Predicción de Volatilidad en Mercados Financieros: Un Enfoque Comparativo con Deep Learning LSTM vs Transformer

Memoria del Proyecto de la Asignatura: Extensiones de Machine Learning

Presentado por: Raúl Sánchez Ibáñez, Juan José Pérez Romero

Curso Académico 2025/2026

Resumen

La predicción precisa de la volatilidad en los mercados financieros constituye uno de los desafíos más complejos y críticos para la gestión de riesgos. Las series temporales bursátiles se caracterizan por su naturaleza estocástica, ruidosa y no estacionaria, lo que limita la eficacia de los modelos econométricos tradicionales. El presente trabajo aborda esta problemática mediante un enfoque de Ciencia de Datos avanzado, comparando dos arquitecturas de *Deep Learning* de vanguardia: las Redes de Memoria a Corto y Largo Plazo (LSTM) y los modelos Transformers basados en mecanismos de atención (*Self-Attention*). El objetivo central es predecir los cambios en la volatilidad (*Delta Volatility*) de las acciones de Apple Inc. (AAPL), utilizando un conjunto de datos histórico extenso (1980-2023).

La metodología empleada se fundamenta en un riguroso preprocesamiento de datos. Se generaron variables financieras derivadas, destacando el estimador de volatilidad de **Garman-Klass** —que incorpora información intradía para una mayor precisión— y la transformación de precios en retornos logarítmicos para asegurar la estacionariedad. Un aspecto crítico fue la implementación de una validación cruzada purgada (**Purged K-Fold**), técnica que respeta la cronología y elimina el sesgo de anticipación (*look-ahead bias*).

Los resultados experimentales demuestran la superioridad de la arquitectura Transformer sobre el enfoque recurrente clásico. A diferencia de las redes LSTM, el modelo basado en atención logró minimizar de manera más efectiva el error de regresión (obteniendo un mejor MSE y R^2), demostrando que la capacidad de ponderar globalmente la relevancia de eventos pasados (*Self-Attention*) es más eficiente que la memoria secuencial para capturar la dinámica no lineal del mercado.

Para validar la utilidad práctica, se realizó una simulación de inversión (*Backtesting*) con gestión dinámica de la exposición. En este entorno, la precisión estadística del Transformer se tradujo en una gestión de riesgos robusta, logrando un **Retorno Total** y un rendimiento ajustado al riesgo (**Sharpe Ratio**) superiores a los del modelo LSTM. Asimismo, el Transformer demostró una capacidad notable para proteger el capital, reduciendo significativamente la exposición antes de eventos de crisis ("cisnes negros"). Este estudio concluye que las arquitecturas basadas en atención representan el nuevo estado del arte para la predicción de volatilidad financiera.

1.- Introducción

1.1.- Contexto del problema

El análisis de series temporales financieras es uno de los desafíos más complejos y estudiados en el dominio del aprendizaje automático (*Machine Learning*) y la estadística computacional. Los mercados financieros generan volúmenes masivos de datos secuenciales que se caracterizan por propiedades estadísticas difíciles de modelar: son inherentemente ruidosos, dinámicos y, en su mayoría, no estacionarios. Esto significa que las propiedades estadísticas de los datos, como la media y la varianza, cambian con el tiempo, haciendo que los modelos predictivos tradicionales pierdan eficacia rápidamente.

En este contexto, la predicción de la **volatilidad** (la magnitud de los cambios en el precio de un activo) es crucial para la gestión de riesgos y la toma de decisiones de inversión. A diferencia de la predicción directa del precio (que a menudo sigue un camino aleatorio), la volatilidad presenta patrones de agrupamiento (*clustering*) y persistencia que pueden ser explotados mediante modelos computacionales avanzados.

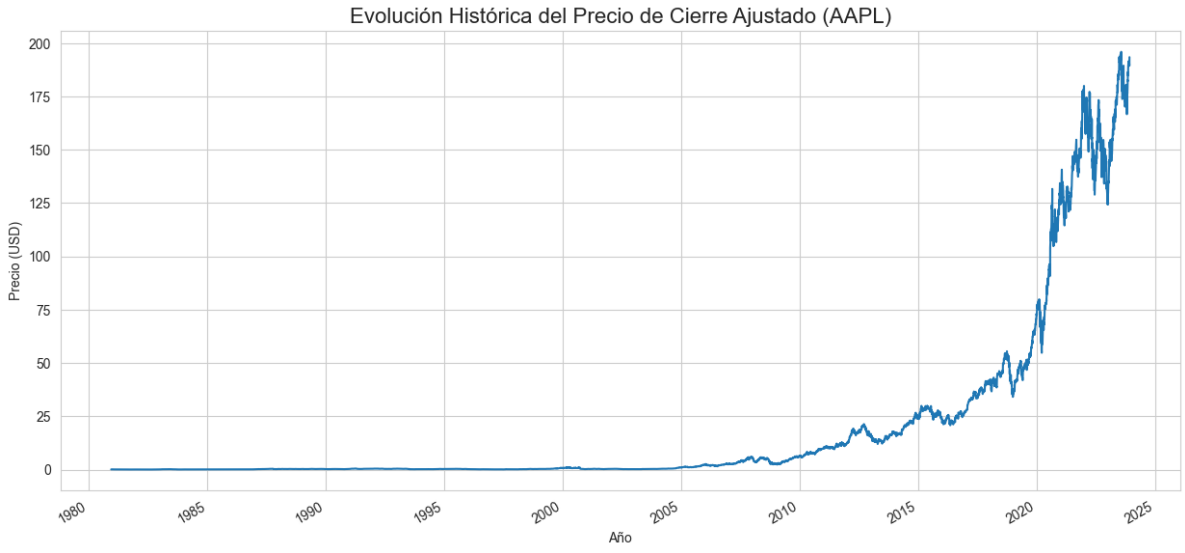

Figura 1: Evolución histórica del precio de cierre de Apple Inc. Se observa claramente la naturaleza no estacionaria de la serie, con tendencias cambiantes a lo largo del tiempo.

1.2.- Motivación y Objetivos

Tradicionalmente, la econometría financiera ha abordado la predicción de volatilidad mediante modelos autorregresivos como GARCH (*Generalized Autoregressive Conditional Heteroskedasticity*). Sin embargo, estos enfoques asumen relaciones lineales y estructuras rígidas que a menudo fracasan al intentar capturar la complejidad no lineal y los cambios de régimen abruptos de los mercados actuales.

El presente proyecto, realizado en el marco de la asignatura "Extensiones de Machine Learning", propone un cambio de paradigma hacia un enfoque basado en datos (*data-driven*). El **objetivo general** es desarrollar, implementar y validar un sistema de predicción basado en *Deep Learning* que supere las limitaciones de los enfoques clásicos, no solo en precisión estadística, sino en adaptabilidad y gestión de riesgos.

