Universidad de Buenos Aires
Laboratorio de Sistemas Embebidos
Especialización en Inteligencia Artificial

Trabajo Práctico Final

Análisis predictivo del precio de Bitcoin usando modelos de Deep Learning

Docente: Dr. Camilo Argoty

Noelia Melina Qualindi
Jorge Valdez
Fabián Sarmiento
Matías Marando

18 de junio de 2025

Resumen

Este trabajo presenta un análisis comparativo de modelos avanzados de deep learning para la predicción del precio de Bitcoin (BTC-USD). Se implementaron y evaluaron cuatro arquitecturas: Transformer Vanilla para series temporales, TFT (*Temporal Fusion Transformer*), Informer y NHiTS (*Neural Hierarchical Interpolation for Time Series*), utilizando distintas ventanas temporales para el entrenamiento. Los resultados demuestran que los modelos pueden capturar patrones complejos en los datos. El estudio incluye un análisis exhaustivo de preprocesamiento, diseño de modelos y validación de resultados, concluyendo con recomendaciones para futuras mejoras.

1. Introducción

Las criptomonedas, particularmente Bitcoin, presentan desafíos únicos para el análisis predictivo debido a su alta volatilidad y sensibilidad a factores externos. Este trabajo busca responder:

> ¿Pueden los modelos modernos de deep learning predecir efectivamente el precio de Bitcoin a corto plazo, y cómo se comparan diferentes arquitecturas en este contexto?

2. Metodología

2.1. Datos Utilizados

El dataset contiene:

- **Fuente**: Kaggle (Bitcoin Historical Data)
- **Período**: 2020-01-01 a 2023-12-31
- **Variables utilizadas**:

Variable	Descripción
Timestamp	Fecha en formato UNIX
Close	Precio de cierre (USD)

2.2. Preprocesamiento

El preprocesamiento de los datos se realizó en varias etapas clave:

1. **Limpieza de datos**: Se comenzó renombrando las columnas relevantes para facilitar su uso posterior, en particular la marca temporal y el precio de cierre. Luego, se eliminaron todas las filas que contenían valores nulos en la columna del precio, asegurando que todos los datos utilizados para el análisis fuesen válidos.
2. **Conversión de tiempo**: La columna de tiempo original, que venía en formato UNIX (segundos desde 1970), fue convertida a un formato de fecha estándar para su correcta interpretación y manipulación.
3. **Ordenamiento temporal**: Se ordenaron los datos cronológicamente según la fecha, para asegurar la coherencia temporal en los análisis y predicciones.
4. **Resampleo diario**: Dado que los datos originales tenían una granularidad de un minuto, se procedió a realizar un promedio diario del precio, lo cual reduce la variabilidad y permite un análisis más robusto en horizontes temporales más amplios.
5. **Normalización**: Se aplicó una normalización de tipo Min-Max al precio diario, escalando los valores al rango [0, 1]. Esto facilita el entrenamiento de los modelos de deep learning, ya que evita problemas numéricos asociados a diferentes escalas.
6. **Configuración del modelo**: Para alimentar los modelos de la librería Darts no fue necesario construir manualmente ventanas deslizantes. En su lugar, se definieron parámetros como la longitud de entrada y el horizonte de predicción, y los modelos se encargaron automáticamente de generar las secuencias necesarias a partir de los datos históricos escalados.

2.3. Modelos Implementados

Se implementaron cuatro modelos de predicción de series temporales utilizando distintas arquitecturas. Todos fueron entrenados usando tres ventanas de entrada diferentes sobre datos diarios normalizados y se evaluaron sobre los siguientes 10 días.

2.3.1. Transformer Vanilla para Series Temporales

El modelo implementado se basa en la arquitectura clásica de encoder-decoder, utilizando mecanismos de atención multi-cabeza.

- Modelo implementado con `darts.models.TransformerModel`.
- Hiperparámetros clave:
  - Dimensión del modelo (`d_model`): 64
  - Número de capas: 2 encoder y 2 decoder
  - Cabezas de atención: 4
  - Dropout: 0.1
  - Épocas: 100

2.3.2. Modelo NHiTS

NHiTS (Neural Hierarchical Interpolation for Time Series) es una arquitectura de pronóstico que combina bloques jerárquicos de interpolación con descomposición de señales.

