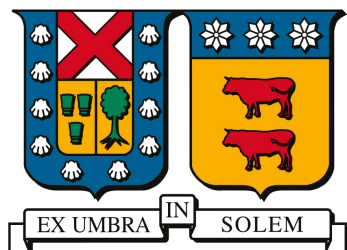


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INDUSTRIAS
VALPARAÍSO - CHILE



**ANÁLISIS DE EFICIENCIA DEL MERCADO DE
BITCOIN A TRAVÉS DE LA PREDICTIBILIDAD DE LOS
RETORNOS MEDIANTE MODELOS DE MACHINE
LEARNING**

FERNANDO ANDRÉS MAASS AGUILERA

**MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL INDUSTRIAL**

PROFESOR GUÍA : WERNER KRISTJANPOLLER RODRÍGUEZ
PROFESOR CORREFERENTE : DIEGO CANESSA RICH

DICIEMBRE 2025



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

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

Título del trabajo: Análisis de Eficiencia del Mercado de Bitcoin a Través de la Predictibilidad de los Retornos Mediante Modelos de Machine Learning

Nombre del candidato(a): Fernando Andrés Maass Aguilera

Carrera / Grado: Ingeniería Civil Industrial

Campus: Casa Central

Departamento: Industrias

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

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

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

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

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

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

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

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

4.- FIRMAS

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

Fecha: 22-12-2025

Firma:

Estudiante o Candidato(a):

Fecha: 22-12-2025

Firma:

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

Índice general

1	Resumen ejecutivo	6
2	Introducción	8
3	Objetivos	10
3.1	Objetivo General	10
3.2	Objetivos Específicos	10
4	Marco Teórico	11
4.1	Criptomonedas y Mercado Digital	11
4.1.1	Origen y Evolución de las Criptomonedas	11
4.1.2	Tecnología Blockchain	13
4.1.3	Bitcoin y los Eventos de Halving	14
4.2	Series Temporales	16
4.2.1	Introducción a las Series Temporales.....	16
4.2.2	Propiedades Estadísticas de las Series Temporales Financieras	17
4.3	Modelos Econométricos Tradicionales de Predicción.....	19
4.3.1	Modelo Autorregresivo Integrado de Medias Móviles (ARIMA).....	19
4.3.2	Modelo GARCH.....	20
4.3.3	Limitaciones de los Modelos Econométricos Tradicionales	21
4.4	Modelos de Aprendizaje Profundo para Series Temporales.....	22
4.4.1	Redes Neuronales Recurrentes (RNN)	22
4.4.2	Long Short-Term Memory (LSTM)	24
4.4.3	Gated Recurrent Unit (GRU).....	25
4.4.4	Limitaciones de las Arquitecturas Recurrentes	27
4.4.5	Mecanismo de Atención.....	28
4.4.6	Arquitectura del Modelo Transformer	29
4.4.7	Adaptación del Modelo Transformer a Series Temporales	34
4.4.8	Modelo Informer	35
4.5	Eficiencia de los Mercados Financieros.....	40
4.5.1	Hipótesis de los Mercados Eficientes (HME)	40
4.5.2	Hipótesis de Mercados Adaptativos (HMA)	42
5	Metodología.....	43
5.1	Construcción del Conjunto de Datos	43
5.2	Descripción de las Variables.....	44
5.2.1	Variable Objetivo	45

5.2.2	Variables predictoras	45
5.3	Partición Temporal del Conjunto de Datos	52
5.4	Configuración Modelos	53
5.4.1	Horizontes de predicción	53
5.4.2	Hiperparámetros	54
5.4.3	Función de perdida	55
5.4.4	Métricas de evaluación	56
5.4.5	Benchmarking	57
5.5	Implementación técnica	58
5.5.1	Entorno y Herramientas computacionales	58
5.5.2	Hardware	59
6	Resultados.....	60
6.1	Resultados para horizonte de predicción de 1 día	60
6.2	Resultados para horizontes de predicción de 7 a 90 días	61
6.3	Análisis de resultados para horizontes de predicción de 1 día	62
6.3.1	Capacidad predictiva	62
6.3.2	Capacidad direccional	64
6.4	Análisis de resultados para horizontes de predicción de 7 a 90 días	65
6.4.1	Desempeño en MAE y MSE	65
6.4.2	Capacidad direccional	67
6.5	Síntesis comparativa global: ranking y trade-off para la discusión	68
6.5.1	Ranking por métrica y horizonte	68
6.6	Implicancias para eficiencia débil	69
7	Conclusiones	71
A	Anexos	73
A.1	Gráficos complementarios	73
	Referencias	79

Índice de tablas

4.1	Eventos de halving en Bitcoin	15
5.1	Resumen de las variables macroeconómicas y financieras incluidas en el estudio	51
5.2	Partición temporal del conjunto de datos	52
5.3	Hiperparámetros fijos utilizados en el modelo Informer	54
6.1	Resultados promedio para horizonte de predicción de 1 día.	60
6.2	Resultados promedio para horizontes de predicción de 7 a 90 días.	61
6.3	Modelo ganador por horizonte de predicción y métrica. El modelo ganador corresponde al mejor desempeño promedio según cada métrica, considerando la minimización de MAE y MSE y la maximización de MDA.	68

Índice de figuras

4.1	Arquitectura básica de una red neuronal recurrente (RNN)	23
4.2	Arquitectura interna de una celda LSTM	25
4.3	Arquitectura interna de una celda GRU	27
4.4	Arquitectura Transformer	29
4.5	Mecanismo de atención del modelo Transformer. (a) Producto Punto Escalado, (b) Atención Multi Cabeza.....	32
4.6	Estructura de un <i>stack</i> individual dentro del codificador	37
4.7	Arquitectura general del modelo Informer	39
5.1	Precio histórico de Bitcoin (BTCUSD) y sus ciclos de halving	44
5.2	Tramos de entrenamiento, validación y testeo	52
6.1	Retornos observados vs. pronosticados para el horizonte $h = 1$ día en distintos tramos de volatilidad. Comparación Informer-LSTM	62
6.2	Distribución empírica de los errores de predicción (residuales) para el horizonte $h = 1$. Comparación entre modelos Informer y LSTM	63
6.3	Evolución temporal del acierto direccional (MDA rodante) para $h = 1$ día. Se reporta MDA con ventana móvil y bandas de confianza por remuestreo. Comparación entre Informer y LSTM	64
6.4	Perfil de MAE por paso de predicción (<i>lead time k</i>) en pronóstico multi-step. Comparación entre modelos.	66
6.5	Curva MDA vs horizonte para todos los modelos, con barras de error (variabilidad entre las 3 iteraciones)	67
A.1	Volumen de negociación	73
A.2	Direcciones activas de Bitcoin.....	73
A.3	Evolución del hash rate de Bitcoin	74
A.4	Evolución del índice del dólar estadounidense	74
A.5	Evolución del índice de volatilidad implícita (VIX).....	74
A.6	Evolución de la tasa de interés de USA	75
A.7	Evolución de la volatilidad del mercado de bonos de USA	75
A.8	Evolución del rendimiento del bono del tesoro a 10 años de USA	75
A.9	Evolución de la inflación implícita a 10 años de USA	76
A.10	Evolución del precio del oro	76
A.11	Evolución de la oferta monetaria de USA.....	76
A.12	Retornos observados vs pronosticados para el horizonte $h=1$ día. Tramo de caída (proxy). <i>Nota. Elaboración propia.</i>	77

A.13 Perfil de MSE por paso de predicción (<i>lead time k</i>) en pronóstico multi-step. Modelo Informer. <i>Nota. Elaboración propia.</i>	77
A.14 Perfil de MSE por paso de predicción (<i>lead time k</i>) en pronóstico multi-step. Modelo LSTM. <i>Nota. Elaboración propia.</i>	77
A.15 Perfil de MSE por paso de predicción (<i>lead time k</i>) en pronóstico multi-step. Modelo GRU. <i>Nota. Elaboración propia.</i>	78

1. Resumen ejecutivo

Este trabajo analiza la eficiencia débil del mercado de Bitcoin. El sustento es que si es posible predecir el retorno logarítmico utilizando únicamente información pública pasada, entonces existe evidencia de predictibilidad y, en consecuencia, de ineficiencia débil, al menos bajo ciertas condiciones. La motivación se sustenta en la elevada volatilidad del mercado cripto y en la presencia de eventos estructurales como los halvings, que podrían alterar transitoriamente la dinámica de precios y la velocidad con que la información se incorpora al mercado.

Para abordar el objetivo, se construye un dataset diario para el período 01/12/2014–31/10/2025, integrando series provenientes de Bitcoin (precio, halvings y métricas de la red blockchain) y variables macroeconómicas/financieras. La variable objetivo corresponde al retorno logarítmico diario del precio de Bitcoin, mientras que el conjunto de predictores se organiza en cinco bloques: variables del mercado cripto, indicadores de análisis técnico, métricas de red blockchain, variables asociadas al halving y variables macro/financieras. La evaluación se realiza con una partición temporal estricta en los tramos (*train*, *validation* y *test*) para evitar sesgos de anticipación y asegurar comparabilidad entre modelos.

El modelo principal es el Informer, comparado contra LSTM, GRU y LSTNet como *benchmarks*. Se consideran seis horizontes de predicción ($h = 1, 7, 14, 30, 60, 90$ días), permitiendo contrastar desempeño desde el corto plazo hasta escalas donde podrían emerger patrones más persistentes. La evaluación se efectúa mediante MAE y MSE (precisión en magnitud) y MDA (precisión direccional), dado que la dirección del movimiento es particularmente relevante para discutir eficiencia débil.

Los resultados evidencian un trade-off consistente entre minimizar error y capturar dirección. En horizontes muy cortos, los modelos recurrentes tienden a mostrar ventajas en error promedio, mientras que Informer presenta un desempeño competitivo y, en horizontes medios, destaca por su comportamiento direccional promedio. No obstante, las mejoras direccionales observadas son

moderadas, por lo que la evidencia respecto a la eficiencia débil se interpreta con cautela, la predictibilidad existe, pero su magnitud sugiere que cualquier señal explotable sería limitada y requeriría pruebas adicionales de robustez y significancia para sostener conclusiones más fuertes.

2. Introducción

La crisis financiera de 2008 intensificó la desconfianza hacia el sistema financiero tradicional y revitalizó el interés por alternativas que redujeran la dependencia de intermediarios y autoridades centrales. En este contexto surge Bitcoin, dando origen a un ecosistema de activos digitales basado en criptografía y redes distribuidas. Desde entonces, el mercado de Bitcoin ha exhibido ciclos pronunciados de expansión y contracción, junto con episodios de alta volatilidad, los cuales se asocian a factores de adopción, condiciones macroeconómicas, sentimiento de mercado e innovaciones del propio ecosistema. Un rasgo estructural distintivo es el *halving*, evento programado que reduce periódicamente la emisión del activo y que puede modificar la dinámica de oferta, influyendo sobre expectativas y comportamiento de participantes.

Estas características motivan una pregunta central en finanzas: ¿en qué medida el precio (o el retorno) de Bitcoin refleja de forma inmediata la información pública disponible? Bajo la hipótesis de eficiencia de mercado en su forma débil, la información contenida en los precios pasados ya estaría incorporada en el precio actual, por lo que no debería ser posible obtener ventajas sistemáticas utilizando únicamente datos históricos. Sin embargo, en mercados con elevada volatilidad, cambios estructurales y participación heterogénea, es predecible que la eficiencia no sea constante, sino que varíe en el tiempo, mostrando períodos de mayor o menor predictibilidad.

En paralelo, el pronóstico de series temporales financieras ha evolucionado desde enfoques econométricos tradicionales hacia métodos de aprendizaje profundo, capaces de modelar relaciones no lineales y dependencias complejas. En este marco, el presente trabajo adopta una perspectiva aplicada: evaluar la eficiencia débil de Bitcoin a partir de la capacidad predictiva de modelos de deep learning entrenados con información pública histórica. El criterio es directo, menor predictibilidad sugiere mayor eficiencia, mientras que mayor precisión (especialmente direccional) sugiere la presencia de señal potencialmente explotable, compatible con episodios de ineficiencia débil.

Bajo esta perspectiva, el presente trabajo adopta un enfoque empírico para analizar la

eficiencia débil del mercado de Bitcoin a partir de su grado de predictibilidad. En particular, se evalúa la capacidad de distintos modelos de aprendizaje profundo para anticipar el retorno logarítmico diario de Bitcoin utilizando exclusivamente información pública histórica. El criterio subyacente es directo, una baja capacidad predictiva es consistente con un mercado cercano a la eficiencia débil, mientras que una mayor precisión, especialmente en términos direccionales, sugiere la presencia de señal potencialmente explotable y, por ende, episodios de ineficiencia débil.

Asimismo, el análisis se extiende a distintos horizontes de predicción, lo que permite examinar cómo varía la eficiencia del mercado en escalas temporales cortas y medias. Esta dimensión resulta particularmente relevante, ya que la eficiencia puede manifestarse de forma diferenciada según el horizonte considerado. Adicionalmente, se incorpora el estudio de los eventos de halving como posibles puntos de quiebre estructural, dado que representan cambios anticipados en la política monetaria del protocolo y constituyen un escenario propicio para evaluar cómo el mercado incorpora información conocida con antelación.

3. Objetivos

3.1. Objetivo General

- Analizar la eficiencia débil del mercado de Bitcoin mediante la aplicación de modelos de deep learning, con el propósito de evaluar empíricamente su grado de predictibilidad durante el período 2014-2025 y de comparar su evolución en distintos momentos del ciclo de halving.

3.2. Objetivos Específicos

- Construir y depurar un conjunto de datos consolidado para Bitcoin (2014-2025), incorporando variables de mercado, indicadores de análisis técnico, variables macroeconómicas y de actividad en la red blockchain, garantizando su alineación temporal para el entrenamiento de modelos predictivos.
- Implementar y entrenar modelos de deep learning aplicables a series de tiempo (LSTM, GRU, LSTNet e Informer), evaluando distintos horizontes de predicción.
- Evaluar el desempeño predictivo de cada modelo mediante métricas tradicionales (MAE, MSE) y métricas direccionales (MDA), comparando su comportamiento según el modelo y el horizonte de predicción.
- Determinar si los patrones predictivos aportan evidencia a favor o en contra de la hipótesis de eficiencia débil del mercado de Bitcoin, analizando además el efecto que los eventos de halving podrían tener sobre su predictibilidad.

4. Marco Teórico

4.1. Criptomonedas y Mercado Digital

4.1.1. Origen y Evolución de las Criptomonedas

El desarrollo de las criptomonedas es parte de la evolución del dinero digital, que puede ser entendido como cualquier medio de pago que no requiere soporte físico y que puede transferirse a través de redes electrónicas. Desde finales del siglo XX, diversas iniciativas exploraron la posibilidad de reemplazar o complementar el dinero tradicional mediante sistemas digitales más rápidos, eficientes y seguros (Narayanan et al., 2016).

Surgieron propuestas prometedoras como *DigiCash*, creado por David Chaum en 1989, que introdujo transacciones electrónicas basadas en criptografía y orientadas a la privacidad, sin embargo, nunca alcanzó la adopción masiva por parte de los usuarios (Chaum, 1989).

Posteriormente en 1996, apareció *E-Gold*, fundado por Douglas Jackson y Barry Downey, el cual permitía a los usuarios abrir cuentas en línea y realizar transferencias respaldadas por reservas físicas de oro almacenadas en bóvedas. Este sistema alcanzó varios millones de cuentas activas a comienzos de los años 2000 y se convirtió en una de las primeras plataformas de pago digital ampliamente utilizadas a escala global. Su atractivo radicaba en ofrecer una alternativa electrónica al dinero fiduciario con respaldo tangible, lo que generaba confianza entre los usuarios. Sin embargo, al ser un sistema centralizado, dependía de la administración de la empresa que lo controlaba y, con el tiempo, su popularidad atrajo también usos ilícitos. Estas limitaciones y la presión regulatoria llevaron finalmente a su clausura en 2009, dejando en evidencia tanto el potencial como las limitaciones de los modelos de dinero electrónico centralizados (Nian & Chuen, 2015).

En un contexto de creciente digitalización de la economía y de desconfianza hacia los

intermediarios financieros tradicionales, especialmente tras la crisis subprime de 2008 (Sapienza & Zingales, 2012), apareció una propuesta que cambiaría de manera definitiva el panorama del dinero digital. En octubre de ese mismo año, un autor bajo el seudónimo de Satoshi Nakamoto publicó el documento *Bitcoin: A Peer-to-Peer Electronic Cash System*, donde planteó por primera vez un sistema monetario completamente descentralizado, capaz de resolver el problema del doble gasto sin necesidad de una autoridad central (Nakamoto, 2008).

La innovación de Nakamoto consistió en combinar técnicas criptográficas con un mecanismo de consenso de prueba de trabajo (*Proof of Work*) y el uso de un registro distribuido e inmutable denominado blockchain, que se convirtió en el soporte tecnológico esencial de las criptomonedas. En enero de 2009 se minó el primer bloque de Bitcoin, denominado Genesis, marcando el inicio del ecosistema criptográfico moderno y estableciendo las bases para el desarrollo de nuevas criptomonedas y aplicaciones descentralizadas.

A partir de allí, se diversificó el ecosistema con nuevas propuestas. Entre ellas, Litecoin (2011) que introdujo un protocolo de validación más rápido y un algoritmo alternativo de minería, consolidándose como la primera altcoin relevante y un espacio de experimentación para mejoras técnicas. Poco después, Ripple (2012) se enfocó en optimizar los pagos internacionales y las transferencias entre instituciones financieras, buscando conectar la tecnología blockchain con el sistema bancario tradicional.

Un avance significativo representó la creación de Ethereum (2015), al ampliar las aplicaciones de la tecnología blockchain mediante los contratos inteligentes, los cuales permitieron la creación de programas autoejecutables y sentaron las bases de los actuales sistemas de finanzas descentralizadas (DeFi) (Buterin, 2015). Posteriormente, surgieron otras criptomonedas de relevancia como Cardano (2017), destacada por su desarrollo basado en investigación académica y Binance Coin (2017), vinculada al mayor exchange de criptomonedas del mundo y convertida luego en la base de un ecosistema propio. Más recientemente, Polkadot (2020) introdujo la interoperabilidad entre múltiples cadenas de bloques, mientras que Solana (2020) se consolidó por su alta velocidad y bajo costo en

transacciones, lo que la hizo atractiva para aplicaciones DeFi y el mercado de tokens no fungibles (NFTs).

4.1.2. Tecnología Blockchain

La tecnología *blockchain* o cadena de bloques constituye la base tecnológica sobre la cual operan las criptomonedas y otros sistemas basados en registros distribuidos.

Un registro distribuido (*distributed ledger*) puede entenderse como una base de datos compartida entre múltiples participantes, donde cada nodo de la red mantiene y actualiza una copia sincronizada del registro, sin depender de un servidor central. Esta arquitectura permite que la información se almacene y valide de forma colectiva, garantizando transparencia, trazabilidad y resistencia a la manipulación. En este contexto, blockchain representa una aplicación específica de los registros distribuidos, cuya principal característica es la organización de la información en bloques encadenados de manera secuencial y segura.

Cada conjunto de transacciones se agrupa en un bloque que contiene, además de los datos de las operaciones, un identificador único (*hash*), una marca de tiempo (*timestamp*) y la referencia al bloque anterior, formando una secuencia continua e inalterable de registros. Esta propiedad de inmutabilidad es una de las principales razones por las que la blockchain se considera una de las tecnologías más seguras para el almacenamiento de datos y la transferencia de valor (Crosby et al., 2016).

