GENERACIÓN DE SERIES DE TIEMPO FINANCIERAS SINTÉTICAS PARA “DATA AUGMENTATION” USANDO REDES NEURONALES GENERATIVAS ADVERSARIAS (GAN)

GENERATION OF SYNTHETIC FINANCIAL TIME SERIES FOR “DATA AUGMENTATION” USING GENERATIVE ADVERSARIAL NETWORKS (GAN)

Elaborado por: Edwin Fernando Villarraga Ossa

Tutor: Fernán Alonso Villa Garzón. Phd.

Trabajo Final para optar al título de Maestría en Ingeniería

MAESTRIA EN INGENIERIA – ANALÍTICA

UNIVERSIDAD NACIONAL DE COLOMBIA

SEDE MEDELLÍN

2021

TABLA DE CONTENIDO

RESUMEN

GENERACIÓN DE SERIES DE TIEMPO FINANCIERAS SINTÉTICAS PARA “DATA AUGMENTATION” USANDO REDES NEURONALES GENERATIVAS ADVERSARIAS (GAN).

Los modelos GAN se han usado de forma exitosa para realizar aumento de datos en problemas relacionados con imágenes, audio y video, pues logran representar adecuadamente las propiedades de los datos reales, pero incorporando suficiente diversidad en los datos sintéticos generados como para poder mejorar el desempeño de los modelos de machine learning y deep learning en las evaluaciones por fuera de muestra. Las series de tiempo financieras se requieren para la modelación y solución de problemas en finanzas, sin embargo, dada la escasez de datos históricos, no solo originados por problemas de recolección de datos, sino también porque una serie de tiempo es solamente la realización de un proceso estocástico y por ende se presenta un sub muestreo. En este trabajo se generaron series de tiempo sintéticas usando DCGAN y cCGAN para generar datos de rendimientos, volúmenes, bid-ask spread, y precios con transformación fraccional, de acciones de Estados Unidos de América, con periodicidad diaria e intradiaria. Se pudo verificar que estos modelos GAN logran generar series simuladas que representan adecuadamente las propiedades distribucionales de las series históricas. Estas series sintéticas generadas pueden servir como insumo del tipo data augmentation en modelos de machine learning y deep learning para mejorar su desempeño con datos por fuera de muestra.

Palabras Clave: GAN, Data Augmentation, Series de Tiempo, Simulación, Deep Learning.

ABSTRACT

GENERATION OF SYNTHETIC FINANCIAL TIME SERIES FOR “DATA AUGMENTATION” USING GENERATIVE ADVERSARIAL NETWORKS (GAN).

GAN models have been used successfully as a data augmentation method applied to problems related to images, audio and video, since they manage to adequately represent the properties of the real data, but incorporating diversity in the synthetic data generated in order to improve the out-of-sample performance of Machine Learning and Deep Learning models. Financial time series are required for modeling and solving problems in finance, however, given the scarcity of historical data, not only caused by data collection problems, but also because a time series is the realization of only one stochastic process and therefore a subsampling is presented. In this work, synthetic time series were generated using DCGAN and cCGAN to generate data on yields, volumes, bid-ask spread, and prices with fractional transformation, of shares of the United States of America, with daily and intraday periodicity. It was possible to verify that these GAN models manage to generate simulated series that adequately represent the distributional properties of the historical time series. These generated synthetic time series can serve as data augmentation to machine learning and deep learning models to improve their out-of-sample performance.

Key Words: GAN, Data Augmentation, Time Series, Simulation, Deep Learning.

1. JUSTIFICACIÓN

Sezer *et al*. realizaron una revisión sistemática de cómo se ha utilizado el deep learning en finanzas entre los años 2015 a 2019, clasificándolas como aplicaciones de trading algorítmico, riesgo financiero, detección de fraude, construcción y administración de portafolios, valoración de activos financieros y derivados, criptomonedas y blockchain, análisis de sentimientos y finanzas de comportamiento, y minería de texto de noticias financieras [1]. Existen múltiples estrategias de inversión en los mercados financieros, algunos autores como Kakushshadze [2] citan categorías tales como una gran diversidad de portafolios, coberturas y especulación con opciones; estrategias de momento, estrategias de reversión a la media, pair trading, estrategias de arbitraje, estrategias pasivas de replicación, estrategias con volatilidades, estrategias con commodities, estrategias con futuros, estrategias con instrumentos estructurados, etc. La mayoría de casos de uso tienen en común la necesidad de construir, evaluar y calibrar sus modelos con datos históricos de activos financieros.

Se observa entonces que existe un gran interés en el ámbito financiero de aplicar aprendizaje de máquinas y aprendizaje profundo para aplicaciones diversas, sin embargo, existen riesgos en la utilización de estas metodologías. Brooks [3] considera que con la disponibilidad actual de grandes cantidades de datos financieros se incurre en un gran riesgo de realizar “p-hacking”, que consiste en la obtención artificial de significancia estadística de modelos que finalmente, no pueden generalizar bien a pesar de la significancia estadística y se trata esto de un problema que aún debe solucionar la ciencia de datos en finanzas, aunque, también considera que las finanzas empíricas deben considerar como inevitable algún grado de data mining.

En la modelación de series de tiempo en finanzas se busca evitar el fenómeno conocido como data snooping o data mining. White [4] explica que el data snooping ocurre cuando un dato se usa más de una vez para propósitos de inferencia o selección de modelos y este problema, para White, es inherente a las series de tiempo financieras, puesto que los datos corresponden a una sola ejecución histórica de la variable de interés.

Los modeladores financieros deben trabajar con datos escasos para modelar, dado que las series financieras son sólo una realización de un proceso estocástico, y por esto se recurre frecuentemente al backtesting para ajustar modelos, lo cual es precisamente la fuente del data snooping, que sería el equivalente en cuanto a sus efectos, al overfitting en terminología de machine learning, en el sentido de que evita que los modelos puedan generalizar correctamente. Para López de Prado [5], el objetivo de los backtesting es solo descartar modelos con mal desempeño, pero nunca deben ser usados para mejorar dichos modelos, pues usarlos para mejorarlos de manera iterativa, conduce a una filtración de datos.

Los modelos de Machine Learning y Deep Learning han tenido un buen desempeño resolviendo problemas de clasificación o regresión con diferentes tipos de datos

estructurados y/o desestructurados, pero se debe tener precaución de evitar el sobre entrenamiento que produzca problemas en la generalización. En el caso particular del procesamiento de imágenes, se han usado técnicas de "data augmentation" con el fin de mejorar la generalización y robustez de los modelos de clasificación y de este modo disminuir el riesgo de overfitting. Usualmente en problemas con imágenes, transformaciones simples como rotaciones, recortes, adición de filtros, cambios de tamaño y otras, han logrado con éxito mejorar el desempeño de la predicción por fuera de la muestra de entrenamiento.

Las Redes Neuronales Adversarias Generativas (Generative Adversarial Networks -GAN) han sido exitosas en la creación de modelos generadores de datos diferentes a los datos muestrales pero que representan sus características distribucionales. Incluso sus resultados han sido tan realistas en generar imágenes, videos y audio, que uno de los campos de investigación actuales consiste en la identificación de fake data (imágenes, video, audio) mediante algoritmos, dado el potencial de usos peligrosos de esta tecnología.

Es posible que el uso de técnicas de data augmentation, pueda mejorar el desempeño de los modelos que usan series de tiempo, en cuanto a su desempeño por fuera de los datos de entrenamiento y validación. No obstante, las propiedades de los datos seriales, impiden que transformaciones simples, como las aplicadas a las imágenes, puedan ser usadas para aumentar datos. Se requiere entonces construir modelos generadores de datos, que puedan capturar las características de las series de tiempo financieras, y de esta manera se puedan usar como mecanismo para aumentar los datos disponibles para el entrenamiento de modelos de aprendizaje de máquina.

La propuesta de generar datos financieros sintéticos usando redes GAN, con periodicidad diaria e intradiaria, brinda mayor cantidad de datos que conservan las propiedades distribucionales de las series originales, pero a su vez brindan suficiente diversidad para poder mejorar el desempeño por fuera de los datos de entrenamiento de los modelos de machine learning y deep learning. Adicionalmente, tener entrenados modelos generadores de series de tiempo que cumplan simultáneamente las condiciones de capturar el comportamiento de las series originales y sin embargo posean la suficiente diversidad para generar realmente nuevos datos, permitiría su uso en modelos de optimización y control tradicionales, o su utilización con técnicas de aprendizaje reforzado para mejorar las decisiones de los usuarios financieros o de robots.

Las figuras 1.1 y 1.2 muestran esquemáticamente como la mayor disponibilidad de datos generados de forma sintética podría cambiar el enfoque de entrenamiento y uso de modelos de aprendizaje automático en finanzas. La figura 1.1 muestra un proceso iterativo de generación de una cantidad *n* de modelos financieros con los mismos datos históricos. Pese a que haya una separación entre datos de entrenamiento y datos de backtesting, existe un gran riesgo de que se encuentren relaciones espurias entre variables o patrones no generalizables dado que los datos históricos se usan de forma reiterada hasta hallar un modelo adecuado.

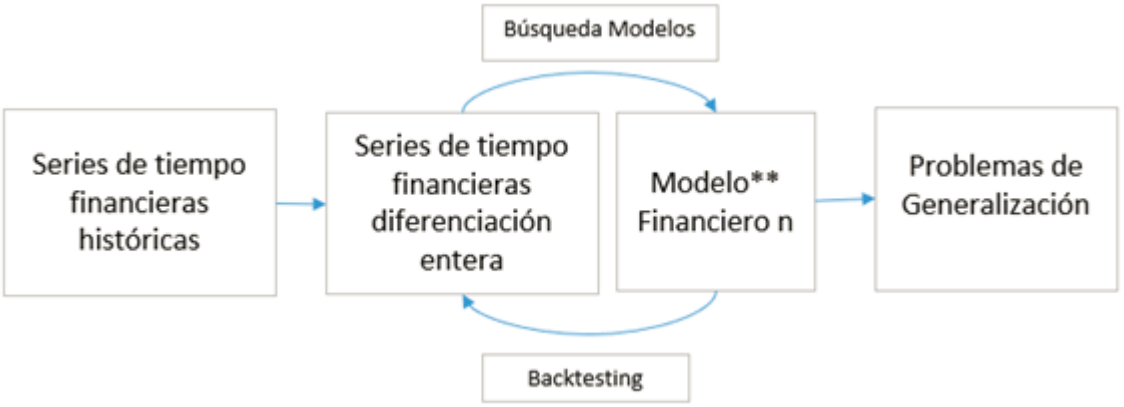

Figura 1.1: Backtesting con datos históricos - proceso iterativo

La figura 1.2 muestra a manera de ejemplo uno de los posibles usos de un buen generador de datos sintéticos para el entrenamiento de modelos financieros. En este caso se usan los datos generados con GAN para el entrenamiento y validación de modelos y una vez se tenga un buen candidato, se evalúa con un backtesting final cuyo objetivo es descartar o no el modelo pero en ningún caso mejorarlo. Este modelo debería tener mejor desempeño por fuera de la muestra por cuanto utilizó mayor cantidad de datos en su entrenamiento y principalmente por que no se calibró usando varias veces la series de tiempo histórica.

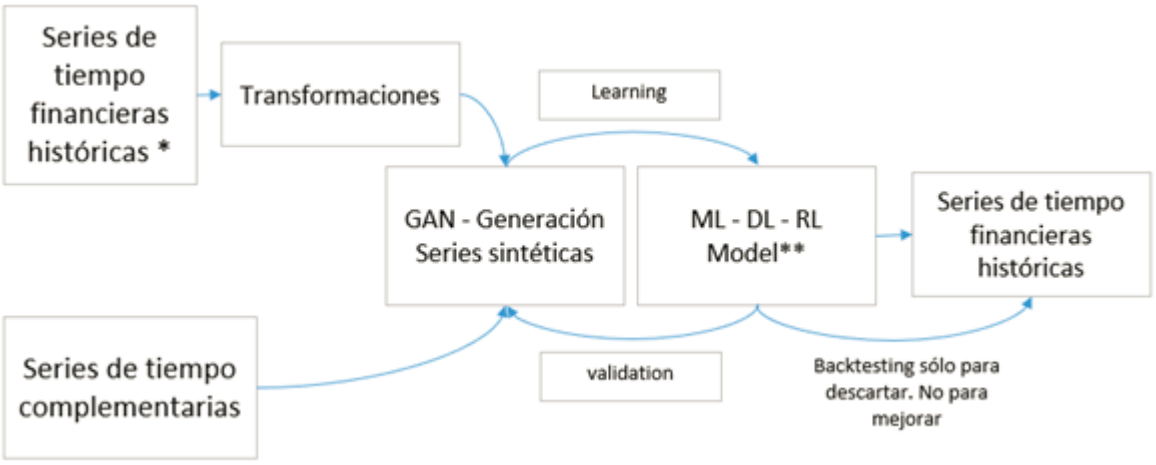

Figura 1.2: Backtesting con datos históricos - proceso iterativo

1.1 APORTE ACADÉMICO

- La utilización de series financieras con diferenciación (integración) fraccional para que el proceso generador del GAN pueda beneficiarse simultáneamente del balance entre memoria y estacionariedad para el "Data Augmentation". Usualmente en finanzas se utilizan series con diferenciación entre e igual a uno, al transformar los precios a rendimientos con lo cual se obtienen series estacionarias sin memoria.

- Modelos generadores GAN para series de datos intradiarios para generación y evaluación de métricas usadas en la literatura de microestructura de mercados financieros.

1.2 APORTE PARA PRACTITIONERS

Las series de tiempo sintéticas le brindan la posibilidad de utilizar modelos de Machine Learning, Deep Leaning o Reinforcement Learning con, potencialmente, mejor desempeño por fuera de la muestra, al reducir el riesgo de data snooping u overfitting.

2. CARACTERÍSTICAS DE LAS SERIES DE TIEMPO FINANCIERAS

2.1 SERIES DE TIEMPO FINANCIERAS DIARIAS

Gooijer *et al.* [6] realizaron un compendio de las técnicas históricamente usadas para predicción de series de tiempo, la mayoría se sustenta en la metodología Box-Jenkins [7] que propone como uno de sus fundamentos la transformación de las series de tiempo para buscar la estacionariedad. En la modelación de series de tiempo financieras usualmente se realiza una transformación de los precios para convertirlos en rendimientos.

López de Prado [5] manifiesta que, prácticamente toda la literatura de series de tiempo financieras, se sustenta en la premisa de transformar las series de tiempo de precios, no estacionarias, a series de tiempo estacionarias mediante transformaciones, como pueden ser los rendimientos, para obtener propiedades estadísticas de invariabilidad de parámetros. Para López, existe un dilema en las series financieras entre la estacionariedad y memoria, dado que lo que hace la serie de tiempo no estacionaria es la presencia de memoria por lo tanto el precio de hacer la serie estacionaria es la pérdida de memoria de la serie de tiempo y por lo tanto capacidad predictiva. Propone el uso de transformaciones fraccionales con el fin de obtener series de tiempo financieras estacionarias desde la perspectiva del test ADF (Augmented Dickey Fuller Test), pero perdiendo la mínima cantidad de memoria posible. De esta manera la estacionariedad no es una propiedad de las series de tiempo financieras sino el resultado de una transformación que facilita la modelación estadística.

En este estudio, para datos con periodicidad diaria, se realizará la simulación de las series de tiempo financieras en la forma de rendimientos, por ser la manera más fácil de contrastar los resultados con los métodos existentes. Sin embargo, también se realizarán transformaciones fraccionales en el límite del test ADF con el fin de evaluar si existe alguna ventaja al conservar memoria en la generación de datos sintéticos.

El estudio de series de tiempo de los retornos de las acciones, ha sido objeto mucha investigación en economía y finanzas, usándose una gran diversidad de modelos, tanto discretos como continuos, donde cada uno de ellos lleva implícitos una cantidad de supuestos que, en muchos casos, no son consistentes con las características distribucionales de los rendimientos observados. A continuación se muestran algunas de las propiedades consideradas hechos estilizados de los rendimientos financieros con frecuencia diaria [8][9][10]:

- Alto exceso de curtosis
- Colas pesadas respecto a una distribución normal.
- Los retornos de los índices tienen menor volatilidad que las acciones individuales
- Los rendimientos mensuales presentan una mayor volatilidad que los rendimientos diarios.
- Asimetría
- Clusters de volatilidad
- La volatilidad de los retornos está negativamente correlacionada con los retornos del activo. "Leverage Effect".
- Decaimiento de la autocorrelación de retornos como función de los rezagos.
- Correlación entre retornos y volumen

En los datos diarios usualmente se tienen a disposición el precio de apertura, precio de cierre, precio más alto, precio más bajo, volumen y cantidad de negociaciones, aunque en la literatura financiera se usa principalmente los precios de cierre.

2.2 SERIES DE TIEMPO FINANCIERAS INTRADIARIAS

En los datos intradiarios no existe consenso respecto a los hechos estilizados de los rendimientos. [11] Dacorogna *et al.* describe que en alta frecuencia, los precios están sujetos a efectos de microestructura, como el conocido fenómeno del rebote entre los niveles bid-ask. En esta escala los efectos de la formación de precios prevalecen sobre efectos que se observan en bajas frecuencias, también se encuentra que los rendimientos presentan colas pesadas de manera creciente con la frecuencia de las transacciones.

Los datos intradiarios pueden ser del tipo "tick-by-tick", en cuyo caso se tiene un registro de cada negociación individual, en el tiempo en el cual sucedió. Los datos registrados son la cantidad transada y el precio de negociación, en algunos mercados se cuenta también con los tipos de participantes en la negociación. Los datos intradiarios pueden contener también el valor de las cotizaciones en el momento de la transacción (bid-ask Price) y la profundidad (cuantas acciones están dispuestos a comprar o vender los mejores bid-ask).[12]

También se usan datos intradiarios con agregaciones de escala de 1 minuto o 5 minutos. López [5] propone realizar agrupaciones no sólo por tiempo sino con otro tipo de proxy de la llegada de información como son:

- Agrupación por volumen transado: un muestreo como función de la actividad de trading permite obtener retornos más cercanos a la distribución normal. Comportamiento importante pues muchos métodos estadísticos asumen este supuesto.[12]

- Agrupación por volumen en unidades monetarias

Cartea *et al* [13], en su libro, Market and High Frequency Trading, tienen un amplio desarrollo de las propiedades de las series de tiempo financieras de alta frecuencia de acuerdo con la teoría de la microestructura de mercado y según las interacciones de tres tipos de participantes en los mercados:

- Traders de liquidez o ruido: (incluye aquí a los inversionistas que se fundamentan en fundamentales que no toman decisiones por eventos de corto plazo).

- Traders con información: este tipo de inversionista obtiene utilidades por incorporar información al precio.

- Market Makers: inversionistas profesionales que facilitan los intercambios de un activo.

Cartea *et al* [14] muestra que la variable Order Flow (volumen de compras menos volumen de ventas) es significativa para explicar los rendimientos intradiarios de una acción, lo cual es consistente con los resultados de Agudelo *et al* [15] quienes toman el Order Flow como base para el cálculo del PIN intradiario en los mercados latinoamericanos que a su vez se sustentan en los modelos teóricos de Glosten y Milgron y Kyle[16,17].

Otra característica de los retornos intradiarios es la autocorrelación negativa que implica un proceso de reversión a la media que se puede explicar por un fenómeno conocido como bid-ask bounce. [18] Aldridge expone las siguientes cinco características de los datos de alta frecuencia: 1) Datos voluminosos (los datos de transacciones de alta frecuencia en un día equivale a 30 años de datos diarios). 2) Precios sujetos al rebote del bid y el ask (a diferencia de los datos diarios donde se maneja un solo precio de cierre). 3) No se observa normalidad en los rendimientos ni log-normalidad en los precios por lo que los modelos que asumen estas distribuciones no aplican. 4) La llegada de datos son asíncronos. 5) Los datos no contienen las etiquetas de la dirección de la transacción (si se originó en una compra o una venta). Respecto a esta última característica [18] Aldridge enumera cuatro de los métodos ampliamente usados para estimar la dirección de la transacción:

- La regla del tick: se compara el precio de la transacción con el precio de la transacción precedente y se etiqueta la transacción como Uptick (precio de la última transacción mayor que el precio de la anterior), Downtick (precio de la

última transacción menor que la anterior), Zero-uptick (si el precio no se mueve pero la última transacción con cambio de precio fue Uptick), Zero-downtick (el precio no se mueve pero la última transacción con cambio de precio de Downtick)

- La regla de la cotización: Una transacción es una compra (venta) si el precio de transacción está sobre (debajo) el promedio del bid-ask. Si la transacción está exactamente en el promedio no se clasifica.

- La regla de Lee-Ready [19]: este método primero clasifica las transacciones con la regla de la cotización y las no clasificadas se les aplica la regla del tick.

- El método de BVC (Bulk Volume Classification) [20]: mediante un modelos probabilístico se asigna una dirección a una agrupación de transacciones por volumen o tiempo.

La información de transacciones financieras de alta frecuencia usualmente se divide en dos niveles. El nivel I incluye el mejor precio bid, el mejor precio ask, el tamaño del mejor bid, el tamaño del mejor ask y el tamaño y precio de la última transacción. El nivel II incluye todos los cambios en el libro de órdenes.

Otras variables usualmente utilizadas en los modelos de microestructura son:

- Spread Observado: mide el costo de ejecutar una transacción de forma agresiva. Costo de la inmediatez.

- Precio Medio: estimador del precio real del mercado. $St = \frac{1}{2}\ (bid\ promedio\ +\ ask\ promedio)$

- Spread Efectivo: dos veces la diferencia en valor absoluto entre el precio pagado y el precio medio

- Intraday Depth: logaritmo de la cantidad de acciones posteadas en el bid y en el ask

- Microprice: es un indicador de desbalance de órdenes. $Pmicro = Pb\left(\frac{Qa}{Qa+Qb}\right) + Pa\left(\frac{b}{Qa+Qb}\right)$, donde Pa y Pb son los precios ask y bid. Qa y Qb es la cantidad de acciones en el ask y el bid.

- Interarrival times: tiempo entre movimientos del bid o del ask.

- Limit Order Order Imbalance: es la razón entre el desbalance de volumen cotizado y el total del volumen cotizado para un nivel de profundidad del libro de órdenes.

[21]Vanstone *et al* enumera varios test para aplicar a las series de tiempo financieras intradiarias los cuales se muestran a continuación:

- Test de Eficiencia de Mercado: Test de razón de varianza
- Test de Aleatoriedad: Nonparametric run test
- Test de dependencia serial: ACF (Ljung-Box test) y PACF

Sirignano y Cont [22] mediante el uso del Deep learning crearon modelos para mejorar el entendimiento sobre la formación de precios en los mercados financieros usando datos intradiarios de 500 acciones del NASDAQ, concluyendo que los modelos de Deep leaning superan a los modelos lineales en términos de Accuracy pero, principalmente, encontraron que existe relación de estacionariedad de largo plazo entre el flujo de órdenes y los cambios de precio de las acciones. Además, los modelos de Deep learning encontraron patrones universales sobres todas las acciones que les permitió generalizar bien. Otra conclusión importante es que existe un patrón de dependencia entre la dinámica del precio y la historia de flujo de órdenes que no depende solamente del estado reciente del libro de órdenes límite sino también sino de su historia (horas). Esto es otro argumento para utilizar modelos GAN en la generación de datos sintéticos pues podría capturar relaciones entre variables que usualmente un modelador con técnicas tradicionales, podría pasar por alto.

3. CARACTERIZACIÓN DE LOS MODELOS GAN

3.1 MODELOS GAN

3.1.1 Modelo GAN Original

Ian Goodfellow [23] publicó en 2014 su paper titulado “Generative Adversarial Networks”, y fue el inicio de los modelos GAN. Básicamente un modelo GAN se compone de dos sub modelos, un modelo generador y un modelo discriminador.

