



Facultad de Ciencias Económicas y Empresariales

**ANÁLISIS Y DETECCIÓN DE ANOMALÍAS USANDO
TÉCNICAS DE DEEP LEARNING: CASO DE
ESTUDIO EN COTIZACIONES DE PFIZER Y
MODERNA**

Autor: Ignacio Arnott Iglesias

Tutora: Jenny Alexandra Cifuentes Quintero

Madrid, Junio 2022

**ANÁLISIS Y DETECCIÓN DE ANOMALÍAS USANDO
TÉCNICAS DE DEEP LEARNING: CASO DE
ESTUDIO EN COTIZACIONES DE PFIZER Y
MODERNA**

Autor: Ignacio Arnott Iglesias

Tutora: Jenny Alexandra Cifuentes Quintero

Madrid, Junio 2022

Universidad Pontificia Comillas

Facultad de Ciencias Económicas y Empresariales



COMILLAS
UNIVERSIDAD PONTIFICIA

RESUMEN

Por medio de técnicas de machine learning, se realizará la detección de anomalías en el precio de las acciones de las compañías Pfizer y Moderna, y se analizará, de forma temporal, cada uno de los eventos encontrados. Para cada uno de los puntos anómalos identificados, se realizará un análisis de la información reportada en los medios con el fin de establecer los eventos que pudiesen causar algún impacto.

ABSTRACT

By means of machine learning techniques, anomaly detection in the share prices of the companies Pfizer and Moderna will be carried out, and each of the events found will be analyzed temporarily. For each of the anomalous points identified, an analysis of the information reported in the media will be carried out in order to establish the events that could cause some impact.

PALABRAS CLAVE

Machine learning, Algoritmos, Base de datos, Anomalías, Detección, Investigación.

KEYWORDS

Machine learning, Algorithms, Database, Anomalies, Detection, Investigation.

Agradecimientos

Primero que nada, quiero agradecer a mi familia el apoyo incondicional que me han brindado durante toda mi etapa universitaria. Gracias Juan, Norma y Sofía.

Finalmente, quiero agradecer a mi tutora oficial, Jenny Alexandra Cifuentes, por el tiempo dedicado en tutorías, charlas y correcciones sobre el trabajo. Gracias por tu amabilidad y paciencia.

ÍNDICE DE CONTENIDOS

1 INTRODUCCIÓN	10
1.1 MOTIVACIÓN	10
1.2 OBJETIVOS	12
1.3 ORGANIZACIÓN DE LA MEMORIA	13
2 ESTADO DEL ARTE	14
2.1 TIPOS DE DETECCIÓN DE ANOMALÍAS	15
2.1.1 Detección de anomalías de manera supervisada	15
2.1.2 Detección de anomalías semi-supervisadas	16
2.1.3 Detección de anomalías no supervisadas	16
2.2 DESAFÍOS EN LA DETECCIÓN DE ANOMALÍAS	18
3 DETECCIÓN DE ANOMALÍAS EN SERIES TEMPORALES	20
3.1 TIPOS DE DATOS: SERIES DE TIEMPO	21
3.1.1 PATRONES DE SERIES TEMPORALES	21
3.1.2 DESCOMPOSICIÓN DE SERIES DE TIEMPO	24
3.2 TIPOS DE ANOMALÍAS	25
3.3 TIPOS DE ENFOQUE EN LA DETECCIÓN DE ANOMALÍAS PARA SERIES TEMPORALES	27
3.3.1 ENFOQUE ESTADÍSTICO	27
3.3.1.1 Modelo autorregresivo (AR)	28
3.3.1.2 Modelo Média Móvil (MA)	29
3.3.1.3 Modelo Autorregresivo Média Móvil (ARMA)	29
3.3.1.4 Modelo ARIMA	29
3.3.1.5 Suavizado exponencial (SES)	30
3.3.1.6 Suavizado doble exponencial (DES)	30
3.3.1.7 Suavizado triple exponencial (TES)	30
3.3.2 ENFOQUE MACHINE LEARNING CLÁSICO	31
3.3.2.1 K means clustering	32
3.3.2.2 Clustering basado de densidad	32
3.3.2.3 Algoritmo local outlier factor (LOF)	32
3.3.2.4 Bosque de aislamiento	33
3.3.3 ENFOQUE MACHINE LEARNING - DEEP LEARNING	33
3.3.3.1 Backpropagation	36
3.3.3.2 Descenso de gradiente (gradient descent)	36
3.3.3.3 Convolutional Neural Networks (CNN)	37
3.3.3.4 Red Neuronal Residual	37
3.3.3.5 Red Neuronal Recurrente (RNN)	38
4 METODOLOGÍA	43
4.1 ENUNCIADO DEL PROBLEMA	43
4.2 INSTALACIÓN LIBRERÍAS	44
4.3 IMPLEMENTACIÓN Y DESARROLLO DE LA BASE DE DATOS	45

4.3.2 SELECCIÓN DE VENTANAS DE ENTRENAMIENTO - PRUEBA	46
4.3.3 ESCALAMIENTO DE LOS DATOS Y GENERACIÓN DE LOS CONJUNTOS DE DATOS	49
4.3.4 MODELO AUTOENCODER LSTM	50
4.3.4.1 ENTRENAMIENTO	50
4.3.4.2 THRESHOLD	51
5 RESULTADOS CASO DE ESTUDIO	53
5.1 ADQUISICIÓN Y PROCESAMIENTO DE LOS DATOS	53
5.2 ENTRENAMIENTO DEL MODELO	55
5.3 SELECCIÓN DEL UMBRAL DE DETECCIÓN	57
5.3 ANOMALÍAS RESULTANTES	59
6 CONCLUSIONES	65

1 INTRODUCCIÓN

1.1 MOTIVACIÓN

Hoy en día, vivimos en un mundo de digitalización donde la cantidad de datos es tan extensa que supera la capacidad humana para estudiarlos manualmente. Este crecimiento implica la necesidad de crear análisis automatizados que permitan detectar, analizar y comprender los datos. Los avances tecnológicos en el área de la informática, incluidos los sistemas de bases de datos, el aprendizaje automático y el almacenamiento en la nube, han potenciado que los datos se conviertan en información y conocimiento de gran valor para respaldar la toma de decisiones [1]. Estos avances han ayudado al desarrollo de sistemas informáticos capaces de almacenar, analizar y gestionar una cantidad de datos cada vez mayor. Esta serie de datos se pueden representar de distintas formas, desde los más sencillos, como numéricos y nominales, hasta los más complejos, como datos de audio y vídeo. Sin embargo, la detección y análisis de la información temporal, que permite la organización cronológica de los datos recolectados, es una de las representaciones más famosas entre los investigadores y que más ha impulsado la creación de inmensas bases de datos.

Una de las tareas más relevantes del análisis de datos temporales es la detección de anomalías, sobre todo gracias a su gran variedad de aplicaciones [2], que incluyen la salud médica, el fraude con tarjetas de crédito, la seguridad informática y la limpieza de datos. Por ejemplo, una transmisión de red anormal puede suponer que un sistema informático sea atacado por piratas informáticos o un virus, una transacción anómala de una tarjeta de crédito puede implicar un uso no autorizado de ella, o la detección de una actividad geológica inesperada en la naturaleza, puede ser precursora de un terremoto o un tsunami.

La minería de datos temporales es un proceso para obtener información útil de series temporales. La detección es una de las actividades en la minería de datos, la cual está motivada por el desafío de reducir la incertidumbre de las anomalías, especialmente debido a la volatilidad de determinados fenómenos [3]. Desde el punto de vista financiero, una aplicación de gran interés es la detección de un cambio brusco del precio de una acción en un día en concreto o un aumento considerado del desempleo en un trimestre del año. De esta

manera, la detección y análisis de las anomalías son un estudio fundamental en muchas actividades para la toma de decisiones, como estrategias socioeconómicas, financieras y comerciales, siempre y cuando se haga buen uso de estas.

Las herramientas para la detección de anomalías en las series temporales se basan en que existe información histórica acerca del fenómeno investigado que permite realizar un análisis y posteriormente establecer conclusiones. El diseño de un modelo para la detección de anomalías en series temporales se centra en la aplicación de algoritmos. Se suponen ciertas condiciones sobre los datos y el modelo concreta las variables involucradas, resumiéndolas en una estructura matemática potencialmente flexible.

Existen varios enfoques que permiten llevar a cabo este tipo de análisis sobre series temporales, el más conocido es la aplicación de métodos estadísticos como el modelo de autorregresión y medias móviles, que llevan desarrollándose durante más de cincuenta años [4]. Los algoritmos creados a partir de estos métodos asumen una distribución conocida de los datos, y los parámetros de la función se concretan para ajustarles un modelo, pero el desarrollo de este tipo de metodologías requiere un profundo conocimiento matemático.

Durante los últimos años, varios investigadores han empleado modelos no paramétricos de aprendizaje automático (*Machine Learning*) para el estudio de series temporales, principalmente porque este enfoque no presupone la naturaleza de los datos. También ha surgido una rama del *Machine Learning* conocida como *Deep Learning*, la cuál va aún más allá e introduce un enfoque asociado a modelos mucho más complejos, inspirados en la estructura y función del cerebro, conocidos como redes neuronales artificiales. A pesar de que no se haya demostrado empíricamente que los modelos de *Machine Learning* siempre superan los estadísticos en cuanto a resultados, numerosas investigaciones indican que los algoritmos de *Machine Learning*, en general, son más competitivos para realizar estudios de series temporales.

En este trabajo de fin de grado, se quiere profundizar sobre una técnica de *Deep Learning* que ha demostrado su eficacia en la detección de anomalías en series de tiempo, además, se pretende implementar y estudiar un caso de uso de interés en el ámbito financiero. A continuación, se presentan detalladamente los objetivos del trabajo y se realiza la descripción detallada de la memoria.

1.2 OBJETIVOS

El principal objetivo de este trabajo es explorar las técnicas basadas en Deep Learning para la detección de anomalías en series de tiempo y consecuentemente aplicarlo a un caso de estudio, en el que se detecte, analice y contraste el resultado de anomalías en el precio de las acciones de dos compañías farmacéuticas: Pfizer y Moderna. Este proceso se realizará a través de un modelo de *Deep Learning* denominado *Autoencoder LSTM*. Para ello, se van a estudiar los siguientes objetivos específicos:

- Planificar y realizar una revisión sistemática y un análisis del estado del arte correspondiente a la aplicación desarrollada.
- Realizar un análisis de los conceptos asociados a la detección de anomalías en series de tiempo
- Estudiar y comparar diferentes metodologías para la detección de anomalías como el enfoque estadístico, el enfoque de *Machine Learning* y, específicamente, de *Deep Learning*.
- Profundizar en los conceptos asociados al desarrollo e implementación de redes neuronales artificiales
- Posteriormente, implementar un algoritmo de Deep Learning para la detección de anomalías en series temporales
- Detectar y analizar las anomalías existentes en la base de datos de Pfizer y Moderna con el objetivo de interpretar los resultados y establecer conclusiones.

1.3 ORGANIZACIÓN DE LA MEMORIA

La memoria está compuesta de los siguientes capítulos:

- **Introducción:** en este capítulo se introduce el tema de estudio, las diferentes aplicaciones y metodologías en la detección de anomalías y un breve resumen del modelo a desarrollar en este trabajo.
 - **Estado del arte:** que incluye el estado actual de la detección de anomalías y todas las definiciones teóricas relevantes para el trabajo, incluidos los distintos tipos de detección de anomalías.
 - **Detección de anomalías en series de tiempo:** en este capítulo se describe el marco teórico del presente trabajo de fin de grado.
 - **Metodología:** el cual explica el diseño que se realizó para desarrollar la implementación del caso de estudio, destacando la metodología usada para la técnica analizada.
 - **Resultados del caso de uso:** en el cual se comentan los resultados de desarrollo de cada etapa, junto con los resultados finales.
 - **Conclusiones:** se establecen una serie de conclusiones sobre la temática tratada y del caso de uso.
-

2 ESTADO DEL ARTE

El análisis de anomalías es de gran interés para diversos campos de investigación, incluida la minería de datos y el aprendizaje automático. Su objetivo es identificar a partir de una base de datos aquellas regiones cuyos comportamientos o patrones no se ajustan a los valores esperados [5]. Los comportamientos inesperados, que son significativamente diferentes a los del resto de los datos proporcionados, se denominan comúnmente anomalías. No obstante, no existe una definición formal ampliamente aceptable de este concepto. En la literatura, una anomalía también se conoce como un valor atípico, un objeto discordante, una excepción, una aberración o una peculiaridad.

Probablemente, la primera definición fue dada por Grubbs en 1969: *Una observación atípica es aquella que parece desviarse notablemente de otros miembros de la muestra en la que ocurre*. Aunque esta definición sigue siendo válida hoy en día, la motivación para detectar estos valores atípicos es muy diferente ahora. En aquel entonces, el motivo principal de la detección de las anomalías era eliminarlas de los datos de entrenamiento, ya que los algoritmos de reconocimiento de patrones eran muy sensibles a los valores atípicos en los datos. Este procedimiento también se denomina limpieza de datos. Después del desarrollo de clasificadores más robustos, el interés en la detección de anomalías disminuyó considerablemente. Sin embargo, hubo un punto de inflexión alrededor del año 2000, cuando los investigadores comenzaron a interesarse más en las anomalías en sí, ya que a menudo se asocian con eventos particulares interesantes o registros de datos sospechosos [6]. Desde entonces, se han desarrollado numerosos algoritmos. En este contexto, la definición de Grubbs también se amplió de tal manera que hoy en día se sabe que las anomalías tienen dos características importantes:

1. Las anomalías son diferentes de la norma con respecto a sus características.
2. Son “atípicas” en un conjunto de datos en comparación con los datos normales.

Los algoritmos de detección de anomalías ahora se utilizan en muchos dominios de aplicaciones y, a menudo, mejoran los sistemas tradicionales de detección basados en reglas.

Particularmente, en la detección de anomalías, las soluciones generalmente se clasifican en tres aspectos según la naturaleza del aprendizaje, incluida la detección de anomalías

supervisada, no supervisada y parcialmente supervisada. A continuación se explican en detalle las características de cada uno de ellos.

2.1 TIPOS DE DETECCIÓN DE ANOMALÍAS

2.1.1 *Detección de anomalías de manera supervisada*

Una vez conocemos las anomalías existentes en nuestros datos, utilizamos las técnicas de clasificación supervisada. A la hora de realizar un análisis y detección de anomalías se tienen dos conjuntos de datos, unos de entrenamiento y otros de test. Como se conoce toda la información necesaria, estos datos se etiquetan en función de si son una anomalía o un dato normal dentro del umbral que hemos seleccionado. En primer lugar, las etiquetas de datos generalmente no están disponibles o son demasiado costosas para obtenerlas en muchos escenarios. En segundo lugar, es posible que las etiquetas de datos no estén balanceadas debido a que las muestras normales superan en gran medida a las anormales, lo que da como resultado un sesgo importante en el modelo de clasificación que puede degradar el rendimiento de la detección de anomalías. Este tipo de problemas se conocen como clasificación desbalanceada.

No obstante, estos problemas se pueden solucionar a través de métodos basados en instancias y basados en algoritmos. Por lo que una vez recopilados estos datos que se han mencionado, se construye un modelo capaz de aprender las diferencias entre la información normal y anómala. Teóricamente, la detección supervisada de anomalías es generalmente más precisa debido a la clara comprensión de la normalidad y la anomalía.

La detección de anomalías supervisada en series temporales es muy común en la actualidad para modelos de *Deep Learning*, en el artículo *Efficient time series anomaly detection by multiresolution self-supervised discriminative network* [7], presentan una red discriminativa supervisada de resolución múltiple (MS2D-Net) para la detección eficiente de anomalías de series temporales, en donde los resultados experimentales muestran que MS2D-Net tiene un costo computacional mucho más bajo que otras líneas llevadas a cabo en métodos de *Deep learning*.

2.1.2 Detección de anomalías semi-supervisadas

En las técnicas semi-supervisadas se conoce la existencia de las anomalías, pero estas no se contemplan en el conjunto de datos. Con el conjunto de datos se llega a definir la normalidad del ambiente, un supuesto en el que es necesario establecer un comportamiento normal se da a cabo cuando hay tráfico en datos de red, en el que para afrontar este problema hay que saber modelar el comportamiento con el objetivo de detectar las anomalías. Para ello existen varias técnicas, basadas en clasificación, en reglas de asociación y en máquinas de soporte vectorial.

Los modelos de clasificación en el conjunto de datos equilibrados se ven perjudicados por cometer demasiados falsos positivos, este es el caso de cuando un nuevo dato entra en el modelo y este es clasificado incorrectamente, de manera que se considera una anomalía y no tiene porqué ser cierto. Este problema se soluciona con clasificadores basados en reglas, los cuales proporcionan más información acerca de una errónea clasificación, como en qué grado se considera dicho dato anómalo, por ejemplo, en un 70%. Otra manera de abordar el problema es a través de métodos basados en reglas, se consigue buscando patrones frecuentes y reglas de asociación. Un modelo es *LERAD*, el cual comprende las reglas que detectan eventos anómalos en una serie temporal. Por último, existe el método de las máquinas de soporte vectorial, el cuál asume una región de datos preestablecida en el espacio a través del cuál se considera todo punto que caiga en esa región como dato normal [6]. Por tanto, todo dato que caiga fuera de la región se considera anómalo.