El proceso de validación de las transacciones dentro de la red se realiza mediante mecanismos de consenso, siendo el más conocido la Prueba de Trabajo (*Proof of Work, PoW*), introducida por Nakamoto en el protocolo de Bitcoin. En este mecanismo, los nodos denominados “mineros” compiten para resolver un problema criptográfico complejo que requiere un alto poder computacional. El primero en resolverlo valida el bloque, lo añade a la cadena y recibe una recompensa en forma de nuevas monedas. Este procedimiento, además de garantizar la seguridad de la red, regula la emisión

monetaria de manera descentralizada, reemplazando el rol de los bancos centrales en la creación y control del dinero.

Sin embargo, el alto consumo energético ha impulsado el desarrollo de nuevos mecanismos de consenso más sostenibles. Entre ellos destacan la Prueba de Participación (*Proof of Stake, PoS*) y sus variantes, que sustituyen la competencia computacional por un proceso de selección basado en la cantidad de criptomonedas que posee cada validador. Estos sistemas reducen significativamente el gasto energético, mejoran la eficiencia de la red y facilitan la escalabilidad, siendo adoptados por criptomonedas como Ethereum tras su transición a Ethereum 2.0 en 2022.

Además del ámbito monetario, la tecnología blockchain ha permitido la expansión de nuevos sectores y modelos económicos. Entre los más destacados se encuentran las Finanzas Descentralizadas (*Decentralized Finance, DeFi*), que replican servicios financieros tradicionales —como préstamos, intercambios o seguros— sin intermediarios, y los Tokens No Fungibles (*Non-Fungible Tokens, NFTs*), que posibilitan la certificación y propiedad digital de activos únicos.

Estas aplicaciones demuestran que la blockchain trasciende el uso de las criptomonedas, configurándose como una infraestructura tecnológica versátil para múltiples industrias. En síntesis, la blockchain combina tres propiedades esenciales: descentralización, transparencia e inmutabilidad.

4.1.3. Bitcoin y los Eventos de Halving

Bitcoin constituye el activo digital más representativo del ecosistema cripto y el principal referente para el análisis de los mercados descentralizados. Su relevancia deriva no solo en haber sido la primera criptomoneda en operar de forma autónoma, sino que también por la forma de su estructura económica, que combina una política monetaria programada, donde la oferta es limitada y la emisión es un proceso decreciente a lo largo del tiempo.

En el código fuente de Bitcoin se establece que la cantidad máxima de unidades en circulación no puede superar los 21 millones de bitcoins (BTC).

La emisión de nuevos bitcoins se realiza mediante la recompensa que reciben los mineros al validar bloques de transacciones, y dicha recompensa se reduce a la mitad en un evento denominado *halving*, que ocurre cada 210.000 bloques minados, lo que equivale aproximadamente a un período de cuatro años. Este mecanismo introduce una tasa de inflación, entendida como el crecimiento de la oferta en BTC, decreciente y completamente predecible, generando una dinámica de escasez que influye tanto en la oferta como en las expectativas del mercado.

Hasta la fecha, se han registrado cuatro eventos de halving en la red de Bitcoin, los cuales reducen de manera programada la recompensa otorgada a los mineros por cada bloque validado. La siguiente tabla (4.1) resume cada evento y su correspondiente ajuste en la emisión monetaria:

Tabla 4.1: Eventos de halving en Bitcoin

Nº de Halving	Fecha	Recompensa Anterior (BTC)	Recompensa Posterior (BTC)
1º	28/11/2012	50	25
2º	09/07/2016	25	12.5
3º	11/05/2020	12.5	6.25
4º	19/04/2024	6.25	3.125

Fuente: Elaboración propia.

Inicialmente, la recompensa por bloque minado era de 50 BTC, esta ha ido disminuyendo en cada halving a la mitad. Actualmente, tras el cuarto evento de halving ocurrido en abril de 2024, los mineros reciben una recompensa de 3,125 BTC por bloque validado, además de las comisiones asociadas a las transacciones de dicho bloque.

Cada reducción en las recompensas implica un ajuste estructural en el sistema, ya que disminuye los ingresos de los mineros y afecta la relación entre los costos de minería, la oferta monetaria y el precio de mercado.

En términos empíricos, los halvings han coincidido históricamente con fases de apreciación

del precio seguidas de períodos de corrección, configurando ciclos de expansión y contracción que reflejan la reacción del mercado ante cambios en la oferta.

Bajo esta perspectiva, los eventos de halving representan un experimento natural sobre los efectos de la escasez programada en Bitcoin. La reducción sistemática de la emisión de nuevos Bitcoins, combinada con la transparencia y predictibilidad del protocolo, proporciona un entorno ideal para evaluar cómo la información anticipada afecta las decisiones de inversión y la predictibilidad de los precios. Este aspecto resulta especialmente relevante para el análisis de la eficiencia del mercado cripto, ya que permite examinar si los cambios en la estructura de oferta son incorporados de forma inmediata y racional por los participantes.

En síntesis, la política monetaria predefinida de Bitcoin y la periodicidad de los eventos de halving constituyen los pilares de su dinámica económica. Ambos elementos determinan su comportamiento cíclico, su relación con la volatilidad y su papel como activo de referencia en el estudio de la eficiencia de los mercados digitales.

4.2. Series Temporales

4.2.1. Introducción a las Series Temporales

El análisis de las series temporales constituye uno de los pilares de la econometría moderna, ya que permite modelar procesos dinámicos cuyos valores presentan dependencia secuencial. Una serie temporal se puede definir como el conjunto ordenado de observaciones $\{y_t\}_{t=1}^T$, donde cada valor y_t corresponde a una variable aleatoria registrada en el tiempo t (Box et al., 2016).

Una serie temporal puede entenderse como la realización de un proceso estocástico subyacente, cuyo objetivo es capturar la relación entre observaciones pasadas y futuras.

En el ámbito financiero, las series temporales se utilizan para modelar y analizar distintos tipos de variables como los precios de los activos, tipos de cambio, tasas de interés, rendimiento de

los bonos, y más recientemente, el valor de las criptomonedas.

4.2.2. Propiedades Estadísticas de las Series Temporales Financieras

Las series temporales financieras se caracterizan por poseer propiedades estadísticas distintas de otros conjuntos de datos, por lo que es fundamental tenerlas en cuenta al momento de analizarlas o aplicar algún modelo. Entre las propiedades más importantes se encuentran:

- **Estacionariedad:** Una serie financiera es estacionaria cuando sus propiedades estadísticas (media, varianza y covarianza) permanecen constantes, independiente del momento en que se observan los datos. Para que una serie $\{y_t\}$ sea estacionaria, debe cumplir tres condiciones fundamentales (Box et al., 2016):

$$\mathbb{E}[y_t] = \mu, \quad \forall t, \quad (1)$$

$$\text{Var}[y_t] = \sigma^2, \quad \forall t, \quad (2)$$

$$\text{Cov}[y_t, y_{t+k}] = \gamma_k, \quad \forall t, k, \quad (3)$$

La estacionariedad en las series financieras depende de si se analizan los precios o los rendimientos. En las series de precios de la mayoría de los activos financieros, éstas son no estacionarias, debido a que los valores de la serie están correlacionados fuertemente con el valor inmediatamente anterior. Por el contrario, las series de rendimientos (o retornos), son en general estacionarias débiles, condición fundamental para utilizar la mayoría de los modelos econométricos aplicados a series de tiempo. El retorno logarítmico (r_t) entre el período $t - 1$ y t se calcula mediante la siguiente expresión:

$$r_t = \ln \left(\frac{P_t}{P_{t-1}} \right), \quad (4)$$

Donde P_t es el precio del activo en el momento t . Este tipo de retorno es a menudo el preferido en análisis de series de tiempo financieras por sus propiedades que facilitan la modelación.

- **Dependencia temporal:** La relación estadística entre los valores pasados y los valores futuros de una serie temporal financiera permite identificar dependencias y patrones. Esta puede ser calculada mediante la función de autocorrelación (ACF). La función autocorrelación al rezago k se define como:

$$\rho_k = \frac{\mathbb{E}[(Y_t - \mu)(Y_{t+k} - \mu)]}{\sigma_Y^2} = \frac{\text{Cov}(Y_t, Y_{t+k})}{\text{Var}(Y_t)}, \quad (5)$$

Donde ρ_k es el coeficiente de autocorrelación al rezago k . En otras palabras, mide la correlación entre los valores Y_t e Y_{t+k} de la serie (Box et al., 2016).

- **Heterocedasticidad:** La volatilidad de los rendimientos de un activo no es constante, sino que cambia a lo largo del tiempo, presentando periodos de alta y baja dispersión. Este fenómeno se conoce como agrupamiento de volatilidad (*volatility clustering*). Su presencia invalida los métodos de estimación tradicionales, haciendo necesario el uso de modelos ARCH/GARCH (Bollerslev, 1986; Engle, 1982).
- **Distribución no normal:** Los rendimientos de los activos financieros suelen desviarse de la distribución normal, exhibiendo exceso de curtosis (colas pesadas) y asimetría. Esto implica que los eventos extremos (grandes variaciones) son más probables de lo que predice una distribución Gaussiana, un hecho crucial para la gestión de riesgos y la correcta especificación de modelos como los GARCH.

$$\text{Exceso de Curtosis} = \frac{\mathbb{E}[(X - \mu)^4]}{\sigma^4} - 3 > 0. \quad (6)$$

4.3. Modelos Econométricos Tradicionales de Predicción

El análisis de series temporales financieras ha estado históricamente dominado por modelos lineales que permiten capturar relaciones dinámicas en los datos y realizar pronósticos bajo ciertos supuestos como la estacionariedad. Entre los más clásicos y utilizados destacan los modelos ARIMA, ARCH y GARCH, ampliamente aplicados en contextos económicos y financieros para modelar el comportamiento de los rendimientos y volatilidad en los mercados financieros.

4.3.1. Modelo Autorregresivo Integrado de Medias Móviles (ARIMA)

El modelo *AutoRegressive Integrated Moving Average*, representado como $ARIMA(p, q, d)$, se trata de una metodología econométrica estocástica diseñada para la modelación y predicción de series temporales univariadas (Box et al., 2016). Su estructura combina tres parámetros enteros no negativos (p, q, d) , cada componente con su respectiva función:

- **AR (Autoregresivo, p)**: modela la independencia lineal de la serie respecto a sus p valores anteriores.
- **I (Integrado, d)**: representa el número d de diferenciaciones requeridas para transformar la serie original no estacionaria en una serie estacionaria.
- **MA (Medias Móviles, q)**: modela la relación entre el valor actual y los errores pasados del modelo.

Matemáticamente, un modelo $ARIMA(p, d, q)$ se expresa como:

$$\Phi_p(L)(1-L)^d Y_t = \Theta_q(L) \varepsilon_t, \quad (7)$$

donde:

- Y_t : valor de la serie en el tiempo t ,
- L : operador rezago ($LY_t = Y_{t-1}$),
- $\Phi_p(L) = 1 - \phi_1 L - \phi_2 L^2 - \dots - \phi_p L^p$: polinomio autorregresivo (AR) de orden p ,
- $(1 - L)^d$: parte integrada (I) que representa las diferencias aplicadas para lograr estacionariedad,
- $\Theta_q(L) = 1 + \theta_1 L + \theta_2 L^2 + \dots + \theta_q L^q$: polinomio de medias móviles (MA) de orden q ,
- ε_t : proceso de ruido blanco con media cero y varianza constante.

4.3.2. Modelo GARCH

El modelo GARCH (*Generalized Autoregressive Conditional Heteroskedasticity*) fue desarrollado con el propósito de modelar la varianza condicional de una serie temporal, permitiendo capturar de manera más realista la heterocedasticidad observada en los rendimientos financieros.

Matemáticamente, un modelo GARCH(1,1) se expresa como:

$$Y_t = \mu + \varepsilon_t, \quad \varepsilon_t \mid \mathcal{I}_{t-1} \sim N(0, h_t), \quad (8)$$

$$h_t = \omega + \alpha \varepsilon_{t-1}^2 + \beta h_{t-1}, \quad (9)$$

donde:

- h_t : varianza condicional en el tiempo t ,
- $\omega > 0$: parámetro constante,
- α : coeficiente asociado al impacto de los choques pasados (ε_{t-1}^2),
- β : coeficiente de persistencia de la varianza,

- $\alpha + \beta < 1$: condición de estacionariedad.

Generalized ARCH o GARCH (Bollerslev, 1986), surge como una extensión del modelo ARCH (Engle, 1982), al incorporar rezagos adicionales de la varianza condicional, lo que permite observar de forma más eficiente la persistencia temporal de la volatilidad en las series financieras.

4.3.3. Limitaciones de los Modelos Econométricos Tradicionales

A pesar de su amplia utilización durante un largo periodo, los modelos econométricos clásicos como ARIMA y GARCH presentan limitaciones al momento de analizar series financieras modernas, sobre todo cuando se habla de mercados volátiles y no lineales, como por ejemplo, las criptomonedas.

En el caso de ARIMA, su desempeño se ve afectado cuando la serie presenta no linealidades o interacciones de alta frecuencia entre las observaciones, mientras que los modelos GARCH, aunque son adecuados para modelar la heterocedasticidad, asumen una varianza condicional que depende solo de los valores pasados, dejando de lado la capacidad de representar movimientos abruptos en la volatilidad.

Frente a estas limitaciones, el desarrollo de enfoques basados en aprendizaje automático (*Machine Learning*) y aprendizaje profundo (*Deep Learning*) ha permitido abordar de mejor manera la naturaleza no lineal y dinámica de las series financieras. Estos modelos aprovechan su capacidad de procesar grandes volúmenes de datos para detectar patrones que los modelos econométricos tradicionales no pueden capturar, mejorando la precisión predictiva y la adaptabilidad a los cambios del mercado.

4.4. Modelos de Aprendizaje Profundo para Series Temporales

El aprendizaje profundo (*deep learning*) ha revolucionado el análisis de series financieras debido a que permite modelar relaciones con alta no linealidad y dependencias de largo plazo que los modelos econométricos tradicionales no logran capturar. El deep learning opera mediante el uso de redes neuronales artificiales con múltiples capas ocultas, lo que les confiere la capacidad de aprender automáticamente representaciones jerárquicas y no lineales directamente a partir de los datos crudos (Goodfellow et al., 2016).

En un comienzo destacan las redes neuronales recurrentes (RNN) y sus variantes LSTM (*Long Short-Term Memory*) y GRU (*Gated Recurrent Unit*), diseñadas para capturar dependencias temporales en secuencias de datos. Mas recientemente, los modelos basados en atención (*Attention Mechanisms*), como los *Transformers*, han introducido un nuevo paradigma al reemplazar la recurrencia por mecanismos que ponderan la relevancia de cada observación pasada, permitiendo un aprendizaje ms eficiente.

4.4.1. Redes Neuronales Recurrentes (RNN)

Las redes neuronales recurrentes (RNN) surgen como una extensión de las redes neuronales tradicionales, incorporando conexiones cíclicas que permiten mantener información de estados anteriores. Esto las convierte en herramientas especialmente adecuadas para modelar datos secuenciales, donde las observaciones pasadas influyen en las futuras, como ocurre en las series financieras.

En su forma más simple, una RNN actualiza su estado oculto h_t en cada instante t mediante la relación:

$$h_t = f(W_h h_{t-1} + W_x x_t + b_h), \quad (10)$$

donde:

- h_t : estado oculto en el tiempo t ,
- h_{t-1} : estado oculto en el instante anterior,
- x_t : vector de entrada en el instante t ,
- W_h y W_x : matrices de pesos para el estado oculto y la entrada, respectivamente,
- b_h : vector de sesgo asociado a la combinación lineal,
- $f(\cdot)$: función de activación no lineal, normalmente $\tanh$ o $ReLU$.

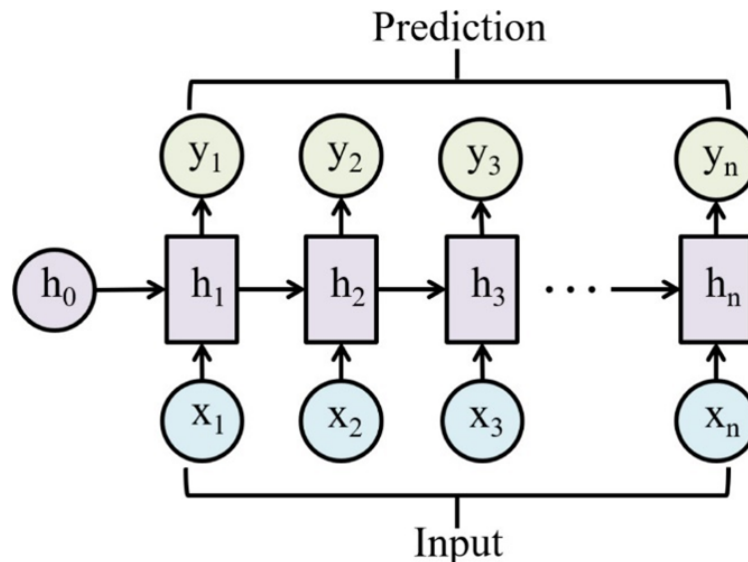


Figura 4.1: Arquitectura básica de una red neuronal recurrente (RNN)

Nota. Tomado de Mienye et al. (2024).

La principal limitación de las RNN en el análisis de series financieras extensas es el problema del desvanecimiento del gradiente (*vanishing gradient*), que impide a la red capturar dependencias significativas a largo plazo, ya que la influencia de pasos de tiempo lejanos se diluye exponencialmente (Zhang et al., 2021).

4.4.2. Long Short-Term Memory (LSTM)

Para mitigar el problema del desvanecimiento del gradiente, se desarrolló la arquitectura *Long Short-Term Memory* (LSTM) (Hochreiter & Schmidhuber, 1997). Las redes LSTM son una forma especializada de RNN, diseñadas para retener información durante periodos extensos de tiempo, resolviendo el desafío de modelar dependencias de largo alcance en series financieras extensas.

El modelo LSTM incorpora una estructura interna denominada celda de memoria C_t , que permite almacenar, olvidar o actualizar información relevante a lo largo del tiempo mediante un conjunto de puertas: la puerta de entrada (*input gate*), la puerta de olvido (*forget gate*) y la puerta de salida (*output gate*). Estas puertas utilizan la función sigmoide (σ) para producir valores entre 0 y 1, que controlan selectivamente que información se almacena, descarta o utiliza para el cálculo del estado oculto. El flujo de información de la red se regula mediante las siguientes ecuaciones, donde σ representa la función sigmoide y $\tanh$ la función de activación hiperbólica:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f), \quad (11)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), \quad (12)$$

$$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C), \quad (13)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t, \quad (14)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \quad (15)$$

$$h_t = o_t * \tanh(C_t), \quad (16)$$

donde:

- f_t : puerta de olvido, controla qué información se descarta de la celda,

- i_t : puerta de entrada, decide qué nueva información se almacena,
- C_t : estado de la celda en el instante t ,
- o_t : puerta de salida, determina qué información se transmite como salida,
- h_t : estado oculto que se propaga al siguiente paso temporal.

