

TESIS DOCTORAL

2023



**MODELO COMBINADO BASADO EN
REDES NEURONALES
RECURRENTE Y REDES
CONVOLUCIONALES DE GRAFOS
PARA LA PREDICCIÓN DE SERIES
TEMPORALES ECONÓMICAS**

ANA LAZCANO DE ROJAS

**PROGRAMA DE DOCTORADO EN
INGENIERÍA DE SISTEMAS Y CONTROL**

**DIRECTORES:
PEDRO JAVIER HERRERA CARO
MANUEL ÁNGEL MONGE MORENO**

**Modelo combinado basado en redes neuronales
recurrentes y redes convolucionales de grafos
para la predicción de series temporales
económicas**

*Memoria que presenta para optar al grado de Doctor en
Ingeniería de Sistemas y Control*

Ana Lazcano de Rojas

Dirigida por los doctores

Pedro Javier Herrera Caro

Manuel Ángel Monge Moreno

Departamento de Ingeniería de Software y Sistemas Informáticos
Universidad Nacional de Educación a Distancia

Madrid 2023

“What we want is a machine that can learn from experience.”

Alan Turing

*Dedicado a
mi familia: Adela, Gonzalo, Pollito y Carlos, por ser mis pilares.*

AGRADECIMIENTOS

A lo largo de este camino como doctorando me han acompañado muchas personas a las que no puedo dejar de agradecer haber estado a mi lado:

En primer lugar, queridos hijos es difícil reflejar en pocas líneas lo mucho que os quiero, estas palabras son para agradecer vuestra comprensión por las horas que os he robado, por susurrar cuando estaba con el ordenador, por los besos de ánimo y los dibujos mientras trabajaba.

Carlos, gracias por tu apoyo incondicional, por hacer de papá y de mamá para que yo tuviera más horas, gracias por hacerme llegar hasta aquí, por confiar en mí desde el primer día incluso cuando ni yo lo hacía, por tus palabras de aliento y por acompañarme tantas horas, sin ti no hubiera sido posible. Este trabajo también es tuyo. Te quiero.

A mis directores, el Dr. Pedro Javier Herrera y el Dr. Manuel Monge, por acompañarme en este camino. A Javier por enseñarme la paciencia y a ser rigurosa con los detalles. A Manuel, director, mentor y amigo además de brillante y trabajador incansable, por inculcarme la pasión por la investigación, por la ayuda día y noche, por las largas charlas y las palabras de ánimo cuando las necesitaba, espero poder seguir tu ejemplo.

A Marta, gracias por formar parte de mi vida, por estar a mi lado durante esta locura.

A Elvira y Verónica, gracias por creer en mí desde hace tantos años.

A mis padres, por vuestro apoyo y lecciones en todos los pasos que he dado. A mi hermana Adela, seguro que estás orgullosa.

Al Padre Justo Gómez L.C y al Padre Gabriel Guajardo L.C, por mantenerme cerca de Dios en un proceso tan científico.

Por acompañarme en el camino, por compartir mis alegrías y mis penas, a mis amigos doctorandos: León, gracias por escuchar siempre; y Pablo, ojalá sigamos siempre compartiendo camino. A Berta por todo lo que compartimos. A Natalia por vivir mis alegrías como propias.

Al Dr. Carlos Poza por invitarme a iniciar esta tesis, que nunca pensé que sería tan emocionante, y por la confianza depositada desde el primer momento.

Al Dr. Álvaro José García Tejedor, uno de los investigadores que más admiro. Gracias por ayudarme, apoyarme y guiarme, por tus consejos y consuelo.

A la Universidad Francisco de Vitoria, por darme mi primer título académico, por hacerme sentir en casa. En especial al Dr. José Antonio Verdejo como Secretario General y amigo.

A la Dr. María Elena Bárcena y al Dr. Timothy Martin por tanto apoyo, ayuda y cariño.

A todos los que de alguna manera habéis formado parte de esto.

RESUMEN

En las últimas décadas se ha afianzado un conjunto de metodologías que, inspiradas en sistemas neuronales y biológicos, pueden resolver problemas como el reconocimiento de formas e imágenes y la toma de decisiones en distintos ámbitos mediante soluciones robustas.

Las redes neuronales artificiales, gracias a su facilidad de aplicación práctica y resultados, han generado especial interés entre investigadores estadísticos, matemáticos y analistas de datos, que ya las incorporan al conjunto de herramientas empleadas en tareas de análisis y predicción. Hasta el momento, las investigaciones que incorporaban predicciones en sus estudios recurrían a las técnicas estadísticas tradicionales, las cuales aplican modelos de regresión para lograr, mediante el análisis de datos pasados, los posibles valores futuros.

Una red neuronal artificial se define como un conjunto de neuronas que mediante una combinación de pesos y funciones es capaz de realizar tareas de predicción, destacando la predicción de series temporales. Las series temporales consisten en un conjunto de valores que se consideran observaciones tomadas a lo largo de un periodo de tiempo y con una periodicidad determinada.

A lo largo del siglo XX se desarrollaron distintos tipos de redes neuronales que, basadas en algoritmos matemáticos con distintos objetivos, lograron confirmar la precisión de las redes neuronales para la predicción. En función de las características de la red neuronal, esta tendrá la capacidad de realizar pronósticos en un determinado campo, como procesamiento de imágenes, series temporales y lenguaje natural. Estos desarrollos han llevado en los últimos años a la investigación acerca de la viabilidad en la combinación de distintos modelos para una determinada labor de predicción, de forma que mediante la

creación de modelos híbridos se mejoren los resultados obtenidos en los modelos por separado.

En este trabajo de investigación se desarrolla el estudio de un modelo híbrido combinado BiLSTM-GCN que permite la predicción de valores en el campo de la economía. El modelo propuesto combina redes de tipo *Bidirectional Long-short Term Memory* (BiLSTM), *Graph Convolutional Network* (GCN) y *Long-short Term Memory* (LSTM) aportando las características de ambos tipos para la obtención de resultados más precisos. Las redes de tipo LSTM y BiLSTM han demostrado ampliamente en la literatura su capacidad de pronóstico de series temporales con resultados altamente precisos y ajustados, mientras que las redes GCN se centran en las predicciones de modelos de grafos, realizando la predicción del siguiente nodo en función de los enlaces con los vecinos. La combinación de los dos modelos de redes neuronales permite captar las características de las series temporales desde varias perspectivas.

Las series temporales analizadas en este documento parten del ámbito económico. Este tipo de series temporales se caracterizan por no presentar estacionariedad ni una tendencia clara, de forma que es requerido aplicarle diferentes tipos de pruebas para obtener series temporales óptimas para su predicción.

Para la evaluación del modelo propuesto se procedió a realizar una comparativa con diferentes métodos estadísticos clásicos y los modelos empleados por separado, permitiendo así validar la calidad de la precisión obtenida con el nuevo modelo.

Palabras clave: *Aprendizaje profundo; Redes neuronales; Predicción de series temporales; LSTM; GCN; Capas ocultas; Precisión.*

ABSTRACT

In recent decades, a set of methodologies has been consolidated that, inspired by neural and biological systems, can solve problems such as recognition of shapes and images and decision-making in different fields through robust solutions.

Among these methodologies, neural networks stand out, which, thanks to its ease of practical application and results, has generated special interest among statistical researchers, mathematicians, and data analysts, who already incorporate neural networks into the set of tools used in analysis tasks and prediction. Until now, the investigations that incorporated predictions in their studies resorted to traditional statistical techniques, which apply regression models to achieve, through analysis of past data, possible future values.

A neural network is defined as a set of neurons that through a combination of weights and functions is capable of performing prediction tasks, highlighting the prediction of time series. Time series consist of a set of values that are considered observations taken over a period of time and with a certain periodicity.

Throughout the 20th century, different types of neural networks were developed which, based on mathematical algorithms with different objectives, managed to confirm the precision of neural networks for prediction. Depending on the characteristics of the neural network, it will have the ability to make forecasts in a certain field, such as images, time series and language. These developments have led in recent years to research on the feasibility of combining different models for a certain prediction task, so that by creating hybrid models the results obtained in the separate models are improved.

In this research work, the study of a combined BiLSTM-GCN hybrid model that allows the prediction of values in the field of economics is developed. The proposed model

combines Bidirectional Long-short Term Memory (BiLSTM), Graph Convolutional Network (GCN) and Long-short Term Memory (LSTM) type networks, providing the characteristics of both types to obtain more accurate results. LSTM and BiLSTM type networks have widely demonstrated in the literature their ability to forecast time series with highly accurate and adjusted results, while GCN networks focus on graph model predictions, predicting the next node based on of links with neighbours. The combination of the two neural network models allows capturing the characteristics of the time series from various perspectives.

The time series analysed in this document start from the economic sphere. This type of time series is characterized by not presenting stationarity or a clear trend, so it is necessary to apply different types of tests to obtain optimal time series for its prediction.

For the evaluation of the model, a comparison has been made with different classical statistical methods and with the models used separately, to have a complete vision of the quality of the precision obtained with the new model.

Keywords: *Deep learning; Neural networks; Time series forecasting; LSTM; GCN; Hidden layers; Precision.*

ÍNDICE

I. BLOQUE I: INTRODUCCIÓN, FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE	25
1 INTRODUCCIÓN	27
1.1 ANTECEDENTES	27
1.2 DESCRIPCIÓN DEL PROBLEMA	28
1.3 MOTIVACIÓN DEL TRABAJO.....	28
1.4 OBJETIVOS.....	29
1.5 METODOLOGÍA.....	29
1.6 APORTACIONES DE LA INVESTIGACIÓN	30
1.7 ORGANIZACIÓN DE LA MEMORIA DE TESIS	31
2 REDES NEURONALES ARTIFICIALES	33
2.1 PERCEPTRÓN MULTICAPA	36
2.2 REDES NEURONALES RECURRENTEES	38
2.2.1 ARQUITECTURA RNN.....	40
2.2.2 ALGORITMO BACKPROPAGATION.....	41
2.2.3 LONG SHORT TERM MEMORY.....	43
2.2.4 BIDIRECCIONAL LSTM	46
2.3 REDES CONVOLUCIONALES	47
2.4 REDES NEURONALES DE GRAFOS.....	50
2.4.1 REDES CONVOLUCIONALES DE GRAFOS	51
2.4.2 REDES CONVOLUCIONALES DE GRAFOS ESPECTRALES.....	53
2.4.3 REDES NEURONALES CONVOLUCIONALES DE GRAFOS ESPACIALES	54
3 SERIES TEMPORALES	57

3.1	DESCRIPCIÓN DE LAS SERIES TEMPORALES	57
3.1.1	PROCESO ESTOCÁSTICO	59
3.1.2	ESTUDIO DE LA TENDENCIA	60
3.1.3	COMPONENTE ESTACIONAL.....	61
3.2	ANÁLISIS DE SERIES TEMPORALES	61
3.2.1	VISUALIZACIÓN DE LAS SERIES TEMPORALES.....	61
3.2.2	DESCOMPOSICIÓN DE LA SERIE TEMPORAL.....	62
3.2.2.1	ESTACIONARIEDAD	63
3.2.2.1.1	TEST DE DICKEY FULLER	64
3.2.2.1.2	TEST DE PHILLIPS-PERRON	64
3.2.2.1.3	DIFERENCIACIÓN DE SERIES TEMPORALES	65
3.2.3	MODELOS AUTORREGRESIVOS	66
3.2.3.1	MODELO DE AUTORREGRESIÓN LINEAL.....	66
3.2.3.2	MODELO DE MEDIAS MÓVILES	66
3.2.3.3	MODELO ARMA.....	67
3.2.3.4	MODELO ARIMA	67
3.2.3.5	MODELO SARIMA	68
3.2.3.6	MODELO ARFIMA	68
3.2.3.7	CRITERIOS DE INFORMACIÓN PARA LA ELECCIÓN DEL MODELO.....	69
4	TÉCNICAS DE PREDICCIÓN DE SERIES TEMPORALES	71
4.1	TÉCNICAS CLÁSICAS DE PREDICCIÓN	71
4.2	TÉCNICAS DE SUAVIZADO	73
4.2.1	TÉCNICAS DE PROMEDIADO.....	74
4.2.2	TÉCNICAS DE SUAVIZADO EXPONENCIAL	74
4.3	MÁQUINAS DE VECTOR SOPORTE.....	75
4.3.1	HIPERPLANO	76
4.3.1.1	HIPERPLANO PARA LA CLASIFICACIÓN BINARIA	76
4.3.2	CLASIFICADOR DE VECTORES SOPORTE	77
4.3.2.1	MÁQUINAS DE VECTOR SOPORTE	78
4.4	FACEBOOK PROPHET	78
4.5	TÉCNICAS DE PREDICCIÓN CON REDES NEURONALES ARTIFICIALES	80
4.5.1	OVERFITTING Y UNDERFITTING	81
II.	BLOQUE II: SOLUCIÓN PROPUESTA, RESULTADOS Y CONCLUSIONES	85

5	MODELO BiLSTM-GCN	87
5.1	ARQUITECTURA DE UNA RED NEURONAL	87
5.2	ARQUITECTURA DEL MODELO BiLSTM-GCN	89
5.2.1	PARÁMETROS DEL MODELO BiLSTM-GCN	92
5.2.2	FUNCIÓN DE ACTIVACIÓN DEL MODELO BiLSTM-GCN	96
5.2.3	GRAFO DE VISIBILIDAD	97
5.2.4	COMPLEJIDAD DEL MODELO BiLSTM-GCN	98
6	EXPERIMENTACIÓN Y ANÁLISIS DE RESULTADOS	101
6.1	HERRAMIENTAS EMPLEADAS	101
6.2	TRATAMIENTO DE LOS DATOS	102
6.2.1	DIVISIÓN DE LOS DATOS	102
6.2.2	NORMALIZACIÓN DE LOS DATOS	103
6.3	MÉTRICAS DE ERROR EMPLEADAS	104
6.3.1	RAÍZ DEL ERROR CUADRÁTICO MEDIO	104
6.3.2	ERROR CUADRÁTICO MEDIO	105
6.3.3	ERROR PORCENTUAL ABSOLUTO MEDIO	105
6.3.4	COEFICIENTE DE DETERMINACIÓN.....	105
6.4	PRECIOS DEL PETRÓLEO.....	106
6.4.1	TÉCNICAS DE PREDICCIÓN DE PRECIOS DEL PETRÓLEO	107
6.4.2	DATOS UTILIZADOS	108
6.4.2.1	ESTUDIO DE LA SERIE TEMPORAL WTI.....	109
6.4.3	RESULTADOS	111
6.5	PRECIOS DE LAS MATERIAS PRIMAS RARAS	118
6.5.1	TÉCNICAS DE PREDICCIÓN DE PRECIOS DE LAS MATERIAS PRIMAS RARAS	119
6.5.2	DATOS UTILIZADOS	120
6.5.2.1	ESTUDIO DE LA SERIE TEMPORAL REE.....	120
6.5.3	RESULTADOS	123
6.6	PRECIOS DE LAS MATERIAS PRIMAS	129
6.6.1	TÉCNICAS PREDICCIÓN DE PRECIOS MATERIAS PRIMAS	130
6.6.2	DATOS UTILIZADOS	131
6.6.2.1	ESTUDIO DE LA SERIE TEMPORAL BCTR.....	132
6.6.3	RESULTADOS	133
7	CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS	141

BIBLIOGRAFÍA	143
---------------------------	------------

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Modelo de neurona artificial	34
Ilustración 2. Modelos de perceptrón. (a) Perceptrón simple; (b) Perceptrón multicapa.	37
Ilustración 3. Red Neuronal Recurrente	39
Ilustración 4. Arquitectura de Red Neuronal Recurrente	40
Ilustración 5. Modelo de la red neuronal multicapa retropropagación (Pajares et al., 2021).....	42
Ilustración 6. Estructura de una celda de una red LSTM	44
Ilustración 7. Estructura de una red BiLSTM	47
Ilustración 8. Estructura de una Red Convolutiva	49
Ilustración 9. Estructura de un grafo	51
Ilustración 10. Estructura de Red Convolutiva de Grafos	52
Ilustración 11. Gráfico de la deuda nacional de Inglaterra (Playfair, 1786).	62
Ilustración 12. Descomposición de una serie temporal.	63
Ilustración 13. Arquitectura de una red neuronal	89
Ilustración 14. Arquitectura del modelo BiLSTM-GCN.....	90
Ilustración 15. Arquitectura del modelo BiLSTM.....	91
Ilustración 16. Arquitectura del modelo GCN-LSTM.....	91
Ilustración 17. Arquitectura del modelo combinado BiLSTM-GCN.....	92
Ilustración 18. Comparativa RMSE respecto al número de capas	95
Ilustración 19. Comparativa tiempo respecto al número de capas	95
Ilustración 20. Ejemplo de una serie temporal con 12 datos y el grafo asociado mediante el algoritmo de visibilidad	98
Ilustración 21. División de los datos WTI en los conjuntos de entrenamiento, validación y prueba	109
Ilustración 22. Representación de la serie temporal WTI	110
Ilustración 23. Representación de la media móvil y la desviación estándar WTI.....	110
Ilustración 24. Descomposición de la serie WTI.....	111

