UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Posgrados

Reconstrucción de datos basada en Redes Neuronales Generativas Adversarias (GANs) en una empresa de retail de productos plásticos

Proyecto de Titulación

Addis Scarlett Abalco Dias

Julio Ibarra Fiallo

Director de Trabajo de Titulación

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

Quito, 02 de diciembre de 2024

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ
COLEGIO DE POSGRADOS

HOJA DE APROBACIÓN DE TRABAJO DE TITULACIÓN

Reconstrucción de datos basada en Redes Neuronales Generativas Adversarias (GANs) en una empresa de retail de productos plásticos

Addis Scarlett Abalco Dias

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

Nombre del estudiante: Addis Scarlett Abalco Dias

Código de estudiante: 00339323

C.I.: 1720136009

Lugar y fecha: Quito, 02 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UNPUBLISHED DOCUMENT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.

DEDICATORIA

A la vida, al amor, a mi yo del pasado por no rendirse y a todos esos seres humanos que se sintieron perdidos alguna vez como yo y que a pesar de todo nunca dejaron de intentarlo

AGRADECIMIENTOS

A mi madre Inés por su guía con infinito amor, sabiduría y ser mi fuente de fortaleza e inspiración, a mi padre Ernesto por su amor incondicional y ser uno de mis héroes favoritos, a mi hermana Johanna por ser mi cómplice de aventuras, a mi querido tutor Julio Ibarra por compartir su pasión por la matemática e inspirarnos a ser mejores seres humanos con su extraordinaria forma de ser y ver el mundo, a mis amigos y amigas por su cariño y compañía en esta hermosa aventura llamada vida, a Carito por creer en mi más de lo que yo misma he creido y ser uno de los seres humanos más hermosos que he conocido, a mi jefe Fausto por su invaluable comprensión y apoyo en este proceso, a Julio Pico por creer en esta idea y permitir crear un puente entre el ámbito académico y empresarial.

RESUMEN

El presente trabajo se enfoca en la aplicación de las redes neuronales generativas adversarias (GANs) para la interpolación de datos en una empresa de retail de plásticos que cuenta con un portafolio de aproximadamente 800 productos, se utilizaron datos diarios desde el 22 de enero del 2021 hasta el 22 de octubre del 2024, la empresa ha tenido problemas con datos faltantes debido a problemas en el sistema para el ingreso de información de la cantidad vendida, olvido del personal para ingresar cada venta, cortes energéticos. Luego de realizar varios experimentos, la arquitectura que tuvo resultados más efectivos fue la utilizó un generador y un discriminador basados en redes Long Short – Term Memory (LSTM), que tienen como objetivo principal capturar las dependencias temporales. De manera general, el generador emplea ruido y contexto previo para generar secuencias plausibles, mientras que el discriminador evalúa la autenticidad de las secuencias generadas. La capacidad de las GANs para aprender distribuciones de datos complejas mediante el proceso adversarial entre el generador y el discriminador representa una ventaja para las reconstrucciones de las series temporales en dominios donde los datos están incompletos, como es el caso de lo que requiere esta empresa.

Palabras clave: GANs, LSTM, empresa de plásticos, reconstrucción, interpolación, series temporales incompletas

ABSTRACT

This study focuses on the application of Generative Adversarial Networks (GANs) for data interpolation in a plastic retail company with a portfolio of approximately 800 products. Daily data from January 22, 2021, to October 22, 2024, were used. The company has faced issues with missing data due to system failures in recording sales information, staff oversight in entering sales, and power outages. After conducting multiple experiments, the architecture that yielded the most effective results employed a generator and a discriminator based on Long Short-Term Memory (LSTM) networks, whose primary objective is to capture temporal dependencies. In general, the generator uses noise and prior context to generate plausible sequences, while the discriminator evaluates the authenticity of the generated sequences. The ability of GANs to learn complex data distributions through the adversarial process between the generator and the discriminator represents a significant advantage for reconstructing time series in domains with incomplete data, such as the requirements of this company.

Key words: GANs, LSTM, plastic company, reconstruction, interpolation, incomplete time series

TABLA DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

Reconstrucción de datos basada en Redes Neuronales Generativas Adversarias (GANs) en una empresa de retail de productos plásticos

Addis Scarlett Abalco Dias, Julio Ibarra Fiallo

***Abstract*—El presente trabajo se enfoca en la aplicación de las redes neuronales generativas adversarias (GANs) para la interpolación de datos en una empresa de retail de plásticos que cuenta con un portafolio de aproximadamente 800 productos, se utilizaron datos diarios desde el 22 de enero del 2021 hasta el 22 de octubre del 2024, la empresa ha tenido problemas con datos faltantes debido a problemas en el sistema para el ingreso de información de la cantidad vendida, olvido del personal para ingresar cada venta, cortes energéticos. Luego de realizar varios experimentos, la arquitectura que tuvo resultados más efectivos fue la utilizó un generador y un discriminador basados en redes Long Short – Term Memory (LSTM), que tienen como objetivo principal capturar las dependencias temporales. De manera general, el generador emplea ruido y contexto previo para generar secuencias plausibles, mientras que el discriminador evalúa la autenticidad de las secuencias generadas. La capacidad de las GANs para aprender distribuciones de datos complejas mediante el proceso adversarial entre el generador y el discriminador representa una ventaja para las reconstrucciones de las series temporales en dominios donde los datos están incompletos, como es el caso de lo que requiere esta empresa.**

***Index Terms*—GANs, LSTM, empresa de plásticos, reconstrucción, interpolación, series temporales incompletas.**

I. Introducción

El presente trabajo se enfoca en realizar una propuesta mediante la aplicación de redes neuronales generativas, para resolver una de las problemáticas a la que se enfrentan la mayoría de pequeñas y medianas empresas en Ecuador respecto a los datos faltantes de las ventas diarias de los productos que usualmente es ocasionado por diferentes motivos como falta de recursos económicos, tecnológicos para implementar un sistema transaccional adecuado para la captura de las ventas diarias, la falta de cultura organizacional, procesos, políticas para el correcto manejo e ingreso de los datos en el sistema.

En la mayoría de las empresas, las causas de los datos faltantes, se debe a la falla en los sensores, interrupciones en los sistemas transaccionales que dificulta la adecuada recolección de datos [1], el uso de técnicas avanzadas para la interpolación de datos permite mejorar la consistencia y calidad de datos, lo cual es indispensable para el desarrollo de los modelos de pronóstico y el sustento de la toma de decisiones.

La interpolación de datos contribuye con la reducción de costos relacionados a la incorrecta toma de decisiones debido a datos incompletos. En los campos principalmente de salud, la industria manufacturera, energía, asegurar la integridad de los datos tiene un impacto directo en la mejora de la eficiencia operativa y la toma de decisiones estratégicas [1] [2]. Un ejemplo específico en el ámbito de salud, en el caso de los monitoreos clínicos, al adecuado manejo de los datos faltantes, marcará la diferencia entre diagnósticos oportunos y precisos y los malos diagnósticos que perjudiquen al paciente [2].