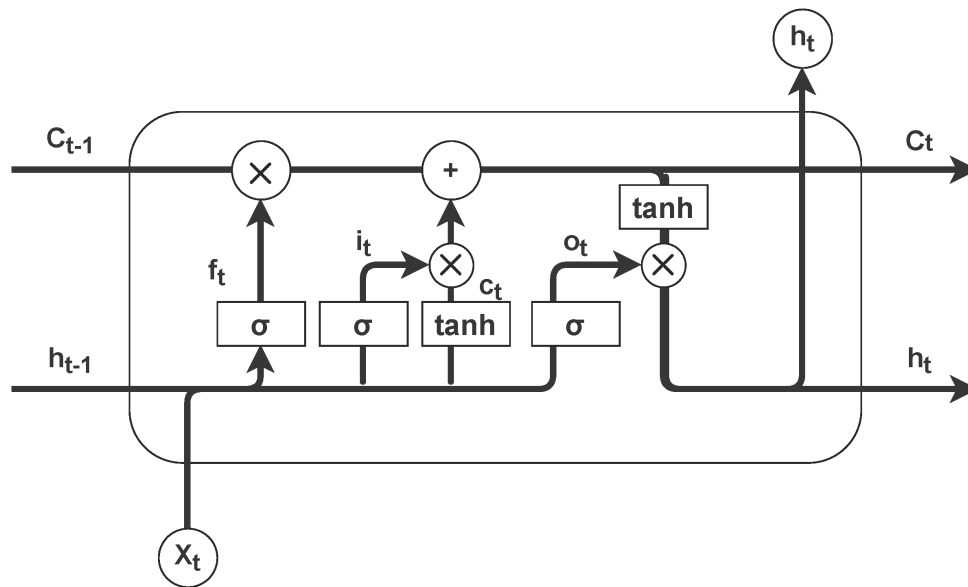


Figura 4.2: Arquitectura interna de una celda LSTM

Nota. Tomado de Mienye et al. (2024).

Gracias a esta arquitectura, las LSTM son capaces de capturar dependencias de largo plazo sin que los gradientes se desvanezcan, lo que las hace especialmente útiles para modelar series financieras, donde las condiciones pasadas pueden tener influencia persistente sobre los valores futuros (Zhang et al., 2021).

4.4.3. Gated Recurrent Unit (GRU)

El modelo Gated Recurrent Unit (GRU), fue propuesto como una alternativa más simple y computacionalmente más eficiente que la arquitectura LSTM (Cho et al., 2014).

A diferencia de las redes LSTM, las redes GRU combinan las funciones de las puertas de entrada y olvido en una sola puerta denominada puerta de actualización (*update gate*), y eliminan la celda de memoria separada. Esto permite que el modelo pueda entrenarse de manera más eficiente, sobre todo cuando se trata de grandes volúmenes de datos.

Matemáticamente, las ecuaciones que definen una celda GRU son:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z), \quad (17)$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r), \quad (18)$$

$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h), \quad (19)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t, \quad (20)$$

donde:

- z_t : puerta de actualización, regula cuánto del estado anterior se conserva,
- r_t : puerta de reinicio, controla cuánta información previa se olvida,
- $\tilde{h}_t$: estado candidato, representa la nueva información propuesta,
- h_t : estado oculto final en el instante t ,
- σ : función sigmoide y $\tanh$: función de activación hiperbólica.

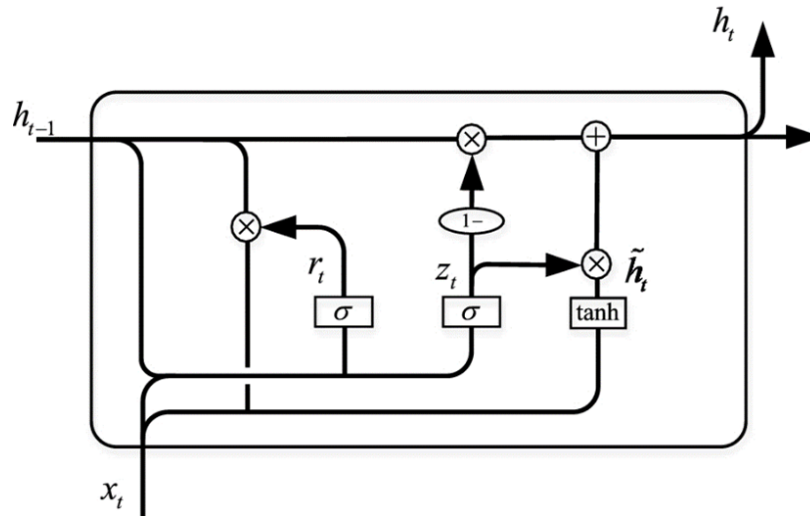


Figura 4.3: Arquitectura interna de una celda GRU

Nota. Tomado de Mienye et al. (2024).

4.4.4. Limitaciones de las Arquitecturas Recurrentes

A pesar de que las redes LSTM y GRU lograron resolver de manera efectiva el problema del gradiente desvanecido y mejoraron la capacidad de modelar dependencias de mediano y largo plazo, ambas mantienen dos limitaciones clave que impiden su aplicación óptima en series temporales de alta frecuencia y gran volumen como las criptomonedas:

- **Dependencia secuencial y baja paralelización:** Todas las variantes de RNN procesan la información de manera secuencial (*token a token*), lo que impide aprovechar completamente el paralelismo computacional. Esta restricción aumenta los tiempos de entrenamiento y limita la capacidad de procesar *batches* de datos de manera eficiente, dificultando el manejo de grandes volúmenes de series temporales.
- **Limitación en la escala de dependencias:** Si bien las redes LSTM y GRU son capaces de modelar dependencias de largo plazo, su rendimiento tiende a degradarse a medida que la longitud de la secuencia de entrada se vuelve excesivamente grande, ya que la información de pasos muy lejanos pierde influencia sobre el estado actual.

Estas limitaciones motivaron el desarrollo de arquitecturas con una visión más global y menos secuencial del procesamiento de la información, con el objetivo de mejorar la eficiencia computacional y la capacidad de capturar relaciones de largo alcance (Vaswani et al., 2017).

4.4.5. Mecanismo de Atención

El Mecanismo de Atención (*Attention Mechanism*) fue propuesto inicialmente para tareas de Procesamiento del Lenguaje Natural (NLP) (Bahdanau et al., 2015), aunque el concepto se fue generalizando para distintos tipos de datos secuenciales. En el contexto de las series temporales, el mecanismo de atención resuelve el cuello de botella de las arquitecturas recurrentes, ya que elimina la necesidad de procesar la información de manera estrictamente secuencial.

Este enfoque permite al *decoder* enfocarse dinámicamente en diferentes partes de la secuencia codificada, asignando pesos de importancia a cada *token* según su relevancia contextual. A diferencia de las arquitecturas recurrentes que dependen de estados ocultos propagados paso a paso, la atención permite que el modelo acceda directamente a toda la secuencia de entrada y asigne distintos niveles de relevancia a cada instante temporal, mejorando significativamente la capacidad de capturar dependencias de largo plazo y patrones no lineales.

Unos años después el concepto fue ampliado con el surgimiento de los Mecanismos de Autoatención (*Self-Attention*), con el objetivo de eliminar por completo la necesidad de estructuras recurrentes o convolucionales para modelar dependencias dentro de una secuencia (Vaswani et al., 2017). A diferencia de la atención clásica, en la que el *decoder* asigna pesos de importancia sobre la secuencia codificada por el *encoder*, la autoatención permite que cada elemento de la secuencia compare su relación con todos los demás elementos de la misma entrada. Este principio dio origen a arquitecturas como la del modelo *Transformer*, que reemplaza completamente la recurrencia mediante el uso exclusivo de mecanismos de autoatención.

4.4.6. Arquitectura del Modelo Transformer

Transformer es un modelo de deep learning publicado por investigadores de Google Brain con una arquitectura de tipo encoder-decoder (Figura 4.4) que se distingue por eliminar completamente la necesidad de recurrencia y convolución, basando toda su capacidad de procesamiento por mecanismos de autoatención y capas feed-forward, que permiten paralelizar el cómputo a lo largo de toda la secuencia y así, capturar dependencias de corto y largo plazo con mayor eficiencia (Vaswani et al., 2017).

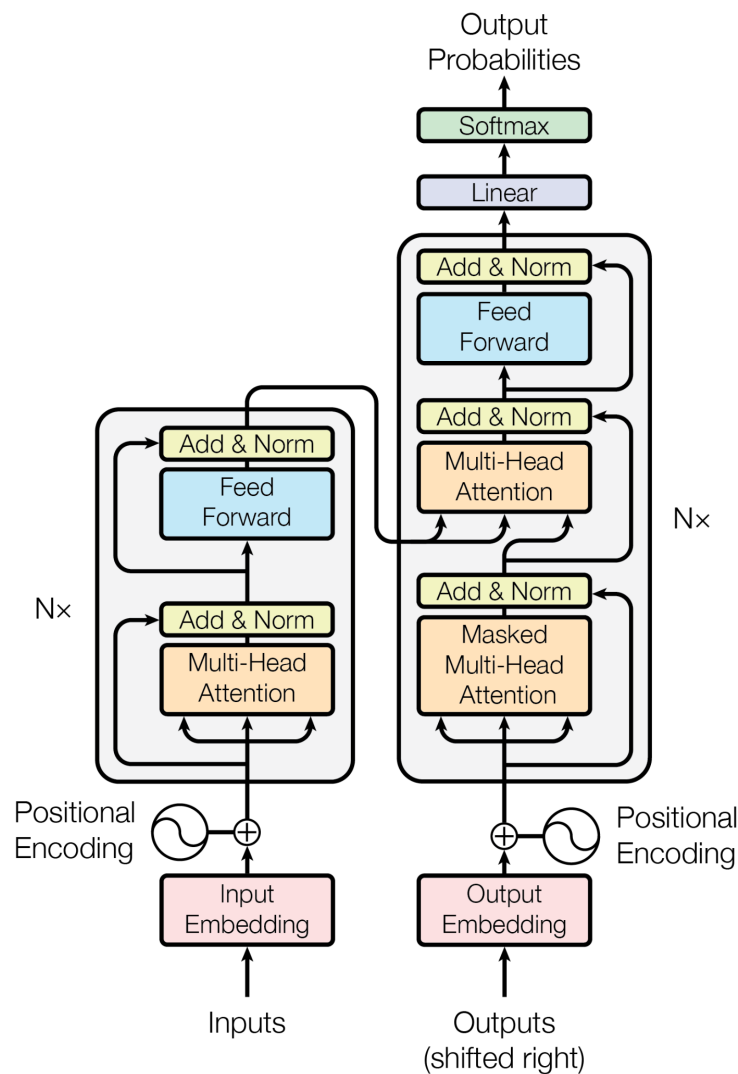


Figura 4.4: Arquitectura Transformer

Nota. Tomado de Vaswani et al. (2017).

a. Etapa de entrada: *Embedding* y Codificación Posicional

Antes de ingresar al encoder y al decoder, la secuencia de entrada y la secuencia objetivo (desplazada a la derecha), respectivamente, pasan por una etapa de representación numérica. Cada elemento de la secuencia se transforma en un vector continuo $Embedding(x)$, de dimensión d_{model} mediante una capa de embedding.

Dado que el Transformer no incorpora recurrencia ni convolución, es necesario agregar información sobre el orden de los elementos dentro de la secuencia. Para ello se utiliza la codificación posicional (*Positional Encoding*), calculada como:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right), PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right), \quad (21)$$

donde pos representa la posición del elemento en la secuencia e i el índice de la dimensión del vector. Estas codificaciones se suman a los vectores de embedding, resultando en la entrada final al primer bloque del encoder:

$$X_{input} = Embedding(x) + PE, \quad (22)$$

donde $X \in \mathbb{R}^{L \times d_{model}}$, de manera análoga, el decoder recibe una secuencia de salida desplazada (*output shifted right*) con su correspondiente codificación posicional, asegurando que las predicciones se generen de izquierda a derecha sin acceso a posiciones futuras.

b. Bloque Encoder

El bloque codificador (encoder) tiene como objetivo procesar y generar una representación contextual de la secuencia de entrada. Está compuesto por N capas idénticas, donde cada capa consta de dos subcapas principales:

1. **Atención Multi-Cabeza (*Multi-Head Attention*):** Permite que cada token (instante de tiempo) en la secuencia de entrada evalúe su relación con todos los demás tokens de la misma

secuencia. Este proceso se formaliza mediante la proyección de la entrada X en tres matrices:

$$Q = XW^Q, \quad (23)$$

$$K = XW^K, \quad (24)$$

$$V = XW^V, \quad (25)$$

donde:

- **Q (Query o consulta):** representa lo que el modelo está buscando de la secuencia,
- **K (Key o clave):** índice que describe las características de los elementos disponibles,
- **V (Value o valor):** contiene la información asociada a cada clave, que se utilizará para generar la salida,
- **W^Q , W^K y W^V :** matrices de pesos aprendibles, encargadas de transformar los vectores de entrada X a los espacios latentes de consultas, claves y valores. Estas matrices permiten que el modelo aprenda que aspectos de los datos deben ser enfatizados o ignorados durante su proceso de atención,
- $W^Q, W^K, W^V \in \mathbb{R}^{d_{model} \times d_k}$, donde d_k corresponde a la dimensión de las claves (keys) y actúa como factor de escalado en el producto punto para estabilizar los gradientes durante el entrenamiento, evitando que los valores del softmax se vuelvan extremadamente pequeños o grandes a medida que aumenta la dimensión del espacio de representación.

La similitud entre Q y K se mide mediante el producto punto escalado (Figura 4.5):

$$\text{Similitud}(Q, K) = \frac{QK^\top}{\sqrt{d_k}}, \quad (26)$$

Los pesos de atención obtenidos a partir de esta similitud se utilizan para ponderar los valores V :

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \quad (27)$$

Para capturar distintos tipos de relaciones en paralelo, se utilizan múltiples cabezas de atención (*Multi-Head Attention*) (Figura 4.5):

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O, \quad (28)$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V), \quad (29)$$

donde W^O es otra matriz de pesos aprendible que proyecta la concatenación de las distintas cabezas de atención al espacio de salida del modelo. Este diseño multi cabeza permite que el modelo atienda a diferentes subespacios de características de manera simultánea.

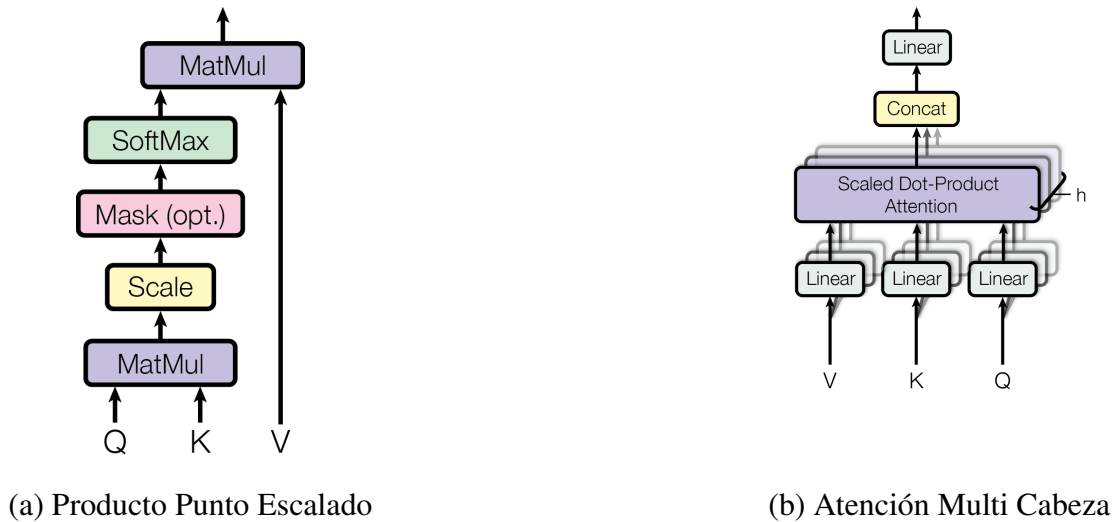


Figura 4.5: Mecanismo de atención del modelo Transformer. (a) Producto Punto Escalado, (b) Atención Multi Cabeza

Nota. Tomado de Vaswani et al. (2017).

2. Red Feed-Forward y Normalización: Cada posición procesada por los mecanismos de atención pasa por una red *feed-forward* independiente (Figura 4.4):

$$FFN(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2, \quad (30)$$

Finalmente, se aplican conexiones residuales y normalización por capas para estabilizar el entrenamiento y mejorar la propagación del gradiente:

$$X^{(l)} = \text{LayerNorm}(X^{(l-1)} + \text{Sublayer}(X^{(l-1)})), \quad (31)$$

El resultado final del codificador corresponde al vector $Z = X^{(L)}$, el cual sintetiza la información contextual de toda la secuencia de entrada.

c. Bloque Decoder

El bloque decodificador (decoder) genera la secuencia de salida de manera autorregresiva, prediciendo cada elemento a partir de los anteriores y utilizando la información proveniente del encoder. Cada una de sus N capas consta de tres subcapas:

1. **Autoatención Enmascarada (*Masked Self-Attention*):** Similar a la atención del encoder, pero con una diferencia importante, se incluye una máscara triangular que impide cualquier observación en elementos futuros de la secuencia (Figura 4.5). De este modo, el modelo solo puede usar la información de los tokens previos, garantizando que la generación de la salida respete el orden temporal.
2. **Atención Cruzada (*Encoder-Decoder Attention*):** Permite que el decoder pondere los vectores de representación enviados por el encoder. En este caso, las consultas vienen del decoder (Q_{dec}), mientras que las claves y valores provienen del encoder (K_{enc}, V_{enc}):

$$\text{Attention}(Q_{dec}, K_{enc}, V_{enc}) = \text{softmax}\left(\frac{Q_{dec}K_{enc}^\top}{\sqrt{d_k}}\right)V_{enc}, \quad (32)$$

3. **Red Feed-Forward y Normalización:** Igual que en el encoder, cada subcapa se acompaña de conexiones residuales y normalización.

Finalmente, la salida del decoder pasa por una capa lineal y una función softmax, que transforman los vectores de representación en probabilidades sobre el espacio de salida.

4.4.7. Adaptación del Modelo Transformer a Series Temporales

Originalmente, el modelo Transformer fue diseñado para tareas de Procesamiento del Lenguaje Natural (NLP). Para su utilización en tareas de predicción y modelado de series temporales, se requiere realizar una serie de adaptaciones específicas:

1. En primer lugar, se reemplaza la capa de embedding por un embedding temporal, donde cada observación multivariable $\mathbf{x}_t \in \mathbb{R}^d$ se proyecta a un espacio latente de dimensión d_{model} mediante:

$$E_t = x_t W_E + b_E, \quad (33)$$

donde $W_E \in \mathbb{R}^{d \times d_{model}}$ es la matriz de pesos que aprende la representación latente de las variables, y b_E es un vector de sesgo.

2. Se incorporan codificaciones posicionales que preservan el orden temporal, comúnmente expresadas mediante funciones trigonométricas (seno y coseno) que asignan una representación única a cada instante t .
3. Se implementan técnicas de reducción de las secuencias de entrada (*windowing* o *downsampling*), que agrupan subconjuntos de datos, reduciendo la longitud de la secuencia y, con ello, la carga computacional del modelo.
4. Ajuste de los hiperparámetros del Transformer, tales como la longitud L de las secuencias de entrada y salida, el número de capas encoder-decoder y el número de cabezas de atención h .