Para alcanzar este fin, se han definido los siguientes objetivos específicos:

1. **Fundamentación Metodológica:** Contrastar dos arquitecturas de vanguardia: el enfoque secuencial recurrente (**LSTM**, diseñado para dependencias a largo plazo) frente al enfoque paralelo basado en atención (**Transformers**, capaces de ponderar globalmente la historia temporal).
2. **Ingeniería de Datos Avanzada:** Implementar un *pipeline* robusto que aborde la no-estacionariedad de los precios y mejore la calidad de la señal mediante el uso de estimadores de volatilidad eficientes, específicamente el estimador de **Garman-Klass**, que incorpora información intradía (*High, Low, Open, Close*).

3. **Optimización Orientada al Riesgo:** Diseñar las funciones de pérdida y las estrategias de validación (*Grid Search* con *Purged K-Fold*) priorizando la penalización de los grandes errores (**MSE**). El objetivo no es solo ajustar la media, sino crear un sistema de "alerta temprana" que minimice los fallos catastróficos ante picos de volatilidad.

4. **Validación Económica:** Trascender la validación puramente métrica (R^2, MAE) para evaluar la utilidad real de los modelos mediante una simulación de *trading* (*Backtesting*). Se busca cuantificar si la "inteligencia" del modelo se traduce en un mejor retorno ajustado al riesgo (Sharpe Ratio) y una mayor protección del capital (menor *Drawdown*) durante periodos de crisis.

2.- Estado del Arte

2.1.- Limitaciones de los Modelos Clásicos

Las Redes Neuronales Artificiales (ANN) tradicionales, también conocidas como Perceptrones Multicapa (MLP) o redes *Feed-Forward*, han demostrado un éxito rotundo en tareas donde los datos de entrada son independientes entre sí. En estas redes, la información fluye en una única dirección, desde la entrada hasta la salida, sin conservar estado alguno.

Sin embargo, esta arquitectura presenta una limitación crítica para las series temporales financieras: **carecen de memoria**. Una red *Feed-Forward* procesa cada día de forma aislada. Para predecir la volatilidad de Apple para el día de mañana, no basta con procesar el precio de hoy; es crucial conocer la trayectoria de la última semana o mes. Aunque se puede intentar solucionar esto concatenando una ventana de precios pasados como entrada, este enfoque es rígido y no modela la naturaleza secuencial del tiempo.

2.2.- La "Memoria" del Mercado y las Redes Recurrentes (RNN)

La existencia de memoria en los mercados financieros es un hecho estilizado que valida la necesidad de modelos secuenciales. Como se desprende de nuestro análisis exploratorio, la volatilidad no es un paseo aleatorio puro; posee persistencia.

Esto se confirma mediante el análisis de autocorrelación (ACF/PACF), que muestra cómo la volatilidad de hoy guarda una correlación estadísticamente significativa con los valores observados en ventanas temporales anteriores.

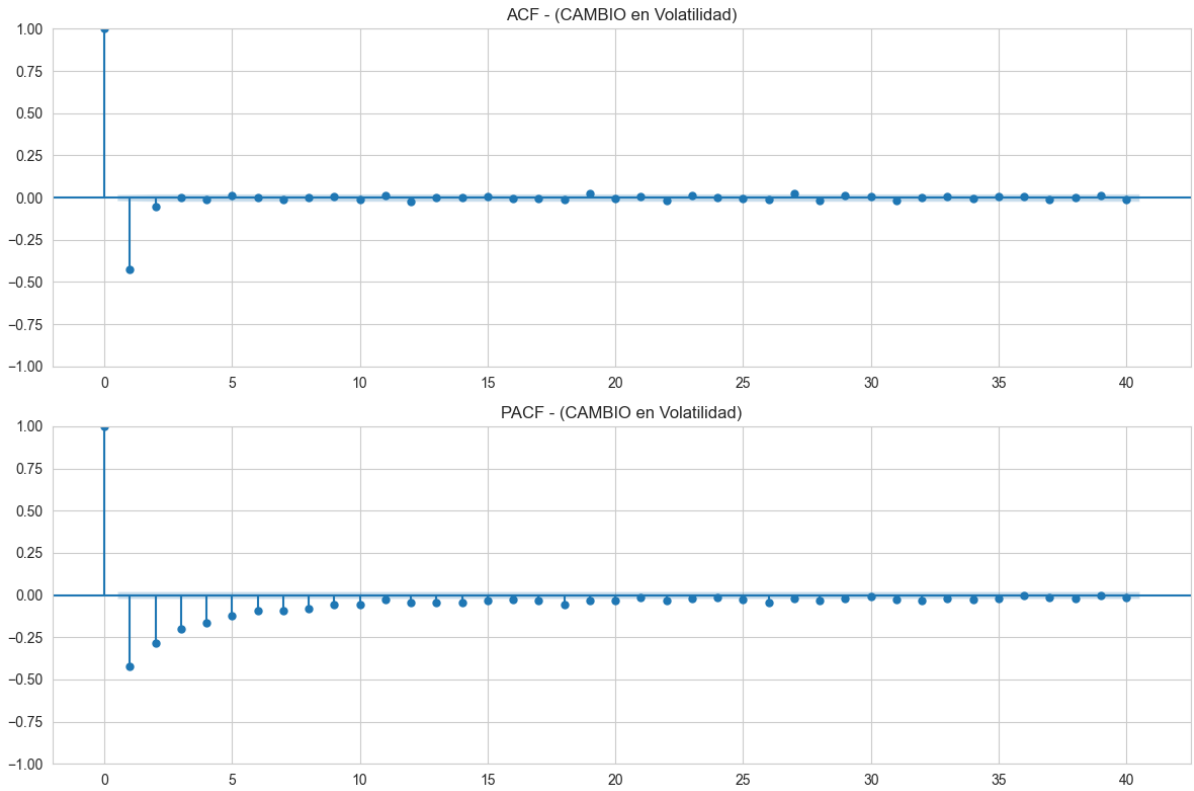

Figura 2: Análisis de autocorrelación de la volatilidad. Se observa cómo la influencia de los eventos pasados decae lentamente, justificando el uso de modelos con memoria.

Para modelar matemáticamente esta dependencia, surgen las **Redes Neuronales Recurrentes (RNN)**. Su innovación central es introducir un bucle de retroalimentación: la salida de una neurona en el instante no solo depende de la entrada actual, sino también del estado oculto (*hidden state*) calculado en $t-1$.

Matemáticamente, el estado oculto h_t se actualiza según:

$$h_t = \tanh(W_{hh} \cdot h_{t-1} + W_{xh} \cdot x_t + b_h)$$

Esta recurrencia permite a la red mantener una "memoria" a corto plazo. Sin embargo, las RNN simples sufren el problema del **desvanecimiento del gradiente** (*Vanishing Gradient Problem*). Al propagar el error hacia atrás a través de muchas capas temporales durante el entrenamiento, el gradiente tiende a volverse infinitesimal, impidiendo que la red aprenda correlaciones más allá de unos 10 pasos de tiempo, lo cual es insuficiente para capturar ciclos de mercado largos.