El uso de técnicas de detección de anomalías semi-supervisadas en series temporales ha permitido desarrollar y mejorar procesos en numerosas actividades, en el artículo *Semi-supervised Anomaly Detection with an Application to Water Analytics* se desarrolla un enfoque novedoso basado en agrupaciones restringidas para la detección de anomalías que se aplica en el monitoreo de agua de supermercados para conocer el uso del agua.

2.1.3 Detección de anomalías no supervisadas

Las técnicas no supervisadas se emplean normalmente en una situación en la que no se conoce ningún conocimiento previo del conjunto de datos, por lo tanto no se presenta información de la etiqueta. En este caso, existen varias técnicas alternativas como:

aproximaciones gráficas, aproximaciones paramétricas, aproximaciones basadas en vecinos cercanos y aproximaciones basadas en clustering.

El método basado en aproximaciones gráficas, como su nombre indica, consiste en observar visualmente los datos con el fin de detectar los datos anómalos. Es una técnica cuyo principal inconveniente es la cantidad de tiempo que es necesario invertir y la subjetividad con la que se juzgan los datos. Una de las aproximaciones más útiles es el *biplot* [7], la cuál muestra una proyección a dos dimensiones de la varianza que aporta cada dato.

Las aproximaciones paramétricas se obtienen a través de un modelo paramétrico que describe la distribución de los datos y utiliza test estadísticos para establecer si un punto es una anomalía y su nivel de significancia. En esta técnica se encuentran varios tipos de test como el *test de Grubb* y el *test Tietjen y Moore*.

Las aproximaciones gráficas y paramétricas asumen una distribución normal de los datos, pero existen casos en los que si se añade, por ejemplo, un número de dimensiones muy altas, los datos no tienden a seguir una distribución multivariante específica [8]. En este caso hay dos posibilidades de obtener los vecinos, mediante una función de distancia entre dos puntos o asignando una puntuación de anomalía a un punto con respecto a su distancia frente al resto de puntos vecinos.

Finalmente, las técnicas de clustering miden la distancia de un punto a su baricentro¹ para determinar si se trata de una anomalía. Esta distancia se debe establecer con la distancia relativa, la cual relaciona la distancia del baricentro a la distancia mediana de todos los puntos del cluster desde el baricentro.

La detección de anomalías en series de tiempo con técnicas no supervisadas han superado numerosos problemas de las técnicas tradicionales, en el artículo *DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series* [9] explican cómo han abordado el problema para detectar anomalías puntuales periódicas y relacionadas con la estacionalidad que ocurren comúnmente en la transmisión de datos. Otro supuesto de uso de esta técnica en la detección de anomalías de series temporales lo encontramos en el artículo

¹ Punto donde se considera concentrada el área total de una figura, donde se supone está ubicado el centro geométrico del cuerpo plano y homogéneo.

FluxEV: A Fast and Effective Unsupervised Framework for Time-Series Anomaly Detection [10], en el cual realizan una investigación de la técnica *FluxEV*, una técnica de detección de anomalías no supervisado basado en aproximaciones paramétricas rápido y eficaz, el cual convierte las anomalías no extremas en valores extremos con el objetivo de lograr una gran mejora en la precisión de detección. No solo encontramos artículos que mejoren procesos o detecten las anomalías con una mayor precisión, también existen aplicaciones prácticas que gracias a estas mejoras consiguen aplicarlo en casos de la vida cotidiana, como en el trabajo *Learning Representations from Healthcare Time Series Data for Unsupervised Anomaly Detection* [11], en el cuál se desarrolla un marco no supervisado para la detección de anomalías en datos de series temporales usando *clustering* para detectar latidos cardíacos anómalos de una manera totalmente no supervisada, al tiempo que proporciona representaciones de datos estructurados y expresivos.

2.2 DESAFÍOS EN LA DETECCIÓN DE ANOMALÍAS

En resumen, el comportamiento anómalo es una desviación del comportamiento normal. Por lo tanto, para detectar desviaciones del comportamiento normal, se necesita un criterio o medida para definirlo. Sin embargo, definir un rango o criterio para el comportamiento normal y distinguirlo del comportamiento anormal es muy complicado en la práctica. En esta sección, explicaremos una serie de desafíos que se encuentran a la hora de desarrollar trabajos relacionados con la detección de anomalías en series temporales:

- **Definir la región normal:** resulta especialmente desafiante cuando necesitamos trazar una línea entre el conjunto de datos normales y el conjunto de datos anómalos. En muchos casos, los datos en una región normal que están cerca de un límite pueden ser indicador de una anomalía, y viceversa.
- **Detección de acciones maliciosas:** El comportamiento malicioso se puede adaptar para imitar el comportamiento normal en el sistema, lo que dificulta distinguir el comportamiento normal del comportamiento anormal.
- **Diferentes actividades requieren diferentes nociones:** una anomalía se observa desde distintos puntos de vista según el área de aplicación. Por ejemplo, mientras que las fluctuaciones en los valores del mercado de valores pueden considerarse normales, en el campo de la medicina, las ligeras desviaciones de los valores normales en un

análisis de sangre pueden considerarse anormales. Esto puede generar que se aplique una técnica eficiente de detección de anomalías en una aplicación, en otras que no son adecuadas.

- **Distinguir el ruido de las anomalías:** el ruido puede llegar a ser similar al conjunto de datos anómalos, lo que hace que la diferenciación entre estos dos sea una tarea desafiante.

En este capítulo hemos comentado el estado actual de la detección de anomalías y todas las definiciones teóricas relevantes para el trabajo, así como las distintas técnicas de detección de anomalías y sus diferentes y versátiles aplicaciones en diferentes campos. Finalmente, planteamos los desafíos que conlleva desarrollar estudios en la detección de anomalías. A continuación, se realizará la descripción del marco teórico de este campo, necesario para el desarrollo del caso de uso.

3 DETECCIÓN DE ANOMALÍAS EN SERIES TEMPORALES

Para seleccionar y aplicar un método de detección de anomalías apropiado, se deben considerar varios factores, incluida la naturaleza del flujo de datos generado, el tipo de anomalía y la disponibilidad de los datos de entrenamiento. Presentamos un diagrama de bloques sistemático del sistema de detección de anomalías en la Figura 1.

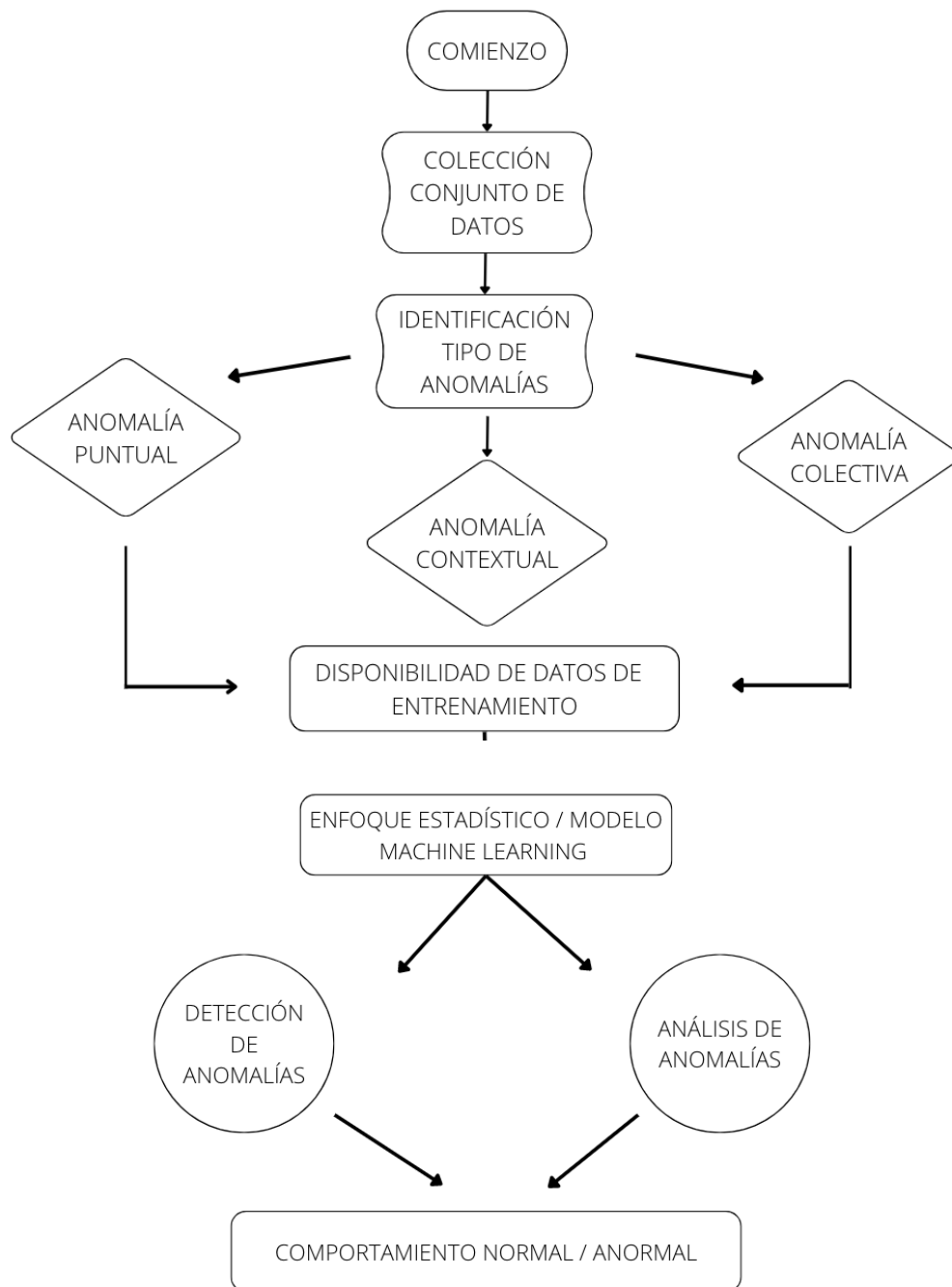


Figura 1. Esquema del proceso en la detección de anomalías. Fuente: elaboración propia

En un sistema de detección de anomalías (como se muestra en la Figura 1), el primer paso es comprender la naturaleza del flujo de datos recopilados, los cuales pueden ser binarios, continuos o discretos. Además, los datos presentan esta estructura de relación con su naturaleza, que indica si se trata de datos de series temporales, datos espaciales o datos gráficos. Esta dependencia ayuda a seleccionar el método apropiado de detección, análisis o predicción de anomalías. El segundo paso es determinar el tipo de anomalía del conjunto predefinido (es decir, anomalía puntual, anomalía contextual y anomalía colectiva). Posteriormente, con base en el tipo de datos y el tipo de anomalía, se realiza la selección de la técnica de aprendizaje de máquina que mejor se adapta a estas características. Finalmente se realiza la implementación para la detección y análisis de anomalías.

3.1 TIPOS DE DATOS: SERIES DE TIEMPO

Un aspecto clave y fundamental de cualquier técnica de detección de anomalías es la naturaleza del conjunto de datos de destino. Esencialmente, un conjunto de datos es una colección de instancias de datos u observaciones. Según los escenarios de aplicación específicos, un conjunto de datos puede ser un número, un registro, un video, una canción, un gráfico, una imagen, un evento y un perfil. Todas estas formas dispares de datos deben transformarse en tipos de datos generales para la detección de anomalías. En el estudio de caso abordado en este trabajo se trabajará particularmente con series temporales.

Una serie temporal es un conjunto de observaciones de una misma variable ordenadas en el tiempo. Existen una variedad de patrones que ayudan a reflejar el resultado de los datos en una serie de tiempo, son patrones utilizados constantemente en la vida cotidiana de forma gráfica. Entre estos encontramos los siguientes patrones: tendencias, estacionales, cíclicas, estacionarios e irregulares. Por otro lado, encontramos la descomposición de series temporales, que no es más que un proceso de identificación y cálculo de los diversos componentes existentes en una serie de tiempo y la forma en que estas se relacionan entre sí.

3.1.1 PATRONES DE SERIES TEMPORALES

El patrón de tendencia generalmente se conoce como un crecimiento o decrecimiento en las series temporales sobre un largo periodo de tiempo. Un supuesto sería la evolución del PIB

(Producto interior Bruto) de Alemania, el cuál evoluciona con una tendencia alcista constante tal y como se muestra en la Figura 2.

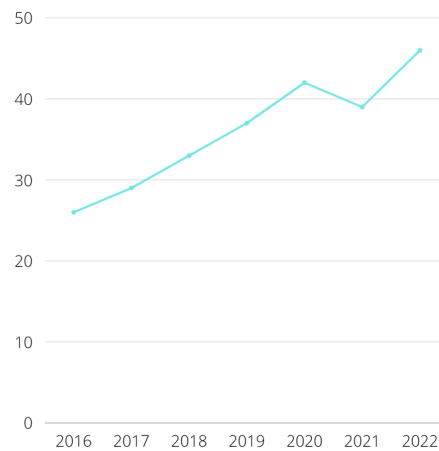


Figura 2. Aproximación gráfica del PIB de Alemania (miles de millones). Fuente:elaboración propia

Cuando una serie de tiempo experimenta cambios regulares y predecibles en períodos fijos, se dice que contiene un patrón estacional. Este es el caso del aumento en ventas de bañadores en verano o el aumento de artículos navideños en Diciembre, como por ejemplo se observa en la Figura 3.

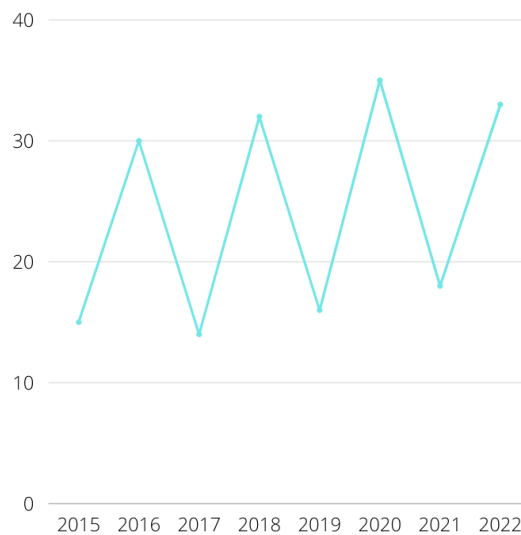


Figura 3. Ejemplo de patrón estacional, venta de bañadores en verano. Fuente:elaboración propia

En ocasiones las series temporales muestran otras oscilaciones predecibles, pero que no tienen un período fijo. Este tipo de variación es un patrón cíclico, en el cual los ciclos no se repiten en intervalos regulares y no tienen la misma forma. El ejemplo típico de un patrón

cíclico es un ciclo económico, en el que la economía experimenta periodos de crecimiento y periodos de recesión tal y como en la Figura 4.

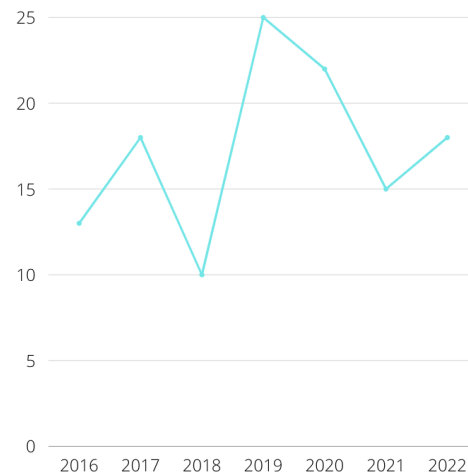


Figura 4. Patrón ciclo económico. Fuente:elaboración propia

Cuando los valores de una variable oscilan de manera constante a lo largo del tiempo , existe un patrón estacionario. Un supuesto bastante común dentro de este patrón, son las ventas de sal, las cuáles no suelen aumentar ni disminuir a lo largo de los años. Gráficamente es una línea prácticamente horizontal.

El patrón restante se conoce como patrón irregular o residuos, el cuál refleja cambios de una serie temporal en el corto plazo y, que debido a su aleatoriedad son complicados de modelar matemáticamente y por consecuencia no se proyectan a futuro para realizar pronósticos. Matemáticamente, está compuesto por: un patrón de tendencia, Tt , un patrón cíclico, Ct , un patrón estacional, St , y un recordatorio, Rt , que denota las variaciones estocásticas no capturadas por los otros componentes. Un supuesto son las variaciones observadas en la serie histórica de precios de la divisa EUR/TRY (Euro/Lira turca), la cuál se ha visto sometida a especulaciones debido a problemas sociales, acontecimientos políticos, etc. que influyen en las decisiones de los inversores (Figura 5).