En el caso de las empresas de retail, los datos faltantes detectados en las series temporales tienen un impacto negativo en los modelos de pronóstico, la gestión de inventarios y costos operativos asociados, para mitigar este efecto negativo se hace utilizan diferentes métodos de interpolación que permiten garantizar la coherencia con los datos históricos existentes.

Existen varios métodos tradicionales y modernos para la interpolación de datos en series temporales. Entre los métodos tradicionales más utilizados se encuentran:

- **Interpolación lineal y polinómica:** La interpolación lineal tiene el enfoque de conectar puntos con líneas rectas, mientras que la polinómica hace uso de polinomios para lograr un mejor ajuste. No obstante, estos métodos suelen presentar problemas el momento de capturar patrones complejos en series temporales de alta variabilidad [3].

- **Splines cúbicos:** Por otra parte, los splines cúbicos se utilizan para la interpolación de datos temporales que presentan variaciones suaves, una de las consideraciones que se debe tener presente es que este tipo de métodos puede tener un costo computacionalmente alto [4].

Entre los métodos modernos basados en redes neuronales, se mencionan los siguientes:

- **Redes neuronales recurrentes (RNN):** Fueron diseñadas para el manejo de datos secuenciales, lo cual resulta de gran utilidad para el manejo

y modelamiento de dependencias temporales. Sin embargo, el problema del gradiente evanescente representa una limitación para el uso de este tipo de redes en dependencias a largo plazo [5].

- **Long Short - Term Memory (LSTM):** Es una variación de las RNN, las LSTM diseñada se encuentran enfocadas en solucionar el problema del gradiente evanescente y con la capacidad de "recordar" a largo plazo, mediante el uso de puertas que controlan el flujo de información, mejorando el desempeño en las tareas de interpolaciones temporales [6].

- **Transformers:** El enfoque está basado en mecanismos de autoatención, logrando superar las limitaciones de secuencialidad presentada en las RNNs, lo cual permite modelar relaciones más complejas en series temporales con una alta capacidad de paralelización [3].

- **Redes Neuronales Generativas Adversarias (GANs):** Este tipo de redes neuronales fueron originalmente diseñadas para la generación de nuevos datos, entre sus aplicaciones se ha visto la creación de nuevas piezas musicales y actualmente están siendo adaptadas para la interpolación de datos faltantes, dado que al combinar generadores y discriminadores, las GANs son capaces de producir estimaciones realistas de datos faltantes en series temporales de manera más robusta. Tiene varias aplicaciones, como la reconstrucción de series temporales en el campo de finanzas [7].

La empresa en la cual se basa el desarrollo de este trabajo, se caracteriza por ser una mediana empresa de retail de productos pláticos, con una trayectoria de 10 años en el mercado, cuenta con un portafolio de aproximadamente 800 items. La herramienta que utiliza la empresa para la recolección de los datos de las ventas diarias de los productos es mediante el complemento de Excel llamado Script Lab.

La recolección de los datos es diaria desde hace 7 años atrás con respecto a toda la información de venta de los productos, por lo cual considera que tiene la suficiente cantidad de datos para comenzar a realizar análisis más profundos respecto al comportamiento temporal de la venta de los productos y también para generar modelos de pronósticos más avanzados que permitan mejorar la eficiencia de las operaciones de abastecimiento de la empresa.

Sin embargo, cuando la empresa comenzó a utilizar los datos con la finalidad de construir modelos de pronóstico robustos, notó que existian datos faltantes entre un día y otro, lo cual dificultaba poder aplicar cualquier técnica de forecasting o cualquier otro tipo de análisis y obtener resultados confiables, por consiguiente, la problemática será solucionada mediante la aplicación de métodos de interpolación de datos.

En base al comportamiento de las ventas de los productos desde enero 2021 hasta octubre 2024 y luego de analizar los diferentes métodos tradicionales y modernos, se concluye que para la reconstrucción de series temporales y recuperar los datos faltantes de las ventas, una alternativa adecuada es el uso de las redes neuronales generativas adversarias (GANs).

La naturaleza de las series de tiempo de ventas suelen presentar comportamientos no lineales, debido a las fluctuaciones del mercado, promociones y otras variables exógenas, es por tal motivo que se seleccionan las redes neuronales generativas adversarias (GANs) dado que tienen la capacidad de capturar patrones complejos no lineales, que otros métodos como la interpolación lineal o los splines cúbicos no logran hacerlo con precisión [8].

El funcionamiento de las GANs consiste en que la red aprende patrones complejos a través de un proceso adversarial entre un generador y un discriminador [8], al utilizar un discriminador que fuerza al generador a producir datos indistinguibles de los datos reales, esto representa una gran ventaja al trabajar con datos que presentan ruido [9].

Considerando que existen diversos tipos de variaciones de las GANs como Time-series Generative Adversarial Networks (TimeGAN), Continuous Recurrent Neural Network GAN (C-RNN-GAN), WaveGAN, Recurrent Generative Adversarial Network (R-GAN), se construyó varios modelos con diferentes hiperparámetros y enfoques de entrenamiento.

De forma general, utilizando los datos de venta diaria de la empresa del catálogo de productos, se implementara una Red Neuronal Generativa Adversaria (GAN) especializada en la reconstrucción de series temporales utilizando un enfoque basado en Long Short-Term Memory (LSTM), la arquitectura del modelo está conformada por un generador que es una red LSTM encargada de utilizar ruido y contexto para predecir los valores enmascarados, la salida para por capas densas y activación para generar valores realistas y el discriminador por otro lado es una red completamente conectada que clasifica si un valor proviene de los datos reales o ha sido generado.

El objetivo de esta implementación es utilizar las GANs con ciertas variaciones mejoradas, para interpolar los valores faltantes y de esta manera reconstruir la serie temporal de las ventas diarias de los productos, estos resultados van a permitir reconstruir la serie, asegurando una mayor coherencia y consistencia en los datos, para posteriormente generar modelos de pronósticos con la finalidad de mejorar el abastecimiento de la empresa, reduciendo costos operativos, ventas perdidas e incrementando las ventas.

En la investigación bibliográfica realizada, se encontró que este tipo de GANs actualmente no son aplicadas ampliamente en el campo del retail, tampoco existen aplicaciones en empresas de retail de productos plásticos,

razón por la cual la contribución de este trabajo se vuelve tan importante porque permite resolver uno de los problemas más complejos de manejar en las empresas.

A menudo el problema respecto a los datos faltantes requieren de soluciones e implementaciones tecnólogicas que son altamente costosas y pensadas únicamente para que se encuentren al alcance de las grandes empresas, sin embargo este enfoque representa una alternativa para las pequeñas y medianas empresas buscan mejorar la calidad de sus datos y fortalecer sus modelos de predicción para mejorar su gestión.