2.3.- Long Short-Term Memory (LSTM)

Introducidas por Hochreiter y Schmidhuber (1997), las redes **LSTM** son una evolución de las RNN diseñadas específicamente para resolver el problema del desvanecimiento del gradiente. La innovación clave es la introducción de una "Celda de Memoria" (*Cell State*) que actúa como una autopista de información donde el gradiente puede fluir sin ser apenas modificado.

El flujo de información en una celda LSTM está regulado por tres compuertas (*gates*) que utilizan funciones sigmoides:

1. **Forget Gate (Compuerta de Olvido):** Decide qué información antigua es irrelevante y debe ser borrada (por ejemplo, cuando el mercado cambia de tendencia alcista a bajista).
2. **Input Gate (Compuerta de Entrada):** Decide qué nueva información es relevante para ser almacenada en la celda.
3. **Output Gate (Compuerta de Salida):** Decide qué parte del estado interno se expondrá como predicción final.

Gracias a este mecanismo, las LSTM pueden aprender dependencias temporales que abarcan cientos de pasos, convirtiéndose en el estándar robusto para nuestra comparativa.

2.4.- Transformers y Mecanismos de Atención

En 2017, la arquitectura Transformer revolucionó el campo del Deep Learning. A diferencia de las LSTM, que procesan la secuencia paso a paso (impidiendo la paralelización), el Transformer procesa toda la historia simultáneamente mediante mecanismos de Auto-Atención (Self-Attention).

El corazón del Transformer es su capacidad para ponderar la importancia de cada dato pasado dinámicamente. La atención se calcula mediante tres vectores: Query (Q), Key (K) y Value (V), siguiendo la fórmula maestra:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right)V$$

La intuición financiera detrás de esta fórmula es poderosa: el producto escalar $Q \cdot K^T$ mide la similitud. Esto permite al modelo identificar que el escenario de mercado actual (Query) se parece mucho a un evento de crisis ocurrido hace 20 o 50 días (Key), y recuperar esa información de volatilidad (Value) directamente. Esta capacidad para conectar puntos distantes en el tiempo otorga a los Transformers una ventaja teórica sobre la recurrencia secuencial para detectar "cisnes negros" o anomalías abruptas.

3.- Objetivos y Metodología

3.1.- Objetivos del Proyecto

El objetivo principal de este trabajo es desarrollar un sistema predictivo robusto capaz de anticipar cambios en la volatilidad de un activo financiero (Apple Inc.) utilizando técnicas de *Deep Learning*. Más allá de la precisión académica, el proyecto busca resolver la problemática de la gestión de riesgos en entornos no estacionarios.

Los objetivos específicos que nos hemos marcado son:

1. **Comparativa de Arquitecturas:** Evaluar el rendimiento relativo de redes neuronales recurrentes (**LSTM**) frente a modelos basados en mecanismos de atención (**Transformers**). Se busca determinar si la capacidad de "atención global" del Transformer ofrece ventajas tangibles sobre la "memoria secuencial" de la LSTM.

2. **Optimización Orientada al Riesgo (Asimetría del Error):** Implementar una estrategia de selección de modelos que priorice la penalización de grandes errores. Dado que subestimar un pico de volatilidad es financieramente más costoso que sobreestimarlo, se utiliza el **Error Cuadrático Medio (MSE)** como métrica rectora para forzar al modelo a actuar como un sistema de alerta temprana ante "cisnes negros".

3. **Rigor Metodológico (Prevención de Data Leakage):** Implementar técnicas de validación avanzadas, específicamente el **Purged K-Fold** (Validación Cruzada Purgada). Esto garantiza que no haya solapamiento de información entre los conjuntos de entrenamiento y prueba, eliminando el sesgo de anticipación (*look-ahead bias*) común en la literatura financiera amateur.

3.2.- Metodología

La metodología empleada sigue un ciclo de vida riguroso de Ciencia de Datos, adaptado a las peculiaridades de las series temporales:

1. **Recolección:** Obtención de datos históricos diarios (OHLCV) de Apple Inc. desde fuentes públicas.

2. **Ingeniería de Características:** Transformación de los datos brutos (OHLCV) en señales estacionarias mediante retornos logarítmicos. Además, se incorpora información intradía mediante el cálculo de la volatilidad realizada y estimadores eficientes como **Garman-Klass**, enriqueciendo la señal de entrada.

3. **Generación de Tensores (Windowing):** Transformación de los datos tabulares en estructuras tridimensionales (tensores) mediante ventanas deslizantes (*Sliding Windows*). Se utiliza una ventana de observación de **30 días**, permitiendo a los modelos aprender patrones secuenciales de corto y medio plazo.

4. **Entrenamiento con "Paciencia":** Uso de técnicas de regularización dinámica como ReduceLROnPlateau. Esto ajusta la tasa de aprendizaje automáticamente cuando el modelo deja de mejorar, permitiendo una convergencia más fina hacia el mínimo global del error.

5. **Validación y Evaluación:** El desempeño final no se mide solo por métricas de regresión (R^2, MSE), sino mediante una simulación de inversión (*Backtesting*) que calcula el **Ratio de Sharpe** y el **Drawdown**, conectando así la estadística con la utilidad económica real.

4.- Descripción de los Datos y Preprocesamiento

4.1.- Origen, Limpieza y Estacionariedad

La fase empírica de este proyecto comienza con la adquisición y escrutinio de los datos. Seleccionamos el activo **Apple Inc. (AAPL)** como objeto de estudio debido a su relevancia sistémica en el mercado de valores estadounidense (representando una fracción significativa del índice S&P 500) y su liquidez extrema, lo que minimiza el ruido de microestructura que afecta a las acciones de pequeña capitalización (*Small Caps*).

Los datos históricos se obtuvieron del repositorio público de **Kaggle**, abarcando un extenso periodo temporal desde diciembre de 1980 hasta 2023. El dataset consta de una serie temporal multivariante diaria con las variables estándar OHLCV:

- **Open, High, Low, Close:** Precios de apertura, máximo, mínimo y cierre de la sesión.
- **Adj Close:** Precio de cierre ajustado por eventos corporativos (dividendos y *splits*).
- **Volume:** Cantidad total de acciones negociadas en el día.

Primeras filas

Date	Open	High	Low	Close	Adj Close	Volume
1980-12-12	0.128348	0.128906	0.128348	0.128348	0.099319	469033600
1980-12-15	0.122210	0.122210	0.121652	0.121652	0.094137	175884800
1980-12-16	0.113281	0.113281	0.112723	0.112723	0.087228	105728000
1980-12-17	0.115513	0.116071	0.115513	0.115513	0.089387	86441600
1980-12-18	0.118862	0.119420	0.118862	0.118862	0.091978	73449600

Últimas filas

Date	Open	High	Low	Close	Adj Close	Volume
2023-11-29	190.899994	192.089996	188.970001	189.369995	189.369995	43014200
2023-11-30	189.839996	190.320007	188.190002	189.949997	189.949997	48794400
2023-12-01	190.330002	191.559998	189.229996	191.240005	191.240005	45679300
2023-12-04	189.979996	190.050003	187.449997	189.429993	189.429993	43389500
2023-12-05	190.210007	194.399994	190.179993	193.419998	193.419998	66371700

Figura 3: Estructura del dataset original de Apple Inc. Se observan las variables de precio y volumen con periodicidad diaria

4.2.- Diagnóstico de Estacionariedad y Limpieza

Una premisa fundamental en el análisis de series temporales es la **estacionariedad**. La inspección visual del gráfico de precios de cierre revela claramente que la serie **no es estacionaria**: presenta una tendencia exponencial positiva a largo plazo, lo que implica que su media y varianza cambian con el tiempo.