Ilustración 25. Métricas de resultados WTI: (a) MSE; (b) RMSE; (c) R2; (d) MAPE; y (e) Tiempo.	113
Ilustración 26. Ajuste de los resultados obtenidos frente al valor real del índice WTI por: (a) ARIMA; (b) PROPHET; (c) BiLSTM; (d) GCN-LSTM; y (e) BiLSTM-GCN	114
Ilustración 27. Ajuste de los resultados obtenidos por BiLSTM, GCN-LSTM y BiLSTM-GCN frente al valor real del índice WTI en el año 2020.	116
Ilustración 28. Valores atípicos	118
Ilustración 29. División datos REE en los conjuntos de entrenamiento, validación y prueba.	121
Ilustración 30. Representación de la serie temporal REE	121
Ilustración 31. Representación de la media móvil y la desviación estándar REE.....	122
Ilustración 32. Descomposición de la serie REE.....	123
Ilustración 33. Métrica de resultados REE: (a) MSE; (b) RMSE; (c) R2; (d) MAPE; y (e) Tiempo.	124
Ilustración 34. Ajuste de los resultados obtenidos frente al valor real del índice de REE por: (a) ARIMA; (b) PROPHET; (c) BiLSTM; (d) GCN-LSTM; y (e) BiLSTM-GCN.	125
Ilustración 35. Ajuste de los resultados obtenidos por BiLSTM, GCN-LSTM y BiLSTM-GCN frente al valor real del índice REE.	127
Ilustración 36. Valores atípicos serie REE	129
Ilustración 37. Serie temporal BCTR en los conjuntos de entrenamiento, validación y prueba.	132
Ilustración 38. Representación de la serie temporal BCTR	132
Ilustración 39. Representación de la media móvil y la desviación estándar BCTR.....	133
Ilustración 40. Descomposición de la serie BCTR.....	134
Ilustración 41. Métrica de resultados BCTR: (a) MSE; (b) RMSE; (c) R2; (d) MAPE; y (e) Tiempo.	135
Ilustración 42. Ajuste de los resultados obtenidos frente al valor real del índice BCTR por: (a) ARIMA; (b) PROPHET; (c) BiLSTM; (d) GCN-LSTM; y (e) BiLSTM-GCN.	136
Ilustración 43. Ajuste de los resultados obtenidos por BiLSTM, GCN-LSTM y BiLSTM-GCN frente al valor real del índice BCTR.	138
Ilustración 44. Valores atípicos serie BCTR	140

ÍNDICE DE TABLAS

Tabla 1. Parámetros seleccionados	94
Tabla 2. Complejidad de los modelos	98
Tabla 3. Tabla de herramientas utilizadas en la investigación.	101
Tabla 4. Resultados Serie Temporal WTI	111
Tabla 5. Resultados obtenidos por el modelo BiLSTM-GCN en la predicción de precios del petróleo WTI, frente a ARIMA, PROPHET, BiLSTM y GCN-LSTM.	112
Tabla 6. Test de Friedman serie WTI.....	116
Tabla 7. Test Wilcoxon serie WTI	117
Tabla 8. Valores estadísticos resultados WTI	117
Tabla 9. Resultados Serie Temporal REE	122
Tabla 10. Resultados obtenidos por el modelo BiLSTM-GCN en la predicción de precios de REE, frente a ARIMA, PROPHET, BiLSTM y GCN-LSTM.	123
Tabla 11. Test Friedman serie REE.....	127
Tabla 12. Test Wilcoxon serie REE	128
Tabla 13. Valores estadísticos resultados REE	128
Tabla 14. Resultados Serie Temporal BCTR	133
Tabla 15. Resultados obtenidos por el modelo BiLSTM-GCN en la predicción de precios de materias BCTR, frente a ARIMA, PROPHET, BiLSTM y GCN-LSTM... ..	134
Tabla 16. Test Friedman serie BCTR.....	138
Tabla 17. Test Wilcoxon serie BCTR	139
Tabla 18. Valores estadísticos resultados serie BCTR.....	139

LISTA DE SÍMBOLOS, ABREVIATURAS Y SIGLAS

ADF	Augmented Dickey Fuller	Dickey Fuller Aumentado
AIC	Akaike Information Criteria	Criterio de Información de Akaike
AIE	International Energy Agency	Agencia Internacional de Energía
ANN	Artificial Neural Network	Red Neuronal Artificial
AR	Autorregresion	Autorregresión
ARMA	AutoRegressive Moving Average models	Modelos de media móvil autorregresiva
ARIMA	AutoRegresive Integrated Moving Average	Media móvil integrada autorregresiva
ARFIMA	Autoregressive Fractionally Integrated Moving Average	Media móvil autorregresiva fraccionalmente integrada
BCTR	Bloomberg Commodities Total Return	
BIC	Bayesian Information Criteria	Criterios de Información Bayesiano
BiLSTM	Bidirectional LSTM	LSTM Bidireccional
CNN	Convolutional Neural Network	Red Neuronal Convolutacional
CM	Correlation Matrix	Matriz de Correlación
CRM	Critical Rare Materials	Materias Primas Críticas
DNN	Deep Neural Network	Red Neuronal Profunda
EE.UU	United States	Estados Unidos
ELU	Exponential Linear Unit	Unidad Lineal Exponencial
GCN	Graph Convolutional Network	Red Convolutacional de Grafos
GNN	Graph Neural Network	Red Neuronal de Grafos
LMS	Last Min Square	Mínimos Cuadrados
LSTM	Long-Short Term Memory	Memoria a Corto Plazo
MA	Moving Average	Media Móvil
MAPE	Mean Absolute Percentage Error	Error Porcentual Absoluto Medio
MBPD	Million Barrels Per Day	Millones de Barriles por Día
MLP	Multi Layer Perceptron	Perceptrón Multicapa
MSE	Mean Square Error	Error Cuadrático Medio

OCDE	Organization for Economic Co-operation and Development	Organización para la Cooperación y el Desarrollo Económicos
PP	Phillips-Perron	
PReLU	Parametric Rectified Linear Unit	Unidad Lineal Rectificada Parametrizada
REE	Rare Earth Materials	Materias Primas Raras
ReLU	Rectified Linear Unit	Unidad Lineal Rectificada
RMSE	Root Mean Square Error	Raíz del Error Cuadrático Medio
RMSProp	Root Mean Square Propagation	Propagación del Error Cuadrático Medio
RNA		Red Neuronal Artificial
RNN	Recurrent Neural Network	Red Neuronal Recurrente
SVC	Support Vector Classifier	Clasificador de Vector Soporte
SVM	Support Vector Machines	Máquinas de Vector Soporte
SVR	Support Vector Regression	Regresión de Vector Soporte
WTI	West Texas Intermediate	

BLOQUE I

INTRODUCCIÓN, FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

Este bloque se organiza en cuatro capítulos; en primer lugar, se realiza una introducción a la presente tesis doctoral, detallando su motivación y objetivos, así como la problemática abordada, la metodología seguida y las principales aportaciones realizadas; en el segundo capítulo se describen las redes neuronales artificiales y cómo han evolucionado progresivamente desde sus orígenes a mediados del siglo XX. El tercer capítulo se centra en las series temporales, exponiendo sus fundamentos teóricos, así como una descripción de su análisis, descomposición y principales técnicas de modelado. Por último, el cuarto capítulo supone la puesta en común de los dos aspectos anteriores, mostrando así el estado del arte en lo que a la predicción de series temporales se refiere sobre distintos tipos de técnicas y empleando redes neuronales artificiales, junto a las principales aplicaciones existentes en la actualidad.

1 INTRODUCCIÓN

1.1 ANTECEDENTES

Las series temporales son una sucesión de datos ordenados cronológicamente con el fin de caracterizar un sistema observado y predecir su comportamiento futuro. Técnicas como las redes neuronales, prometen conocimientos que los enfoques tradicionales no pueden alcanzar.

El Aprendizaje Automático es una rama de la Inteligencia Artificial, que tiene como objetivo dotar a los ordenadores de la capacidad de aprendizaje, creando sus propias reglas sin la necesidad de un componente humano y logrando, de ese modo, la representación de datos y creación de modelos que originen respuesta a los problemas planteados.

El Aprendizaje Profundo constituye una rama importante del Aprendizaje Automático, abriendo un campo de conocimiento muy prometedor en continuo auge, gracias en parte a los avances tecnológicos que permiten el procesamiento de grandes cantidades de datos con estructuras complejas, y cuyo máximo exponente son las redes neuronales, y más específicamente las catalogadas como *profundas* (Pajares et al., 2021). Las redes neuronales profundas gozan de un gran auge en los últimos años gracias a su posibilidad de aplicación en diferentes campos y resultados con gran precisión, superando en la mayoría de los casos a los obtenidos por las técnicas estadísticas tradicionales.

Los primeros modelos de redes neuronales se basaron en el propuesto por McCulloch y Pitts (1943), investigación en la que trataban de emular el funcionamiento de una neurona a través de las matemáticas, organizándolas en capas de forma similar al córtex cerebral,

no siendo hasta 1986 cuando la investigación del algoritmo de retropropagación (*backpropagation* en terminología inglesa) provocó un gran aumento de la popularidad de estos sistemas (Rumelhart et al., 1986).

1.2 DESCRIPCIÓN DEL PROBLEMA

La importancia de las series temporales en la economía viene determinada por la necesidad de recoger las variables a lo largo del tiempo, no en un periodo concreto, siendo el orden de los datos otro factor determinante. Las observaciones no son independientes o, expresado de otra forma, se puede decir que las observaciones pasadas pueden afectar a las futuras, y por lo tanto estas pueden llegar a ser predichas.

Los economistas se enfrentan al reto de realizar una toma de decisiones en un contexto de incertidumbre. De ese modo, la posibilidad de realizar la predicción de variables cobra gran importancia en este contexto, permitiendo reducir la incertidumbre y mejorar la información arrojada, ya que cuando la incertidumbre aumenta se produce una demanda de previsiones.

1.3 MOTIVACIÓN DEL TRABAJO

La gran cantidad de datos disponibles gracias al *Big Data*, que hace referencia a una cantidad inmensa de información disponible en un periodo de tiempo breve y en ocasiones en tiempo real, permiten que los sistemas de redes neuronales realicen predicciones en base a los datos históricos almacenados. Este escenario, en el que se da una situación de datos tan abundante, es reciente ya que hasta hace no mucho uno de los principales problemas en las tareas de predicción era la escasez de información en la que basar los pronósticos. Todo ello ha transformado el problema actual en realizar una extracción y tratamiento de los datos eficiente.

De la necesidad de realizar predicciones cada vez más precisas, logrando el mayor grado de exactitud posible en las predicciones económicas con las que poder anticipar acontecimientos económicos con un margen pequeño de error, surge la motivación de esta investigación, en la que combinando diferentes modelos de redes neuronales

profundas, que han arrojado por separado resultados prometedores en la predicción de series temporales frente a técnicas clásicas, se logra un modelo híbrido que reduce las métricas de error y proporciona resultados más ajustados en la evaluación del modelo.

1.4 OBJETIVOS

Los principales objetivos del presente trabajo de investigación pueden desglosarse en los siguientes puntos:

- a) Realizar una revisión de la literatura acerca de las series temporales y las técnicas de análisis y modelado.
- b) Investigar el estado del arte de las redes neuronales artificiales aplicadas a las tareas de predicción de series temporales.
- c) Construir un modelo de red neuronal híbrido que permita realizar pronósticos de series temporales económicas con independencia de las características de la serie, permitiendo realizar predicciones ajustadas y logrando mejores resultados en las métricas de error seleccionadas en la comparativa con otros modelos de redes neuronales y de técnicas de predicción clásicas.

En concreto, en el enfoque propuesto se combina un modelo de memoria de largo-corto plazo (más conocido por *Long-short Term Memory*, LSTM, en terminología inglesa), junto con un modelo de red neuronal de grafos (*Graph Convolutional Network*, GCN, en terminología inglesa), capturando las características que hacen de los dos modelos técnicas precisas de pronóstico de series temporales. El uso de este tipo de redes viene justificado no solo por la literatura existente sino como modelo innovador en el uso de las redes convolucionales de grafos para la predicción de series temporales, convirtiendo los datos en grafos para su interpretación.

1.5 METODOLOGÍA

Durante el desarrollo de la presente tesis doctoral, se parte del análisis de las diferentes técnicas de predicción de series temporales, para lo que es requerido un estudio detallado

acerca de las características que conforman las series temporales y cómo dichas características impactan de forma directa en los métodos de predicción empleados.

Una vez llevados a cabo diversos experimentos sobre la predicción de valores económicos con redes neuronales recurrentes, se decide profundizar en la investigación mediante la incorporación de un modelo de red neuronal de grafos, que permitiera mejorar los resultados existentes en la literatura mediante una disminución en las métricas de error.

Para comprobar la precisión del modelo propuesto se plantearon tres experimentos, en donde se realizó la predicción de tres series temporales económicas sobre: petróleo, materias primas raras y el índice bursátil de materias primas. Primero, mediante el análisis de las características de las series temporales y, posteriormente, la tarea de predicción mediante los modelos ARIMA, PROPHET, BiLSTM y GCN-LSTM en contraposición con el modelo propuesto BiLSTM-GCN, obteniendo menores tasas de error en todos los casos.

1.6 APORTACIONES DE LA INVESTIGACIÓN

La principal aportación de esta tesis doctoral consiste en la incorporación a la literatura de una nueva metodología para la predicción de series temporales derivada de la creación de un modelo híbrido que combina redes neuronales recurrentes y redes neuronales de grafos.

Dicha aportación se encuentra recogida en el artículo titulado “*A Combined Model Based on Recurrent Neural Networks and Graph Convolutional Networks for Financial Time Series Forecasting*” (Lazcano et al., 2023), publicado en la revista *Mathematics*, indexada en el *Journal Citation Reports* (JCR-WOS) y ubicada en el primer cuartil (Q1) en el último año publicado, 2021. En concreto, esta publicación recoge el modelo híbrido propuesto con una detallada descripción de su arquitectura. Para la comprobación de la precisión del modelo se lleva a cabo la predicción del precio del petróleo a partir del índice *West Texas Intermediate* (WTI) partiendo de los datos desde enero de 1983 hasta junio de 2022. Los resultados obtenidos pueden encontrarse en el capítulo 6 (apartado 6.4).

Por otro lado, el artículo titulado “*Commodity Prices after COVID-19: Persistence and Time Trends*” (Monge y Lazcano, 2022), y publicado en la revista *Risks* indexada en el *Scimago Journal Rank* (SJR) y ubicada en el segundo cuartil (Q2) en el último año publicado (2021) recoge, desde un punto más económico, el análisis y estudio de las series temporales derivadas del índice de materias primas *Bloomberg Commodities Total Return* (BCTR), con el objetivo de conocer el comportamiento de la serie temporal, con datos que abarcan desde enero de 1991 hasta abril de 2021, tras el inicio de la pandemia de COVID-19. Para ello, se realizó un análisis comparando una técnica clásica, ARFIMA, con un modelo de perceptrón multicapa para realizar la predicción de la serie, la cual reflejaba una recuperación de los valores tras el descenso producido durante la crisis sanitaria. Este trabajo permitió confirmar que un modelo basado en redes neuronales podía mejorar las técnicas clásicas, sentando las bases de la investigación desarrollada en la presente tesis doctoral.

1.7 ORGANIZACIÓN DE LA MEMORIA DE TESIS

La memoria de tesis se presenta estructurada en capítulos, siguiendo el orden natural de la investigación. Los capítulos se encuentran organizados dentro de dos bloques: el primero (*Introducción, Fundamentación Teórica y Estado del Arte*) aglutina los cuatro primeros capítulos, y el segundo bloque (*Solución Propuesta, Resultados y Conclusiones*) incluye los tres últimos capítulos. La distribución de los capítulos es la que se presenta a continuación:

El primer capítulo, *Introducción*, plantea los antecedentes, la descripción de la problemática abordada y su motivación, así como los objetivos de la tesis, la metodología seguida y las principales aportaciones, tal y como se ha expuesto en los apartados anteriores.

El segundo capítulo, *Redes Neuronales Artificiales*, contiene una descripción de las redes neuronales y la evolución a lo largo de la historia de los diferentes modelos generados y sus principales características.

En el tercer capítulo, *Series Temporales*, se enumeran los fundamentos teóricos de las series temporales con una elaborada descripción del análisis, descomposición y técnicas de modelado.

CAPÍTULO 1. INTRODUCCIÓN

En el cuarto capítulo, *Técnicas de Predicción de Series Temporales*, se repasa el estado del arte en la predicción de series temporales sobre técnicas clásicas y de redes neuronales con sus principales aplicaciones.

En el quinto capítulo, *Modelo BiLSTM-GCN*, se describe el modelo híbrido propuesto partiendo de la descripción de la arquitectura de las redes neuronales, y finalmente justificando las decisiones adoptadas a la hora de diseñar el modelo propuesto.

En el sexto capítulo, *Experimentación y Análisis de Resultados*, se recogen los experimentos realizados. Comenzando por la descripción y la base teórica de los datos recogidos, llevando a cabo un análisis del tipo de información y de las técnicas empleadas en la literatura para la predicción de estas series temporales, la descripción del tratamiento de los datos y los resultados obtenidos en el experimento, así como la comparativa de las métricas entre los distintos métodos empleados.

El séptimo capítulo, *Conclusiones*, detalla las principales conclusiones extraídas a partir de los experimentos realizados y los resultados obtenidos. También incluye las líneas de trabajo que se podrían desarrollar a futuro tomando como base la investigación que recoge esta tesis.

Por último, la bibliografía científica consultada durante el desarrollo de este trabajo es convenientemente referenciada al final del documento.

2 REDES NEURONALES ARTIFICIALES

Alan Turing, considerado el padre de la inteligencia artificial, fue el primero en estudiar el cerebro como una forma de ver el mundo de la computación, allá por los años 30 del siglo pasado. No obstante, el concepto de Red Neuronal Artificial (RNA) o *Artificial Neural Network* (ANN) en terminología inglesa, se ubica más adelante (en la década de los 50), cuando Rosenblatt concibió la primera idea del perceptrón, un modelo matemático semejante al funcionamiento de las neuronas cerebrales (Rosenblatt, 1958). Desde estas primeras investigaciones se desarrolló un creciente interés en las redes neuronales, surgiendo importantes avances de la comunidad científica.

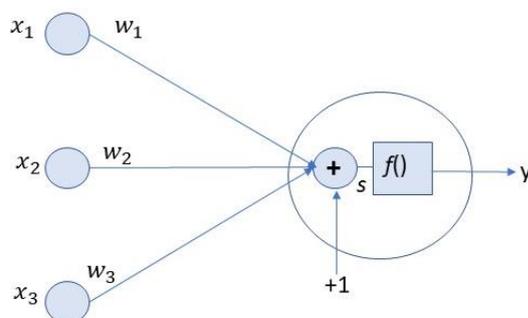
Una red neuronal artificial es un complejo sistema para el procesamiento de información inspirado en una red neuronal biológica (Palmer y Montaña, 1999). Una red neuronal se compone de una gran cantidad de neuronas, que son los elementos de procesamiento de la red, y se organizan en capas. Las neuronas se comunican con otras mediante uniones que les permiten comunicarse, estas uniones tienen un peso que contiene la información relevante para la toma de decisiones y la resolución de problemas.