En resumen, este trabajo se encuentra orientado para ser una alternativa viable que contribuya a solucionar el problema de la falta de datos en las series temporales de la venta de los productos, el proyecto se encuentra alojado en un repositorio público en Github, con la finalidad de facilitar su réplica o que permita ser una línea base para mejoras futuras, la información se encuentra en el siguiente link:

https://github.com/SkaDataScience/Interpolacion

II. Estado del arte

La gestión de datos incompletos es todo un desafío por lo cual la interpolación de datos es esencial en las industrias donde los datos se caracterizan por ser ruidosos, incompletos o irregulares, en virtud de los dificultades que enfrentan los métodos tradicionales ante patrones complejos temporales, el enfoque de las GANs brindan la alternativa generar datos sintéticos que preserven las relaciones complejas. Las aplicaciones de las Redes Neuronales Generativas Adversarias (GANs) han demostrado ser una solución alternativa frente a la falta de datos en las industrias.

Entre algunos de los ejemplos, se tiene que las GANs se han empleado para simular la secuencia de compras que un cliente potencialmente podría realizar en un determinado tiempo, haciendo uso de las Redes Neuronales Recurrentes (RNN) para aprender el comportamiento de clientes a partir de su historial transaccional, una GAN genera posibles combos de productos para las semanas futuras, esto permite actualizar el estado del cliente con cada nuevo combo generado, lo cual facilita la simulación de futuras secuencias de compras [10].

En otro de los ejemplos se tiene el modelo eCommerceGAN (ecGAN), de forma sencialla este modelo ayuda a grandes tiendas como Amazon o Alibaba, entre otras afines, para incrementar el cross - selling, dado que estas tiendas tienen millones de pedidos, peros esos pedidos represntan solo una pequeña muestra de todas las posibles combinaciones de cosas que las personas podrían comprar, lo que hace este modelo es imaginar o crear pedidos falsos como si fueran nuevos pedidos que parecen reales [11].

Lo que hace la eCommerceGAN (ecGAN) es predecir si el cliente que compró una computadora y un mouse, también tiene el suficiente potencial de estar interesado en la compra de un teclado, otra de las cosas que está en la capacidad de hacer es que si una tienda lanza un nuevo producto (un nuevo teléfono, marca de un nuevo proveedor), el modelo puede imaginar qué otros productos tienen una alta posibilidad de comprarse en conjunto con ese teléfono como audífonos o estuches [11].

El sustento técnico para el desarrollo de estas aplicaciones se detalla a continuación, de manera general las GANs constan de dos redes:

- **Generador (G):** Produce datos sintéticos a partir del ruido latente z en datos sintéticos $G(z)$, con el objetivo de replicar la distribución de los datos reales.
- **Discriminador (D):** Clasifica entre los datos reales x y generados (sintéticos) $G(z)$, ayudando a mejorar al generador.

El proceso se define como un juego minimax, donde G intenta minimizar su error al engañar a D, y D intenta maximizar su precisión al distinguir entre x y $G(z)$:

$$\min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))].$$

En donde:

- $p_{\text{data}}(x)$: Distribución real de los datos.
- $p_z(z)$: Distribución de ruido latente
- $D(x)$: Probabilidad de que x sea real.
- $G(z)$: Dato sintético generado a partir de z.

El funcionamiento consiste en que G genera datos sintéticos $G(z)$ usando $z \sim p_{\text{z}}(z)$, por otro lado D evalúa x y $G(z)$, devolviendo las probabilidades de autenticidad, las pérdidas $\mathcal{L}_G$ y $\mathcal{L}_D$ se actualizan iterativamente para mejorar ambos modelos, en conclusión el generador G mejora al miniminizar la probabilidad de que D identifique datos sintéticos, mientras que el discriminador intenta maximizar su precisión, creando un equilibrio que resulta en datos sintéticos realistas [8].

En resumen, en la investigación realizada las GANs han sido utilizadas en diferentes ámbitos de la salud, finanzas, ventas, entre otros y esto se debe principalmente a estas dos fortalezas:

- **Generación de datos:** Para este objetivo se han desarrollado modelo como TimeGAN con la finalidad de generar series de tiempo realistas que capturen las características estadísticas y las dependencias temporales de los datos históricos.

- **Imputación de datos:** Modelos como GAIN, e ImputeGAN han demostrado su eficacia en la imputación de datos faltantes en series de tiempo multivariadas

Luego de un análisis técnico del funcionamiento de las GANs y sus aplicaciones, se determina que la diferencia de este trabajo en comparación con los anteriores radica en la combinación de técnicas avanzada de aprendizaje profundo (GANs y LSTMs) para la reconstrucción de series temporales de las ventas de los productos, a partir de datos enmascarados para una empresa de retail de productos plásticos.

El modelo que se ha planteado implementar es una Red Neuronal Adversaria (GAN) enfocada en la reconstrucción de series temporales, la red utiliza un generador basado en LSTM (Long Short - Term Memory) para generar datos temporales artificiales y un discriminador para evaluar la calidad de los datos generados. El objetivo principal de la red es reconstruir de manera precisa las secuencias de datos de la cantidad vendida a partir de datos enmascarados, utilizando un contexto temporal y ruido.

El enfoque propuesto en este trabajo es innovador porque no existen registros de aplicación de las GANs en empresas de retail de plásticos, por otro lado el diseño del modelo también interesante porque combina ténicas de aprendizaje profundo como GANs y LSTMs para para la restauración de series temporales incompletas o deterioradas, se convierte en un gran aporte y con el potencial de ser aplicado en la industria de retail a gran escala para mejorar las predicciones de la demanda.

III. Materiales y Metodología

Luego de detectar el problema principal al cual se enfrenta la empresa respecto a la falta de datos en ciertos días, se realiza la propuesta del modelo a implementar para recuperar la información perdida, por consiguiente se solicita a la empresa la información requerida, en este caso la venta diaria de los productos desde hace cuatro años atrás para tener una cantidad adecuada de datos que permita el entrenamiento de la red neuronal,

La empresa entregó cuatro archivos en formato excel, que se detallan a continuación:

- **2021-PLASTICOS:** Este archivo tiene un tamaño de 2.314 KB, se encuentra conformado por 10 pestañas y se va a utilizar la hoja con el nombre *'REGISTRO_FACTURAS'* en la que se encuentra una matriz de 23 columnas por 11843 filas con información de las ventas diarias de los productos del año 2021.

- **2022-PLASTICOS-depurado:** Este archivo tiene un tamaño de 4.344 KB, se encuentra conformado por 10 pestañas y se va a utilizar la hoja con el nombre *'REGISTRO_FACTURAS'* en la que se encuentra una matriz de 23 columnas por 22793 filas con información de las ventas diarias de los productos del año 2022. Algo importante a mencionar con respecto a este archivo fue que en el proceso de limpieza se detectó datos que no correspondían al año 2022, se puso en conocimiento de la empresa y en base a esta alerta, envió un nuevo archivo depurado.

- **2023-PLASTICOS-depurado:** Este archivo tiene un tamaño de 3.423 KB, se encuentra conformado por 10 pestañas y se va a utilizar la hoja con el nombre *'REGISTRO_FACTURAS'* en la que se encuentra una matriz de 23 columnas por 16477 filas con información de las ventas diarias de los productos del año 2023. Algo importante a mencionar con respecto a este archivo fue que en el proceso de limpieza se detectó datos que no correspondían al año 2023, se puso en conocimiento de la empresa y en base a esta alerta, envió un nuevo archivo depurado.

