen, B. R. Chang, and C. A. Li, "An intelligent model for pairs trading using genetic algorithms," *Comput. Intell. Neurosci.*, vol. 2015, 2015, doi: 10.1155/2015/939606.
- [31] A. Karathanasopoulos, K. A. Theofilatos, G. Sermpinis, C. Dunis, S. Mitra, and C. Stasinakis, "Stock market prediction using evolutionary support vector machines: an application to the ASE20 index," *Eur. J. Financ.*, vol. 22, no. 12, pp. 1145–1163, 2016, doi: 10.1080/1351847X.2015.1040167.
- [32] F. Ye, L. Zhang, D. Zhang, H. Fujita, and Z. Gong, "A novel forecasting method based on multi-order fuzzy time series and technical analysis," *Inf. Sci. (Ny)*, vol. 367–368, pp. 41–57, 2016, doi: 10.1016/j.ins.2016.05.038.
- [33] Y. Li, B. Zheng, T. T. Chen, and X. F. Jiang, "Fluctuation-driven price dynamics and investment strategies," *PLoS One*, vol. 12, no. 12, pp. 1–15, 2017, doi: 10.1371/journal.pone.0189274.
- [34] R. C. Brasileiro, V. L. F. Souza, and A. L. I. Oliveira, "Automatic trading method based on piecewise aggregate approximation and multi-swarm of improved self-adaptive particle swarm optimization with validation," *Decis. Support Syst.*, vol. 104, pp. 79–91, 2017, doi: 10.1016/j.dss.2017.10.005.
- [35] S. H. Kim, H. S. Lee, H. J. Ko, S. H. Jeong, H. W. Byun, and K. J. Oh, "Pattern matching trading system based on the dynamic time warping algorithm," *Sustain.*, vol. 10, no. 12, pp. 1–18, 2018, doi: 10.3390/su10124641.
- [36] C. Avdoulas, S. Bekiros, and S. Boubaker, "Evolutionary-based return forecasting with nonlinear STAR models: evidence from the Eurozone peripheral stock markets," *Ann. Oper. Res.*, vol. 262, no. 2, pp. 307–333, 2018, doi: 10.1007/s10479-015-2078-z.
- [37] A. Pimenta, C. A. L. Nametala, F. G. Guimarães, and E. G. Carrano, "An Automated Investing Method for Stock Market Based on Multiobjective Genetic Programming," *Comput. Econ.*, vol. 52, no. 1, pp. 125–144, 2018, doi: 10.1007/s10614-017-9665-9.
- [38] Q. Huang, Z. Kong, Y. Li, J. Yang, and X. Li, "Discovery of trading points based on Bayesian modeling of trading rules," *World Wide Web*, vol. 21, no. 6, pp. 1473–1490, 2018, doi: 10.1007/s11280-018-0534-9.
- [39] R. Drezewski, G. Dziuban, and K. Pajak, "The bio-inspired optimization of trading strategies and its impact on the efficient market hypothesis and sustainable development strategies," *Sustain.*, vol. 10, no. 5, 2018, doi: 10.3390/su10051460.
- [40] M. R. Alimoradi and A. Husseinzadeh Kashan, "A league championship algorithm equipped with network structure and backward Q-learning for extracting stock trading rules," *Appl. Soft Comput. J.*, vol. 68, pp. 478–493, Jul. 2018, doi: 10.1016/j.asoc.2018.03.051.
- [41] J. C. P. M'Ng and M. Mehralizadeh, "Forecasting East Asian Indices Futures via a Novel Hybrid of Wavelet-PCA Denoising and Artificial Neural Network Models," *PLoS One*, vol. 11, no. 6, pp. 1–29, 2016, doi: 10.1371/journal.pone.0156338.
- [42] S. Pyo, J. Lee, M. Cha, and H. Jang, "Predictability of machine learning techniques to forecast the trends of market index prices: Hypothesis testing for the Korean stock markets," *PLoS One*, vol. 12, no. 11, pp. 1–17, 2017, doi: 10.1371/journal.pone.0188107.
- [43] L. Troiano, E. M. Villa, and V. Loia, "Replicating a Trading Strategy by Means of LSTM for Financial Industry Applications," *IEEE Trans. Ind. Informatics*, vol. 14, no. 7, pp. 3226–3234, 2018, doi: 10.1109/TII.2018.2811377.