McCulloch y Pitts (1943) desarrollaron un primer modelo sencillo de neurona artificial mediante circuitos eléctricos, como se muestra en la Ilustración 1, y lanzaron una teoría acerca de la forma de trabajar de las neuronas. Las entradas x_i son multiplicadas por los pesos sinápticos w_i y a la suma de todo ello se le aplica la función de activación no lineal sigmoide f de la siguiente manera:

$$y = f\left(\sum_{i=1}^N w_i x_i + w_0\right) \quad (2.1)$$

donde w_0 es la representación del sesgo mostrado en la Ilustración 1, y toma el valor +1.

Ilustración 1. Modelo de neurona artificial



No obstante, Donald Hebb (1949) fue el primero en explicar los procesos del aprendizaje desde un punto de vista psicológico. Su idea de que el aprendizaje ocurría cuando eran activados ciertos cambios en una neurona, sigue siendo el fundamento de la mayoría de las funciones de aprendizaje que pueden hallarse en una red neuronal artificial. Los trabajos de Hebb sentaron las bases de la Teoría de las Redes Neuronales.

Posteriormente, Rosenblatt comenzó el desarrollo del Perceptrón. Pese a ser la red neuronal más antigua, sigue utilizándose hoy en día como identificador de patrones ya que es capaz de generalizar (Rosenblatt, 1958). Es decir, posee la capacidad de reconocer patrones con los que no ha sido entrenado (pero que presentan cierta similitud con otros con los que sí se ha entrenado previamente). Poco después, Widrow y Hoff (1960) desarrollaron el modelo Adaline (*ADaptative LINear Elements*), basado en el concepto de neurona de McCulloch–Pitts. La diferencia entre Adaline y el perceptrón simple es que, en la fase de aprendizaje, los pesos se ajustan de acuerdo con la suma ponderada de las entradas. En el perceptrón simple, la red pasa a la función de activación y la salida de la función se usa para ajustar los pesos.

Pese a los avances producidos en las décadas anteriores, en el año 1969 se produce un hecho que paraliza una década el estudio de estas redes. En concreto, Minsky y Papert (1969) probaron matemáticamente que el perceptrón no permite resolver tareas no lineales. Esta incapacidad para resolver problemas que no son separables linealmente, suponía una limitación importante ya que impedía que la red aprendiera tareas relativamente sencillas, como por ejemplo resolver la función lógica XOR.

El trabajo desarrollado por Rumelhart, Hinton y Williams (1986) posibilitó un aumento del interés por las redes neuronales artificiales, aplicando una mejora al perceptrón simple de Rosenblatt. En concreto, en su estudio desarrollaron un algoritmo de retropropagación (*backpropagation*) para redes neuronales multicapa en la que se aprecia el potencial de este tipo de redes. De hecho, el Perceptrón Multicapa (*Multilayer Perceptrón*, MLP, en terminología inglesa) ha sido y, posiblemente siga siendo a día de hoy, el paradigma más extendido y utilizado. Con el teorema enunciado por Kolmogorov (1957) como base, lograron demostrar que los modelos de perceptrón multicapa con varias capas ocultas y un número no determinado de nodos puede considerarse un aproximador de funciones universal (Hornik et al., 1989).

A lo largo de las décadas de los 80 y 90 del siglo XX se comenzaron a desarrollar otros métodos, como las Máquinas de Vector Soporte (*Support Vector Machines*, SVM por sus siglas en inglés), introducidas por Kohonen (1982) o los bosques aleatorios (Random Forest, RF) descritos por Ho (1995). Estos algoritmos demostraron una alta eficiencia y resultados similares a los obtenidos por las redes neuronales artificiales desarrolladas hasta ese momento (Lecun y Bengio, 1995).

Más recientemente, Hinton y Salakhutdinov (2006) demostraron que era posible entrenar redes neuronales artificiales con gran cantidad de capas profundas, inicializando de forma adecuada los pesos en lugar de empleando valores aleatorios. Este proceso comenzaba por entrenar cada una de las capas de forma no supervisada y posteriormente se continuaba con el entrenamiento supervisado utilizando los pesos resultantes de las capas pre entrenadas como valores iniciales. Por otro lado, Glorot y Bengio (2010) propusieron un esquema de inicialización eficiente de pesos, comúnmente conocido como “*Inicialización de Xavier*”, con la capacidad de inicializar los pesos sin la necesidad de un entrenamiento no supervisado, y que se ha convertido en el estándar del aprendizaje profundo. Además, demostraron un gran impacto en el entrenamiento y mejora de la precisión con la elección de la función de activación no lineal. Este hecho propició una nueva línea de investigación centrada en encontrar funciones de activación adecuadas como la Unidad Lineal Rectificada (*Rectified Linear Unit*, ReLU) (Jarrett et al., 2009; Nair y Hinton, 2010; Glorot et al., 2011).

El campo de aplicación de las redes neuronales artificiales abarca en la actualidad la resolución de problemas de predicción de series temporales, la clasificación y el reconocimiento de patrones, el procesamiento de datos o la robótica, en distintos campos

entre los que se incluyen la agricultura, la biología, la medicina o la economía, entre otros (Pajares et al., 2010; Herrera et al., 2016; Le et al., 2019; Nazir et al., 2020; Namasudra et al., 2021), con resultados que mejoran los obtenidos por los modelos estadísticos clásicos (De Lillo y Meraviglia, 1998; Takahashi et al., 1999). Además, la capacidad de procesamiento paralelo permite a las redes neuronales el aprendizaje de relaciones entre las variables sin suposiciones previas.

A continuación, nos centraremos en una serie de modelos de red neuronal artificial que han venido utilizándose históricamente para la predicción de series temporales. El estudio de diversas tipologías de red como el perceptrón multicapa, las redes recurrentes, redes convolucionales y redes basadas en grafos, así como distintos modelos pertenecientes a las categorías anteriores, han sido el fundamento sobre el que se apoya el modelo híbrido propuesto en esta memoria de tesis, y que se detallará en concreto en el capítulo cinco.

2.1 PERCEPTRÓN MULTICAPA

Los modelos de perceptrón multicapa (MLP) se componen de al menos una capa de entrada, una capa oculta y una de salida con valores binarios de entrada y salida. Gracias al uso de algoritmos de retropropagación (Rumelhart et al., 1986), es posible realizar el entrenamiento de redes multicapa mediante un enfoque supervisado calculando el error en la capa de salida, realizando el ajuste de pesos en un proceso iterativo, para el abordaje del aprendizaje complejo de problemas. Este proceso se realiza mediante el aprendizaje supervisado, en el que se proporciona a la red tanto los datos de entrada como la salida esperada y la red se encarga de realizar la asociación por medio del aprendizaje.

El perceptrón multicapa es una red de tipo *feedforward* en la que las entradas alimentan la red propagándose hacia las salidas en una única dirección, con el objetivo de enviar los datos desde la capa de entrada a la de salida y atravesando todas las capas ocultas que existan en el modelo, que serán las encargadas del procesamiento de la información y de extraer los rasgos característicos.

Mediante algoritmos de aprendizaje automático (*Machine Learning*, ML) es posible predecir series temporales utilizando perceptrones multicapa en los que podemos llamar perceptrones a cada una de las neuronas conectadas que forman la red neuronal, que

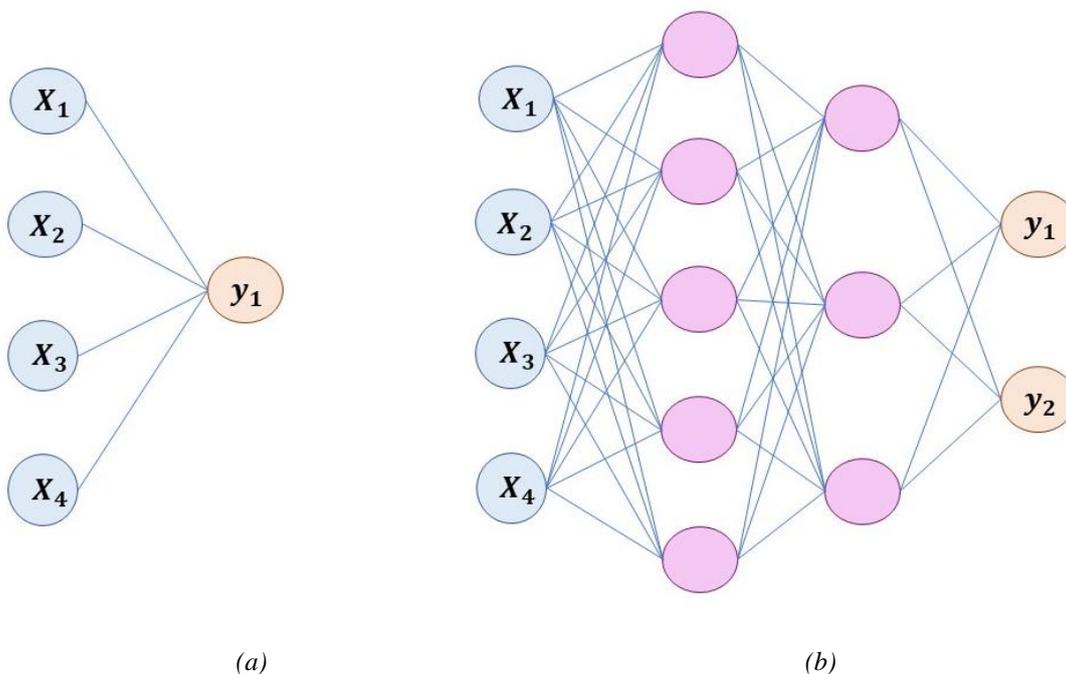
Palmer y Montaña (1999) definen como sistemas para el procesado de información con estructura y funcionamiento similar al de una red neuronal biológica.

Como se ha mencionado anteriormente, el modelo más simple de perceptrón multicapa consta de tres capas, que es un conjunto de nodos que aplican una transformación seguida por una activación no lineal en las entradas recibidas. Los pesos de cada nodo son diferentes, lo que hace posible aplicar una función diferente. Las transformaciones realizadas por cada uno de los nodos y los pesos de estas transformaciones son aprendidos a lo largo del entrenamiento.

La única capa que compone los modelos de una capa se llama de salida, y en ella cada neurona estará conectada con todas las neuronas de la capa anterior, pero no entre ellas.

Las capas que están completamente conectadas en los perceptrones multicapa se denominan *capa densa* y los datos son denotados como *capa de entrada*, aunque no se trate de una capa como tal (Wanchen, 2020). En la Ilustración 2 se observa la estructura de un perceptrón simple (por tanto, de una única capa y con un único nodo de salida) y de un modelo multicapa con estas interconectadas con los nodos de salida.

Ilustración 2. Modelos de perceptrón. (a) Perceptrón simple; (b) Perceptrón multicapa.



Las entradas de una neurona i en una capa l de un perceptrón de L capas, x_i^l , son multiplicados por los pesos w_{ij}^l , que definen la relación entre la neurona i de la capa $l-1$

y la neurona j de la capa l . La salida de una neurona x_j^{l+1} es una función sigmoide de la suma de sus entradas:

$$x_j^{l+1} = f\left(\sum_i w_{i,j}^l x_i^l\right) \quad (2.2)$$

Las funciones de activación no lineales restringen la salida a un dominio apropiado en los nodos de salida, mientras que el número de nodos en la capa de salida determinará la tarea de aprendizaje en particular. Si se trata de una tarea de regresión y clasificación binaria, la capa de salida estará formada por un solo nodo. Sin embargo, para problemas de aprendizaje más generales como clasificación de clases múltiples y lenguaje, el número de nodos de salida puede llegar a ser mucho mayor.

La correcta definición de la arquitectura en redes MLP es importante ya que un déficit de conexiones puede provocar que la red no sea capaz de encontrar solución al problema, mientras que el exceso puede causar un sobreajuste de la red con los datos de entrenamiento. Esto es habitual que suceda cuando se utiliza un gran número de capas y neuronas (Lins y Ludermir, 2005); el número adecuado de capas dependerá del problema planteado a resolver por la red neuronal (Eğrioğlu et al., 2008).

Una de las principales limitaciones de las redes de tipo MLP es que no tienen la capacidad de explorar la estructura de los datos en aplicaciones como el procesamiento del lenguaje natural y la predicción de series temporales, y el número de entradas y salidas son corregidas impidiendo que sea posible aplicarlas a problemas con entradas y salidas de tamaño variable como ocurre en el pronóstico de series temporales.

2.2 REDES NEURONALES RECURRENTES

Por su efectividad en gran número de aplicaciones prácticas las Redes Neuronales Recurrentes (*Recurrent Neural Networks*, RNN) han sido ampliamente estudiadas en investigaciones científicas (Pajares et al., 2021). Se trata de una red neuronal de aprendizaje automático supervisado con uno o más bucles de retroalimentación (Haykin, 1994), y basadas en el trabajo de Rumelhart et al. (1986). Un tipo especial de RNN serían las Redes de Hopfield (Hopfield, 1982), ampliamente utilizadas en múltiples aplicaciones (Rojas, 1996; Herrera et al., 2011).

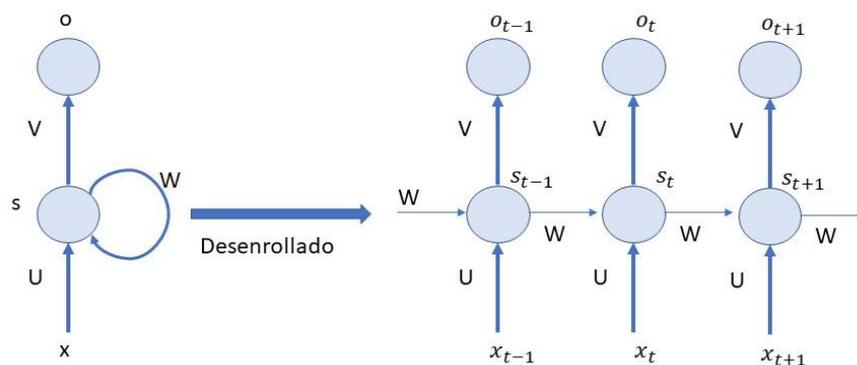
Las RNN se crearon como una variante de las ANN para el pronóstico de datos dependientes en el tiempo. Las redes tipo MLP no tienen la capacidad de tener en cuenta las posibles relaciones temporales existentes en los datos introducidos, mientras que las RNN son capaces de conectar cada uno de los pasos temporales con los anteriores para lograr modelizar la relación existente entre ellos (Elman, 1990), pudiendo clasificar datos secuenciales.

Las RNN observan cada dato de entrada y analizan su importancia relativa en el pronóstico, por lo que el aprendizaje no se basa únicamente en patrones de entrada y salida, sino que también es capaz de aprender los patrones internos de la secuencia, lo que hace las RNN las más utilizadas para el análisis y pronóstico de series temporales en distintos ámbitos de aplicación como mercados de valores, medicina, etc. (Kim and Chi, 2018).

A diferencia de otro tipo de redes neuronales, las RNN pueden almacenar un estado e incluso aprender información en ventanas de contexto de cualquier longitud, admitiendo secuencias temporales y existiendo un espacio entre la capa oculta en el momento actual, y la capa oculta en el momento siguiente.

Este tipo de redes están compuestas por un conjunto de unidades no lineales en donde al menos una conexión entre unidades forma un ciclo dirigido. Los bucles de retroalimentación son ciclos recurrentes a lo largo del tiempo como se observa en la Ilustración 3.

Ilustración 3. Red Neuronal Recurrente



El entrenamiento de una ANN se realiza mediante la introducción de los conjuntos de datos de entrada y datos esperados, con el objetivo de que la diferencia entre estos

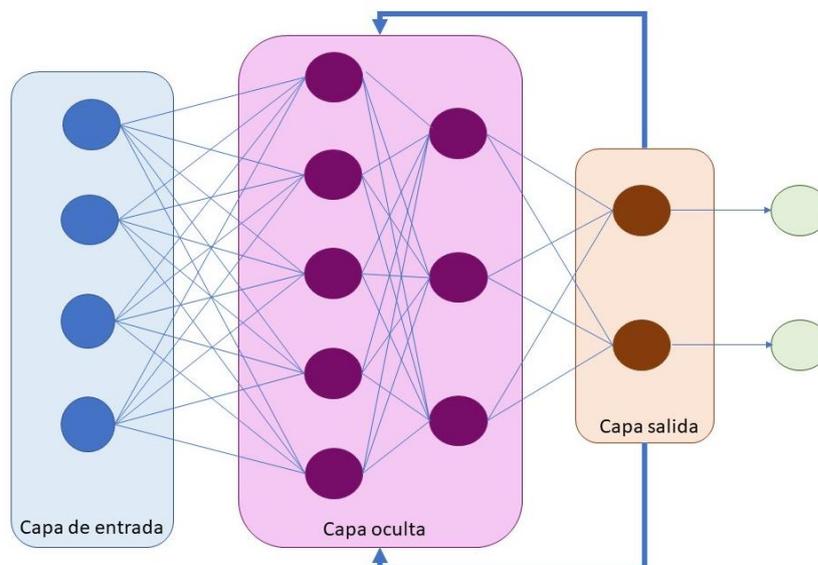
conjuntos sea mínima, logrando minimizar la pérdida de información mediante la optimización de los pesos de las neuronas.

2.2.1 ARQUITECTURA RNN

En lugar de usar un número fijo de vectores en la entrada como en las redes de tipo MLP, las RNN puede hacer uso de toda la información de entrada disponible hasta el momento actual para realizar predicciones. La cantidad de información capturada por estas redes depende de su estructura y del algoritmo utilizado para realizar el entrenamiento (Schuster and Paliwal, 1997).