A pesar de sus avances, el Transformer convencional enfrenta limitaciones importantes al trabajar con series temporales extensas (Zhou et al., 2021):

- Su complejidad computacional es cuadrática, del orden $O(L^2)$, en función de la longitud de la secuencia L . Esto significa que al aumentar el número de pasos temporales, el costo de cálculo crece de manera no lineal, lo que vuelve al modelo ineficiente para series temporales largas o de alta resolución.
- El elevado consumo de memoria limita la escalabilidad en entornos de datos de alta frecuencia y gran volumen, como los mercados de criptomonedas.
- Hay una limitación inherente de la arquitectura encoder-decoder tradicional, que resulta en una caída de la velocidad debido a la decodificación dinámica paso a paso.

4.4.8. Modelo Informer

El modelo *Informer* surge como una extensión del Transformer, optimizado para la predicción de series temporales de largo horizonte (*Long Sequence Time-Series Forecasting, LSTF*) (Zhou et al., 2021). Su diseño aborda las limitaciones de complejidad cuadrática y el elevado consumo de memoria del Transformer original, manteniendo su capacidad para capturar dependencias globales mediante mecanismos de atención.

A nivel estructural, el modelo Informer conserva la arquitectura encoder-decoder, pero introduce tres innovaciones fundamentales:

1. **Autoatención Probabilística Esparsa (ProbSparse Self-Attention):** Para optimizar el cálculo de atención del Transformer, el modelo Informer introduce el mecanismo de Autoatención Probabilística Esparsa (*ProbSparse Self-Attention*), basado en la hipótesis de que solo un subconjunto de queries contribuye significativamente al mapa de atención.

Cada query q_i se evalúa mediante una métrica de esparcidad M :

$$M(q_i, K) = \max_j \left(\frac{q_i k_j^\top}{\sqrt{d_k}} \right) - \frac{1}{L} \sum_{j=1}^L \frac{q_i k_j^\top}{\sqrt{d_k}}, \quad (34)$$

donde:

- q_i : query del elemento i ,
- K : matriz de claves,
- k_j : key del elemento j ,
- L : longitud de la secuencia,
- d_k : dimensión de las claves,
- $\max_j \left(\frac{q_i k_j^\top}{\sqrt{d_k}} \right)$: valor máximo de similitud entre query q_i y alguna key k_j .

Mientras mayor sea la esparcidad $M(q_i, K)$, más concentrada e informativa es la atención asociada a la query q_i . El modelo solo selecciona las *queries* con los valores más altos de $M(q_i, K)$, reduciendo el cálculo a un subconjunto $Q' \subset Q$:

$$A_{\text{ProbSparse}}(Q', K, V) = \text{Softmax} \left(\frac{Q' K^\top}{\sqrt{d_k}} \right) V, \quad (35)$$

Con ello, la complejidad temporal y espacial del mecanismo de atención se reduce de $O(L^2)$ a $O(L \log L)$, sin pérdida sustancial de precisión.

2. **Destilación de Autoatención (Self-Attention Distiling)**: Después de la atención probabilística, se aplica un proceso de *distiling* que reduce progresivamente la longitud de la secuencia y elimina información redundante. Formalmente, tras la capa j , la salida del encoder se actualiza mediante:

$$X_{j+1} = \text{MaxPool} \left(\text{ELU} \left(\text{Conv1D} \left([X_j]_{AB} \right) \right) \right), \quad (36)$$

donde:

- $[X_j]_{AB}$: representa la salida del bloque de atención (*Attention Block*) aplicado a X_j ,
- $\text{Conv1D}(\cdot)$: corresponde a una convolución unidimensional aplicada sobre el eje temporal.

Esta operación extrae patrones locales mediante filtros de tamaño K ,

- $\text{ELU}(\cdot)$: función de activación *Exponential Linear Unit*,
- $\text{MaxPool}(\cdot)$: operación de *pooling* que reduce la longitud de la secuencia a la mitad en cada capa, disminuyendo la complejidad total del encoder,
- X_{j+1} : corresponde a la salida de la capa j , es decir, la secuencia transformada tras la convolución, activación y submuestreo, que servirá como entrada a la siguiente capa o al módulo jerárquico del encoder.

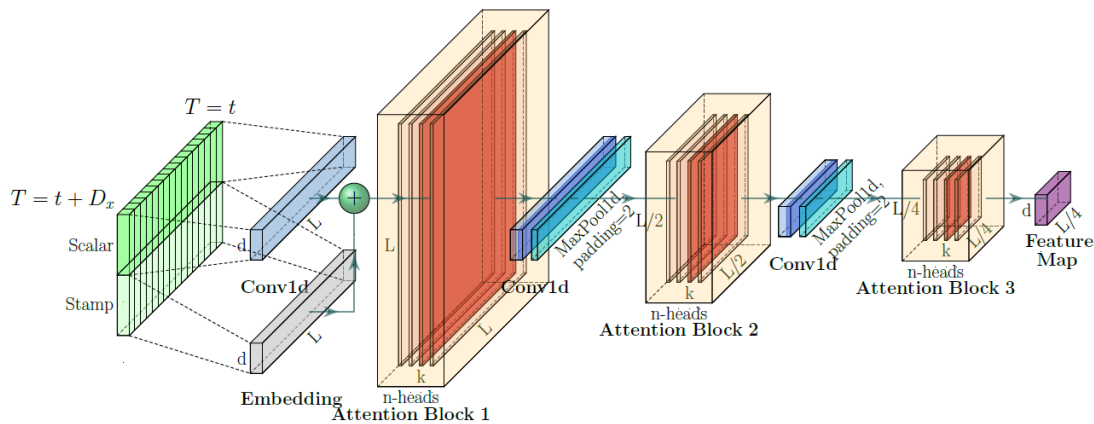


Figura 4.6: Estructura de un *stack* individual dentro del codificador

Nota. Tomado de Zhou et al. (2021).

La representación anterior (Figura 4.6), muestra la estructura de un *stack* individual del codificador (*encoder*) asociado al modelo Informer, que representa una unidad básica del proceso de codificación jerárquica.

En primer lugar, cada *stack* comienza con las entradas de la serie temporal que son transformadas mediante capas de embedding que incluyen operaciones Conv1D, destinadas a generar las representaciones iniciales que ingresan al primer bloque de atención (*Attention Block*).

Cuando la secuencia entra al bloque de atención, se aplica el mecanismo de Autoatención Probabilística Esparsa (*ProbSparse Self-Attention*), en el cual el modelo determina que partes de la secuencia son más relevantes para cada elemento de esta. Posteriormente, la secuencia transformada pasa por el proceso de Destilación de Autoatención (*Self-Attention Distilling*), que

consta de tres suboperaciones: Conv1D, ELU y MaxPool. Es precisamente el módulo MaxPool el que reduce la longitud de la secuencia a la mitad en cada stack, implementando así la codificación jerárquica.

El resultado final de este proceso es una secuencia más corta, con un contexto profundo, y una complejidad reducida, lista para pasar al siguiente bloque de atención hasta alcanzar la última capa del Encoder, antes de ser procesada por el decoder (Zhou et al., 2021).

3. **Decoder Generativo Eficiente:** El *decoder* del modelo Informer conserva la estructura básica del Transformer, pero combina una atención probabilística esparsa con un mecanismo de enmascaramiento temporal, además de un enfoque de predicción no autorregresiva. Esta combinación permite mantener la direccionalidad de la secuencia y reducir la complejidad computacional del proceso de atención. De este modo, el modelo puede generar toda la secuencia de salida en una sola operación.

La entrada (*input*) al decoder es una concatenación de:

$$X_{\text{dec}} = \text{Concat}(X_{\text{token}}, X_0), \quad (37)$$

donde:

- X_{dec} : vector o matriz de entrada del decoder,
- X_{token} : fragmento de la secuencia anterior, usualmente llamados *start token* o contexto previo,
- X_0 : marcador inicial de ceros para rellenar posiciones futuras en las que el modelo debe predecir.

Durante el entrenamiento, se aplica una máscara triangular M para asegurar la direccionalidad temporal:

$$A_{\text{masked}}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}} + M\right) V, \quad (38)$$

A diferencia del enfoque autorregresivo clásico, el Informer predice el conjunto completo de salidas $\hat{Y}$ en una única *forward pass*, evitando la acumulación de errores y reduciendo el tiempo de inferencia de $O(L)$ a $O(1)$.

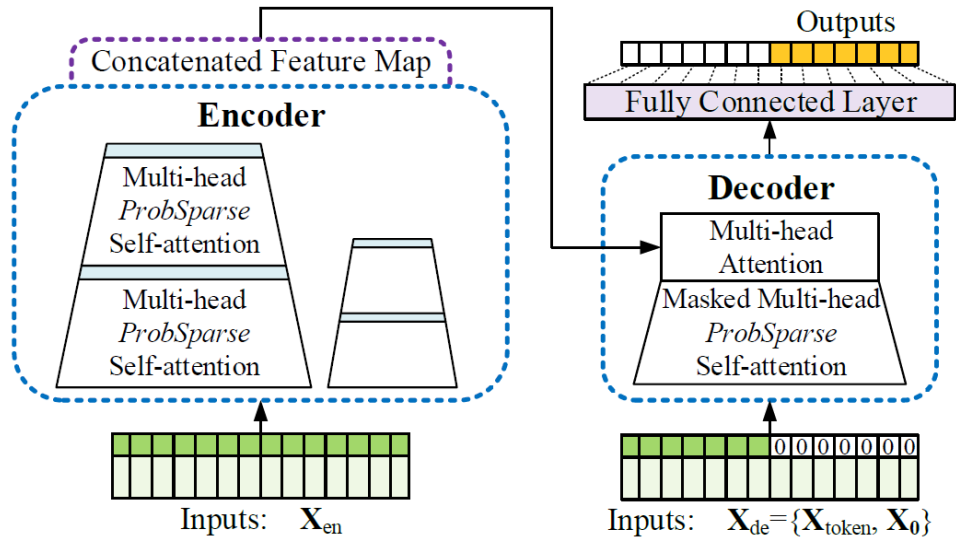


Figura 4.7: Arquitectura general del modelo Informer

Nota. Tomado de Zhou et al. (2021).

En la Figura 4.7, se presenta la estructura general del modelo Informer, que integra los componentes descritos anteriormente. En ella se observa un encoder basado en mecanismos de autoatención probabilística esparsa y un proceso jerárquico de destilación que reduce la longitud de la secuencia, junto con decoder que aplica atención enmascarada y predicción no autorregresiva para generar la salida completa en una única pasada de inferencia (*forward pass*).

Además de su arquitectura, el modelo emplea durante el entrenamiento una función de pérdida estándar que permite cuantificar la discrepancia entre las predicciones y los valores reales, favoreciendo el ajuste de los parámetros del modelo.

Función de Pérdida: La función de pérdida que se utiliza en el modelo Informer corresponde al error cuadrático medio (MSE):

$$\mathcal{L}_{\text{MSE}} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2, \quad (39)$$

donde:

- $\mathcal{L}_{\text{MSE}}$: valor de la función de pérdida, que cuantifica el error promedio cuadrático entre las predicciones $\hat{Y}$ y los valores reales Y .
- N : número total de observaciones utilizadas en el cálculo.

4.5. Eficiencia de los Mercados Financieros

El estudio de la eficiencia de los mercados financieros constituye uno de los pilares fundamentales de la teoría moderna de las finanzas. Su estudio busca determinar en qué medida los precios de los activos reflejan de manera completa toda la información disponible en el mercado (Fama, 1970).

Un mercado se considera eficiente cuando los precios de los activos se ajustan de forma inmediata ante nueva información, impidiendo la obtención sistemática de rendimientos anormales mediante estrategias de inversión basadas en aprovechar dicha información para comprar o vender el activo.

4.5.1. Hipótesis de los Mercados Eficientes (HME)

La Hipótesis de los Mercados Eficientes (*Efficient Market Hypothesis, EMH*), formulada por Eugene Fama (1970), sostiene que en un mercado eficiente los precios de los activos ya incorporan

toda la información relevante existente, de modo que ninguna estrategia de *trading* basada en información pasada o pública puede generar beneficios superiores al promedio del mercado.

La EMH se clasifica comúnmente en tres formas de eficiencia, según el tipo de información que se considera incorporada en los precios:

- **Eficiencia débil:** Plantea que los precios actuales reflejan completamente toda la información contenida en los precios pasados. En consecuencia, los rendimientos históricos no poseen poder predictivo sobre los rendimientos futuros, por lo que las estrategias basadas en análisis técnico carecen de eficacia.
- **Eficiencia semifuerte:** Establece que los precios de los activos incorporan tanto la información pasada como la información pública disponible, incluyendo reportes financieros, noticias económicas, y otros datos de acceso general. En esta forma de eficiencia, el análisis fundamental tampoco permitiría obtener rendimientos anormales.
- **Eficiencia fuerte:** Postula que los precios reflejan toda la información existente, tanto pública como privada (interna o privilegiada). Si esta condición se cumpliera, ni siquiera los inversionistas con información exclusiva podrían obtener rendimientos superiores al promedio del mercado.

En la práctica, la hipótesis de eficiencia de mercado (HME) ha sido objeto de amplio debate. Si bien explica el funcionamiento ideal de los mercados financieros, diversos estudios han mostrado evidencias de comportamientos no aleatorios, anomalías de calendario, burbujas y efectos de sobre reacción que cuestionan su validez empírica. Estos fenómenos sugieren que los mercados pueden presentar grados variables de eficiencia a lo largo del tiempo bajo distintas condiciones, de volatilidad, liquidez y participación institucional.

4.5.2. Hipótesis de Mercados Adaptativos (HMA)

A partir de las limitaciones de la hipótesis de eficiencia tradicional, Andrew Lo (2004) propuso la Hipótesis de Mercados Adaptativos (*Adaptive Market Hypothesis, AMH*), la cual plantea que la eficiencia no es una condición estática, sino más bien un proceso dinámico que actúa conforme los participantes del mercado se adaptan al entorno económico, tecnológico y conductual. Según esta perspectiva, los mercados pueden ser eficientes en ciertos periodos y volverse ineficientes en otros, dependiendo de factores como la competencia, la innovación o la disponibilidad de información (Lo, 2004).

Este enfoque se puede evidenciar más fácilmente en mercados como el de criptomonedas, que se caracteriza por ser altamente volátil, descentralizado y estar abierto todo el tiempo. Debido a lo anterior, el mercado de criptomonedas se encuentra expuesto a variaciones estructurales derivadas de ciclos de mercado y del evento de *halving* de Bitcoin, que reduce la emisión de nuevas monedas aproximadamente cada cuatro años.

Si bien el evento de halving es exclusivo de Bitcoin, cualquier movimiento en Bitcoin, en general, afecta a todas las demás criptomonedas. Estos ciclos de halving generan alteraciones en la dinámica de oferta y demanda, afectando la liquidez, la percepción de escasez y, en consecuencia, el comportamiento de los inversionistas. Como resultado, la eficiencia del mercado tiende a fluctuar entre fases de alta y baja previsibilidad, reflejando la naturaleza adaptativa descrita por Lo (2004).

En este contexto, el uso de modelos dinámicos de predicción, como el *Informer*, permite capturar estas variaciones adaptativas y analizar cómo cambia la eficiencia a lo largo del tiempo.

En resumen, ambos modelos constituyen el marco conceptual que orienta el análisis de eficiencia implícita del mercado de criptomonedas en esta investigación. La eficiencia se evaluará de manera empírica a través de la capacidad predictiva del modelo *Informer*, considerando que una menor capacidad de predicción indicará un mercado más eficiente, mientras que una mayor precisión reflejará un comportamiento ineficiente o predecible.

5. Metodología

5.1. Construcción del Conjunto de Datos

Para la implementación de los modelos de pronóstico se construyó un conjunto de datos (*dataset*) específicamente diseñado para los objetivos de esta investigación. Este dataset integra información del mercado de Bitcoin, indicadores de análisis técnico, variables macroeconómicas y financieras relevantes, y métricas asociadas al funcionamiento de la red, con el propósito de modelar el retorno logarítmico diario de la criptomoneda.

El período analizado abarca desde el 1 de diciembre de 2014 hasta el 31 de octubre de 2025, cubriendo múltiples ciclos de mercado y contextos macroeconómicos, episodios de alta y baja volatilidad, además de eventos estructurales relevantes, como tres ciclos de halving, y sucesos globales críticos, entre los que destaca la pandemia de COVID-19.

Los datos provienen en su mayoría de dos sitios web. La información histórica asociada a Bitcoin fue descargada desde *Yahoo Finance* y *Blockchain Web*, mientras que las series macroeconómicas y financieras fueron recopiladas en su mayoría desde la base de datos de la *Federal Reserve Economic Data* (FRED), la cual ofrece series oficiales y actualizadas de instituciones económicas relevantes.

Tras la recopilación de los datos, se llevaron a cabo procesos de depuración, estandarización, transformación y normalización. Estas etapas resultaron fundamentales para asegurar la consistencia temporal entre las distintas fuentes antes de consolidar el dataset final y desarrollar los modelos de predicción. El dataset definitivo utilizado para esta investigación está compuesto por 3.988 observaciones diarias y 22 variables, lo que equivale a un total de 87.736 valores registrados.

5.2. Descripción de las Variables

Con el objetivo de determinar los principales factores que potencialmente influyen en la dinámica del precio de Bitcoin, se seleccionó un conjunto de variables provenientes tanto del mercado de criptomonedas como del ámbito técnico, macroeconómico, financiero y del comportamiento propio de la red.

La selección de estas variables se fundamenta en literatura económica y financiera que señala que el precio de Bitcoin puede estar influido por elementos como la liquidez, el sentimiento de mercado, las condiciones monetarias, las expectativas de inflación, la percepción de riesgo, la tasa de interés y factores estructurales o tecnológicos vinculados a la red blockchain, entre otros.

Analizar los ciclos de halving de Bitcoin cobra especial relevancia, pues estos eventos han ejercido una influencia estructural en la dinámica de oferta del activo, afectando consecuentemente sus patrones de valoración a mediano y largo plazo. El gráfico ilustra claramente la coincidencia temporal entre los halvings y cambios significativos en la estructura de precios. (Figura 5.1).



Figura 5.1: Precio histórico de Bitcoin (BTCUSD) y sus ciclos de halving

Nota. Elaboración propia en base a datos de Yahoo Finance.

5.2.1. Variable Objetivo

La variable objetivo corresponde al **retorno logarítmico** (Ecuación 4) diario del precio de Bitcoin. El uso de retorno logarítmico es estándar en finanzas debido a que los niveles de precios suelen ser no estacionarios, en cambio, las diferencias como los retornos logarítmicos tienden a ser estacionarios, con una varianza estabilizada, y por lo tanto, son más adecuadas para modelos de series de tiempo (Hamilton, 1994). Cabe destacar que, si bien la estacionariedad no es un requisito estricto para modelos modernos de deep learning, si contribuye a mejorar la calidad y estabilidad de las predicciones.

5.2.2. Variables predictoras

A partir de la variable objetivo, las variables predictoras se agrupan en las siguientes categorías:

1. Mercado de criptomonedas
2. Indicadores de análisis técnico.
3. Métricas asociadas al funcionamiento de la red blockchain.
4. Variables de halving.
5. Variables macroeconómicas y financieras.