Figura 5. Cotización divisa EUR/TRY. Fuente: Investing.com

3.1.2 DESCOMPOSICIÓN DE SERIES DE TIEMPO

En esencia, existen dos métodos para extraer la tendencia y el componente estacional de la serie temporal, el método paramétrico y no paramétrico:

Método Paramétrico

- Propone modelos paramétricos para representar la relación entre tendencias y componentes de estacionalidad a lo largo del tiempo.
- Ajusta estos modelos a series de tiempo, por ejemplo, con el método de mínimos cuadrados.
- Aísla la tendencia y el componente estacional por medio de los modelos ajustados.

Método no paramétrico

- Asume "suavidad" en la relación existente entre la tendencia y el componente estacional a lo largo del tiempo.
- Aísla la tendencia y el componente estacional con la suavización del gráfico de secuencia (implementado, por ejemplo, la técnica de medias móviles).

3.2 TIPOS DE ANOMALÍAS

En series de tiempo, existen una serie de anomalías más simples en las que el detector de anomalías puede analizar cada punto de datos sin tener en cuenta ningún otro punto de datos en el conjunto de datos de entrada. La anomalía puntual es una observación x o y que se desvía notablemente de X según algunos criterios predefinidos, donde $x \in X$ e $y \notin X$ [2]. Por ejemplo, la ausencia de un trabajador a la reunión del lunes por la mañana durante el horario laboral es una anomalía puntual, porque a diferencia de los demás empleados, este no se encuentra en la oficina. Es posible establecer límites para detectar automáticamente estas anomalías cuando se desvían del umbral especificado previamente. Un ejemplo gráfico de una anomalía puntual puede observarse en la Figura 6.

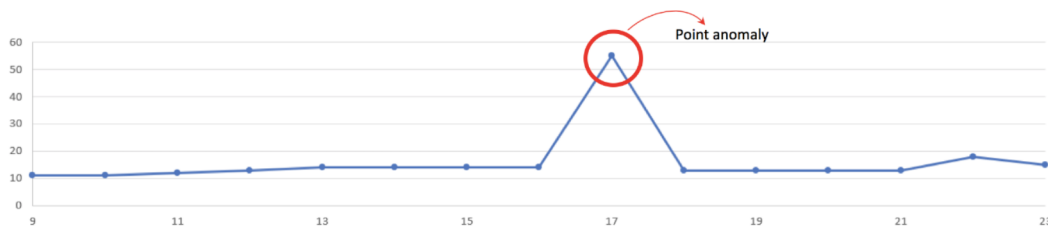


Figura 6. Temperatura por horas de un edificio. Fuente: [2]

Por otra parte, para detectar una anomalía contextual, el detector debe considerar el contexto en el que se produjo el punto de datos. Este colectivo de datos en el conjunto anómalo pueden ser normales, pero juntos representan una anomalía. En algunos casos, los algoritmos diseñados para la detección de anomalías individuales se pueden usar para las anomalías contextuales en caso de que se incluyan nuevas características, como el número de mes. Un supuesto en el que se puede observar una anomalía contextual es en el tráfico de automóviles a lo largo del día en una carretera, tal y como se muestra en la Figura 7:

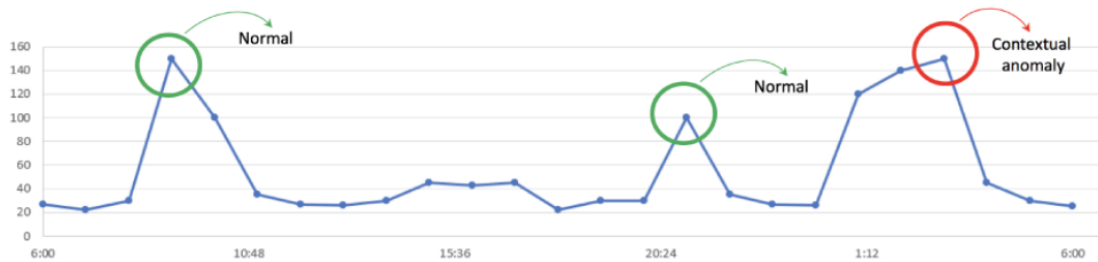


Figura 7. Número de coches en la carretera a diferentes horas del día. Fuente: [2]

Por último existen las anomalías colectivas. Se analiza una secuencia de observaciones para conocer el comportamiento selectivo del flujo de datos. Cualquier desviación del patrón normal puede dar lugar a anomalías colectivas con respecto a patrones de datos completos en intervalos de tiempo consecutivos. Por ejemplo, una sola observación de un intervalo de tiempo no es suficiente para determinar el comportamiento del corazón, mientras que las señales colectivas pueden determinar el comportamiento normal o anormal, como se muestra en la Figura 8.

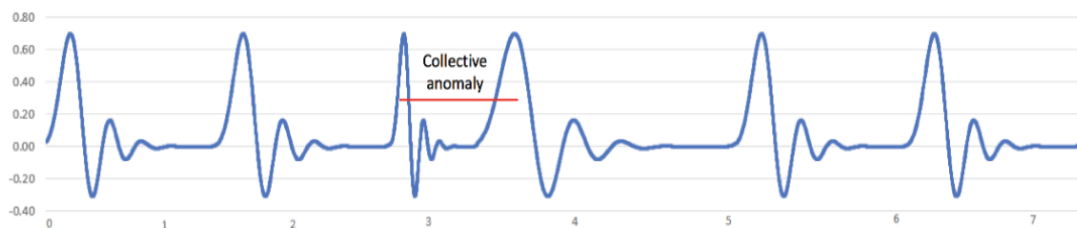


Figura 8. Frecuencia cardíaca con todos los patrones de datos durante intervalos de tiempo consecutivos. Fuente [2]

Para resumir, el tipo de anomalía más sencilla de detectar son las puntuales debido a que no es necesario encontrar una secuencia o contexto, además de no tener en cuenta ningún otro punto de datos en el conjunto de datos de entrada. Al detectar anomalías secuenciales, es común transformar subsecuencias de los datos en puntos que contienen información sobre la subsecuencia. Sin embargo, cómo construir subsecuencias es un problema en sí mismo y puede ser difícil de resolver. Por esta razón, en este trabajo se realizará la detección de anomalías puntuales.

3.3 TIPOS DE ENFOQUE EN LA DETECCIÓN DE ANOMALÍAS PARA SERIES TEMPORALES

En esta sección se introducen varios métodos de detección de anomalías de series de tiempo, los cuales se dividen en los siguientes enfoques: estadístico, *Machine Learning* clásico y *Deep Learning*. En la siguiente tabla (Figura 9) se puede observar una comparativa general de las características entre el enfoque estadístico y el enfoque de Machine Learning:

Datos	Estadística	Machine larning
Origen	Matemáticas	Inteligencia Artificial e Informática
Propósito	Inferencia sobre relaciones entre variables	Realizar predicciones
Uso de datos	- Datos con - Atributos	+ Datos , + Atributos, + Observaciones
Suposiciones en datos	+Suposiciones	No requiere
Esfuerzo humano	Alto	Bajo

*Figura 9. Tabla comparativa de las características del enfoque estadístico y el enfoque de Machine Learning.
Fuente:elaboración propia*

3.3.1 ENFOQUE ESTADÍSTICO

En los conjuntos de datos del mundo real, a menudo sucede que algunos puntos se comportan de manera diferente a la mayoría de los datos. Dichos puntos de datos se denominan anomalías en el aprendizaje automático o Machine Learning y valores atípicos en las estadísticas [12].

El objetivo principal de la estadística en series temporales es encontrar un ajuste que sea cercano al ajuste que se hubiera encontrado sin los valores atípicos. Luego podemos

identificar los valores atípicos por su gran 'desviación' (por ejemplo, su distancia o residual) de ese ajuste preseleccionado.

Como enfoques estadísticos, se han seleccionado algunos modelos regresivos comúnmente investigados como AR, MA, ARMA, ARIMA y algunos de los modelos que han funcionado correctamente en las competencias Makridakis² (también conocidas como M-Competitions). Aunque M-Competitions compara métodos de pronóstico estadístico, los métodos de detección de anomalías en series temporales están estrechamente relacionados con los enfoques de pronóstico. En este sentido, proporcionan una buena referencia para algoritmos estadísticos efectivos en el análisis de series de tiempo. A continuación, en la Figura 10, se presenta una tabla resumen con las principales ventajas y desventajas de este enfoque:

VENTAJAS	DESVENTAJAS
Método de trabajo sistemático	Gran inversión de tiempo y esfuerzo
Evita realizar afirmaciones sin base alguna	Se pueden llegar a tomar decisiones negativas que no ayuden a la mejora de los procesos.
Las afirmaciones se sostienen con datos objetivos	Aumenta el error humano

Figura 10. Tabla de ventajas y desventajas del enfoque estadístico. Fuente: elaboración propia

3.3.1.1 Modelo autorregresivo (AR)

Un modelo autorregresivo es una representación de un proceso estocástico. Es decir, la variable independiente y la variable dependiente son la misma con la disparidad que la variable dependiente estará en un momento de tiempo posterior a la variable independiente.

El modelo se conoce como AR(p), donde p (orden) es equivalente al número de períodos que se va a retroceder para conseguir el pronóstico de nuestra variable. En la vida real, se utiliza comúnmente para la proyección de ventas de una empresa o la proyección del PIB (Producto Interior Bruto) de un país.

² Son una serie de competencias abiertas organizadas por equipos dirigidos por el investigador de pronósticos Spyros Makridakis y cuyo objetivo es evaluar y comparar la precisión de diferentes métodos de pronóstico.

3.3.1.2 Modelo Média Móvil (MA)

El modelo de média móvil consiste en utilizar promedios de las observaciones pasadas de una serie temporal específica. Por cada período, el cálculo del promedio se actualiza añadiendo la nueva observación y eliminando la más antigua. Esta técnica resulta apropiada cuando la aleatoriedad de los datos es elevada y su correlación baja. Un supuesto en el que es muy popular su uso, como indicadores técnicos de tendencia, es en el trading, debido a que actúan como “suavizadores” de los precios tanto a corto, medio y largo plazo.

3.3.1.3 Modelo Autorregresivo Média Móvil (ARMA)

El modelo ARMA es una herramienta utilizada (a diferencia del modelo MA) para predecir futuros valores de una serie temporal. Se trata de un proceso estocástico³ y estacionario formado por el AR y el MA. En comparación con los modelos AR y MA puros, los modelos ARMA proporcionan el modelo lineal más eficaz de las series temporales estacionarias, ya que son capaces de modelar el proceso desconocido con el mínimo número de parámetros.

3.3.1.4 Modelo ARIMA

Uno de los principales problemas con los conjuntos de datos es el hecho de que pueden ser no estacionarios. El modelo ARIMA es una generalización del modelo ARMA, utilizado con el fin de proporcionar métodos más sofisticados para crear modelos de los componentes de tendencia y estacionales, además, disponen de la ventaja añadida de poder incluir variables predictoras en el modelo.

Supongamos que una compañía de ventas por catálogo quiere llevar a cabo un modelo de predicción, pues bien, esta empresa ha recopilado datos de las ventas trimestrales de prendas femeninas junto a otros datos que pueden explicar las variaciones en las ventas. Estos predictores pueden ser el número de catálogos enviados por redes sociales, el número de prendas que se ofertan en dichos catálogos, el número de empleados activos en ventas telefónicas o el capital invertido en publicidad. Con el modelo ARIMA se observa si existe

³ Proceso cuyo comportamiento no es determinista, en la medida en que el subsiguiente estado del sistema se determina tanto por las acciones predecibles del proceso como por elementos aleatorios.

alguna diferencia significativa en su capacidad de predicción en comparación a un modelo suavizado exponencial sin predictores.

3.3.1.5 Suavizado exponencial (SES)

El suavizado exponencial es un método de predicción de datos temporales cuando se caracterizan por una sola variable independiente. En términos sencillos, este enfoque permite pronosticar la demanda de un producto en un período dado, de manera que estima que la demanda será igual a la media de los consumos históricos en un período concreto, estableciendo una mayor ponderación a los valores cercanos en el tiempo. Las formas de suavizado exponencial amplían el análisis para modelar datos con tendencias y componentes estacionales. Además, tiene en cuenta el error de pronóstico actual en los posteriores pronósticos.

3.3.1.6 Suavizado doble exponencial (DES)

DES agrega soporte particularmente para tendencias en la serie de tiempo univariada. Combinado con las tendencias aditivas, se conoce convencionalmente como el *modelo de tendencia lineal de Holt*. El nombre se deriva del nombre del desarrollador del método Charles Holt. Este método ayuda a cambiar las tendencias con el tiempo de diferentes maneras, ya sea de forma aditiva o multiplicativa, contando si la tendencia es lineal o exponencial.

3.3.1.7 Suavizado triple exponencial (TES)

Es un método que agrega explícitamente soporte a la serie temporal univariante para la estacionalidad y permite que los patrones de nivel, tendencia y estacionalidad cambien con el tiempo, ya que es un método adaptativo.

3.3.2 ENFOQUE MACHINE LEARNING CLÁSICO

El machine learning es una rama de la Inteligencia Artificial (IA) y la informática en la que se trabaja con el uso de datos y algoritmos, y que tiene por objetivo mejorar y aprender de manera más precisa que los seres humanos.

Los algoritmos de aprendizaje automático (*machine learning*) detectan anomalías en conjuntos de datos de series temporales sin asumir un modelo generador⁴ específico y tienen como objetivo aprender y reconocer automáticamente patrones en grandes cantidades de datos. En este tipo de técnicas, a diferencia de los modelos estadísticos, no es necesario conocer el proceso subyacente de los datos para poder realizar predicciones de series temporales y detección de anomalías en series temporales. Por lo tanto, estos métodos están muy avanzados fuera del campo de la estadística [13].

A continuación, se presenta una tabla resumen de las ventajas y desventajas de este enfoque en el análisis automático de datos (Figura 11):

VENTAJAS	DESVENTAJAS
Automatización de procesos	Dificultad de acceso a datos actualizados y fiables
Reduce el error humano	Falta de profesionales cualificados
Potencia la creatividad	Su desarrollo es costoso
Agiliza la toma de decisiones	-
Menor esfuerzo humano	-

Figura 11. Tabla de ventajas y desventajas del enfoque de Machine Learning. Fuente: elaboración propia

Tras presentar brevemente los algoritmos de aprendizaje automático, vamos a explicar varios métodos de detección de anomalías univariados en esta sección:

⁴ En probabilidades y estadística, un modelo generador es aquel que genera valores aleatorios de un dato observable, típicamente dados algunos parámetros ocultos.

3.3.2.1 K means clustering

El clustering consiste en la agrupación automática de datos, que tiene como fin detectar cuál es el mejor agrupamiento. K means es el algoritmo de clustering más usado dada su escalabilidad con la cantidad de los datos, por lo que permite especificar el número de grupos (k) que queremos encontrar. Resulta recomendable normalizar previamente los datos porque estos grupos se forman a partir de distancias, es decir, cada dato se asigna al baricentro más cercano, formando de esta manera varios grupos. Normalizar quiere decir que los datos sean de escalas similares y, de esta manera, los datos con una escala mayor no dominan las distancias [11]. Las técnicas más utilizadas para normalizar los datos son: escalar cada dato de nuevo en el rango (0,1) o asumir que cada dato sigue una distribución normal para conseguir que los datos tengan media = 0 y varianza= 1.

3.5.2.2 Clustering basado de densidad

Este tipo de técnica distingue las distintas áreas que existen en un conjunto de puntos, de tal forma que etiqueta el área en función de si hay concentraciones de puntos o si hay puntos separados por áreas vacías o con escasos puntos. Los puntos que no forman parte de un cluster o grupo, se etiquetan como ruido⁵. Pongamos un ejemplo para comprenderlo mejor, suponga que somos un analista deportivo de la NBA (Liga de baloncesto profesional en Estados Unidos) y que tenemos a mano todos los datos de tiros que se han enceestado y que no se han enceestado, pues bien, procedemos a incorporar todos estos datos en nuestra herramienta de clustering basada en densidad y obtenemos una serie de patrones de cada jugador sobre sus tiros encecados y no encecados. Esta técnica nos permite analizar estos patrones y poder realizar estrategias de juego para los próximos partidos.

3.3.2.3 Algoritmo local outlier factor (LOF)

Es un método de detección de anomalías no supervisado que calcula la desviación de densidad local de un punto de datos dado con respecto a sus vecinos. Considera como anomalías todas las muestras que tienen una densidad sustancialmente más baja que sus vecinas.

⁵ Variabilidad inexplicable dentro de una muestra de datos.