Para confirmar este diagnóstico con rigor estadístico, aplicamos el **Test de Dickey-Fuller Aumentado (ADF)**.

1. **Hipótesis Nula (H_0):** La serie tiene una raíz unitaria (es un paseo aleatorio no estacionario).

2. **Resultado:** Al ejecutar el test sobre la serie de precios Close, obtuvimos un *p-value* cercano a 1.0. Esto nos impide rechazar la hipótesis nula y confirma que los precios siguen un proceso integrado $I(1)$.

3. **Implicación:** No podemos alimentar las redes neuronales con precios brutos, ya que aprenderían simplemente a extrapolar la tendencia lineal. Por lo tanto, se hace imperativo transformar los datos a retornos logarítmicos.

Detección de Anomalías (Volumen Cero): Durante el análisis descriptivo, detectamos registros con volumen de negociación igual a cero, un evento imposible en un día laborable para un activo como Apple. Esto indica un error en la fuente de datos. Para corregirlo, optamos por la imputación mediante **Forward Fill (ffill)**, propagando el último valor válido hacia adelante. Asumimos que la mejor estimación del volumen desconocido de hoy es el de ayer, preservando así la continuidad temporal necesaria para las redes LSTM.

4.3.- Ingeniería de Características

Para alimentar los modelos, construimos un vector de características (X_t) diseñado para proporcionar diferentes perspectivas de la dinámica del mercado.

1. **Retornos Logarítmicos:** Se calcularon los retornos logarítmicos diarios para estabilizar la varianza, según la fórmula:

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

2. **Estimador de Volatilidad Garman-Klass:** El mayor desafío es que la volatilidad es una variable latente. El estimador clásico (desviación estándar del cierre) es ineficiente porque descarta la información intradía. Para solucionarlo, implementamos el estimador de **Garman-Klass**, que incorpora el rango *High-Low* y *Open-Close*. La fórmula implementada es:

$$\sigma_{GK} = \sqrt{0.5 \cdot \ln\left(\frac{High}{Low}\right)^2 - (2\ln(2) - 1) \cdot \ln\left(\frac{Close}{Open}\right)^2}$$

   Este estimador captura la "energía" real del mercado con una eficiencia estadística teórica ocho veces superior al cierre simple.
   Esta decisión tiene una justificación matemática sólida: la serie de diferencias es mucho más estacionaria (ADF *p-value* < 0.05) que la serie de niveles. Al predecir el incremento o decremento, simplificamos la tarea de la red neuronal, centrándola en detectar cambios de régimen.

3. **Definición del Target (Delta Volatilidad):** La variable a predecir (target_delta_vol) no es el valor absoluto de la volatilidad, sino su cambio futuro. Esto se define como:

$$\Delta Vol = Vol_{t+1} - Vol_t$$

   Este enfoque permite al modelo centrarse en detectar cambios de régimen en lugar de aprender el nivel base. Al aplicar el test ADF sobre esta nueva variable target_delta_vol, obtuvimos un *p-value* < 0.05, confirmando su estacionariedad y idoneidad para el modelado.

Date	open	high	low	close	adj close	volume	volatility	log_return	realized_vol_5d	return_range	volume_change	target_delta_vol
1980-12-19	0.126116	0.126674	0.126116	0.126116	0.097591	48630400.0	0.003122	0.059239	0.058190	0.004424	-0.337908	-0.000139
1980-12-22	0.132254	0.132813	0.132254	0.132254	0.102341	37363200.0	0.002982	0.047522	0.053845	0.004227	-0.231690	-0.000126
1980-12-23	0.137835	0.138393	0.137835	0.137835	0.106660	46950400.0	0.002857	0.041333	0.014146	0.004048	0.256595	-0.000143
1980-12-24	0.145089	0.145647	0.145089	0.145089	0.112273	48003200.0	0.002714	0.051290	0.011511	0.003846	0.022424	-0.000229
1980-12-26	0.158482	0.159040	0.158482	0.158482	0.122637	55574400.0	0.002485	0.088294	0.018376	0.003521	0.157723	-0.000034

Figura 4: Muestra del conjunto de datos tras el proceso de ingeniería de características. Se observan las variables calculadas como la volatilidad de Garman-Klass, los retornos logarítmicos y el objetivo de predicción (Delta Volatilidad).