1. Mercado de criptomonedas

Las variables agrupadas en esta categoría buscan capturar la dinámica propia del mercado de Bitcoin, entendida como el resultado de la interacción entre oferta y demanda, liquidez, volatilidad intradía, actividad económica en la red y señales estructurales asociadas al comportamiento de inversionistas y usuarios. Las variables asociadas a esta categoría son:

- **Volumen transado:** Representa el volumen de Bitcoins transados diariamente. Es un indicador clave de liquidez y de participación de mercado. Los cambios abruptos en este indicador suelen asociarse a fases de euforia, pánico o movimientos significativos en la estructura de ordenes. La literatura empírica del mercado bursátil muestra que incrementos sostenidos en volumen suelen coincidir con cambios de tendencia o rupturas de rango (Karpoff, 1987). (Figura A.1)
- **Indicador Intradía de Parkinson:** Mide la amplitud relativa entre el máximo y mínimo del día. Este indicador es una aproximación directa al componente central del estimador de Parkinson (Parkinson, 1980), capturando la volatilidad intradía y permitiendo al modelo identificar entornos de alta incertidumbre. Bitcoin es un activo con volatilidad históricamente elevada, por lo que este tipo de señal es especialmente relevante para modelos que buscan anticipar retornos de corto plazo. Matemáticamente, este rango intradía puede expresarse como:

$$\ln \left(\frac{H_t}{L_t} \right), \quad (40)$$

- **Volatilidad móvil a 7 días:** La volatilidad del retorno calculada en una ventana móvil de 7 días. Esta variable introduce una medida de volatilidad suavizada, permitiendo que el modelo distinga fases de calma o turbulencia de manera menos sensible al ruido diario. Matemáticamente la fórmula es:

$$\sigma_{7,t} = \left(\frac{1}{6} \sum_{i=0}^6 (r_{t-i} - \bar{r}_{7,t})^2 \right)^{1/2}, \quad (41)$$

2. Indicadores de análisis técnico

Diseñados para capturar tendencias, momentum, cambios en la fuerza relativa del mercado, variaciones en la volatilidad, entre otros. Estos indicadores son ampliamente utilizados en finanzas cuantitativas y trading algorítmico, ya que permiten caracterizar con precisión el estado dinámico del activo y facilitar la detección de puntos de inflexión o aceleración en la trayectoria del precio,

lo que permite tomar decisiones con mayor sustento. Se escogieron los siguientes indicadores de análisis técnico:

- **Promedio Móvil Exponencial (EMA 7 y EMA 30):** Diferencia relativa entre el precio de Bitcoin y sus promedios móviles exponenciales calculados a 7 y 30 días, respectivamente. Cuando el precio se sitúa persistentemente por encima de estas medias, suele interpretarse como una fase de impulso alcista. Por otro lado, las desviaciones negativas prolongadas reflejan presión bajista.
- **Índice de Fuerza Relativa (RSI 14):** El RSI 14 (*Relative Strength Index*) es un indicador que evalúa la magnitud relativa de las ganancias y pérdidas de los últimos 14 días para determinar condiciones de sobrecompra o sobreventa (Wilder, 1978).
- **Histograma del MACD:** El MACD (*Moving Average Convergence Divergence*) mide la interacción entre dos promedios móviles exponenciales y su línea de señal, reflejando el impulso direccional del precio. El histograma, que corresponde a la diferencia entre ambas curvas, captura la aceleración del momentum. Valores positivos crecientes indican fortalecimiento de la presión compradora, mientras que valores negativos crecientes sugieren un aumento en la presión vendedora.
- **Indicador %B de Bandas de Bollinger:** Este indicador describe la posición relativa del precio dentro de las Bandas de Bollinger, permitiendo evaluar zonas de presión alcista o bajista y detectar cambios en la volatilidad. Un valor cercano a 1 indica proximidad a la banda superior, lo que puede asociarse a impulso alcista o sobrecompra, mientras que valores cercanos a 0 reflejan cercanía a la banda inferior, vinculada a presión bajista o sobreventa (Bollinger, 2002).

3. Métricas asociadas al funcionamiento de la red blockchain

Proviene directamente de la actividad interna del protocolo Bitcoin. A diferencia de los indicadores financieros tradicionales, estas métricas permiten capturar dimensiones tecnológicas esenciales, tales como la seguridad del sistema, el nivel de competencia entre mineros, la actividad económica registrada en cadena y la intensidad de uso de la red.

- **Direcciones activas:** El número de direcciones activas en la red Bitcoin. Las active addresses son una aproximación al uso real de la red blockchain. Se ha documentado que cuando se incrementa la cantidad de direcciones activas suelen asociarse a fases de adopción creciente o demanda orgánica por transacciones, diferenciándose del mero comportamiento especulativo (Liu et al., 2025). (Figura A.2)
- **Tasa de hash:** El hash rate es un indicador fundamental para medir la capacidad computacional dedicada al minado. La tasa de hash refleja tanto la seguridad de la red como el nivel de inversión en infraestructura minera. Fluctuaciones importantes en este indicador pueden anticipar cambios estructurales vinculados a costos de producción, presión vendedora de mineros o variaciones de incentivos económicos (Hayes, 2019). (Figura A.3)

4. Variables de halving

Destinadas a capturar la estructura cíclica propia de Bitcoin derivada de los eventos de halving. Cada halving reduce a la mitad la recompensa por bloque que reciben los mineros, generando cambios discretos en la oferta nueva de bitcoins y, potencialmente, en los incentivos de participación y en la dinámica de precio.

- **Days since halving:** Variable que mide el número de días transcurridos desde el último halving de Bitcoin. Esta métrica captura la posición temporal dentro del ciclo de halving que históricamente ha generado cambios relevantes en oferta, dinámica de precios y sentimiento de mercado. Al incluir esta variable, el modelo incorpora explícitamente la dimensión cíclica propia del funcionamiento de Bitcoin.
- **Days to next halving:** Cuantifica los días que faltan para el siguiente halving de Bitcoin. Esta variable representa la fase del ciclo previa al ajuste de emisión y permite incorporar al modelo posibles dinámicas asociadas a la anticipación del evento por parte de los participantes del mercado.

5. Variables macroeconómicas y financieras

Las variables incluidas buscan capturar condiciones monetarias, financieras y de riesgo que influyen indirectamente en el comportamiento de Bitcoin. Incorporarlas permite que los modelos identifiquen los patrones y las tendencias macroeconómicas que moldean los precios, especialmente en un periodo caracterizado por shocks significativos como la política monetaria después de la pandemia de COVID-19, episodios de inflación elevada y variaciones abruptas en tasas de interés.

- **Índice del Dólar Estadounidense (DTWEXBGS):** Representa la fortaleza global del dólar frente a una canasta amplia de monedas. Este indicador sintetiza condiciones externas relevantes para los mercados financieros, ya que el dólar suele actuar como activo refugio en escenarios de aversión al riesgo. (Figura A.4)
- **Índice de volatilidad implícita VIX (VIXCLS):** El VIX (*Volatility Index*), también conocido como Índice de Volatilidad del CBOE, es un indicador que mide la volatilidad esperada del mercado estadounidense durante los próximos 30 días, basada en los precios de las opciones del S&P 500. El VIX es una medida ampliamente utilizada para representar la incertidumbre y el riesgo percibido en el mercado bursátil estadounidense (Whaley, 1993). (Figura A.5)
- **Tasa de Política Monetaria (FEDFUNDS):** Es la tasa objetivo definida por la Reserva Federal de Estados Unidos y constituye el principal instrumento de política monetaria. Aumentos en la FEDFUNDS suelen asociarse a condiciones financieras más estrictas, disminución de la liquidez y depreciación de activos riesgosos. Por el contrario, tasas bajas favorecen la expansión de crédito y el apetito por riesgo. En el contexto de Bitcoin, numerosos estudios han encontrado que la política monetaria influye indirectamente en los precios al modificar tanto la liquidez agregada como las expectativas de retorno ajustado por riesgo (Jiang et al., 2022). (Figura A.6)
- **Índice MOVE (volatilidad del mercado de bonos):** Mide la volatilidad implícita del mercado de bonos del Tesoro estadounidense. Es una señal clave del nivel de incertidumbre en los mercados

de renta fija, particularmente sensible a expectativas de inflación, cambios en tasas y tensiones macroeconómicas. Un aumento en el MOVE suele reflejar mayor inestabilidad financiera, lo que puede traducirse en ajustes en carteras globales (Kumar, 2017) y, en algunos casos, en ventas de activos de alto riesgo. Para modelos predictivos, esta variable ayuda a capturar entornos de estrés sistémico o cambios abruptos en el ciclo monetario. (Figura A.7)

- **Rendimiento del bono del tesoro a 10 años (DGS10):** Es una referencia para el costo del dinero a largo plazo y sintetiza expectativas de crecimiento, inflación y riesgo macroeconómico. Incrementos sostenidos en la tasa tienden a encarecer el financiamiento y reducir el atractivo de activos sin flujo de caja como Bitcoin. Por el contrario, tasas decrecientes suelen estimular la búsqueda de alternativas con mayor retorno potencial. (Figura A.8)
- **Inflación implícita a 10 años (T10YIE):** Corresponde al *breakeven inflation rate*, que mide la inflación esperada a 10 años implícita en los precios de bonos protegidos frente a inflación (TIPS). Refleja las expectativas del mercado respecto de la evolución futura del nivel de precios. Bitcoin ha sido frecuentemente debatido como un potencial refugio frente a inflación, por lo que incorporar una medida de expectativas inflacionarias permite evaluar si cambios en este componente macroeconómico se correlacionan con el precio de la criptomoneda. (Figura A.9)
- **Precio del oro (GOLD):** Considerado un activo refugio y un depósito de valor en periodos de incertidumbre económica. Dado que parte del discurso en torno a Bitcoin lo posiciona como “oro digital”, analizar su relación con el comportamiento del oro físico resulta relevante para evaluar similitudes o divergencias entre ambos activos. Flujos de capital hacia el oro frente a activos riesgosos como el Bitcoin pueden ofrecer señales sobre la dirección de movimientos, como señalan algunos trabajos anteriores (Long et al., 2021). (Figura A.10)
- **Oferta monetaria (M2SL):** La oferta monetaria M2 refleja la cantidad de liquidez en la economía. Crecimientos acelerados suelen asociarse con condiciones financieras expansivas, mientras que contracciones o estabilización de M2 corresponden a entornos más restrictivos. Teóricamente, una mayor liquidez podría fomentar la inversión en activos especulativos, aunque este vínculo

no es siempre estable. Su inclusión permite capturar cambios generales en la disponibilidad de dinero en la economía. (Figura A.11)

Tabla 5.1: Resumen de las variables macroeconómicas y financieras incluidas en el estudio

Variable	Descripción
DTWEXBGS	Índice del valor del dólar estadounidense frente a una canasta global de monedas. Captura condiciones financieras internacionales y flujos hacia activos de riesgo.
VIXCLS	Índice de volatilidad implícita del mercado bursátil estadounidense. Refleja percepciones de incertidumbre, aversión al riesgo y tensiones financieras.
FEDFUNDS	Tasa de fondos federales establecida por la Reserva Federal. Representa el nivel de restricción monetaria y disponibilidad de liquidez.
MOVE	Volatilidad implícita del mercado de bonos del Tesoro de EE. UU. Indicador clave de expectativas sobre tasas de interés y estrés macroeconómico.
DGS10	Rendimiento del bono del Tesoro a 10 años. Resume expectativas de crecimiento, inflación y costo del dinero a largo plazo.
T10YIE	Inflación implícita a 10 años (breakeven). Refleja las expectativas inflacionarias del mercado para la próxima década.
GOLD	Precio del oro, considerado un activo refugio. Permite comparar el comportamiento de Bitcoin con activos tradicionales de resguardo.
M2SL	Oferta monetaria M2. Captura cambios en la liquidez agregada y efectos de políticas monetarias expansivas o contractivas.

Nota. Elaboración propia.

5.3. Partición Temporal del Conjunto de Datos

La separación temporal garantiza que cada conjunto respete estrictamente el orden cronológico, reproduciendo de manera fiel el escenario real de predicción donde el futuro nunca puede utilizarse para estimar el pasado. El dataset se separó cronológicamente en *train*, *validation* y *test*. Los modelos se entrenaron solo con información del primer tramo, se ajustaron en el segundo y se evaluaron en un período posterior que el modelo nunca vio, asegurando ausencia de sesgo de anticipación (*look-ahead bias*).

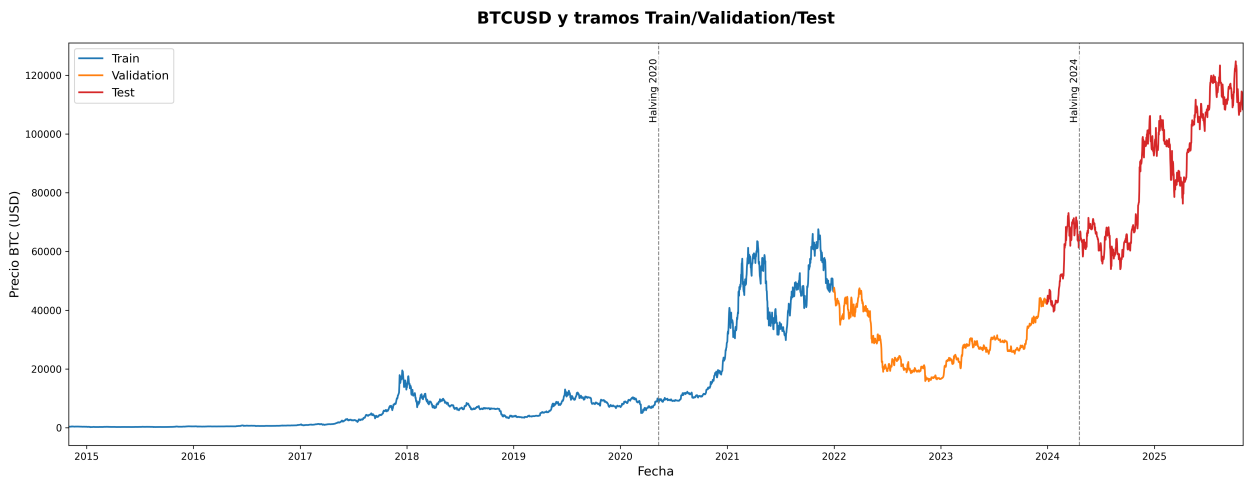


Figura 5.2: Tramos de entrenamiento, validación y testeo

Nota. Elaboración propia.

Tabla 5.2: Partición temporal del conjunto de datos

Tramo	Período	Días	Porcentaje
Train	01/12/2014 → 31/12/2021	2.588	64.8 %
Validation	01/01/2022 → 31/12/2023	730	18.3 %
Test	01/01/2024 → 31/10/2025	670	17.0 %
Total	01/12/2014 → 31/10/2025	3.988	100 %

Nota. Elaboración propia.

5.4. Configuración Modelos

Cada modelo fue entrenado sobre el conjunto de entrenamiento descrito en la sección anterior y ajustado utilizando el conjunto de validación. La evaluación final se realizó exclusivamente sobre el tramo de testeo, evitando cualquier forma de sesgo de anticipación. El enfoque principal de esta investigación es el modelo Informer, mientras que las demás arquitecturas se utilizan posteriormente como referencia comparativa para medir el poder predictivo de cada modelo en el mercado del Bitcoin.

5.4.1. Horizontes de predicción

Se utilizaron seis horizontes de predicción: 1, 7, 14, 30, 60 y 90 días. Estos intervalos permiten evaluar el desempeño predictivo en distintas escalas temporales, desde el muy corto plazo hasta proyecciones de mediano plazo, donde cada uno de ellas representa un escenario predictivo distinto.

El horizonte de 1 día corresponde a predicciones de muy corto plazo, donde la información reciente del mercado suele tener un rol más relevante. Este horizonte permite analizar si existen dependencias temporales inmediatas en los retornos. En caso de que existan, podrían ser aprovechadas por modelos basados en series históricas.

Los horizontes de 7 y 14 días permiten extender el análisis hacia variaciones de corto plazo que pueden estar asociadas a ciclos semanales, fluctuaciones de liquidez o episodios breves de volatilidad. Estos plazos requieren que el modelo integre información que trascienda la dinámica diaria, evaluando si existen patrones que se mantengan durante períodos ligeramente más largos.

Por su parte, los horizontes de 30, 60 y 90 días permiten evaluar escenarios de mediano plazo, en los cuales intervienen elementos adicionales como cambios de tendencia, condiciones macroeconómicas y métricas relacionadas con el funcionamiento de la red Bitcoin.

El uso de múltiples horizontes permite caracterizar cómo varía la capacidad predictiva del modelo a través del tiempo. Además, ofrece un marco para analizar si la información histórica contiene elementos potencialmente útiles para anticipar retornos futuros. Desde la perspectiva de la eficiencia débil del mercado, un activo cuyos precios reflejan plenamente la información pasada no debería ofrecer oportunidades sistemáticas de predicción en ninguno de estos horizontes. Por ello, la comparación de desempeño del modelo entre distintos plazos constituye una base metodológica que permitirá discutir, en capítulos posteriores, si el comportamiento del mercado de Bitcoin se alinea o no con esta hipótesis.

5.4.2. Hiperparámetros

Tabla 5.3: Hiperparámetros fijos utilizados en el modelo Informer

Hiperparámetro	Valor
Secuencia de entrada	160
Longitud etiqueta	80
Número de capas codificador	2
Número de capas decodificador	1
Dimensión del modelo	512
Dimensión capa feed-forward	2048
Número de cabezas de atención	8
Dropout	0.1
Epochs	30
Optimización	Adam
Tasa de aprendizaje inicial	0.0001
Batch size	32
Función de pérdida	Huber ($\delta = 0.005$)

Nota: Los hiperparámetros se mantuvieron constantes en todos los horizontes de predicción.

5.4.3. Función de pérdida

La función de pérdida determina cómo se mide la diferencia entre las predicciones del modelo y los valores reales observados. Durante el entrenamiento, el modelo ajusta sus parámetros con el objetivo de minimizar esta medida de error, por lo que la elección de la función de pérdida influye directamente en la estabilidad y calidad del proceso de aprendizaje.

En esta investigación se utilizó la función de pérdida Huber. Esta función combina elementos de MSE (Error Cuadrático Medio) y MAE (Error Absoluto Medio), es decir, actúa de manera cuadrática para errores pequeños y de manera lineal para errores grandes según el valor del parámetro δ . Esta propiedad la hace particularmente adecuada para series financieras como Bitcoin, donde la presencia de valores atípicos y movimientos bruscos puede afectar negativamente a funciones más sensibles como MSE. La Huber permite un entrenamiento más robusto frente a outliers sin perder estabilidad en el cálculo de gradientes.

Matemáticamente, la función de pérdida Huber para una observación i es:

$$L_{\delta}(y_i, \hat{y}_i) = \begin{cases} \frac{1}{2}(y_i - \hat{y}_i)^2, & \text{si } |y_i - \hat{y}_i| \leq \delta, \\ \delta |y_i - \hat{y}_i| - \frac{1}{2}\delta^2, & \text{si } |y_i - \hat{y}_i| > \delta. \end{cases}, \quad (42)$$

Para obtener la pérdida total utilizada en el entrenamiento, se promedia esta expresión sobre todas las observaciones del conjunto:

$$\text{Huber} = \frac{1}{N} \sum_{i=1}^N L_{\delta}(y_i, \hat{y}_i), \quad (43)$$

En todos los experimentos se fijó $\delta = 0.005$, garantizando consistencia en la comparación entre horizontes y reduciendo la influencia de valores extremos sin comprometer la estabilidad numérica del entrenamiento.

5.4.4. Métricas de evaluación

Para evaluar el rendimiento predictivo de los modelos se utilizaron tres métricas que evalúan distintos aspectos: MAE, MSE y MDA. Estas métricas permiten analizar la precisión numérica y la capacidad direccional del modelo en distintos horizontes de predicción.