Una RNN simple está compuesta por tres capas: capa de entrada, capas ocultas recurrentes y capa de salida como se observa en la Ilustración 4.

Ilustración 4. Arquitectura de Red Neuronal Recurrente



La capa de entrada tiene N unidades de entrada, estas entradas son una secuencia de vectores a través del tiempo t como $\{..., X_{t-1}, X_t, X_{t+1}, \dots\}$, en donde $X_t = (X_{t1}, X_{t2}, \dots, X_{tn})$. Las capas de entrada de una Red Neuronal Recurrente están conectadas a las unidades ocultas de la siguiente capa, en donde las conexiones están definidas con una matriz de pesos W_{IH} .

Las capas ocultas tienen M unidades ocultas $h_t = (h_{t1}, h_{t2}, \dots, h_{tm})$ que se encuentran conectadas entre sí por la temporalidad con conexiones recurrentes. La inicialización de

las unidades ocultas mediante elementos distintos de cero puede mejorar el rendimiento general y la estabilidad de la red (Sutskever et al., 2013).

El estado de memoria de la capa oculta se define como:

$$h_t = f_H(o_t) \quad (2.3)$$

Donde:

$$o_t = W_{IH}x_t + W_{HH}h_{t-1} + b_h \quad (2.4)$$

f_H es la función de activación de la capa oculta y b_h es el vector de polarización (*bias*) de las unidades ocultas. Las unidades ocultas están conectadas a la capa de salida con conexiones ponderadas W_{HO} . La capa de salida tiene P unidades $Y_t=(Y_1, Y_2, \dots, Y_p)$ que se calculan como:

$$y_t = f_o(W_{HO}h_t + b_o) \quad (2.5)$$

Donde f_o es la función de activación de la capa de salida y b_o el vector de polarización. Sutskever et al. (2011) definieron el estado oculto de una RNN como un conjunto de valores que resume toda la información necesaria sobre los estados pasados de la red durante muchos pasos de tiempo. Esta información integrada define el comportamiento de la red y permite hacer predicciones precisas en la capa de salida.

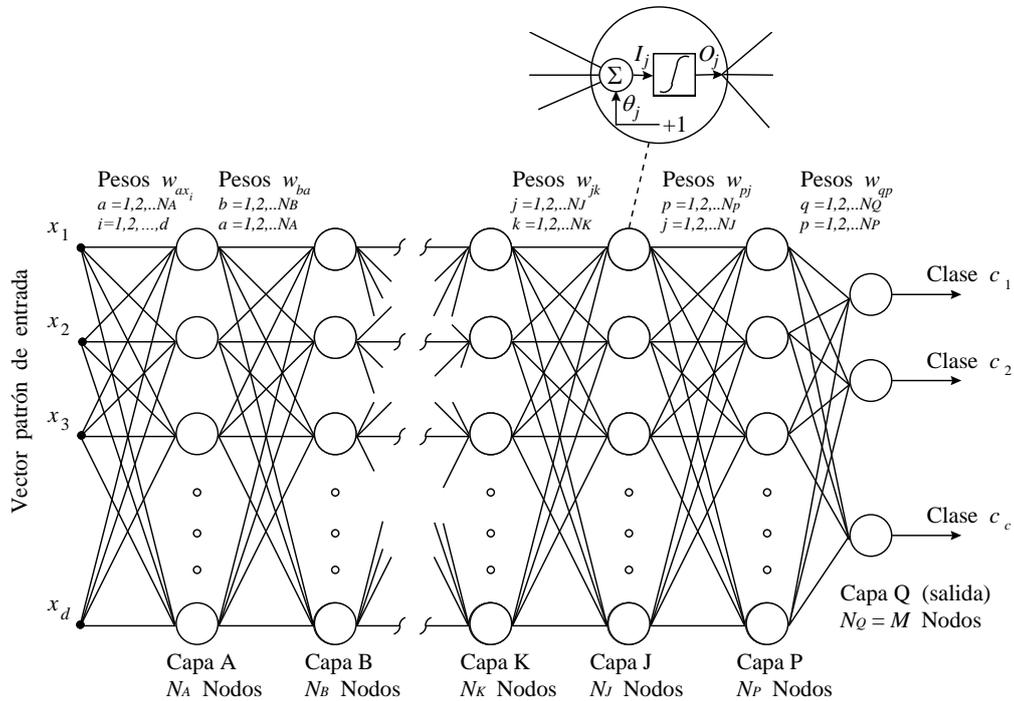
2.2.2 ALGORITMO BACKPROPAGATION

Uno de los objetivos de las redes neuronales es ajustar los pesos de cada neurona de forma que el error sea minimizado. El algoritmo *backpropagation* investigado por Rumelhart et al., (1986) indicará el grado de implicación de cada neurona sobre el error global cometido.

El cálculo se lleva a cabo desde la última capa del modelo, Las neuronas situadas en la capa oculta reciben una parte del error calculado en base a la aportación que cada neurona ha realizado a la salida, dotando a la red de la capacidad para modificar en las neuronas los pesos en las capas intermedias, adaptándolos para llevar a cabo el aprendizaje de las relaciones entre los conjuntos de entrada y las salidas obtenidas. La Ilustración 5 muestra

la arquitectura básica de un modelo de red neuronal con retropropagación (*backpropagation*).

Ilustración 5. Modelo de la red neuronal multicapa retropropagación (Pajares et al., 2021)



Se trata de un proceso iterativo que se realiza en cada capa hasta que cada una de las neuronas hayan recibido una señal de error que especifique cuál ha sido su aportación al error total a partir del cálculo de la función de coste de cada una de las variables, mediante sus derivadas parciales. Se realizará la actualización de los pesos de cada una de las relaciones de las neuronas tratando de que la red sea capaz de modelar las relaciones de los datos de entrenamiento (García Martínez et al., 2003).

La primera fase del algoritmo *backpropagation* consta de una propagación hacia delante, que en este proceso es iniciada cuando se introduce la información en el modelo a través de la capa de entrada. Cada una de las neuronas de esta capa se corresponderá con cada elemento del vector introducido como entrada y se realizará el cálculo del valor de los pesos en esta capa. Posteriormente el resto de las capas realizarán el mismo proceso en función de la entrada recibida en cada una de ellas. Dado un patrón p de entrada $X_p = X_{p1}, \dots, X_{pi}, \dots, X_{pN}$ es transmitido a través de la asignación de pesos W_{ij} desde la capa de la primera capa a la capa oculta.

La entrada recibida por una neurona j en una capa oculta se define como:

$$I_j^P = \sum_{i=1}^N W_{ij} X_i^P + \theta_j \quad (2.6)$$

En donde θ_j Representa valor de una neurona, considerado como el peso de una neurona ficticia cuyo valor en la salida es 1. La salida de la neurona oculta j , O_j^P es obtenida a partir de aplicar una función f sobre la entrada:

$$O_j^P = f(I_j^P) \quad (2.7)$$

A partir de la expresión anterior se puede definir la medida de error como:

$$E = \sum_{p=1}^P E^p \quad (2.8)$$

La técnica del descenso del gradiente supone la base del algoritmo *backpropagation*. E^p es la representación del total de los pesos contenidos en la red, su gradiente se denota como un vector resultado de la derivada parcial de E^p respecto a cada uno de esos pesos. A continuación, este gradiente tomará la dirección que determine el incremento del error de forma más rápida, mientras que el decremento más rápido se encontrará en la dirección opuesta.

La modificación de los pesos se lleva a cabo una vez finalizados todos los patrones de entrenamiento, lo que se denomina aprendizaje por lotes (*batch*, en terminología inglesa).

Este proceso de iteraciones sobre los parámetros proporcionales al valor negativo del gradiente en el punto actual es conocido como descenso por gradiente. El principal problema de esta técnica reside en la posibilidad de que el gradiente adquiera valores muy pequeños y que esto impida el cambio de valor el peso y terminar por interrumpir el entrenamiento de la red; este problema fue detectado por Hochreiter (1991).

2.2.3 LONG SHORT TERM MEMORY

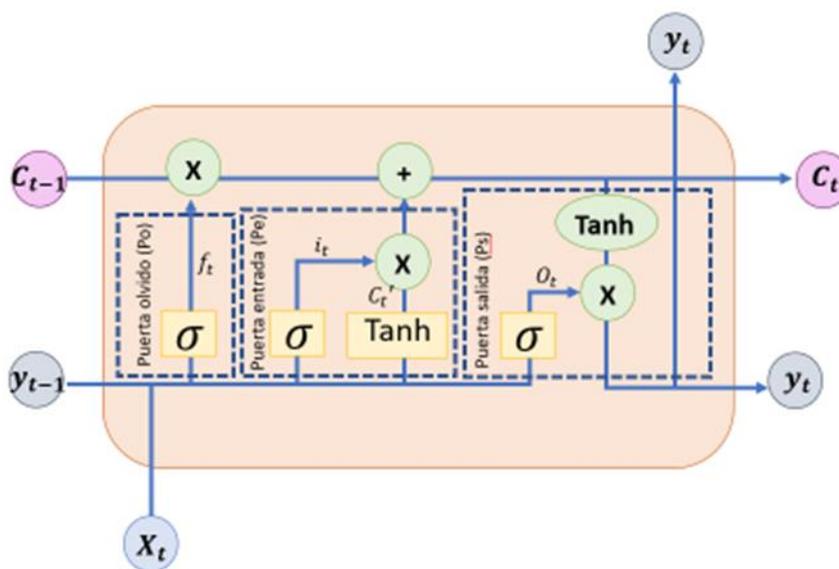
Las redes *Long-Short Term Memory* (LSTM) estudiadas por Hochreiter y Schmidhuber (1997), son un tipo de RNN cuya principal característica es que la información persiste en la red mediante la introducción de bucles en el diagrama de la red, por lo que adquieren la capacidad para recordar estados anteriores y usar esta información para decidir cuál

será el siguiente estado. Modelan dependencias temporales en horizontes más amplios sin olvidar los patrones a corto plazo y son capaces de resolver el problema del descenso del gradiente.

El impacto de las redes LSTM ha sido notable en diversos ámbitos de aplicación como el modelado del lenguaje, transcripción de voz a texto, traducción automática, etc. (Lin y Tegmark, 2016). Por ejemplo, Zhao et al. (2017b) describen cómo las redes LSTM tratan la capa oculta como una unidad de memoria, por lo que puede hacer frente a la correlación dentro de las series temporales tanto a corto como a largo plazo.

Las celdas de memoria se encuentran en el centro de la unidad. La Ilustración 6 refleja cómo la entrada son los datos conocidos y la salida es el resultado del pronóstico. Existen tres puertas en la unidad de memoria: puerta de entrada (*input gate*), puerta de olvido (*forget gate*) y puerta de salida (*output gate*).

Ilustración 6. Estructura de una celda de una red LSTM



La celda de memoria está representada en la Ilustración por la entrada C_{t-1} y la salida C_t que recorre de forma horizontal la celda en la parte superior y almacenará la información durante largos periodos de tiempo.

La capacidad de almacenar u olvidar viene regulada por las puertas que conforman la celda, permitiendo añadir o eliminar información a la memoria en un momento dado. Están compuestas por una capa con una función sigmoide como activación y una operación de suma o multiplicación. Un resultado de 0 en la operación sigmoide

implicaría no dejar pasar la información. Una celda LSTM tiene tres de estas puertas para controlar el estado de la memoria.

La puerta del olvido decidirá qué información debe permanecer y cuál debe ser olvidada. Esto se logra gracias a la función sigmoide, la cual tiene un dominio de 0 a 1 mostrando la relevancia de la información en ese rango. En esta puerta se concatenan el estado oculto de la celda anterior con la nueva entrada y la red decidirá si el estado de la memoria se alterará eliminando algún elemento de la memoria o se conserva.

$$f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.9)$$

La puerta de entrada actualizará el estado oculto de la celda, para ello la nueva entrada es añadida al estado oculto de un tiempo anterior. La cantidad de información a almacenar se determina con la función sigmoide transformando los valores entre 0 y 1. Una pequeña red con función \tanh (tangente hiperbólica) que crea un vector de valores nuevos candidatos C_t que se añadirían al estado. Los dos resultados se combinan para añadirse al estado.

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.10)$$

$$C_t' = \tanh(w_c \cdot [h_{t-1}, x_t] + b_c) \quad (2.11)$$

En ese momento el estado de la memoria cambiará de C_{t-1} a C_t y el estado anterior será multiplicado por f_t y sumando $i_t * C_t'$, actualizando la memoria con los nuevos valores.

$$C_t = f_t * C_{t-1} + i_t * C_t' \quad (2.12)$$

La puerta de salida es la encargada de decidir cuál será el estado oculto de la celda en el estado siguiente haciendo uso de las funciones *sigmoide* y *tanh*. La función *tanh* comprime los valores entre -1 y 1 con el objetivo de evitar que los valores aumenten o disminuyan en exceso y así evitar los problemas de desvanecimiento de gradiente durante el entrenamiento.

$$o_t = \sigma(w_o [h_{t-1}, x_t] + b_o) \quad (2.13)$$

$$h_t = o_t * \tanh(C_t) \quad (2.14)$$

2.2.4 BIDIRECCIONAL LSTM

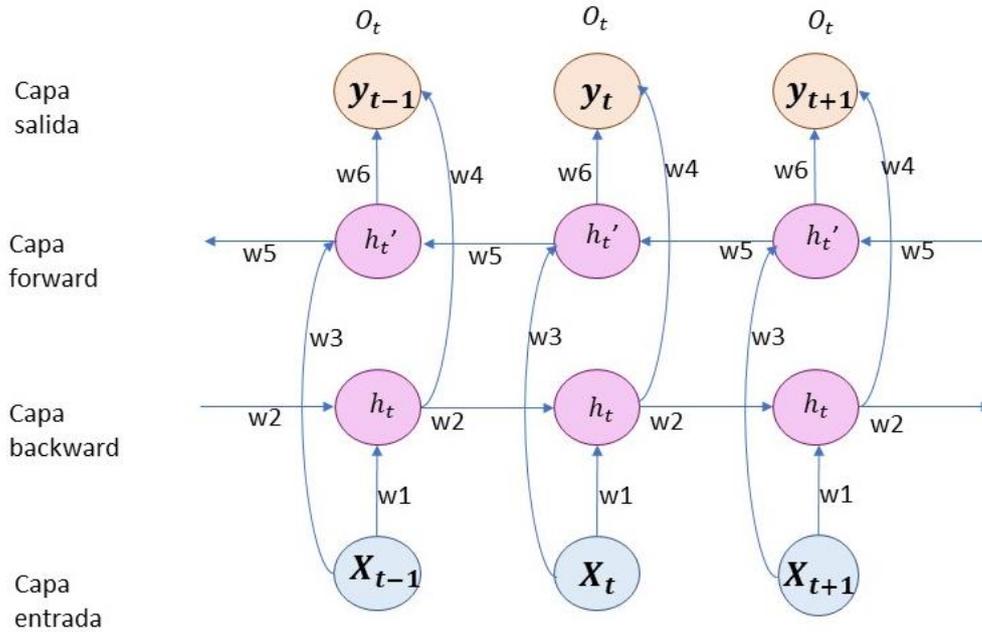
Las redes LSTM bidireccionales, conocidas como BiLSTM (del inglés, *Bidirectional LSTM*), son una variante de las arquitecturas simples LSTM que permiten un entrenamiento adicional gracias a que los datos atraviesan dos veces la entrada, en primer lugar, de izquierda a derecha y de derecha a izquierda. En realidad, son dos redes, una hacia delante capturando información pasada y otra hacia atrás capturando información futura, tal y como describen Pajares et al. (2021).

En la investigación llevada a cabo por Siami-Namini et al. (2019), trataron de explorar hasta qué punto es beneficioso añadir capas adicionales para el entrenamiento del modelo, concluyendo que el modelo BiLSTM ofrece mejores resultados que las predicciones de una red LSTM. Estos resultados son coincidentes con las conclusiones obtenidas en el estudio llevado a cabo por Kim y Moon (2019) en el que, a través de una red neuronal, se realizaban predicciones con los datos procedentes de una serie temporal sobre inversiones. Yang y Wang (2022) lograron unos resultados similares realizando una comparación de las predicciones de series temporales financieras arrojadas por una red BiLSTM, una máquina de vector de regresión (SVR) y ARIMA.

Las redes BiLSTM utilizan los datos preprocesados como entrada, y éstos pasan a través de una capa LSTM hacia adelante y otra capa LSTM hacia atrás, generando el resultado de la predicción. BiLSTM tiene la capacidad de resolver la dependencia a largo plazo de las RNN y LSTM, a partir de la combinación de las dos direcciones diferentes de los datos. En estos modelos existe una capa hacia adelante (*forward*), en la que el cálculo se realiza desde el momento 1 hasta el momento t y de este se obtiene y guarda la salida de la capa oculta hacia adelante en cada momento; y una capa hacia atrás (*backward*) en la que el cálculo se invierte a lo largo del tiempo t al tiempo 1 para obtener la salida de la capa oculta.

Como se muestra en la Ilustración 7, las entradas van a la capa oculta *forward* y a la capa oculta *backward* (w_1 y w_3) y cada capa oculta lleva la entrada a sí misma (w_2 y w_5). Tanto la capa *forward* como la capa *backward* envían sus salidas a la capa de salida (w_4 y w_6). Por último, por cada tiempo t la salida final es obtenida mediante la combinación de las salidas de las capas *forward* y *backward* como una red LSTM bidireccional.

Ilustración 7. Estructura de una red BiLSTM



Las fórmulas que describen la estructura anterior son las siguientes:

$$h_t = f(w_1 x_t + w_2 h_{t-1}) \quad (2.15)$$

$$h'_t = f(w_3 x_t + w_5 h_{t+1}) \quad (2.16)$$

$$o_t = g(w_4 h'_t + w_6 h_t) \quad (2.17)$$

2.3 REDES CONVOLUCIONALES

Las Redes Neuronales Convolucionales (en inglés, *Convolutional Neural Networks*, CNN) son un tipo de red creada en su origen para tareas de visión artificial, y que son capaces de procesar imágenes de dimensiones variables. Poseen una topología basada en rejilla para el procesamiento de datos (sonidos, imágenes, vídeos, etc.), sobre los que se aplican operaciones basadas en rejillas de dimensiones 1-D, 2-D, 3-D y superiores, configuradas la mayor parte de las veces como tensores (Pajares et al., 2021).