- **2024-PLASTICOS:** Este archivo tiene un tamaño de 2.525 KB, se encuentra conformado por 10 pestañas y se va a utilizar la hoja con el nombre *'REGISTRO_FACTURAS'* en la que se encuentra una matriz de 23 columnas por 12652 filas con información de las ventas diarias de los productos del año 2024.

Luego de verificar los archivos proporcionados por la empresa, se concluye que si se analiza por cada uno de los productos no se van a tener suficientes datos para entrenar el modelo, considerando que en un día determinado se pueden vender unos productos y otros no. En base a esto, se determina que se necesita una limpieza de datos previa, para consolidar en un solo archivo y agrupar las ventas de cada producto por día, para lo cual se verifica que en cada pestaña con el nombre de: *'REGISTRO_FACTURAS'*, se mantiene el mismo formato de matriz y consta de las siguientes variables:

- **FECHA:** La fecha en cual se registra que se realizó la venta.
- **NUM FACTURA:** Número de factura.
- **CLIENTE:** Nombre del cliente.
- **CÉDULA/RUC:** Datos del cliente.
- **CÓDIGO:** Es el código del producto (por sugerencia del líder del proyecto de la empresa nos indicó que se tome esta variable y no la del nombre del producto).
- **DESCRIPCIÓN:** Es el nombre que describe la funcionalidad del producto.
- **STOCK_ANTERIOR:** Cantidad de inventario anterior a esta venta.
- **CANTIDAD:** Es la cantidad vendida al cliente.
- **P. VENTA:** El precio de venta al público.
- **P. COMPRA:** Precio al cual se compró al proveedor.
- **GANANCIA_U:** Es la ganancia unitaria de ese producto en esa venta.
- **GANANCIA_T:** Es la ganancia total de la venta de ese producto.
- **SUBTOTAL:** Valor antes de impuestos.
- **TOTAL:** Es el valor incluido impuestos.

- **OBSERVACIONES:** Si existió alguna novedad con la venta.
- **VENDEDOR:** Nombre del vendedor que realizó la venta.
- **TIPO_PAGO:** Si el pago fue en efectivo o tarjeta de crédito.
- **FECHA_PAGO:** Sin registros.
- **DIAS_CREDITO:** Sin registros.
- **SRI_SI/NO:** Sin registros.
- **OBSERVACIONES2:** Sin registros.
- **%:** Sin registros.
- **INVERSION_TOTAL:** Sin registros.

En el proceso de limpieza de datos, se comienza por remover las filas de totales, se valida la consistencia de los datos que deben existir en cada una las columnas, es decir que si es una variable cuantitativa no debe existir texto o viceversa si es una variable categórica no deberían existir números, por ejemplo en el nombre del cliente o del proveedor no deberían existir números.

Adicionalmente, en la validación de la consistencia de los datos, las anomalías detectadas fueron alertadas a la empresa y se solicitó la aclaración y directriz del tratamiento de esos datos, por ejemplo si en la variable de cantidad vendida existían valores negativos, se reemplazó con 0, en base a la explicación y solicitud de la empresa, por otro lado, considerando que existe la misma estructura de los datos en cada una de las pestañas, se decide eliminar las otras variables y trabajar únicamente con las citadas a continuación:

- **FECHA:** La fecha en cual se registra que se realizó la venta.
- **CÓDIGO:** Es el código del producto (por sugerencia del líder del proyecto de la empresa nos indicó que se tome esta variable y no la del nombre del producto).
- **CANTIDAD:** Es la cantidad vendida al cliente.
- **P. VENTA:** El precio de venta al público.

En base a estas variables se hace una agrupación de los datos en base a la FECHA, se suma la variable CANTIDAD, se obtiene el promedio de la variable P. VENTA, de esta manera no se tienen las ventas de forma individualizada por cada uno de los productos y se obtienen las ventas de la empresa agrupadas de forma global de forma diaria desde enero 2021 hasta octubre 2024.

El mismo proceso de agrupación se realiza para cada una las pestañas *'REGISTRO_FACTURAS'* de los archivos excel 2021, 2022, 2023 y 2024, luego de la limpieza y preprocesamiento de datos, se consolida en un solo archivo depurado llamado *'Consolidada_depurada'* que tiene un peso de 373 KB y 1317 registros.

Luego de obtener el archivo depurado, se hace uso de la herramienta de Colab, se abre un nuevo documento de Google Colab para comenzar con el análisis de datos y generar posibles modelos, para lo cual se carga el archivo excel *'Consolidada_depurada'* en el Colab, adicionalmente se valida que el conjunto de datos se encuentre en el formato establecido para cotinuar con el análisis de los posibles modelos que se podrían utilizar.

Se comienza con el desarrollo de la propuesta de algunos tipos de modelos para abordar la problemática de los datos faltantes, con la finalidad de probar diferentes hiperparámtros, arquitecturas y determinar cuál sería la mejor propuesta, a continuación se detallarán los siete modelos que fueron desarrollados.

En el Modelo 1 utiliza una arquitectura GAN mejorada con LSTM que se especializa en trabajar con datos secuenciales como series temporales, el enfoque de esta capacitación es combinar la capacidad de las redes LSTM para modelar dependencias temporales con la estructura competitiva de las GAN para generar datos realistas.

Este modelo que consta del Generador, funciona de la siguiente manera, la secuencia de entrada, de tamaño 'input_lenght' = 30, se procesa paso a paso a través de la capa LSTM. La salida de la LSTM, específicamente el último estado oculto, se pasa a través de dos capas densas (fully connected). La primera capa tiene 64 neuronas y utiliza una función de activación ReLU para introducir no linealidad, mientras que la capa final utiliza Tanh para limitar la salida generada al rango [-1,1], compatible con los datos normalizados. La salida del generador es una secuencia de 15 valores predichos, que representa los pasos temporales futuros.

Por otra parte, el discriminador recibe como entrada tanto secuencias reales como generadas, luego procesa cada secuencia a través de la LSTM para generar un estado oculto final, que luego se transforma en una probabilidad escalar, este valor indica la confianza del discriminador en que la secuencia evaluada sea real.

Por consiguiente, el generador y el discriminador interactúan en un proceso competitivo. El objetivo del generador es intentar engañar al discriminador produciendo secuencias que se asemejen a las reales, mientras que el discriminador intenta mejorar su capacidad para diferencias entre secuencias reales y las generadas.

La pérdida utilizada para entrenar el discriminador es la entropía cruzada binaria (BCELoss), que mide qué tan bien clasifica las secuencias como reales o generadas, en el caso del generador la función de perdida tiene dos componentes: una perdida adversarial (BCELoss) que mide qué tan exitoso es en engañar al discriminador y una pérdida de reconstrucción (MSELoss) que asegura que las secuencias generadas se asemejen a los datos reales, la perdida total del generador es la suma de las dos pérdidas.