Error Absoluto Medio (MAE). El MAE mide la magnitud promedio de los errores absolutos:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|, \quad (44)$$

Error Cuadrático Medio (MSE). El MSE penaliza de forma más intensa los errores grandes debido a su naturaleza cuadrática:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2, \quad (45)$$

Mean Directional Accuracy (MDA). El MDA mide la proporción de casos en que el modelo acierta la dirección del movimiento del retorno:

$$\text{MDA} = \frac{1}{N} \sum_{i=1}^N \mathbf{1}[\text{sign}(y_i) = \text{sign}(\hat{y}_i)], \quad (46)$$

Donde $\text{sign}(y)$ es la función signo, la cual permite evaluar si el modelo acierta la dirección del retorno, independiente del valor de la magnitud. Está definida como:

$$\text{sign}(y) = \begin{cases} +1, & \text{si } y > 0, \\ 0, & \text{si } y = 0, \\ -1, & \text{si } y < 0, \end{cases}$$

Estas métricas fueron aplicadas de forma uniforme en todos los modelos y horizontes, permitiendo una comparación directa del desempeño predictivo.

5.4.5. Benchmarking

Para evaluar de manera objetiva el rendimiento del modelo Informer, se incorporó un conjunto de modelos comparativos ampliamente utilizados en la literatura de predicción de series temporales: LSTM, GRU y LSTNet.

- **LSTM (Long Short-Term Memory):** Arquitectura clásica en el análisis de series temporales, reconocida por su capacidad para retener información relevante durante largos intervalos mediante un sistema de compuertas. Su uso extendido en finanzas lo convierte en un punto de comparación natural y confiable.
- **GRU (Gated Recurrent Unit):** Corresponde a una variante más ligera del LSTM, con menor cantidad de parámetros y un entrenamiento más eficiente. Este modelo ofrece una perspectiva complementaria respecto de arquitecturas recurrentes con menor complejidad.
- **LSTNet:** Combina componentes convolucionales y recurrentes, permitiendo capturar patrones locales y dependencias de mayor plazo. Su diseño híbrido resulta útil como punto de comparación frente a arquitecturas puramente recurrentes o basadas en atención.

Comparar el Informer con estas tres arquitecturas permite evaluar si su estructura basada en mecanismos de atención aporta mejoras reales frente a modelos tradicionales de series temporales.

5.5. Implementación técnica

5.5.1. Entorno y Herramientas computacionales

El desarrollo de esta investigación se realizó en un entorno virtual aislado, ocupando como base WSL2 (Ubuntu), logrando asegurar estabilidad y reproducibilidad. Los scripts se programaron con el lenguaje Python debido a la capacidad de soportar paquetes avanzados fundamentales para el aprendizaje profundo. Además, se uso Visual Studio y JupyterLab para manipular la data y construir gráficos.

Se utilizaron principalmente las siguientes librerías:

- **PyTorch**: framework principal utilizado para construir, entrenar y evaluar los modelos de aprendizaje profundo: Informer, LSTM, GRU y LSTNet.
- **NumPy**: herramienta base para operaciones numéricas y manejo eficiente de arreglos multidimensionales.
- **pandas**: Empleada para la limpieza, transformación y manipulación del dataset, facilitando el trabajo con series temporales.
- **scikit-learn**: Utilizada para tareas de preprocesamiento, cálculos de métricas y funciones auxiliares.
- **Matplotlib**: Destinada a la generación de gráficos estáticos para visualizar tendencias y resultados.
- **Seaborn**: Empleada para visualizaciones estadísticas más estéticas y exploratorias.

El modelo Informer fue descargado desde el repositorio oficial de GitHub, creado por los autores del paper (Zhou et al., 2021). Luego, este fue adaptado para admitir las series temporales con los horizontes definidos en esta investigación y para cargar el dataset creado para esta investigación.

5.5.2. Hardware

El hardware utilizado en todas las ejecuciones contiene las siguientes características técnicas:

- **Procesador (CPU):** Intel Core i7-12650H de 12^a generación, compuesto por 10 núcleos y 16 hilos, lo que facilita el manejo de cargas de trabajo paralelas durante la ejecución de los experimentos.
- **Memoria RAM:** 16 GB DDR5 a 4800 MT/s.
- **Unidad de Procesamiento Gráfico (GPU):** NVIDIA GeForce RTX 3060 con 6 GB de memoria dedicada, compatible con CUDA, lo que permitió acelerar el entrenamiento en PyTorch mediante cómputo en GPU.
- **Sistema operativo anfitrión:** Windows 11 utilizando WSL2 (Ubuntu).

6. Resultados

El análisis de los resultados se estructura separando la evaluación de la capacidad predictiva de los modelos, medida a través del error de predicción, del análisis de eficiencia débil del mercado, evaluado mediante la capacidad direccional de los retornos pronosticados. Los resultados corresponden al promedio de tres iteraciones independientes por cada configuración, con el fin de mitigar la variabilidad estocástica inherente al entrenamiento de modelos de aprendizaje profundo.

El modelo principal es el Informer debido a que posee la arquitectura de interés de la investigación. Los modelos LSTM, GRU y LSTNet se reportan como benchmarks, permitiendo contextualizar el desempeño del Informer frente a alternativas neuronales recurrentes (LSTM/GRU) y una arquitectura híbrida (LSTNet). Las métricas utilizadas son MAE y MSE para cuantificar magnitud de error y MDA para evaluar la consistencia de los signos en los retornos, conectando directamente con la noción de predictibilidad direccional asociada a eficiencia débil.

6.1. Resultados para horizonte de predicción de 1 día

Tabla 6.1: Resultados promedio para horizonte de predicción de 1 día.

Modelo	MAE	MSE	MDA	Nº pasos
LSTM	0.018266	0.000653	0.509896	640
GRU	0.018811	0.000673	0.502604	640
Informer	0.021335	0.000817	0.503125	640
LSTNet	0.024205	0.001011	0.495312	640

Nota. Cada resultado es el promedio de tres iteraciones. Elaboración propia.

6.2. Resultados para horizontes de predicción de 7 a 90 días

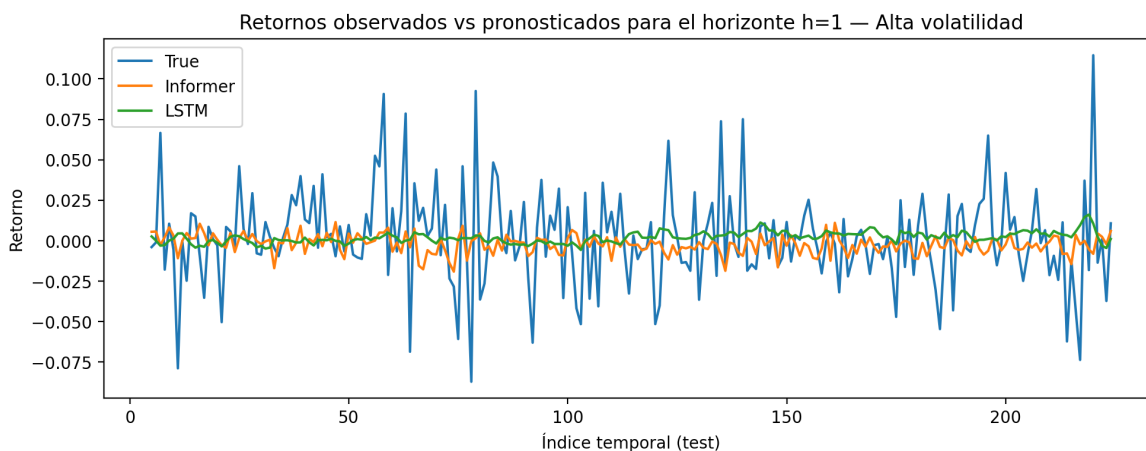
Tabla 6.2: Resultados promedio para horizontes de predicción de 7 a 90 días.

Modelo	Horizonte	MAE	MSE	MDA	N° pasos
LSTM	7	0.018430	0.000656	0.499330	4480
Informer	7	0.019173	0.000701	0.506027	4480
GRU	7	0.018983	0.000681	0.495089	4480
LSTNet	7	0.029938	0.001460	0.498512	4480
LSTM	14	0.018479	0.000657	0.498735	8960
GRU	14	0.019154	0.000691	0.506250	8960
Informer	14	0.019072	0.000678	0.517188	8960
LSTNet	14	0.031456	0.001646	0.501525	8960
LSTM	30	0.018571	0.000656	0.506771	19200
Informer	30	0.018496	0.000651	0.514340	19200
GRU	30	0.019038	0.000678	0.495833	19200
LSTNet	30	0.034954	0.002009	0.497882	19200
LSTM	60	0.018628	0.000657	0.509320	36480
Informer	60	0.018654	0.000649	0.515351	36480
GRU	60	0.020036	0.000736	0.501590	36480
LSTNet	60	0.035065	0.002032	0.499580	36480
LSTM	90	0.018897	0.000674	0.503054	51840
Informer	90	0.019278	0.000669	0.513201	51840
GRU	90	0.019010	0.000702	0.496528	51840
LSTNet	90	0.036787	0.002243	0.500868	51840

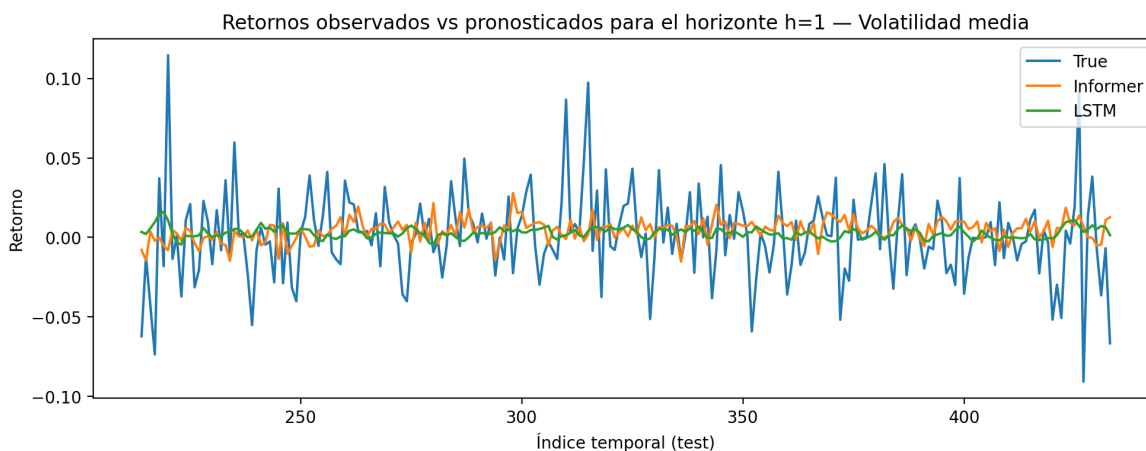
Nota. Cada resultado es el promedio de tres iteraciones. Elaboración propia.

6.3. Análisis de resultados para horizontes de predicción de 1 día

6.3.1. Capacidad predictiva



(a) Alta volatilidad.



(b) Volatilidad media.

Figura 6.1: Retornos observados vs. pronosticados para el horizonte $h = 1$ día en distintos tramos de volatilidad. Comparación Informer-LSTM

Nota. Elaboración propia.

En el horizonte de predicción de un día $h = 1$, el modelo LSTM exhibe el mejor desempeño en términos de métricas de error, superando consistentemente al modelo Informer. En particular, el Informer registra un error absoluto medio aproximadamente un 16,8 % mayor y un error cuadrático medio 25,1 % superior respecto de LSTM en este horizonte, lo que indica una menor capacidad para

ajustarse a la dinámica de los retornos diarios.

No obstante, la Figura 6.1 muestra un comportamiento relevante desde el punto de vista visual. En los distintos escenarios de volatilidad evaluados, las trayectorias pronosticadas por el modelo Informer muestran una mayor capacidad para seguir la dinámica de corto plazo de los retornos observados, reaccionando de forma más evidente ante fluctuaciones abruptas y cambios de signo. Por otra parte, el modelo LSTM presenta predicciones considerablemente más suavizadas, con una menor amplitud frente a variaciones locales de la serie.

Esta diferencia sugiere la existencia de un compromiso entre capacidad de seguimiento dinámico y estabilidad estadística. Mientras que el modelo Informer logra capturar con mayor fidelidad la forma temporal de la señal, dicha mayor variabilidad se traduce en errores de mayor magnitud bajo métricas cuadráticas. Por el contrario, el sesgo inductivo del modelo LSTM favorece estimaciones más conservadoras, cercanas al promedio, lo que resulta beneficioso en términos de MAE y MSE en un mercado dominado por ruido de alta frecuencia (Figura 6.2).

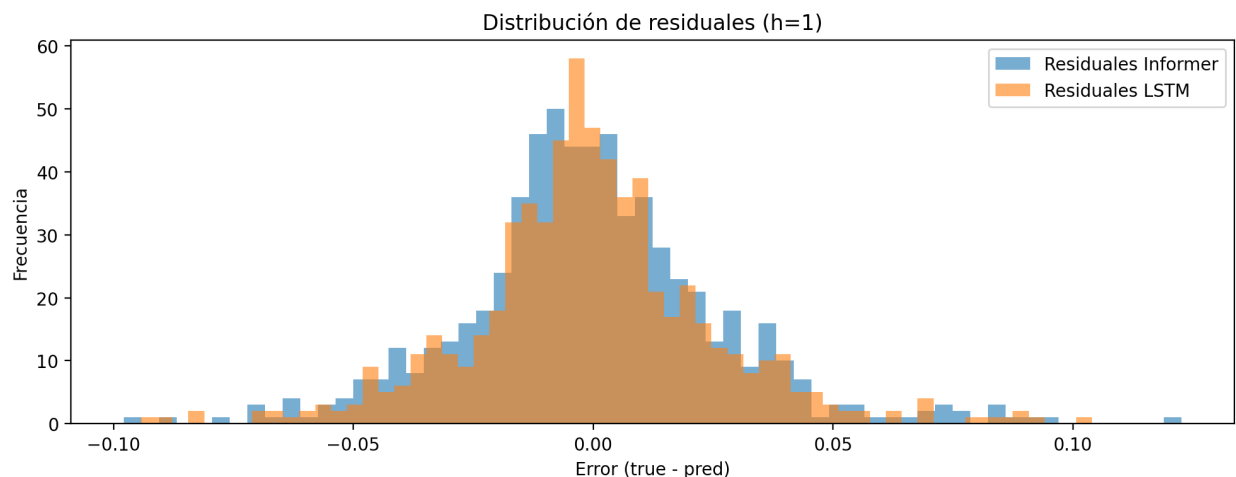


Figura 6.2: Distribución empírica de los errores de predicción (residuales) para el horizonte $h = 1$. Comparación entre modelos Informer y LSTM

Nota. Elaboración propia.

El modelo GRU alcanza resultados cercanos a los de el modelo LSTM, confirmando que arquitecturas recurrentes con compuertas presentan un equilibrio entre flexibilidad y regularización

para horizontes muy cortos. En contraste, LSTNet exhibe el peor desempeño, lo que sugiere limitaciones para adaptarse a la dinámica del retorno diario bajo la configuración empleada.

6.3.2. Capacidad direccional

En cuanto a la capacidad direccional, el modelo LSTM obtiene la mayor cantidad de aciertos en el signo del retorno con un 50,98 %, seguido por Informer con 50,31 % y GRU con 50,26 %. LSTNet se ubica bajo el umbral neutral con 49,43 %. Si bien valores sobre 0,5 sugieren acierto direccional por encima del azar, las magnitudes observadas en $h = 1$ son moderadas y cercanas al 50 %, lo que es coherente con un escenario donde el retorno diario contiene un componente elevado de impredecibilidad.

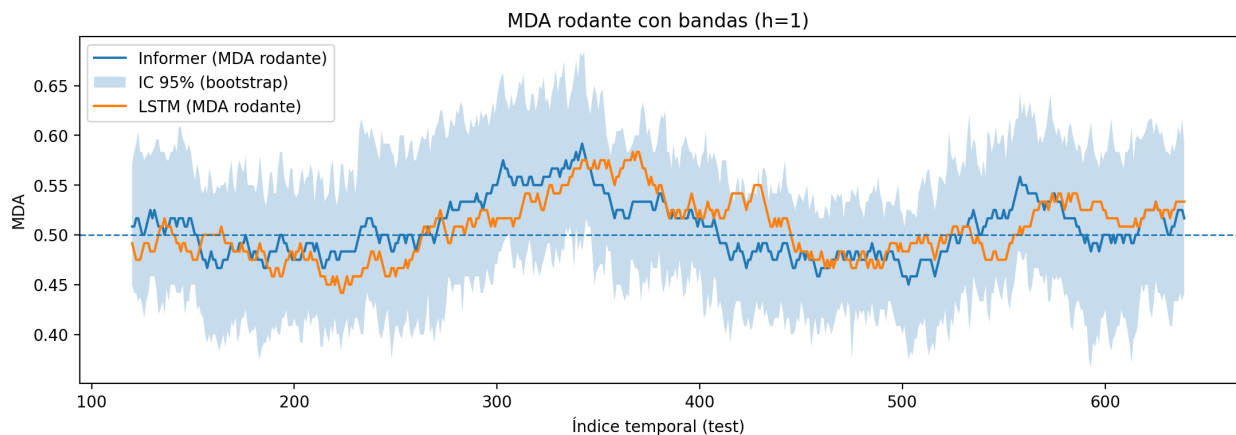


Figura 6.3: Evolución temporal del acierto direccional (MDA rodante) para $h = 1$ día. Se reporta MDA con ventana móvil y bandas de confianza por remuestreo. Comparación entre Informer y LSTM

Nota. Elaboración propia.

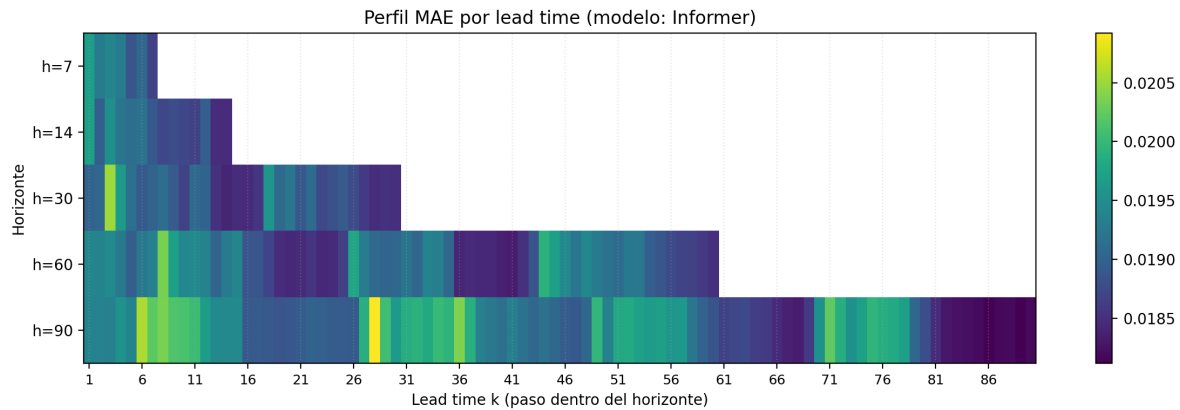
6.4. Análisis de resultados para horizontes de predicción de 7 a 90 días

6.4.1. Desempeño en MAE y MSE

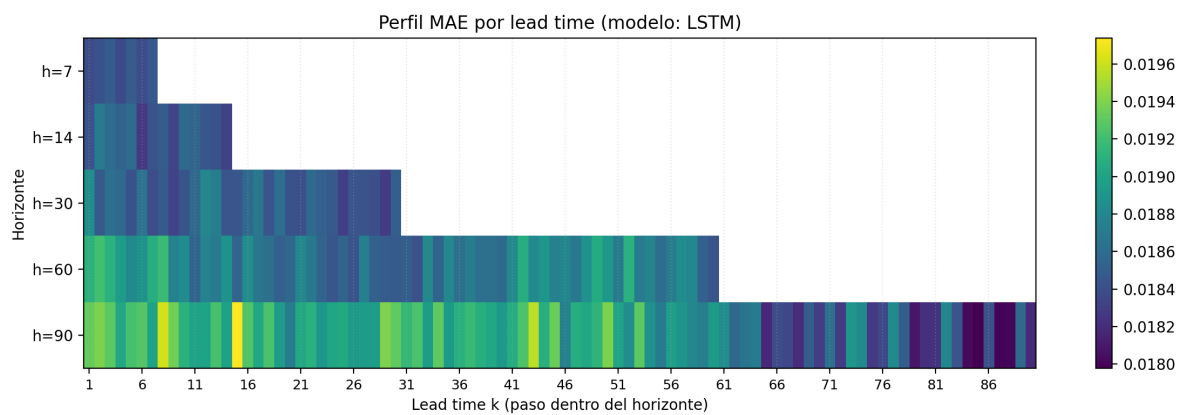
A nivel de MAE, se observa en la Tabla 6.2 que los valores para los modelos LSTM, GRU e Informer se mantienen en un rango relativamente estrecho (aprox. 0,018–0,020) incluso al aumentar h . Esta “estabilidad” debe interpretarse con cuidado dado que el conteo de observaciones (N_{pasos}) crece proporcionalmente al horizonte (por ejemplo, 4480 para $h = 7$, 8960 para $h = 14$, etc.), la métrica se está promediando sobre todos los pasos del vector de predicción, lo cual mezcla errores de pasos cercanos y lejanos.