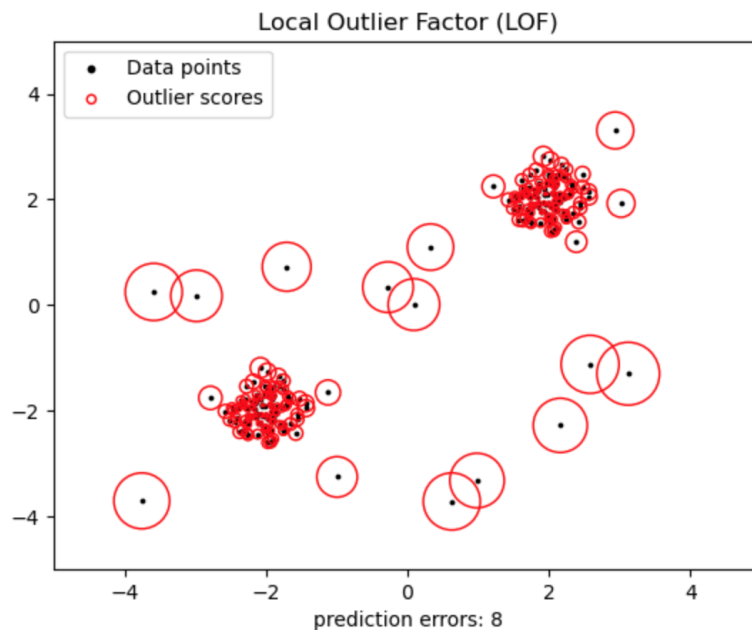


Figura 12. Representación del funcionamiento del algoritmo LOF. Fuente:[2]

3.3.2.4 Bosque de aislamiento

Bosque de aislamiento o Isolation Forest en inglés, es uno de los algoritmos más recientes en la industria del machine learning que se propuso en el año 2008. Se trata de un algoritmo basado en árboles que fue construido alrededor de la teoría de árboles de decisión y bosques aleatorios [14]. El funcionamiento comienza una vez se obtiene el conjunto de datos, ya que el algoritmo inmediatamente divide los datos en dos partes iguales o desiguales, el umbral escogido es aleatorio. Esta división de los datos se realiza continuamente (formando ramas de un árbol) hasta que el algoritmo se ejecuta a través de todos los datos, y consecuentemente el algoritmo filtra los puntos de datos que tomaron menos pasos que otros y los aísla como anomalías.

3.3.3 ENFOQUE MACHINE LEARNING - DEEP LEARNING

El aprendizaje automático trata que los ordenadores puedan realizar tareas sin estar programados explícitamente, pero los ordenadores aún piensan y actúan como máquinas. El Deep Learning o aprendizaje profundo, que es un subconjunto del aprendizaje automático,

llega un paso más lejos e introduce un enfoque extremadamente sofisticado para el aprendizaje automático con algoritmos inspirados en la estructura y función del cerebro llamados redes neuronales artificiales [15].

Las dendritas, el soma y el axón son los componentes que tiene una neurona humana: las dendritas se encargan de captar los impulsos nerviosos que emiten otras neuronas. Estos impulsos se procesan en el soma y se transmiten a través del axón, el cuál emite un impulso nervioso hacia las neuronas contiguas. En la siguiente Figura 13 se puede observar la representación de una neurona humana:

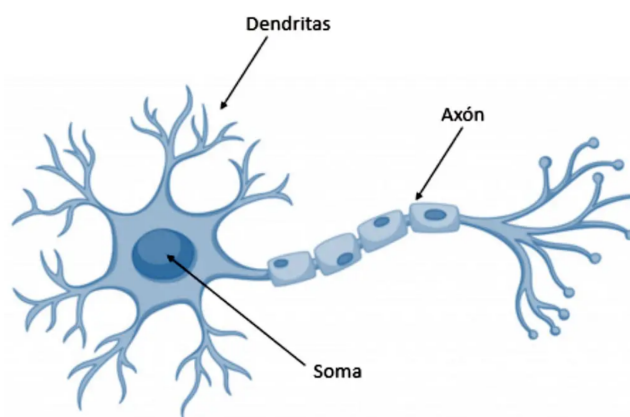


Figura 13. Representación de una neurona humana. Fuente:[16]

Las redes neuronales artificiales están basadas en este funcionamiento y a nivel esquemático se representa de la siguiente manera en la Figura 14:

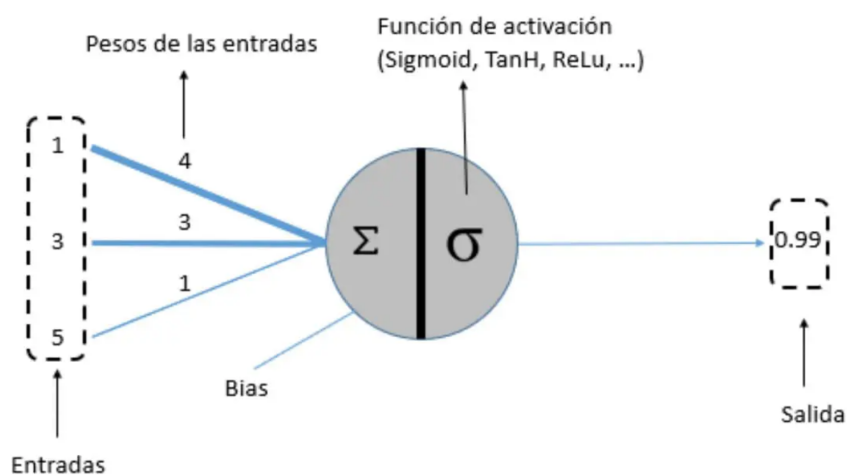


Figura 14. Representación del funcionamiento de una Red Neuronal Artificial. Fuente: :[16]

En el caso de las redes neuronales artificiales, la suma de las entradas multiplicadas por sus pesos asociados establece el “impulso nervioso” que recibe la neurona [16]. Este valor se procesa en el interior de la célula mediante una función de activación que produce un resultado con información como salida de la neurona.

De forma sencilla, puede decirse que nuestro cerebro está compuesto por neuronas interconectadas entre sí, mientras que las neuronas artificiales están conectadas y agrupadas entre sí en niveles que denominamos capas. Una capa es un conjunto de neuronas cuyas entradas se originan de los datos de entrada en caso de la primera capa y de una capa anterior para las demás, las salidas son la entrada de una capa posterior.

En la siguiente Figura 15 se puede observar una red con cuatro capas:

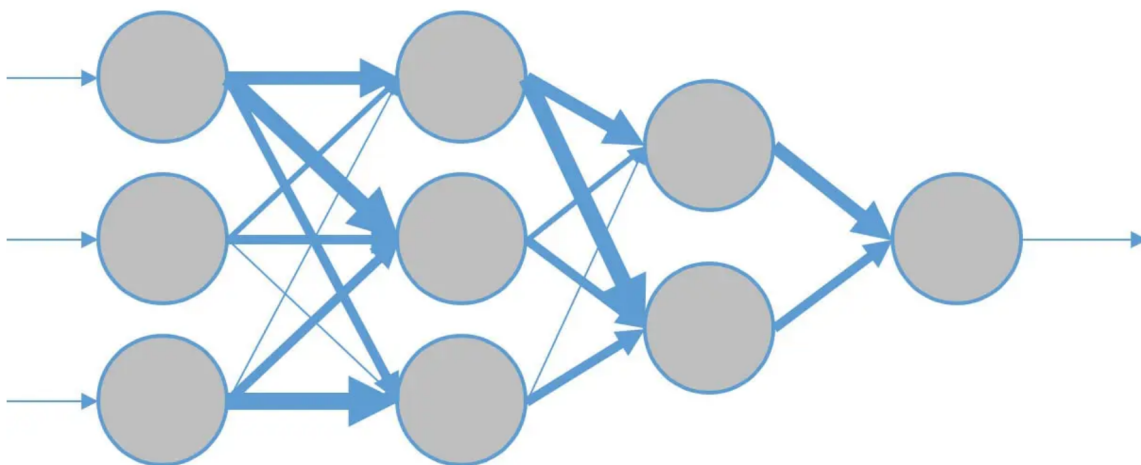


Figura 15. Representación Red Neuronal Artificial (perceptrón multicapa). Fuente :[16]

Esta figura muestra la red neuronal artificial más simple dentro del campo, reconocida como el perceptrón multicapa, el cuál está compuesto por múltiples capas: en la primera capa, las neuronas reciben como entrada los datos reales que alimentan la red neuronal, es por ello que recibe el nombre de capa de entrada. La última capa o capa de salida es el resultado obtenido de la red y la capa entre la capa de entrada y la capa de salida se le denomina capa oculta, debido a que se desconocen los datos de entrada y de salida.

Las redes neuronales no siempre resultan eficaces en primera instancia, es por ello que a las redes neuronales es necesario entrenarlas [17]. Entrenar una red neuronal consiste en ajustar cada uno de los pesos de las entradas de cada neurona, con el objetivo de que las respuestas de la capa de salida se aproximen lo máximo posible a los datos conocidos, o lo que es lo mismo, que tengan el menor error posible. De esta manera el modelo es capaz de identificar aquellos predictores con una mayor influencia y de qué manera están relacionados entre ellos y con la variable respuesta.

Las redes neuronales intentan representar la solución de problemas complejos no como una serie de pasos, sino como el desarrollo de sistemas informáticos inspirados en el funcionamiento del cerebro humano, y por tanto dotados de una cierta "inteligencia". Una combinación de una gran cantidad de neuronas interconectadas que funcionan completamente en paralelo para resolver problemas relacionados con la detección de formas o patrones, predicción, codificación, clasificación, control y optimización (Araceli Campos Ortiz, 1998). A pesar de que la idea parece sencilla, alcanzar una forma de implementarla ha requerido la combinación de múltiples métodos matemáticos, en concreto, el algoritmo de retropropagación (*backpropagation*) y la optimización por descenso de gradiente (*gradient descent*).

3.3.3.1 Backpropagation

Es un algoritmo capaz de cuantificar la influencia de cada peso y bias de la red en sus predicciones. Esto se consigue haciendo uso de la regla de la cadena (*chain rule*), que simplemente se trata del vector formado por las derivadas parciales de una función. En el caso de las redes, la derivada parcial del error respecto al peso o bias cuantifica la "carga" que han tenido en el error final [18]. De esta forma, se puede determinar qué pesos de la red hay que cambiar para mejorarla, posteriormente hay que establecer cuánto y cómo modificar los pesos (optimización).

3.3.3.2 Descenso de gradiente (gradient descent)

Es un algoritmo que permite minimizar el gradiente e ir actualizando los pesos y bias del modelo para reducir el error. Este proceso puede resultar muy costoso si se quiere calcular el error del modelo para todas las observaciones de entrenamiento, sin embargo, existe una

alternativa al descenso de gradiente llamado gradiente estocástico (*stochastic gradient descent, SGD*), el cuál divide el conjunto de entrenamiento en lotes (batch) y actualiza los pesos y bías con cada uno.

Una vez explicado qué son las redes neuronales, cómo funcionan y qué aspectos las complementan, se introducen a continuación una serie de las redes más trabajadas y estudiadas en el análisis con series de tiempo

3.3.3.3 Convolutional Neural Networks (CNN)

CNN es un tipo de red neuronal artificial que ha demostrado un rendimiento excelente en muchas aplicaciones, especialmente en tareas relacionadas con imágenes. Las aplicaciones de CNN incluyen clasificación de imágenes, segmentación semántica de imágenes, detección de objetos en imágenes, etc. Consiste en una red con aprendizaje supervisado que procesa sus capas imitando al córtex visual del ojo humano para identificar distintas características en las entradas [19], que en definitiva hacen que pueda identificar objetos y “ver”. Para conseguir este propósito, la CNN contiene varias capas ocultas especializadas y con una jerarquía: esto quiere decir que las primeras capas pueden detectar líneas, curvas y se van especializando hasta llegar a capas más profundas que reconocen formas complejas como un rostro o la silueta de un animal.

Por otro lado, existen redes neuronales convolucionales profundas con otra serie de aplicaciones como por ejemplo WaveNet. Se trata de una técnica creada en 2016 por la compañía de inteligencia artificial DeepMind. WaveNet ha resultado capaz de generar sonidos realistas como voces humanas modulando formas de onda mediante redes neuronales entrenadas directamente a partir de muestras del habla humana real [20].

3.3.3.4 Red Neuronal Residual

La red neuronal residual también conocida como Resnet fue propuesta por el equipo de Microsoft Research liderado por Kaiming He en su artículo *Deep Residual Learning for Image Recognition*, del año 2016. Esta red está basada en construcciones obtenidas de las células de la pirámide de la corteza cerebral. Se trata de la primera red neuronal de retroalimentación muy profunda que funciona con cientos de capas, las redes neuronales

artificiales consiguen las construcciones a través de atajos o “conexiones de salto” para moverse sobre varias capas (HighwayNets también puede aprender los pesos de salto por sí mismos a través de una matriz de peso adicional para sus puertas) [19]. Los modelos típicos de ResNet se implementan con saltos de doble o triple capa que contienen no linealidades (ReLU) y normalización por lotes en el medio.

Desde mi punto de vista, existen dos razones principales para agregar conexiones de salto:

- En primer lugar, para evitar el problema de desvanecimiento del gradiente, lo cuál permite que las redes neuronales sean más fáciles de optimizar y las funciones de activación faciliten el flujo de información a través de un gran número de capas (“autopistas de la información”).
- En segundo lugar, para mitigar el problema de degradación, en el que agregar más capas en un modelo profundo conduce a un mayor error de entrenamiento.

Las fuertes capacidades de representación de las redes neuronales residuales ha permitido que la clasificación de imágenes no haya sido la única aplicación de visión que ha utilizado Resnet, sino que también tiene una gran aplicación en el reconocimiento facial y la detección de objetos[19].

3.3.3.5 Red Neuronal Recurrente (RNN)

Las redes neuronales recurrentes fueron creadas en la década de los 80 y están basadas en el trabajo de David Rumelhart en 1986 [20]. Las redes neuronales recurrentes son redes neuronales con conexiones recurrentes entre ellas o conexiones con *feedback*.

Este tipo de redes neuronales son sistemas dinámicos, no tienen una estructura de capas definida sino que permiten conexiones aleatorias entre las neuronas, incluso pudiendo crear ciclos para que la red obtenga memoria a corto plazo. Las RNN son utilizadas para una amplia variedad de actividades informáticas, entre ellas la predicción y modelamiento de sistemas dinámicos, esto se debe a las conexiones recurrentes que se encuentran entre las redes de la red neuronal, usadas para reconocer o reproducir secuencias. Todo problema donde el tiempo o el contexto pueda ser relevante son candidatos a ser resueltos a través de RNN [21].

Con el fin de entender y poner en contexto la definición de este tipo de red, es necesario comparar previamente las redes neuronales recurrentes con las redes neuronales clásicas, mencionadas en la sección anterior:

Las Redes neuronales clásicas son simplemente aquellas cuya función de activación solo funciona en una dirección, hacia adelante, desde la capa de entrada hacia la de salida. Esto quiere decir que no recuerdan valores previos, todo lo que entra después sale y se olvida.

Una Red Neuronal Recurrente (RNN) lo más simple posible, compuesta por una sola neurona, funciona de la misma manera, solo que cuando produce una salida esta la envía de nuevo a sí misma. Esto implica que la RNN tiene memoria para recordar información pasada y capacidad de aprendizaje a la hora de solventar errores pasados. Este tipo de red ofrece muchas ventajas; es capaz de procesar entradas de cualquier longitud, la dimensión de las entradas no incrementa el tamaño del modelo, los cálculos consideran la información histórica y los pesos son compartidos a lo largo del tiempo [22]. El aspecto negativo, es que el hecho de que no tenga en cuenta la información futura para obtener las predicciones y que se complique el acceso a los datos obtenidos anteriormente en el procesamiento, crea grandes limitaciones a la hora de ofrecer resultados acertados en muchos problemas de procesamiento de lenguaje.

A lo largo de las últimas décadas, muchos trabajos se centraron en mejorar la arquitectura de las RNN, y actualmente son ampliamente utilizadas en el campo del procesamiento del lenguaje natural, *speech recognition* y modelado de series temporales, por ejemplo, para predicción del número de personas interesadas en un producto, o para hacer predicción de valores de acciones.

SOLUCIÓN AL PROBLEMA DE LAS DEPENDENCIAS A LARGO PLAZO

Como se ha comentado anteriormente, las Redes Neuronales Recurrentes presentan el problema fundamental de que no poseen memoria a largo plazo. A medida que aumentan las iteraciones temporales, las entradas que se obtienen en etapas tempranas en la red tienen poca o nula influencia en los resultados más tardíos. Este fenómeno se debe al conocido gradiente desvaneciente, el cuál provoca que el modelo presente complicaciones cuando ajusta los pesos lejanos durante el backpropagation ya que las derivadas empiezan a tender a cero [23].

Las Gated Recurrent Units (GRU) y su generalización, las Long Short-Term Memory units (LSTM) redujeron considerablemente el gradiente desvaneciente encontrado en las RNN clásicas, utilizando celdas de memoria. Un concepto de compuertas que le permiten al modelo decidir si conservar o no cierta información durante la propagación sucesiva.

REDES NEURONALES LSTM (LONG SHORT-TERM MEMORY)

Las Long Short-term memory (LSTM) son un tipo de redes neuronales recurrentes usadas en el campo de investigación del Deep Learning, que mediante conexiones de feedback le permite procesar grandes conjuntos de datos de información; son frecuentemente utilizadas para el reconocimiento, predicción y detección de anomalías.

