UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Posgrados

Súper Resolución de Series de Tiempo basada en Transformers

Proyecto de Titulación

Andrés Cárdenas Anchatipán

Julio Ibarra Fiallo

Director de Trabajo de Titulación

Trabajo de titulación de posgrado presentado como requisito para la obtención del título de Magíster en Ciencia de Datos

Quito, 01 de diciembre de 2024

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ
COLEGIO DE POSGRADOS

HOJA DE APROBACIÓN DE TRABAJO DE TITULACIÓN

Súper Resolución de Series de Tiempo basada en Transformers

Andrés Cárdenas

Nombre del Director del Programa:	Felipe Grijalva
Título académico:	Ph.D. en Ingeniería Eléctrica
Director del programa de:	Ciencia de Datos
Nombre del Decano del colegio Académico:	Eduardo Alba
Título académico:	Doctor en Ciencias Matemáticas
Decano del Colegio:	Ciencias e Ingenierías
Nombre del Decano del Colegio de Posgrados:	Dario Niebieskikwiat
Título académico:	Doctor en Física

Quito, diciembre 2024

© DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en la Ley Orgánica de Educación Superior del Ecuador.

Nombre del estudiante:	Andrés Cárdenas
Código de estudiante:	00339422
C.I.:	1721714234
Lugar y fecha:	Quito, 01 de diciembre de 2024.

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en http://bit.ly/COPETheses.

UNPUBLISHED DOCUMENT

Note: The following graduation project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on http://bit.ly/COPETheses.

RESUMEN

Nuestra arquitectura incluye una codificación de entrada con información posicional, una pila de bloques de codificadores Transformer con mecanismos de atención multi-cabezal y una capa de proyección lineal final para la reconstrucción de la salida. El modelo fue entrenado con 1000 series temporales de 5000 puntos cada una, utilizando un enfoque de aprendizaje autorregresivo enmascarado, y evaluado mediante la reconstrucción de versiones subsampleadas con 250 puntos. Se probaron dos configuraciones: una línea base con dimensiones más pequeñas y una versión más compleja con parámetros aumentados. Ambas configuraciones se entrenaron en una TPU v2-8.

Los resultados demuestran que el modelo identifica con éxito los patrones subyacentes de las series temporales, incluyendo variaciones en frecuencia y amplitud. La segunda configuración, que incrementa la dimensionalidad del modelo y el número de cabezas de atención, mejora la fidelidad de la reconstrucción, especialmente en la amplitud de la serie original. Sin embargo, esta mejora conlleva un mayor uso de recursos, alcanzando 190 GB de RAM y un tiempo de entrenamiento de 12 horas. El intento de seguir aumentando los parámetros del modelo estuvo limitado por los recursos computacionales disponibles.

Estos hallazgos resaltan el potencial de los modelos basados en Transformers para la super-resolución de series temporales, a la vez que subrayan las compensaciones entre complejidad del modelo, desempeño y uso de recursos.

Palabras clave: Series de Tiempo, Suér-resolución, Transformers, BERT, predicción enmascarada, mecanismo de atención, reconstrucción de alta resolución.

ABSTRACT

Our architecture includes input encoding with positional information, a stack of Transformer encoder blocks with multi-head self-attention mechanisms, and a final linear projection layer for output reconstruction. The model was trained on 1000 time series of 5000 points each, using masked autoregressive learning, and evaluated by reconstructing their downsampled versions containing 250 points. Two configurations were tested: a baseline with smaller model dimensions and a more complex version with increased parameters. Both configurations were trained on a TPU v2-8.

The results demonstrate that the model successfully identifies the underlying patterns of the time series, including variations in frequency and amplitude. The second configuration, which uses increased model dimensionality and attention heads, improves reconstruction fidelity, particularly in capturing the amplitude of the original series. However, this improvement comes at the cost of higher resource consumption, reaching 190 GB of RAM and 12 hours of training time. Further enhancements were limited by computational resources.

These findings highlight the potential of Transformer-based models for time series super-resolution while also underscoring the trade-offs between model complexity, performance, and resource utilization.

Key words: Time series, super-resolution, Transformers, BERT, masked prediction, attention mechanism, high-resolution reconstruction.

TABLA DE CONTENIDO

I Introducción 10

II Estado del arte 10
- II-A El Transformer: Bases, Funcionamiento y Aplicaciones . . . 10
- II-B Transformers en Series de Tiempo . . . 11
- II-C Imputación de valores faltantes en Series de Tiempo . . . 12

III Materiales y Metodología 13
- III-A Series de Tiempo sintéticas . . . 13
- III-B Descripción de la Arquitectura del Transformer . . . 14
- III-C Parámetros de entrenamiento . . . 15
- III-D Reconstrucción de Series de Tiempo . . . 15

IV Resultados 16
- IV-A Modelo 1 . . . 16
- IV-B Modelo 2 . . . 16
- IV-C Parámetros y Rendimiento . . . 17
- IV-D Detección de eventos . . . 17

V Conclusion 17

References 18

ÍNDICE DE TABLAS

I Parámetros del Modelo 15
II Parámetros del Modelo y Uso de Memoria 17

ÍNDICE DE FIGURAS

1 Valores faltantes en Serie temporal. 12
2 Series de Tiempo Sintéticas. 13
3 Eventos en Series de Tiempo. 14
4 Esquema Transformer BERT. 14
5 Masking Series de **entrada** 14
6 Reconstrucción 1 16
7 Reconstrucción 2 16
8 Detección de eventos 17

Súper Resolución de Series Temporales Basado en Transformers

Andrés Cárdenas Anchatipán, Julio Ibarra Fiallo

***Abstract*—La super-resolución de series temporales es una tarea desafiante que consiste en reconstruir datos de series temporales de alta resolución a partir de observaciones de baja resolución. En este trabajo, proponemos una arquitectura basada en Transformers, inspirada en el modelo BERT, adaptada para series temporales. El modelo está diseñado para predecir valores enmascarados dentro de una serie, capturando eficazmente las dependencias tanto locales como globales a lo largo del tiempo. Los datos de entrada consisten en series temporales sintéticas con alta variabilidad en amplitud y frecuencia, lo que permite evaluar rigurosamente la capacidad de generalización del modelo.**

Nuestra arquitectura incluye una codificación de entrada con información posicional, una pila de bloques de codificadores Transformer con mecanismos de atención multi-cabezal y una capa de proyección lineal final para la reconstrucción de la salida. El modelo fue entrenado con 1000 series temporales de 5000 puntos cada una, utilizando un enfoque de aprendizaje autorregresivo enmascarado, y evaluado mediante la reconstrucción de versiones subsampleadas con 250 puntos. Se probaron dos configuraciones: una línea base con dimensiones más pequeñas y una versión más compleja con parámetros aumentados. Ambas configuraciones se entrenaron en una TPU v2-8.