Este tipo de redes aprenden a extraer características significativas de los datos mediante una operación convolucional: una operación lineal altamente especializada. Se trata de un procesamiento mediante el cual se crean mapas de características y tiene como objetivo extraer patrones repetidos. La principal diferencia con los métodos tradicionales de

extracción es que las CNN no necesitan extraer las características manualmente. Este proceso proporciona a las CNN una característica denominada invarianza de distorsión; las características se extraen independientemente de dónde se encuentren en los datos, haciendo que este tipo de redes sean idóneas para el tratamiento de datos unidimensionales como las series temporales.

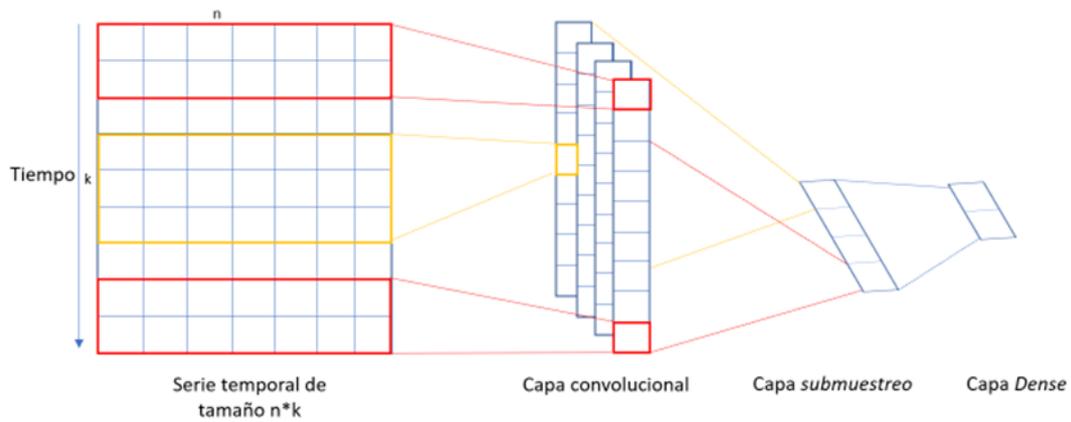
Las CNN se componen de una serie de capas convolucionales, capas de agrupación (*pooling*) y capas totalmente conectadas. A diferencia de las MLP, cada nodo se encuentra conectado a una región de la entrada, conocida como campo receptivo. Las neuronas en las mismas capas comparten la misma matriz de peso convolucional. Borovykh et al. (2019) investigaron sobre cómo son estas propiedades las que permiten que las CNN tengan una cantidad de parámetros entrenables menor en comparación con una RNN, convirtiéndola en una red más eficiente. Otras investigaciones, como la llevada a cabo por Yang et al. (2015), proponen las CNN como extractores de características solos o junto con bloques recurrentes para proporcionar pronósticos.

Una red CNN es una ANN *feedforward* que interpreta las entradas como matrices 2-D. A diferencia de las redes totalmente conectadas como las MLP, la ubicación de los datos en el vector de entrada es importante, por esta razón los datos adyacentes dentro de la matriz deben ser elegidos cuidadosamente. Por esta razón las CNN son especialmente utilizadas en problemas de clasificación de imágenes, ya que los píxeles vecinos están relacionados entre sí en ambas direcciones (LeCun et al., 1989).

Las CNN contienen unos núcleos con la capacidad de representar receptores con la habilidad de reaccionar ante múltiples características; las señales de activación que superen un rango establecido serán transmitidas a la neurona de la siguiente capa gracias a la función de activación.

La Ilustración 8 muestra cómo las redes CNN constan de dos tipos de capas, una convolucional y otra de submuestreo, y está formada por sucesivas capas de estos tipos. En la capa de convolución se aplica la operación de convolución y los resultados son transferidos a la siguiente capa, para posteriormente en la capa de submuestreo reducir los parámetros y el tamaño espacial de la representación. La última capa de convolución está conectada a una capa totalmente conectada, finalizando con la capa de clasificación en la que la toma de decisiones se realiza como en los clasificadores tradicionales.

Ilustración 8. Estructura de una Red Convolutiva



A diferencia de las redes neuronales tradicionales, las neuronas de una capa no están directamente relacionadas con las de la capa anterior, sino con un número reducido de estas neuronas, como consecuencia es necesaria la reducción de parámetros y la aceleración de la convergencia. Es posible que un conjunto de neuronas tenga los mismos pesos, lo que representa un menor número de parámetros.

La convolución es un paso imprescindible para la extracción de características, en esta fase se llevan a cabo cálculos de multiplicaciones y sumas entre la capa original y cada uno de los n filtros (o núcleos) responsables de la generación de un mapa de características. Los resultados de la convolución se pueden llamar mapas de características, estas características coinciden con la localización en la imagen original del filtro. Al configurar un núcleo de convolución con un tamaño determinado puede derivar en la pérdida de información.

Las investigaciones más recientes han demostrado que el uso de CNN para la clasificación de series temporales tiene ventajas importantes sobre otros métodos, como la resistencia al ruido y la extracción profunda de características muy informativas que son independientes del tiempo (Zhao et al., 2017a). En estos casos, los núcleos de convolución tienen el mismo ancho que la serie temporal, y la longitud es de tamaño variable. Por esta razón el *kernel* o núcleo se mueve en una dirección desde el comienzo de la serie temporal hasta el final realizando una convolución.

2.4 REDES NEURONALES DE GRAFOS

Los grafos permiten reflejar estructuras de datos como un conjunto de nodos y aristas, que representan objetos y sus relaciones respectivamente. En los últimos años se ha desarrollado ampliamente la literatura existente sobre Redes Neuronales de Grafos (*Graph Neural Network*, GNN por sus siglas en inglés), esto es debido al gran poder de los grafos para expresar relaciones que permiten denotar un gran número de sistemas a través de diversas áreas (Zhou et al., 2020). Entre las aplicaciones de este tipo de redes se encuentran las redes sociales (Han et al., 2020) y la física (Sánchez-González et al., 2018), centrándose en tareas como la clasificación de nodos, la predicción de enlaces y la agrupación de éstos.

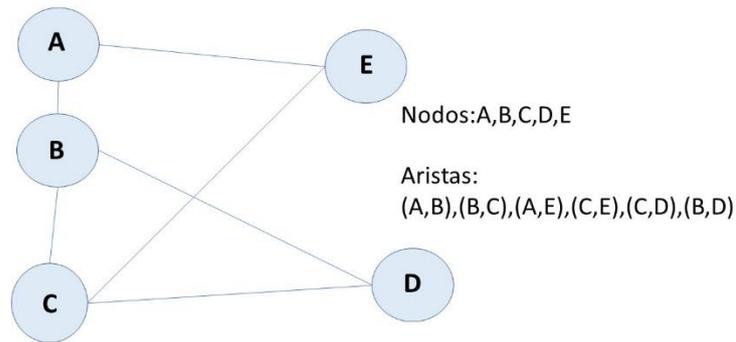
Gori et al. (2005) realizaron las primeras investigaciones sobre GNN, siendo posteriormente desarrolladas por Scarselli et al. (2008) y Gallicchio et al. (2010), mientras que Sperduti y Starita (1997) aplicaron por primera vez los grafos acíclicos proporcionando los primeros estudios sobre GNN.

Un grafo es representado como $G = (V, E)$ en donde V es el conjunto de vértices o nodos y E es el conjunto de aristas. Los nodos se denotan como $v_i \in V$ y las aristas desde v_i a v_j son denotados como $e_{ij} = (v_i, v_j) \in E$. Los vecinos de un nodo v son definidos como $N(v) = \{u \in V | (v, u) \in E\}$.

La matriz de adyacencia A es una matriz de $n \times n$ elementos con $A_{ij} = 1$ si $e_{ij} \in E$; y $A_{ij} = 0$ y $e_{ij} \notin E$. Un grafo puede tener atributos de nodo, donde $X \in R^{n \times d}$ es una matriz de características del nodo con $x_v \in R^d$ representando el vector de características del nodo v . Un grafo también puede disponer de atributos de arista X^e en donde $X^e \in R^{m \times c}$ es una matriz de características de aristas con $x_{v,u}^e \in R^c$ que representa el vector de características de una arista (v, u) . La Ilustración 9 representa la estructura de nodos y aristas de un grafo.

Wu et al. (2021) describían cómo, en las primeras representaciones de las GNN, los modelos aprenden la representación de un nodo objetivo mediante la propagación de la información contenida en los nodos vecinos de forma iterativa hasta que un punto fijo estable es alcanzado.

Ilustración 9. Estructura de un grafo



Para los desarrolladores de ANN e investigadores de diversas áreas como finanzas, economía y energía, el estudio y modelización de las series temporales se ha convertido en un tema central de investigación. Por ejemplo, Wu et al. (2020) propusieron un diseño específico para series temporales dentro de un marco general de GNN, de forma que se realizara de forma automática la extracción de relaciones entre las variables implicadas a través de un módulo de aprendizaje de grafos. Deng y Hooi (2021) llevaron a cabo un estudio con el objetivo de realizar la predicción del comportamiento esperado de las series temporales para la detección de eventos anómalos. A partir de un enfoque combinando GNN con pesos de atención proporcionaban explicación a las anomalías detectadas, y lograban demostrar que este método proporcionaba mayor precisión que los enfoques de referencia y tenía la capacidad de capturar con mayor exactitud las correlaciones entre las variables. En la investigación realizada por Wang et al. (2021), reflejaban cómo los componentes y las relaciones de los datos financieros se pueden representar como grafos, ya que permiten reflejar tanto características individuales como relaciones más complejas. La complejidad y la volatilidad asociadas con las series de tiempo económicas a menudo dan como resultado un grafo heterogéneo o variable, lo cual es un desafío para las redes neuronales de grafos.

2.4.1 REDES CONVOLUCIONALES DE GRAFOS

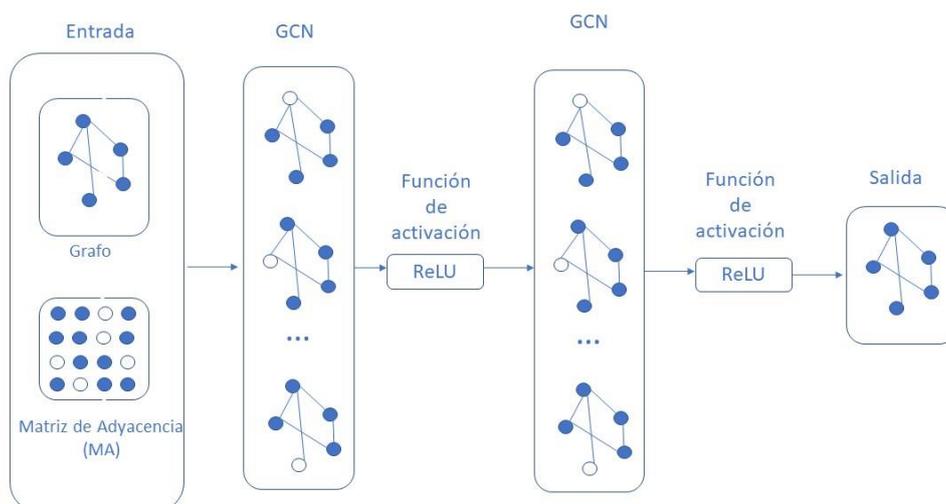
La principal idea en los modelos GNN es que la información de la señal del nodo de entrada puede ser combinada con la propagación de la información a los nodos vecinos para transmitir la información mejor el estado oculto de la entrada original. El concepto general de las GNN es intercambiar mensajes constantemente con sus vecinos hasta que

logran alcanzar un equilibrio estable, de forma similar a lo que ocurre en las RNN, en donde estos pesos se comparten en cada paso recurrente. Por el contrario, las Redes Convolucionales de Grafos (*Graph Convolutional Networks*, GCN por sus siglas en inglés) no comparten los pesos entre las capas ocultas.

Las GCN pueden considerarse una generalización de las GNN para datos estructurados en grafos. En el módulo de convolución de grafos la información de dependencia es obtenida de los datos de entrada a través de la matriz de adyacencia. Cada nodo de la matriz es combinado con información del nodo dependiente para obtener la información de dependencia de cada nodo y los nodos relacionados.

Con el objeto de aplicar formulaciones matemáticas a las GCN, Jiang y Luo (2022) introducían las siguientes notaciones. D se define como la matriz de grados, en la que cada uno de los elementos es $D_{ii} = ||N(v_i)||$. $L = D - A$ describe la matriz Laplaciana de un grafo no dirigido, siendo A la matriz de adyacencia. $\tilde{L} = I_N - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ es como se denota la matriz Laplaciana normalizada, en donde I_N es la matriz identidad con tamaño N . La matriz de características del nodo de un grafo, sin ser considerado el índice de paso de tiempo, puede ser simplificada como $X \in R^{N \times d}$ en donde N es el número de nodos y d la dimensión del vector de características del nodo. La estructura de una GCN puede verse representada en la Ilustración 10.

Ilustración 10. Estructura de Red Convolutiva de Grafos



Es posible categorizar las GCN en *Redes Convolucionales de Grafos Espectrales* y en *Redes Convolucionales de Grafos Espaciales*.

2.4.2 REDES CONVOLUCIONALES DE GRAFOS ESPECTRALES

Bruna et al. (2013) propusieron la primera red convolucional de grafos espectrales notable. Inspirada en la CNN, este modelo profundo basado en grafos contiene varias capas convolucionales espectrales.

Las GCN espectrales implican la descomposición propia de la matriz Laplaciana. Esta descomposición permitirá comprender la estructura subyacente del grafo con la que pueden ser identificados grupos y subgrupos del grafo. Se trata de una matriz de adyacencia A normalizada de una manera especial, mientras que la descomposición propia consiste en una forma de encontrar esos componentes que componen el grafo.

Las GCN espectrales pueden ser definidas formalmente como:

$$X^{(l+1)} = V(V^t X^{(l)} \odot V^t W_{spectral}^{(l)}) \quad (2.18)$$

En donde V son los vectores propios de la matriz Laplaciana L , que pueden ser hallados mediante la fórmula $L = V \Lambda V^t$, siendo Λ los elementos propios de L y $W_{spectral}$ los filtros a aplicar, cuya dimensión depende del número de nodos N del grafo.

Es posible representar $W_{spectral}$ como la suma de K funciones predefinidas, en lugar de aprender N valores de W , se aprenderán K coeficientes α de la siguiente suma:

$$W_{spectral}^{(l)} \approx \sum_{k=1}^k \alpha_k f_k \quad (2.19)$$

La dimensionalidad de f_k depende del número de nodos N , estas funciones son fijas por lo que lo aprendido son los coeficientes α , de modo que $W_{spectral}$ ya no depende de N .

Las GCN estudiadas por Kipf y Welling (2017) son una aproximación de primer orden, esto significa que la métrica utilizada para determinar la similitud entre dos nodos se basa en los vecinos inmediatos del nodo, a diferencia de la CNN espectral de Chebyshev (ChebNet) investigado por Defferrard et al. (2016), la cual consiste en una aproximación del filtro mediante los polinomios de Chebyshev de la matriz diagonal de valores propios. Con el objetivo de evitar un sobreajuste de la red, en las GCN se utiliza $K = 1$. Esto evita

calcular la descomposición propia y los filtros ya no están asociados a los vectores propios.

La operación de convolución $*G$ de las GCN se formaliza como:

$$X_{*G} = W \left(I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \right) X \quad (2.20)$$

donde W representa una matriz con los parámetros del modelo. Para evitar el problema de explosión de gradiente las operaciones de convolución se desarrollan de la siguiente manera.

$$X_{*G} = W (\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}) X \quad (2.21)$$

En donde $\tilde{A} = A + I_N$ y $D_{ii} = \sum_j \tilde{A}_{ij}$.

2.4.3 REDES NEURONALES CONVOLUCIONALES DE GRAFOS ESPACIALES

El enfoque alternativo consiste en las GCN basadas en el espacio, en las que las convoluciones del grafo son definidas mediante la propagación de la información. El mayor reto de este tipo de enfoques es la definición de la operación de convolución con los vecinos, manteniendo la invarianza local de las CNN.

Danel et al. (2020) describían en su trabajo cómo la convolución espacial trabaja sobre los vecinos locales de los nodos y comprende las propiedades de un nodo en función de sus k vecinos. La principal diferencia con la convolución espectral es que resulta más simple realizar el cálculo de las convoluciones y producen resultados prometedores en las tareas de clasificación de grafos.

La premisa es que cada nodo v_i es identificado por sus coordenadas $p_i \in \mathbb{R}^t$. Las convoluciones de grafos espaciales son definidas como:

$$\bar{h}_i(U, b) = \sum_{j \in N_i} \text{ReLU}(U^T(p_j - p_i) + b) \odot h_j \quad (2.22)$$

En donde $U \in \mathbb{R}^{t \times d}$, $b \in \mathbb{R}^d$ son parámetros entrenables, d es la dimensión de h_j y \odot es la multiplicación por elementos. El par (U, b) realiza las funciones de filtro convolucional que opera en los vecinos de v_i . Las posiciones relativas en los vecinos son transformadas utilizando una operación lineal combinada con una función ReLU no lineal. El valor servirá para obtener los pesos de los vectores de características h_j de un vecindario.

No obstante, Balcilar et al. (2020) han propuesto recientemente un enfoque en el que tratan de aproximar ambos modelos (espectral y espacial), permitiendo disminuir el número de parámetros entrenables.

En los últimos años las GCN y otras variantes surgidas con posterioridad han logrado resultados prometedores en distintas áreas de aplicación como redes sociales (Chen et al., 2018), procesamiento del lenguaje natural (Yao et al., 2019; Zhang et al., 2018) y visión artificial (Keskes y Noumeir, 2021; Cao et al, 2022).