El modelo generador toma un tensor aleatorio (usualmente gaussiano) como fuente de datos, desde un espacio aleatorio sin ningún significado antes del entrenamiento. Pero después del entrenamiento, estos puntos antes aleatorios, se convierten en puntos con significado en el espacio de dominio del problema a simular y se denomina espacio latente, el cual contiene variables importantes para el problema pero que son inobservables. De esta manera después del entrenamiento el modelo generador es capaz de transformar variables aleatorias para generar nuevas muestras. El modelo discriminador toma como entrada un ejemplo del problema de dominio sea éste real o simulado y lo clasifica con una variable binaria como real o ficticio.

Se entrena luego en una sola red ambos modelos, intentando simultáneamente, mejorar cada uno de ellos, y se inicia un proceso de búsqueda de equilibrios en un juego de adversarios donde el modelo generador intenta crear datos que puedan “engañar” al modelo discriminador y a su vez, el modelo discriminador, aprende a diferencia los datos reales de los ficticios. Al finalizar el entrenamiento se separan los modelos generador y discriminador, y se tiene entonces por un lado un buen modelo generador de datos y por otro un buen modelo discriminador. Un aspecto muy importante a resaltar es que el modelo generador nunca tuvo acceso a los datos reales y su único mecanismo de aprendizaje es mediante su interacción con el discriminador.

El modelo generador se puede representar así [24]:

$G: G(z) \rightarrow R^{|X|}$ donde $z \in R^{|Z|}$ es una muestra del espacio latente $X \in R^{|X|}$

y el modelo discriminador: $D: D(X) \rightarrow (0, 1)$

La figura 3 ilustra la estructura de una red neural GAN.

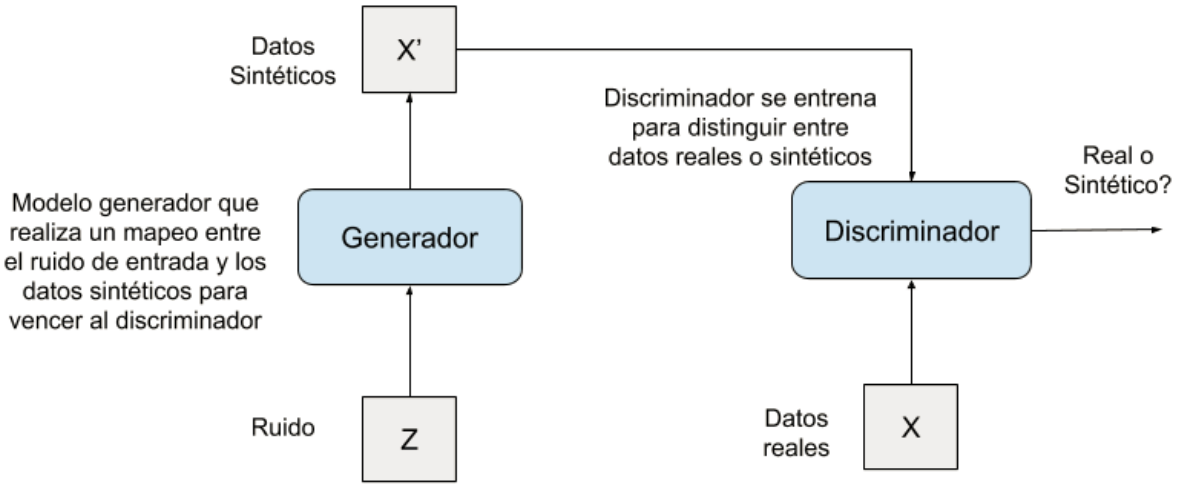

Figura 3.1: Esquema de un modelo GAN Básico

Después de la aparición del paper seminal del primer GAN, han aparecido gran castidad de modificaciones al mismo con el fin de resolver una variedad de problemas diferenes. Salehi et al [25,26] realizan una clasificación de las diferentes variantes desarrolladas de GAN en tres categorías con el fin de agruparlas según la técnica utilidad. La tabla 3.1 muestra la clasificación propuesta por Salehi para los diferentes tipos de GAN. A continuación se explican de modo resumido algunas de estas técnicas. Es importante para el modelador que entienda en qué casos es preferible usar uno respecto a otro.

Tabla 3.1: tipos de modificaciones a los modelos GAN. fuente: [26]

Tipo modificación GAN	Ejemplo de Modelos GAN
Basada en convolución	DCGAN
Basada en variables condicionales	CGAN, infoGAN, ACGAN, Semi-Supervised GAN.
Basada en Autoencoders	AAE, BiGAN, ALI, VAE-GAN.
Basada en optimización de la función objetivo	Unrolled GAN, f-GAN, Mode-Regularized GAN, Leat-Square GAN, EBGAN, WGAN, WGAN-GP, WGAN-LP

3.1.2 Modelo DCGAN

[27]Radford et al. crearon el modelo DCGAN (Deep Convolutional Generative Adversarial Networks) que ha tenido mucha aplicación en la generación de imágenes realistas. En esta arquitectura se usan capas de convolución tanto en el modelo generador como en el modelo discriminador. Específicamente en el modelo

generador se usan capas llamadas de convolución transpuesta (algunas veces llamada deconvolución) que realiza operaciones inversas a la convolución y por ello amplifica el tamaño de las imágenes. La siguiente tabla 3.2 muestra los cambios incorporados en el modelo DCGAN respecto a capas estándar de convolución del modelo original GAN. [26]

Tabla 3.2: Cambios en las capas convolucionales en el DCGAN. fuente: [26]

Característica	**CNN estándar**	**CNN modelo generador del GAN**	**CNN modelos discriminador del GAN**
Capa de reducción de dimensionalidad	Pooling Layers	Convoluciones con Fractional-Strided	Convoluciones con Stride
Batch Normalization	No se requiere	Se requiere	Se requiere
Función de Activación	Diversas funciones de activación	ReLU en capas intermedias y Tanh en la final	Leaky ReLU en todas la capas
Capa FCL	Sí	No	No

3.1.3 Modelos GAN con Variables Condicionales

Una de las limitaciones del modelo GAN original, es que no se tiene control de la forma como el generador construye los datos sintéticos, pues toma datos aleatorios desde el espacio latente. Se han creado varias arquitecturas para tener mayor control sobre los datos generador

Mirza y Osidero [28] propusieron el modelo de GAN condicional (cGAN) en el cual condicionan a los modelos generador y discriminador respecto a una variable externa y de esta manera condicionar los resultados de la red neuronal obteniéndose mayor control. La variable externa que condiciona al modelo puede ser del tipo categórica, numérica o de otro tipo. La figura 4, extraída del artículo de Mirza et al. muestra como la variable y condiciona tanto el modelo generador como el discriminador.

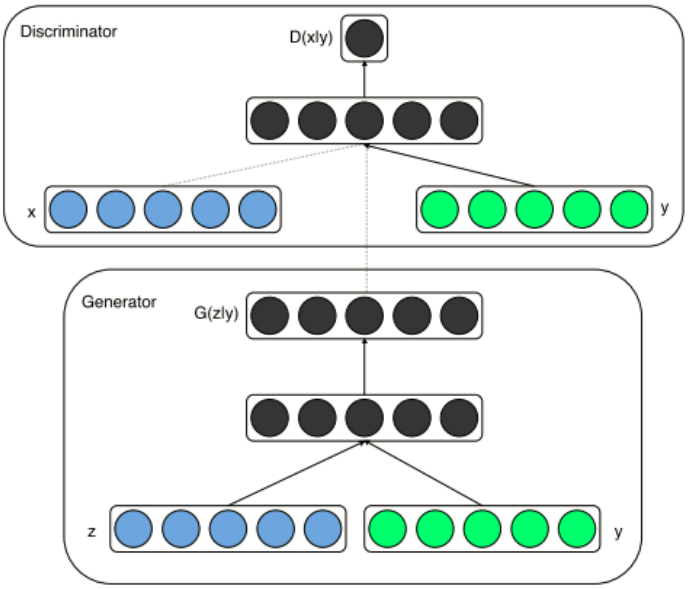

Figura 3.2: Esquema de modelo CGAN. Fuente: [28]

Los modelos cGAN permiten tener control sobre los datos sintéticos generados mientras que en el GAN original no se contaba con dicho control. A manera de ejemplo se puede citar el caso típico de la generación de dígitos usando los datos de MNIST y usando una arquitectura cGAN, se puede adicionar como variable con información adicional un dígito del 0 a 9, y así el modelo queda condicionado para generar dicho dígito.

Los modelos Information maximizing GAN (infoGAN)[29] se desarrollaron para tener mayor control sobre el proceso generador y se sustenta en la maximización de la información mutua. Información mutua consiste en la cantidad de información que puede ser inferida de una variable mediante la observación de otra variable. En el modelo GAN original, para generar modificaciones en los datos sintéticos de forma controlada se tendrían que generar combinaciones lineales de puntos en el espacio latente que a su vez generaron datos con las características a combinar, sin embargo puesto que los puntos provienen de ruido y el mapeo entre la dimensión latente y los ejemplo generados es compleja, en realidad no se tiene control sobre la forma en la que el generador puede generar datos con características deseada. en la figura 5 se observa como el infoGAN incorpora entonces una segunda variable o fuente de información en el modelo generador (latent code), al igual que lo hace el cGAN, pero en este caso el dato en lugar de ser arbitrario, es estimado por el modelo discriminador, al que se le acopla una red neuronal densa, llamada Q Network, la cual trata de predecir la distribución de este código latente (latent code).

Finalmente al manipular estos códigos latentes, que son más fácilmente interpretables que el vector de ruido, se pueden identificar que tipo de característica de los datos creados se controla con cada componente del vector C. La figura 3.3 esquematiza un modelo infoGAN.

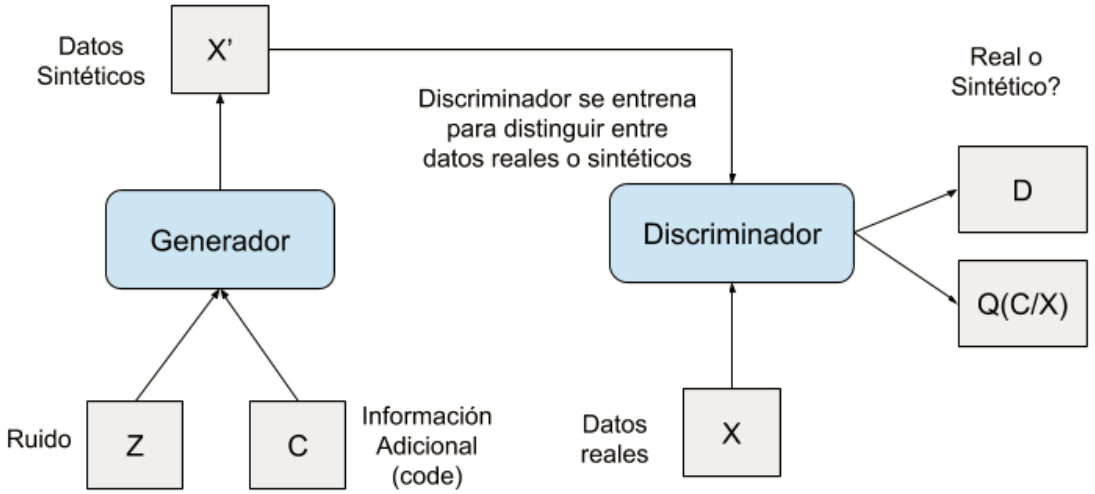

Figura 3.3: esquema de un modelo infoGAN

Los modelos Auxiliary Classifier GAN [30] (ACGAN) son una extensión de los CGAN en el que el discriminador, además de clasificar los datos reales de los sintéticos, también predice la clase C en lugar de usarla como un dato de entrada. Pero el dato C también es un dato de entrada del modelo generador. En los semi-supervised GAN [31] (SGAN) se utiliza un conjunto de datos con con etiquetas y otros sin etiquetas. Los datos sin etiquetas se utilizan para la discriminacion entre real o sintético, los datos etiquetados se usan para optimizar mejorar el proceso de entrenamiento del modelo. Esta última modelación es útil cuando se cuenta con cantidad limitada de datos etiquetados.

La figura 3.4 muestra un esquema comparativo de los cuatro modelos GAN condicionales que se han mencionado. [32]

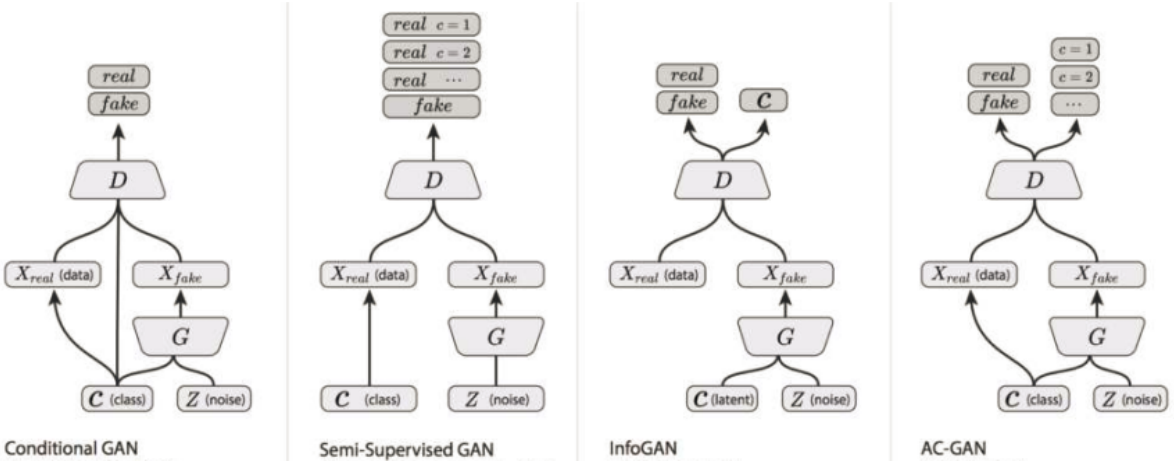

Figura 3.4: Comparativo de modelos GAN condicionales. Fuente:[32]

3.1.4 Modelos GAN con Autoencoders

En los modelos GAN condicionales se busca mejorar el control sobre las imágenes o datos creados por el generador. El uso de autoencoders con GAN pretende mejorar el entendimiento y mapeo del espacio latente de la capa cuello de botella de los autoencoders.

Los adversarial autoencoders [33] (AAE) son una combinación de un modelo GAN con un autoencoder. En este caso el modelo GAN se usa como un regularizador para mejorar la representación latente del autoencoder. [25] Donahue et al. en su artículo Adversarial Feature Learning, propusieron el modelo Bidirectional GAN (BiGAN) con el que buscan identificar un mapeo inverso al del GAN original, es decir, obtener la representación latente partiendo de un dato real y este enfoque sirve para el aprendizaje de características de los datos. [34] Las redes tipo adversarially learned inference (ALI) también realizan un doble mapeo desde la dimensión latente hacia la dimensión de datos y en sentido contrario. [35] Los VAE-GAN buscan encontrar un dimensión latente en el modelo del variational autoencoder de tal manera que se pueda modelar adecuadamente las características de los datos en la dimensión latente y usa el modelo GAN para que este sirva como medida de similaridad más robusta que las medidas típicas de error de reconstrucción. En el caso de las imagenente por ejemplo, una rotación de una imagen arroja un gran error de reconstrucción desde la perspectiva del error cuadrático, pero para el modelo GAN no existe tanta diferencia entre las dos imágenes.

3.1.5 Modelos GAN con Modificación de la Función Objetivo

Este grupo de modificaciones se enfocan en disminuir dos problemas comúnmente encontrados en los modelos GAN:

- Modo de colapso: cuando el modelo generador no tiene diversidad y genera datos repetidos.
- Proceso de entrenamiento inestable: dificultad en la obtención de un equilibrio de Nash.

En el caso del unrolled GAN [36] se redefine la función objetivo del generador respecto a una optimización “unrolled” del discriminador y se mejora el desempeño de la red en cuanto al modo de colapso y la estabilización. [37]Wasserstein GAN también es una variación muy usada para mejorar la estabilidad del modelo, en este caso se usa la distancia Wasserstein-1, o Earth-Mover (EM) y se muestra que tiene un mejor comportamiento de los gradientes en la optimización. [38] Petzka et al. introdujeron una modificación al WGAN con restricciones de Lipschitz en la optimización de la minimización de la distancia entre el modelo y la distribución de probabilidad empírica (WGAN-LP).

3.1.6 Otros Modelos GAN

[39]Zhu et al crearon el Cycle GAN el cual se usa para traducción o transformación de imágenes a imágenes, aun cuando no se tenga datos emparejados de imágenes para el entrenamiento. Para este fin el objetivo del aprendizaje es un mapeo del modelo generador entre las imágenes X hacia las imágenes Y, de tal manera que la distribución de imágenes de $G(X)$no sea distinguible de la distribución de imágenes de Y. También se realiza un mapeo inverso desde Y a X y se introduce un ciclo de consistencia de la pérdida de tal manera que $F(G(X)) \approx X$ y $F(G(Y)) \approx Y$. El modelo Cycle GAN fue un avance respecto al modelo Pixels-to-Pixels [40] que ya había mostrado la flexibilidad de los modelos GAN para diversos usos que requieren la transformación de imágenes.

Los modelos GAN también son utilizados para tareas de super resolución de imágenes[41,42]. En este sentido se usan para predecir cómo sería la imagen en caso de requerirse mayores detalles.

[43]Petroski et al. ha propuesto un nuevo enfoque para modelos generadores, llamado GTN (Generative Teaching Networks) que en lugar de buscar la competencia entre dos modelos, busca su cooperación. Es una línea de investigación prometedora para la generación de datos sintéticos.

3.2 METODOLOGIA PARA EVALUACION DE GAN

Borji [44] presenta 24 métodos cuantitativos para tratar de evaluar la calidad y diversidad de los datos generados con GAN y posteriormente plantea los motivos por los cuales cada uno de ellos no es apropiado. Sin embargo no hay consenso respecto a una métrica universal, pero los métodos más ampliamente usados son Inception Score y Fréchet Inception Distance.

El Inception Score [45] es una métrica para la evaluación de la calidad y diversidad de las imágenes generadas por un modelo GAN. El autor de esta medida la creó como una alternativa a la subjetividad de la evaluación humana. El modelo usa una red neuronal Inception V3 pre entrenada para calcular la probabilidad de que las imágenes generadas pertenezcan a una categoría, esta probabilidades se resumen en el Inception Score con la premisa de que las imágenes clasificadas en la misma categoría deben tener una distribución de probabilidad $P(X/Y)$ con baja entropía.

La medida Fréchet Inception [46] Distance calcula la distancia entre vectores de características de las imágenes reales y las generadas. La métrica resume las distancia entre los vectores de características de incepción entre las imágenes reales y las sintéticas.

Las métricas de desempeño en la literatura se concentran en la evaluación de imágenes y no existe un método para la evaluación de la calidad de series de tiempo generadas.

3.3 USO DE MODELOS GAN PARA DATA AUGMENTATION

Frid-Adar [47] crearon un modelo DCGAN para crear imágenes sintéticas de lesiones de hígado. Se realizó un modelo independiente para cada tipo de lesión, logrando un incremento del 7% en las métricas de desempeño de clasificación de lesiones respecto a las técnicas tradicionales para aumentar datos. Shrivastaba *et al.* [48] propusieron un modelo que combina el aprendizaje no supervisado con el aprendizaje con datos simulados (simGAN) con el fin de entrenar modelos de machine learning. Su enfoque pretende mejorar el realismo de las imágenes generadas por un simulador usando imágenes reales sin etiquetas. Su método tiene como insumo un dato o imagen simulada, que proviene de un modelo externo y se mejora la imagen con un modelo GAN refinador que es una red neuronal con función de pérdida adversaria, similar a los GAN, y como salida se tienen imágenes que no son distinguibles de las reales.

Data Augmentation Generative Adversarial Network [49] (DAGAN) fue creado para mejorar la diversidad de los datos aumentados frente a las técnicas tradicionales de data augmentation de imágenes. La figura 3.5 muestra cómo funciona el DAGAN.

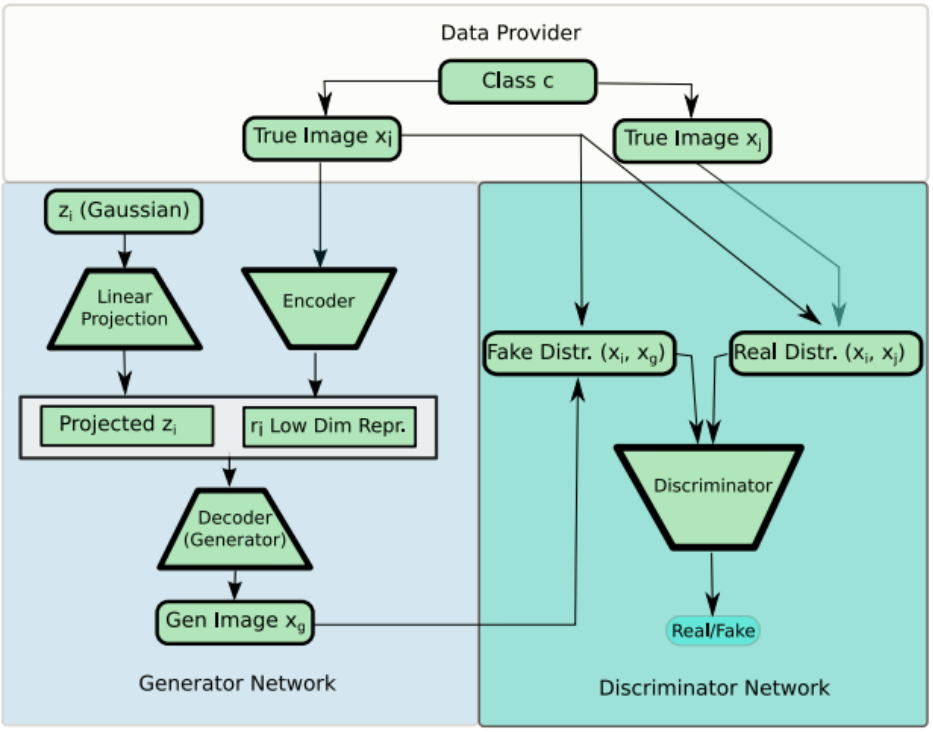

Figura 3.5: Esquema del modelo DAGAN. Fuente: [49]

La parte izquierda de la figura 6 corresponde a la parte generadora del DAGAN. Se toma una imágen de la clase C y se transforma a una dimensión latente a través de un encoder. Simultáneamente se toma un vector aleatorio Zi y se proyecta a la misma dimensión latente y se concatena con la representación de la imagen. Este último tensor se transforma a través de un decoder en una imagen que es una reconstrucción de la imagen original pero tiene elementos aleatorios para incorporar aleatoriedad. La parte izquierda de la figura 6 corresponde al modelo discriminador que toma como insumo sintético a la imagen original Xi y la imagen sintética, y como dato real la misma imagen real Xi y una imagen real Xj de la misma clase. Al entrenar el modelo se producen imágenes Xg que pertenecen a la clase C pero con suficiente diversidad para que se considere data augmentation.

Motamed y Khalvati [50] se inspiraron en el modelo DAGAN para crear su modelo Inception Augmentation GAN (IAGAN) con el cual mejoraron el AUC de 0.83 a 0.84 mediante data augmentation de imágenes de rayos X de neumonía. Este modelo incorpora dos bloques residuales en la parte generadora del modelo, además de incorporar imágenes con anotaciones como entrada al modelo y una capa de atención para la localización de las lesiones.

Lee et al. [51] usaron seis modelos tipo GAN para realizar Data Augmentation de imágenes satelitales de meteorología. Se usaron los modelos GAN, CGAN, DCGAN, InfoGAN, Least Square GAN (LSGAN) y WGAN-GP. El modelo WGAN-GP resultó ser más eficiente para la tarea pues le tomó 8 minutos el entrenamiento, frente a 17 horas del InfoGAN y un rango de 7 a 12 horas de los demás modelos.