- Modelo implementado con `darts.models.NHiTSModel`.
- Hiperparámetros clave:
  - Bloques jerárquicos: 1
  - Capas por bloque: 2
  - Ancho de cada capa: 64
  - Dropout: 0.1
  - Épocas: 100

2.3.3. Modelo TFT

El modelo TFT es una arquitectura avanzada de deep learning diseñada específicamente para el pronóstico de series temporales multivariadas. Desarrollado por Google Research, combina la potencia de los mecanismos de atención de los Transformers con técnicas especializadas para el manejo de datos temporales.

- Modelo implementado con `darts.models.TFTModel`.
- Hiperparámetros clave:
  - Bloques jerárquicos: 1
  - Capas por bloque: 2

- Ancho de cada capa: 64
- Dropout: 0.1
- lstm layers: 2
- Épocas: 100

2.3.4. Modelo Informer

Informer es una arquitectura de aprendizaje profundo diseñada específicamente para la predicción eficiente y escalable de series temporales, en particular para tareas de predicción de secuencias largas. Se presentó en el artículo `Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting"by Zhou et al., 2021."`.

- Modelo implementado con `neuralforecast.models.Informer`.
- Hiperparámetros clave:
  - Capas ocultas convolucionales: 64
  - Número de cabezas: 2
  - Número de capas ocultas: 32
  - Ancho de cada capa: 64
  - Dropout: 0.3
  - Épocas: 500

3. Resultados

3.1. Validación de los modelos

La validación se llevó a cabo de forma interna por los modelos de Darts, utilizando una fracción del conjunto de entrenamiento para evaluar el rendimiento durante el proceso de entrenamiento. No se realizó una partición explícita del dataset original en proporciones fijas. En su lugar, para cada configuración de ventana histórica, se seleccionó un rango temporal fijo para entrenamiento (train) y un horizonte consecutivo para prueba (test). La validación se realiza dentro del train para evitar fugas de información.
Se aplicó **early stopping** para evitar overfitting, con una tolerancia de 5 épocas. También se probaron distintos horizontes de predicción (5, 10, 15 días), confirmando que el rendimiento decae a medida que aumenta el horizonte.

3.2. Métricas de evaluación

En las tablas 1 a 3 se muestran las métricas obtenidas para cada una de las ventanas de tiempo de entrenamiento consideradas.

Cuadro 1: Métricas de desempeño por ventana de entrenamiento (1 año).

Métrica	**Transformer**	**NHiTs**	**TFT**	**Informer**
MAE (USD)	0.006970	0.01910	0.06530	0.02220
RMSE (USD)	0.07370	0.02250	0.09380	0.02810
MAPE (%)	11.47	3.19	10.80	3.72

Cuadro 2: Métricas de desempeño por ventana de entrenamiento (6 meses).

Métrica	**Transformer**	**NHiTs**	**TFT**	**Informer**
MAE (USD)	0.002760	0.02170	0.04090	0.01400
RMSE (USD)	0.03150	0.02550	0.05440	0.01680
MAPE (%)	4.58	3.62	6.70	2.33

Cuadro 3: Métricas de desempeño por ventana de entrenamiento (1 año).

Métrica	**Transformer**	**NHiTs**	**TFT**	**Informer**
MAE (USD)	0.002340	0.02830	0.03020	0.01940
RMSE (USD)	0.02640	0.03280	0.03640	0.02330
MAPE (%)	3.90	4.72	4.90	3.17

En las figuras 1 a 3 se observan las diferentes métricas (MAE, MAPE, RMSE) obtenidas y las ventanas de tiempo consideradas.