Luego se experimentó como el Modelo 2, que de igual manera es la variación de una GAN mejorada con LSTM diseñada para abordar la reconstrucción completa de una serie temporal a partir de una única ventana de entrada, a diferencia del modelo anterior, que trabaja con múltiples pares entrada-salida, este enfoque se centra en generar una

secuencia extendida de predicciones que cubran la totalidad de los datos posteriores a la ventana inicial.

Las diferencias clave con el Modelo 1 se encuentran en que el Modelo 2 utiliza una sola ventana de entrada fija para construir toda la serie, es decir que considerando que el enfoque es global, el modelo debe ser capaz de aprender patrones globales en los datos y generalizar más allá de las dependencias locales, mientras que el Modelo 1 tiene un enfoque local dado que utiliza múltiples ventanas deslizantes para generar pares entrada-salida.

En el Modelo 2, el generador y el discriminador necesitan mayor capacidad para manjear la longitud extendida de las secuencias, con la finalidad de modelar tanto dependencias a corto como a largo plazo, por esta razón se utilizan dos capas LSTM (no sólo una) para capturar dependencias jerárquicas y modelas las relaciones temporales en escalas más amplias.

El Modelo 3 tiene un enfoque diferente para reconstruir ventanas de una serie temporal a partir de un conjunto reducido de puntos seleccionados aleatoriamente, el modelo se basa en una GAN mejorada con LSTM, donde el generador aprende a reconstruir las partes faltantes de la serie a partir de puntos de referencia, mientras que el discriminador valida si las ventanas reconstruidas son plausibles, esto implica un gran desafío porque el modelo debe aprender a interpolar secuencias a partir de información dispersa.

El generador recibe una entrada reducida por ejemplo 5 puntos y genera los 25 puntos restantes necesarios para completar una ventana de tamaño 30, la arquitectura está conformada por: dos capas LSTM que procesan los puntos dispersos para capturar patrones subyacentes y dependencias temporales y por capas fully connected que transforman el estado oculto final de la LSTM para producir los 25 puntos restantes con ativación Tanh para mantener las salidas en el rango normalizado.

El discriminador clasifica ventanas completas como reales o generadas, la arquitectura es idéntica al generador, excepto que en la salida producto un escalar ([0,1]) mediante un función sigmoide que representa la probabilidad de que la ventana evaluada sea real.

En el Modelo 4 se agrega una capa interactiva para incorporar la idea que el usuario ingrese las fechas de rangos, fecha inicio y fecha fin y entre esas dos fechas reconstruya los datos faltantes o desconocidos basándose en punto de entrada seleccionados aleatoriamente dentro del intervalo, este enfoque agrega una visión de aplicación práctica en el mundo empresarial, abriendo la posibilidad que las personas puedan ingresar diferentes fechas y se recupere la información intermedia.

En comparación con las anteriores propuestas, el Modelo 4 se distingue por su enfoque progresivo e interactivo, su arquitectura se encuentra adaptada para reconstruir series temporales de manera iterativa dentro de un rango dinámico de fechas definido por usuario, la arquitectura del generador utiliza dos capas LSTM para procesar un conjunto inicial de puntos seleccionados aleatoriamente como entrada y generar un primer valor faltante en la secuencia, el valor generado se desnormaliza y se incorpora como nueva entrada en un esquema de ventana deslizante, seemplazando el punto más antiguo.

El generador repite el mismo proceso de manera iterativa, avanzando paso a paso hasta completar la reconstrucción del intervalo, mientras tanto el discriminador también compuesto por dos capas LSTM y capas fully connected, evalúa ventanas completas (reales o generadas) para clasificar si son auténticas o generadas, retroalimentando al generador para mejorar la calidad de sus predicciones, a diferencia de otros modelos que generan ventanas o secuencias completas en un solo paso, este enfoque progresivo permite manejar intervalos de fechas dinámicas y trabajar de manera más efectiva con datos incompletos o parciales.

En el caso del Modelo 5 presenta una arquitectura especializada para reconstruir puntos enmascarados dentro de una serie temporal, este modelo se basa en una R-GAN mejorada con LSTM, diseñada para aprender a interpolar valores específicos utilizando el contexto de punto adyacentes, este modelo incorpora ruido condicional como entrada al generador, lo que permite capturar distribuciones más complejas.

Este modelo trabaja enfocado en reconstrucción focalizada, en lugar de trabajar con ventas completas como algunos de los modelos anteriores (Modelo 1 y 2) o puntos dispersos seleccionados aleatoriamente (Modelo 3), en este caso selecciona un único punto central como objetivo de predicción, mientras utiliza los puntos anteriores y posteriores al enmascarado como contexto.

En este enfoque, el generador no solo utiliza el contexto como entrada, sino que también incorpora un vector de ruido de dimensión fija, lo que le permite generar predicciones con mayor variabilidad y robustez, adicionalmente el discriminador en comparación con los anteriores modelos que procesaban ventanas completas, lo que hace es clasificar únicamente el punto reconstruido como real o generado.

El generador utiliza dos capas LSTM con una dimensión oculta de 512, lo que permite capturar patrones temporales complejos en el contexto, las salidas de las LSTM son procesadas por capas densas que transforman el estado oculto final en un valor único correspondiente al punto enmascarado. En cambio, el discriminador es completamente denso y procesa directamente el punto generado o real, evaluando la plausibilidad mediante una salida sigmoide.

El Modelo 6 tiene el enfoque de tener una arquitectura GAN mejorada con LSTM para reconstruir completamente una serie temporal a partir de una única ventana de entrada, este modelo está diseñado para generar reconstrucciones de largo plazo, utilizando una venta fija inicial como base para reconstruir la totalidad de los datos restantes.

En donde, el generador utiliza dos capas LSTM para capturar las dependencias temporales de largo plazo presentes en la ventana inicial y produce una secuencia extendida que corresponde a la totalidad de los datos restantes de la serie, de forma paralela, el discriminador evalúa la similitud con la realidad de las secuencias generadas, clasificándolas como reales o falsas.

En el modelo 2, el generador produce una ventana de salida fija inmediatamente posterior a la ventana inicial, diseñada para capturar patrones locales de corto y mediano alcance, esto significa que para reconstruir una serie completa, sería necesario iterar sucesivamente extendiendo la salida con nuevas ventanas, lo que aumenta el costo computacional y depende de la precisión acumulativa en cada predicción.

Este modelo se encuentra optimizado para generar directamente la serie completa restante en un solo paso, a partir de la misma ventana inicia, esto requiere un generador capaz de manejar salidas mucho más grandes y de capturar patrones globales de largo plazo que se detectan en las series, reduciendo así la necesidad de iteraciones y mejorando la eficiencia en tareas de predicción global.

El último modelo desarrollado número 7, utiliza una arquitectura GAN con LSTM diseñada para reconstruir series temporales a partir de ventanas deslizantes, el tamaño de la ventana fija es 50, y el tamaño de las solapadas es 10, cada ventana es tratada como una unidad independiente durante el entrenamiento y posteriormente se combinan las ventanas generadas para reconstruir la serie completa, esto permite capturar patrones locales y globales en la serie temporal, manteniendo una estructura coherente al ensamblar las ventanas.