Los resultados demuestran que el modelo identifica con éxito los patrones subyacentes de las series temporales, incluyendo variaciones en frecuencia y amplitud. La segunda configuración, que incrementa la dimensionalidad del modelo, el número de cabezas de atención y la dimensión del feed forward mejora la fidelidad de la reconstrucción, especialmente en la amplitud de la serie original.

Estos hallazgos resaltan el potencial de los modelos basados en Transformers para la super-resolución de series temporales, a la vez que subrayan las compensaciones entre complejidad del modelo, desempeño y uso de recursos.

***Index Terms*—Time series, super-resolution, Transformers, BERT, masked prediction, attention mechanism, high-resolution reconstruction**

I. Introducción

En el contexto contemporáneo del análisis de datos, las series temporales constituyen un elemento esencial en un amplio espectro de disciplinas. Desde la economía y las finanzas hasta la meteorología y la ingeniería, estas secuencias ordenadas de observaciones en el tiempo permiten modelar patrones, realizar pronósticos y extraer valiosa información sobre la dinámica de los sistemas.

Estas series, que representan la evolución de variables a lo largo del tiempo, permiten comprender y predecir fenómenos complejos. Sin embargo, un desafío clave radica en la resolución de los datos, ya que muchas veces se dispone de series de tiempo de baja resolución que no capturan adecuadamente las variaciones a corto plazo ni los detalles necesarios para análisis precisos.

La reconstrucción de series de tiempo consiste en generar datos adicionales para aumentar la frecuencia de muestreo de una serie existente, pasando de baja a alta resolución. Tradicionalmente, enfoques como la interpolación lineal o el suavizado exponencial han sido utilizados para abordar este problema; sin embargo, estos métodos suelen fallar en capturar patrones complejos, dinámicas no lineales y relaciones contextuales presentes en los datos.

En este trabajo, se propone un enfoque basado en Transformers, utilizando un modelo inspirado en BERT, modificado para series de tiempo. Mientras que los Generative Adversarial Networks (GANs) han mostrado resultados prometedores en tareas similares, los Transformers ofrecen ventajas significativas en la captura de dependencias a largo plazo y relaciones bidireccionales dentro de las series temporales. Estas capacidades permiten al modelo reconstruir valores faltantes o enmascarados utilizando información tanto previa como posterior en el tiempo, mejorando la coherencia y precisión de las predicciones.

La metodología presentada entrena un modelo BERT modificado para que, mediante un esquema de enmascaramiento aleatorio, aprenda a predecir valores faltantes en series de baja resolución. Este enfoque aprovecha la atención bidireccional para capturar patrones globales y locales en los datos, asegurando que las predicciones sean consistentes con el contexto completo de la serie.

El objetivo principal de este trabajo es demostrar la eficacia de los Transformers para la reconstrucción de series temporales de baja resolución, evaluando su desempeño en tareas de super-resolución. Los resultados obtenidos validan el potencial de este enfoque para aplicaciones prácticas en una amplia variedad de dominios, desde monitoreo industrial hasta predicción climática y análisis financiero.

II. Estado del arte

A. El Transformer: Bases, Funcionamiento y Aplicaciones

El artículo *"Attention is All You Need"* presenta una nueva arquitectura de red neuronal llamada **Transformer** [1], la cual se basa completamente en mecanismos de

atención, eliminando la necesidad de redes recurrentes o convolucionales.

Arquitectura del Transformer: El Transformer sigue una estructura típica de codificador-decodificador presente en la mayoría de los modelos de transducción de secuencias neuronales más competitivos.

- **Codificador:** Mapea una secuencia de entrada de representaciones de símbolos $(x_1, \ldots, x_n)$ a una secuencia de representaciones continuas $z = (z_1, \ldots, z_n)$.
- **Decodificador:** Genera una secuencia de salida $(y_1, \ldots, y_m)$ de símbolos uno a la vez, basándose en la secuencia codificada z. El modelo es *autoregresivo*, es decir, utiliza los símbolos generados previamente como entrada adicional al generar el siguiente.

El Transformer implementa esta arquitectura utilizando capas apiladas de autoatención y capas completamente conectadas para el codificador y el decodificador, como se muestra en la Figura 1 del artículo.

Autoatención: El Corazón del Transformer: La **autoatención**, también conocida como atención intra-secuencia, permite que el modelo relacione diferentes posiciones dentro de una misma secuencia para calcular una representación de la secuencia.

Tipos de Atención: El Transformer utiliza la atención multi-cabeza de tres maneras:

1) **Atención codificador-decodificador:** Las consultas provienen de la capa anterior del decodificador, y las claves y valores de memoria provienen de la salida del codificador. Esto permite que cada posición en el decodificador atienda a todas las posiciones de la secuencia de entrada.
2) **Autoatención en el codificador:** Todas las claves, valores y consultas provienen del mismo lugar, en este caso, la salida de la capa anterior en el codificador. Cada posición en el codificador puede atender a todas las posiciones en la capa anterior del codificador.
3) **Autoatención en el decodificador:** Similar a la autoatención en el codificador, pero con una máscara que impide que las posiciones atiendan a posiciones futuras, para preservar la propiedad autoregresiva.