La idea central de las LSTM gira en torno al uso de celdas de memoria, las cuales fueron creadas para ayudar a propagar la información de las capas tempranas a capas más profundas en la red. Las redes LSTM son memorias que llegan a decidir qué información del pasado debe ser borrada y qué información debe ser recordada.

REDES NEURONALES GRU (GATED RECURRENT UNITS)

Las Gated Recurrent Unit (GRU) son una versión simplificada de la celda LSTM, con la diferencia que, para solucionar el problema del gradiente desvaneciente, GRU utiliza una puerta de actualización y una puerta de reinicio. Básicamente, estas dos puertas deciden qué información se debe pasar a la salida al igual que las LSTM. Lo especial de esta red es que pueden ser entrenadas para mantener la información a largo plazo (Kostadinov, 2017).

AUTOENCODER

Un *Autoencoder* es un tipo de red neuronal artificial compuesto por un codificador y un decodificador, utilizado para tareas de aprendizaje no supervisado. El principal objetivo de este modelo es codificar y decodificar una entrada de datos. Es particularmente útil cuando se trata de representar los datos con un conjunto más pequeño de características y cuando se trata de reconstruirlos mientras se eliminan las características no deseadas (I. Goodfellow et al., 2016). Los *Autoencoder* es posible entrenarlos para que ofrezcan como resultado

versiones modificadas de los datos de entrada, esto se produce debido a que codifican los datos y luego los reconstruyen mediante el decodificador.

La detección de anomalías con *Autoencoders* es una estrategia no supervisada para identificar anomalías cuando los datos no están etiquetados, es decir, si estos son anomalías o no son anomalías. Un método para detectar anomalías es reducir la dimensionalidad de los datos y proyectarlos en un espacio de menor dimensión, es decir, el espacio latente, donde quedan más variables correlacionadas y una mayor información [24]. La manera en la que consiguen minimizar la pérdida global de información es buscando un nuevo espacio en el que la mayoría de observaciones puedan ser bien representadas. Los *Autoencoders* crean una función que mapea la posición que ocupa cada observación en el espacio original con el que ocupa en el nuevo espacio generado. Este mapeo funciona en ambas direcciones, permitiendo ir del espacio generado al espacio original [25].

Por lo tanto, la proyección hacia el espacio original mostrará diferencias significativas en algunos puntos de datos, que representan el conjunto de datos anómalos. Esto hace que el *Autoencoder* sea ideal para la detección de anomalías y, especialmente, en datos secuenciales como series de tiempo.

APLICACIONES AUTOENCODERS

La Red Neuronal Recurrente *Autoencoder* tiene muchas otras aplicaciones como, por ejemplo, la extracción de funciones. Russ Felker, director del departamento tecnológico de una compañía de servicios de logística y gestión de carga (Globaltranz), comentó que el uso de *Autoencoders* como un extractor de características elimina la necesidad de pasar horas de laboriosa ingeniería después de la limpieza de datos. Esto puede permitir que la clasificación de datos se complete más fácilmente. "Al reunir elementos similares, está permitiendo que el sistema pueda realizar rápidas recomendaciones sobre cuál debería ser el resultado", dijo Felker [26].

Los investigadores también están comenzando a explorar formas en que los *Autoencoders* puedan usarse para mejorar las relaciones de compresión de video e imágenes [27]. Se están desarrollando *Autoencoders* que pueden comprimir imágenes tomadas con una resolución

muy alta en una cuarta parte o menos del tamaño requerido con las técnicas estadísticas tradicionales.

Los codificadores automáticos han sido utilizados por empresas como PayPal para crear sistemas de detección de fraude mediante la extracción de características clave que determinan si una transacción es fraudulenta. Por otro lado, aquellos que se usan para la detección de anomalías son de gran utilidad en la industria bancaria para ayudar a automatizar la generación de algoritmos de recomendación de préstamos. Por ejemplo, si un banco tiene una gran cantidad de datos sobre sus clientes y préstamos, entonces estos datos se pueden usar para caracterizar cómo son los mejores préstamos.

Tras haber analizado enfoques como el estadístico, el *Machine Learning* clásico y el *Deep Learning* y dentro de ellos numerosos modelos con diferentes aplicaciones, la técnica que mejor se adapta al estudio del caso práctico de este trabajo es la Red Neuronal Recurrente *Autoencoder*. En este caso práctico se quiere detectar y analizar las diferentes anomalías en el precio de las acciones de las compañías Pfizer y Moderna, dos empresas farmacéuticas muy involucradas en la crisis del Covid-19 debido a la influencia obtenida por las vacunas que suministraron alrededor del todo el planeta. Para ello, se ha creado un modelo con redes neuronales recurrentes *Autoencoder* en Google Collab que ha permitido detectar y analizar las causas de estas anomalías para establecer conclusiones.

4 METODOLOGÍA

En este capítulo, primeramente se presenta el enunciado del problema que pretendemos estudiar. Seguidamente, se explica la metodología llevada a cabo de manera conceptual, con el propósito de facilitar la comprensión de cómo hemos desarrollado el modelo *Autoencoder LSTM* para la detección de las anomalías en series temporales. Además, describimos las variables empleadas y su forma de medición, la aplicación del estudio y el análisis de los datos (programa, fiabilidad, rapidez).

4.1 ENUNCIADO DEL PROBLEMA

En este estudio, como se ha mencionado a lo largo de los diferentes capítulos, se quiere detectar y analizar las anomalías existentes en el precio de las acciones de dos compañías: Pfizer y Moderna. Como consecuencia de la crisis sanitaria del Covid-19, las farmacéuticas pasaron a colocarse en el primer plano de la actualidad debido al importante papel que desempeñaron en la gestión de la pandemia. La investigación de estas anomalías permite obtener información que nos ayude a comprender por qué han sido causadas. Así mismo, el precio de las cotizaciones se representa por medio de datos de series temporales, por lo que la implementación de un modelo de redes neuronales recurrentes es necesario para su análisis.

Por otra parte, el entorno macroeconómico global actualmente se está viendo perjudicado y vulnerable a las diferentes crisis, guerras y otras numerosas razones que no son estudio de la investigación, pero es necesario mencionar lo probable que es que la cotización de Pfizer y Moderna se vea relativamente arrastrada por el ambiente económico global. Por ejemplo, el *MSCI World* es uno de los índices bursátiles de referencia a nivel global que representa la economía y mercados financieros del mundo desarrollado, y la cotización del índice ha descendido un 14,28% a día 25 de mayo de 2022 desde su máximo histórico a finales del año 2021. Ambas compañías se encuentran en países desarrollados ya que Pfizer y Moderna son empresas estadounidenses. Esta información es importante para que el modelo de Redes Neuronales Recurrentes no distorsione la realidad objetiva de los datos para la detección de las anomalías.

4.2 INSTALACIÓN LIBRERÍAS

En este trabajo, crearemos una red neuronal recurrente *Autoencoder LSTM* en Google Colab a través de Python con Keras y Tensorflow. Python es un lenguaje de programación que se utiliza para desarrollar aplicaciones de todo tipo, a diferencia de otros lenguajes como Java o .NET, se trata de un lenguaje interpretado, lo que significa que no necesita ser compilado y traducido para ejecutar aplicaciones escritas en Python. Se ejecuta directamente en la computadora mediante un programa llamado intérprete, por lo que no es necesario "traducirlo" a lenguaje de máquina [28]. Python ha ido ganando apoyo gracias a su sencillez y a sus amplias posibilidades, sobre todo en los últimos años, ya que facilita trabajar con inteligencia artificial, big data, machine learning y data science, entre muchos otros campos en auge.

Por su parte, Keras es una biblioteca de redes neuronales de código abierto escrita en Python que está diseñada para ir construyendo por bloques la arquitectura de cada red neuronal, tanto redes convolucionales como recurrentes, que son aquellas que permiten entrenar modelos de *Deep learning*. Es un sistema capaz de ejecutarse sobre Tensorflow, Microsoft cognitive, Toolkit o Theano. Sin embargo, el mejor sistema para usar con Keras es Tensorflow, debido a que gestiona múltiples *GPUs* (unidades de procesamiento gráfico) en el entrenamiento de datos si la arquitectura lo permite. Tensorflow es también una biblioteca de código abierto para *Deep learning* que reúne una serie de modelos y algoritmos para combinarlos y hacerlos útiles, su uso es adecuado para clasificar, descubrir predicciones, identificar patrones y aplicar percepciones y creación. Se ha utilizado en aplicaciones como el cuidado de la salud, productos propios de Google, los medios sociales, los anuncios o la detección de anomalías.

Las librerías de Keras y Tensorflow permiten generar, entre otros modelos de machine learning, redes neuronales con arquitectura de *Autoencoder LSTM*, por lo que el primer paso se trata de instalar estas dos librerías en Python para generar nuestro modelo.

4.3 IMPLEMENTACIÓN Y DESARROLLO DE LA BASE DE DATOS

Una vez ya se han instalado las librerías para comenzar a generar nuestro modelo, se han obtenido los precios de cotización diarios de Pfizer y Moderna desde la página web de Yahoo Finance. Se han descargado los precios de las acciones de los últimos cinco años (20/05/2017 - 20/05/2022) en caso de la compañía Pfizer, y aproximadamente tres años y medio de la cotización bursátil de Moderna (07/12/2018 - 20/05/2022). Estos valores se han incorporado al modelo como datos de entrada en un archivo CSV, que es un tipo de documento simple de formato abierto que presenta datos como una tabla con columnas separadas por comas y filas separadas por saltos de línea. Este archivo creado en Python, que se ha desarrollado para la realización del estudio, se encuentra en el siguiente link como código abierto para el lector: <https://github.com/ignacio-300/Deteccion-anomalias.git>

La detección de anomalías para series temporales consta de dos problemas: encontrar aquellos datos anómalos y etiquetar correctamente la serie objetivo. El trabajo tiene como objetivo encontrar todos los valores atípicos, etiquetarlos y posteriormente analizar cuáles han sido las causas de estas anomalías en el precio de las acciones.

Se incorporan los valores de cotización de ambas compañías y se definen las diferentes variables de los datos continuos en la Figura x que se observa a continuación, la cuál corresponde a una muestra del conjunto de datos de entrada para explicar las variables que los componen:

	Open	High	Low	Close	Adj Close	Volume
Date						
2017-05-22	30.834915	30.834915	30.474382	30.474382	25.340321	26725962
2017-05-23	30.474382	30.626186	30.436434	30.493359	25.356104	18517410
2017-05-24	30.521822	30.626186	30.332067	30.407970	25.285099	23576821
2017-05-25	30.417458	30.550285	30.332067	30.512335	25.371878	18227244
2017-05-26	30.531309	30.635674	30.455408	30.493359	25.356104	12330746

Figura 16. Tabla variables del modelo. Fuente:elaboración propia

Variables

- *Open* es el precio al que cotiza la acción al abrir el mercado.
- *High* el precio más alto que se alcanzó en una sesión bursátil.
- *Low* el precio más bajo que se alcanzó en una sesión bursátil.
- *Close* el precio al que cerró la acción en dicha sesión bursátil.
- *Adj Close* cambia el precio de cierre de una acción para reflejar el valor de esa acción después de contabilizar las acciones corporativas
- *Volume* corresponde al número de transacciones que se han realizado en una sesión bursátil.

4.3.2 SELECCIÓN DE VENTANAS DE ENTRENAMIENTO - PRUEBA

Una vez incorporados todos los datos de entrada, necesitamos agrupar los datos en secuencias más pequeñas para nuestro modelo, pero primero, los dividiremos en dos grupos: los datos de entrenamiento y los datos de test. Se selecciona el tanto por ciento de dichos datos que queremos entrenar (*Train size*), y el tanto por ciento restante serán los datos que se utilicen para probar el test (*Test size*). Teniendo en cuenta la afluencia de vacunas debido al COVID 19 a partir del año 2021, se ha asumido, solamente con el fin de ejemplificar el proceso, una ventana a partir de este año que podría caracterizar datos anómalos. Por ello se ha definido una ventana del 2017 al 2021 para el conjunto de entrenamiento y una del 2021 en adelante como ventana de prueba (Test)

Queremos hacer clasificación usando un algoritmo supervisado, con lo cual tendremos:

PFIZER

X_train con 1.008 registros para entrenar

y_train con las “etiquetas” de los resultados esperados de X_train

X_test con 252 registros para test

y_test con las “etiquetas” de los resultados de X_test

MODERNA

X_train con 695 registros para entrenar

y_train con las “etiquetas” de los resultados esperados de X_train

X_test con 174 registros para test

y_test con las “etiquetas” de los resultados de X_test

La muestra de datos de entrenamiento y de test que incorporamos en el modelo observar para cada una de las empresas se pueden gráficamente en la Figura 19 y Figura 20.

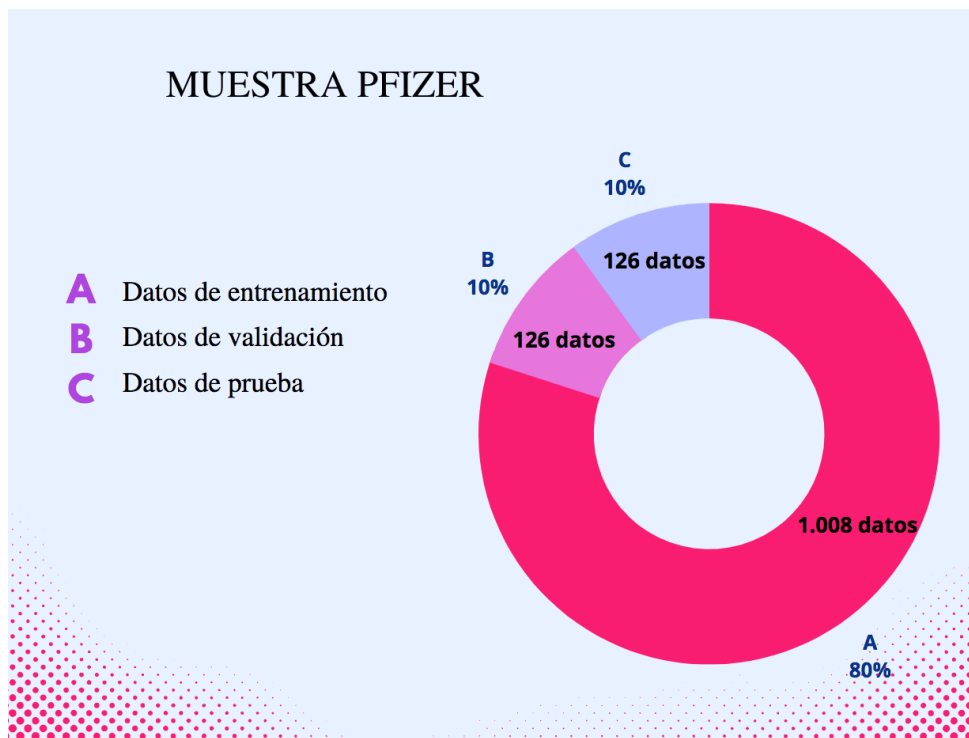


Figura 19. Gráfico de la muestra de datos de Pfizer para el modelo. Fuente: elaboración propia

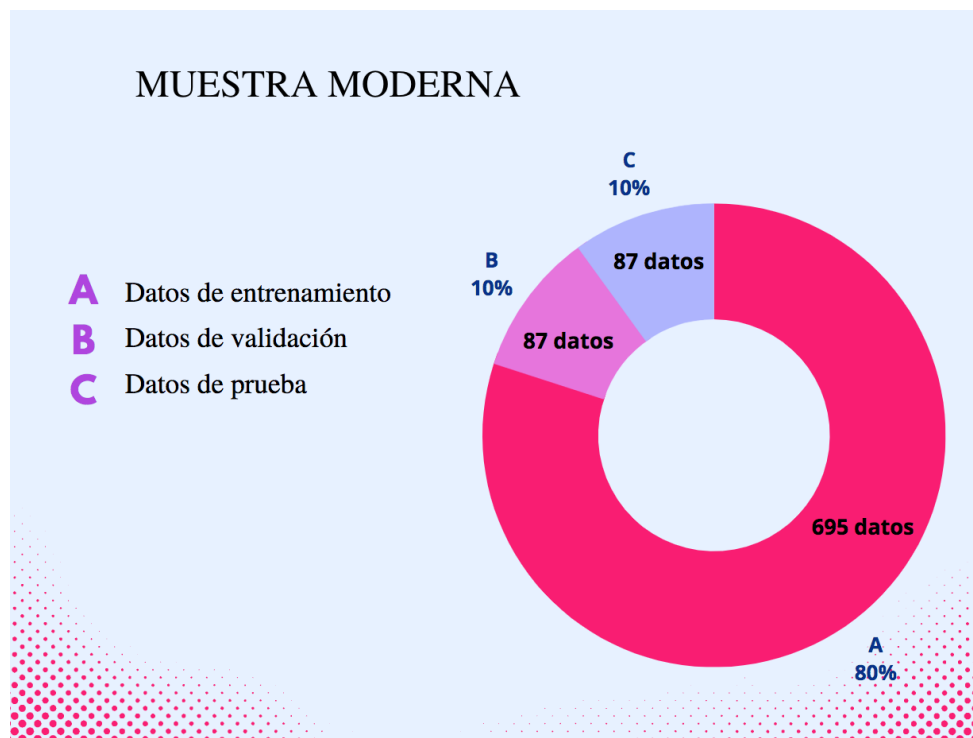


Figura 20. Gráfico de la muestra de datos de Moderna para el modelo. Fuente: elaboración propia

El set de validación no lo consideramos como otro grupo debido a que pertenece al conjunto de entrenamiento, este se utilizará durante iteraciones que haremos con el set de entrenamiento. En el gráfico superior hemos dividido los datos del conjunto de entrenamiento en datos de validación y test ya que al final se están usando los datos para las iteraciones, tal y como se muestra en la siguiente Figura 21:

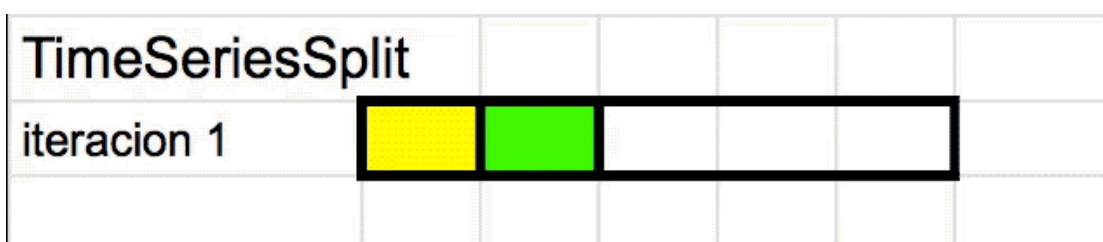


Figura 21. En amarillo las muestras para entrenar y en verde el conjunto de Validación. Fuente: Aprende Machine Learning [29]

4.3.3 ESCALAMIENTO DE LOS DATOS Y GENERACIÓN DE LOS CONJUNTOS DE DATOS

El ajuste de los pesos de la red neuronal se facilita cuando los datos de entrada están normalizados en un rango de (0,1) o de (-1,1). Es importante recordar que el escalador debe entrenarse con los datos de entrenamiento para evitar sesgos, y que deberá aplicarse de igual forma tanto a los datos de entrenamiento como a los de prueba (test).