- [44] O. B. Sezer and A. M. Ozbayoglu, "Algorithmic financial trading with deep

- convolutional neural networks: Time series to image conversion approach,” *Appl. Soft Comput.*, vol. 70, pp. 525–538, Sep. 2018, doi: 10.1016/J.ASOC.2018.04.024.
- [45] W. Long, Z. Lu, and L. Cui, “Deep learning-based feature engineering for stock price movement prediction,” *Knowledge-Based Syst.*, vol. 164, pp. 163–173, 2019, doi: 10.1016/j.knosys.2018.10.034.
- [46] F. Rundo, F. Trenta, A. L. Di Stallo, and S. Battiato, “Advanced Markov-based machine learning framework for making adaptive trading system,” *Computation*, vol. 7, no. 1, 2019, doi: 10.3390/computation7010004.
- [47] X. Zhong and D. Enke, “Predicting the daily return direction of the stock market using hybrid machine learning algorithms,” *Financ. Innov.*, vol. 5, no. 1, 2019, doi: 10.1186/s40854-019-0138-0.
- [48] G. Jeong and H. Y. Kim, *Improving financial trading decisions using deep Q-learning: Predicting the number of Shares, action Strategies, and transfer learning*, vol. 117. Elsevier Ltd, 2019. doi: 10.1016/j.eswa.2018.09.036.
- [49] F. Zhou, H. min Zhou, Z. Yang, and L. Yang, “EMD2FNN: A strategy combining empirical mode decomposition and factorization machine based neural network for stock market trend prediction,” *Expert Syst. Appl.*, vol. 115, pp. 136–151, 2019, doi: 10.1016/j.eswa.2018.07.065.
- [50] K. Y. Shen and G. H. Tzeng, “Fuzzy Inference-Enhanced VC-DRSA Model for Technical Analysis: Investment Decision Aid,” *Int. J. Fuzzy Syst.*, vol. 17, no. 3, pp. 375–389, 2015, doi: 10.1007/s40815-015-0058-8.
- [51] L. Dymova, P. Sevastjanov, and K. Kaczmarek, “A Forex trading expert system based on a new approach to the rule-base evidential reasoning,” *Expert Syst. Appl.*, vol. 51, pp. 1–13, 2016, doi: 10.1016/j.eswa.2015.12.028.
- [52] P. C. Chang, J. L. Wu, and J. J. Lin, “A Takagi-Sugeno fuzzy model combined with a support vector regression for stock trading forecasting,” *Appl. Soft Comput. J.*, vol. 38, pp. 831–842, 2016, doi: 10.1016/j.asoc.2015.10.030.
- [53] Jesús-Rodrigo Navia-Rodríguez; Carlos-Alberto Cobos-Lozada; Martha-Eliana Mendoza-Becerra, “Trading Algorítmico para la Predicción de Series de Tiempo Financieras : Una Revisión Sistemática,” vol. 58, pp. 337–358, 2020.
- [54] T. Théate and D. Ernst, “An application of deep reinforcement learning to algorithmic trading,” *Expert Syst. Appl.*, vol. 173, no. January, 2021, doi: 10.1016/j.eswa.2021.114632.
- [55] H. Van Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double Q-Learning,” *30th AAAI Conf. Artif. Intell. AAAI 2016*, pp. 2094–2100, 2016, doi: 10.1609/aaai.v30i1.10295.
- [56] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.
- [57] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *32nd Int. Conf. Mach. Learn. ICML 2015*, vol. 1, pp. 448–456, 2015.
- [58] F. Feng, X. He, X. Wang, C. Luo, Y. Liu, and T. S. Chua, “Temporal relational ranking for stock prediction,” *ACM Trans. Inf. Syst.*, vol. 37, no. 2, 2019, doi: 10.1145/3309547.
- [59] L. Zhang, C. Aggarwal, and G. J. Qi, “Stock price prediction via discovering multi-frequency trading patterns,” *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. Part F1296, pp. 2141–2149, 2017, doi: 10.1145/3097983.3098117.
- [60] Y. R. Wei Bao, Jun Yue, “A deep learning framework for financial time series using stacked autoencoders and long- short term memory,” *PLoS One*, pp. 1–24, 2017, doi: 10.6084/m9.figshare.5028110.
- [61] J. Alberg and Z. C. Lipton, “Improving Factor-Based Quantitative Investing by

Forecasting Company Fundamentals,” no. Nips, 2017, [Online]. Available: <http://arxiv.org/abs/1711.04837>