La arquitectura del generador recibe un vector de ruido latente *(z)* de dimensión latent_dim = 100 y genera una ventana de datos de tamaño window_size. Este vector pasa por dos capas LSTM con hidden_dim = 256, seguidas por capas fully connected que transforman la salida final en una ventana reconstruida normalizada, por otro lado, el discriminador recibe ventanas reales o generadas, procesándolas a través de dos capas LSTM y capas densas que clasifican las ventanas como reales o falsas mediante una salida sigmoide.

En comparación con los otros modelos, éste se basa en ruido latente para generar ventanas en lugar de depender exclusivamente de datos históricos o puntos dispersos, la combinación de ventanas solapadas, asegura continuidad entre las predicciones locales, trabajar con ventanas independientes es adecuado para series largas o segmentadas, la combinación de ventanas solapadas asegura continuidad entre las predicciones locales.

Luego de realizar varios experimentos con los diferentes modelos, se presenta una tabla comparativa entre cada uno de los experimentos, la finalidad de esto es tener la suficiente información de las arquitecturas, hiperparámetros utilizados para poder decidir cuál de estos modelos se adaptaría de mejor forma al enfoque inicial de un modelo que a partir de poca información, se reconstruya la serie temporal completa.

Table I
Comparación de Modelos GAN para Reconstrucción de Series Temporales (Modelos 1 al 4)

Modelo	Hiperparámetros	Arquitectura y Funcionamiento	Diferencias y Ventajas frente a Otros Modelos
1	• **Input Length:** 30 • **Output Length:** 15 • **Hidden Dim:** 128 • **Num Layers:** 1 • **Learning Rate:** 0.001 • **Patience:** 10	Modelo básico de GAN con generador y discriminador basados en LSTM. El generador aprende a reconstruir ventanas deslizantes cortas, mientras que el discriminador evalúa la autenticidad de las secuencias generadas.	Enfoque simple y eficiente para reconstrucción local, pero limitado para series más extensas y complejas. Útil como punto de partida por su baja complejidad computacional.
2	• **Input Length:** 30 • **Output Length:** Completa • **Hidden Dim:** 128 • **Num Layers:** 2 • **Learning Rate:** 0.0005 • **Patience:** 20	Extensión del Modelo 1 con un generador que reconstruye toda la serie restante a partir de una única ventana inicial. Usa LSTM más profundo con dos capas.	Capaz de capturar patrones globales en la serie. Reduce la iteración sobre ventanas, optimizando el procesamiento de series largas. Ideal para predicciones más amplias.
3	• **Input Points:** 5 aleatorios • **Hidden Dim:** 128 • **Num Layers:** 2 • **Learning Rate:** 0.0005 • **Patience:** 20	El generador reconstruye ventanas a partir de puntos de entrada seleccionados aleatoriamente, mientras que el discriminador evalúa la coherencia entre los puntos reales y generados.	Versátil para series incompletas o con datos faltantes. Sobresale en la interpolación de valores faltantes dentro de ventanas. Mejora la robustez frente a ruido.
4	• **Input Length:** 30 • **Output Length:** 30 • **Hidden Dim:** 128 • **Num Layers:** 2 • **Learning Rate:** 0.0005 • **Patience:** 20	Modelo GAN con ventanas deslizantes completas y bidireccionales. Permite la reconstrucción de un rango definido de ventanas por el usuario.	Ofrece flexibilidad para reconstrucción personalizada en rangos específicos. Útil para tareas con control explícito sobre las ventanas reconstruidas.

Luego de este análisis, se concluye que el Modelo 3 es la mejor opción el objetivo de este trabajo, dado que este enfoque implementa una arquitectura de Redes Adversariales

Table II
Comparación de Modelos GAN para Reconstrucción de Series Temporales (Modelos 5 al 7)

Modelo	Hiperparámetros	Arquitectura y Funcionamiento	Diferencias y Ventajas frente a Otros Modelos
5	• **Mask Size:** 20 • **Noise Dim:** 10 • **Hidden Dim:** 512 • **Num Layers:** 2 • **Learning Rate:** 0.0001 • **Patience:** 50	Implementación R-GAN con ruido y enmascaramiento. El generador combina un vector de ruido con el contexto de la serie, mientras que el discriminador evalúa su realismo.	Diseñado para datos incompletos o ruidosos. Sobresale en la coherencia global de series temporales extensas. Beneficia la generalización y la interpolación precisa.
6	• **Input Length:** 30 • **Output Length:** Completa • **Hidden Dim:** 128 • **Num Layers:** 2 • **Learning Rate:** 0.0005 • **Patience:** 20	GAN clásica con LSTM bidireccional. El generador produce la reconstrucción de toda la serie desde una única ventana de entrada inicial.	Similar al Modelo 2, pero mejora la precisión mediante ajustes en las pérdidas. Mayor capacidad para capturar patrones globales.
7	• **Window Size:** 50 • **Stride:** 10 • **Latent Dim:** 100 • **Hidden Dim:** 256 • **Num Layers:** 2 • **Learning Rate:** 0.0005 • **Batch Size:** 64	Arquitectura GAN con espacio latente. El generador utiliza ventanas deslizantes superpuestas para la predicción de datos globales y locales.	Capaz de capturar relaciones complejas en series extensas. Ideal para reconstrucción de series con patrones altamente variables.

Generativas (GAN) optimizada para la reconstrucción de series temporales a partir de puntos de datos incompletos. Su diseño combina las capacidades de un generador, que predice valores faltantes en ventanas temporales, y un discriminador, que evalúa la coherencia entre las series generadas y las reales. A continuación, se describe en detalle cómo está construido y cómo funciona este modelo.

A continuación se detalla el funcionamiento de la arquitectura implementada, para una mejor comprensión a profundidad de los fundamentos técnicos que lo sustentan:

Fundamento General de las GAN

Una GAN clásica está compuesta por dos redes neuronales: el generador (G) y el discriminador (D). El generador intenta aprender la distribución de los datos reales para producir datos sintéticos que sean indistinguibles de los reales. El discriminador, por su parte, evalúa si los datos provienen de los reales ($y = 1$) o fueron generados ($y = 0$). En el contexto del Modelo 3, estas redes están adaptadas para manejar ventanas de series temporales, donde:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z)))]$$

Arquitectura del Modelo

La arquitectura de forma general, se observa de la siguiente forma:

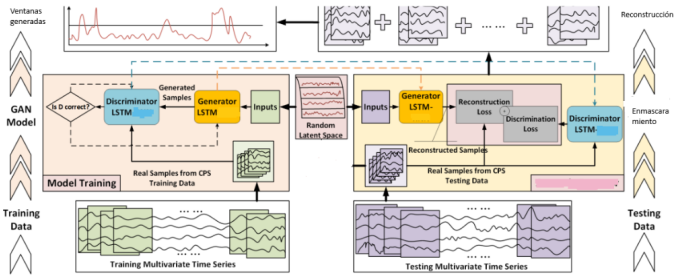

Figure 1. Comportamiento de las ventas de la serie original

El diseño del Modelo 3 se fundamenta en la elección estratégica de hiperparámetros y arquitecturas optimizadas para reconstruir ventanas temporales. Los componentes principales son:

El Generador (G) El generador toma como entrada un subconjunto de puntos seleccionados aleatoriamente dentro de una ventana temporal y produce los puntos faltantes. Su arquitectura está diseñada para capturar dependencias temporales utilizando Long Short-Term Memory (LSTM). Matemáticamente, el generador realiza las siguientes operaciones:

1) **Entrada y LSTM**: Se recibe un tensor $z \in \mathbb{R}^{B \times 5 \times 1}$, donde B es el tamaño del batch y 5 son los puntos seleccionados. La LSTM procesa la secuencia temporal y produce una representación latente:

$$h_t, c_t = \text{LSTM}(z_t, h_{t-1}, c_{t-1}),$$

donde h_t y c_t son el estado oculto y de memoria, respectivamente. La dimensión de salida está definida por el hiperparámetro `hidden_dim = 128`.