Con el fin de realizar la detección de los puntos anómalos dentro de la serie de tiempo, se entrenará en primer lugar un algoritmo de predicción. Este modelo, entrenado con los datos que consideramos sin anomalías, permitirá predecir en la ventana de test un comportamiento “sin anomalías” de la serie de tiempo. Este algoritmo predictor será la red neuronal recurrente LSTM, al tener una configuración que permite modelar las dependencias temporales de los datos. Preparar los datos para el pronóstico de series temporales (en particular, LSTM) puede ser complicado. Intuitivamente, si necesitamos predecir el valor en el paso de tiempo actual, se usará el historial de los datos (n pasos de tiempo anterior al valor actual). Dado que la forma de aprendizaje de esta red requiere el procesamiento con muchas muestras (series de tiempo), el conjunto de entrenamiento y validación se generará con sub series de tiempo obtenidas a través de una ventana de n muestras que se desplaza respectivamente por el conjunto de entrenamiento y de validación. A modo ilustrativo, a continuación, se muestra el proceso de creación de los conjuntos de entrenamiento y validación (Figura 22):

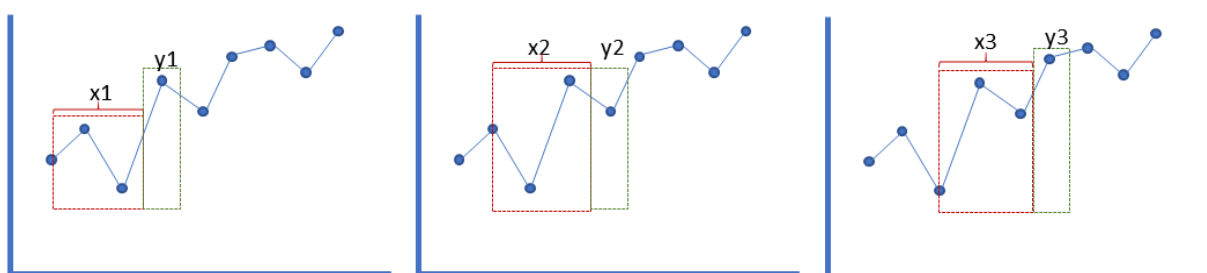


Figura 22. Gráfico del proceso realizado con el conjunto de datos. Fuente: elaboración propia

SeriesLength es el período de la serie de tiempo donde separamos los datos para su análisis en ventanas de tiempo. Para este caso de estudio, se seleccionó una ventana de 64 pasos de tiempo.

Estructura de la red neuronal: la entrada de la reunión neuronal se compone de 64 valores de la serie de tiempo, por 32 observaciones, donde la serie de tiempo es el tamaño de la

ventana que usamos para crear el conjunto de entrenamiento y las observaciones es el número de sub-series de tiempo que se generaron.

Una vez procesada esta información en el modelo, ya tenemos nuestras secuencias de datos creadas. Ahora, debemos saber usarlas para realizar las predicciones.

4.3.4 MODELO AUTOENCODER LSTM

Es la hora de poner a entrenar la red neuronal *Autoencoder LSTM* y gracias a Keras es bastante sencillo. Usaremos la capa *LSTM* en un modelo secuencial para poder realizar las predicciones y reducir el problema del gradiente desvaneciente, lo que haremos es modificar nuestro flujo de entrada del archivo CSV, que contiene una columna de unidades de salida (neuronas de salida), y lo convertiremos en varias columnas. De hecho, lo que vamos a hacer es tomar nuestras series de tiempo y convertirlas en "tareas de tipo supervisado" para que podamos alimentar nuestra red neuronal y entrenarla a través del Backpropagation. El modelo lo hemos compuesto de secuencias con 32 neuronas de entrada para entrenar la red (el valor se adapta a las necesidades del modelo), y por cada secuencia que se ha entrenado obtenemos un valor de salida.

4.3.4.1 ENTRENAMIENTO

En este paso del caso, debemos entrenar la red neuronal con el objetivo de que las respuestas de la capa de salida se aproximen a los datos conocidos. Lo más importante que hay que recordar mientras se entrenan modelos de series temporales es no entremezclar los datos de entrenamiento, ya que el orden de estos es importante. En Python, creamos un código compuesto por una serie de funciones que permiten que el modelo se vaya entrenando, para ello se dividen los datos en "lotes" de tamaño *batch_size* se especifica el número de épocas o pases completos (*epochs*) que todo el conjunto de datos de entrenamiento debe procesarse para el aprendizaje del algoritmo. Gracias a esto, el modelo es capaz de identificar qué predictores tienen mayor influencia y de qué forma están relacionados entre ellos y con la variable respuesta. Pero estos valores no se escogen aleatoriamente, en este caso se ha seleccionado un *batch_size = 16* y un *epochs_size = 50*, principalmente debido a que el modelo aprende mejor con estos valores que con otros con los que se realizó la etapa de validación.

Ponemos en marcha el entrenamiento de la red neuronal y automáticamente ajusta los pesos de las entradas de cada neurona, para que estas tengan el menor error posible

4.3.4.2 THRESHOLD

Una vez que se ha entrenado el algoritmo predictor de la serie de tiempo. Se procederá a hacer la detección de los datos anómalos. Debido a que el predictor fue entrenado con datos que consideramos “normales”, la idea fundamental de este modelo es que la predicción tendrá errores de reconstrucción relativamente pequeños para los datos normales, pero grandes errores de reconstrucción para los datos anormales. Para ello, se debe definir un umbral que permita determinar si el punto analizado de la serie se encuentra dentro de un comportamiento “normal” de acuerdo a la dinámica histórica de la serie o no [30]. Para saber cual es el valor más adecuado del *Threshold*, se desarrollan los siguiente pasos:

- Encontramos el error entre los valores predichos por la red \hat{y} y los valores reales de la serie y . Para ello, hallamos el error absoluto medio MAE (*MAE loss*) definido por la siguiente fórmula:

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$

donde n es el número de muestras de la serie de tiempo analizado

- Se analiza la distribución del error, y se selecciona el valor de error que engloba aproximadamente el 98 % de los datos de error. Los valores usados en la literatura varían del 95% al 99%. En el caso de Pfizer se ha seleccionado un $Threshold = 3,15$ y en Moderna un $Threshold = 1$
- Si la pérdida de reconstrucción de una muestra es mayor que este valor de umbral, podemos afirmar que el modelo está viendo un patrón con el que no está familiarizado y etiquetamos esta muestra como una anomalía.

Con la explicación de la metodología para desarrollar el modelo Autoencoder LSTM entendemos que tipo de datos estamos usando (cuantitativos continuos), como estos se implementan en el modelo en bloques, dividiendo entre datos de entrenamiento y datos de test y de qué manera (escalado). También hemos comprendido el proceso por el que hay que someter la muestra de datos en la red neuronal que hemos creado y el umbral adecuado en función del valor máximo de pérdida de MAE. A partir de este paso, la red neuronal *Autoencoder LSTM* facilita las anomalías detectadas.

5 RESULTADOS CASO DE ESTUDIO

En este capítulo se ofrece una descripción general de los experimentos y se presentan los datos obtenidos en un orden lógico, utilizando tablas y gráficos. Recordemos el propósito de este trabajo, detectar y analizar las anomalías en el precio de las acciones de Pfizer y Moderna a través de una red neuronal recurrente *Autoencoder LSTM*. En esta sección, se observará el resultado del caso de estudio seleccionado para ejemplificar el procedimiento de detección automática de anomalías.

5.1 ADQUISICIÓN Y PROCESAMIENTO DE LOS DATOS

Se han descargado los precios de las acciones de los últimos cinco años (20/05/2017 - 20/05/2022) en caso de la compañía Pfizer, y aproximadamente tres años y medio de la cotización bursátil de Moderna (07/12/2018 - 20/05/2022) desde la página web oficial de Yahoo Finance. Estos valores se han procesado como datos numéricos continuos de entrada en Python, con ellos se han elaborado gráficas de tiempo de ambas compañías con la variable *close*. Además, esta serie de tiempo se ha dividido en datos de entrenamiento y validación o prueba, tal y como mencionamos en el anterior capítulo.

COTIZACIÓN HISTÓRICA PFIZER & MODERNA

Ahora que sabemos cuales son las variables que componen los datos, vamos a realizar una demostración gráfica de la cotización histórica de ambas compañías usando la variable *Close*, con el fin de observar la serie temporal con la que trabajaremos:

PFIZER

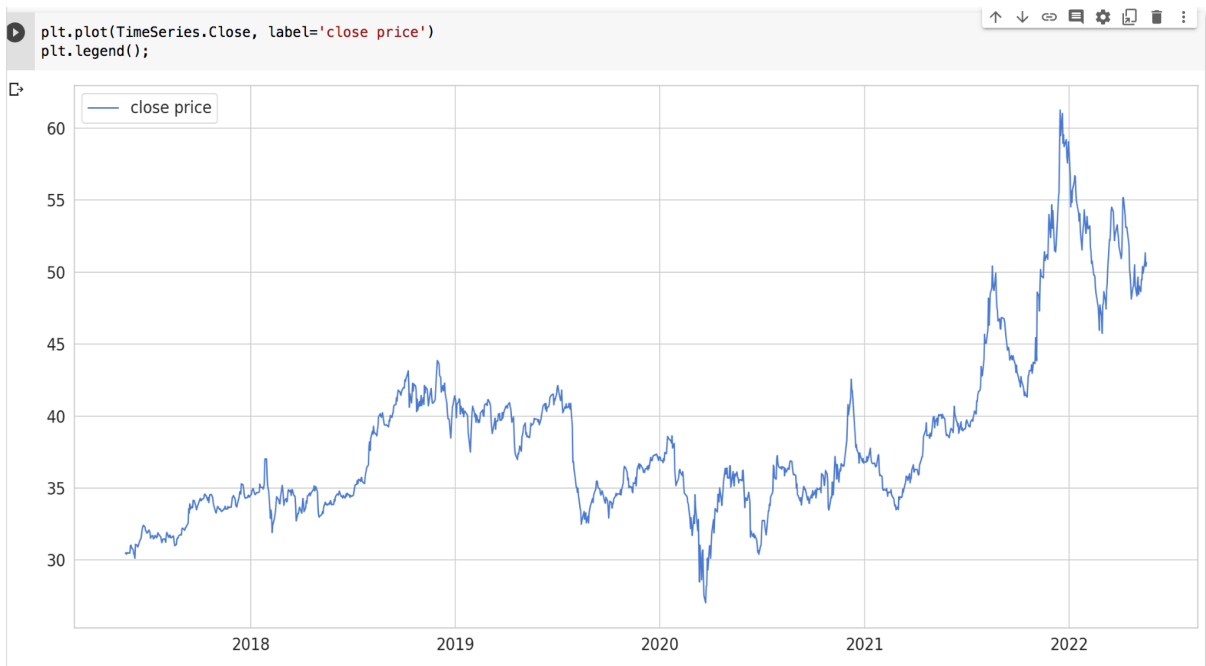


Figura 23. Gráfico lineal de la cotización bursátil de Pfizer. Fuente:elaboración propia

En la Figura 23 observamos la serie temporal entre 2017 y 2022 de la cotización bursátil de Pfizer. Se ha creado a partir de los datos *Close price*, es decir, de los precios de cierre de cada sesión bursátil.

MODERNA



Figura 24. Gráfico lineal de la cotización bursátil de Moderna. Fuente:elaboración propia

En la Figura 24 contemplamos la serie temporal entre 2018 y 2022 con datos *Close* de Moderna.

La figura 25 es un ejemplo gráfico de la clasificación de los datos de entrenamiento, validación y test para la serie de tiempo de Pfizer y Moderna. La línea gris es un ejemplo de la predicción del modelo en el precio de la acción una vez se ha realizado todo el proceso y entrenamiento de los datos.



Figura 25. Gráfico del proceso realizado con el conjunto de datos. Fuente: elaboración propia

5.2 ENTRENAMIENTO DEL MODELO

A partir de las consideraciones descritas en el capítulo anterior, se entrenó la red neuronal. En la siguiente gráfica, puede observarse el error de entrenamiento y validación, en cada época de análisis. Como puede observarse, el error va decreciendo a medida que analiza más datos y sus valores finales de entrenamiento-validación, son relativamente cercanos, evitando

problemas de sobreajuste. Si estas líneas paralelas comienzan a alejarse constantemente, podría ser una señal para detener el entrenamiento en una época anterior, o lo que se conoce como early-stopping. Se puede observar en la Figura 26 y Figura 27.

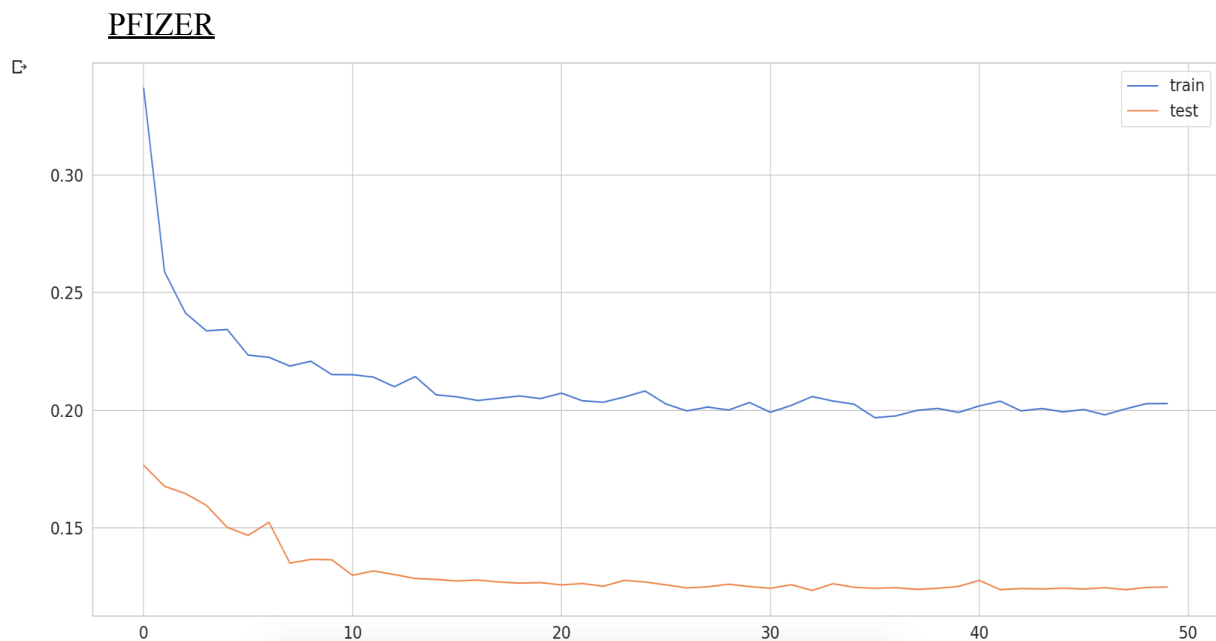


Figura 26. Gráfica error entrenamiento/validación Moderna. Fuente:elaboración propia.