3.4 USO DE MODELOS GAN EN SERIES DE TIEMPO

3.4.1 Predicción de series de tiempo con GAN

Zhou et al. [10] utilizaron un modelo GAN con redes tipo LSTM en la parte generadora, usando como datos de entrada además del precio de las acciones, varios indicadores técnicos tradicionales, con el fin de realizar la predicción de precios con un horizonte de 1 minuto. Zhang et al [52] también utilizan un modelo GAN con LSTM para la predicción de precios a nivel diario (estandarizados, lo que equivale a una transformación en búsqueda de estacionariedad). Utiliza como datos de entrada además del precio de cierre diario, el precio de apertura, el mayor precio, el menor precio, el volumen de negociación, turnover ratio (proporción de las acciones negociadas respecto al total de acciones en bolsa) y la media móvil de 5 días (una semana de negociaciones).

[53]Koshiyama et al. proponen el uso de cGAN (GAN condicional) para calibrar estrategias de trading con periodicidad diaria, usando el exceso de retorno como variable a simular. Realizan una comparación de resultados con el desempeño de modelos de bootstrap estacionario. Koshiyama et al. [54] construyeron un cGAN

para pronóstico de series multivariadas con un enfoque probabilístico y realizan pronósticos sobre series de consumo de electricidad y sobre series de tasas de cambio.

3.4.2 Generación de series de tiempo con GAN

Wise [54] desarrolló un tipo de modelo GAN llamado Quant GAN en el que utiliza redes temporales convolucionales TCN para capturar las dependencias de largo plazo de los rendimientos financieros, tales como los clusters de volatilidad y correlaciones seriales. Wise cataloga a su modelo Quant GAN como ubicado en el medio entre los modelos basados en datos, como las simulaciones Monte Carlo y los modelos con supuestos subyacentes sobre el comportamiento estocástico como los modelos de Black-sholes, Hestos o Levy. De Meer [55] utiliza un Wasserstein's GAN para crear series sintéticas de acciones individuales del S&P500 y el índice CBOE VIX. También modelan usando Relativistic GAN simultáneamente el índice S& 500 y el VIX, todo con periodicidad diaria.

Yoon propone [55] un modelo que combina la flexibilidad de los GAN con el control que permiten los modelos autoregresivos. Este modelo denominado timeGAN. El autor dice que la aplicación de los modelos GAN condicionales o incondicionales directamente a series de tiempo no es un enfoque correcto pues no garantiza que la red capture la dinámica temporal de las series de tiempo.

Hyand [56] crea un modelo denominado RGAN (Recurrent GAN) y un modelos RCGAN (Recurrent Conditional GAN) con aplicación en generación de datos sintéticos en series multidimensionales médicas. Utilizan redes recurrentes tanto en la parte generadora como discriminadora del GAN. [57] K.G. Hartman usa GAN para la generación de datos sobre señales de encefalogramas y para evaluar la calidad de los datos generados utiliza métricas como Inception Score, Frechet Inception Distance, Sliced Wasserstein Distance y muestran que se puede generar series de encefalogramas realistas.

Magnus et al. [58] construyeron su simulador de mercado de opciones financiera basado en modelos GAN usando arquitecturas de convoluciones univariadas y redes recurrentes. Proponen el uso del simulador del mercado de opciones para mejorar las estrategias de trading sobre estos activos.

Takashi et al. [59] utilizan modelos GAN para crear series financieras con un enfoque data driven y verifican que las series generadas conservan propiedades como las colas pesadas en la distribución de retornos, clusters de volatilidad, asimetría , e impredecibilidad lineal.

3.4.3 Otros usos de GAN con series de tiempo

Zijian utilizó redes GAN [60] para la imputación de datos sobre series de tiempo multivariadas obteniendo resultados que supera el desempeño de otros métodos. Marti [61] propuso un modelo al que denominó CorrGAN con el cual genera matrices de correlación sintéticas usando modelos GAN, las cuales pueden mejorar el backtesting de modelos de trading o portafolios financieros. Li et al [62] aplican metodología GAN para la detección de anomalías en series de tiempo multivariadas con aplicación a IoT.

4. TRANSFORMACIÓN DE DATOS E IDENTIFICACIÓN DE PROPIEDADES DE LAS SERIES FINANCIERAS HISTÓRICAS

Para la selección de acciones se tomaron las 50 acciones más transadas cada día de junio de 2020 en el mercado IEX y se obtuvo un consolidado de 101 instrumentos, de los cuales se eliminaron dos por ser nuevas acciones cotizando en las bolsa de Estados Unidos y otra que fue retirada de Nasdaq por fraude contable (Luckin Coffee Inc).

Los datos intradiarios son de libre acceso y se descargaron de http://iextraing.com con el fin de que las conclusiones puedan tener reproducibilidad y verificación. Los datos diarios se obtuvieron de http://finance.yahoo.com. A continuación se muestran las características de las series diarias e intradiarias de estas acciones.

4.1 DATOS DIARIOS

Los modelos GAN para datos diarios se aplicarán a los datos de cuatro acciones, para los cual se utilizó un clustering jerárquico sobre las correlaciones de los rendimientos de las acciones. Se observa entonces que con un punto de corte de la distancia cercano a un se pueden observar cuatro grupos de acciones con comportamiento similar.

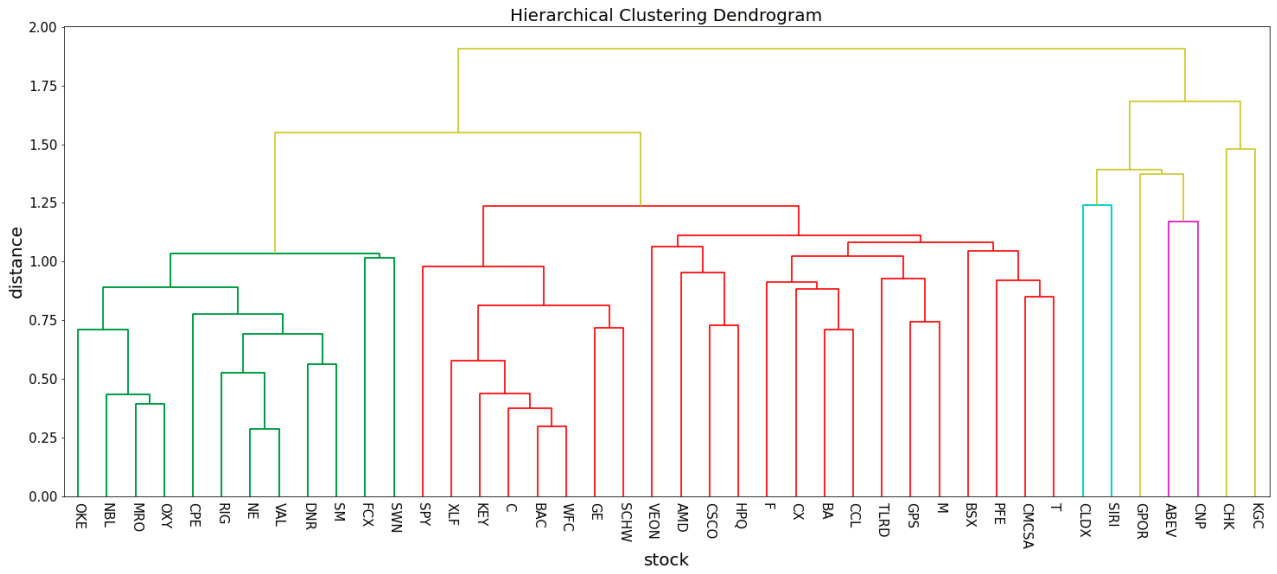

Figura 4.1: Dendrograma para conglomerado jerárquico de las acciones

A continuación se grafican dos pares de series de tiempo de precios de acciones pertenecientes cada uno a un mismo cluster. FCX y SWN pertenecen a un mismo cluster y HPQ y CSCO a otros. Se observa que tiene cada uno similaridad en el comportamiento, lo cual no implica que sean iguales. La figura 10 muestra de forma horizontal acciones de un mismo cluster.

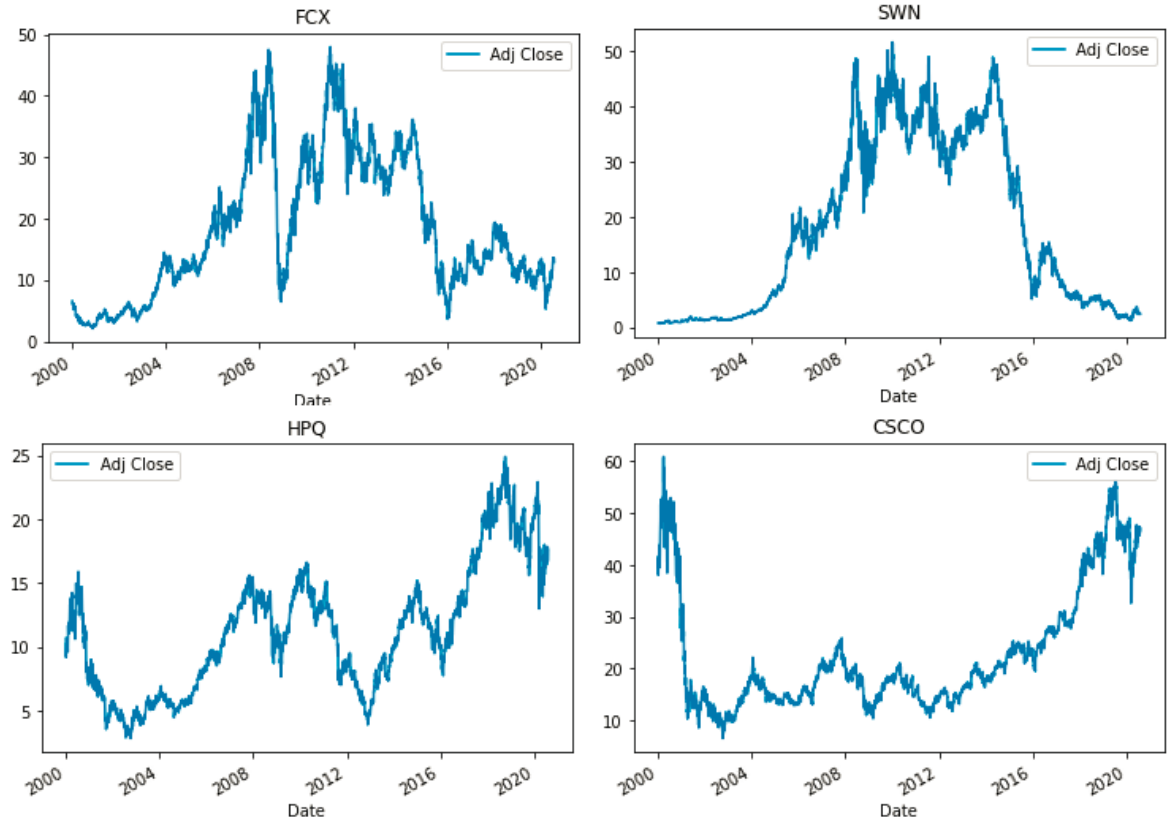

Figura 4.2: similaridad de comportamiento de precios de acciones de un mismo cluster.

Con esta metodología se eligen entonces las siguientes cuatro acciones para la modelación de los GAN para datos diarios:

- BAC (Bank of America Corp.)
- CNP (Centerpoint Energy Inc)
- HPQ (Hewlett Packard)
- SWN (Southwestern Energy Co)

4.1.1 Datos para Modelo DCGAN Sobre Rendimientos

Del total 99 acciones de las cuales se tiene datos diarios, se conservaron solamente las que estaban listadas desde el primero de enero de 2000, reduciéndose el número de acciones a 41 donde cada una tiene un total de 5.178 datos diarios de precio de apertura, precio de cierre, máximo y mínimo diario, además del volumen transado y precio ajustado.

Tabla 4.1: Resumen de momentos de los rendimientos diarios

MEDIDA	Media de Rendimientos	Desviación Estándar de Rendimientos	Asimetría de Rendimientos	Curtosis de Rendimientos
conteo	41	41	41	41
media	0.000494	0.031857	2.30217	120.28
desviación estándar	0.000367	0.013832	7.924985	484.83
mínimo	-0.000062	0.012567	-1.262095	4.54
25%	0.000236	0.024193	0.074853	9.26
50%	0.000482	0.027663	0.452535	18.46
75%	0.000684	0.036222	1.12904	44.29
máximo	0.00155	0.092412	49.091977	3101.55

De acuerdo con el resumen del cálculos de los primeros 4 momentos de los rendimientos diarios, se observa que los rendimientos diarios de las 41 acciones de la muestra presentan una media cercana a cero con poca dispersión entre los datos. En cuanto a la desviación estándar las acciones de la muestra presentan valores entre 1.25% diario a 9.2% diario, aunque este último valor de 9.2% corresponde a un dato atípico de la acción de Gulfport Energy Corporation (GPOR). En cuanto a la asimetría se presenta principalmente sesgo positivo (la distribución presenta cola pesada a la derecha) aunque también hay acciones con sesgo negativo.

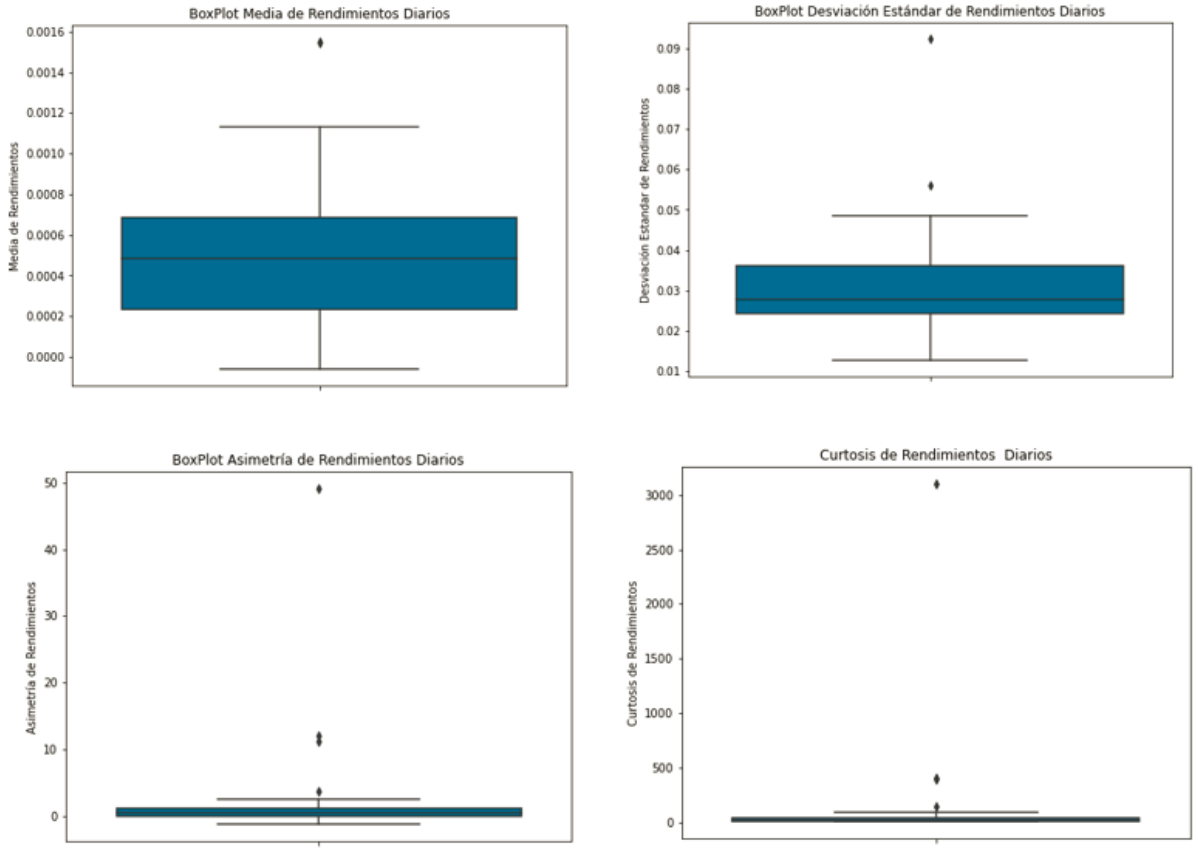

Figura 4.3: Boxplot de los momentos de los rendimientos diarios

Los valores de la curtosis van desde 4.54 hasta 3.101 por lo que es evidente un exceso de curtosis de los rendimientos.

La tabla 4 muestra las autocorrelaciones calculadas para los rendimientos de las 41 acciones de la muestra. Se resalta en gris las celdas que presentan autocorrelación significativa. En la fila final se puede verificar que efectivamente disminuye la cantidad de autocorrelaciones significativas con los rezagos. La figura 8 muestra un correlograma para el activo con ticker.

Se concluye que en general se pudo verificar que los rendimientos diarios de las acciones de la muestra, cumplen con los hechos estilizados esperados para estos datos.

Tabla 4.2: Autocorrelaciones de rendimientos diarios

TICKER	REZAGO									
	1	2	3	4	5	6	7	8	9	10
ABEV	-0.0248	-0.0500	0.0117	-0.0285	-0.0016	-0.0033	0.0136	0.0137	0.0224	-0.0002
AMD	0.0025	0.0281	-0.0206	0.0147	-0.0036	-0.0395	0.0410	-0.0059	-0.0020	0.0032
BA	0.0318	0.0612	0.0040	-0.0475	-0.0562	-0.0375	0.0039	0.0286	0.0391	-0.0183
BAC	-0.0185	0.0300	-0.0361	-0.0107	-0.0803	-0.0379	0.0249	-0.0161	0.0386	0.0626
BSX	-0.0315	-0.0307	-0.0076	0.0044	-0.0030	-0.0140	-0.0061	-0.0242	-0.0230	-0.0255
C	0.0247	0.0331	-0.0301	-0.0753	0.0129	-0.0343	-0.0109	-0.0055	0.0111	0.0025
CCL	0.0039	-0.0234	0.0066	-0.0159	-0.0282	-0.0901	0.0617	-0.0166	0.0329	0.0153
CLDX	-0.0356	0.0296	0.0213	-0.0092	-0.0123	0.0169	0.0128	-0.0061	0.0199	0.0072
CMCSA	-0.0889	-0.0164	-0.0050	-0.0411	-0.0216	-0.0362	0.0079	-0.0088	-0.0132	-0.0025
CNP	-0.0444	-0.0423	0.0871	-0.0589	-0.0431	-0.0407	0.0169	0.0068	-0.0256	0.0397
CPE	0.0461	0.0136	-0.0252	-0.0289	-0.0566	-0.0109	0.0044	-0.0267	0.0168	0.0353
CSCO	-0.0742	-0.0331	-0.0042	-0.0014	-0.0221	-0.0077	0.0221	-0.0466	0.0055	0.0066
CX	0.0579	-0.0050	-0.0301	-0.0072	0.0165	-0.0134	0.0691	-0.0063	-0.0018	0.0161
DNR	0.0233	-0.0304	-0.0358	0.0175	0.0341	-0.0166	-0.0200	-0.0175	0.0232	0.0141
F	0.0183	0.0550	0.0271	0.0344	-0.0571	-0.0423	0.0193	-0.0224	0.0104	0.0316
FCX	-0.0070	-0.0013	0.0047	-0.0377	0.0021	-0.0275	0.0306	0.0029	0.0031	0.0267
GE	-0.0275	0.0139	0.0029	0.0080	-0.0326	-0.0090	0.0140	-0.0136	-0.0020	0.0297
GPOR	-0.0698	-0.0722	-0.0037	0.0354	-0.0303	-0.0218	0.0247	-0.0227	-0.0324	-0.0117
GPS	0.0009	-0.0438	0.0145	-0.0053	-0.0463	-0.0290	0.0156	0.0244	-0.0037	-0.0003
HPQ	-0.0311	-0.0016	-0.0037	-0.0109	-0.0246	-0.0249	0.0010	-0.0671	-0.0037	0.0021
KEY	-0.0679	-0.0171	-0.0758	-0.0498	-0.0098	-0.0776	0.0168	-0.0039	0.0166	0.0877
KGC	-0.1058	-0.0210	0.0113	0.0023	-0.0139	0.0050	0.0140	-0.0036	-0.0187	-0.0039
M	0.0275	-0.0531	0.0168	-0.0041	0.0275	-0.0416	0.0119	0.0525	-0.0073	-0.0118
MRO	-0.0403	-0.0453	0.0326	-0.0193	-0.0033	0.0082	0.0158	-0.0017	0.0280	0.0503
NBL	0.0153	-0.0131	0.0211	-0.0063	-0.0168	-0.0428	0.0196	-0.0230	0.0083	0.0058
NE	0.0660	-0.0304	-0.0598	0.0571	0.0106	-0.0408	-0.0127	-0.0383	0.0243	-0.0101
OKE	-0.0075	-0.0061	0.0252	0.0251	0.0020	-0.0224	0.0640	-0.0247	-0.0084	0.0287
OXY	-0.0281	-0.0326	0.0118	-0.0366	-0.0181	0.0001	0.0139	0.0062	0.0310	0.0404
PFE	-0.0355	-0.0847	0.0084	-0.0088	-0.0047	-0.0270	0.0005	0.0134	0.0016	0.0012
RIG	0.0558	-0.0333	-0.0202	-0.0402	0.0097	-0.0021	0.0365	-0.0293	-0.0006	-0.0011
SCHW	-0.0933	-0.0364	0.0355	-0.0401	-0.0217	-0.0453	0.0181	0.0057	0.0069	0.0115
SIRI	0.0150	-0.0625	0.0035	-0.0260	0.0243	0.0076	0.0429	-0.0216	0.0192	-0.0084
SM	0.0153	-0.0361	-0.0265	0.0018	0.0157	0.0284	0.0242	-0.0286	0.0194	0.0020
SPY	-0.0978	-0.0258	0.0071	-0.0246	-0.0141	-0.0360	0.0349	-0.0279	0.0246	-0.0094
SWN	-0.0114	-0.0440	0.0191	-0.0130	0.0053	-0.0059	-0.0053	-0.0154	0.0070	-0.0170
T	-0.0196	-0.0389	-0.0188	-0.0344	-0.0117	-0.0373	-0.0036	0.0045	0.0003	-0.0038
TLRD	0.0129	0.0194	0.0026	-0.0170	0.0002	-0.0104	-0.0040	0.0101	-0.0039	-0.0132
VAL	0.1678	-0.0660	-0.0461	-0.0064	0.0059	-0.0180	0.0158	-0.0154	0.0377	0.0031
VEON	-0.0251	-0.0397	0.0102	-0.0002	-0.0365	-0.0202	-0.0312	-0.0212	0.0063	-0.0181
WFC	-0.1070	0.0260	-0.0524	-0.0212	-0.0274	-0.0759	0.0453	-0.0174	0.0314	0.0472
XLF	-0.1057	0.0121	-0.0159	-0.0426	-0.0423	-0.0448	0.0455	-0.0102	0.0109	0.0370
Número de Autocorrelaciones	25	28	15	18	14	21	11	9	9	13

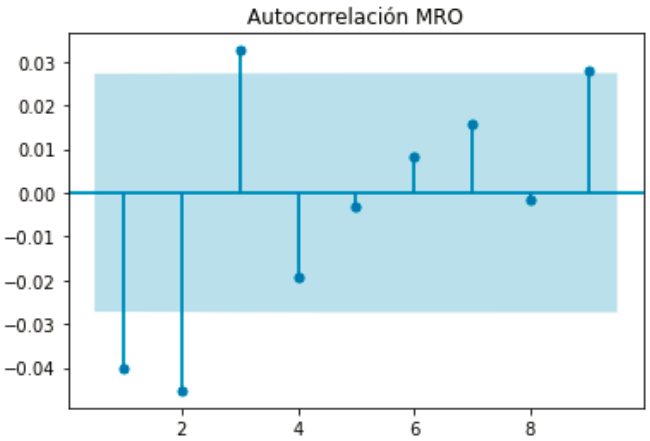

Figura 4.4: Autocorrelograma de los retornos diarios de la acción con ticker MRO

4.1.2 Datos Modelo DCGAN con Diferencias Fraccionales

Para evaluar la capacidad de los modelos GAN para generar series de tiempo que conserven simultáneamente las características de estacionariedad y memoria, se utiliza la transformación fraccional sobre las series de precios de las cuatro acciones seleccionadas. Se elige la metodología de transformación con ventana fija eliminando los coeficientes o pesos de transformación que no superen el valor de 0.003. Para el caso de un valor d=0.3 como parámetro de transformación fraccional se requieren entonces 72 datos históricos para transformar el dato 73. (En el caso de d=1, se requieren sólo dos datos coeficientes (1, -1) y por lo tanto dos datos de precios).