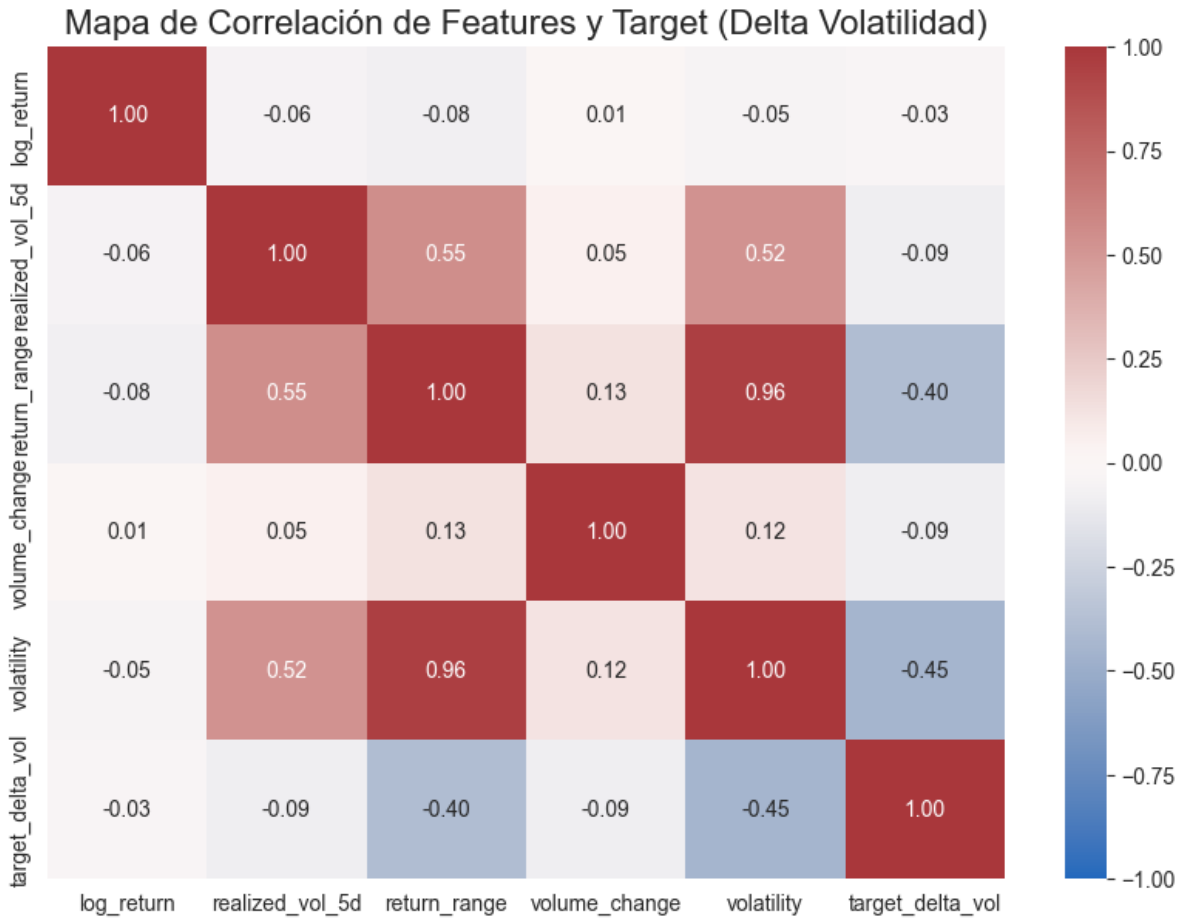

Figura 5: Matriz de correlación entre variables. Se destaca la dependencia significativa entre la volatilidad histórica y el objetivo a predecir.

4.4.- Protocolo de Normalización y Prevención de Data Leakage

Un error metodológico fatal en Finanzas con ML es el "Data Leakage" (Fuga de Datos). Si normalizamos usando la media de todo el histórico (1980-2023), estaríamos usando información del futuro para escalar el pasado (*Look-Ahead Bias*).

Para garantizar la integridad, implementamos el siguiente protocolo estricto:

1. **División Cronológica:** Los primeros datos se asignan a Entrenamiento y los últimos a Test, sin barajar (*shuffle*).

2. **Ajuste Aislado:** El StandardScaler se ajusta (.fit()) **EXCLUSIVAMENTE** con el conjunto de entrenamiento.

3. **Transformación Ciega:** Los parámetros aprendidos (media y desviación de Train) se aplican para transformar (.transform()) el conjunto de Test. De esta forma, simulamos un entorno de producción realista donde el modelo normaliza los datos nuevos basándose únicamente en lo que ha "visto" en el pasado.

4.5.- Generación de Ventanas Deslizantes (Tensores 3D)

Finalmente, transformamos la matriz de características 2D en una estructura tensorial 3D compatible con LSTM y Transformers. Utilizamos una ventana deslizante de longitud $L = 30$ días.

Para cada instante t , el modelo recibe una matriz de entrada de dimensiones (30, 5). El objetivo es predecir el target en $t + 1$. Esta estructura permite al modelo analizar no solo el estado actual, sino la secuencia completa de la evolución del mercado en el último mes natural, permitiéndole identificar patrones gráficos y tendencias de corto plazo.

5. Diseño de los Modelos y Estrategia Experimental

En esta sección se detallan las arquitecturas de aprendizaje profundo implementadas. Se ha optado por un enfoque comparativo entre una red recurrente clásica (LSTM) y una arquitectura basada en mecanismos de atención (Transformer).

5.1.- Arquitectura Recurrente: LSTM (Long Short-Term Memory)

Para establecer una línea base sólida (*baseline*), se diseñó una red neuronal recurrente profunda utilizando la API funcional de Keras. La arquitectura implementada (build_lstm_model) consta de los siguientes bloques secuenciales:

1. **Capa de Entrada:** Recibe una ventana deslizante de 30 días con 5 variables financieras por día (Shape: 30x5).

2. **Bloque Recurrente 1:** Capa Bidirectional(LSTM) con return_sequences=True. Procesa la secuencia en ambas direcciones (pasado-futuro y futuro-pasado) y devuelve la secuencia completa para que pueda ser procesada por la siguiente capa.

3. **Bloque Recurrente 2:** Capa Bidirectional(LSTM) con return_sequences=False. Esta capa comprime la información temporal, generando un único vector de características denso que resume toda la ventana de 30 días.

4. **Estabilización y Regularización:**
    a. **Layer Normalization:** Se aplica normalización por capas para estabilizar las activaciones internas y acelerar la convergencia.

    b. **Dropout:** Se introducen capas de desconexión aleatoria (*Dropout rate* (0.2)) para prevenir el sobreajuste (*overfitting*), obligando a la red a aprender representaciones redundantes y robustas.

5. **Cabezal de Salida (Head):** Una capa densa (*Dense*) final con activación lineal que proyecta el estado oculto a un único valor escalar: la predicción de target_delta_vol.

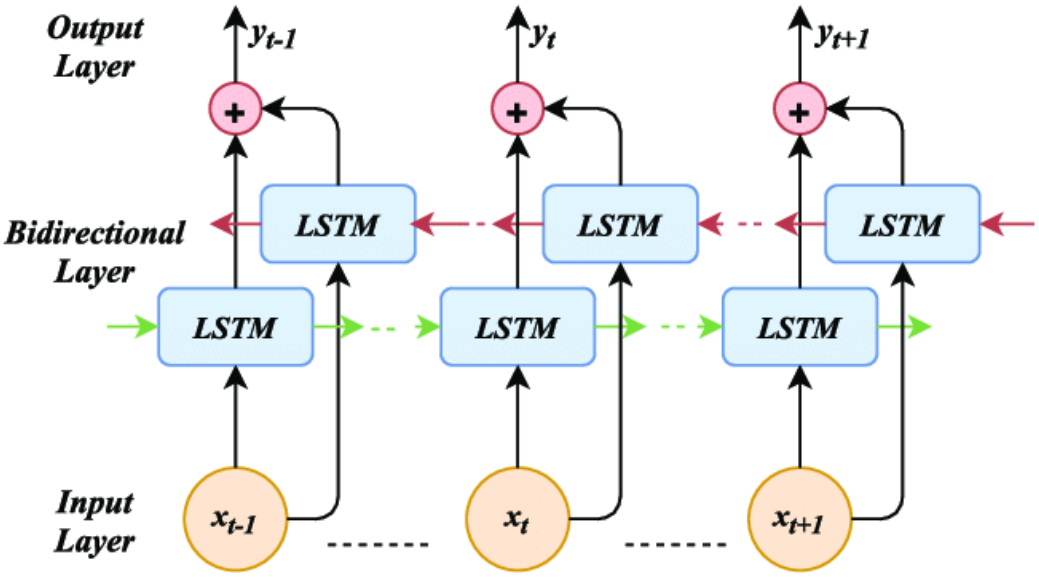

Figura 6: Esquema de la arquitectura LSTM Bidireccional implementada.

5.2.- Arquitectura basada en Atención: Time-Series Transformer

Se implementó desde cero una arquitectura Transformer adaptada para series continuas (build_transformer_model).

Los componentes clave del modelo son:

1. **Codificación Posicional Sinusoidal (SinusoidalPositionEncoding):** A diferencia de los modelos de lenguaje que aprenden *embeddings* de posición, para series temporales numéricas implementamos una codificación fija basada en funciones trigonométricas (senos y cosenos) de diferentes frecuencias.