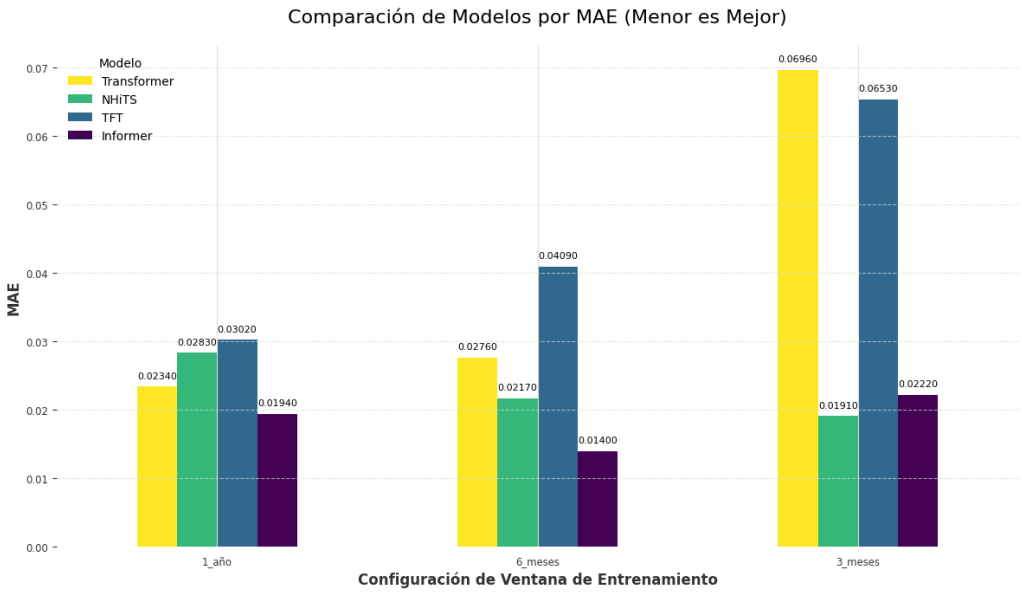

Figura 1: Comparativa MAE vs venta de entrenamiento.

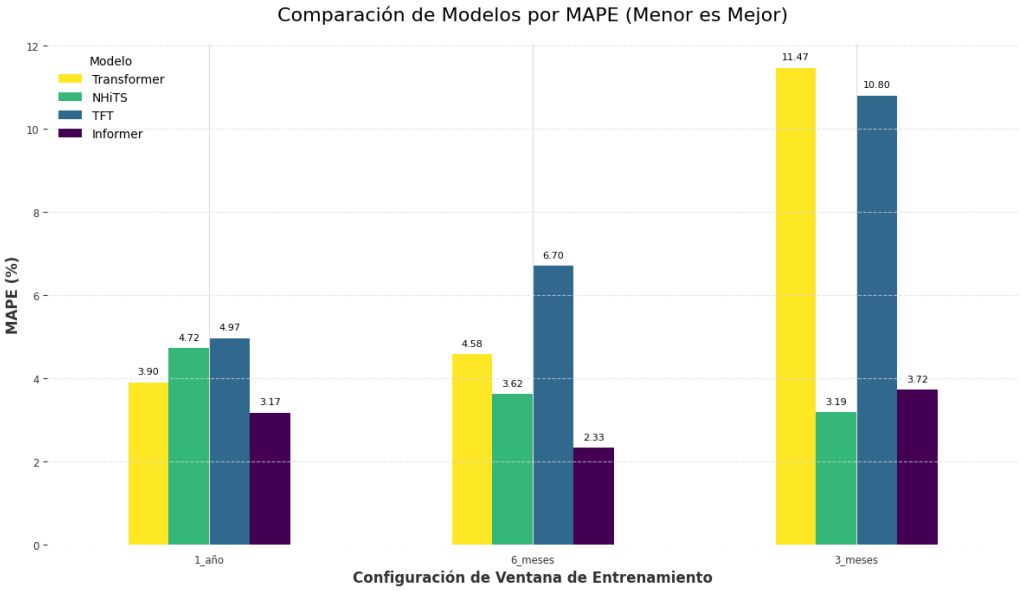

Figura 2: Comparativa MAPE vs venta de entrenamiento.