Como observamos en este gráfico, el modelo ha llegado a un valor estacionario del error una vez el conjunto de datos ha sido procesado 20 veces o *epochs* por el proceso de entrenamiento del algoritmo, debido a que la distancia entre estos a partir de las 20 *epochs* ya no decrece. La diferencia de distancia entre la línea de entrenamiento y la línea del test es normal, ya que asumimos un pequeño error de predicción del algoritmo.

MODERNA

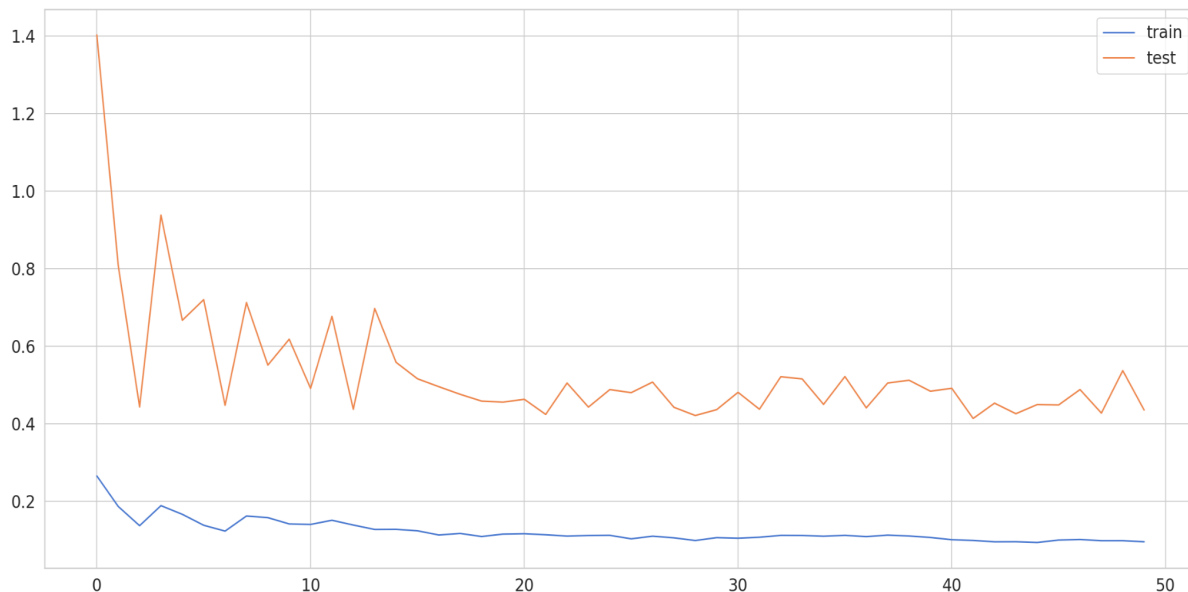


Figura 27. Gráfica error entrenamiento/validación Moderna. Fuente: elaboración propia.

En el caso de Moderna, la línea naranja se encuentra por encima de la línea azul a diferencia del caso de Pfizer. Esto implica que el modelo está aprendiendo dada su constancia en el movimiento horizontal a lo largo de cada época. En este caso aunque en los dos modelos decrece el error, puede considerarse un ligero sobreajuste de los datos.

5.3 SELECCIÓN DEL UMBRAL DE DETECCIÓN

El histograma que se puede observar a continuación (Figura 28) muestra la distribución del error en la predicción de los datos de entrenamiento:

PFIZER

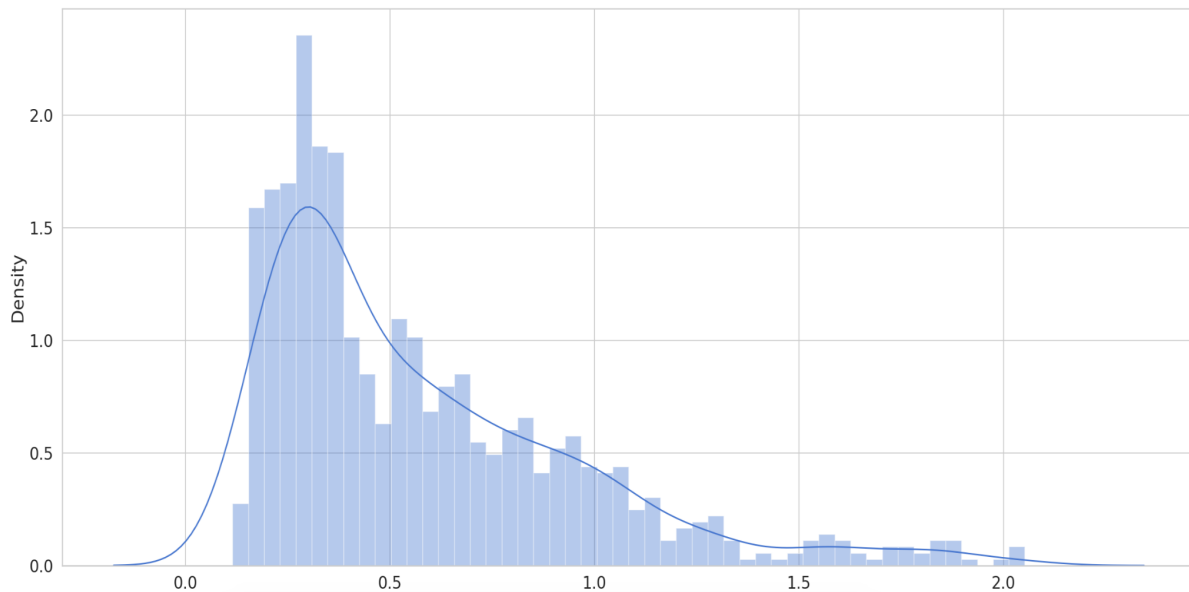


Figura 28. Distribución del error en Pfizer. Fuente: elaboración propia.

Esta distribución del error con sesgo a la derecha muestra que la mayor parte de los datos del entrenamiento se sitúan entre los valores $[0,2]$ del error, y claramente la densidad de los datos es mayor cuando se acerca a cero, lo que quiere decir que la mayoría de estos datos tienen un error muy bajo en la predicción. Aquellos datos que se alejan de cero por el eje horizontal están “peor” predichos por el modelo. Esta información es de gran utilidad porque nos permite establecer un umbral para detectar las anomalías, es por ello que en el capítulo anterior se ha seleccionado un *threshold* de 3,15, por ser el valor máximo para el cual se acumula aproximadamente el 98 % de la distribución. Esto quiere decir que cualquier dato que se aleje expresamente de 3,15 una vez se están ejecutando los datos de prueba se tomará como un valor atípico [31].

MODERNA

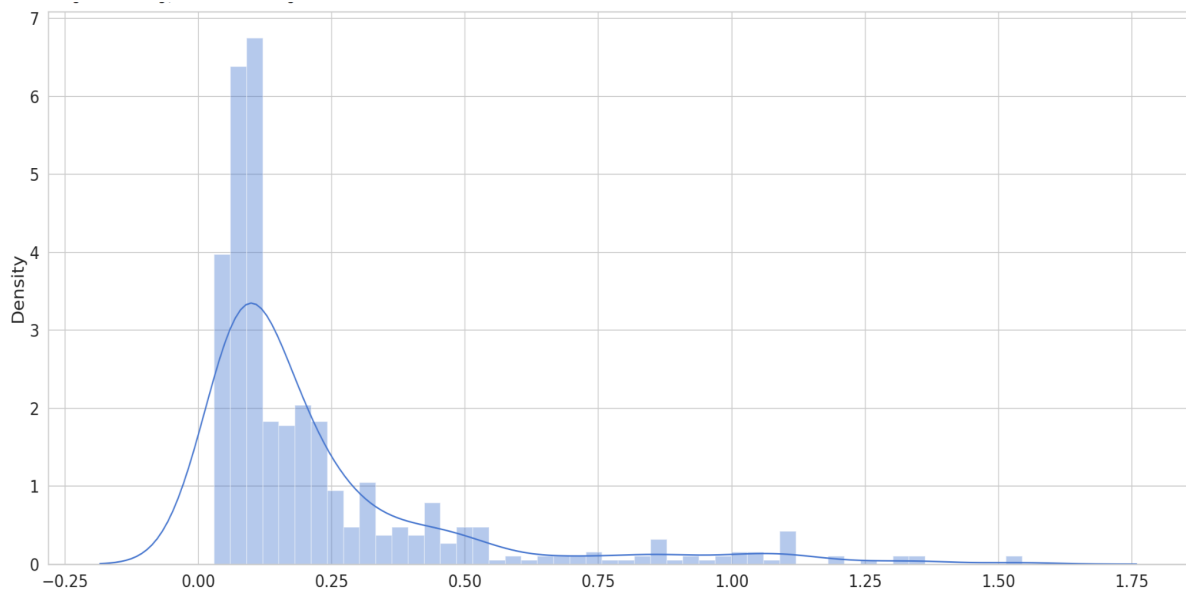


Figura 29. Distribución del error en Pfizer. Fuente:elaboración propia

La distribución del error en el caso de Moderna (Figura 29) está sesgada a la derecha y muestra un comportamiento similar al caso de Pfizer. Sin embargo, a diferencia de Pfizer, la predicción de los datos de entrenamiento mantienen una mayor densidad al acercarse a cero, o lo que es lo mismo, que estos datos tienen menores errores de predicción. Es por esto por lo que en el capítulo anterior se seleccionó una pérdida máxima relativamente menor que la de Pfizer, con un *Threshold* = 1.

5.3 ANOMALÍAS RESULTANTES

Si la pérdida de distribución de la muestra es mayor que el *Threshold*, la muestra se etiqueta como anomalía. La franja de valores que superan la línea naranja, es decir, el umbral que seleccionamos con anterioridad, se clasifican como anomalías. El resultado se puede observar gráficamente en la Figura 30 y Figura 31:

PFIZER

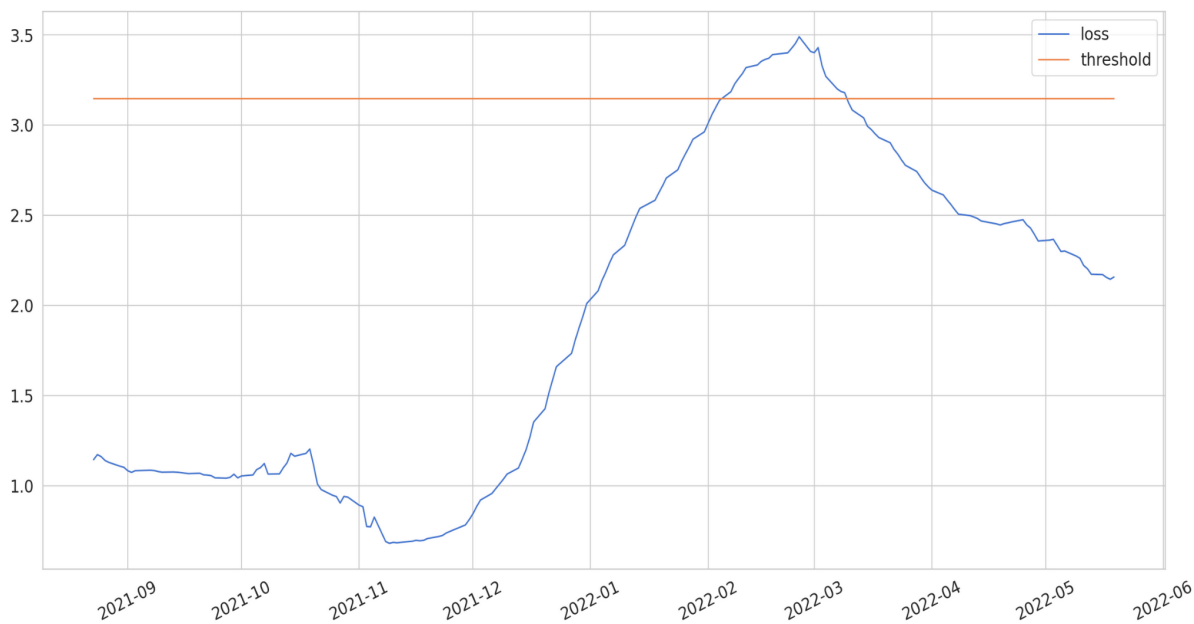


Figura 30. Pérdida de la muestra Pfizer. Fuente:elaboración propia

MODERNA

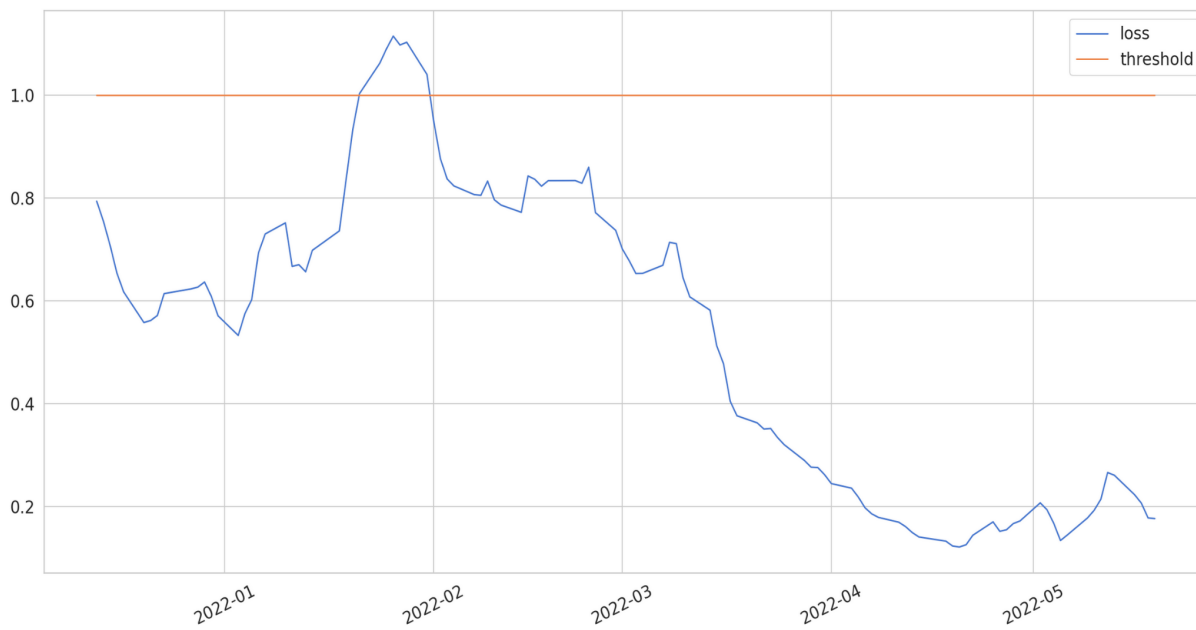


Figura 31. Pérdida de la muestra Moderna. Fuente:elaboración propia

Este conjunto de anomalías puntuales que superan el Threshold se han detectado con éxito en las Figuras 32 y 33:

PFIZER

	loss	threshold	anomaly	Close
Date				
2022-02-07	3.185809	3.15	True	5.422433
2022-02-08	3.229087	3.15	True	4.944971
2022-02-09	3.258417	3.15	True	4.872244
2022-02-10	3.285282	3.15	True	4.597149
2022-02-11	3.319940	3.15	True	4.654065
2022-02-14	3.334314	3.15	True	4.344189
2022-02-15	3.354075	3.15	True	4.341027
2022-02-16	3.364652	3.15	True	4.306245
2022-02-17	3.371540	3.15	True	4.059609
2022-02-18	3.391727	3.15	True	3.942614
2022-02-22	3.401540	3.15	True	3.626413
2022-02-23	3.426443	3.15	True	3.417721
2022-02-24	3.453135	3.15	True	3.129978
2022-02-25	3.490716	3.15	True	3.686492
2022-02-28	3.408964	3.15	True	3.439855
2022-03-01	3.402219	3.15	True	3.063576
2022-03-02	3.430782	3.15	True	3.680168
2022-03-03	3.331982	3.15	True	3.721274
2022-03-04	3.271579	3.15	True	3.980559
2022-03-07	3.201856	3.15	True	3.768704
2022-03-08	3.187317	3.15	True	3.597955
2022-03-09	3.180300	3.15	True	4.012178

Figura 32. Anomalías detectadas en la cotización bursátil de Pfizer

MODERNA

	loss	threshold	anomaly	Close
Date				
2022-01-21	1.002808	1	True	0.795655
2022-01-24	1.061683	1	True	0.766643
2022-01-25	1.089837	1	True	0.716788
2022-01-26	1.114393	1	True	0.742134
2022-01-27	1.097124	1	True	0.675731
2022-01-28	1.102446	1	True	0.789371
2022-01-31	1.039927	1	True	0.892642