¿Por qué Autoatención?: Las capas de autoatención ofrecen ventajas sobre las capas recurrentes y convolucionales en tres aspectos clave:

1) **Complejidad computacional:** Las capas de autoatención conectan todas las posiciones con un número constante de operaciones ejecutadas secuencialmente, mientras que las capas recurrentes requieren $O(n)$ operaciones secuenciales.
2) **Dependencias a larga distancia:** La autoatención puede modelar dependencias entre posiciones distantes dentro de una secuencia sin un aumento en el costo computacional, a diferencia de las capas recurrentes y convolucionales.
3) **Interpretabilidad:** La autoatención puede proporcionar información sobre qué partes de la secuencia de entrada son más relevantes para cada posición de salida, lo que puede ayudar a comprender mejor el funcionamiento del modelo.

Atención Escalada de Producto Punto: El Transformer utiliza una función de atención llamada *Atención escalada de producto punto*. La entrada consiste en:

- Consultas (Q)
- Claves (K)
- Valores (V)

La salida se calcula como una suma ponderada de los valores, donde el peso asignado a cada valor se determina por la compatibilidad entre la consulta y la clave correspondiente.

Fórmula de la atención escalada de producto punto:

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

donde:

- d_k es la dimensión de las claves.
- El factor de escala $1/\sqrt{d_k}$ evita que los productos punto se vuelvan demasiado grandes para valores grandes de d_k.

Atención Multi-Cabeza: En lugar de utilizar una sola función de atención, el Transformer emplea la atención multi-cabeza, que realiza la función de atención en paralelo en múltiples "cabezas".

Ventajas de la atención multi-cabeza:

- Permite que el modelo atienda conjuntamente a información de diferentes subespacios de representación en diferentes posiciones.
- Aumenta la capacidad del modelo para capturar diferentes tipos de relaciones entre las posiciones de la secuencia.

El Transformer se basa completamente en la autoatención para modelar las relaciones entre las posiciones dentro de una secuencia. La autoatención ofrece ventajas sobre las capas recurrentes y convolucionales en términos de complejidad computacional, capacidad para modelar dependencias a larga distancia e interpretabilidad. El Transformer ha logrado resultados de vanguardia en diversas tareas de PNL y a continuación se abordará cómo se adaptan a problemas relacionados con series de tiempo.

B. Transformers en Series de Tiempo

Los Transformers, inicialmente diseñados para el Procesamiento del Lenguaje Natural (PNL), también se pueden aplicar al análisis de series de tiempo. A pesar de que los Transformers carecen de recurrencia, se les puede dotar de la capacidad de procesar información secuencial mediante la **codificación posicional**. Esta técnica consiste en incorporar vectores que representan la posición de cada elemento en la secuencia de entrada.

Para su aplicación en series de tiempo, se han realizado diversas modificaciones a la arquitectura del Transformer original [2]:

- **Codificación Posicional:** A diferencia de las redes LSTM o RNN, el Transformer original no tiene recurrencia. Para modelar la información secuencial, se añade una codificación posicional a las instrucciones de entrada. Existen tres enfoques principales para la codificación posicional en series de tiempo:
  - **Vanilla Encoding:** Algunos trabajos simplemente utilizan la codificación posicional original del Transformer, que se suma a las incrustaciones de las series de tiempo de entrada y se introduce en el Transformer. Sin embargo, este método no aprovecha al máximo las características de las series de tiempo.
  - **Learnable Encoding:** Se ha demostrado que aprender las incrustaciones posicionales a partir de los datos de las series de tiempo puede ser mucho más efectivo. Estas incrustaciones son más flexibles y se adaptan mejor a las tareas específicas.
  - **Timestamp Encoding:** En escenarios reales, a menudo se dispone de información sobre la marca temporal, como marcas de tiempo de calendario y marcas especiales para eventos. Esta información, que no se utiliza en los Transformers originales, es muy útil en aplicaciones reales. Para abordar este problema, se han propuesto métodos que codifican las marcas temporales como codificación posicional mediante capas de incrustación aprendibles.
- **Módulo de Atención:** El módulo de autoatención es fundamental en los Transformers. Sin embargo, su complejidad temporal y de memoria $O(N^2)$ (donde N es la longitud de la serie de tiempo) puede ser un problema al trabajar con secuencias largas. Para mejorar la eficiencia, se han experimentado estrategias como:
  - Introducir un sesgo de dispersión en el mecanismo de atención, como en LogTrans y Pyraformer.
  - Aprovechar la propiedad de bajo rango de la matriz de autoatención para acelerar el cálculo, como en Informer y FEDformer.
- **Innovación de la Atención Basada en la Arquitectura:** Diversos estudios han propuesto la introducción de una arquitectura jerárquica en el Transformer para tener en cuenta la naturaleza multirresolución de las series de tiempo. Por ejemplo, Informer introduce capas de "max-pooling" entre los bloques de atención para reducir la resolución de las series, mientras que Pyraformer utiliza un mecanismo de atención basado en un árbol C-ary para capturar dependencias temporales en diferentes resoluciones.

La super-resolución de series temporales ha sido un campo activo de investigación en el procesamiento de señales y aprendizaje profundo, con aplicaciones en audio, genómica y otros dominios [3]. Los métodos convencionales incluyen enfoques basados en interpolación como lineal y cúbica, los cuales presentan limitaciones al modelar dinámicas complejas y dependencias temporales no lineales.

C. Imputación de valores faltantes en Series de Tiempo

Uno de los desafíos clave en el análisis de series de tiempo es la imputación de valores faltantes, un problema que surge con frecuencia debido a fallos, errores de sensores o recopilación de datos incompleta (Figura 1). [4]

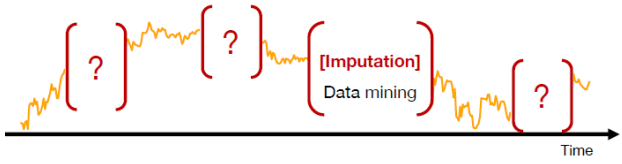


Figure 1: Valores faltantes en Serie temporal.

La imputación de series de tiempo puede considerarse como un problema de superresolución, donde el objetivo es reconstruir una serie de tiempo de alta resolución a partir de mediciones de baja resolución que contienen valores faltantes.