3 SERIES TEMPORALES

En este capítulo, en primer lugar, se realizará una descripción en profundidad de las series temporales, estableciendo los fundamentos teóricos necesarios sobre las características de las series para su análisis, que será descrito en segundo lugar.

3.1 DESCRIPCIÓN DE LAS SERIES TEMPORALES

Una serie temporal se define como una secuencia de N observaciones equidistantes y con un orden cronológico, sobre una o múltiples características, lo que es denominado como series univariantes o multivariantes por cada unidad observable en distintos momentos (Hamilton, 2020).

Las series temporales univariantes pueden representarse como:

$$y_1, y_2, \dots, y_N; (y_t)_{t=1}^N; (y_t: t = 1, \dots, N) \quad (3.1)$$

donde y_t es una observación t ($1 \leq t \leq N$) de la serie y N es el total de observaciones que contiene de la serie temporal. Las N observaciones y_1, y_2, \dots, y_N pueden ser reflejadas en un vector columna $y=[y_1, y_2, \dots, y_N]'$ de tamaño $N \times 1$.

En cuanto a las series temporales multivariantes, estas pueden representarse como:

$$y_1, y_2, \dots, y_N; (y_t)_{t=1}^N; (y_t: t = 1, \dots, N) \quad (3.2)$$

donde $y_t = [y_{t1}, y_{t2}, \dots, y_{tM}]'$ ($M \geq 2$) es la observación t ($1 \leq t \leq N$) de la serie y N el número de elementos de la serie. Las observaciones pueden representarse en una matriz Y de tamaño $N \times M$:

$$Y \equiv \begin{bmatrix} y'_1 \\ y'_2 \\ \dots \\ y'_N \end{bmatrix} \equiv \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1M} \\ y_{21} & y_{22} & \dots & y_{2M} \\ \dots & \dots & \dots & \dots \\ y_{N1} & y_{N2} & \dots & y_{NM} \end{bmatrix}$$

En donde y_{sj} es la observación número s para $1 \leq s \leq N$ sobre la variable número j siendo $1 \leq j \leq M$, para constante en todo el tiempo t .

En el análisis clásico de las series temporales se consideran cuatro componentes (Lansangan y Barrios, 2009):

- *Tendencia* (T): Comportamiento regular de la serie a largo plazo.
- *Componente estacional* (E): Oscilaciones a corto plazo de un periodo de tiempo regular.
- *Componente aleatoria* (I): No se corresponde con los patrones de comportamiento, es la consecuencia de elementos aleatorios que suceden de forma aislada a lo largo de la serie temporal.
- *Variaciones cíclicas* (C): Movimientos a medio plazo en torno a la tendencia; puede presentar cierta regularidad.

De estos cuatro componentes la tendencia, la variación cíclica y el componente estacional son considerados determinísticos, mientras que la componente aleatoria no. Una serie de tiempo en función de sus componentes puede ser denotada como:

$$X_t = T_t + E_t + C_t + I_t \quad (3.3)$$

donde T_t representa la tendencia, E_t la componente estacional e I_t la componente aleatoria.

A su vez las series temporales pueden ser clasificadas como *estacionarias* o *no estacionarias*. Una serie temporal es considerada estacionaria cuando permanece estable. Esto quiere decir que su media E y su varianza V son constantes a lo largo de todo el periodo de tiempo t , y el resultado de la covarianza Y entre dos momentos de tiempo t depende únicamente de la distancia entre ellos. Se considerará que una serie temporal no

es estacionaria cuando la media E o la varianza V se modifican a lo largo del tiempo (Maçaira et al., 2018).

$$E(X_t) = E(X_{t+k}) = \mu \quad (3.4)$$

$$V(X_t) = V(X_{t+k}) = \sigma^2 \quad (3.5)$$

$$Y_k = E[(X_t - \mu)(X_{t+k} - \mu)] \quad (3.6)$$

En donde la covarianza de X_t y X_{t+k} está representada por Y_k .

A continuación, se describen con más detalle el proceso estocástico, la tendencia y la componente estacional, por su importancia dentro del análisis de series temporales.

3.1.1 PROCESO ESTOCÁSTICO

Un proceso estocástico es definido como una secuencia de variables aleatorias asociadas a distintos instantes en el tiempo. Cada una de estas variables es un proceso estocástico y se corresponde a un índice de tiempo dado t con su propia distribución de probabilidad (Watkins, 2019).

Un proceso estocástico Y_t es descrito mediante la *esperanza matemática*, la *varianza* y la *autocovarianza* y los *coeficientes de autocorrelación*. En concreto, los coeficientes de autocorrelación son determinados por los coeficientes de correlación entre pares de variables:

$$\rho_{k,t+k} = \frac{Y_{k,y+k}}{\sqrt{Var(Y_t)Var(Y_{t+k})}}; t \in \mathbb{N} \quad (3.7)$$

Un proceso de *ruido blanco* es considerado como una serie de variables aleatorias en un proceso estocástico idénticas y no correlacionadas por lo que su media es cero y la varianza es constante. El ruido blanco es denotado por ε_t .

$$\varepsilon_t \sim (0, \sigma^2); cov(\varepsilon_{t_i}, \varepsilon_{t_j}) = 0; \forall t_i \neq t_j \quad (3.8)$$

El camino aleatorio o *random walk* es aquel proceso por el que una serie temporal X_t resulta en ruido blanco al aplicársele la primera diferencia.

3.1.2 ESTUDIO DE LA TENDENCIA

El análisis del movimiento a largo plazo de una serie temporal X_t debe realizarse cuando esta consta de un alto número de observaciones, ya que se podrían obtener resultados erróneos. Entre los distintos métodos están (Gujarati y Porter, 2011):

- *Método de las medias móviles*: Se realiza la sustitución de la serie original X_t por una serie suavizada tomada como línea de tendencia. No es posible obtener predicciones con este método, ya que únicamente proporciona el valor de la tendencia en el tiempo de la serie.

Dada una serie temporal $X_{it}; t \equiv (t_1, t_2, t_3, \dots, t_n), i = 1, 2, 3, \dots, k$, para aplicar el método de medias móviles, se realiza el promedio de cada valor con algunas de las observaciones anteriores y posteriores. El cálculo de medias móviles se realizará en periodos de k (número de subperiodos en que se divide el año, p.e. $k=4$: trimestres, $k=2$: semestres), y determinará la longitud de la media móvil.

Para llevar a cabo el método de la media móvil se realizará la sustitución de x_t por la media móvil \bar{x}_t . Se tomarán las k primeras observaciones, se calculará la media y se asignará a un valor central. Para realizar este proceso se deben tener en cuenta dos casos:

- k es impar: Los subíndices de las medias móviles estarán compuestos por números enteros, por lo que la serie de las medias móviles estará centrada, derivando en la pérdida de $(k - 1)$ datos. En este caso la tendencia estará formada por la unión de los puntos (t, \bar{x}_t) .
 - k es par: En este caso los subíndices no serán números enteros en todas las ocasiones, y como resultado la serie no estará centrada. Se realizará el cálculo de la serie aritmética entre dos valores consecutivos de las medias móviles calculadas y se representará por $\bar{\bar{x}}_t, t = \frac{k}{2} + 1, \frac{k}{2} + 2 \dots$. La línea de tendencia estará formada por $(t, \bar{\bar{x}}_t)$.
- *Método del ajuste analítico*: Se realiza un ajuste por regresión de los valores de la serie a una función de tiempo. El supuesto más habitual es el de la tendencia lineal; se trata de una línea recta que muestra cómo la tendencia aumenta o disminuye a un ritmo constante $Z(t) = a + bt$ que se ajusta correctamente a los datos.

3.1.3 COMPONENTE ESTACIONAL

Se trata de movimientos periódicos que se repiten en intervalos cortos de tiempo con duración casi constante.

Los principales objetivos para determinar las componentes estacionales son:

- Conocer los componentes estacionales.
- Eliminarlos, proceso conocido como *desestacionalización*. De esta manera es posible observar mejor el proceso, aislándolo de influencias estacionales.

La mayoría de las series se clasifican en un esquema multiplicativo, esto es que se pueden definir en base a los cuatro componentes de una serie temporal como:

$$(X_{it} = T_{it} \cdot E_{it} \cdot C_{it} \cdot A_{it}) \quad (3.9)$$

Mientras que otras series pueden mostrar un carácter aditivo:

$$(X_{it} = T_{it} + E_{it} + C_{it} + A_{it}) \quad (3.10)$$

En primer lugar, se determinará el movimiento de medias móviles de orden k como el número de periodos que son observados en un año. Se denotarán con y' los nuevos valores y se puede expresar como: $Y = Y' \cdot E + I$. Dividiendo la ecuación por Y' y despejando la componente estacional se refleja como:

$$E = \frac{Y}{Y'} - \frac{I}{Y'} \quad (3.11)$$

En último lugar, se determinará la media para cada periodo estacional, con lo que se obtendrá un valor representativo de cada uno de ellos. Los valores obtenidos serán los componentes estacionales.

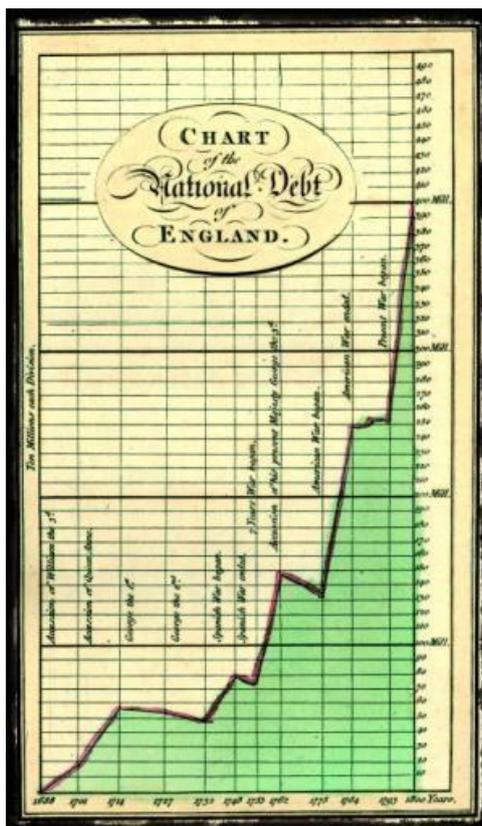
3.2 ANÁLISIS DE SERIES TEMPORALES

3.2.1 VISUALIZACIÓN DE LAS SERIES TEMPORALES

Para el análisis de series temporales una de las primeras técnicas utilizadas es la representación gráfica de los datos, la cual permite obtener cierta información acerca de los mismos.

Playfair (1786) fue un economista precursor de las técnicas gráficas para el análisis estadístico, desarrollando técnicas como los gráficos de líneas, barras y tarta como se observa en la Ilustración 11. En su publicación *The Commercial and Political Atlas* contenía 43 series temporales y representaba un gráfico de barras acerca de diferentes aspectos económicos de Inglaterra, promoviendo la primera representación visual de los números en vez de mediante tablas.

Ilustración 11. Gráfico de la deuda nacional de Inglaterra (Playfair, 1786).



3.2.2 DESCOMPOSICIÓN DE LA SERIE TEMPORAL

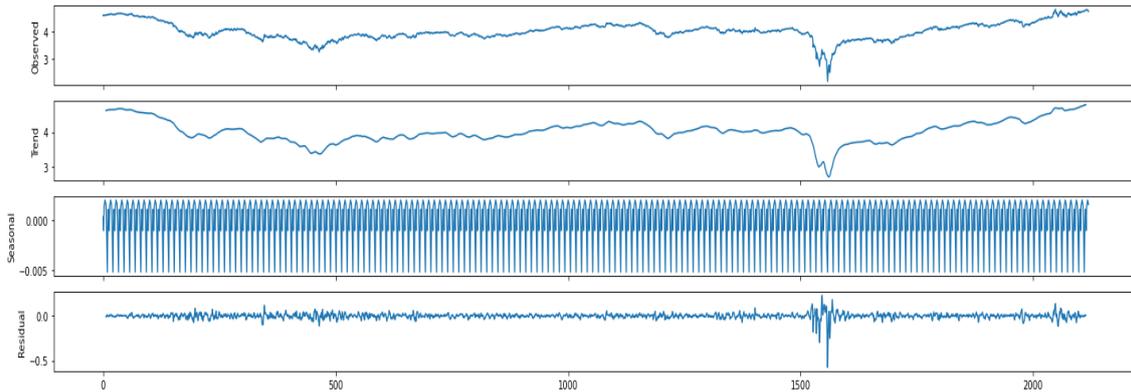
Los métodos de descomposición clásicos tienen su origen a principios del siglo XX. Es un proceso que constituye el inicio de los métodos de descomposición de las series temporales (Young, 1996). La descomposición consta de las siguientes fases:

- Identificación del tipo de modelado: Se tratará de un modelo multiplicativo cuando en función del tiempo se modifique la variación estacional.
- Identificación de la tendencia: Se realizará el cálculo a partir de las medias móviles.
- Identificación de la componente estacional: Se calcula mediante el promedio de los datos en frecuencia estacional y la resta de la tendencia.

- Identificación de la componente irregular: La señal obtenida tras restar de la serie original el componente de la tendencia y de la estacionariedad.

En la Ilustración 12 se observa la descomposición de la serie temporal relativa a los precios del petróleo de West Texas Intermediate con frecuencia diaria.

Ilustración 12. Descomposición de una serie temporal.



3.2.2.1 ESTACIONARIEDAD

La estacionariedad es un proceso mediante el cual se mantienen de forma constante en el tiempo la media, la varianza y la covarianza de la serie temporal. Un proceso estocástico será considerado como estacionario si no se muestran cambios en su media, no existe tendencia y su varianza y variaciones periódicas permanecen estables o han sido eliminadas en caso de que existieran (Rhif et al., 2019).

Con una notación matemática se puede definir que un proceso Y_t es estacionario cuando sus propiedades estadísticas en una secuencia $Y_{t1}, Y_{t2}, \dots, Y_{tn}$ ($n \geq 1$) son similares a la secuencia $Y_{t1+h}, Y_{t2+h}, \dots, Y_{tn+h}$ para cualquier número entero h .

Una serie estacionaria carece de tendencia, su media es constante y la varianza es representada gráficamente con una amplitud constante. En función del aumento del tiempo t el proceso tiende al equilibrio.

Un proceso Y_t no es estacionario cuando las propiedades estadísticas de la secuencia $Y_{t1}, Y_{t2}, \dots, Y_{tn}$ ($n \geq 1$) son diferentes a las de $Y_{t1+h}, Y_{t2+h}, \dots, Y_{tn+h}$.

3.2.2.1.1 TEST DE DICKEY FULLER

Para detectar la estacionariedad de una serie temporal existen algunos test de hipótesis que permiten confirmar si es estacionaria o no. Los más ampliamente utilizados son los test de raíces unitarias de Fuller (1976) y Dickey Fuller Aumentado (ADF) (Dickey y Fuller, 1979), que tratan de corroborar la existencia de raíces unitarias en la serie temporal. Una *raíz unitaria* es una tendencia estocástica en la serie temporal, por lo que si la serie presenta una raíz unitaria se trata de un patrón sistemático que no es posible predecir.

La hipótesis nula de esta prueba $H(0)$ no es rechazada cuando existe una raíz unitaria. Y puede ser definida como:

$$y_t = \mu + \rho y_{t-1} + \varepsilon_t \quad (3.12)$$

donde μ y ρ son parámetros a estimar y ε_t es el término de error que se define como ruido blanco de la serie. ρ denota un coeficiente de autocorrelación que toma valores $-1 < \rho < 1$. En el caso de que $\rho=1$, entonces la serie y no es estacionaria. La hipótesis de estacionariedad puede ser evaluada si $\rho < 1$. El test ADF se basa en la especificación:

$$\Delta Y_t = \delta + \gamma Y_{t-1} + \beta_1 t + \sum_{i=2}^p \beta_i \Delta Y_{t-i+1} + \varepsilon_t \quad (3.13)$$

tratando de probar la hipótesis nula que $\gamma = 0$.

Una alternativa a los test de raíz unitaria son los contrastes de estacionariedad. Los más empleados son los investigados por Kwiatkowski et al. (1992) y Leybourne y McCabe (1994) cuya diferencia radica en cómo se realiza el tratamiento de los parámetros de la autocorrelación de la serie temporal.

3.2.2.1.2 TEST DE PHILLIPS-PERRON

La hipótesis nula de la prueba de Phillips-Perron (PP) (1988) es que existe una raíz unitaria, existiendo la alternativa de que no existe. Si el *p-valor* se encuentra por encima de un tamaño crítico, no se puede rechazar el valor nulo y la serie tiene una raíz unitaria.

La diferencia con la prueba ADF es que la regresión estimada incluye solo un residuo de la variable dependiente, además de los términos de tendencia. Los *p-valores* se obtienen mediante la aproximación de superficie de regresión de MacKinnon (1994) utilizando las

tablas de 2010. Si el p -valor está cerca de ser significativo, los valores críticos deben usarse para decidir si se debe rechazar el valor nulo.

3.2.2.1.3 DIFERENCIACIÓN DE SERIES TEMPORALES

La importancia del manejo de series temporales estacionarias reside en que los modelos para la predicción de series temporales se basan en series estacionarias. Si las características de la serie temporal cambian a lo largo del tiempo resulta complicado representar eventos futuros en base a los pasados mediante un modelo lineal sencillo.

En su mayoría, las series económicas no proceden de procesos estacionarios, manteniendo una tendencia ascendente o descendente, pero esta limitación puede ser resuelta mediante la aplicación de diferencias en una o más etapas. En el caso de que la serie no presente estacionariedad en la varianza, también será necesario realizar transformaciones aplicando logaritmos antes de la diferenciación.