Figura 33. Anomalías detectadas en la cotización bursátil de Moderna

Las anomalías que se han detectado en los datos de prueba se visualizan gráficamente como en las Figuras 34 y 35:

PFIZER

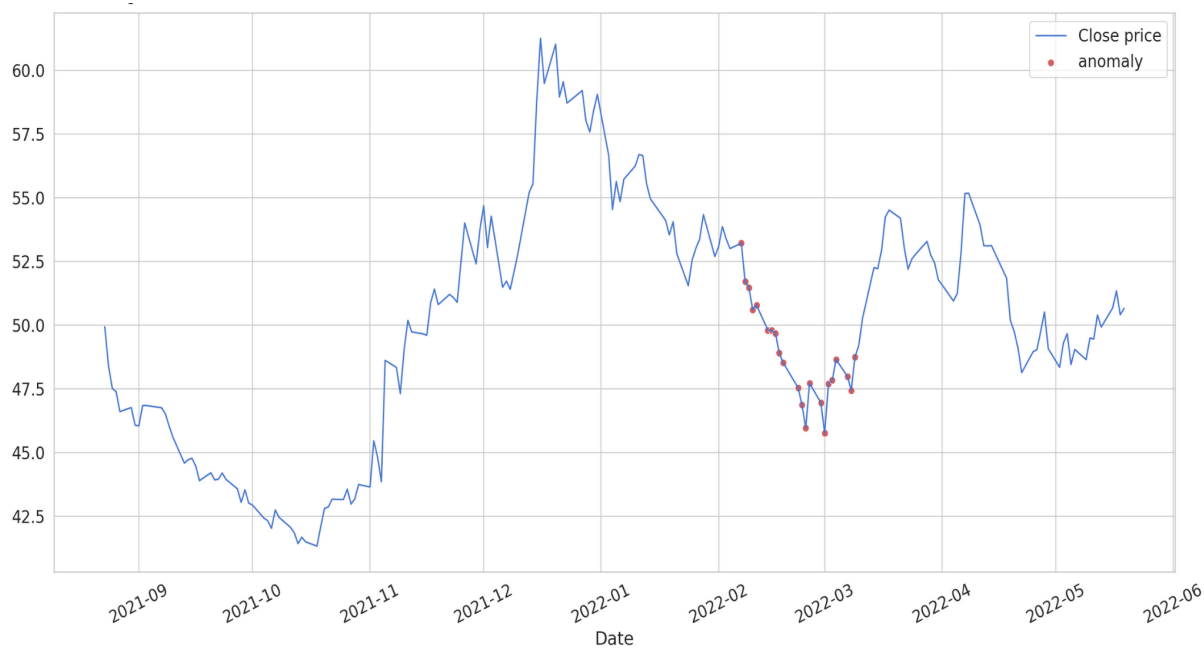


Figura 34. Anomalías detectadas en Pfizer. Fuente: elaboración propia.

MODERNA

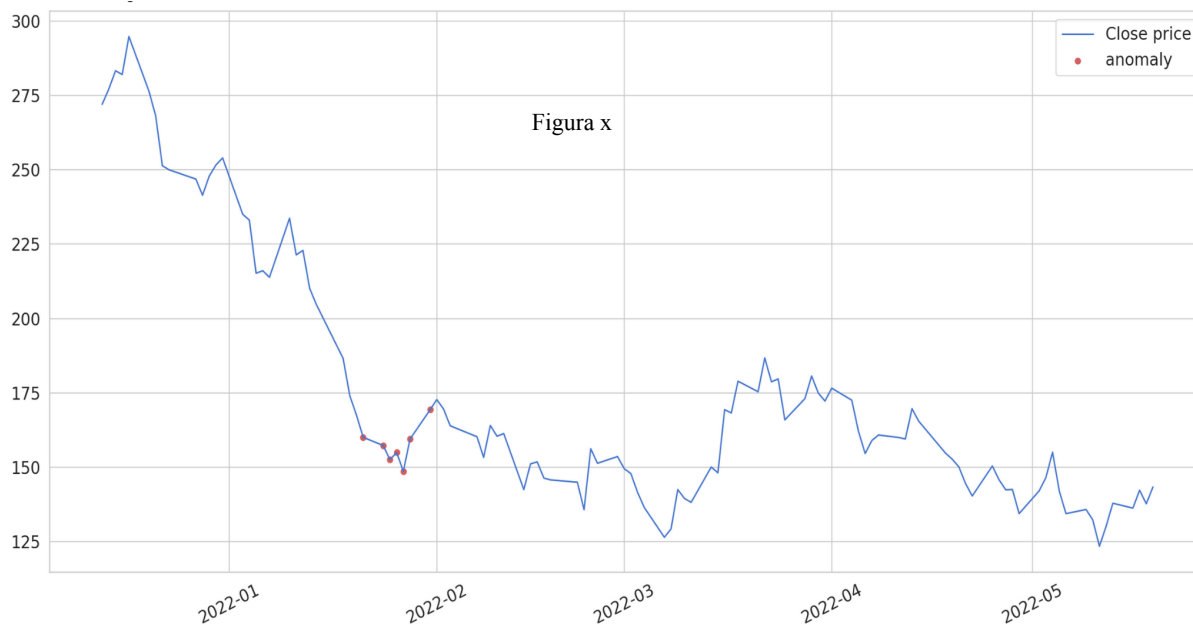


Figura x

Figura 35. Anomalías detectadas en Moderna. Fuente: elaboración propia.

Las anomalías puntuales detectadas en el precio de las acciones de Pfizer, tal y como se observaba en la Figura x, se concentran en un intervalo de tiempo bastante concreto (07/02/2022 - 09/03/2022). Tras una serie de investigaciones que nos permitieran saber las causas de estas anomalías en el precio de la acción, se ha encontrado información acerca de una de las posibles razones por las que se han detectado ese gran número de anomalías en un período tan corto de tiempo, y es que el día 8 de febrero de 2022, la farmacéutica estadounidense Pfizer anunciaba los resultados del primer trimestre del año. Pfizer dijo que esperaba generar un récord de entre 98 mil millones de dólares y 102 mil millones de dólares en ingresos en 2022, ayudado por sus 32 mil millones de dólares en ventas de vacunas Covid-19 y 22 mil millones de dólares en ventas de píldoras antivirales. Esto no alcanzó la estimación de consenso de 103,2 mil millones de dólares y los analistas esperaban ver 34 mil millones de dólares en ventas de vacunas en 2022. Los resultados provocaron una venta masiva de acciones, representando una corrección del 7% en tan solo unos días (Estrategias de inversión, 2022). La caída de las acciones se prolongó hasta el 1 de marzo de 2022 donde registró el precio de 45,53\$ por acción, un descenso del 11,55% desde los 51,68\$ por acción registrado el 8 de febrero de 2022.

A partir del 1 de marzo de 2022, la red neuronal detectó 6 anomalías más en el precio de la acción. Tras haber analizado posibles causas que provocaron estas anomalías hasta el 9 de marzo de 2022, se ha llegado a la conclusión de que el modelo, tras haber considerado una caída en el precio de la acción tan brusca, un movimiento en la dirección contraria, un movimiento alcista, provocó que esos datos se considerasen anomalías, no debido a alguna causa en específico. Esta conclusión la podemos confirmar ya que como se ha visto en la Figura x, la red neuronal se ha entrenado correctamente, sin presentar problemas como el gradiente desvaneciente, el cual hubiese provocado una capacidad de aprendizaje más lenta y por tanto posibles fallos en la detección de las anomalías.

En el caso de la compañía Moderna, se detectaron siete anomalías entre el 21 y el 31 de enero de 2022. Tras haber analizado las posibles causas de estas anomalías en el precio de la acción, se detectó que la farmacéutica Moderna anunciaba el 24 de enero los resultados de un ensayo clínico contra la esclerosis múltiple con aparente éxito (Sergi Alcalde, 2022), además, el 25 de enero la compañía anunciaba que la vacuna contra el VIH que se encontraba en desarrollo, entraba en fase prueba con humanos (Nancy Álvarez, 2022). A partir de estos anuncios, el precio de la acción ascendió de 146,91\$ a 168,45\$, es decir, un 14,86%. Las siguientes anomalías detectadas, los días 26, 27, 28 y 31 de enero acompañan el “furor” provocado por las noticias mencionadas anteriormente, por lo que se detectan como anomalías, pero no han tenido un evento en particular estos días.

6 CONCLUSIONES

En este trabajo, nos hemos centrado principalmente, desde un punto de vista financiero, en la detección y análisis de anomalías en series de tiempo.

En primer lugar, se puede concluir que el uso de Machine Learning abarca una amplia gama de algoritmos y herramientas para crear modelos que se utilizan en una gran variedad de aplicaciones, especialmente en la predicción y detección de anomalías en series temporales. A pesar de que no se haya demostrado empíricamente que este tipo de modelos tienen un mejor rendimiento que los modelos estadísticos, hemos observado la fiabilidad y la sencillez con la que se ha desarrollado la detección de las anomalías a través de una red neuronal *Autoencoder LSTM*. Las redes neuronales artificiales están compuestas de capas e interconexiones que permiten obtener automáticamente resultados de alta calidad, sin importar el volumen de datos que se pretenda estudiar, mientras que otro tipo de técnicas no son capaces de procesarlos debido a su complejidad, provocando altos márgenes de error.

Posteriormente, se puede concluir que el *Deep Learning* es un método muy funcional para numerosas áreas de trabajo, su estructura inspirada en el cerebro humano ofrece modelos que tienen una capacidad de aprendizaje impresionante, son capaces de aprender de sus errores pasados para predecir los sucesos futuros. Observando los resultados finales, se ha podido observar que el modelo, a medida que se entrenaba, era capaz de aprender con más facilidad para detectar las anomalías con precisión. Como aspecto negativo, cabe mencionar que el modelo, realmente, ha captado una serie de anomalías en las que no ha sucedido ningún evento en particular y sin embargo se detectaron como tal debido a que acompañaban el “movimiento” del precio de las acciones tal y como pudimos observar en los resultados.

Finalmente, se concluye que las técnicas de Deep Learning son desde luego muy eficaces en la detección de anomalías de series temporales, para futuros trabajos se pretende estudiar dos enfoques empíricamente, un enfoque estadístico y un enfoque de *Deep Learning* más complejo para comprobar cual de ellos obtiene mejores resultados en la detección de anomalías en series temporales.

BIBLIOGRAFÍA

[1]Animesh Patcha, Jung-Min Park, “An overview of anomaly detection techniques: Existing solutions and latest technological trends”, *Computer Networks*, Volume 51, Issue 12, 2007, Pages 3448-3470, ISSN 1389-1286.

[2]Muhammad Fahim and Alberto Sillitti, “Anomaly Detection, Analysis and Prediction Techniques in IoT Environment: A Systematic Literature Review”, volume 7, 2019, Page 81665.

[3]Andrew A. Cook; Goksel Misirli; Zhong Fan; “Anomaly Detection for IoT Time-Series Data: A Survey”; Volume: 7, Issue 7, July 2020, Page(s): 6481 - 6494.

[4]Antonio Rafael Sabino Parmezan, Vinicius M.A. Souza, Gustavo E.A.P.A. Batista: *Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model*, January 2019.

[5] Mohammad Braei, “Anomaly detection in univariate Time-series: A survey on the state-of-the-art”, *Department of Computer Science*, April 2020

[6]FU, Zhouyu; HU, Weiming; TAN, Tieniu. *Similarity based vehicle trajectory clustering and anomaly detection*. En *IEEE International Conference on Image Processing 2005*. Ieee, 2005. p. II-602.

[7]HUANG, Desen, et al. *Efficient time series anomaly detection by multiresolution self-supervised discriminative network*. *Neurocomputing*, 2022, vol. 491, p. 261-272.

[8]Vercruyssen, V., Meert, W., Verbruggen, G., Maes, K., Baumer, R., & Davis, J. (2018). *Semi-supervised anomaly detection with an application to water analytics*. In *2018 IEEE International Conference on Data Mining (ICDM) (Vol. 2018, pp. 527-536)*. IEEE.

[9]MUNIR, Mohsin, et al. *DeepAnT: A deep learning approach for unsupervised anomaly detection in time series*. *Ieee Access*, 2018, vol. 7, p. 1991-2005. TECNICA NO SUPER.

[10]LI, Jia, et al. *FluxEV: a fast and effective unsupervised framework for time-series anomaly detection*. En *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 2021. p. 824-832. TEC NO SUPER

[11]PEREIRA, Joao; SILVEIRA, Margarida. *Learning representations from healthcare time series data for unsupervised anomaly detection*. En *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*. IEEE, 2019. p. 1-7.

[12]Antonio Rafael Sabino Parmezan, Vinicius M.A. Souza, Gustavo E.A.P.A. Batista, "Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model", *Information Sciences*, Volume 484, 2019, Pages 302-337.

[13]SINGH, Amanpreet; THAKUR, Narina; SHARMA, Aakanksha. "A review of supervised machine learning algorithms". En *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. Ieee, 2016. p. 1310-1315.

[14]Luan, Siyu, et al. "Out-of-distribution detection for deep neural networks with isolation forest and local outlier factor". *IEEE Access*, 2021, vol. 9, p. 132980-132989.

[15]Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio, “How to Construct Deep Recurrent Neural Networks, Department of Information and Computer Science”, Aalto University School of Science, April 2014.

[16]Reproducido de “Redes Neuronales artificiales: Qué son y cómo se entrenan” por Oscar García-Olalla Olivera, 2019. Obra de Dominio Público. FIGURAS
<https://www.xeridia.com/blog/redes-neuronales-artificiales-que-son-y-como-se-entrenan-part-e-i>

[17]Uday Shankar Shanthamallu, Andreas Spanias, Cihan Tepedelenlioglu, and Mike Stanley, “A Brief Survey of Machine Learning Methods and their Sensor and IoT Applications” SenSIP Center, School of ECEE, Arizona State University, NXP Semiconductors, March 2018.

[18]Goldstein M, Uchida S (2016) “A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data”. PLoS ONE 11(4): e0152173.

[19] Anita Yadava, C K Jhaa, Aditi Sharanb “Optimizing LSTM for time series prediction in Indian stock market”, International Conference on Computational Intelligence and Data Science, 2019, pp. 10.

[20] van den Oord, Aaron and Dieleman, Sander and Zen, Heiga and Simonyan, Karen and Vinyals, Oriol and Graves, Alex and Kalchbrenner, Nal and Senior, Andrew and Kavukcuoglu, Koray. "WaveNet: A Generative Model for Raw Audio", 2016. sep, {arXiv:1609.03499}

[21]Chunyong Yin; Sun Zhang; Jin Wang; Neal N.Xiong; “Anomaly Detection Based on Convolutional Recurrent Autoencoder for IoT Time Series”; Volume: 52, Issue 1, Jan. 2022, Page(s): 112 - 122.

[22]Baur, Christoph; ALBARQOUNI, Shadi; NAVAB, Nassir. “Semi-supervised deep learning for fully convolutional networks”. En International conference on medical image computing and computer-assisted intervention. Springer, Cham, 2017. p. 311-319.

[23] Alex Sherstinsky, “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network”, *Physica D: Nonlinear Phenomena*, Volume 404, 2020, 132306, ISSN 0167-2789.

[24] Zhang, C., Li, S., Zhang, H., & Chen, Y. (2019). *Velc*: “A new variational autoencoder based model for time series anomaly detection”.

[25] Oleksandr I. Provotar; Yaroslav M. Linder; Maksym M. Veres; “Unsupervised Anomaly Detection in Time Series Using LSTM-Based Autoencoders”, Dec. 2019.

[26] Markus Thill, Wolfgang Konen, Hao Wang, Thomas Bäck, “Temporal convolutional autoencoder for unsupervised anomaly detection in time series, *Applied Soft Computing*, Volume 112, 2021, 107751, ISSN 1568-4946.

[27] H.D. Nguyen, K.P. Tran, S. Thomassey, M. Hamad; “Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management”; *International Journal of Information Management*, Volume 57, 2021, 102282, ISSN 0268-4012.

[28] Python official web page. (2022).

Recuperado de: <https://www.python.org/about/apps/>

[29] *Aprende Machine Learning*, march 2020.

<https://www.aprendemachinelarning.com/sets-de-entrenamiento-test-validacion-cruzada/>

[x] A. Namaki a, A.H. Shirazi b, R. Raei a, G.R. Jafari “Network analysis of a financial market based on genuine correlation and threshold method”

[30] Zanifa Omary, Fredrick Mtenzi “Machine Learning Approach to Identifying the Dataset Threshold for the Performance Estimators in Supervised Learning”, *International Journal for Infonomics (IJI)*, Volume 3, Issue 3, September 2010.

[31] A. Namaki, A.H. Shirazi, R. Raei, G.R. Jafari, *Network analysis of a financial market based on genuine correlation and threshold method*, *Physica A: Statistical Mechanics and its Applications*, Volume 390, Issues 21–22, 2011, Pages 3835-3841.