Este enfoque enmarca la imputación como un problema de mejora de la resolución, similar a la superresolución de imágenes, donde se reconstruye una imagen de alta resolución a partir de una de baja resolución.

Otros estudios se han centrado principalmente en modelos profundos para diversas tareas de análisis de series de tiempo, incluida la previsión, la clasificación, la imputación y la detección de anomalías. Si bien no se centran explícitamente en la imputación como superresolución, proporcionan información valiosa sobre técnicas y arquitecturas de modelos profundos relevantes para esta tarea.
Modelos de aprendizaje profundo para la imputación: Varios modelos de aprendizaje profundo se han probado para la imputación de series de tiempo, incluidas las redes neuronales recurrentes (RNN), las redes neuronales convolucionales (CNN) y los Transformers. Estos modelos pueden aprender dependencias temporales complejas y correlaciones multivariadas, lo que los hace adecuados para la imputación de valores faltantes.

- **RNNs:** Yu *et al.* [5] presentan un enfoque basado en RNN bidireccionales para la reconstrucción de series de tiempo, utilizando un esquema codificador-decodificador. Los RNN, como las redes de memoria a corto plazo (LSTM), son adecuados para modelar datos secuenciales debido a su capacidad para capturar dependencias a largo plazo. Para la imputación, los RNN pueden aprender los patrones temporales de los datos observados y usarlos para predecir los valores faltantes.
- **CNNs:** Las CNN, conocidas por su eficacia en el procesamiento de imágenes, también se han aplicado con éxito al análisis de series de tiempo. Las

CNN pueden capturar patrones locales en los datos utilizando operaciones de convolución y aprender representaciones jerárquicas de la serie de tiempo.

- **Transformers:** Los Transformers, inicialmente diseñados para el procesamiento del lenguaje natural, han surgido como una arquitectura poderosa para el análisis de series de tiempo. El mecanismo de autoatención de los Transformers les permite capturar dependencias a largo plazo y correlaciones multivariadas de manera eficaz, lo que los hace particularmente prometedores para la imputación de series de tiempo. [6]

Enfoques de superresolución: Varios mecanismos y técnicas de atención se pueden considerar para mejorar la capacidad de los modelos profundos para capturar dependencias temporales y correlaciones multivariadas. Estos avances son relevantes para abordar los estudios de imputación como superresolución.

- **Atención dispersa:** Los modelos como LogTrans e Informer introducen sesgos de dispersión en el mecanismo de atención, lo que permite a los modelos centrarse en los puntos de datos más relevantes y reducir la complejidad computacional al tiempo que capturan dependencias a largo plazo.
- **Atención jerárquica:** Los modelos como Pyraformer emplean arquitecturas jerárquicas para capturar dependencias temporales en múltiples resoluciones. Este enfoque permite que los modelos aprendan patrones a corto y largo plazo de manera eficaz.
- **Atención en el dominio de frecuencia:** Los modelos como FEDformer operan en el dominio de frecuencia, lo que permite capturar relaciones temporales complejas y modelar patrones periódicos de manera eficaz.

Consideraciones para la imputación de superresolución: Al considerar la imputación de series de tiempo como superresolución, es esencial tener en cuenta los siguientes factores:

- **Naturaleza de los valores faltantes:** Comprender el mecanismo subyacente a los valores faltantes (por ejemplo, aleatorio, periódico o dependiente del valor) puede guiar la elección de la arquitectura del modelo y las técnicas de preprocesamiento.
- **Longitud de las brechas:** Los modelos deben diseñarse para manejar la longitud variable de las brechas en los datos, utilizando mecanismos de atención apropiados o técnicas de descomposición para capturar dependencias a largo plazo.
- **Información de la marca de tiempo:** La incorporación de información de la marca de tiempo, como los calendarios o los eventos, puede mejorar la precisión del modelo, especialmente cuando los valores faltantes dependen del tiempo.

Enmarcar la imputación de series de tiempo como superresolución ofrece una nueva perspectiva sobre este problema desafiante. Los avances en modelos profundos, particularmente en arquitecturas y técnicas de atención, proporcionan una base sólida para desarrollar métodos de imputación eficaces y robustos. Al aprovechar estos avances y explorar futuras direcciones de investigación, podemos mejorar significativamente nuestra capacidad para manejar los valores faltantes en los datos de series de tiempo, lo que permite análisis más precisos y una mejor toma de decisiones en diversas aplicaciones del mundo real.

A diferencia de estos trabajos, el presente enfoque utiliza una modificación del modelo BERT, originalmente diseñado para procesamiento de lenguaje natural, adaptado para series temporales. Nuestro modelo se diferencia por su capacidad para reconstruir valores enmascarados de manera bidireccional, aprovechando tanto las dependencias hacia el pasado como hacia el futuro. Esto se logra a través de un esquema de enmascaramiento adaptado para series temporales y una arquitectura optimizada para manejar dominios de datos con resolución variable.

III. Materiales y Metodología

A. Series de Tiempo sintéticas

Para este estudio, se generaron datos sintéticos representativos de series de tiempo con patrones diversos y diseñados para simular escenarios del mundo real. Estas series de tiempo presentan variabilidad tanto en amplitud como en frecuencia, lo que crea un conjunto de datos desafiante para evaluar la capacidad de adaptación del modelo propuesto.
Generación de Datos Sintéticos: Se crearon 1000 series de tiempo, cada una compuesta por 5000 puntos, representando una alta resolución temporal (Imagen 2). Estas series sirvieron como base para el entrenamiento del modelo. Para evaluar el rendimiento, se emplearon las mismas series, pero a una resolución reducida de 250 puntos por serie, simulando escenarios de baja calidad de muestreo.