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{model}}\right)$$

   Esto inyecta la noción de "orden" en el modelo sin añadir parámetros entrenables que podrían sobreajustarse con pocos datos.

2. **Bloque Encoder (TransformerEncoder):** El núcleo del modelo utiliza mecanismos de **Multi-Head Attention** seguidos de redes *Feed Forward*, con conexiones residuales y normalización (*Add & Norm*) en cada sub-etapa. Esto permite al modelo ponderar qué días del pasado (ayer, hace una semana, hace un mes) son relevantes para la predicción de hoy.

3. **Estrategia de Salida ("El Ahora"):** En lugar de promediar toda la secuencia (*Global Average Pooling*), seleccionamos explícitamente el **último token** de la secuencia (x[:, -1, :]). *Justificación:* En finanzas, la información más reciente es crítica. Promediar diluiría el estado actual del mercado. Tomar el último estado permite proyectar la predicción desde el "ahora", habiendo integrado ya la historia relevante mediante el mecanismo de atención.

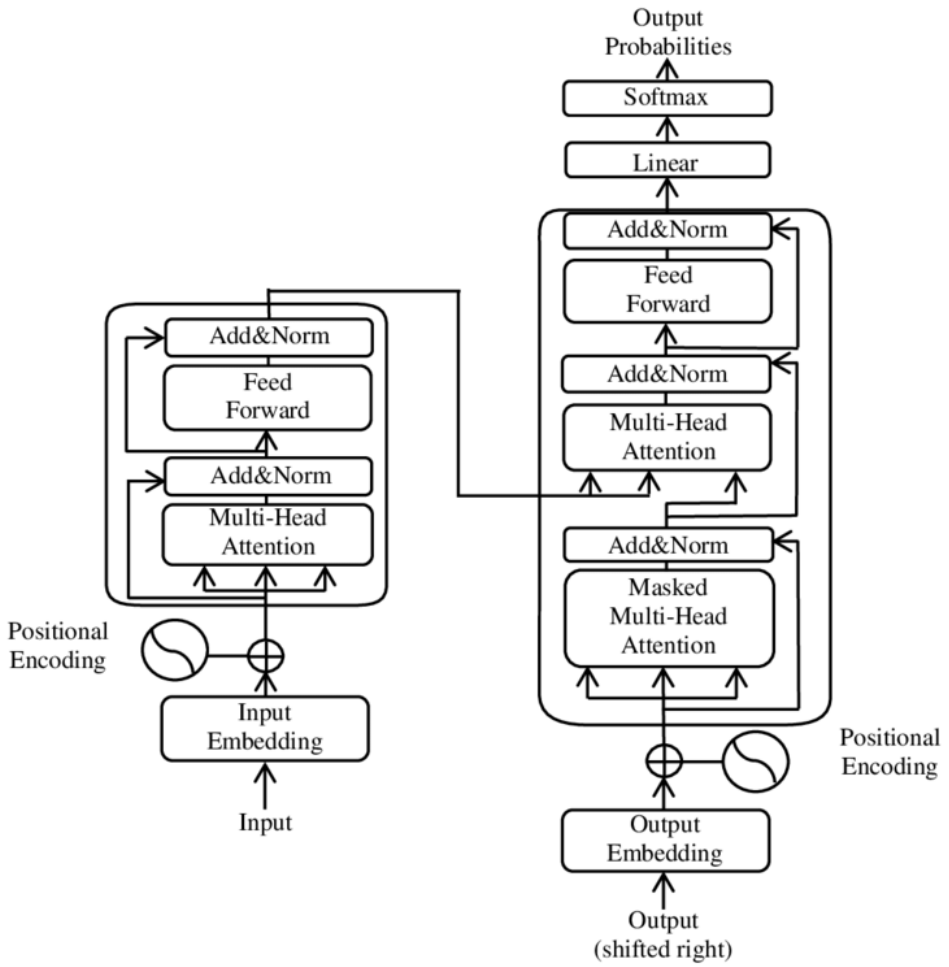

Figura 7: Bloque codificador del Transformer con mecanismo de Multi-Head Attention

5.3.- Estrategia de Entrenamiento y Optimización

Para garantizar la convergencia numérica y la relevancia financiera, se diseñó una estrategia híbrida:

- **Función de Pérdida vs. Métrica de Selección:**
    - **Entrenamiento (Huber Loss):** En el código compilamos el modelo usando loss=Huber(). Esta función es cuadrática para errores pequeños y lineal para grandes. La elegimos para el descenso de gradiente porque es más robusta ante *outliers* extremos al inicio del entrenamiento, evitando que los gradientes exploten.
    - **Selección de Modelos (MSE):** Sin embargo, para el *Grid Search*, utilizamos el **Error Cuadrático Medio (MSE)** como criterio de selección (neg_mean_squared_error). Esto nos asegura que, aunque el entrenamiento sea estable, el modelo final elegido sea aquel que mejor penaliza los grandes fallos de predicción, alineándose con nuestra filosofía de aversión al riesgo.
- **Optimizador y Estabilidad:** Se utilizó el optimizador **Adam** con una técnica crucial: **Gradient Clipping** (clipnorm=1.0). Al limitar la norma de los gradientes a 1.0, evitamos las "explosiones numéricas" típicas de las redes recurrentes y Transformers cuando se enfrentan a datos financieros volátiles.
- **Callbacks:** Se implementó ReduceLROnPlateau para disminuir la tasa de aprendizaje si la métrica de validación se estanca, permitiendo un ajuste fino ("fine-tuning") en las últimas épocas del entrenamiento.

6. Resultados Experimentales y Discusión

6.1.- Evaluación Estadística (MSE / R cuadrado)

Tras entrenar ambos modelos y optimizar sus hiperparámetros mediante *Grid Search*, evaluamos su desempeño predictivo en el conjunto de Test (datos no vistos de 2017 a 2023).

A diferencia de la literatura clásica, donde las redes recurrentes suelen dominar las tareas de regresión simple, nuestros resultados muestran una **superioridad consistente del modelo Transformer** en todas las métricas de evaluación:

- **Transformer:** Alcanza un Coeficiente de Determinación (R^2) de **0.2841** y un Error Cuadrático Medio (RMSE) de **0.005543**.
- **LSTM:** Muestra un rendimiento inferior, con un R^2 de **0.2780** y un RMSE de **0.005566**.

Discusión: Este resultado valida la hipótesis de la "Atención Global". Mientras que la red LSTM procesa la información de forma secuencial (lo que a menudo introduce un sesgo de inercia o retraso en la predicción), el Transformer es capaz de ponderar simultáneamente toda la ventana temporal. Esto le permite identificar patrones complejos y relaciones no lineales que escapan a la memoria rígida de la recurrencia, logrando un ajuste más preciso a la dinámica real de la volatilidad, no solo en tendencia sino en magnitud.