La figura 4.5 muestra que para un valor de d=0.9, sólo se requieren 12 coeficientes mayores a 0.003 para la diferenciación fraccional y no depende de la serie de datos.

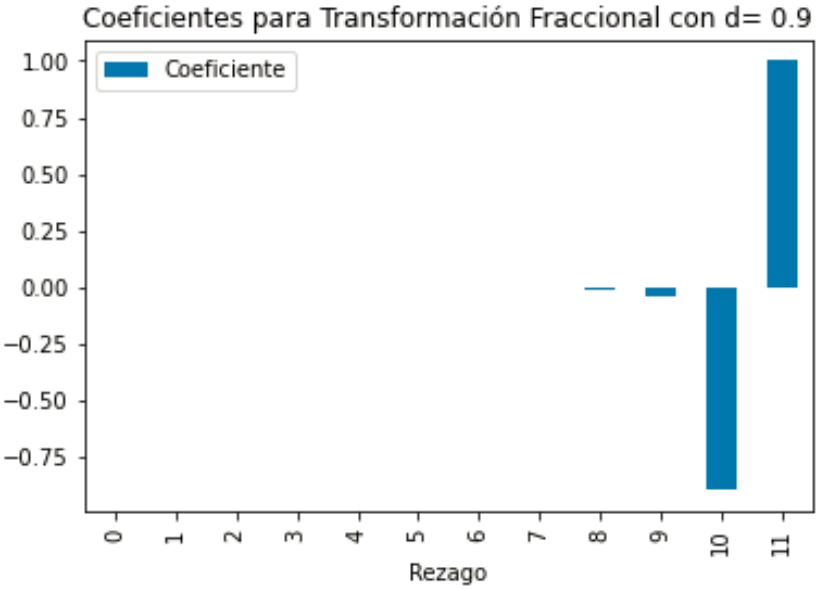

Figura 4.5: Coeficientes para transformación fraccional con d=0.9

Para cada una de las series de datos de precios de las cuatro acciones de la muestra se elige el parámetro óptimo d, seleccionando el mínimo valor de d que conserve la estacionariedad, medida con el estadístico Dicky Fuller Aumentado (adf). La tabla 4.3 muestra los resultados para la acción con ticker BAC. La primera columna contiene los valores de d que fueron evaluados, la segunda columna el valor del estadístico adf para la serie de datos obtenidos con el parámetro d y su respectiva valor p en la columna 3. La columna 4 muestra la reducción de datos válidos en la medida que en cada valor de d utiliza diferente cantidad de rezagos para la transformación de datos. La quinta columna contiene el valor de comparación para un límite de confianza del 95%. Finalmente la columna 6 contiene la correlación entre la serie original y la serie transformada. Se observa

que un valor de d=0.2, mantiene una alta correlación entre la serie original y la serie modificada (correlación = 0.94) y el valor de P y estadístico adf establecen que la serie transformada no contiene raíces unitarias, esto es, se puede asumir como estacionaria.

Tabla 4.3: Resultado de proceso de selección del parámetro d para diferenciación fraccional

d	Estadítico adf	Valor P	Datos Válidos	Límite para 95% conf	Correlación Serie Original y Transformada
0	-2.10	2.43E-01	5176	-2.86	0.97
0.1	-2.96	3.91E-02	5115	-2.86	0.96
0.2	-4.62	1.21E-04	5104	-2.86	0.94
0.3	-7.07	4.87E-10	5111	-2.86	0.91
0.4	-10.61	5.80E-19	5122	-2.86	0.86
0.5	-15.41	3.10E-28	5133	-2.86	0.78
0.6	-21.37	0.00E+00	5143	-2.86	0.66
0.7	-28.40	0.00E+00	5151	-2.86	0.51
0.8	-35.53	0.00E+00	5159	-2.86	0.37
0.9	-42.96	0.00E+00	5165	-2.86	0.22
1	-49.43	0.00E+00	5175	-2.86	0.02

La figura 4.6 en la columna de la izquierda, muestra gráficamente cómo al aumentar el valor del parámetro de diferenciación d desde 0 hasta 1, se va perdiendo memoria al reducirse la correlación entre la serie de datos original y la serie transformada y va aumentado la estacionariedad medida con el uso del estadístico adf. La línea horizontal significa el punto límite para decisión del criterio adf. La línea verde por encima de la línea entrecortada significa que no se puede rechazar la hipótesis de raíces unitarias en la serie de tiempo. Las gráficas de la derecha muestran un comparativo de las series de precios originales y las series con diferenciación fraccional para las acciones CNP, HPQ y SWN. En el caso de CNP y HPQ se eligió un valor de d= 0.2 y para el caso de SWN se utilizó un valor de d=0.3. En todos los casos se realizaron cálculos con la metodología de ventana fija utilizando un umbral mínimo de 0.003 para los pesos de los rezagos.

La figura 4.7 muestra una alta autocorrelación en las series de precios con transformación fraccional para las cuatro acciones, esto contrasta con la ausencia o baja correlación de los datos de la serie con diferenciación con d = 1 que es el caso que ocurre con el cálculo de los rendimientos financieros.

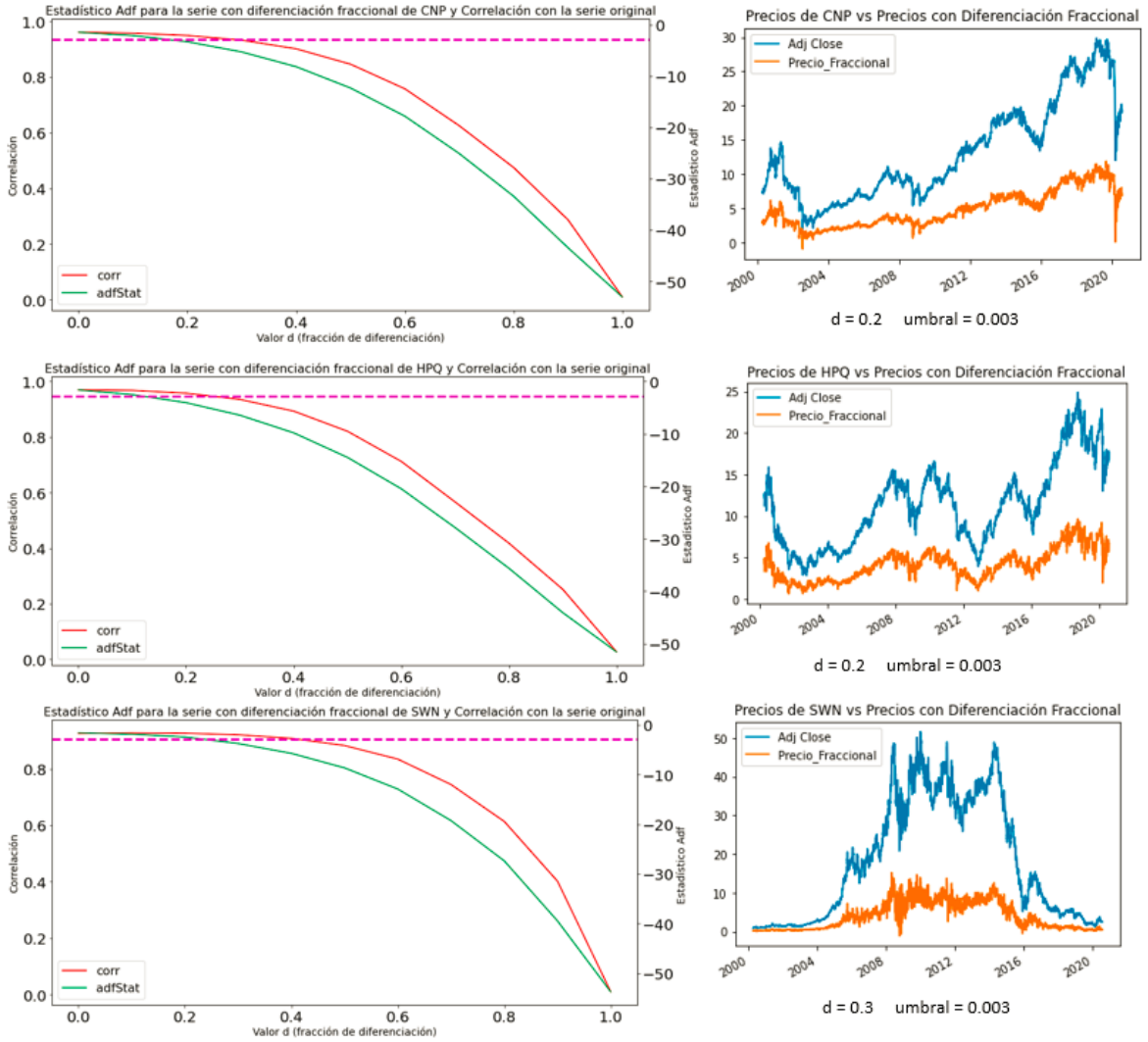

Figura 4.6: Gráfica de visualización de criterio de decisión de d y serie de datos original y transformada

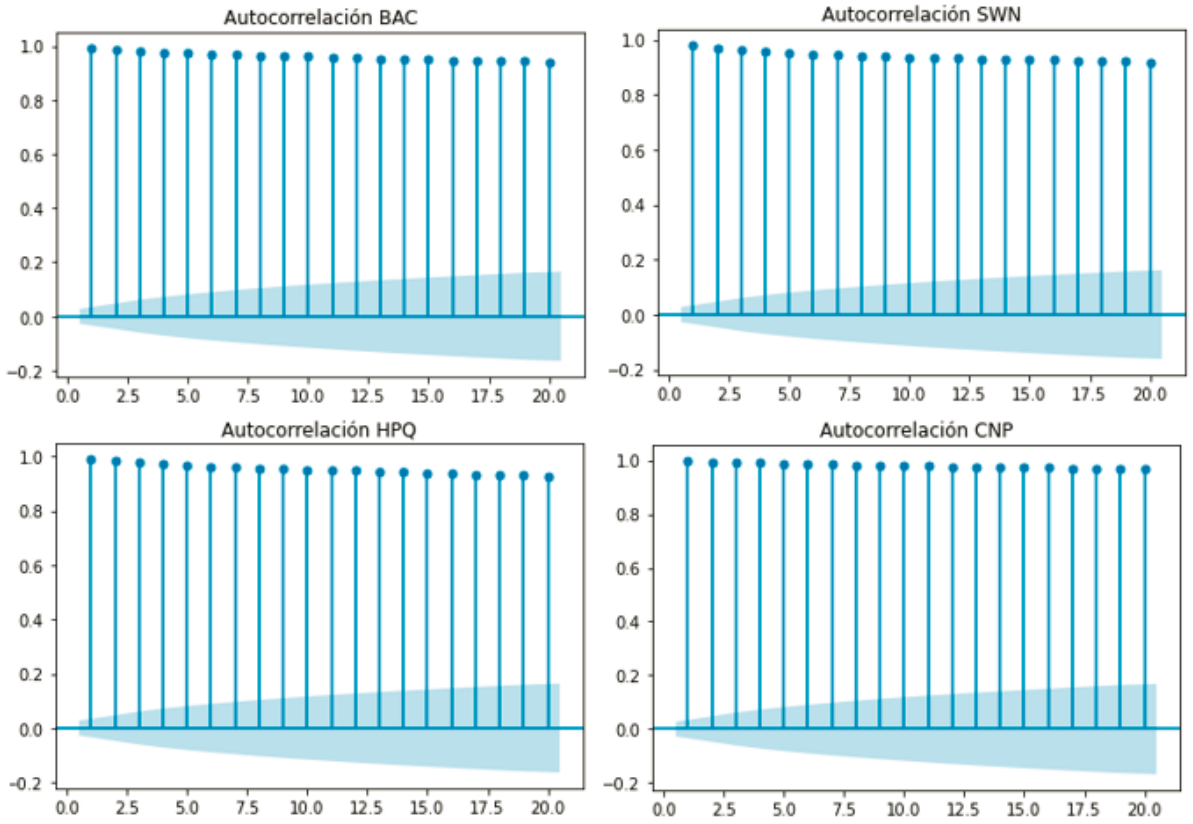

Figura 4.7: Autocorrelación se series de precios con transformación fraccional

4.2 DATOS INTRADIARIOS

Los datos intradiarios se descargaron de la página http://iextraing.com, los cuales se encuentran disponibles de forma libre. Corresponden a datos de ese mercado corresponden al 1.8% de las transacciones del mercado en Estados Unidos. Los datos se obtienen en un formato pcap para cada día de transacciones, el cual fue transformado y convertido a formato csv con scripts de python. Los datos se descargaron desde el 06 de septiembre de 2017 (primeros datos disponibles en este mercado) y el 30 de junio de 2020. Los datos descargados tienen el total de transacciones y cotizaciones, datos que no tienen una agrupación temporal. Se hizo entonces una agrupación cada cinco minutos.

Finalmente los datos transformados son:

- Precios intradiarios
- Rendimientos intradiarios
- Precio medio (promedio del Bid - Ask)
- Volumen intradiario

- Spread intradiadio (Diferencia entre Bid y Offer)

La serie de precios intradiarios y precios medio son casi iguales por lo cual se descarta esta última variable. La figura 4.8 muestra las gráficas para la acción de Bank of America.

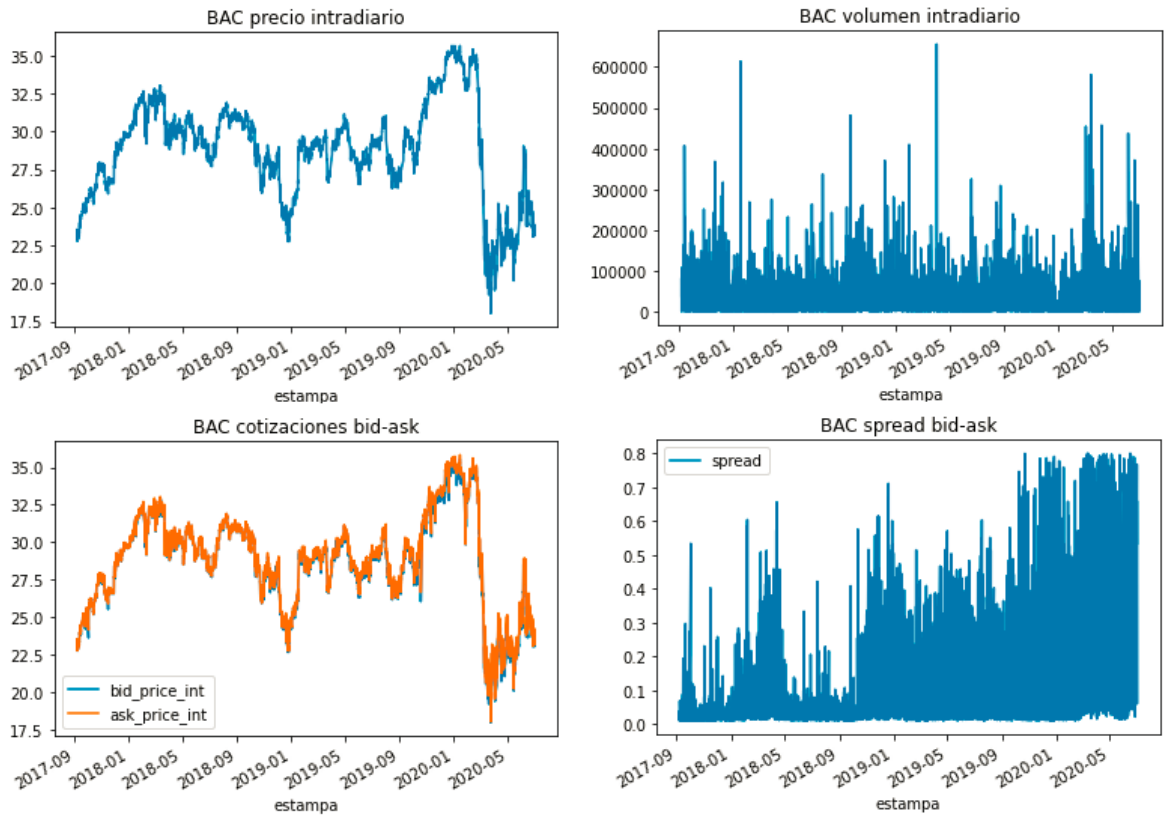

Figura 4.8: Datos intradiarios de la acción de BAC

Para la modelación con las redes neuronales es conveniente el escalamiento. Los precios se transformaron en rendimientos y los volúmenes y spreads se transformaron a una escala entre cero y uno.

5. ENTRENAMIENTO DE MODELOS GAN

5.1 DATOS DIARIOS

5.1.1 Modelo DCGAN No Condicionado Sobre Rendimientos Diarios

El primer modelo GAN a utilizar será no condicionado, esto es, para su entrenamiento se utilizan únicamente los datos de los rendimientos de las acciones seleccionadas.

Para el modelo DCGAN se utiliza un submodelo discriminador con las siguientes características:

- Datos de entrada: Vector con 180 datos de rendimiento diarios. Para cada acción se tiene un total de 167 series de tiempo con 180 datos cada uno.
- Capas intermedias: Se utilizan dos capas convolucionales de una dimensión, cada una con un filtro de tamaño 64, y tamaño de kernel de 3, stride de 2 y padding same. Además de utilizar la función de activación Leaky ReLU con parámetro alpha=0.2. Posteriormente se adiciona después cada capa convolucional un dropout con parámetro 0.4.
- Capas de salida: Se adiciono finalmente una capa flatten para convertir los tensores en vectores de una dimensión y luego una capa fully conected con salida de dimensión uno y función de activación tipo sigmoide.

Como optimizador se usa el optimizador Adam, la función de pérdida es tipo binary crossentropy y como métrica se utiliza el accuracy. Finalmente, este submodelo discriminador contiene un total de 15,489 parámetros entrenables.

Tabla 5.1: Característica del submodelo discriminador del GAN no condicionado

Layer (type)	Output Shape	Param #
conv1d_33 (Conv1D)	(None, 90, 64)	256
leaky_re_lu_71 (LeakyReLU)	(None, 90, 64)	0
dropout_20 (Dropout)	(None, 90, 64)	0
conv1d_34 (Conv1D)	(None, 45, 64)	12352
leaky_re_lu_72 (LeakyReLU)	(None, 45, 64)	0
dropout_21 (Dropout)	(None, 45, 64)	0
flatten_10 (Flatten)	(None, 2880)	0
dense_23 (Dense)	(None, 1)	2881

Total params: 15,489
Trainable params: 15,489
Non-trainable params: 0

También el DCGAN requiere un submodelo generador con las siguientes características:

- Datos de entrada: Las entradas del submodelo generador son vectores de un espacio latente aleatorio desde el cual el modelo generador toma datos y los mapea hacia un espacio de datos con características similares a los datos reales. Se tomó como dimensión de espacio latente igual a 500 y luego se transforma hacia una dimensión proporcional a la dimensión deseada del dato de salida para facilitar la transformación de dimensiones a través del upsampling.
- Capas intermedias: Se utilizaron tres capas tipo convolución de una dimensión transpuesta, con parámetros: Filtros= 128, Tamaño de kerner = 4, Strides = 2, Padding= same, Función de activación = Leaky ReLU.
- Capa de salida: A través de una capa de convolución de una dimensión para ajustar los tensores de salida con la dimensión (128, 1) qué es la dimensión que deben tener los datos simulados e igual a la dimensión que tienen los datos reales.

Este submodelo generador no tiene optimizador, métricas ni función de pérdida, pues a diferencia del submodelo discriminador, este submodelo sólo se entrena en el DCGAN total. Este submodelo generador contiene un total de 260,609 parámetros entrenables.

Tabla 5.1: Característica del submodelo generador del GAN no condicionado

Layer (type)	Output Shape	Param #
dense_24 (Dense)	(None, 5760)	63360
leaky_re_lu_73 (LeakyReLU)	(None, 5760)	0
reshape_13 (Reshape)	(None, 45, 128)	0
conv1d_transpose_38 (Conv1DT	(None, 90, 128)	65664
leaky_re_lu_74 (LeakyReLU)	(None, 90, 128)	0
conv1d_transpose_39 (Conv1DT	(None, 180, 128)	65664
leaky_re_lu_75 (LeakyReLU)	(None, 180, 128)	0
conv1d_transpose_40 (Conv1DT	(None, 360, 128)	65664
leaky_re_lu_76 (LeakyReLU)	(None, 360, 128)	0
conv1d_35 (Conv1D)	(None, 180, 1)	257

Total params: 260,609
Trainable params: 260,609
Non-trainable params: 0

Posteriormente, se integran los submodelos generador y discriminador en el modelo DCGAN unificado con los siguientes parámetros entrenables:

Tabla 5.3: Característica del modelo GAN unificado no condicionado

Layer (type)	Output Shape	Param #
sequential_36 (Sequential)	(None, 180, 1)	260609
sequential_35 (Sequential)	(None, 1)	15489

Total params: 276,098
Trainable params: 260,609
Non-trainable params: 15,489

En el modelo DCGAN unificado, los parámetros del submodelo discriminador aparecen inicialmente como no entrenables, esto se debe a que el entrenamiento total del modelo DCGAN se cumple mediante la alternancia de un aprendizaje del modelo generador-discriminador y del modelo discriminador de forma alternada. (nunca de forma simultánea).

Para el entrenamiento se usaron 500 epochs.

5.1.2 Modelo cCGAN en Series Con Diferenciación Fraccional

Inicialmente, se ejecutó un DCGAN no condicional sobre las series de precios transformadas, sin embargo el modelo no logró capturar en su proceso de generación de datos los diferentes niveles de precios a lo largo del tiempo. La siguiente gráfica muestra cómo el proceso generador de datos presenta una alta variabilidad tratando de suplir las carencias de los datos de entrada.

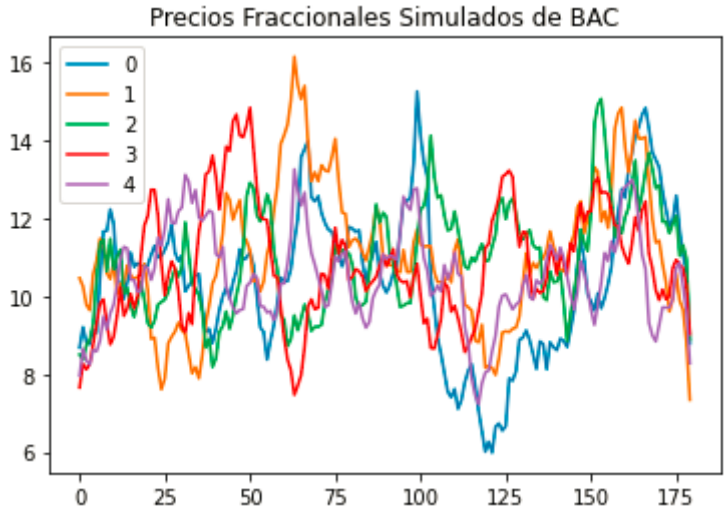

Figura 5.1: Datos sintéticos de precios con transformación fraccional de BAC con GAN no condicional

Se utilizó entonces un modelo cCGAN condicional donde se generó una etiqueta correspondiente al nivel inicial de precios para cada serie de datos, además de realizó un escalamiento de los datos entre cero y uno para cada serie de datos.