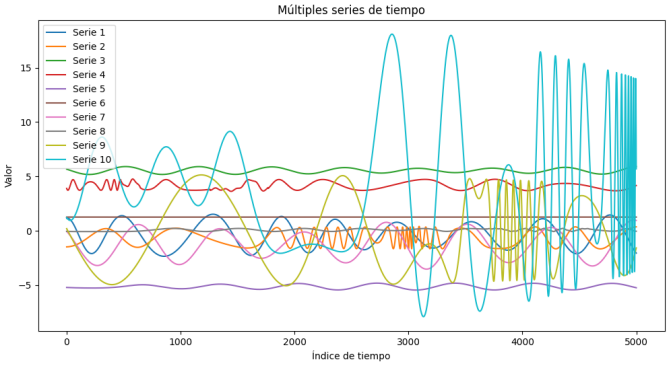
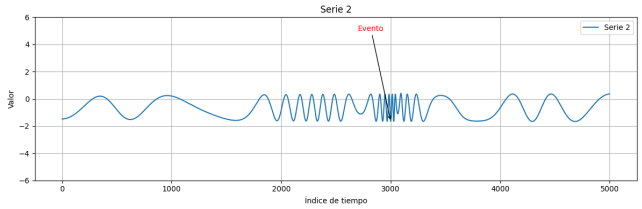
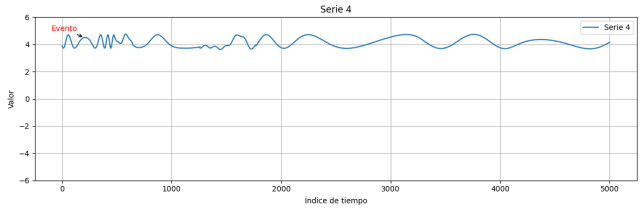


Figure 2: Series de Tiempo Sintéticas.

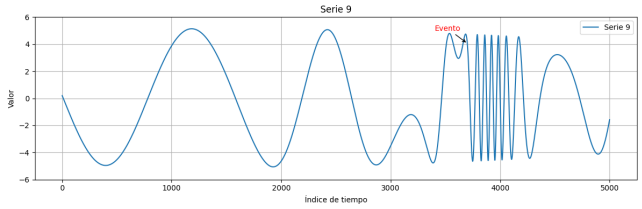
Variabilidad de los Datos: Los datos generados contienen una variación significativa en las frecuencias y amplitudes, lo que permite evaluar la robustez del modelo al trabajar con datos de características diversas y complejas.
Eventos: Las series de tiempo sintéticas utilizadas en este trabajo presentan eventos de corta duración y alta variabilidad (Figura 3). Estos eventos fueron generados intencionalmente para que el modelo Transformer pueda aprender a detectar patrones anómalos y posteriormente predecirlos.



(a) Figura 3A



(b) Figura 3B



(c) Figura 3C

Figure 3: Eventos en Series de Tiempo.

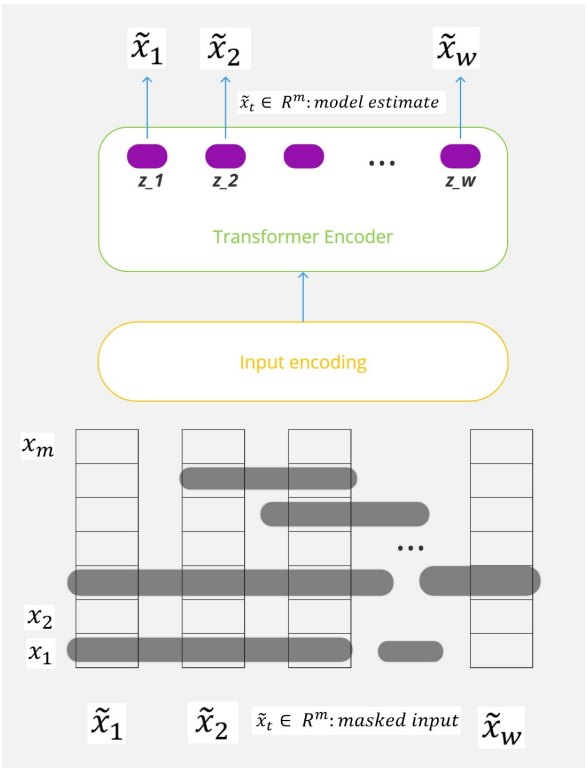


Figure 4: Esquema Transformer BERT. [6]

En una aplicación práctica, estos eventos podrían corresponder a situaciones críticas como fallos en motores industriales, fluctuaciones abruptas en señales financieras, anomalías en sistemas eléctricos o cambios repentinos en procesos físicos. La capacidad del Transformer para identificar estos eventos es fundamental en tareas de mantenimiento predictivo, monitoreo de sistemas complejos y detección temprana de anomalías.

El aprendizaje de estos eventos en un entorno sintético permite validar la efectividad del modelo en la detección de patrones disruptivos y no lineales, garantizando su aplicabilidad en contextos reales con datos ruidosos y complejos.

Arquitectura basada en BERT: El modelo propuesto está basado en el Transformer BERT, adaptado específicamente para series de tiempo. Se utilizó un enfoque de super-resolución temporal que emplea enmascaramiento y predicción de valores faltantes para reconstruir puntos faltantes en las series de tiempo de baja resolución.

B. Descripción de la Arquitectura del Transformer

El modelo **TimeSeriesBERT** propuesto se basa en una arquitectura tipo Transformer modificada para abordar el problema de la super-resolución en series temporales. La arquitectura general del modelo consta de los siguientes componentes principales:

1) **Preprocesamiento de Entrada - Enmascaramiento Aleatorio:**
Antes de alimentar la serie temporal al modelo, una fracción r de los valores de entrada se enmascaran aleatoriamente reemplazándolos por ceros. Esto obliga al modelo a aprender a predecir los valores faltantes en función de los valores no enmascarados y de las dependencias temporales. El proceso se realiza con una máscara binaria M, donde:

$$x_{\text{masked}} = x \odot M \quad \text{con} \quad M_{t,i} \in \{0, 1\}.$$

Aquí, $\odot$ representa la multiplicación elemento a elemento y M es una matriz binaria aleatoria.