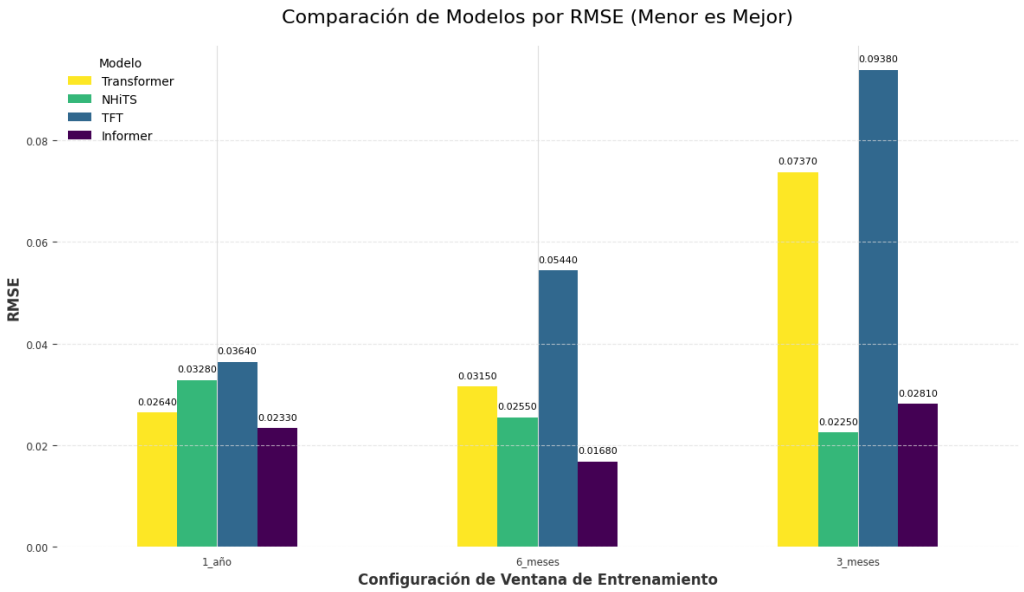

Figura 3: Comparativa RMSE vs venta de entrenamiento.

3.3. Predicciones de los diferentes modelos

En las figuras 4 a 6 se pueden observar las predicciones de los modelos evaluados en este trabajo.