Para la modelación del Conditional Convolutional GAN (cCGAN) se utilizó un sub modelo discriminador que integra dos fuentes de datos: 1) Series de datos de precios con transformación fraccional con reescalamiento. 2) Etiqueta correspondiente al nivel inicial de precios de cada serie de tiempo La figura 5.2 muestra la configuración del submodelos discriminador

- Datos de entrada 1: Vector con 180 precios con transformación fraccional con escalamiento. Para cada acción se tiene un total de 165 series de tiempo con 180 datos cada uno.
- Datos de entrada 2: vector con 165 datos enteros entre 0 y el máximo nivel de precios de la acción en los datos históricos. Este número entero se

transforma con capas de embedding y capas densas para convertir el dato a una dimensión compatible con las longitud de las series de tiempo.

- Capas intermedias: Se usa una concatenación de los dos tipos de datos de entrada y posteriormente se utilizan dos capas convolucionales de una dimensión, cada una con un filtro de tamaño 128, y tamaño de kernel de 3, stride de 2 y padding same. Además de utilizar la función de activación Leaky ReLU con parámetro alpha=0.2 después de cada capa convolucional. Posteriormente se adiciona después cada capa convolucional un dropout con parámetro 0.4.
- Capas de salida: Se adiciona finalmente una capa flatten para convertir los tensores en vectores de una dimensión y luego una capa fully connected con salida de dimensión uno y función de activación tipo sigmoide.

Como función de pérdida se agregó una función de cross entropía binaria, un optimizador Adam y métrica accuracy.

También se agregó un sub modelo generador al cual también incorpora dos tipos de entrada, la primera son los datos aleatorios del espacio latente, y la segunda la misma clase de etiqueta que se incorporó como entrada en el modelo discriminador pero generada de forma aleatoria. la figura 5.3 muestra la arquitectura del modelo GAN donde se observa del detalle del modelo generador.

- Datos de entrada 1: datos aleatorios de una dimensión latente igual a 500 la cual se transforma con una capa densa a una dimensión de 45 la cual será la que se usará para ser concatenada con los datos de entrada 2.
- Datos de entrada 2: datos aleatorios enteros entre cero y el rango máximo de los datos históricos. Mediante capas de embedding y transformación de dimensiones para finalmente llegar a una dimensión de 45.
- Capas intermedias: inicia con una concatenación de los datos de entrada 1 y 2, y luego se agregan dos capas convolucionales 1d transpuestas con función de activación LeakyReLU.
- Capa de salida: con una capa de convolución se genera una salida de dimensión (180,1) que es igual a la dimensión de las series de tiempo reales.

Finalmente se realiza una combinación de los modelos generador y discriminador donde es importante sincronizar la entrada las entradas de datos de etiquetas en los modelos generador y discriminador, como se puede observar en la figura 5.3

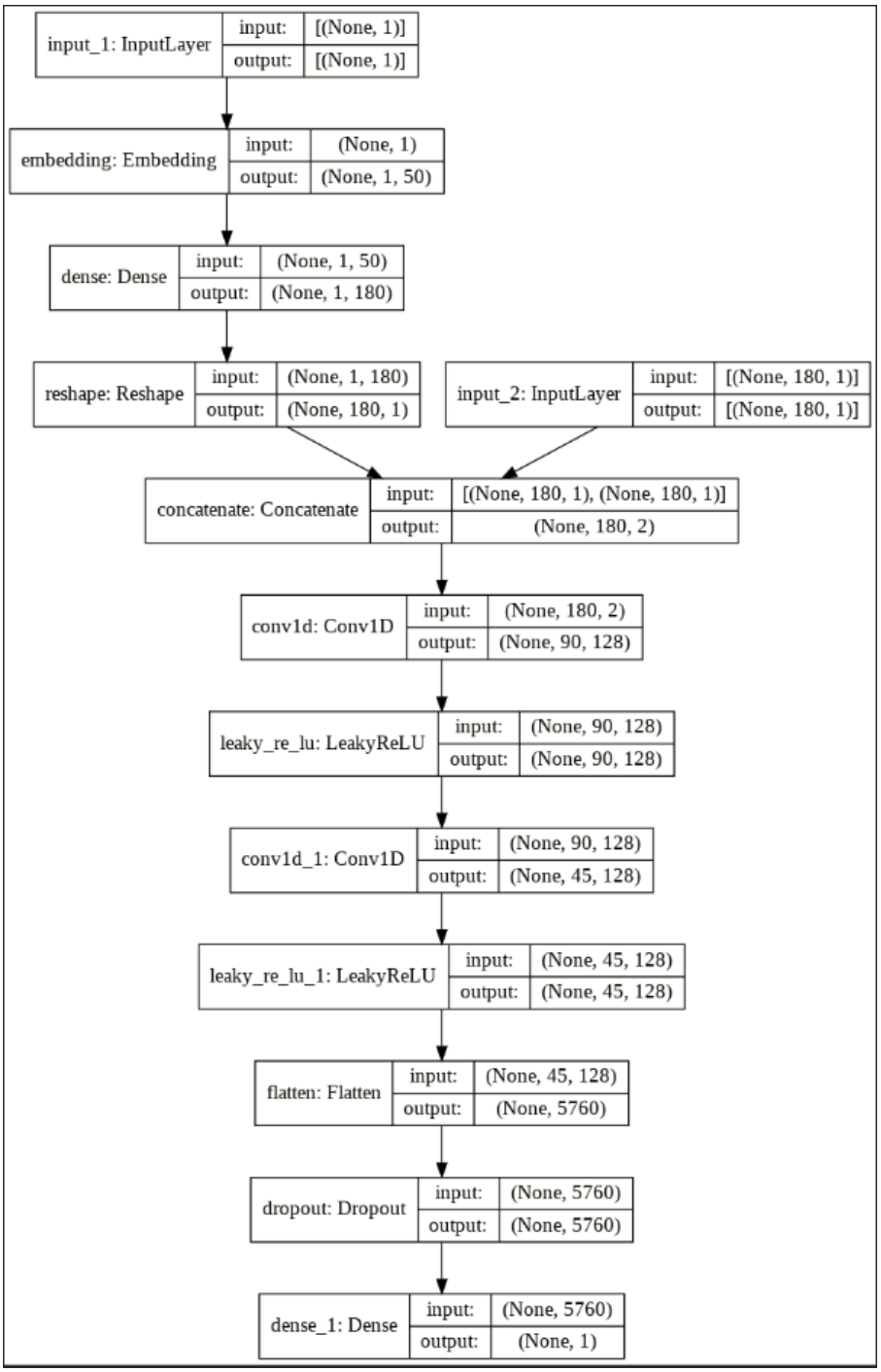

Figura 5.2: Submodelo Discriminador del cCGAN

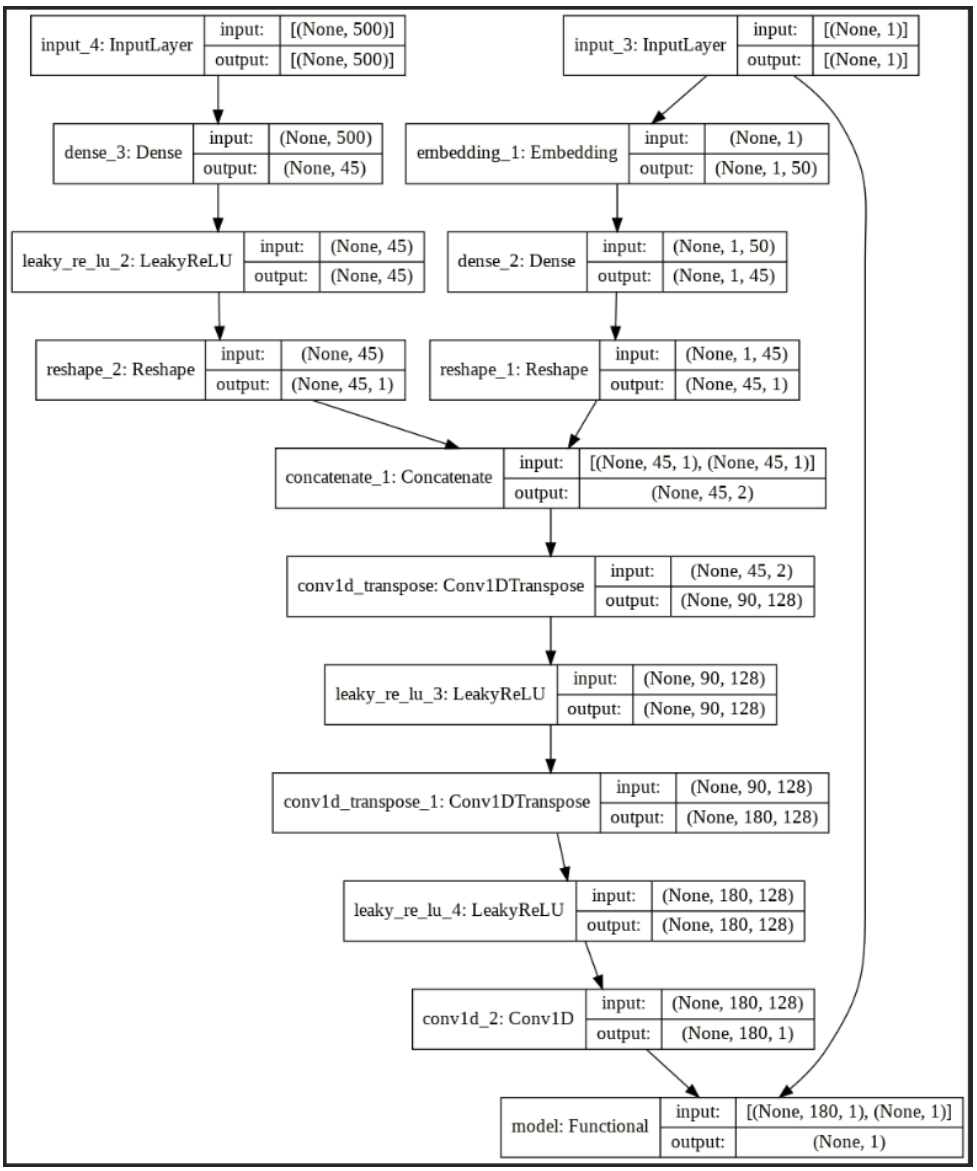

Figura 5.3: Submodelo Generador del cCGAN

5.2 DATOS INTRADIARIOS

5.2.1 Modelo Wasserstein GAN para Rendimientos Intradiarios

El modelo WGAN tiene tres componentes:

- Modelo critico
- Modelo generador
- Modelo GAN

El modelo crítico es similar al modelo discriminador del GAN tradicional tiene una función de pérdida wasserstein. otras diferencias respecto al modelo CGAN son:

- Las etiquetas para los datos reales y simulados con 1 y -1 en lugar de 1 y 0.
- Uso de función de pérdida wassertein tanto para el modelo crítico como para el modelo generador.
- Uso de RSMProp para el gradiente descendente en lugar de Adam.
- Uso de función de activación lineal para la capa final
- Requiere el uso de una función clipping que se incorpora como restricción en las capas convolucionales.

La tabla 5.4 muestra la configuración del modelo crítico, al cual contiene dos capas convolucionales seguidas de capas de normalización de batch y capas de regularización Leaky ReLU. Finalmente se incorpora una capa tipo flatten para convertir los tensores a una sola dimensión y posteriormente con una capa fully connected para obtener una dimensión de salida de 1. Este modelo crítico tiene un total de 15.745 parámetros entrenables.

El modelo generador es similar al modelo generador del CGAN. La tabla 5.5 muestra la configuración del modelo generador. Contiene dos capas de convolución transpuesta para realizar upsampling de los datos aleatorios de la dimensión latente. Se utiliza una capa final de convolución para ajustar los tensores de salida a una dimensión (180, 1) que el el tamaño de la ventana de precio de las series de tiempo intradiarias.

Tabla 5.4: Características de modelo crítico del WGAN

Layer (type)	Output Shape	Param #
conv1d_132 (Conv1D)	(None, 90, 64)	256
batch_normalization_185 (Bat	(None, 90, 64)	256
leaky_re_lu_235 (LeakyReLU)	(None, 90, 64)	0
conv1d_133 (Conv1D)	(None, 45, 64)	12352
batch_normalization_186 (Bat	(None, 45, 64)	256
leaky_re_lu_236 (LeakyReLU)	(None, 45, 64)	0
flatten_42 (Flatten)	(None, 2880)	0
dense_92 (Dense)	(None, 1)	2881

Total params: 16,001
Trainable params: 15,745
Non-trainable params: 256

Tabla 5.5: Característica del modelo generador del WGAN

Layer (type)	Output Shape	Param #
dense_93 (Dense)	(None, 5760)	63360
leaky_re_lu_237 (LeakyReLU)	(None, 5760)	0
reshape_50 (Reshape)	(None, 45, 128)	0
conv1d_transpose_100 (Conv1D	(None, 90, 128)	65664
batch_normalization_187 (Bat	(None, 90, 128)	512
leaky_re_lu_238 (LeakyReLU)	(None, 90, 128)	0
conv1d_transpose_101 (Conv1D	(None, 180, 128)	65664
batch_normalization_188 (Bat	(None, 180, 128)	512
leaky_re_lu_239 (LeakyReLU)	(None, 180, 128)	0
conv1d_134 (Conv1D)	(None, 180, 1)	257

Total params: 195,969
Trainable params: 195,457
Non-trainable params: 512

5.2.2 Modelos DCGAN para Rendimientos, Volumen y Spread Intradiario

Para el modelo DCGAN usado para la generación de rendimientos, volúmenes y spreads intradiarios se utilizó la misma arquitectura usada para la modelación de rendimientos diarios, con una modificación de la función de activación del modelo generador. La tabla 5.6 muestra las diferentes funciones de activación usadas para la generación de cada tipo de serie de tiempo intradiario. Debe existir una correspondencia entre el rango de las series de tiempo a simular y el rango de las funciones de activación a utilizar para la última capa del modelo generador de datos.:

Tabla 5.6: Características función de activación modelo generador intradiario CGAN

Serie de Datos	Función de Activación	Comentario
Retornos Intradiarios	tanh	Los rendimientos son positivos y negativos
Volúmenes Intradiarios	sigmoig	volúmenes escalados entre 0 y 1
Spreads Intradiarios	sigmoid	volúmenes escalados entre 0 y 1

6. GENERACIÓN DE SERIES FINANCIERAS SINTÉTICAS Y EVALUACIÓN DE SUS CARACTERÍSTICAS

6.1 DATOS DIARIOS

Después de finalizar el entrenamiento del modelo GAN, se realiza la exportación del submodelo generador el cual se puede usar de forma independiente para la generación de series sintéticas. Con este modelo generador se realizan entonces simulaciones de series de tiempo con los cuales se consiguieron los siguientes resultados.

6.1.1 Modelo GAN No Condicionado Sobre Rendimientos Diarios

Con el generador de series de tiempo obtenido al entrenar los modelos GAN para las acciones con Ticker BAC, CNP, HPQ, SWN, se guardan los parámetros del sub modelo generador y se exporta. Este modelo generador se utiliza entonces como generador sintético de datos para cada una de las acciones.

Se recuerda que para acción se tenían únicamente 5,178 datos históricos, sin embargo aquí se generaron a manera de ejemplo 1000 series de datos, cada una con 180 datos de rendimientos datos, para un total de 180.000 datos sintéticos, esto es, un total de 34 veces más de datos sintéticos que datos reales históricos. Esta es precisamente la ventaja que tendría el uso del modelo generador tomado del GAN pues se usaría para el entrenamiento y ajuste de modelos sin utilizar varias veces los datos históricos.

La figura 6.1 muestra una comparación de 5 series de 180 datos simulados de rendimientos de la acción de HPQ y los datos reales. En principio parece que el modelo GAN puede generar datos sintéticos de series de tiempo de forma satisfactoria. La tabla 6.1 muestra el resumen del cálculo de los momentos de los rendimientos para las cuatro acciones seleccionadas. En general los valores de la media son cercanos a cero, tanto en las series simuladas como en las históricas. La desviación estándar es cercana al 2% al día en los datos sintéticos y reales. Los modelos GAN generan series de tiempo con asimetría y curtosis similares a los datos históricos aunque en el caso de los rendimientos diarios de la cción de HPQ no capturó totalmente el alto grado de curtosis. La figura 6.2 muestra los autocorrelogramas para las series de datos simulados los cuales muestran que estos datos presentan una dependencia en su proceso generador de datos respecto

a datos pasados, situación que no es tan fuerte en los datos reales, esto es un problema en cuanto a la modelación de la serie sintética pues algoritmos que utilicen estos datos como entrada para la modelación financiera podrían encontrar patrones inexistente en la realidad. En este sentido, a las series generadas les falta aleatoriedad en cuanto a su dependencia en el tiempo.

Figura 6.1: Gráfica de 5 series de 180 datos simulados de rendimientos diarios y rendimientos diarios acumulados y su comparación son datos históricos de la acción de HPQ. Datos reales a la derecha.

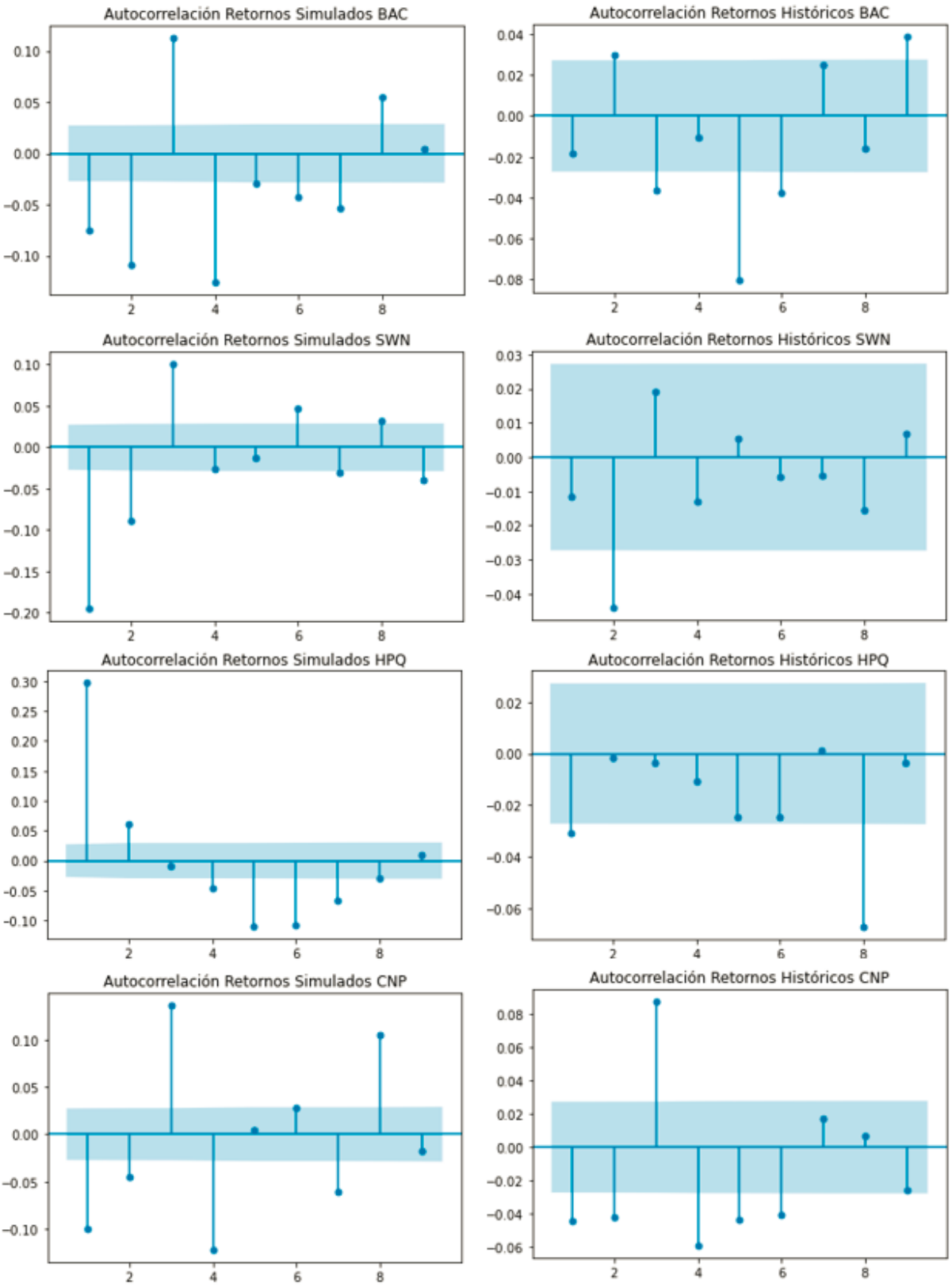

Figura 6.2: Correlograma de los rendimientos diarios históricos y rendimientos simulados.

Tabla 6.1: Resumen de los cuatro primeros momentos de los rendimientos históricos y simulados de cuatro acciones

TICKER		Media	Desviación Estándar	Asimetría	Curtosis	Máximo	Mínimo
BAC	Real	0.0006	0.0293	0.8478	26.06	0.3527	-0.2897
	Simulado	0.0007	0.0206	-0.0833	9.38	0.3676	-0.3888
SWN	Real	0.0008	0.0336	0.3926	8.84	0.3667	-0.2949
	Simulado	0.0043	0.0229	0.9068	12.54	0.4922	-0.2286
BAC	Real	0.0004	0.0243	-0.0697	8.33	0.1729	-0.2003
	Simulado	0.0151	0.0215	0.0215	10.74	0.3197	-0.3602
HPQ	Real	0.0005	0.0232	2.1218	143.20	0.6296	-0.4221
	Simulado	0.0025	0.0167	0.9059	13.19	0.4026	-0.2924

6.1.2 Modelo GAN Condicional en Series Con Diferenciación Fraccional

Se muestran los resultados del modelo cCGAN para los precios con transformación fraccional de las acciones de BAC. El primer hecho a resaltar es que los datos sintéticos presentan un patrón de autorrelaciones similar al de los datos históricos aunque con menor persistencia. Figura 6.3.

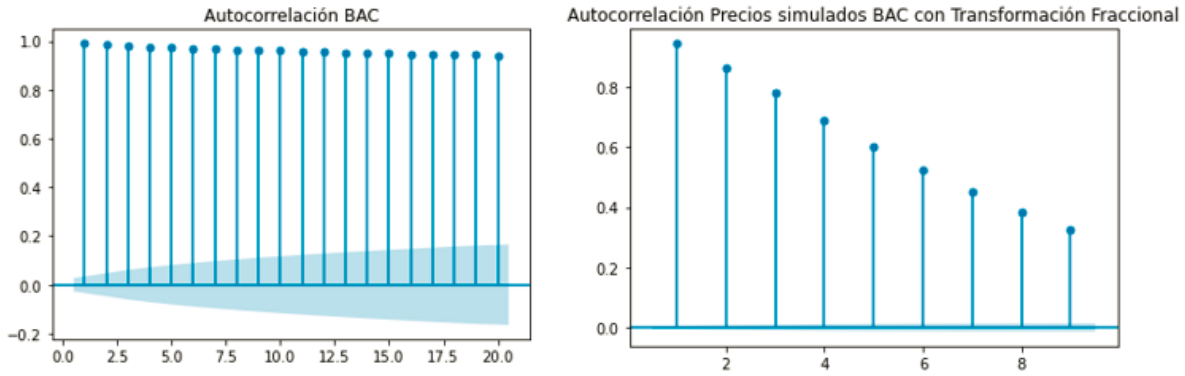

Figura 6.3: Autorrelogramas datos históricos y simulados de BAC

La figura 6.4 muestra cinco series de precios con transformación fraccional para la acción de BAC, para los niveles de precios de 0, 8 y 15. La tabla 6.2 muestra varias estadísticas del comportamiento de los precios de la acción de BAC con transformación fraccional para cada nivel de precios de la acción. Pese a que el número de datos históricos es bajo en cada categoría, aún así se puede ver que los datos simulados en cada una de ellas tienen un comportamiento similar, pero no igual, al de los datos históricos. No se realizó un histograma de los precios puesto que la generación de datos por categorías, y la generación de categorías de forma

aleatoria mediante una distribución uniforme, impone artificialmente que el número de datos por categoría sea aproximadamente igual.