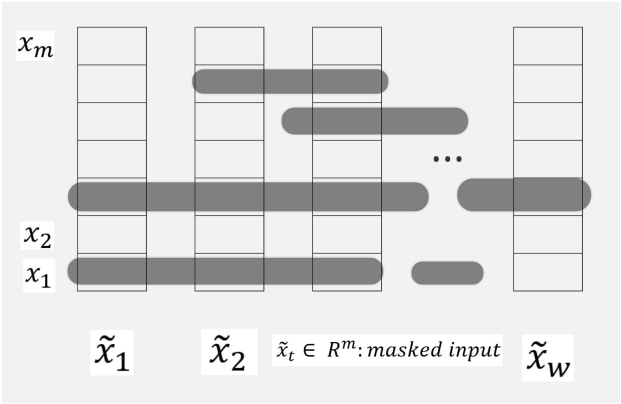


Figure 5: Masking Series de **entrada**

2) **Codificación de Entrada (InputEncoding):**
Cada serie temporal enmascarada se proyecta a un espacio de mayor dimensión mediante una capa lineal de proyección. Además, se incorpora una codificación

posicional sinusoidal para preservar la información temporal de la secuencia. La salida de esta etapa tiene la siguiente forma:

$$x' = \text{InputEncoding}(x_{\text{masked}}) + \text{PositionalEncoding}.$$

donde x_{masked} es la serie enmascarada.

3) **Pila de Encoders (TransformerEncoder):**
La arquitectura del modelo incluye una pila de **N bloques de Encoder**. Cada bloque está compuesto por:
   - **Atención Multi-Cabezal (Multi-Head Attention):** Permite al modelo capturar dependencias a corto y largo plazo entre diferentes puntos de la serie temporal. La salida se calcula como:

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

     donde Q, K, y V son las proyecciones de la entrada usando matrices de parámetros aprendibles.
   - **Conexiones Residuales y Normalización por Capas:** Facilitan el flujo de gradientes durante el entrenamiento y estabilizan el aprendizaje.
   - **Red Feed-Forward:** Cada punto de la secuencia es procesado independientemente a través de una red feed-forward compuesta por dos capas lineales y una activación ReLU intermedia.

   Este proceso puede resumirse de la siguiente manera:

$$x'' = \text{LayerNorm}(x' + \text{Dropout}(\text{MultiHeadAttention}(x'))).$$

$$x_{\text{out}} = \text{LayerNorm}(x'' + \text{Dropout}(\text{FeedForward}(x''))).$$

   donde x_{out} representa la salida de cada bloque de Encoder.

4) **Capa de Proyección de Salida:**
Finalmente, la salida de la pila de encoders se proyecta de regreso al espacio original de la serie temporal mediante una capa lineal. Esto permite reconstruir los valores faltantes en la serie temporal original. La predicción final está dada por:

$$\hat{x} = \text{OutputProjection}(x_{\text{out}}).$$

5) **Función de Pérdida (Masked MSE):**
Durante el entrenamiento, el modelo compara únicamente las predicciones en las posiciones enmascaradas con sus valores reales usando la pérdida de error cuadrático medio (MSE):

$$\mathcal{L}_{\text{MSE}} = \frac{1}{|M|}\sum_{(t,i)\in M}\left(\hat{x}(t,i) - x(t,i)\right)^2.$$

donde M representa los índices de los valores enmascarados, $\hat{x}(t,i)$ son las predicciones del modelo y $x(t,i)$ son los valores reales.

C. Parámetros de entrenamiento

En la presente experimentación, se evaluaron dos configuraciones del modelo TimeSeriesBERT con el objetivo de determinar el impacto de los parámetros de la arquitectura en el desempeño del modelo. Ambos experimentos fueron entrenados utilizando una **TPU v2-8** para acelerar el procesamiento y manejo eficiente de las series temporales.

El uso de la TPU permitió realizar operaciones de atención multi-cabezal y optimización de manera eficiente, lo cual es crucial dado el tamaño de las series temporales y la complejidad del modelo.

La comparación entre ambas configuraciones (Tabla 1) tiene como objetivo analizar el impacto de la dimensionalidad proyectada (`model_dim`) y el número de cabezas de atención (`num_heads`) en la calidad de la reconstrucción de las series temporales.

Table I: Parámetros del Modelo

Parámetro	Configuración 1	Configuración 2
seq_len	5000	5000
input_dim	1	1
model_dim	32	64
num_layers	4	4
num_heads	4	8
ff_dim	128	256
Dropout	0.1	0.1
num_epochs	10	10

D. Reconstrucción de Series de Tiempo

Una vez entrenado el modelo TimeSeriesBERT, la reconstrucción de series de tiempo de baja resolución (250 puntos) a una resolución más alta (5000 puntos) se realiza mediante un proceso estructurado en tres etapas:

label=**0.**

1) **Interpolación Inicial de la Serie de Tiempo:**
   - Cada serie de baja resolución se interpola linealmente para ajustarse a la longitud deseada de 5000 puntos. Esto proporciona una primera aproximación que sirve como entrada al modelo.
   - La interpolación se realiza asegurando que la serie de baja resolución se expanda uniformemente a lo largo del dominio temporal.
2) **Entrada al Modelo:**
   - La serie interpolada se convierte en un tensor de PyTorch con la forma $(1, 5000, 1)$, donde:
     - **1**: Representa el tamaño del lote (batch size).
     - **5000**: Representa la longitud de la serie expandida.
     - **1**: Representa la dimensión de cada punto en la serie (escalares en este caso).
   - Se asegura que el tensor tenga la misma forma esperada por el modelo.
3) **Predicción con el Modelo:**

- El tensor interpolado se pasa al modelo entrenado en **modo evaluación** (model.eval()).
- El modelo procesa la serie interpolada y realiza ajustes a los valores para predecir una reconstrucción de la serie en alta resolución.
- Finalmente, la salida del modelo se convierte de nuevo a un array de NumPy para su análisis y visualización.

4) **Visualización de la Reconstrucción:**
   - Se grafica la serie original de baja resolución junto con la serie reconstruida para evaluar visualmente el desempeño del modelo.
   - La serie original se visualiza con 250 puntos y se superpone con la reconstrucción de 5000 puntos, mostrando cómo el modelo mejora la resolución temporal al predecir valores adicionales.