2) **Capa Densa y Activación**: La salida de la última LSTM (h_T) pasa por capas densas que transforman la representación latente en la reconstrucción de los puntos faltantes:

$$\hat{x} = \text{Tanh}\left(W h_T + b\right),$$

donde W y b son los pesos y sesgos aprendidos. La activación Tanh normaliza los valores generados en $[-1, 1]$.

3) **Dimensión de Salida**: La salida reconstruida tiene dimensión $\mathbb{R}^{B \times 25}$, donde 25 representa los puntos faltantes en cada ventana de tamaño 30.

El Discriminador (D) El discriminador evalúa si una ventana reconstruida es real o generada. Su arquitectura también se basa en LSTM para capturar patrones temporales. Las operaciones principales son:

1) **Entrada y LSTM**: Se recibe un tensor $x \in \mathbb{R}^{B\times 30\times 1}$, donde 30 es el tamaño de la ventana completa. La LSTM extrae características temporales de la secuencia:

$$h_t, c_t = \text{LSTM}(x_t, h_{t-1}, c_{t-1}),$$

donde `hidden_dim = 128`.

2) **Clasificación**: La última salida de la LSTM (h_T) pasa por una capa densa con activación sigmoide para calcular la probabilidad de que la ventana sea real:

$$p(y = 1|x) = \text{Sigmoid}(Wh_T + b).$$

Hiperparámetros Clave

Los hiperparámetros utilizados en el modelo son:

- **`hidden_dim = 128`**: Controla la capacidad de memoria de la LSTM, permitiendo capturar patrones complejos en series temporales.
- **`num_layers = 2`**: Dos capas en la LSTM aseguran la extracción jerárquica de características temporales.
- **`learning_rate = 0.0005`**: Una tasa de aprendizaje baja garantiza una convergencia estable.
- **`patience = 20`**: Early stopping para evitar el sobreajuste.

Flujo de Entrenamiento

El entrenamiento del Modelo 3 sigue un enfoque adversarial, donde G y D se entrenan alternadamente:

1) **Entrenamiento del Discriminador (D)**:
   - Entrada real: Ventanas reales completas ($x \in \mathbb{R}^{30}$).
   - Entrada falsa: Ventanas generadas por G ($\hat{x}$).
   - Pérdida:

$$\mathcal{L}_D = -\mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)] - \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))].$$

2) **Entrenamiento del Generador (G)**:
   - Se busca maximizar la probabilidad de que D clasifique las ventanas generadas como reales ($y = 1$).
   - Pérdida adversarial:

$$\mathcal{L}_G^{\text{adv}} = -\mathbb{E}_{z\sim p_z}[\log D(G(z))].$$

   - Pérdida adicional por error cuadrático medio (MSE) para ajustar los puntos generados:

$$\mathcal{L}_G^{\text{mse}} = \frac{1}{N}\sum_{i=1}^{N} \|\hat{x}_i - x_i\|^2.$$

   - Pérdida total:

$$\mathcal{L}_G = \mathcal{L}_G^{\text{adv}} + \lambda \mathcal{L}_G^{\text{mse}},$$

donde $\lambda = 10$ pondera la importancia del error MSE.

Reconstrucción de Ventanas

En la fase de prueba, se seleccionan aleatoriamente 5 puntos de cada ventana, que sirven como entrada al generador. El generador predice los 25 puntos faltantes, combinándolos con los 5 iniciales para reconstruir la ventana completa.

IV. Resultados y Discusión

El comportamiento original de la serie de tiempo de la venta diaria de productos de forma global de la empresa de retail de productos plásticos, presenta un comportamiento volátil, con patrones en el tiempo difíciles de aprender, tal como aprecia en la Figura 1.

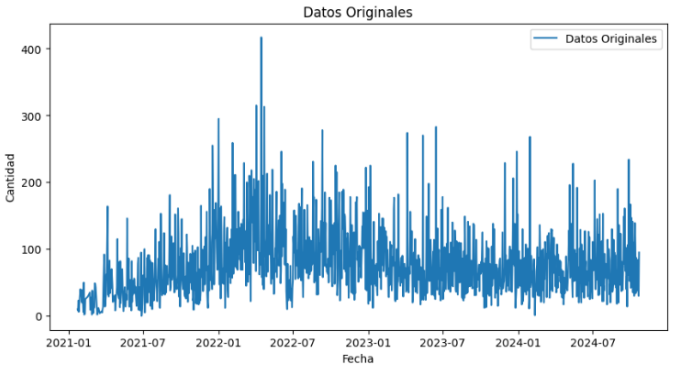

Figure 2. Comportamiento de las ventas de la serie original

Luego de analizar los resultados, se selecciona el Modelo 3 para comprender mejor los resultados obtenidos, el tiempo de procesamiento fue de aproximadamente 10 minutos, el tamaño de del conjunto de entrenamiento fue de 1030 ventanas, el tamaño del conjunto de prueba de 258 ventanas, se imprime las 5 primeros ventanas para observar en la Figura 2, los patrones de los que está aprendiendo el modelo.

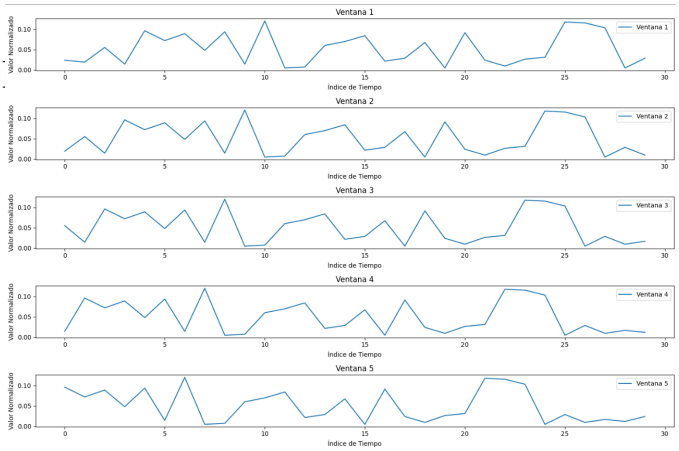

Figure 3. Comportamiento de las ventanas para entrenar el modelo

Los resultados obtenidos del entrenamiento del Modelo 3 muestran que el discriminador equilibra la capacidad

de distinguir entre datos reales y generados, el generador mejora en producir datos que engañan al discriminador y son coherentes con los patrones temporales aprendidos. En conclusión la Figura a continuación, muestra una reducción en la pérdida, lo que quiere decir que la reconstrucción es más precisa con el tiempo y el modelo está aprendiendo.