El uso del modelo condicional de GAN permitió mejorar la generación de datos permitiendo la disminución de las variaciones dentro de cada nivel además de entregarle al modelador una variable de control sobre las series a generar. El modelo GAN sin condicional debe generar un solo proceso generador de datos por lo cual puede incurrir en sobre estimación de varianzas para intentar cubrir posibles diferencias en los procesos generadores subyacentes.

Tabla 6.2: Resumen descriptivo de las series simuladas con cCGAN e históricas de los precios con transformación fraccional de BAC

RESUMEN POR NIVEL DE DATOS SIMULADOS CON cCGAN DE PRECIOS CON DIFERENCIAION FRACCIONAL							
nivel	media	desviacion estándar	sesgo	curtosis	mínimo	máximo	número de series
0	0.54	0.22	1.16	2.72	0.1	2.4	4860
1	1.52	0.21	1.05	1.60	1.1	2.9	4500
2	2.53	0.23	1.10	1.48	2.1	3.9	4860
3	3.55	0.23	1.02	1.41	3.1	4.7	4680
4	4.54	0.23	1.24	2.24	4.1	6.0	7380
5	5.54	0.23	1.60	7.04	5.1	8.3	6660
6	6.56	0.23	1.17	2.39	6.1	8.2	5580
7	7.52	0.22	1.20	2.07	7.1	8.9	5580
8	8.54	0.23	1.11	1.40	8.1	9.8	6840
9	9.53	0.22	1.02	1.03	9.1	10.6	5760
10	10.53	0.22	1.15	2.32	10.1	12.0	5400
11	11.53	0.22	1.10	1.62	11.1	13.0	5400
12	12.54	0.23	1.21	2.15	12.1	14.2	6120
13	13.54	0.22	0.96	1.52	13.1	14.9	5760
14	14.54	0.22	1.08	1.48	14.1	15.8	5760
15	15.55	0.23	1.12	1.79	15.1	17.1	4860

RESUMEN POR NIVEL DE DATOS HISTORICOS DE PRECIOS CON DIFERENCIACIÓN FRACCIONAL							
nivel	media	desviacion estándar	sesgo	curtosis	mínimo	máximo	número de series
0	1.36	1.51	NaN	NaN	0.3	2.4	2
1	1.73	0.14	-1.46	2.07	1.5	1.8	4
2	2.71	0.51	1.65	2.71	2.3	3.7	7
3	3.55	0.38	0.21	0.25	2.9	4.3	12
4	4.39	0.31	-0.59	1.02	3.6	5.0	24
5	5.45	0.36	-0.12	-0.89	4.8	6.1	27
6	6.40	0.46	0.36	-0.99	5.8	7.1	10
7	7.85	0.28	0.08	-2.17	7.5	8.2	6
8	8.51	0.57	0.00	0.35	7.4	9.7	18
9	9.80	0.44	-1.08	0.57	9.0	10.2	7
10	10.62	0.44	-0.62	1.20	9.6	11.4	15
11	11.70	0.25	-0.95	0.54	11.2	12.0	10
12	12.29	0.90	-1.14	1.21	10.4	13.4	10
13	13.67	0.47	0.11	-5.22	13.2	14.2	4
14	14.64	0.30	0.00	-1.28	14.3	15.0	4
15	15.34	0.23	1.15	1.55	15.1	15.7	5

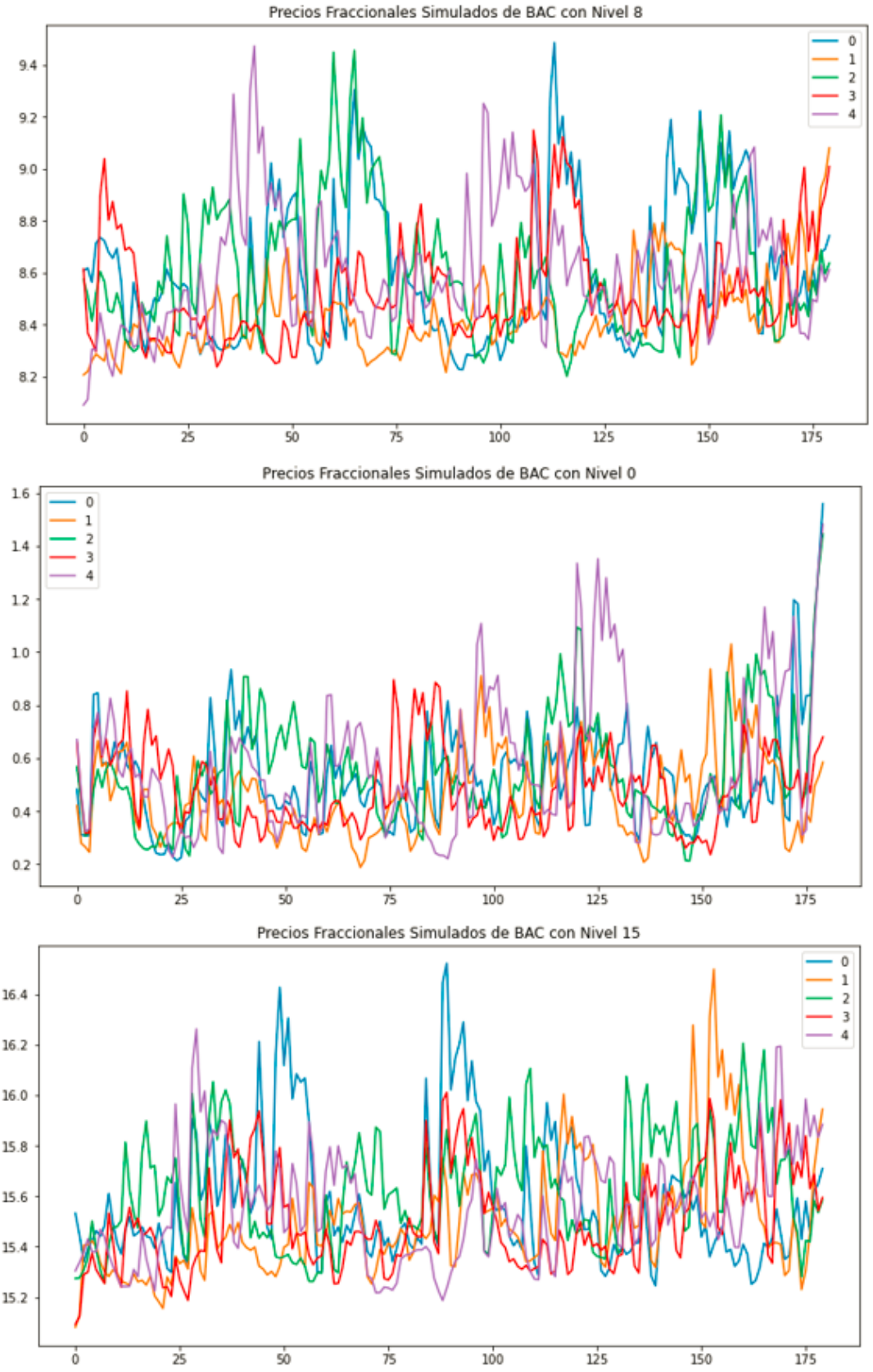

Figura 6.4: Generación de Series de Precios Fraccionales para diferentes niveles

6.2 DATOS INTRADIARIOS

Para la modelación de los datos históricos intradiarios se utilizaron ventanas de 180 datos de 5 minutos.

6.2.1 Modelo Wasserstein GAN para Rendimientos Intradiarios

El modelo WGAN no logró capturar la distribución de los rendimientos intradiarios. Este modelo es particularmente sensible al parámetro de la función de clipping. La figura 6.4 muestra 4 series históricas intradiarias de la acción de Bank of America. Se puede observar que los rendimientos cada 5 minutos se ubican en su mayoría en el rango entre 0.01 y -0.01 con algunos datos extremos que superan 0.002

Figura 6.5: Muestra de series históricas de BAC

En la figura 6.6 se puede observar una comparación de la sensibilidad del comportamiento de las funciones de pérdida de los modelos wasserman para el sub modelo generador y el modelo crítico para las series reales y simuladas. Para un valor de 0.01 en la función de clipping se observa un comportamiento explosivo de las pérdidas, para un valor de 0.001 se observa un desvanecimiento de los gradientes y el modelo GAN deja de aprender. Un valor de 0.001 tiene un mejor comportamiento pero es susceptible de ser mejorado. También en la figura 6.6 se puede ver que los rendimientos simulados no reflejan el comportamiento de los rendimientos intradiarios históricos por al menos dos motivos: 1) El rango típico de los rendimientos es muy alto en comparación con los datos reales. 2) No se observa diversidad en los rendimientos simulados.

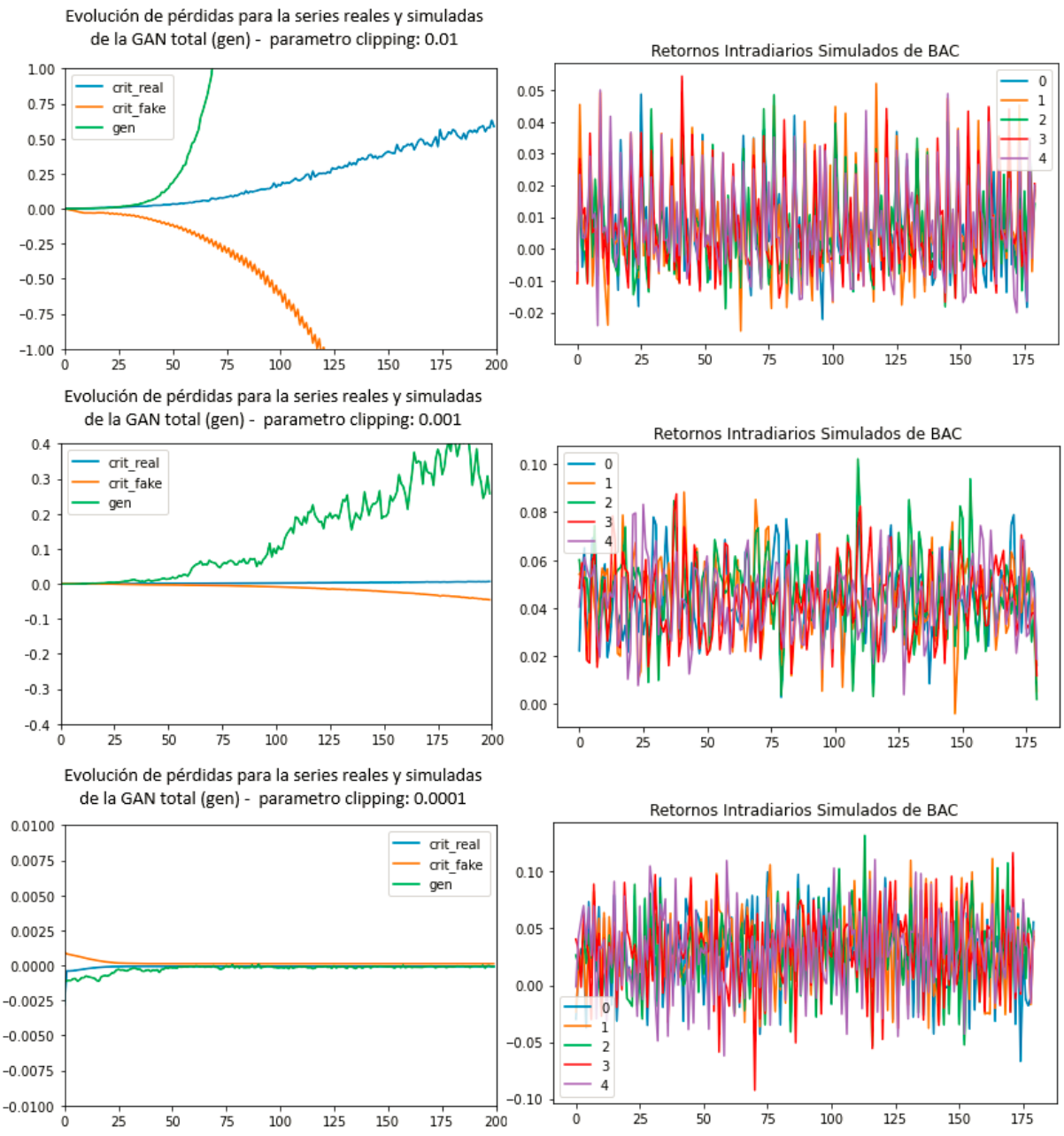

Figura 6.6: Comparación del comportamiento de la pérdidas para diferentes parámetros de clipping

Se realiza entonces la modelación con modelos GAN convolucionales.

6.2.2 Modelo CGAN para Rendimientos Intradiarios

El modelo CGAN sí logró capturar las propiedades distribucionales de los rendimientos intradiarios. La figura 6.7 muestra el comportamiento de las pérdidas de los modelos generador y discriminador durante el entrenamiento del modelo GAN de los rendimientos intradiarios de BAC. Se observa una oscilación entre los

modelos generador y discriminador, dado que tienen objetivos opuestos y adversarios.

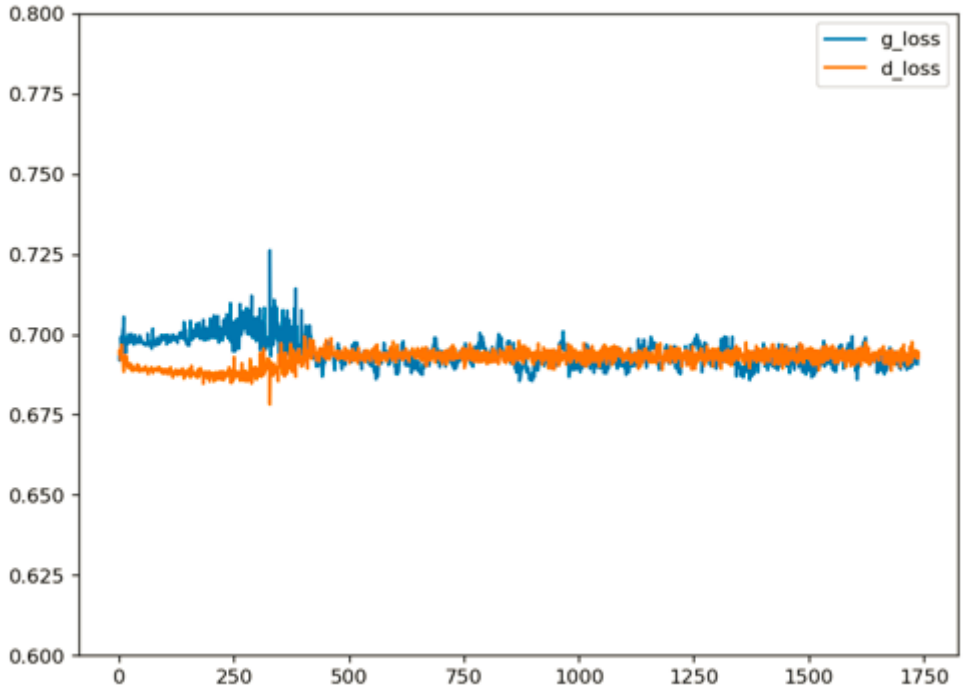

Figura 6.7: Evolución pérdida del generador y del discriminador de CGAN

La figura 6.8 muestra visualmente como las series de datos generadas por el CGAN para los rendimientos intradiarios de BAC, tanto en la escala como en los datos extremos. También los rendimientos acumulados parecen visualmente consistentes.

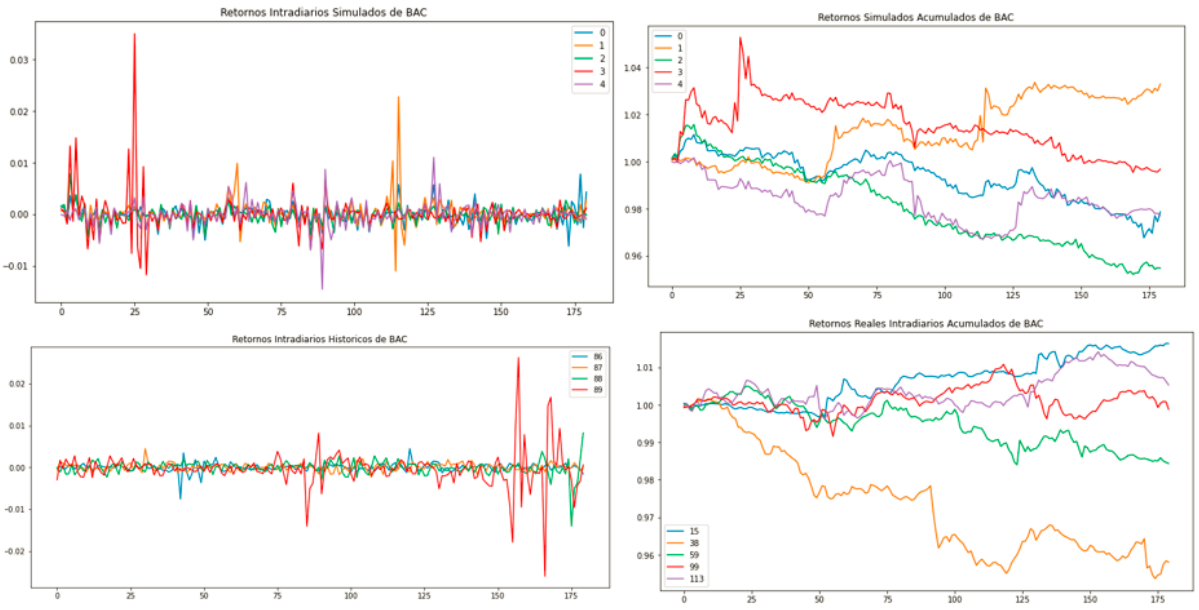

Figura 6.8: Evolución pérdida del generador y del discriminador de CGAN

La figura 6.9 y 6.10 contiene los histogramas de los rendimientos y las autocorrelogramas para las acciones de BAC y SWN, y se comparan las series históricas con las series simuladas. En el caso de de BAC se puede apreciar que la forma de la distribución es simular y que se presenta un patrón de autocorrelaciones similares, con varios rezagos significativos y con una variación de signo entre el primer rezago y los siguientes rezagos.

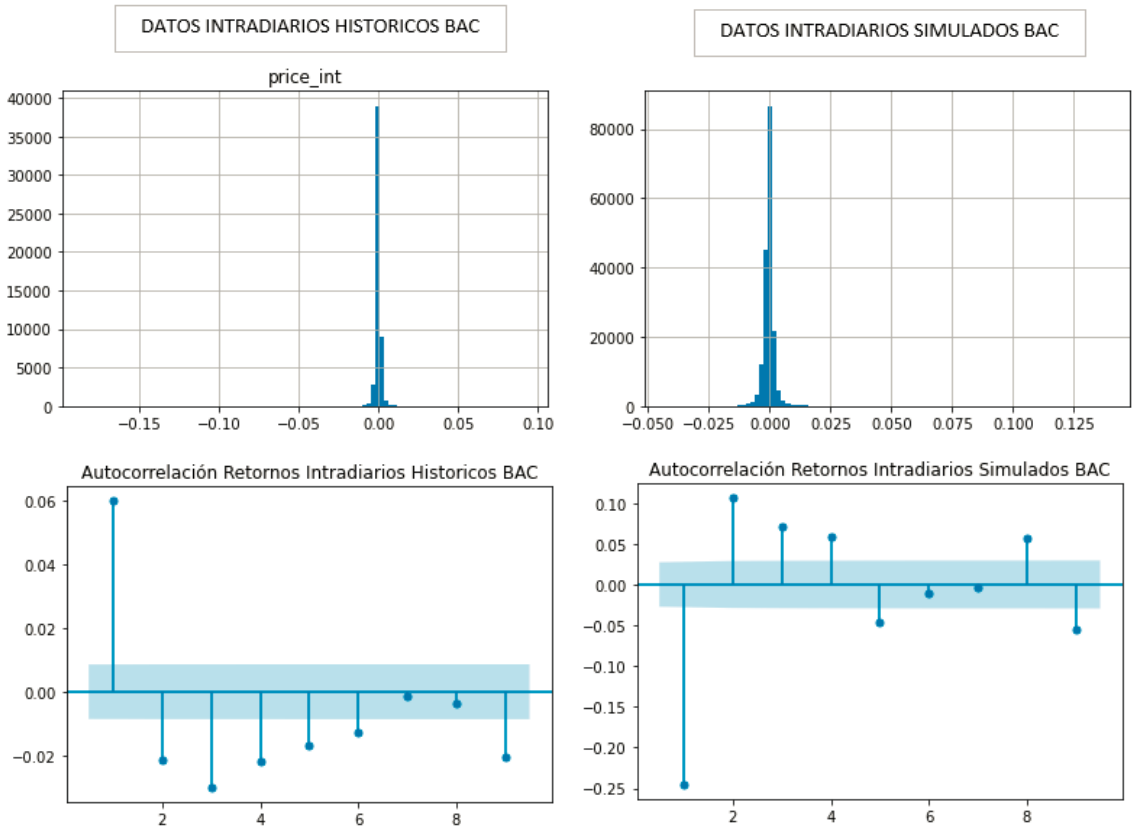

Figura 6.9: Histograma y correlograma rendimientos intradiarios históricos y simulados de BAC

En cuanto el caso de SWN, además de la similaridad de los histogramas, se puede observar que los autocorrelogramas tienen un comportamiento casi idéntico incluyendo los dos primeros rezagos significativos con un cambio de signo.

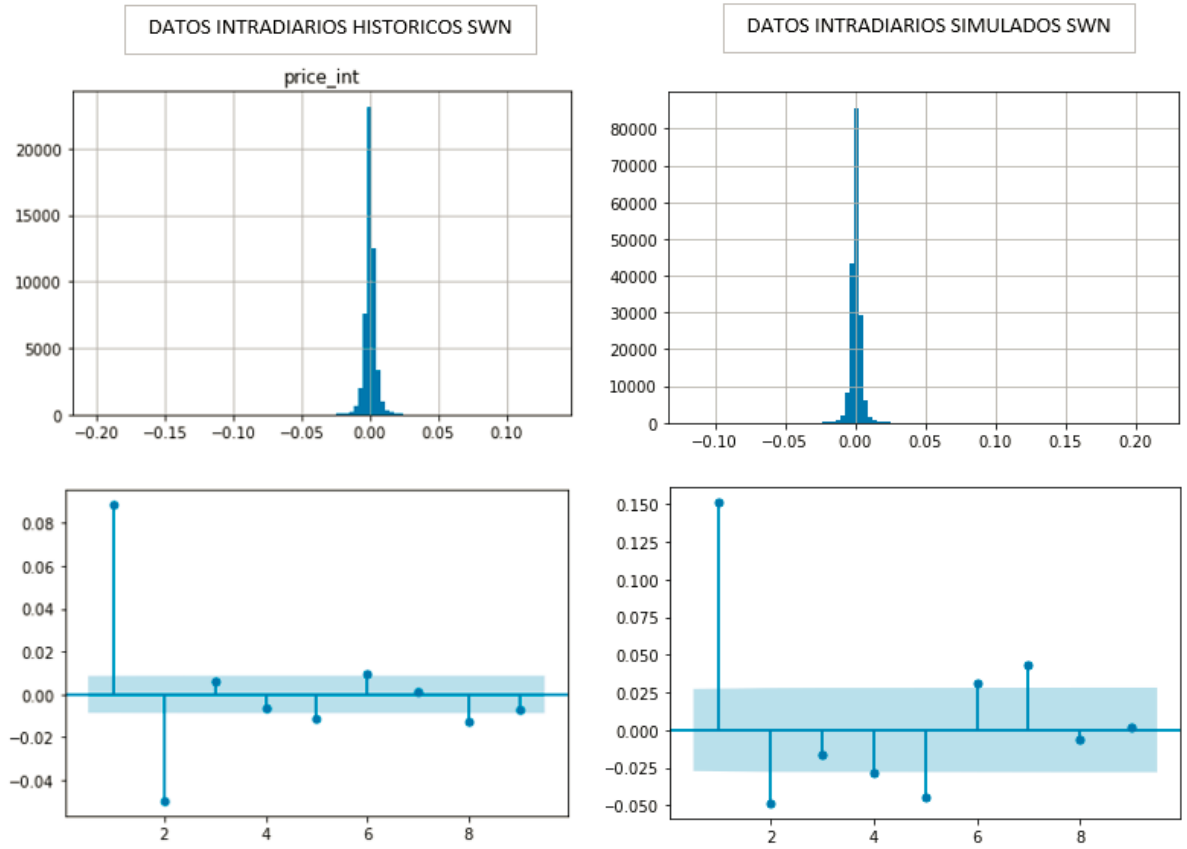

Figura 6.10: Histograma y correlograma rendimientos intradiarios históricos y simulados de BAC

La tabla 6.3 contiene estadísticas descriptivas para las series de datos históricas y simuladas de los rendimientos intradiarios para las acciones de BAC y SWN. En general el modelo GAN captura bien los primeros dos momentos de las distribuciones, en cuanto a la simetría hay cercanía en los parámetros pero no pudo capturar el alto exceso de curtosis de los rendimientos del BAC. Se debe recordar que el periodo de estudio contiene la caída y subida de precios del primer semestre de 2020, lo cual originó un exceso de curtosis nunca antes vista en las series de datos de precios de acciones.