La *diferencia* es la técnica utilizada habitualmente para eliminar la tendencia en una serie temporal. La diferenciación de una serie temporal Y_t consiste en la transformación de Y_t en una nueva serie $D^{(1)}_t$ definida como:

$$D^{(1)}_t = D(Y_t) = Y_t - Y_{t-1} \quad (3.14)$$

Este procedimiento puede llevarse a cabo nuevamente sobre una serie previamente diferenciada, definiéndose como diferencias de segundo orden:

$$D^{(2)}_t = D(D^{(1)}_t) = D^{(1)}_t - D^{(1)}_{t-1} \quad (3.15)$$

Una diferencia de orden m es definida como:

$$D^{(m)}_t = D(D^{(m-1)}_t) = D^{(m-1)}_t - D^{(m-1)}_{t-1} \quad (3.16)$$

3.2.3 MODELOS AUTORREGRESIVOS

Un modelo autorregresivo representa un proceso aleatorio donde la variable de interés depende de sus observaciones pasadas. A continuación, se estudian los principales modelos de autorregresión existentes de manera cronológica, y cómo estos han ido evolucionando con el fin de dar respuesta a las dificultades y problemáticas presentes en el análisis de series temporales (Villavicencio, 2010).

3.2.3.1 MODELO DE AUTORREGRESIÓN LINEAL

En los modelos de autorregresión lineal (AR) los valores predichos dependen linealmente de los predecesores. Se utilizan para el pronóstico sobre observaciones de las que se conoce el valor en determinados momentos del tiempo.

En estos modelos tanto la variable dependiente como la variable explicativa representan la misma variable; la diferencia radica en que ambas están separadas por un instante de tiempo t , siendo la variable dependiente anterior ($t-1$) a la variable explicativa (Granger y Morris, 1976).

Son popularmente conocidos como AR(p) donde p es la etiqueta *orden*, que representa el número de periodos que serán retrocedidos para realizar la predicción. Cuanto mayor sea p más información contendrá el modelo. El modelo AR(p) puede ser definido como:

$$Y_t = \delta + \phi_1 Y_{t-1} + \phi_2 Y_{t-2}, \dots, + \phi_n Y_{t-n} + \varepsilon_t \quad (3.17)$$

En donde los elementos ϕ_i para $i = 1, 2, \dots, n$ y δ son constantes y ε_t representa el ruido blanco.

3.2.3.2 MODELO DE MEDIAS MÓVILES

El en modelo de medias móviles (MA) la dependencia de los valores predichos es lineal con las entradas anteriores. Explica el valor de una variable en un tiempo t en función de un término independiente y una sucesión de términos de error que se corresponden con periodos precedentes. Se denota como modelo MA(q) en donde la etiqueta q representa los términos de error.

Estos modelos centran su importancia no en si la serie es estacionaria, si no en si es invertible. Se trata de una regresión lineal del valor actual en función de error de ruido blanco. El modelo MA(q) se denota como:

$$Y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2}, \dots, + \theta_n \varepsilon_{t-n} \quad (3.18)$$

3.2.3.3 MODELO ARMA

La combinación de los dos modelos anteriores (AR y MA), recibe el nombre de modelo autorregresivo de media móvil (en inglés *AutoRegressive Moving Average model*, ARMA) y se representa como:

$$Y_t = \delta + \phi_1 Y_{t-1} + \phi_2 Y_{t-2}, \dots, + \phi_n Y_{t-n} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2}, \dots, + \theta_n \varepsilon_{t-n} \quad (3.19)$$

En todos los modelos representados el término de ruido blanco ε_t se denomina *innovación* al tratarse de un elemento que no es posible predecir a partir de datos anteriores.

3.2.3.4 MODELO ARIMA

Para la aplicación de los modelos anteriores es necesario que la serie Y_t sea estacionaria, en caso contrario y que sea necesaria la aplicación de diferencias a la serie, se definirá el modelo como $ARIMA(p,d,q)$, que consiste en un modelo $ARMA(p,q)$ al que ha sido necesario aplicarle d diferencias para convertir la serie en estacionaria.

La metodología Box Jenkins (1970) se aplica a los modelos ARMA y ARIMA para tratar de localizar el modelo que mejor se ajusta en base a los datos disponibles. El método original consiste en un enfoque iterativo de tres fases:

- Identificación y selección del modelo: Tras asegurar la estacionariedad de la serie, se parte de la identificación de las diferencias aplicadas para identificar el componente a usar en el modelo.
- Estimación de parámetros: Cálculo de los coeficientes que mejor ajusten al modelo ARIMA. Los más utilizados son el de *verosimilitud* y el de *mínimos cuadrados no lineales*.
- Comprobación del modelo mediante ensayo.

3.2.3.5 *MODELO SARIMA*

En algunos casos las series temporales tienen una componente estacional que genera dificultades a la hora de ser capturada, razón por la que se desarrollaron los modelos SARIMA, que son una generalización de los modelos ARIMA e integran un término de estacionalidad, un modelo SARIMA(P,D,Q) puede ser descrito como:

$$\phi_p(B)\Phi_P(B^s)W_t = \theta_q(B)\Theta_Q(B^s)\varepsilon_t \quad (3.20)$$

El operador de *lag* es denotado por B , mientras que ϕ_p , Φ_P , θ_q y Θ_Q son las componentes autorregresivas y de medias móviles representadas por medio de polinomios de órdenes p,P,q,Q respectivamente, siendo P y Q estacionales. ε_t se refiere al ruido blanco y W_t es el diferenciador de la serie temporal. Por lo que un modelo SARIMA se denomina de orden $(p,d,q)\times(P,D,Q)s$, siendo s el número de periodos determinados en los que se repite el patrón de la serie temporal, la ventana estacional (en caso de datos trimestrales, sería p.e. $s=4$) (Villavicencio, 2010).

3.2.3.6 *MODELO ARFIMA*

La *memoria larga*, que consiste en la persistencia mostrada por las autocorrelaciones en algunas series temporales estacionarias, las cuales decrecen de forma lenta hasta cero, indicando efectos transitorios, está estrechamente relacionada con la persistencia mostrada por las autocorrelaciones de algunas series temporales estacionarias. Tienen un lento ritmo descendente y finalmente convergen en cero, lo que indica que las innovaciones o *shocks* tienen efectos temporales, pero a lo largo de amplios periodos de tiempo.

Granger (1980) y Granger y Joyeux (1980) suscitaron gran interés a partir de sus investigaciones sobre modelos de memoria larga aplicados a series económicas, afirmando que el comportamiento de algunas de estas series aparentemente no estacionarias, mostraban indicios de sobrediferenciación al aplicarles diferencias. Lo que ponía de manifiesto la necesidad de un término medio para aquellas series en las que es necesario aplicar una diferenciación como en los casos del modelo ARIMA con raíces unitarias, pero esta sin embargo tiene efectos contraproducentes como en los modelos ARIMA estacionarios con una media constante.

La solución propuesta se trata de un orden de integración fraccional, un proceso ARMA fraccionalmente integrado que se define como ARFIMA (p,d,q) , en donde d es un número real. Al permitir que se trate de un número no entero, este modelo se define como un término medio entre los procesos ARIMA con raíces unitarias ARIMA($d = 1$), y los procesos estacionarios ARMA ($d = 0$).

Se establece que un proceso estocástico Y_t sigue un proceso ARFIMA (p,d,q) si cumple:

$$Y_t = \phi(B)(1 - B)^d Y_t = \theta_0 + \theta(B)a_t, \quad t = 1, 2, \dots, t \quad (3.21)$$

En donde $\phi(B) = 1 - \phi_1 B - \dots - \phi_p B^p$ y $\theta(B) = 1 - \theta_1 B - \dots - \theta_q B^q$ son los polinomios autorregresivo y de medias móviles, respectivamente, de orden p y q de un proceso ARMA. d es el parámetro de diferenciación fraccional y a_t son variables aleatorias no observables.

El componente ARMA(p,q) del modelo ARFIMA se denomina como la componente de corto plazo.

3.2.3.7 CRITERIOS DE INFORMACIÓN PARA LA ELECCIÓN DEL MODELO

Para llevar a cabo la elección del modelo adecuado, algunos autores desarrollaron metodologías que permiten realizar la identificación de los modelos de una forma más sencilla a partir de la Teoría de la Información. Los criterios más ampliamente utilizados son el *Criterio de Información de Akaike* (AIC) (Akaike, 1970) y el *Criterio de Información Bayesiano* (BIC). Los criterios de información miden el balance entre la capacidad predictiva y la complejidad del modelo, expresándose de la siguiente manera:

$$xIC = \text{complejidad} - \text{ajuste} \quad (3.22)$$

xIC representa cualquiera de los dos criterios (AIC; BIC) que utilizan el criterio de máxima similitud como criterio de bondad de ajuste, y el número de parámetros para medir la complejidad.

- *Criterio de Información Akaike*: El criterio AIC es el utilizado con mayor frecuencia, ya que calcula y compara las puntuaciones AIC, eligiendo el que mejor se adapte a los datos. Identifica la relación que existe entre la varianza y el sesgo en el modelo. Está basado en la medida de información de Kullback-Leibler (1951), que permite

realizar la interpretación entre la distancia de dos variables realizando el cálculo de la log-verosimilitud de un modelo, hallando una función de pérdida que al ser minimizada obtenga el modelo que mejor represente los datos.

Se puede definir como:

$$AIC = 2k - 2Ln(L) \quad (3.23)$$

siendo k el número de parámetros del modelo y $Ln(L)$ la función del log-verosimilitud para el modelo estadístico. Se trata de una función creciente que decrecerá al ser multiplicada por -1 . L es el producto de las probabilidades de cada dato en función del modelo, y es el resultado de multiplicar m valores por 0 y 1. AIC decrecerá en función del aumento de la bondad de ajuste.

- *Criterio de información Bayesiano*: También conocido como criterio Schwarz (1978), se representa mediante:

$$BIC = k \ln(n) - 2Ln(L) \quad (3.24)$$

k representa el número de parámetros, L es el criterio de máxima similitud y n es el número de datos, pero en este caso la complejidad está condicionada a los parámetros k y $\ln(n)$, por lo que un aumento de la complejidad tiene efectos mayores al multiplicar el tamaño muestral, penalizando en mayor medida la complejidad que AIC, buscando el modelo más abstracto y sencillo.

4 TÉCNICAS DE PREDICCIÓN DE SERIES TEMPORALES

Los métodos matemáticos para la predicción de series temporales han evolucionado constantemente desde el inicio de las investigaciones en este campo. Poynting (1884) trató en su estudio de eliminar la tendencia y las fluctuaciones cíclicas realizando promedios sobre un determinado intervalo de tiempo. Más adelante otros investigadores como Hooker (1901) y Spencer (1904) escribieron sobre la eliminación de tendencias mediante la inclusión de polinomios de alto orden.

Todas estas investigaciones seguían la línea de la predicción de valores económicos. Hasta principios del siglo XX las predicciones se hacían en base a promedios de la serie temporal.

La predicción de valores económicos representa un problema clásico pero desafiante, que en las últimas décadas atrajo la atención de investigadores economistas e ingenieros con el propósito común de construir modelos de predicción eficientes y explorar herramientas de aprendizaje automático, siendo estudiado por investigadores como Jiang (2021).

4.1 TÉCNICAS CLÁSICAS DE PREDICCIÓN

En 1927 se establece el inicio de las técnicas clásicas de predicción. Yule (1927) describe la técnica de autorregresión para predecir periodicidades en la aparición de manchas solares (Bengio et al., 1995), con un modelo cuya predicción de un valor se realizaba sobre la suma pesada de los valores predecesores.

Los modelos diseñados posteriormente mantenían la idea principal de que, si la serie tiene un comportamiento establecido, puede distinguirse del término aleatorio realizando un

promedio de los valores pasados (Makridakis et al., 1998). Aparecen entonces los procesos de descomposición de series temporales, existiendo diferentes métodos de descomposición. El principal objetivo de todos ellos es lograr aislar cada componente de la serie temporal de la forma más fiel posible. La idea de la descomposición es lograr eliminar la tendencia y aislar el componente repetitivo o periódico. Los residuos generados se entienden como procesos aleatorios, no siendo posible su predicción, pero sí su identificación.

A los métodos regresivos les siguieron los de suavizado de series que se iniciaron a mediados del siglo XX (Holt, 1957; Magee, 1958). El alisado exponencial parte de la premisa de la dependencia de los valores pertenecientes a una serie, en base a la combinación atenuada exponencialmente de los valores que la preceden. En ambos casos se trata de reducir la estimación cuadrática del error ajustando linealmente el modelo.

Los modelos autorregresivos y de media móvil (ARMA) fueron desarrollados por Box y Jenkins (1970). Su metodología para la aplicación sobre series temporales, la predicción y el control ha adoptado el nombre de Metodología Box-Jenkins para series temporales. Tal y como se expuso en el capítulo anterior, los modelos ARMA combinan un modelo de autorregresión lineal y un modelo de media móvil, y los modelos ARIMA incluyen la diferenciación de la serie para convertirla en estacionaria.

Gran parte de la literatura existente consiste en la predicción de series temporales mediante los modelos ARMA, ARIMA y ARFIMA. Makridakis y Hibon (1997) desarrollaron una investigación para tratar de dar respuesta al peor rendimiento de los modelos ARIMA frente a técnicas más simples, obteniendo como conclusión que el principal problema reside en la conversión de la serie a estacionaria mediante la aplicación de diferencias, mientras que utilizando enfoques diferentes al propuesto por Box y Jenkins los resultados mejoran notablemente.

Junior et al. (2014) aplicaban el modelo ARIMA con el objetivo de evaluar su desempeño en los mercados financieros, a través de la previsión de variaciones futuras en una serie temporal del índice bursátil brasileño, considerando esta tarea como difícil dada la relativa incertidumbre involucrada con las variables en los mercados.

Los procesos fraccionarios son abordados por Bhardwaj y Swanson (2006), presentando una evidencia de pronóstico la cual sugiere que los modelos ARFIMA estimados producen mejores aproximaciones a los verdaderos valores económicos que los modelos

AR, MA, ARMA, ARIMA y modelos relacionados basados en el análisis de errores cuadráticos medios y precisión predictiva.

En esta línea, Chung et al. (2009) basaron su investigación en la realización de predicciones de datos en el mercado de valores y la industria china utilizando el método ARIMA. Ding, et al. (1993) por su parte, estudiaron cómo los modelos ARFIMA estimados producen predicciones fuera de la muestra significativamente mejores que los resultados basados en AR, MA, error cuadrático medio (MSE) y utilizando la técnica de Diebold y Mariano (1995) para medir la precisión de las pruebas. Sun et al. (2007) se basaron en el modelo ARFIMA para el análisis y predicción de los niveles de elevación del Gran Lago Salado, mostrando un mejor desempeño que los modelos ARMA convencionales. Mientras que, en la misma línea de investigación, Li et al. (2007) compararon los modelos ARMA, ARIMA, ARFIMA y GARCH para el análisis de series temporales.

Más recientemente, Tang et al. (2022) afirmaron que las series de tiempo económicas se caracterizan por su comportamiento no lineal, dinámico y caótico, hecho que ha llamado la atención de muchos investigadores en las últimas décadas y ha impulsado el desarrollo de nuevos modelos para su predicción.

4.2 TÉCNICAS DE SUAVIZADO

Las técnicas predictivas de suavizado o alisado resultan de utilidad para realizar pronósticos en horizontes inmediatos con buenos resultados incluso cuando el número de observaciones es pequeño.

Para aplicar los métodos de suavizado no es necesario que la serie temporal a estudiar tenga un comportamiento estacional, existiendo distintos métodos en función del tipo de serie a estudiar, en caso de no tener tendencia ni estacionalidad o de contar con ambas características.

4.2.1 TÉCNICAS DE PROMEDIADO

Estas técnicas se aplican en series temporales que no presentan ni tendencia ni estacionalidad. El comportamiento de este tipo de series se puede describir como estable, siguiendo un patrón subyacente μ con algunas fluctuaciones aleatorias e_t , por lo que pueden ser modelizadas de la siguiente manera:

$$X_t = \mu + e_t \quad (4.1)$$

- *Modelo Naive*: Otorga la misma importancia $\left(\frac{1}{t}\right)$ a todas las observaciones de la serie temporal a la hora de realizar predicciones. La previsión viene dada por la media de las observaciones.

$$\hat{X}_{t+1} = \bar{x} \quad (4.2)$$

- *Modelo de medias móviles*: Se basan en tener en cuenta únicamente las últimas k observaciones, de esta manera es posible dar el mismo peso a los últimos k datos $\left(\frac{1}{k}\right)$ y cero al resto mediante un procedimiento de medias móviles. Sustituye cada dato por una media de los k últimos, lo que produce que la serie temporal se suavice y se elimine el ruido, obteniendo el patrón subyacente de la serie. Cuanto mayor sea el número de observaciones relevantes k , más se suavizará la serie.

$$\hat{X}_{t+1} = \frac{\sum_{i=1}^k x_{t+i-1}}{k} \quad (4.3)$$

4.2.2 TÉCNICAS DE SUAVIZADO EXPONENCIAL

Son adecuadas en problemas de predicciones a corto plazo. Su estructura recursiva permite revisar las predicciones a medida que se dispone de nueva información. Entre los métodos se encuentran los siguientes:

- *Método de alisado simple*: Se aplica a datos cuya media es constante.

$$\hat{X}_{t+1} = \left(\frac{1}{n}\right)X_t + \left(1 - \frac{1}{n}\right)\hat{X}_t = \alpha X_t + (1 - \alpha)\hat{X}_t \quad (4.4)$$

donde $\alpha = \frac{1}{t}$ y $0 < \alpha \leq 1$. Cuanto mayor es α menor es el alisado.

- *Alisado exponencial adaptativo*: Las técnicas de alisado exponencial presentan el inconveniente de tener que estimar el valor de α óptimo. Con el objetivo de evitar este problema se recurre a la técnica adaptativa, que se representa como:

$$\hat{X}_{t+1} = \alpha_t X_t + (1 - \alpha_t) \hat{X}_t \quad (4.5)$$

En donde:

$$\alpha_t = \frac{E_t}{M_t} \quad (4.6)$$

Siendo E_n y M_n las versiones alisadas del error de predicción $e_n = X_t - \hat{X}_t$ y de su valor absoluto.