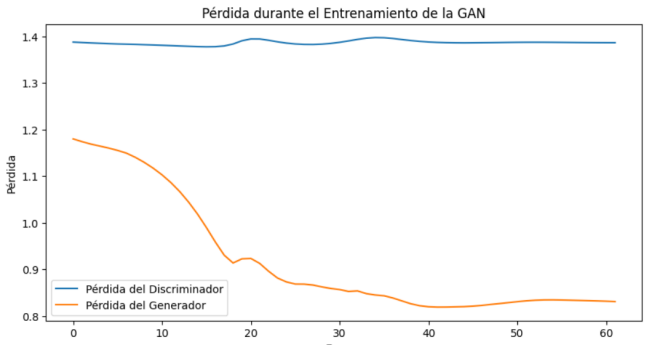

Figure 4. Resultados del entrenamiento

Finalmente en los resultados que se obtienen de la reconstrucción de las series temporales, dados pocos datos, es interesante observar como a partir de pocos datos tiene la capacidad de recuperar la información faltante, como el ejemplo que observamos a continuación en la Figura 4.

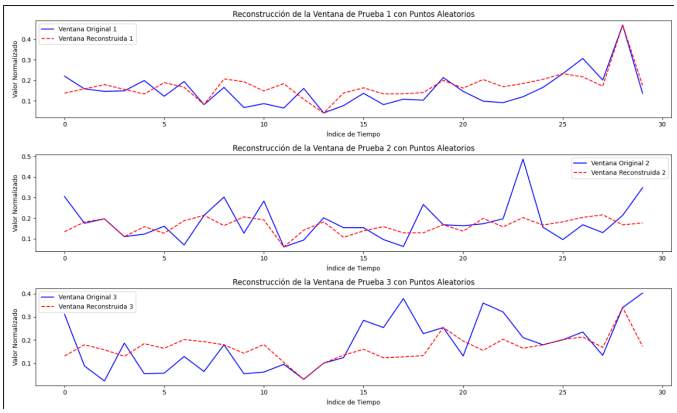

Figure 5. Resultados de la reconstrucción obtenida

En los resultados obtenidos del Modelo 3, se observa como se brinda poca información al modelo e intenta reconstruir los datos faltantes, aunque en los resultados se observa que la red construida, todavía no es capaz de captar, aprender los patrones complejos del comportamiento de la serie original.

V. Conclusiones

Este trabajo representa un gran aporte para el sector empresarial, porque permite recuperar la información entre dos puntos del tiempo definidos, sin embargo es importante tener en cuenta limitaciones como recursos computacionales para el entrenamiento de las redes.

Esta es una solución robusta y eficaz para la reconstrucción de series temporales incompletas. Su diseño combina arquitecturas de redes LSTM y GAN, aprovechando las capacidades de las primeras para capturar dependencias temporales y las de las segundas para garantizar la coherencia entre datos reales y generados. Esto permite reconstruir series temporales incluso en escenarios con datos faltantes o ruido.

El enfoque del modelo, basado en la selección aleatoria de puntos de entrada, demuestra ser altamente adaptable, mejorando su capacidad para manejar series con patrones complejos y discontinuidades. Además, la pérdida MSE incorporada al generador aumenta la precisión de la reconstrucción al minimizar los errores entre los valores reales y generados, logrando resultados más fiables.

Por último, la implementación de early stopping asegura la eficiencia del modelo al detener el entrenamiento cuando se alcanza un punto de convergencia óptimo, evitando sobreentrenamiento y optimizando el uso de recursos computacionales. En conjunto, el Modelo 3 destaca como una herramienta versátil y escalable para resolver problemas de reconstrucción en series temporales, manteniendo un balance adecuado entre precisión, estabilidad y eficiencia.

Agradecimiento

A la vida, al amor, a mi yo del pasado por no rendirse y a todos esos seres humanos que se sintieron perdidos alguna vez como yo y que a pesar de todo nunca dejaron de intentarlo

References

[1] J. B. V. K. Yalavarthi and L. Schmidt-Thieme, "Tripletformer for probabilistic interpolation of irregularly sampled time series," *arXiv preprint arXiv:2210.02091*, 2023. [Online]. Available: https://arxiv.org/abs/2210.02091

[2] T. H. C. X. J. Z. S. P. Yuxi Wei, Juntong Peng and S. Chen, "Compatible transformer for irregularly sampled multivariate time series," *arXiv preprint arXiv:2310.11022*, 2023. [Online]. Available: https://arxiv.org/abs/2310.11022

[3] N. P. J. U. L. J. A. N. G. K. A. Vaswani, N. Shazeer and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, pp. 5998–6008, 2017. [Online]. Available: https://arxiv.org/pdf/1706.03762

[4] T. Lyche and K. Mørken, *Spline Methods for Curve and Surface Fitting*. Cambridge University Press, 2008. [Online]. Available: https://www.cambridge.org/core/books/spline-methods-for-curve-and-surface-fitting

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[6] J. S. F. A. Gers and F. Cummins, "Learning to forget: Continual prediction with lstm," in *Proceedings of the 9th International Conference on Artificial Neural Networks (ICANN)*, vol. 2, 1999, pp. 850–855. [Online]. Available: https://www.researchgate.net/publication/12292425_Learning_to_Forget_Continual_Prediction_with_LSTM

[7] J.-B. A. M. Lepot and F. H. L. R. Clemens, "Interpolation in time series: An introductive overview of existing methods, their performance criteria and uncertainty assessment," *Water*, vol. 9, no. 10, p. 796, 2017.

[8] M. M. B. X. D. W.-F. S. O. A. C. I. Goodfellow, J. Pouget-Abadie and Y. Bengio, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, pp. 2672–2680, 2014. [Online]. Available: https://arxiv.org/pdf/1406.2661v1.pdf

[9] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, vol. abs/1312.6114, 2014. [Online]. Available: https://arxiv.org/pdf/1312.6114v11.pdf

[10] T. Doan, N. Veira, B. Keng, and S. Ray, "Generating realistic sequences of customer-level transactions for retail datasets," *IEEE International Conference on Data Mining Workshops (ICDMW)*, vol. 1, pp. 1234–1240, 2019. [Online]. Available: https://arxiv.org/abs/1901.05577

[11] A. Kumar, A. Biswas, and S. Sanyal, "ecommercegan: A generative adversarial network for e-commerce," *Proceedings of the ACM Conference on Artificial Intelligence*, vol. 1, no. 1, pp. 987–993, 2018. [Online]. Available: https://arxiv.org/abs/1801.03244