IV. Resultados

A. Modelo 1

Para la primera reconstrucción de series de tiempo, se utilizó la Configuración 1 del modelo Transformer, como se describe en la Tabla I. Esta configuración cuenta con una dimensión proyectada (model_dim) de 32, 4 capas de Encoder (num_layers) y 4 cabezas de atención (num_heads), con una tasa de Dropout del 10%. El modelo fue entrenado durante 10 épocas utilizando una TPU v2-8 para acelerar el proceso de entrenamiento.

En la figura 5, se observa el resultado de reconstruir una serie de tiempo de baja resolución (250 puntos) a alta resolución (5000 puntos). La serie original, representada en azul, muestra variaciones significativas tanto en amplitud como en frecuencia, lo cual presenta un desafío considerable para los modelos de super-resolución.

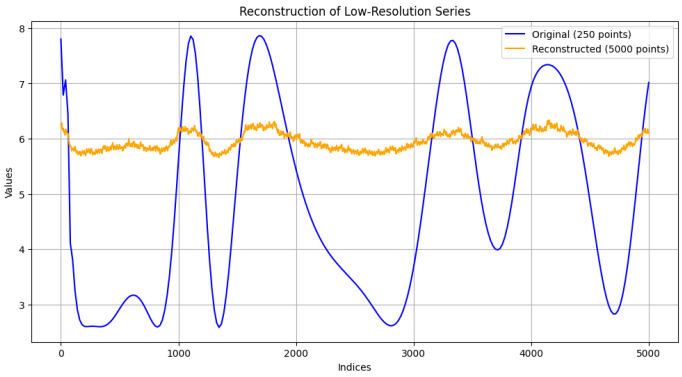


Figure 6: Reconstrucción 1

La serie reconstruida por el modelo, mostrada en naranja, logra capturar de manera satisfactoria los patrones globales presentes en la serie original, reflejando de manera consistente las tendencias y oscilaciones predominantes en los datos. A pesar de la reducción inicial de información en la serie original de baja resolución, el modelo ha logrado reconstruir una serie con un nivel de detalle considerable, suavizando de manera coherente las variaciones a lo largo del tiempo.

Sin embargo, aún existe espacio para mejorar. Aunque el modelo capta de manera efectiva los patrones generales de la serie original, no logra superponerse perfectamente, especialmente en términos de amplitud. Este aspecto es particularmente visible en las regiones donde la serie original presenta grandes oscilaciones, lo que indica la necesidad de ajustar la capacidad del modelo para aprender de manera más precisa los valores extremos y la magnitud de las variaciones.

Este resultado demuestra la capacidad del Transformer para aprender las relaciones temporales de largo plazo dentro de las series de tiempo, pero también resalta la necesidad de optimizar su desempeño para lograr una reconstrucción más precisa que reproduzca no solo la frecuencia, sino también la amplitud de la serie original.

B. Modelo 2

Para la segunda reconstrucción de series de tiempo, se utilizó la Configuración 2 del modelo Transformer, como se describe en la Tabla I. Esta configuración incrementa la dimensión proyectada del modelo (model_dim) de 32 a 64, el número de cabezas de atención (num_heads) de 4 a 8, y el feed forward de 128 a 256, manteniendo el resto de los parámetros constantes, como 4 capas de Encoder, una tasa de Dropout del 10% y 10 épocas de entrenamiento.

En la Figura 6 se observa la comparación entre la serie original de baja resolución (250 puntos) y la serie reconstruida (5000 puntos). El modelo continúa capturando los patrones globales de la serie original, evidenciando una mejora significativa en la reconstrucción de la amplitud de las oscilaciones. A diferencia de la primera reconstrucción, en esta versión, las oscilaciones reconstruidas se aproximan con mayor precisión a la magnitud de las variaciones presentes en la serie original, especialmente en los picos y valles.

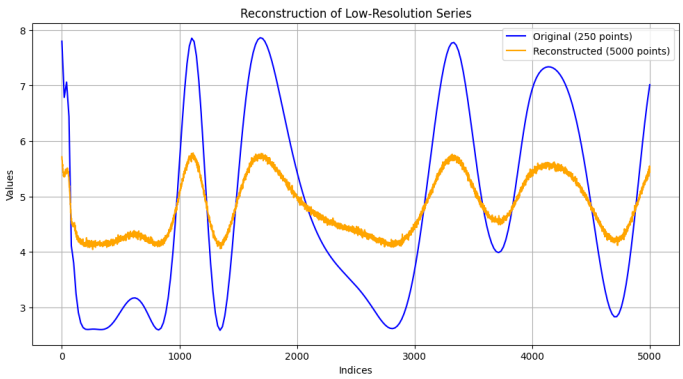


Figure 7: Reconstrucción 2

Esta mejora en la amplitud se debe, principalmente, al aumento de la capacidad del modelo al incrementar la dimensión de proyección y el número de cabezas de atención. Estas modificaciones permiten al Transformer modelar con

mayor precisión las relaciones temporales a largo plazo y ajustar mejor la escala de los valores reconstruidos.

No obstante, aún persisten ligeras diferencias en la reconstrucción detallada de algunas regiones, donde la serie reconstruida no logra superponerse perfectamente a la serie original. Sin embargo, la tendencia general y la amplitud de las oscilaciones se han representado de forma más efectiva, lo que refleja un avance significativo respecto a la configuración anterior.

Esta segunda configuración demuestra que incrementar la capacidad del modelo mejora su habilidad para reconstruir tanto los patrones globales como la amplitud de las series temporales originales.

C. Parámetros y Rendimiento

Table II: Parámetros del Modelo y Uso de Memoria

Parámetro	**Configuración 1**	**Configuración 2**
seq_len	5000	5000
input_dim	1	1
model_dim	32	64
num_layers	4	4
num_heads	4	8
ff_dim	128	256
Dropout	0.1	0.1
num_epochs	10	10
Uso de RAM (GB)	40	190
Tiempo (horas)	5h 39m	12h 6m

Nota: Ambos modelos utilizaron TPU v2-8 de Google Colab.