En términos comparativos, el modelo Informer muestra un desempeño competitivo en error para horizontes medios. En $h = 30$ logra el mejor resultado en error entre los modelos evaluados (MAE = 0,018496, MSE = 0,000651), superando marginalmente a LSTM (MAE = 0,018571, MSE = 0,000656). Para $h = 60$ y $h = 90$, el Informer obtiene los mejores MSE (0,000649 y 0,000669, respectivamente), mientras que LSTM conserva el mejor MAE en esos horizontes (0,018628 y 0,018897) (Figura 6.4). Este patrón sugiere que el Informer tiende a reducir errores grandes (penalizados por MSE) en horizontes largos, aunque no necesariamente minimiza el error absoluto medio en todos los casos. Esto puede variar según cómo se ajuste delta en la función de pérdida Huber. En este trabajo se probaron diversos deltas, sin embargo, en promedio el escogido fue el que presentó errores más bajos.

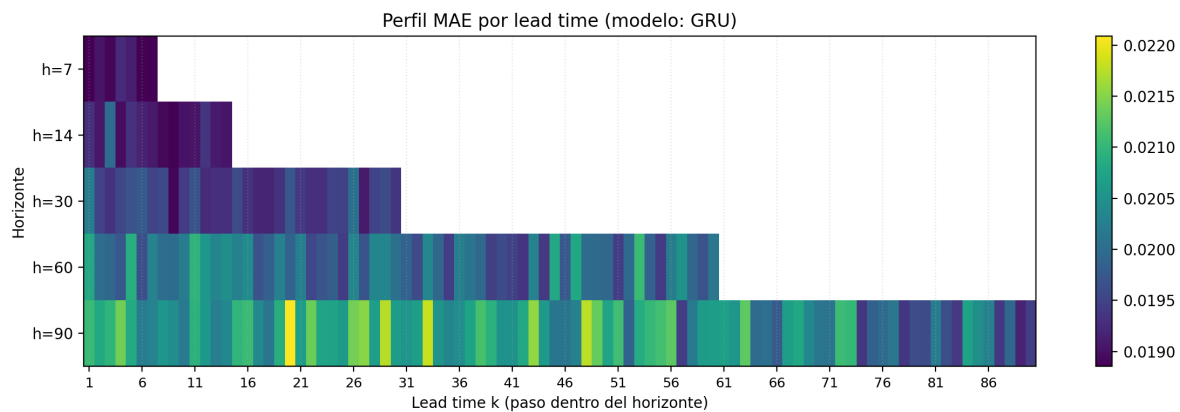
El modelo LSTNet se separa claramente del resto con MAE y MSE sustancialmente mayores en todos los horizontes (por ejemplo, MAE = 0,036787 y MSE = 0,002243 en $h = 90$), lo que indica que bajo la configuración utilizada su aproximación no resulta adecuada para el objetivo de pronóstico del retorno diario en este contexto de altísima volatilidad.



(a) Modelo Informer.



(b) Modelo LSTM.



(c) Modelo GRU.

Figura 6.4: Perfil de MAE por paso de predicción (*lead time k*) en pronóstico multi-step. Comparación entre modelos.

Nota. Elaboración propia.

6.4.2. Capacidad direccional

En contraste con el patrón de MAE/MSE, la métrica direccional muestra un comportamiento particularmente relevante para el objetivo de eficiencia débil. Para horizontes iguales o mayores a 7 días, el Informer obtiene sistemáticamente el mayor MDA entre los modelos comparados (Figura 6.5).

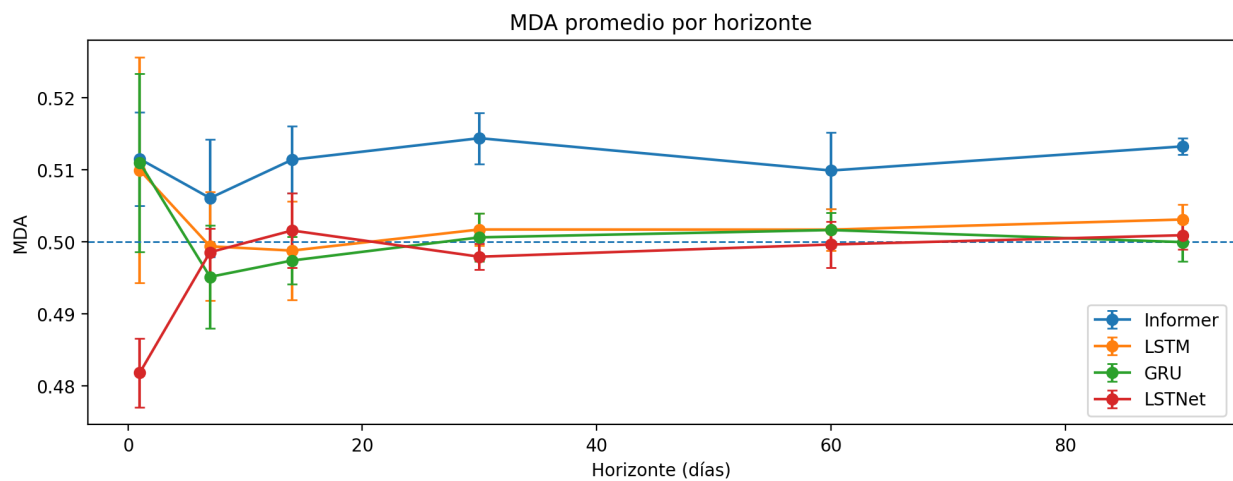


Figura 6.5: Curva MDA vs horizonte para todos los modelos, con barras de error (variabilidad entre las 3 iteraciones)

Nota. Elaboración propia.

Esta evidencia es consistente con la hipótesis de que el Informer puede capturar señales de mediano plazo que ayudan a anticipar la dirección del retorno más allá del ruido de corto plazo.

6.5. Síntesis comparativa global: ranking y trade-off para la discusión

6.5.1. Ranking por métrica y horizonte

Al sintetizar el ranking por métricas, se observa un patrón (Tabla 6.3):

1. Para MAE/MSE, el modelo LSTM domina en el corto plazo y mantiene competitividad en la mayoría de horizontes, mientras que el Informer se vuelve particularmente competitivo en horizontes medios (especialmente $h = 30$) y reduce MSE en horizontes largos ($h = 60$, $h = 90$).
2. Para MDA, el Informer es el mejor modelo en todos los horizontes desde $h = 7$ hasta $h = 90$, lo que lo posiciona como el candidato principal cuando el objetivo es capturar predictibilidad direccional.

Horizonte	MAE	MSE	MDA
1	LSTM	LSTM	Informer
7	LSTM	LSTM	Informer
14	LSTM	LSTM	Informer
30	Informer	Informer	Informer
60	LSTM	Informer	Informer
90	LSTM	Informer	Informer

Tabla 6.3: Modelo ganador por horizonte de predicción y métrica. El modelo ganador corresponde al mejor desempeño promedio según cada métrica, considerando la minimización de MAE y MSE y la maximización de MDA.

Nota. Elaboración propia.

Este resultado produce un trade-off interesante para la discusión. Un modelo puede no ser el mejor en minimizar error promedio, pero sí ser superior en capturar señales de dirección.

6.6. Implicancias para eficiencia débil

En un mercado débilmente eficiente, el retorno (y especialmente su signo) debería ser difícil de anticipar de forma sistemática a partir de información pública pasada. Por ello, un MDA persistentemente superior a 0.5, estable en el tiempo y robusto estadísticamente, constituye evidencia empírica contra la eficiencia débil (o, al menos, evidencia de predictibilidad explotable bajo ciertas condiciones).

A partir de los resultados obtenidos, la evidencia en favor de predictibilidad direccional es en el mejor de los casos moderada. Para $h = 1$ día, los valores de MDA se ubican cercanos al umbral neutral (50 %), lo que es coherente con un entorno dominado por ruido de alta frecuencia y con retornos diarios altamente impredecibles. Para horizontes mayores ($h \geq 7$), se observa que Informer tiende a presentar MDA promedio superior al de los benchmarks (Figura 6.5), sugiriendo que la arquitectura puede capturar señales de mediano plazo que contribuyen a anticipar la dirección del retorno en ciertos tramos del conjunto de prueba.

Sin embargo, existen tres consideraciones que impiden interpretar estos resultados como una refutación concluyente de la eficiencia débil. Primero, la magnitud del efecto es pequeña, incrementos del orden de 0,01–0,02 en MDA, aunque pueden ser consistentes con cierta señal, también pueden ser sensibles a la variabilidad estocástica del entrenamiento y del período en particular evaluado. Segundo, la evaluación multi-step introduce dependencia entre predicciones dentro de un mismo bloque de horizonte, lo que reduce la independencia efectiva de las observaciones y puede inflar la aparente precisión direccional si se interpreta el conteo de aciertos como si fuesen ensayos independientes. Tercero, los diagnósticos visuales muestran un trade-off entre capacidad de seguimiento dinámico y estabilidad. Informer tiende a reaccionar más a fluctuaciones locales, mientras que LSTM presenta trayectorias más suavizadas, lo que se refleja en diferencias entre métricas de error (MAE/MSE) y métricas direccionales (MDA).

En síntesis, los resultados son compatibles con la existencia de predictibilidad direccional

de manera marginal y en horizontes mayores, sobre todo en el caso del modelo Informer, pero dicha evidencia debe interpretarse con cautela debido al tamaño del efecto y a las dependencias inherentes al pronóstico multi-step.

7. Conclusiones

El objetivo de esta investigación fue evaluar la capacidad predictiva de modelos de deep learning para predecir los retornos de Bitcoin, comparando el modelo Informer con LSTM, GRU y LSTNet. Se utilizaron métricas de error (MAE y MSE) y de capacidad direccional (MDA) para evaluar el desempeño de los modelos en distintos horizontes de predicción de 1, 7, 14, 30, 60 y 90 días. Los resultados muestran que Informer sobresale en términos direccionales para horizontes medianos y largos, mientras que LSTM muestra ventajas en el corto plazo y una cierta estabilidad en general.

Para el horizonte de 1 día, el modelo LSTM mostró un mejor desempeño en MAE y MSE, sugiriendo que el retorno diario contiene componentes de ruido que LSTM maneja mejor. Sin embargo, Informer mostró mejoras en MSE para horizontes más largos (30, 60, 90 días), destacando en la reducción de errores grandes, lo que es coherente con su capacidad de capturar dependencias de mediano plazo, y con la configuración de la función de pérdida Huber, que entrega un peso respectivo a MAE y MSE según el delta que se le asigne a Huber.

En términos de predictibilidad direccional, Informer fue el modelo con mejor desempeño en horizontes mayores a 7 días. Aunque la magnitud del efecto fue moderada, esto sugiere que Informer tiene una leve ventaja en la capacidad de predecir la dirección de los retornos en comparación con los modelos recurrentes, como LSTM y GRU. Sin embargo, este resultado no es suficiente para rechazar de forma concluyente la eficiencia débil del mercado.

Este trabajo aporta una evaluación comparativa de modelos de deep learning para la predicción de Bitcoin, destacando el Informer como el modelo con mayor capacidad direccional para horizontes largos. Sin embargo, la evidencia sobre eficiencia débil es moderada y sujeta a varias limitaciones como se describió en los resultados, debido a que, la magnitud del efecto en MDA es

pequeña, no se realizaron pruebas formales de significancia para la predictibilidad direccional, y el análisis depende de la configuración particular del modelo y el período evaluado.

La comparación entre Informer y LSTM mostró que Informer tiende a reaccionar mejor ante fluctuaciones locales en los retornos, mientras que LSTM presenta predicciones más suavizadas. Este comportamiento refleja un trade-off entre capacidad de seguimiento dinámico y estabilidad. Mientras que LSTM minimiza mejor los errores promedio (MAE/MSE) en el corto plazo, Informer captura mejor la dinámica de los retornos, lo que le permite superar a LSTM en términos direccionales para horizontes largos.

Para futuras investigaciones, se recomienda realizar pruebas de significancia sobre MDA utilizando esquemas robustos que controlen dependencia temporal, además de evaluar el desempeño bajo validación multi-step. También sería valioso realizar un análisis de la eficiencia débil en distintos regímenes de mercado (alta y baja volatilidad) y contrastar la predictibilidad direccional con estrategias de trading basadas en el signo predicho.

En resumen, los resultados muestran que la predicción de retornos de Bitcoin en el corto plazo es difícil debido al alto ruido de alta frecuencia. Sin embargo, Informer muestra una leve ventaja direccional en horizontes más largos, lo que sugiere que es posible identificar señales de mediano plazo en el mercado de criptomonedas. A pesar de no rechazar la hipótesis de eficiencia débil, los resultados documentan una predictibilidad marginalmente explotable que debe ser analizada con mayor profundidad bajo esquemas de validación más robustos.

A. Anexos

A.1. Gráficos complementarios



Figura A.1: Volumen de negociación

Nota. Elaboración propia en base a datos de Yahoo Finance.

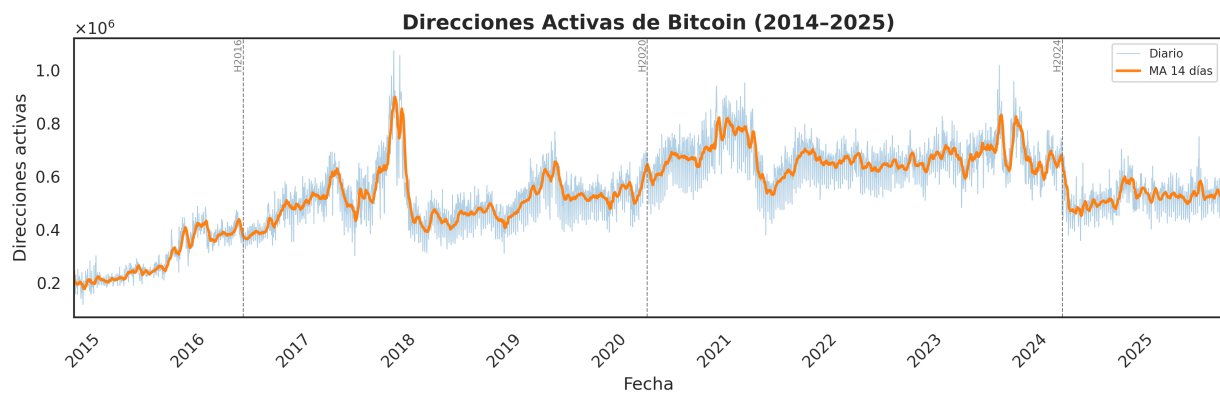


Figura A.2: Direcciones activas de Bitcoin

Nota. Elaboración propia en base a datos de Blockchain Web.

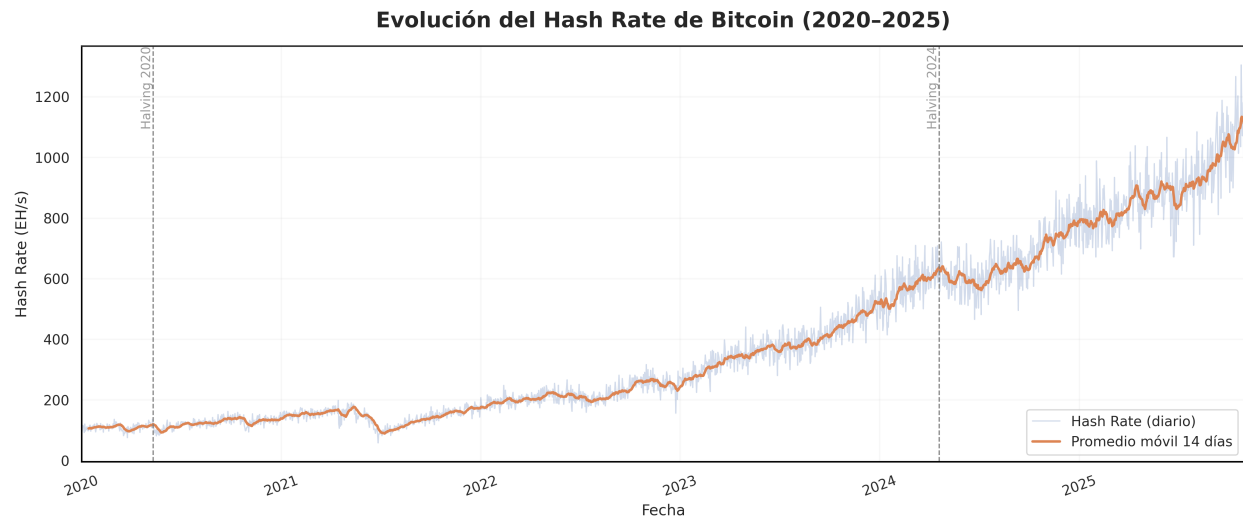


Figura A.3: Evolución del hash rate de Bitcoin

Nota. Elaboración propia en base a datos de Blockchain Web.

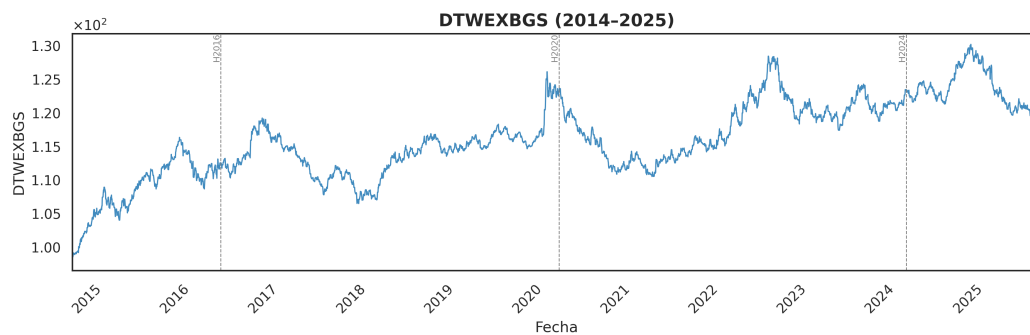


Figura A.4: Evolución del índice del dólar estadounidense

Nota. Elaboración propia en base a datos de FRED.