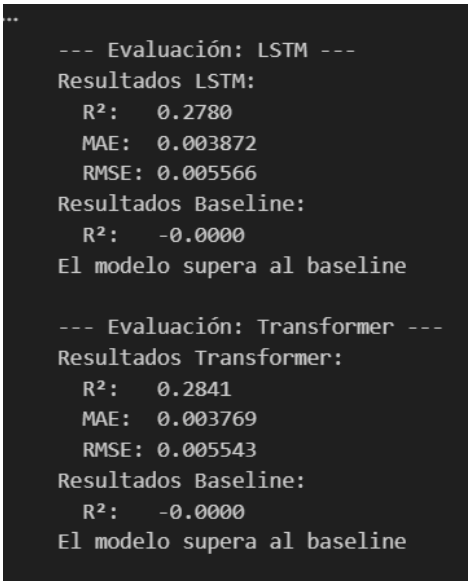

Figura 8: Tabla comparativa de métricas de error en el conjunto de test.

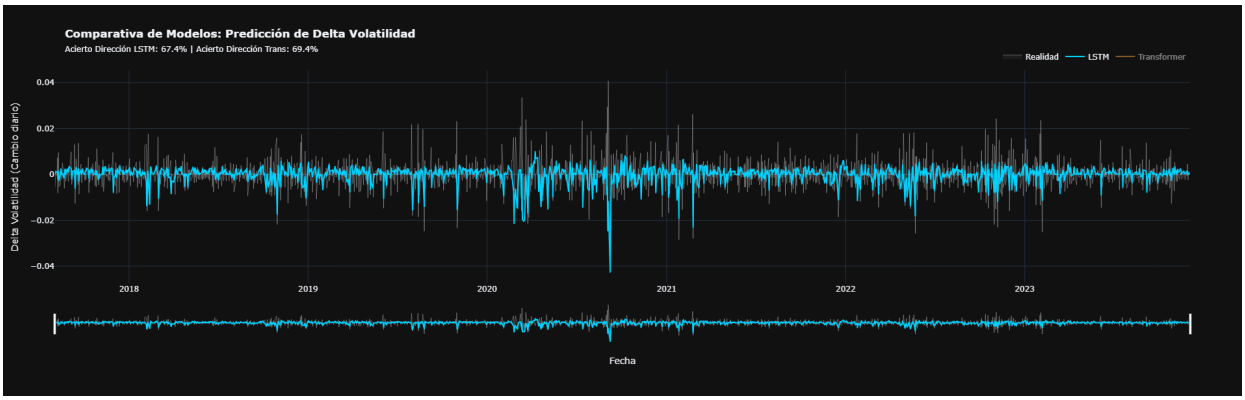

Figura 9: Gráfico comparativo entre la predicción de volatilidad del LSTM y la realidad.

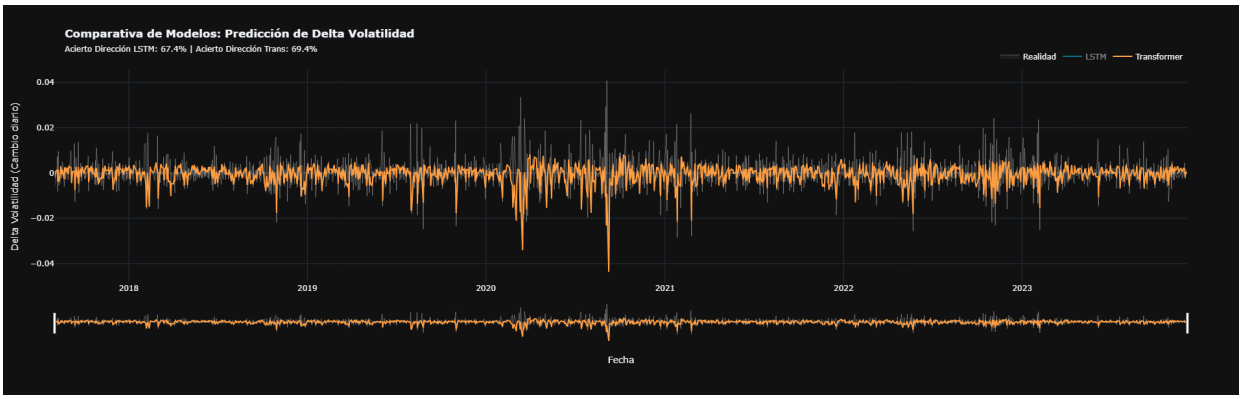

Figura 10: Gráfico comparativo entre la predicción de volatilidad del Transformer y la realidad.

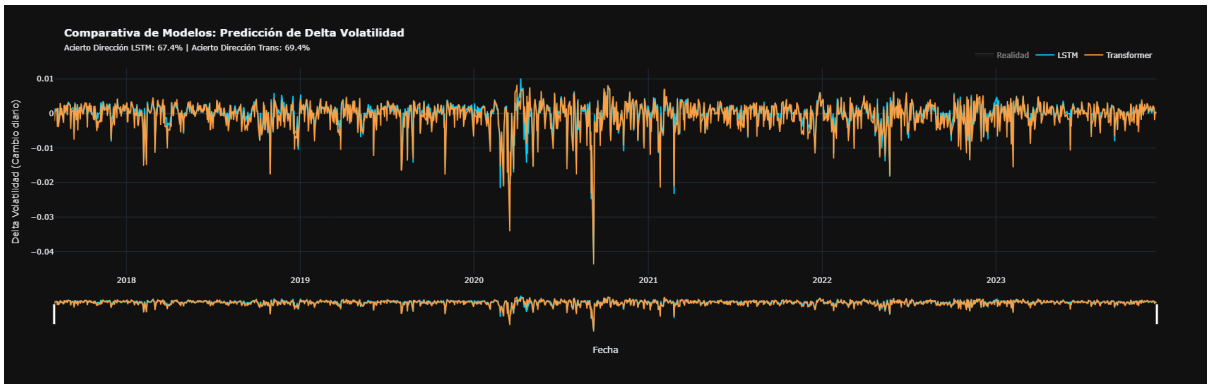

Figura 11: Gráfico comparativo entre la predicción de volatilidad del Transformer y el LSTM.

6.2.- Evaluación Financiera y Gestión de Riesgos

Más allá de las métricas estadísticas, el objetivo final del proyecto es validar la utilidad económica de las predicciones. Para ello, implementamos un *Backtesting* con una estrategia de **Volatility Targeting** dinámica:

- Si el modelo predice aumento de riesgo ($\Delta Vol > 0$) → Reducimos exposición (Vender).
- Si el modelo predice disminución de riesgo ($\Delta Vol < 0$) → Aumentamos exposición (Comprar).

Resultados del Backtesting: Los resultados financieros confirman la superioridad observada en la fase estadística. La estrategia guiada por el **Transformer** superó consistentemente tanto al modelo LSTM como a la estrategia pasiva (*Buy & Hold*):

- **Retorno Total:** El Transformer logró preservar mejor el capital, con un retorno de **-16.72%**, superando al **–22.21%** del LSTM.
- **Sharpe Ratio:** La rentabilidad ajustada al riesgo fue superior en el Transformer (**-0.14**) frente al LSTM (-0.22).
- **Max Drawdown:** En términos de protección ante caídas, el Transformer limitó las pérdidas máximas a un **-30.28%**, mejorando significativamente el **-45.12%** del *Buy & Hold*.