Tabla 6.3: Descripción de las series históricas intradiarias de rendimientos de BAC y SWN

	REDIMIENTOS INTRADIARIOS			
	BAC		SWN	
	Histórico	Simulado	Histórico	Simulado
Media	0.000	0.000	0.000	0.000
Desviación Estándar	0.003	0.003	0.005	0.005
Asimetría	-8.93	5.56	-0.33	4.95
Curtosis	794.39	168.93	130.56	142.03
Máximo	0.092	0.139	0.130	0.214
Mínimo	-0.184	-0.042	-0.201	-0.117

6.2.3 Modelo CGAN para Volúmenes Intradiarios

Con el modelo GAN convolucional también se pueden simular los datos intradiarios de los volúmenes de negociación.

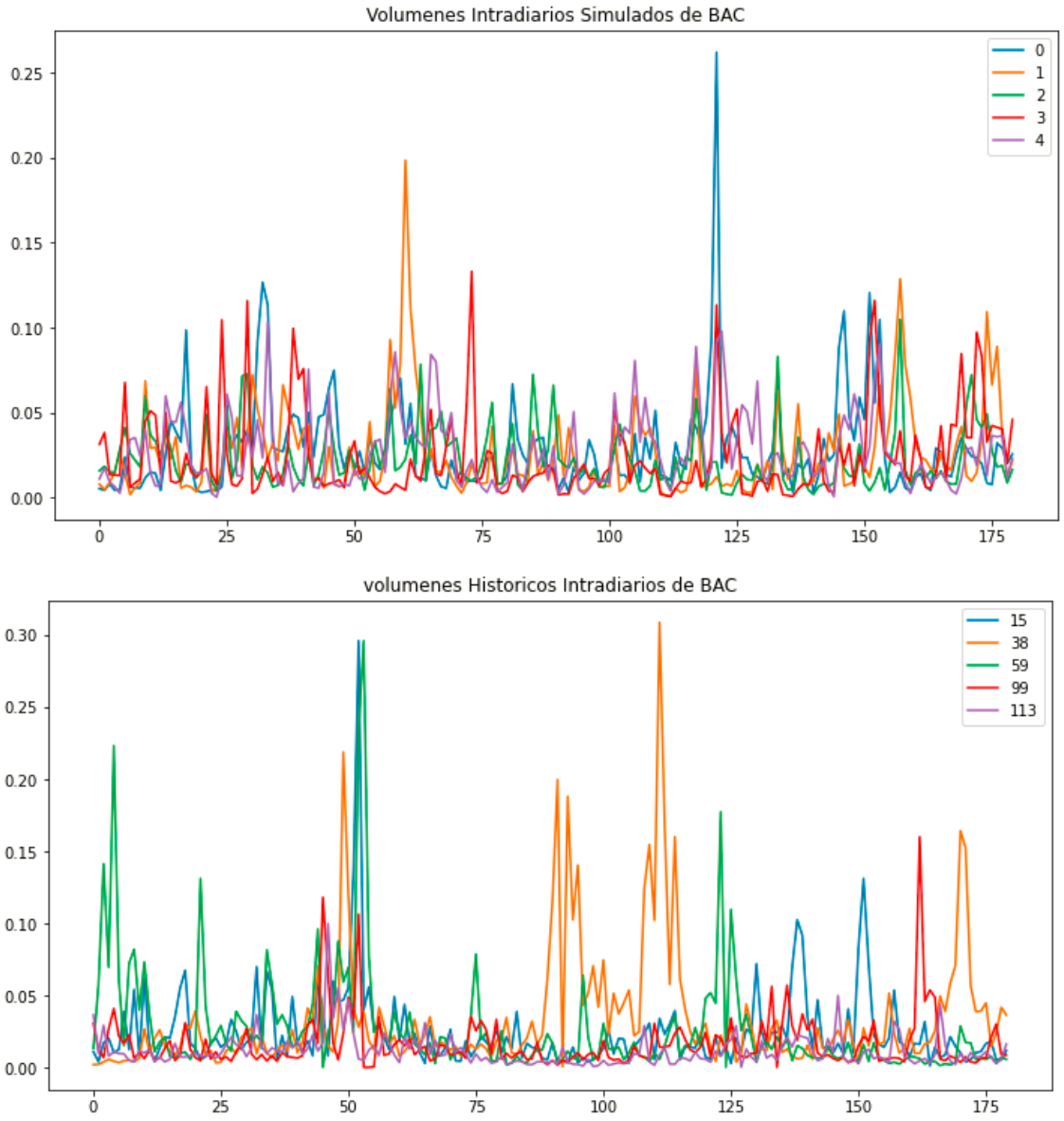

Figura 6.11: Comparativo datos históricos y datos simulados de volúmenes intradiarios de BAC

La figura 6.11 muestran un comparativos de algunas series históricas de volúmenes intradiarios de las acción de BAC., la figura 6.12 contiene el histograma de dichas series y los correlogramas. La tabla 6.4 muestra algunas estadística descriptivas de los volúmenes intradiarios de BAC y SWN. El modelo CGAN logra modelar adecuadamente series de tiempo intradiaria de volúmenes.

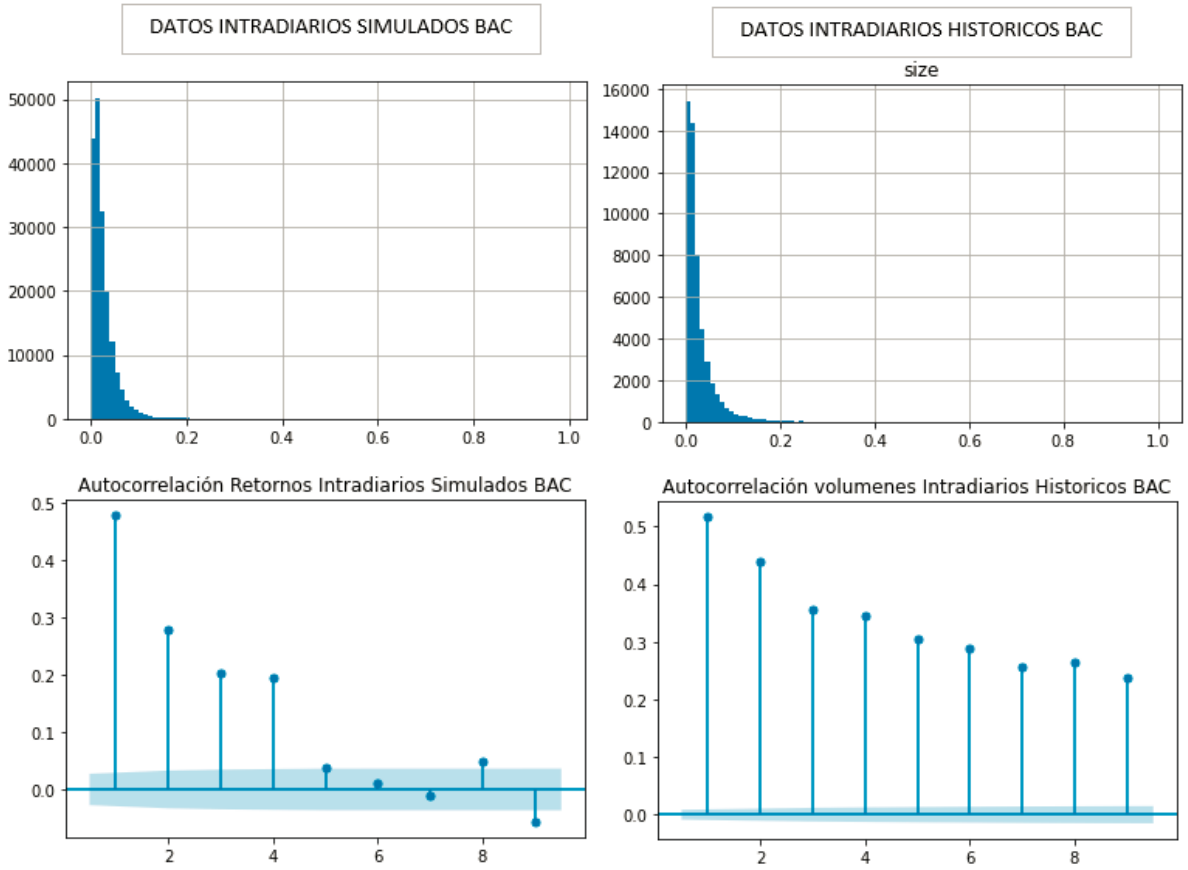

Figura 6.12: Histograma y correlograma volúmenes intradiarios históricos y simulados de BAC

Tabla 6.4: Descripción de las series históricas intradiarias de volúmenes de BAC y SWN

	VOLUMENES INTRADIARIOS			
	BAC		SWN	
	Histórico	Simulado	Histórico	Simulado
Media	0.028	0.026	0.010	0.009
Desviación Estándar	0.038	0.029	0.019	0.019
Asimetría	6.16	7.17	13.08	14.62
Curtosis	77.26	120.55	397.80	401.83
Máximo	1.000	0.987	1.000	0.880
Mínimo	0.000	0.000	0.000	0.000

6.2.4 Modelo CGAN para Spreads Intradiarios

También el modelo CGAN permite modelar los spreads. Las figuras 6.13 y 6.14 muestran un comparativo de las series históricas y simuladas intradiarias de los spreads de la acción de SWN.

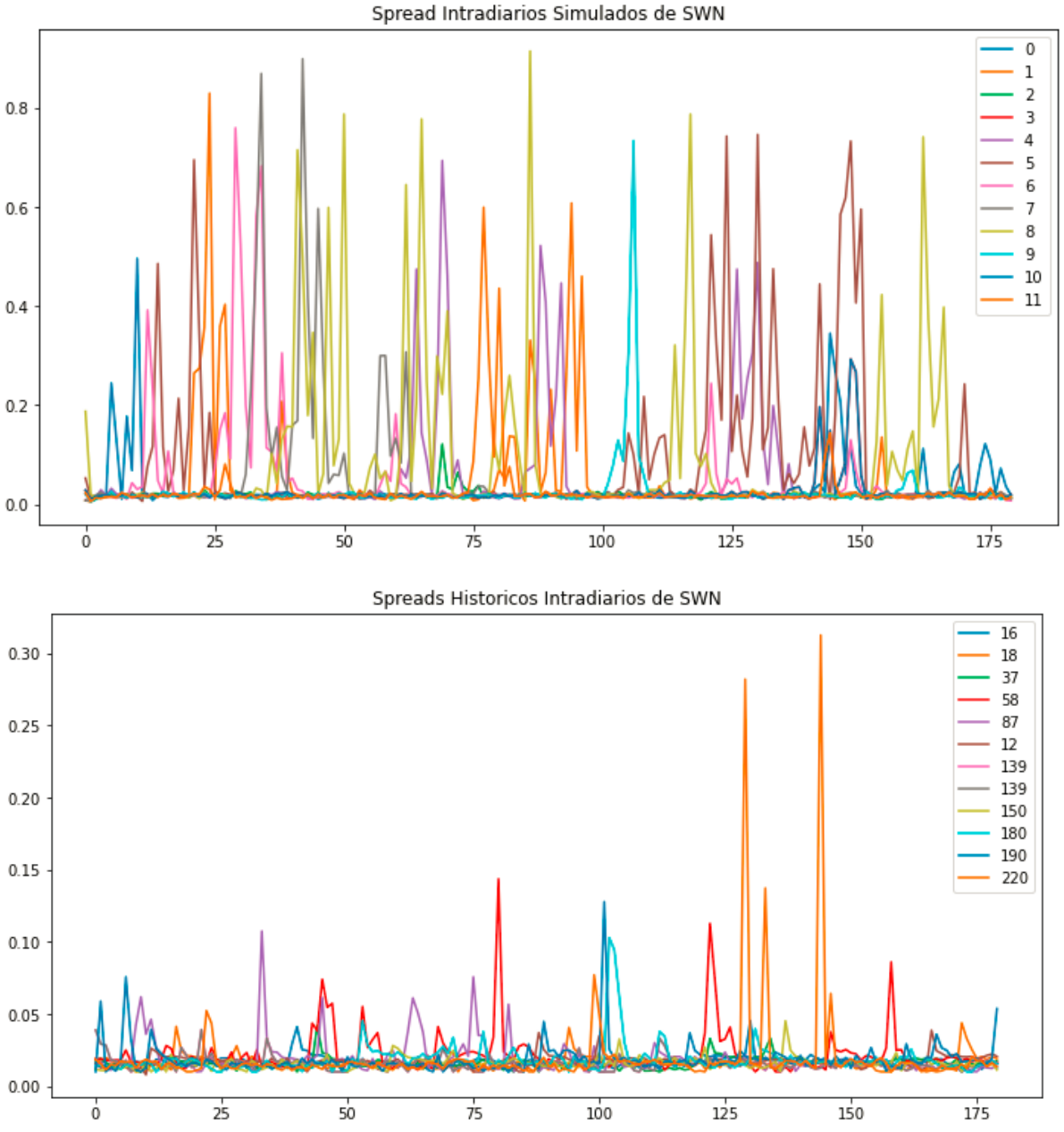

Figura 6.13: Comparativo datos históricos y datos simulados de spreads intradiarios de SWN

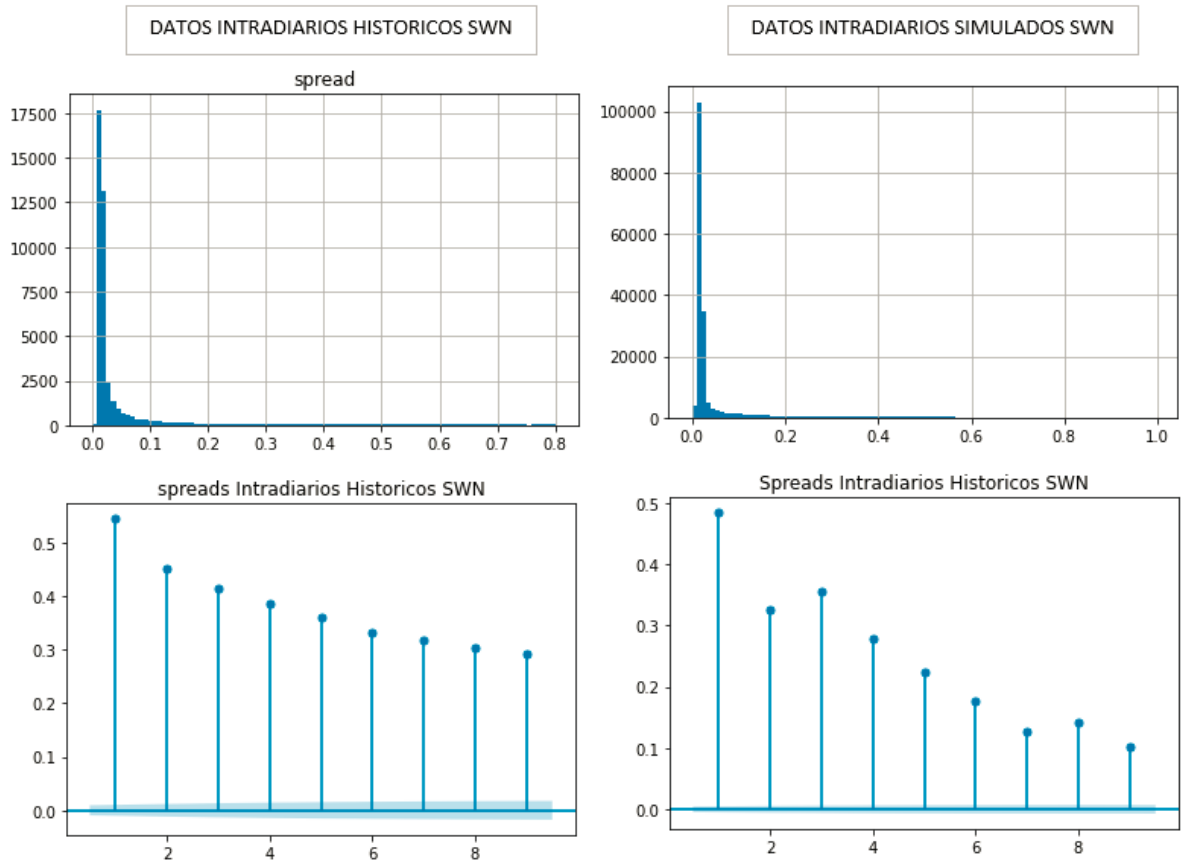

Figura 6.14: Histograma y correlograma spreads intradiarios históricos y simulados de BAC

El modelo CGAN logra capturar de forma adecuada las propiedades distribucionales de las series históricas. En la tabla 6.6 se muestran estadísticas descriptivas de los datos históricos y simulados para dos acciones: SWN y BAC. Se ratifica que el modelo CGAN logra modelar adecuadamente.

Tabla 6.6: Descripción de las series históricas intradiarias de spread de BAC y SWN

	SPREADS INTRADIARIOS			
	BAC		SWN	
	Histórico	Simulado	Histórico	Simulado
Media	0.079	0.042	0.063	0.060
Desviación Estándar	0.117	0.031	0.133	0.128
Asimetría	3.03	2.17	3.55	4.00
Curtosis	10.47	11.51	12.56	17.32
Máximo	0.800	0.633	0.800	0.993
Mínimo	0.009	0.000	0.000	0.000

7. CONCLUSIONES

A continuación se presentan los objetivos del presente trabajo final de maestría y las conclusiones relevantes para cada uno de ellos.

7.1 Objetivo Específico 1

Caracterizar modelos GAN para su uso en la generación de series de tiempo financieras sintéticas.

Los modelos GAN tienen una gran cantidad de usos prácticos debido a su capacidad de capturar las características distribucionales de datos como imágenes, sonidos, videos, series de tiempo, entre otros, sin requerir la modelación manual de características. Sin embargo, existen muchas variaciones de modelos adversarios generativos los cuales son adecuados o no según el tipo de datos a modelar. En el caso de las series de tiempo financieras, funcionó adecuadamente el uso de estructuras de GAN de los tipos DCGAN y cCGAN, y no funcionó bien el tipo WGAN.

La literatura sobre GAN es amplia y creciente, además de que es usual que se generen nuevas abreviaturas con cada modificación propuesta en los diferentes artículos. El modelador debe tener en cuenta que las características esenciales de los GAN son: capacidad de crear un modelo generador de datos y que su entrenamiento se realiza mediante una competencia de este modelo generador con un modelos discriminador a través de un juego de adversarios. Las modificaciones obedecen a la necesidad de: mejorar la optimización y estabilidad del GAN. 2) eludir problemas de desvanecimiento del gradiente o el problema del colapso en un modo (Mode Collapse). 3) Adecuar el GAN a la estructura de datos específica del problema a resolver.

Una de las principales dificultades en la evaluación de modelos GAN es la falta de un criterio estandarizado para calificar la calidad del modelo generador. En los problemas de imágenes se puede manejar la capacidad de engañar al ojo humano o engañar a modelos algorítmicos o neuronales usados para medir el desempeño. En los datos diferentes a las imágenes, debe el modelador establecer sus criterios de evaluación. En el caso de las series de tiempo financieras se optó por utilizar las siguientes métricas: media, desviación estándar, sesgo, curtosis, máximo, mínimo, histograma y autocorrelaciones, asumiendo que si una serie generada a través de una arquitectura GAN presenta unas métricas similares a las series históricas entonces pueden provenir del mismo proceso generador de datos, aunque el proceso en sí sea desconocido para el modelador.

7.2 Objetivo Específico 2

Construir los modelos GAN para generar series de tiempo financieras sintéticas con periodicidad diaria e intradiaria.

En este trabajo se pudo mostrar que ciertas configuraciones de modelos de Redes Generativas Adversarias capturan adecuadamente las características distribucionales de las series de tiempo diarias e intradiarias para acciones de Estados Unidos.

El modelo GAN generado para la simulación de rendimientos diarios de las acciones de Estados Unidos de América utilizó una arquitectura DCGAN. El submodelo generador utilizó capas convolucionales de una dimensión y toma datos aleatorios de una dimensión latente aleatoria y realiza un mapeo a una serie de datos generados. Es importante que el modelador elija correctamente el tipo de función de activación de la capa de salida del generador, para que sea consistente con la escala posible de valores de los datos a generar. A manera de ejemplo, los rendimientos financieros pueden tener valores positivos y negativos y por lo tanto una función de activación del tipo tangente hiperbólica puede ser posible, pero no debería usarse una función de activación del tipo sigmoide el cual es usualmente usado para las imágenes. (valores sólo positivos). También es muy importante realizar una transformación de los datos de entrada al modelo con el fin de llevarlos a una escala compatible con la función de activación a utilizar.

Para la modelación de las series diarias de precio con transformación fraccional, transformación hecha mantener el máximo de memoria de la serie sin perder la estacionariedad, se requiere escalar la serie para que las redes neuronales puedan manejar adecuadamente los datos. Se debió en este caso utilizar una arquitectura de DCGAN condicional con el fin modelar adecuadamente los niveles de precios de la serie. La variable auxiliar o condicional es el nivel de precios y el modelo cCGAN entonces realiza la modelación tanto del nivel de precios como del comportamiento de la serie en ese nivel.

En la modelación de los rendimientos, spread y volúmenes intradiarios, se utilizó una arquitectura DCGAN la cual fue adecuada, variando las funciones de activación según el rango de los datos a modelar, para la creación sintética de series de tiempo intradiarias.

7.3 Objetivo Específico 3

Simular series financieras sintéticas con periodicidad diarias e intradiarias

Después de repetir los ciclos de optimización de los modelos GAN con diferentes cantidades de épocas, se verificó que no existe en este tipo de modelación una convergencia estricta a un óptimo. El juego competitivo puede llevar a diferentes estados del modelo, algunos indeseados como pueden ser una victoria del generador o del discriminador, pero en un escenario deseado se encontrarán diferentes fluctuaciones en las pérdidas del modelo generador o discriminador donde cada uno mejora su desempeño a consta del otro. El modelador debe entonces después de superar un número mínimo de épocas, guardar varios modelos y evaluarlos todos a la luz de las métricas de desempeño elegidas. Algunos modelos generadores capturaban bien la estructura de autocorrelación, pero no el rango de valores. Otros modelos generadores manejaban bien la escala y los clusters pero no capturaban bien la curtosis. Entonces se selecciona el modelo generador que mejor represente a los datos.