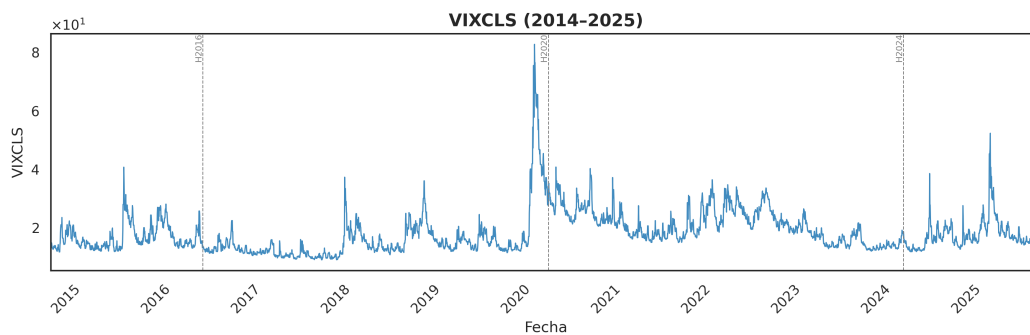


Figura A.5: Evolución del índice de volatilidad implícita (VIX)

Nota. Elaboración propia en base a datos de FRED.

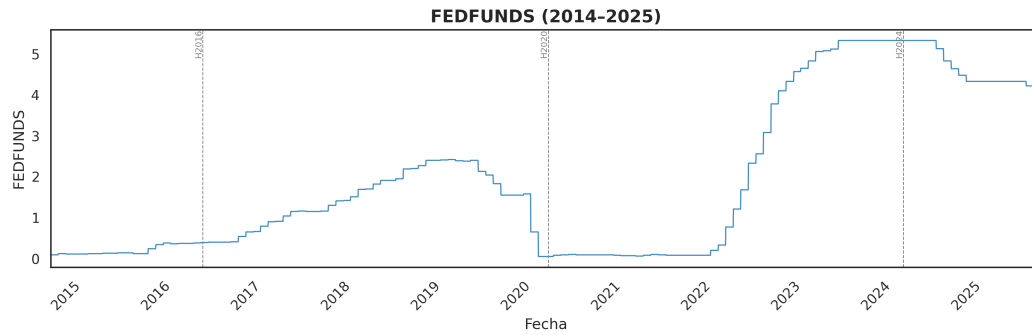


Figura A.6: Evolución de la tasa de interés de USA

Nota. Elaboración propia en base a datos de FRED.

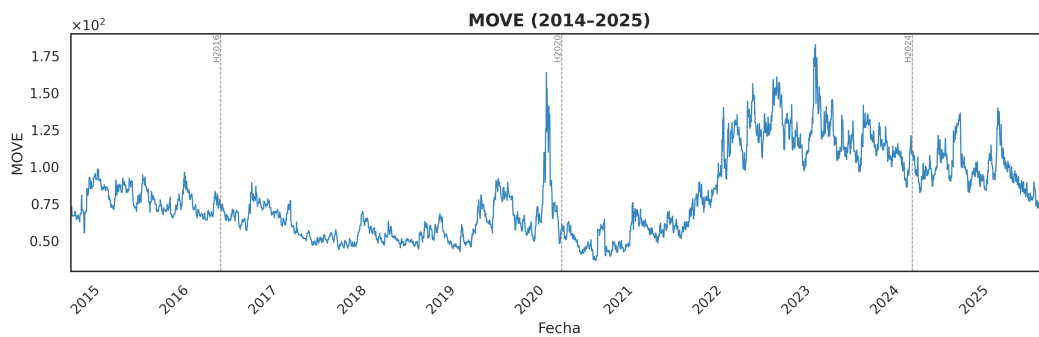


Figura A.7: Evolución del la volatilidad del mercado de bonos de USA

Nota. Elaboración propia en base a datos de FRED.

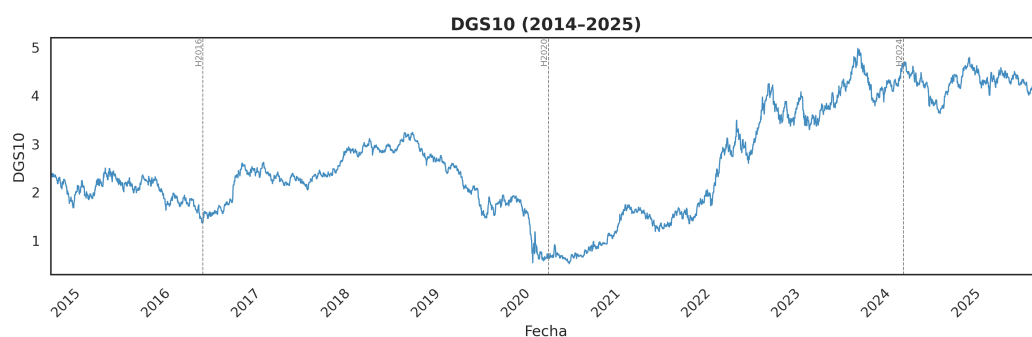


Figura A.8: Evolución del rendimiento del bono del tesoro a 10 años de USA

Nota. Elaboración propia en base a datos de FRED.

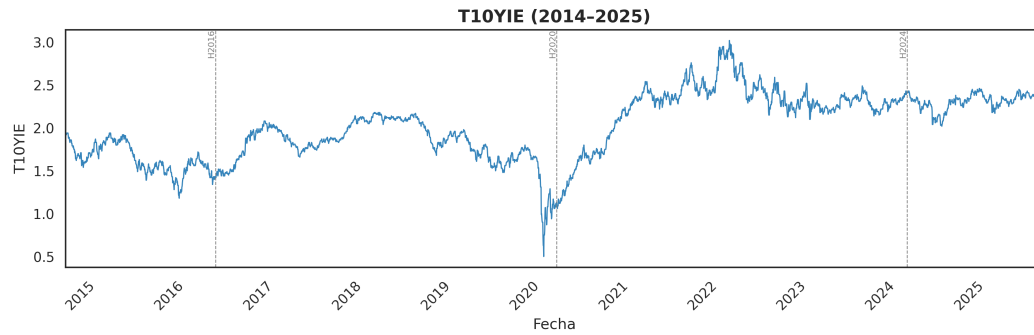


Figura A.9: Evolución de la inflación implícita a 10 años de USA

Nota. Elaboración propia en base a datos de FRED.

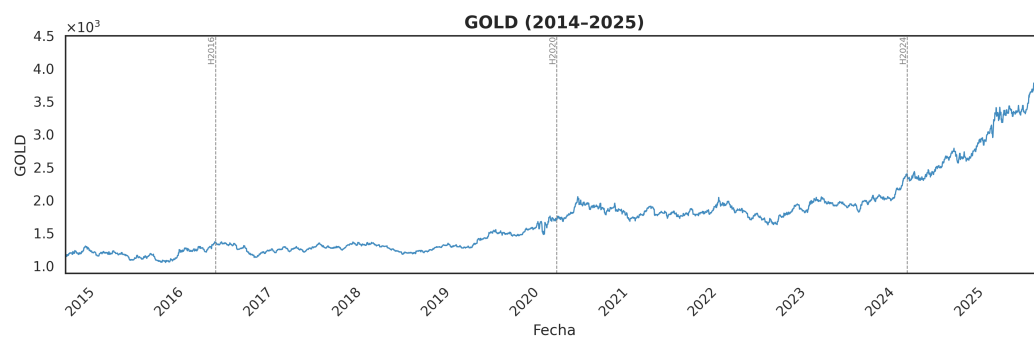


Figura A.10: Evolución del precio del oro

Nota. Elaboración propia en base a datos de Yahoo Finance.

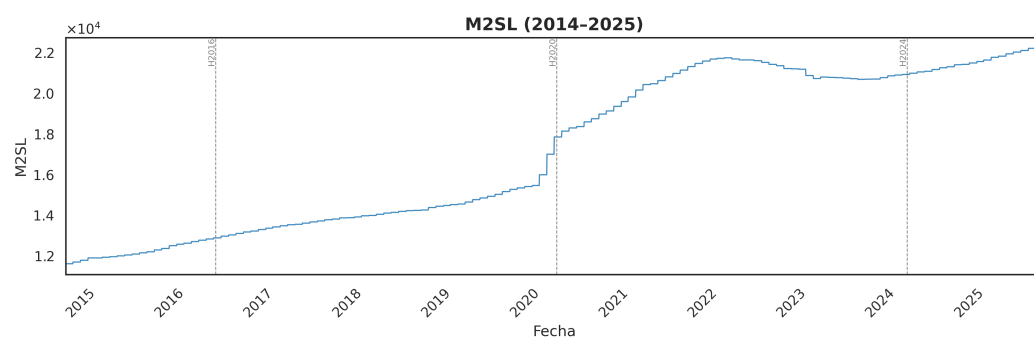


Figura A.11: Evolución de la oferta monetaria de USA

Nota. Elaboración propia en base a datos de FRED.

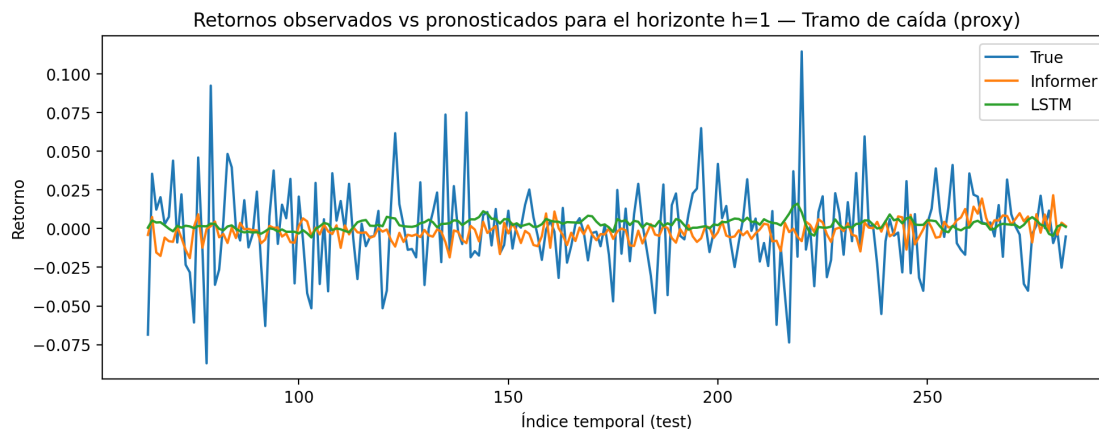


Figura A.12: Retornos observados vs pronosticados para el horizonte $h=1$ día. Tramo de caída (proxy). *Nota. Elaboración propia.*

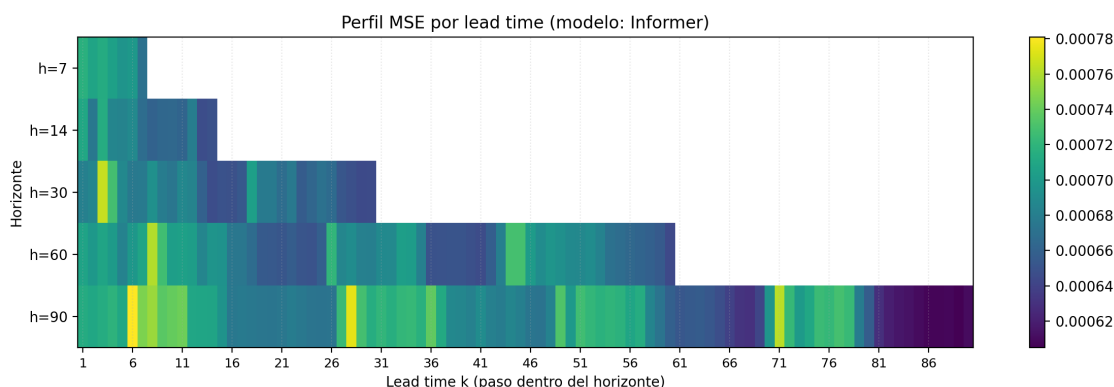


Figura A.13: Perfil de MSE por paso de predicción (*lead time k*) en pronóstico multi-step. Modelo Informer. *Nota. Elaboración propia.*

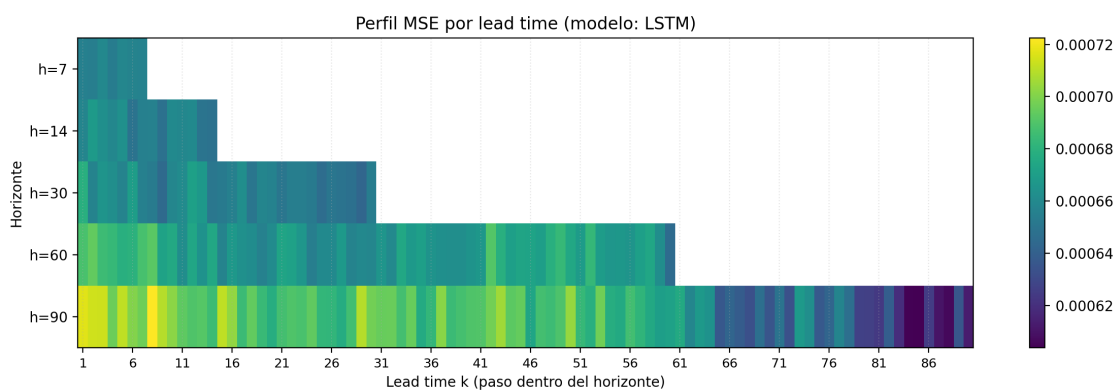


Figura A.14: Perfil de MSE por paso de predicción (*lead time k*) en pronóstico multi-step. Modelo LSTM. *Nota. Elaboración propia.*

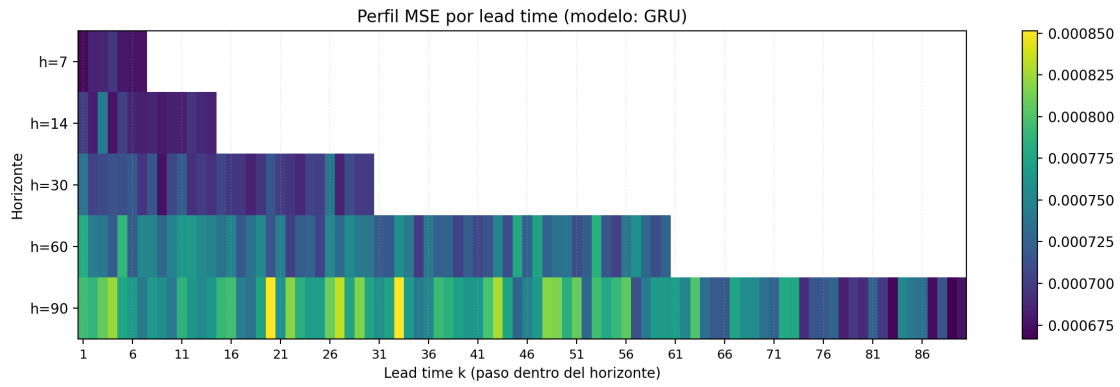


Figura A.15: Perfil de MSE por paso de predicción (*lead time k*) en pronóstico multi-step. Modelo GRU. *Nota. Elaboración propia.*

Referencias

- Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. *3rd International Conference on Learning Representations (ICLR)*.
<https://arxiv.org/abs/1409.0473>
- Bollerslev, T. (1986). Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics*, 31(3), 307-327. [https://doi.org/10.1016/0304-4076\(86\)90063-1](https://doi.org/10.1016/0304-4076(86)90063-1)
- Bollinger, J. (2002). *Bollinger on Bollinger Bands*. McGraw-Hill.
- Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2016). *Time Series Analysis: Forecasting and Control* (5th). John Wiley & Sons.
- Buterin, V. (2015). A Next-Generation Smart Contract and Decentralized Application Platform [White Paper].
- Chaum, D. (1989). Security without identification: Transaction systems to make Big Brother obsolete. *Communications of the ACM*, 28(10), 1030-1044.
- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv preprint arXiv:1406.1078*.
- Crosby, M., Pattanayak, P., Verma, S., & Kalyanaraman, V. (2016). Blockchain Technology: Beyond Bitcoin. *Applied Innovation Review*, (2), 6-10.
- Engle, R. F. (1982). Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*, 50(4), 987-1007. <https://doi.org/10.2307/1912773>

- Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance*, 25(2), 383-417. <https://doi.org/10.2307/2325486>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
<https://www.deeplearningbook.org/>
- Hamilton, J. D. (1994). *Time Series Analysis*. Princeton University Press.
- Hayes, A. (2019). Bitcoin Price and Its Marginal Cost of Production: Support for a Fundamental Value. *Journal of Risk and Financial Management*, 12(2), 1-12.
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735-1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Jiang, Y., Liu, Y., & Lu, X. (2022). The Response of Cryptocurrency Prices to U.S. Monetary Policy News. *Journal of International Money and Finance*, 120, 102-586.
- Karpoff, J. M. (1987). The relation between price changes and trading volume: A survey. *Journal of Financial and Quantitative Analysis*, 22(1).
- Kumar, A. (2017). Market volatility, monetary policy and the term premium. *Bank for International Settlements Working Paper*, (606).
- Liu, K., Manahov, V., & Stafylas, D. (2025). An investigation into the relationship between cryptocurrency active addresses and prices during major geopolitical conflicts. *Journal of Economic Studies*. <https://doi.org/10.1080/1351847X.2025.2585957>
- Lo, A. W. (2004). The Adaptive Markets Hypothesis: Market Efficiency from an Evolutionary Perspective. *Journal of Portfolio Management*, 30(5), 15-29.
<https://doi.org/10.3905/jpm.2004.442611>

- Long, S., Pei, H., Tian, H., & Lang, K. (2021). Can both Bitcoin and gold serve as safe-haven assets? A comparative analysis based on the NARDL model. *International Review of Financial Analysis*, 78, 101914. <https://doi.org/10.1016/j.irfa.2021.101914>
- Mienye, I. D., Swart, T. G., & Obaido, G. (2024). Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications. *Information*, 15(9), 517. <https://doi.org/10.3390/info15090517>
- Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System.
- Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S. (2016). *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press.
- Nian, L. P., & Chuen, D. L. K. (2015). Introduction to Bitcoin. *Handbook of Digital Currency*, 5-30.
- Parkinson, M. (1980). The extreme value method for estimating the variance of the rate of return. *Journal of Business*, 53(1), 61-65.
- Sapienza, P., & Zingales, L. (2012). A Trust Crisis. *International Review of Finance*, 12(2), 123-131. <https://doi.org/10.1111/j.1468-2443.2012.01150.x>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30. <https://papers.nips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>
- Whaley, R. E. (1993). Derivatives on Market Volatility: Hedging Tools Long Overdue. *The Journal of Derivatives*, 1(1), 71-84. <https://doi.org/10.3905/jod.1993.407866>
- Wilder, J. W. (1978). *New Concepts in Technical Trading Systems*. Trend Research.

Zhang, D., Aggarwal, C. C., & Qi, G. (2021). Deep learning for time series forecasting: Advances and challenges. *International Journal of Forecasting*, 37(1), 389-406.

<https://doi.org/10.1016/j.ijforecast.2020.06.008>

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 11106-11115.