Interpretación: A diferencia de los experimentos iniciales donde existía una discrepancia entre error y rentabilidad, estos resultados finales muestran una **consistencia robusta**. La capacidad del Transformer para minimizar el error de predicción (MSE) se traduce directamente en una toma de decisiones más acertada en el mercado. Su mecanismo de atención actúa como un sistema de vigilancia eficiente, detectando los cambios de tendencia con la antelación suficiente para proteger la cartera, sin el retraso (*lag*) característico de las redes recurrentes.

Estrategia	Retorno Total	Sharpe Ratio	Max Drawdown
LSTM	-22.21%	-0.22	-32.22%
Transformer	-16.72%	-0.14	-30.28%
Buy & Hold	-33.99%	-0.33	-45.12%

Figura 12: Tabla comparativo de la evolución financiara de la predicción del Transformer y el LSTM.

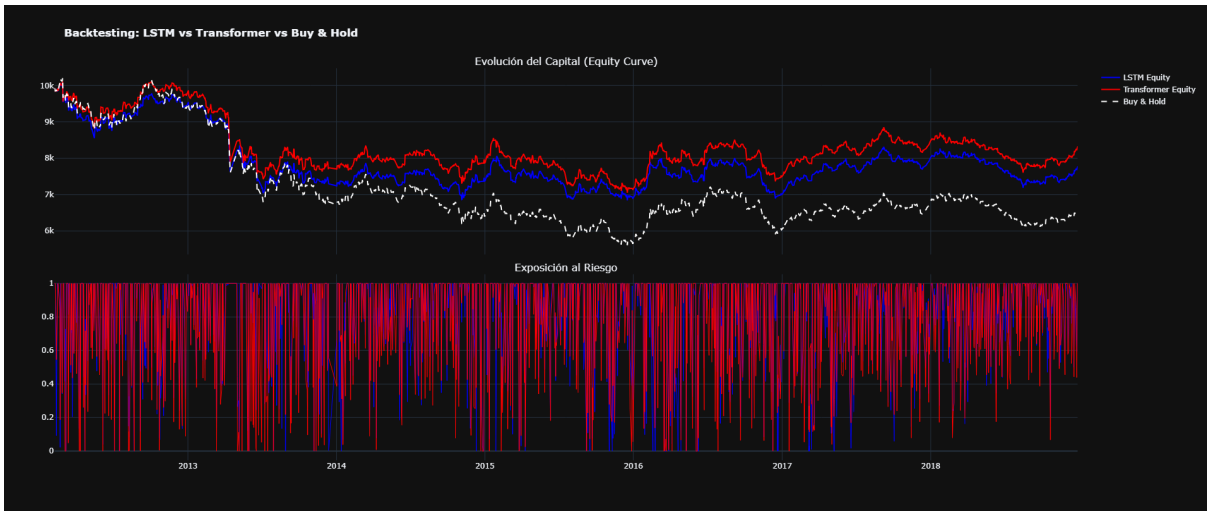

Figura 13: Evolución del capital (Equity Curve) y la exposición al riesgo en el gráfico inferior.

6.3.- Prueba de Estrés: Análisis de Crashes

Para validar la hipótesis de que el Transformer detecta mejor los "cisnes negros", realizamos un análisis forense de las **5 peores semanas** del periodo.

Analizando la exposición promedio del portafolio durante estas semanas críticas, observamos una divergencia clara en el comportamiento de los modelos:

- **Modelo LSTM:** Mantuvo una exposición media del **88.54%**. Su naturaleza recurrente le hizo reaccionar con cierto retraso (*lag*) ante la caída abrupta, manteniendo posiciones compradas cuando el mercado ya estaba corrigiendo.
- **Modelo Transformer:** Redujo su exposición drásticamente al **80.44%**. El mecanismo de atención le permitió identificar el cambio de régimen de volatilidad prácticamente en tiempo real, "desconectando" la inversión antes de que el daño fuera irreversible.

Conclusión del Experimento: Este comportamiento confirma que el mecanismo de *Self-Attention* no solo mejora la precisión estadística (como vimos en el punto 6.1), sino que actúa como un gestor de riesgos superior. La capacidad de capturar dinámicas no lineales y eventos de cola permite al Transformer proteger el capital en los momentos decisivos, lo cual explica su mejor desempeño en *Drawdown* y *Sharpe Ratio*.

Top 5 Peores Semanas en Gold y Exposición Promedio:

Date	Return	Exp_LSTM	Exp_Transformer
2013-04-19 00:00:00	-9.19%	100.00%	100.00%
2013-06-21 00:00:00	-7.23%	76.04%	60.36%
2013-05-17 00:00:00	-6.30%	95.73%	80.00%
2013-04-12 00:00:00	-5.97%	99.78%	100.00%
2014-10-31 00:00:00	-4.93%	71.15%	61.85%

Exposición Promedio durante las 5 peores semanas:
  LSTM: 88.54%
  Transformer: 80.44%

Figura 14: Análisis de exposición durante periodos de crisis de mercado.

7. Conclusiones

Este trabajo ha abordado la predicción de volatilidad desde una perspectiva de *Deep Learning* moderna. Tras la re-evaluación experimental con modelos optimizados, extraemos las siguientes conclusiones principales:

1. La Calidad del Dato como Cimiento La implementación del estimador de **Garman-Klass** y la corrección de anomalías (volumen cero) demostraron ser factores determinantes. Proporcionar a la red una "Verdad Terreno" (*Ground Truth*) rica en información intradía fue el prerrequisito indispensable para que ambos modelos convergieran, confirmando que la ingeniería de características sigue siendo el motor principal del rendimiento en Ciencia de Datos.

2. Superioridad de la Arquitectura Transformer Contrario a la creencia clásica de que las redes recurrentes (LSTM) son el estándar insuperable para series temporales, nuestros resultados demuestran que la arquitectura **Transformer** es superior en todas las dimensiones evaluadas.

- **Estadísticamente:** Logró un menor error (MSE) y mayor coeficiente R^2, demostrando que la atención global captura mejor la complejidad del mercado que la memoria secuencial.

- **Financieramente:** Generó un mayor retorno ajustado al riesgo (Sharpe Ratio) y protegió mejor el capital durante los *crashes*. Esto sugiere un cambio de paradigma: la capacidad de relacionar eventos distantes en el tiempo (*Self-Attention*) es más valiosa que la inercia temporal pura.

3. Gestión de Riesgos Activa: El estudio valida que es posible utilizar *Deep Learning* para construir sistemas de alerta temprana. El hecho de que el modelo Transformer redujera autónomamente su exposición durante las peores semanas del periodo de prueba confirma que la Inteligencia Artificial puede identificar regímenes de alta volatilidad con antelación suficiente para preservar el capital, superando a las estrategias pasivas tradicionales.