7.4 Objetivo Específico 4

Verificar que las series de tiempo financieras generadas con los modelos GAN cumplen los hechos estilizados de las series financieras.

En la modelación de series de rendimientos financieros diarios, se utilizó un modelo GAN convolucional, el cual se probó en acciones de 4 acciones transadas en Estados Unidos, las cuales fueron seleccionadas de tal forma que presentaron diferencia en su comportamiento, usando conglomerados jerárquicos. Como resultado se pudo comprobar a través de la evaluación de histogramas, correlogramas y comparación de medias, desviación estándar, asimetría y curtosis, que el modelo generativo GAN logra capturar adecuadamente el proceso generador de datos para cada una de las acciones de la muestra. Entre los hechos estilizados verificados en las series simuladas se encuentran:

- Media de los rendimientos tendiendo a cero.
- Todas las desviaciones estándar de las series históricas diarias rondaron el valor del 2%. Este valor en sí mismo no es un hecho estilizado, pero si se aprecia que las series simuladas fueron consistentes en este comportamiento de las series históricas.
- Los rendimientos al cuadrado, como proxy de la volatilidad, presentan comportamiento de clusters
- Alto exceso de curtosis (presencia de colas pesadas)
- Baja autocorrelación
- La distribución de probabilidad de los rendimientos unido a las métricas anteriores permiten desestimar algún supuesto de normalidad en los rendimientos, aún sin calcular los estadísticos de Shapiro-Wilks de normalidad o de Shapiro Wilks, los cuales son innecesarios frente al elevado exceso de curtosis observado.

En cuanto a sesgo o asimetría, los resultados fueron mixtos en el sentido de en algunos casos se presentó ausencia de sesgo, y en otros un sesgo positivo.

Para las series de tiempo de los precios diarios de acciones con transformación fraccional, se usó un modelo cCGAN. Esta nueva serie de tiempo, es estacionaria según el criterio del estadístico ADF, pero a su vez presenta memoria. Al usar el modelo en un modelo CGAN no condicional se verificó que la modelación no fue adecuada, esto se debe a que pese a la transformación con diferencias fraccionales, los datos en los 20 años de datos de la muestra, presentan cambios de nivel importante y se podría considerar que no existe un solo proceso generador de datos. La red DCGAN intentó modelar esta serie de tiempo ajustando el nivel de precios a la media e incorporando una elevada varianza en los datos, produciendo series de tiempo muy diferentes a las series históricas. Se utilizó entonces un cCGAN, usando como variable condicional el nivel inicial de cada serie de tiempo, de tal manera que el modelo GAN pueda reconocer los diferentes procesos generadores de datos. La modelación mejoró y se pudieron generar series de tiempo sintéticas para cada nivel de precios.

Para la modelación de series intradiarias se usaron tres variables: rendimiento intradiario, volumen intradiario y spread observado intradiarios. Inicialmente se modeló con un wGAN el cual no logró captar bien las propiedades de las series históricas, especialmente el alto exceso de curtosis. Se usó entonces la arquitectura DCGAN con ligeras modificaciones para cada variable en la función de activación de la última capa del modelo generador. Las series simuladas tiene las siguientes características observadas en las series históricas intradiarias:

- Los rendimientos intradiarios presentan un mayo exceso de curtosis que los datos de rendimientos diarios.
- Los rendimientos intradiarios presentan una mayor autocorrelación que los datos de rendimientos diarios.

Los modelos GAN fueron más difíciles de entrenar con el fin de que el modelo generador pueda simular los elevados excesos de curtosis de los rendimientos intradiarios, se requierio entonces la generación y evaluación de mayor cantidad de modelos generadores hasta obtener algunos con una buena representación de los datos históricos.

En cuanto a la modelos GAN para los volúmenes transados de forma intradiaria se pudo verificar que las series generadas capturaban correctamente los siguientes hechos:

- Alta autorrelación
- Alta curtosis
- Alta asimetría positiva

Los volúmenes transados presentan una cola a la derecha y están trucadas en cero

por la izquierda.

El comportamiento de las series de spreads intradiarios simuladas, también presentan alta autocorrelación, una elevada curtosis (menor a los volúmenes transados), alta asimetría (menor a los volúmenes).

En consecuencia, se verificó que los modelos GAN permiten crear modelos generadores de series de tiempo sintéticas financieras diarias e intradiarias, que tienen un comportamiento en sus propiedades distribucionales, similar al comportamiento histórico de la serie real.

A manera de discusión, y como propuestas para estudios futuros se propone:

Evaluar modelos GAN condicional con variables condicionales tales como: año, mes, día de la semana, varianza de la serie de tiempo, entre otros. La ventaja de los modelos CGAN condicionales es que otorgan control al modelador y facilita el entrenamiento de modelos GAN con múltiples procesos generadores de datos.

La literatura financiera muestra que el desbalance del libro de órdenes presenta algún grado de capacidad de predicción de los rendimientos intradiarios. Se podría evaluar empíricamente la simulación simultánea con GAN de rendimientos intradiarios y desbalance del libro de órdenes con el fin de verificar si es posible generar series sintéticas que tengan esta relación.

También se propone la generación simultánea de series multivariadas, con modelos GAN condicionados o modelos de GAN de maximización de la información (InfoGAN) o modelos GAN con clasificador auxiliar (AC-GAN). Una aplicación práctica de uso de series multivariadas simuladas podría ser por ejemplo el entrenamiento de algoritmo de pair trading o conformación de portafolios sintéticos para arbitraje estadístico.

8. BIBLIOGRAFÍA

1. Sezer OB, Gudelek MU, Ozbayoglu AM. Financial time series forecasting with deep learning : A systematic literature review: 2005–2019. Applied Soft Computing. 2020. p. 106181. doi:10.1016/j.asoc.2020.106181

2. Kakushadze Z, Serur JA. 151 Trading Strategies. 2018. doi:10.1007/978-3-030-02792-6

3. Brooks C, Hoepner AGF, McMillan DG, Vivian A, Simen CW. Financial Data Science: The Birth of a New Financial Research Paradigm Complementing Econometrics? SSRN Electronic Journal. doi:10.2139/ssrn.3580729

4. White H. A Reality Check for Data Snooping. Econometrica. 2000. pp. 1097–1126. doi:10.1111/1468-0262.00152

5. Prado ML de, de Prado ML. Advances in Financial Machine Learning: Lecture 3/10. SSRN Electronic Journal. doi:10.2139/ssrn.3257419

6. Gooijer JGD, De Gooijer JG, Hyndman RJ. 25 years of time series forecasting. International Journal of Forecasting. 2006. pp. 443–473. doi:10.1016/j.ijforecast.2006.01.001

7. Box GEP, Jenkins GM. Time Series Analysis: Forecasting and Control, Revised Ed. 1976.

8. Cont R. Empirical properties of asset returns: stylized facts and statistical issues. Quant Finance. 2001;1: 223–236.

9. Tsay RS. Analysis of Financial Time Series. Wiley Series in Probability and Statistics. 2005. doi:10.1002/0471746193

10. Zhou X, Pan Z, Hu G, Tang S, Zhao C. Stock Market Prediction on High-Frequency Data Using Generative Adversarial Nets. Mathematical Problems in Engineering. 2018. pp. 1–11. doi:10.1155/2018/4907423

11. Dacorogna M et al. An Introduction to High-Frequency Finance. 2001. doi:10.1016/b978-0-12-279671-5.x5000-x

12. Rydberg TH. Realistic Statistical Modelling of Financial Data. Int Stat Rev. 2000;68: 233–258.

13. Cartea Á, Jaimungal S, Ricci J. Algorithmic Trading, Stochastic Control, and Mutually Exciting Processes. SIAM Review. 2018. pp. 673–703. doi:10.1137/18m1176968

14. Cartea Á, Jaimungal S. Modeling Asset Prices for Algorithmic and High Frequency Trading. SSRN Electronic Journal. doi:10.2139/ssrn.1722202

15. Agudelo DA, Giraldo S, Villarraga E. Does PIN measure information? Informed trading effects on returns and liquidity in six emerging markets. International Review of Economics & Finance. 2015. pp. 149–161. doi:10.1016/j.iref.2015.04.002

16. Glosten LR, Milgrom PR. Bid, ask and transaction prices in a specialist market with heterogeneously informed traders. Journal of Financial Economics. 1985. pp. 71–100. doi:10.1016/0304-405x(85)90044-3

17. Kyle AS. Continuous Auctions and Insider Trading. Econometrica. 1985. p. 1315. doi:10.2307/1913210

18. Aldridge I. High-Frequency Trading: A Practical Guide to Algorithmic Strategies and Trading Systems. John Wiley and Sons; 2009.

19. Lee CMC, Ready MJ. Inferring Trade Direction from Intraday Data. The Journal of Finance. 1991. pp. 733–746. doi:10.1111/j.1540-6261.1991.tb02683.x

20. Easley D, de Prado MML, O'Hara M. Flow Toxicity and Liquidity in a High-frequency World. Review of Financial Studies. 2012. pp. 1457–1493. doi:10.1093/rfs/hhs053

21. Vanstone B, Hahn T. Data Characteristics for High-Frequency Trading Systems. The Handbook of High Frequency Trading. 2015. pp. 47–57. doi:10.1016/b978-0-12-802205-4.00003-8

22. Sirignano J, Cont R. Universal Features of Price Formation in Financial Markets: Perspectives From Deep Learning. SSRN Electronic Journal. doi:10.2139/ssrn.3141294

23. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative Adversarial Networks. arXiv [stat.ML]. 2014. Available: http://arxiv.org/abs/1406.2661

24. Creswell A, White T, Dumoulin V, Arulkumaran K, Sengupta B, Bharath AA. Generative Adversarial Networks: An Overview. IEEE Signal Processing Magazine. 2018. pp. 53–65. doi:10.1109/msp.2017.2765202

25. Donahue J, Krähenbühl P, Darrell T. Adversarial Feature Learning. arXiv [cs.LG]. 2016. Available: http://arxiv.org/abs/1605.09782

26. Salehi P, Chalechale A, Taghizadeh M. Generative Adversarial Networks (GANs): An Overview of Theoretical Model, Evaluation Metrics, and Recent Developments. arXiv [cs.CV]. 2020. Available: http://arxiv.org/abs/2005.13178

27. Radford A, Metz L, Chintala S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv [cs.LG]. 2015. Available: http://arxiv.org/abs/1511.06434

28. Mirza M, Osindero S. Conditional Generative Adversarial Nets. arXiv [cs.LG]. 2014. Available: http://arxiv.org/abs/1411.1784

29. Chen X, Duan Y, Houthooft R, Schulman J, Sutskever I, Abbeel P. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. arXiv [cs.LG]. 2016. Available: http://arxiv.org/abs/1606.03657

30. Odena A, Olah C, Shlens J. Conditional Image Synthesis With Auxiliary Classifier GANs. arXiv [stat.ML]. 2016. Available: http://arxiv.org/abs/1610.09585

31. Odena A. Semi-Supervised Learning with Generative Adversarial Networks. arXiv [stat.ML]. 2016. Available: http://arxiv.org/abs/1606.01583

32. Brownlee J. Generative Adversarial Networks with Python: Deep Learning Generative Models for Image Synthesis and Image Translation. Machine Learning Mastery; 2019.

33. Makhzani A, Shlens J, Jaitly N, Goodfellow I, Frey B. Adversarial Autoencoders. arXiv [cs.LG]. 2015. Available: http://arxiv.org/abs/1511.05644

34. Dumoulin V, Belghazi I, Poole B, Mastropietro O, Lamb A, Arjovsky M, et al. Adversarially Learned Inference. arXiv [stat.ML]. 2016. Available: http://arxiv.org/abs/1606.00704

35. Larsen ABL, Sønderby SK, Larochelle H, Winther O. Autoencoding beyond pixels using a learned similarity metric. arXiv [cs.LG]. 2015. Available: http://arxiv.org/abs/1512.09300

36. Metz L, Poole B, Pfau D, Sohl-Dickstein J. Unrolled Generative Adversarial Networks. arXiv [cs.LG]. 2016. Available: http://arxiv.org/abs/1611.02163

37. Arjovsky M, Chintala S, Bottou L. Wasserstein Generative Adversarial Networks. In: Precup D, Teh YW, editors. International Convention Centre, Sydney, Australia: PMLR; 2017. pp. 214–223.

38. Petzka H, Fischer A, Lukovnicov D. On the regularization of Wasserstein GANs. arXiv [stat.ML]. 2017. Available: http://arxiv.org/abs/1709.08894

39. Zhu J-Y, Park T, Isola P, Efros AA. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. arXiv [cs.CV]. 2017. Available: http://arxiv.org/abs/1703.10593

40. Isola P, Zhu J-Y, Zhou T, Efros AA. Image-to-Image Translation with Conditional Adversarial Networks. arXiv [cs.CV]. 2016. Available: http://arxiv.org/abs/1611.07004

41. Ledig C, Theis L, Huszar F, Caballero J, Cunningham A, Acosta A, et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. arXiv [cs.CV]. 2016. Available: http://arxiv.org/abs/1609.04802

42. Wang X, Yu K, Wu S, Gu J, Liu Y, Dong C, et al. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. arXiv [cs.CV]. 2018. Available: http://arxiv.org/abs/1809.00219

43. Such FP, Rawal A, Lehman J, Stanley KO, Clune J. Generative Teaching Networks: Accelerating Neural Architecture Search by Learning to Generate Synthetic Training Data. arXiv [cs.LG]. 2019. Available: http://arxiv.org/abs/1912.07768

44. Borji A. Pros and Cons of GAN Evaluation Measures. arXiv [cs.CV]. 2018. Available: http://arxiv.org/abs/1802.03446

45. Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X, et al. Improved Techniques for Training GANs. In: Lee DD, Sugiyama M, Luxburg UV, Guyon I, Garnett R, editors. Advances in Neural Information Processing Systems 29. Curran Associates, Inc.; 2016. pp. 2234–2242.

46. Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. arXiv [cs.LG]. 2017. Available: http://arxiv.org/abs/1706.08500

47. Frid-Adar M, Klang E, Amitai M, Goldberger J, Greenspan H. Synthetic Data Augmentation using GAN for Improved Liver Lesion Classification. arXiv [cs.CV]. 2018. Available: http://arxiv.org/abs/1801.02385

48. Shrivastava A, Pfister T, Tuzel O, Susskind J, Wang W, Webb R. Learning from Simulated and Unsupervised Images through Adversarial Training. arXiv [cs.CV]. 2016. Available: http://arxiv.org/abs/1612.07828

49. Antoniou A, Storkey A, Edwards H. Data Augmentation Generative Adversarial Networks. arXiv [stat.ML]. 2017. Available: http://arxiv.org/abs/1711.04340

50. Motamed S, Khalvati F. Inception Augmentation Generative Adversarial Network. arXiv [cs.CV]. 2020. Available: http://arxiv.org/abs/2006.03622

51. Lee H, Kim J, Kim EK, Kim S. Wasserstein Generative Adversarial Networks Based Data Augmentation for Radar Data Analysis. NATO Adv Sci Inst Ser E Appl Sci. 2020;10: 1449.

52. Zhang K, Zhong G, Dong J, Wang S, Wang Y. Stock Market Prediction Based on Generative Adversarial Network. Procedia Comput Sci. 2019;147: 400–406.

53. Koshiyama A, Firoozye N, Treleaven P. Generative Adversarial Networks for Financial Trading Strategies Fine-Tuning and Combination. arXiv [cs.LG]. 2019. Available: http://arxiv.org/abs/1901.01751

54. Wiese M, Knobloch R, Korn R, Kretschmer P. Quant GANs: deep generation of financial time series. Quantitative Finance. 2020. pp. 1–22. doi:10.1080/14697688.2020.1730426

55. Yoon J, Jarrett D, van der Schaar M. Time-series Generative Adversarial Networks. In: Wallach H, Larochelle H, Beygelzimer A, d\textquotesingle Alché-Buc F, Fox E, Garnett R, editors. Advances in Neural Information Processing Systems 32. Curran Associates, Inc.; 2019. pp. 5508–5518.

56. Esteban C, Hyland SL, Rätsch G. Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs. arXiv [stat.ML]. 2017. Available: http://arxiv.org/abs/1706.02633

57. Hartmann KG, Schirrmeister RT, Ball T. EEG-GAN: Generative adversarial networks for electroencephalograhic (EEG) brain signals. arXiv [eess.SP]. 2018. Available: http://arxiv.org/abs/1806.01875

58. Wiese M, Bai L, Wood B, Buehler H. Deep Hedging: Learning to Simulate Equity Option Markets. arXiv [q-fin.CP]. 2019. Available: http://arxiv.org/abs/1911.01700

59. Takahashi S, Chen Y, Tanaka-Ishii K. Modeling financial time-series with generative adversarial networks. Physica A: Statistical Mechanics and its Applications. 2019;527: 121261.

60. Guo Z, Wan Y, Ye H. A data imputation method for multivariate time series based on generative adversarial network. Neurocomputing. 2019;360: 185–197.

61. Marti G. CORRGAN: Sampling Realistic Financial Correlation Matrices Using Generative Adversarial Networks. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2020.

doi:10.1109/icassp40776.2020.9053276

62. Li D, Chen D, Jin B, Shi L, Goh J, Ng S-K. MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks. Artificial Neural Networks and Machine Learning – ICANN 2019: Text and Time Series. 2019. pp. 703–716. doi:10.1007/978-3-030-30490-4_56

ANEXO 1 - LISTADO DE ACCIONES

SECTOR	NUMERO DE ACCIONES
Basic Materials	4
Communication Services	10
Consumer Cyclical	12
Consumer Defensive	2
Energy	26
ETF	13
Financial Services	7
Healthcare	6
Industrials	8
Real Estate	3
Technology	6
Utilities	1
Suma total	**98**

TICKER	EMPRESA	SECTOR
AAL	American Airlines Group Inc	Industrials
ABEV	Ambev ADR	Consumer Defensive
AMCR	Amcor PCL	Consumer Cyclical
AMD	Advanced Micro Devices Inc	Technology
APTV	Aptiv PC	Consumer Cyclical
AR	Antero Resources Corporation	Energy
ATNM	Actinium Pharmaceuticals Inc	Healthcare
AZEK	The AZEK Company Inc	Industrials
BA	Boeing	Industrials
BAC	Bank of America Corp	Financial Services
BBD	Banc Bradesco ADR	Financial Services
BKD	Brookdale Senior Living Inc	Healthcare
BSX	Boston Scientific Corporation	Healthcare
C	Citigroup	Financial Services
CCL	Carnival Corp	Consumer Cyclical

CDEV	Centennial Resource Development	Energy
CHK	Chesapeake Energy Corporation	Energy
CLDX	Celldex Therapeutics Inc	Healthcare
CMCSA	Comcast Corp	Communication Services
CNP	Centerpoint Energy Inc	Utilities
COTY	Coty Inc	Consumer Defensive
CPE	Callon Petroleum Co	Energy
CSCO	Cisco	Technology
CTLT	Catalent Inc	Healthcare
CX	Cemex SAB de CV	Basic Materials
DAL	Delta Air Lines Inc	Industrials
DNR	Denbury Resources Inc	Energy
EEM	Emerging Markets Index MSCI Ishares	ETF
EFA	EAFE Index MSCI Ishares	ETF
ET	Energy Transfer Equity LP	Energy
EWZ	Brazil Index MSCI Ishares	ETF
F	Ford Motor Co	Consumer Cyclical
FB	Facebook	Communication Services
FCX	Freeport-McMoRan Inc	Basic Materials
FET	Forum Energy Technologies Inc	Energy
FWONK	Formula One Group	Communication Services
GE	General Electric	Industrials
GLPI	Gaming and Leasure	Real Estate
GNUS	Genius Brands International Inc	Communication Services
GOVT	iShares U.S. Treasury Bond ETF	ETF
GPOR	Gulfport Energy Corporation	Energy
GPS	GAP Inc	Consumer Cyclical
GRUB	GrubHub Inc	Communication Services
GTE	Gran Tierra Energy Inc	Energy
HPE	Hewlett Packard Enterprise Co	Technology
HPQ	Hewlett Packard	Technology

HTZ	Hertz Global Holding Inc	Industrials
HYG	Iboxx High Yield Corporate Bonds Ishares	ETF
IEFA	Ishares Core MSCI EAFE ETF	ETF
ITUB	Itau Unibanco Holding ADR	Financial Services
IVR	Invesco Mortgage Capital Inc	Real Estate
IWM	Ishares Russell 2000 ETF	ETF
KEY	KeyCorp	Financial Services
KGC	Kinross Gold Corporation	Basic Materials
KMI	Kinder Morgan Inc	Energy
LK	Luckin Coffee Inc	Consumer Cyclical
LQD	iShares iBoxx Investment Grade Corporate Bond	ETF
M	Macys Inc	Consumer Cyclical
MIK	The Michael Companies Inc	Consumer Cyclical
MRO	Marathon Oil Corp	Energy
NAKD	Naked Brand Group Limited	Consumer Cyclical
NBL	Noble Energy Inc	Energy
NCLH	Norwegian Cruise Line Holdings	Consumer Cyclical
NE	Noble Corporation plc	Energy
NIO	Nio Inc	Consumer Cyclical
NKLA	Nikola Corporation	Consumer Cyclical
NOG	Northern Oil and Gas Inc	Energy
OAS	Oasis Petroleum Inc	Energy
OKE	ONEOK Inc	Energy
OXY	Occidental Petroleum Corp	Energy
PACD	Pacific Drilling SA	Energy
PE	Parsley Energy Inc A	Energy
PFE	Pfizer	Healthcare
QEP	QEP Resources Inc	Energy
RIG	Transocean LTD	Energy
SABR	Sabre Corporation	Technology
SCHW	SCharles Schwab Corp	Financial Services

SHIP	Seanergy Maritime Holdings Corp	Industrials
SIRI	Sirium XM Holding Inc	Communication Services
SLV	Ishares Silver Trust	ETF
SM	SM Energy Co	Energy
SNAP	Snapchat Inc	Communication Services
SPXS	Direxion Daily S&P 500 Bear 3X Shares	ETF
SPY	SPDR S&P 500 ETF	ETF
SQQQ	ProShares UltraPro Short QQQ	ETF
SWN	Southwestern Energy Co	Energy
T	AT&T Inc	Communication Services
TLRD	Tailored Brands Inc	Consumer Cyclical
TOPS	Top Ships Inc	Industrials
TWO	Two Harbors Investment Corp	Real Estate
UAL	United Airlines Holdings Inc	Industrials
UBER	Uber	Technology
VAL	Valaris plc	Energy
VALE	Vale SA	Basic Materials
VEON	VEON Ltd	Communication Services
VRM	Vroom Inc	Consumer Cyclical
WFC	Wells Fargo & Co	Financial Services
WLL	Whiting Petroleum Corp	Energy
XLF	Financial Select Sector SPDR	ETF
XOG	Extraction Oil & Gas inc	Energy
ZNGA	Zynga	Communication Services

ANEXO 2 - ÍNDICE DE ABREVIATURAS

Abreviatura	Significado
AAE	Adversarial Autoencoders
ACF	Parcial Autocorrelation Function
ACGAN	Auxiliary Classifier GAN
ALI	Adversarially learned inference
BiGAN	Bidirectional GAN
cCGAN - cGAN	Conditional Convolutional Generative Adversarial Network
CGAN - DCGAN	Deep Convolutional Generative Adversarial Network
DL	Deep Learning
GAN	Generative Adversarial Network
infoGAN	Information maximizing GAN
LSTM	Long Short Term Memory
ML	Machine Learning
RGAN	Recurrent GAN
RL	Reinforcement Learning
SGAN	semi-supervised GAN
test ADF	Test Augmented Dickey Fuller
TGAN	Temporal GAN
WGAN	Wasserstein GAN