En la Configuración 1, el modelo utilizó un model_dim de 32 y 4 cabezas de atención, logrando un consumo de 40 GB de memoria RAM y un tiempo de entrenamiento de 5 horas y 39 minutos. Los resultados de esta configuración muestran que el modelo fue capaz de captar los patrones globales de las series originales, pero con limitaciones al momento de reconstruir la amplitud.

Para mejorar la reconstrucción, especialmente en términos de amplitud, se probó la Configuración 2, donde el model_dim se incrementó a 64 y el número de cabezas de atención a 8. Esta configuración mostró una mejora en la precisión de los resultados, logrando una mejor superposición con las series originales. Sin embargo, este incremento en los parámetros del modelo también produjo un aumento significativo en el uso de recursos computacionales, alcanzando 190 GB de RAM y requiriendo 12 horas y 6 minutos de tiempo de ejecución.

A pesar de los avances obtenidos con la Configuración 2, no fue posible explorar configuraciones aún más complejas debido a las limitaciones en la disponibilidad de recursos computacionales. El uso de memoria RAM se disparó considerablemente, y la infraestructura disponible no permitía escalar el modelo más allá de estos límites.

D. Detección de eventos

En la reconstrucción que se muestra en la Figura 8, se observa que el modelo Transformer no solo logra captar los patrones globales de la serie de tiempo original, sino que también detecta los eventos de alta variabilidad y corta duración. Estos eventos, ubicados principalmente en las regiones con oscilaciones rápidas y cambios abruptos, son identificados por el modelo a pesar de no coincidir exactamente en amplitud.

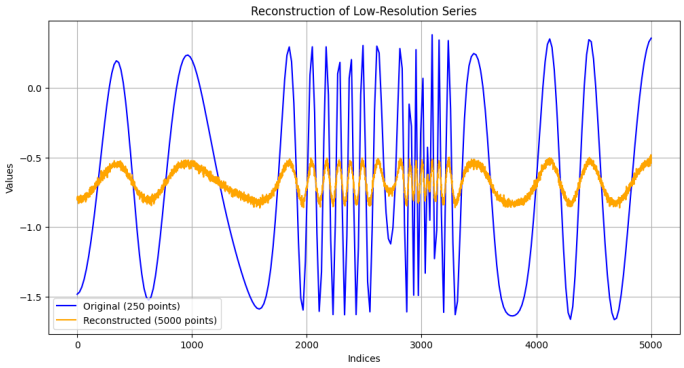


Figure 8: Detección de eventos

La reconstrucción muestra que el modelo aún suaviza los eventos, lo que sugiere que su capacidad de resolución espacial aún debe mejorarse. No obstante, su habilidad para identificar la ubicación y ocurrencia de los eventos representa un avance significativo. En una aplicación real, la detección de estos eventos podría interpretarse como fallas o anomalías críticas, como en el caso de fallos en motores, sistemas de control u otros escenarios industriales.

V. Conclusion

En este trabajo, se ha propuesto una arquitectura basada en Transformers, inspirada en el modelo BERT, para la tarea de super-resolución de series temporales. La metodología implementada demostró ser efectiva en la reconstrucción de datos de alta resolución a partir de versiones de baja resolución, capturando los patrones globales y locales presentes en las series temporales.

Los resultados obtenidos con las dos configuraciones evaluadas muestran que el modelo es capaz de identificar las variaciones en frecuencia y amplitud de las series originales, destacándose especialmente en la segunda configuración, donde el número de parámetros fue incrementado. Este ajuste permitió una mejora significativa en la reconstrucción de la amplitud, aunque con un costo considerable en recursos computacionales, alcanzando un uso máximo de 190 GB de RAM y un tiempo de entrenamiento de 12 horas.

A pesar de las mejoras logradas, se observó que el modelo aún presenta limitaciones en la superposición precisa de la serie reconstruida con la serie original, especialmente en regiones con alta variabilidad. Este aspecto podría ser abordado en trabajos futuros mediante técnicas avanzadas de regularización, estrategias de atención más eficientes, o utilizando arquitecturas híbridas que combinen Transformers con otras técnicas como redes recurrentes o convolucionales.

Finalmente, se destaca la escalabilidad del modelo y su capacidad para trabajar con datos sintéticos de alta complejidad. Sin embargo, el alto consumo de memoria y tiempo de entrenamiento representa un desafío importante. Para resolverlo, futuras investigaciones podrían enfocarse en optimizar la eficiencia computacional del modelo, implementando versiones más ligeras del Transformer o utilizando técnicas de cuantización y *sparse atention*.

Este trabajo sienta las bases para el uso de Transformers en la super-resolución de series temporales, mostrando su potencial para tareas complejas y su aplicabilidad en escenarios donde la resolución de datos es crítica para el análisis temporal.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in \textit{Advances in Neural Information Processing Systems}, vol. 30, pp. 5998–6008, 2017.

[2] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun, "Transformers in time series: A survey," in *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI)*, 2023, pp. 1–9. Available: https://doi.org/10.48550/arXiv.2202.07125.

[3] V. Kuleshov, S. Birnbaum, Z. Enam, and P. W. Koh, "Time Series Super Resolution with Temporal Adaptive Batch Normalization," in \textit{Proceedings}, 2018. Available: \url{https://api.semanticscholar.org/CorpusID:53474966}.

[4] Y. Wang, H. Wu, J. Dong, Y. Liu, M. Long, and J. Wang, "Deep Time Series Models: A Comprehensive Survey and Benchmark," arXiv preprint arXiv:2407.13278, 2024. Available: https://doi.org/10.48550/arXiv.2407.13278.

[5] [1] W. Yu, C. Mechefske, and I. Y. Kim, "Time Series Reconstruction Using a Bidirectional Recurrent Neural Network Based Encoder-Decoder Scheme," 2019.

[6] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A Transformer-based Framework for Multivariate Time Series Representation Learning," *arXiv preprint arXiv:2010.02803*, 2020. Available: https://doi.org/10.48550/arXiv.2010.02803