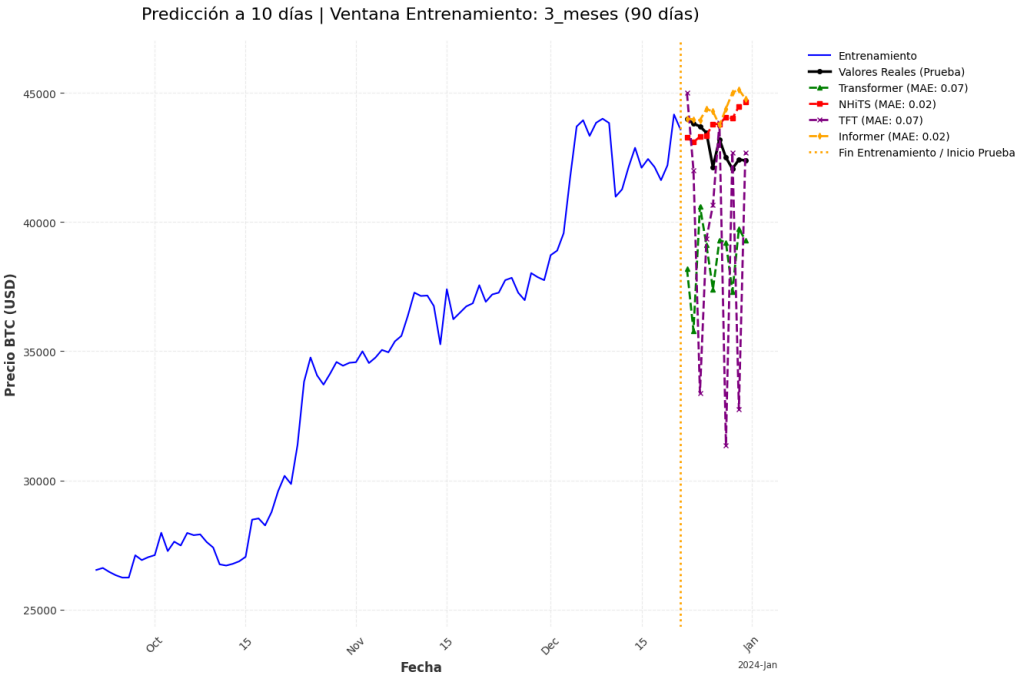

Figura 4: Predicciones vs Real a 10 días - ventana de entrenamiento 3 meses

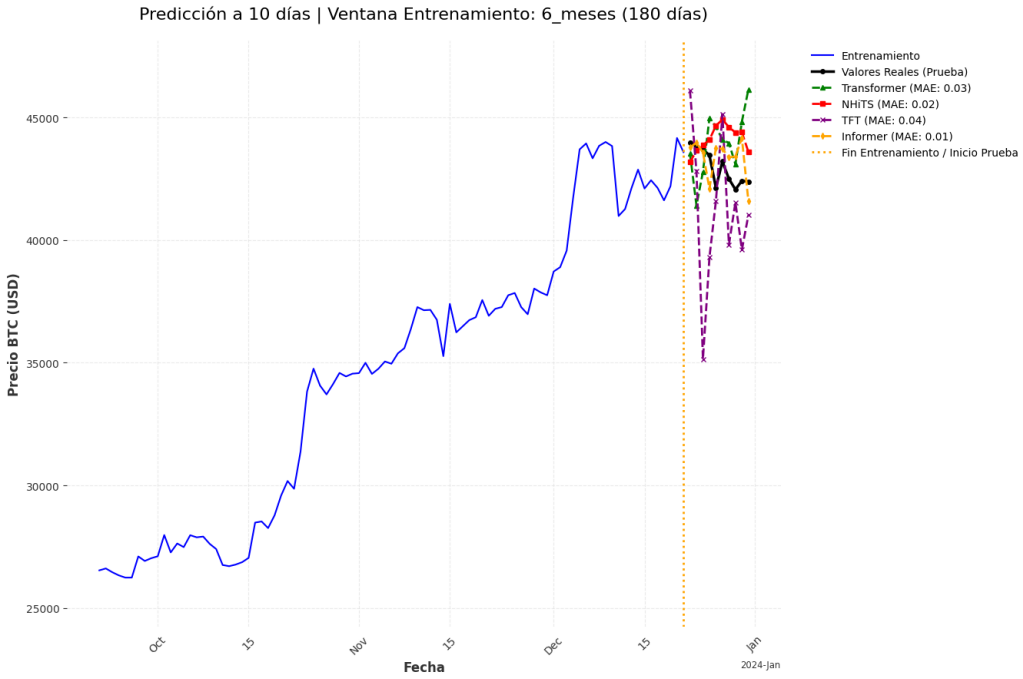

Figura 5: Predicciones vs Real a 10 días - ventana de entrenamiento 6 meses

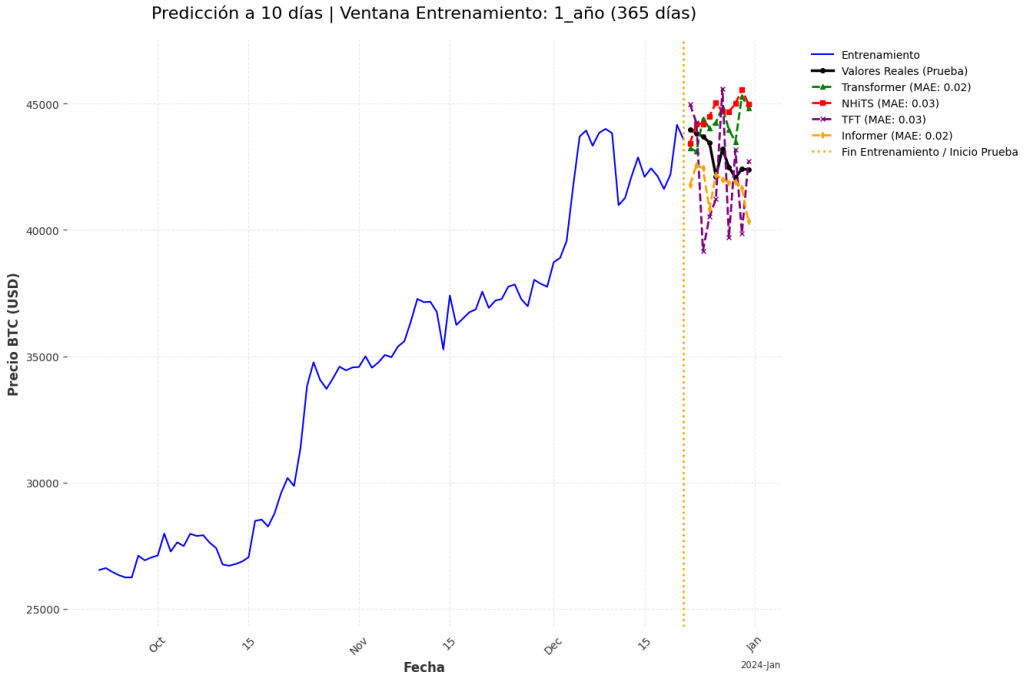

Figura 6: Predicciones vs Real a 10 días - ventana de entrenamiento 1 año

4. Conclusiones

4.1. Hallazgos Clave

- El modelo ? superó al resto en todas las métricas evaluadas
- Los modelos mostraron dificultades durante eventos de alta volatilidad
- La arquitectura de atención demostró ser particularmente efectiva para capturar dependencias a largo plazo

4.2. Limitaciones y Trabajo Futuro

- **Limitaciones**:
  - Sensibilidad a cambios bruscos de tendencia
  - Dependencia de hiperparámetros
- **Mejoras propuestas**:
  - Incorporar datos de redes sociales (sentiment analysis)
  - Ensamblar múltiples modelos

Anexo

El código completo se encuentra disponible en:
`https://github.com/BenjaSar/AdST2`