- *Suavizado exponencial doble*: Es posible realizar un alisado de la versión alisada de los datos para mejorar la predicción, que queda definido como:

$$\widehat{S}'_t = \alpha S'_t + (1 - \alpha) \widehat{S}'_{t-1} \quad (4.7)$$

4.3 MÁQUINAS DE VECTOR SOPORTE

Las Máquinas de Vector Soporte (en inglés *Support Vector Machines*, SVM) fueron desarrolladas en la década de los 90 del siglo XX como un modelo de clasificación-regresión. En su origen se planteaba como un método de clasificación binario, pero posteriormente se ha aplicado a problemas de clasificación múltiple y regresión.

El aprendizaje mediante vectores de soporte fue presentado por Vapnik (1979) como método para la construcción de hiperplanos separadores con máximo margen, pero su uso se desarrolló como solución a problemas de regresión y aproximación (Smola et al., 1998), llegando a usarse para la predicción de series temporales caóticas (Mukherjee et al., 1997).

4.3.1 HIPERPLANO

En un espacio p -dimensional, un hiperplano es definido como un subespacio plano de dimensiones $p-1$, no siendo necesario que el subespacio pase por el origen. La definición matemática para un hiperplano de p dimensiones es:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p = 0 \quad (4.8)$$

Dados los parámetros $\beta_0, \beta_1, \beta_2, \dots, \beta_p$, todos los pares de valores de x para los que se cumple la igualdad, son puntos del hiperplano. Cuando no se satisface la ecuación anterior, obteniendo:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p > 0 \quad (4.9)$$

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p < 0 \quad (4.10)$$

significa que el punto x caerá a un lado o al otro del hiperplano, por lo que se asume que un hiperplano divide un plano p -dimensional en dos mitades. Solo es necesario conocer el signo de la ecuación para saber en qué lado habrá caído el punto x .

4.3.1.1 HIPERPLANO PARA LA CLASIFICACIÓN BINARIA

Si contamos con una matriz de datos $n \times p$, en donde n se corresponde con el número de observaciones y p con el número de predictores, el objetivo es desarrollar un clasificador en base al subgrupo de datos de entrenamiento. Este clasificador tendrá la capacidad de seleccionar de forma correcta nuevas observaciones en función de un hiperplano de separación, en donde la respuesta a obtener tiene dos posibles valores y_1, y_2, \dots, y_n :

$$\sigma_0 + \sigma_1 x_{i1} + \sigma_2 x_{i2} + \dots + \sigma_p x_{ip} > 0 \text{ si } y_i = 1 \quad (4.11)$$

$$\sigma_0 + \sigma_1 x_{i1} + \sigma_2 x_{i2} + \dots + \sigma_p x_{ip} < 0 \text{ si } y_i = -1 \quad (4.12)$$

Considerando $\sigma_0, \sigma_1, \sigma_2$ los coeficientes pertenecientes al hiperplano, una nueva observación x se asignará a un grupo u otro en función del lado del hiperplano en el que se localice. Dependiendo del signo de $f(x) = \sigma_0 + \sigma_1 x_{i1} + \sigma_2 x_{i2} + \dots + \sigma_p x_{ip}$, si $f(x)$ es positivo la observación x será asignada al grupo 1, mientras que si es negativo será asignada a la clase -1 . Este tipo de algoritmos también tienen propiedades informativas;

cuanto más lejano se encuentre el valor de $f(x)$ de 0, más lejano estará x del hiperplano, lo que aporta mayor seguridad en la clasificación de la variable.

En el caso de que los datos a estudiar sean perfectamente separables mediante un hiperplano, existirá un número infinito de hiperplanos, de modo que será necesario escoger el hiperplano más alejado de los datos de entrenamiento, conocido como *hiperplano óptimo de separación*. Este hiperplano puede ser calculado a partir de las distancias en perpendicular de cada una de las observaciones respecto a un hiperplano determinado, en donde la menor distancia se corresponde con la distancia más pequeña de las observaciones a este hiperplano. Este espacio es conocido como *margen*. El hiperplano óptimo será aquel que tenga una distancia mínima mayor en las observaciones al hiperplano o menor margen. Las observaciones que se encuentran más próximas al hiperplano son conocidas como *vectores de soporte*.

4.3.2 CLASIFICADOR DE VECTORES SOPORTE

Las observaciones pertenecientes a dos clases no son necesariamente separables por medio de un hiperplano. En estos casos un clasificador de este estilo puede no ser la solución óptima, ya que la inclusión de una única muestra puede cambiar radicalmente el hiperplano óptimo de separación haciendo que deje de ser el modelo adecuado.

El Clasificador de Vectores Soporte (del inglés *Support Vector Classifier*, SVC), basándose en un hiperplano no realiza la separación perfecta, lo que le dota de la capacidad de ser más robusto a observaciones individuales y una mejor clasificación en las categorías de entrenamiento y prueba. Este tipo de clasificador permite que algunas observaciones se encuentren en el lado incorrecto del hiperplano o del margen. Este modelo incorpora a su diseño un parámetro c , llamado *parámetro de regularización* que controla la permisividad de observaciones situadas en el lado incorrecto del hiperplano. Si $c > 0$, c es el número máximo de observaciones que pueden encontrarse en el lado incorrecto. Cuanto menor sea c , menor será el margen, tendrá poco *bias* pero mucha *varianza*. Si $c = 0$ entonces el SVC es equivalente al *hiperplano óptimo de separación*.

4.3.2.1 MÁQUINAS DE VECTOR SOPORTE

Cuando el problema implica límites no lineales se recurre al uso de las Máquinas de Vector Soporte (del inglés *Support Vector Machines*, SVM) que aumentan la dimensionalidad de una forma específica a partir del uso de núcleos (*kernels*), los cuales se encargan de transformar un espacio de pocas dimensiones en uno mayor mediante transformaciones de los datos. Los *kernels* más populares son:

- *Kernel* lineal:

$$K(x_i, x'_i) = \sum_{j=1}^p x_{ij}, x'_{ij} \quad (4.13)$$

Cuantifica la similitud de dos observaciones usando la correlación de Pearson.

- *Kernel* polinómico:

$$K(x_i, x'_i) = \left(1 + \sum_{j=1}^p x_{ij}, x'_{ij} \right)^d \quad (4.14)$$

En un *kernel* polinómico de grado d , se permite un límite de decisión flexible.

- *Kernel* radial:

$$K(x_i, x'_i) = \exp \left(-\gamma \sum_{j=1}^p (x_{ij}, x'_{ij})^2 \right)^d \quad (4.15)$$

Siendo γ una constante positiva que implica que cuanto mayor es su valor, mayor es la flexibilidad del SVM.

4.4 FACEBOOK PROPHET

Se trata de un modelo presentado por Facebook (Taylor y Letham, 2018), que originalmente fue concebido para realizar predicciones de datos diarios con

estacionalidad semanal y anual, además de poder incluir los efectos de las vacaciones. Dado el éxito del modelo, se modificó con el fin de cubrir un espectro más amplio de datos estacionales. Su funcionamiento es óptimo con series temporales univariadas que constan de una fuerte estacionalidad y cantidad suficiente de datos históricos.

La idea general es un modelo aditivo, la ecuación se ajusta a la tendencia, estacionalidad y las vacaciones por lo que puede ser representada como:

$$y_t = t_t + e_t + h_t + \varepsilon_t \quad (4.16)$$

En donde t_t representa la tendencia de la serie, e_t la estacionalidad, h_t el efecto de las vacaciones, ε_t el término de error y y_t los valores a predecir.

PROPHET permite seleccionar el modelo sobre el que se desea realizar las predicciones, *modelo de crecimiento logístico* o *modelo lineal por partes*; siendo este último el que se emplea por defecto. El modelo de crecimiento logístico está definido como:

$$y_t = \frac{C}{1 + e^{-k(t-m)}} \quad (4.17)$$

donde C representa la cantidad de carga, k la tasa de crecimiento y m es un parámetro de compensación. Mientras que el modelo lineal por partes se ajusta mediante la siguiente ecuación:

$$y = \begin{cases} \beta_0 + \beta_1 x & x \leq c \\ \beta_0 + \beta_2 c + (\beta_1 + \beta_2)x & x > c \end{cases} \quad (4.18)$$

en donde c representa el punto del cambio de tendencia y β_i es un parámetro de tendencia.

Desde el desarrollo del modelo se ha experimentado un aumento en la literatura disponible acerca de predicciones de series temporales mediante el uso de PROPHET. Como por ejemplo, Navratil y Kolkova (2019) que en su investigación trataban de identificar las tendencias estacionales en el desarrollo de ingresos de un segmento de comercio electrónico.

4.5 TÉCNICAS DE PREDICCIÓN CON REDES NEURONALES ARTIFICIALES

En los últimos años el aumento exponencial de los datos disponibles ha convertido a las redes neuronales artificiales (ANN, por sus siglas en inglés), en una técnica cada vez más empleada en las tareas de aprendizaje automático. Existe una extensa literatura sobre el uso de las ANN en los trabajos de predicción como los descritos por Frank et al. (2001), Tealab (2018) y Gasparin et al. (2022) entre otros. Zhang et al. (1998) realizan un resumen sobre las publicaciones en las que se ha empleado las ANN para la predicción de series temporales. Hay un consenso generalizado en que este tipo de redes poseen unas características que las dotan de las capacidades adecuadas para llevar a cabo este tipo de tareas en comparación con las técnicas estadísticas clásicas.

Las ANN son capaces de modelar cualquier forma de relación desconocida en los datos sin suposiciones previas, teniendo la capacidad de generalizar y transferir las relaciones aprendidas a datos futuros y convirtiéndose en *aproximadores universales*. Esto quiere decir que tienen la capacidad de modelar cualquier forma de relación de los datos, en especial en el caso de las relaciones no lineales (Hornik et al., 1989), lo que dota a las ANN de la capacidad de modelar una gran cantidad de funciones.

Por estas razones las redes neuronales artificiales y las catalogadas como *profundas* (del inglés *Deep Neural Networks*, DNN) son cada vez más aplicadas en el estudio de problemas de predicción de series temporales, con una demanda en aumento. Zhang (2003) propuso en su investigación una metodología híbrida mediante la cual combinaba un modelo ARIMA y el uso de DNN para lograr el máximo aprovechamiento de ambos modelos. Para la valoración del modelo empleó conjuntos de datos reales, y los resultados mostraban que el modelo combinado podría ser una forma efectiva de lograr una mejora en la precisión de los dos modelos por separado. Siguiendo esta misma línea, Hill et al. (1996) investigaron acerca del comportamiento de las DNN en contraposición con los pronósticos obtenidos por varios métodos estadísticos clásicos para la predicción de series temporales (Makridakis et al., 1982), logrando demostrar un mejor desempeño de las DNN en comparación con los otros métodos estudiados, logrando las mayores diferencias en las series temporales discontinuas. La investigación llevada a cabo por Gheyas y Smith (2009) se centraba en la generación de un enfoque simple para el pronóstico de series temporales univariadas, basándose en un conjunto de técnicas de regresión generalizadas.

La literatura al respecto se completa con la aportación de Khashei y Bijari (2010), quienes a partir de sus estudios sobre un modelo que combina las técnicas clásicas y las DNN obtiene una mejora en la predicción de series temporales como resultado empírico.

Tratando de mejorar los problemas existentes en las tareas de predicción con series temporales con componentes lineales y no lineales, Yolcu et al. (2013) desarrollaron un modelo híbrido que constaba de ambos tipos de estructura, lineales y no lineales, asumiendo que una serie temporal puede tener los dos tipos de componentes. Sus resultados mostraron una efectividad mayor que los resultados disponibles en la literatura.

Es una tarea crucial determinar la estructura óptima de la red y el método de entrenamiento para el problema de predicción de series temporales y la obtención de la máxima precisión, por esta razón Zhang y Kline (2007) incluyeron en su investigación la elección de variables de entrada como uno de los factores críticos a tener en cuenta.

Los modelos de redes neuronales más utilizados en tareas de predicción de series temporales son el perceptrón multicapa, redes neuronales recurrentes (como las LSTM y BiLSTM) y actualmente se investiga acerca del uso de las redes convolucionales con este fin. Estos modelos se encuentran descritos en el capítulo 2 de este documento.

4.5.1 SOBREAJUSTE E INFRAAJUSTE

Un error de generalización se da cuando un modelo clasifica correctamente las muestras durante el entrenamiento, pero al ser validado con el conjunto de datos de prueba, que no ha sido probado antes, falla procesando de manera incorrecta. Estos errores suelen darse cuando las muestras son incompletas o tienen demasiado ruido.

Para describir el sobreajuste (*overfitting*) y el infraajuste (*underfitting*) es necesario conocer los siguientes conceptos:

- *Sweet spot*: Punto de equilibrio en el aprendizaje del modelo que servirá para asegurar no caer en el *underfitting* y *overfitting*.
- *Bias*: Se trata de un parámetro que nos indica si un modelo no ha tenido en cuenta todos los datos disponibles en el conjunto de datos. Un nivel alto de *bias* indica un *underfitting* del modelo.

- *Varianza*: Señala cómo de sensible es el modelo al conjunto de entrenamiento, un modelo con una varianza alta aprenderá también el ruido del conjunto y producirá *overfitting*.

El *overfitting* o sobreajuste se trata de un fenómeno manifestado porque el modelo se encuentra demasiado adaptado al problema propuesto con los datos de entrenamiento y no es capaz de generalizar con datos desconocidos. Suele darse cuando el número de parámetros es muy alto con respecto al número de muestras de entrenamiento. Cuando un modelo ofrece una precisión muy alta en el entrenamiento y una precisión muy baja en la validación, muy probablemente se trate de un caso de *overfitting*. Las posibles causas del *overfitting* son:

- Modelo demasiado potente: Debe encontrarse un equilibrio entre el *bias* y la *varianza* para que exista el mismo equilibrio entre la precisión y la coherencia de los datos obtenidos.
- Aprendizaje del ruido en el entrenamiento: Este problema se da cuando el conjunto de entrenamiento carece de datos suficientes o consta de demasiado ruido, entendido como excesiva variabilidad estadística inexplicable en los datos.

Algunas soluciones para el *overfitting* pueden ser:

- Simplificación del modelo: Una disminución en el número de capas o neuronas puede simplificar el modelo de forma que se ajuste mejor a los datos de entrenamiento. El mismo resultado puede obtenerse mediante la introducción de capas *Dropout*, que desactivan algunas de las neuronas de la red, simulando una simplificación del modelo. La introducción de *Early Stopping* consiste en la evaluación del modelo tanto en el entrenamiento como en la validación, cuando detecta un empeoramiento en el rendimiento con el conjunto de validación produciendo la parada del modelo.
- *Data Augmentation*: Aumento de los datos a partir de los existentes.
- Eliminación del ruido: Proceder a la estandarización de los datos y eliminación de la información no relevante para el modelo.
- Aumento del conjunto de datos: Aumentar siempre que sea posible el número de observaciones.

En el caso del *underfitting* o infraajuste el modelo no es capaz de encontrar una relación entre los datos de entrenamiento, y por lo tanto no tiene la capacidad de realizar predicciones. Entre las causas más habituales del *underfitting* pueden estar:

- No hay suficientes parámetros para modelar los datos.
- El algoritmo no disponía de suficiente tiempo para modelar el problema.
- Baja entropía en el conjunto de entrenamiento.

Para solucionar el *underfitting* es necesario ampliar el conjunto de entrenamiento, de manera que el modelo es forzado a trabajar con patrones más complejos.

BLOQUE II

SOLUCIÓN PROPUESTA, RESULTADOS Y CONCLUSIONES

Este bloque se organiza en tres capítulos. En primer lugar, se detalla el modelo híbrido propuesto para la predicción de series temporales y que supone la principal aportación de esta tesis doctoral. Se describe la arquitectura que compone la red neuronal, así como las decisiones técnicas tomadas para su elaboración con el objetivo de obtener resultados de gran precisión. En el siguiente capítulo se realiza una detallada descripción de las herramientas empleadas, así como de las métricas de error empleadas para realizar la evaluación del modelo propuesto. Se especifican también los tres experimentos llevados a cabo y la comparativa de resultados con los modelos del estado del arte utilizados para realizar el contraste. En el último capítulo se recogen las principales conclusiones extraídas de la presente investigación y de detallan las futuras líneas de trabajo propuestas.

5 MODELO BiLSTM-GCN

El modelo BiLSTM-GCN representa una evolución de los modelos existentes hasta el momento en la literatura y descritos en los capítulos anteriores.

Tras el estudio y análisis de los principales modelos descritos en la literatura, en esta tesis doctoral se optó por diseñar un modelo híbrido que combinase aquéllos que mostraban un mejor comportamiento, y determinar si un modelo híbrido podía superar a los modelos por separado. Con una mejora significativa con respecto a los resultados reflejados en la literatura con diferentes modelos de predicción, tal y como se presentará en el capítulo 6, es posible inferir que este modelo permite obtener resultados de gran precisión en la predicción de series temporales de carácter económico.

Con el fin de favorecer una mejor comprensión de la arquitectura del modelo de red propuesto en esta tesis doctoral, a continuación, se realiza una breve presentación de la arquitectura de una red neuronal artificial.

5.1 ARQUITECTURA DE UNA RED NEURONAL

El hecho de que las neuronas puedan organizarse en capas a diferentes niveles supone una de las principales características de las redes neuronales. Los niveles que pueden encontrarse en una ANN son la capa de entrada, capas ocultas y una capa de salida (Zhang et al., 2021). En este tipo de arquitecturas debe tenerse en cuenta el número de neuronas que compondrán cada capa, los parámetros de entrenamiento y las métricas que servirán para evaluar el rendimiento del modelo.

La capa de entrada es la responsable de recibir las señales que proceden del exterior y propagarlas hacia las neuronas que se encuentran en la siguiente capa, estas neuronas que

se encuentran contenidas en las capas ocultas realizarán un procesamiento no lineal de la información recibida por las neuronas de la capa anterior. Finalmente, la capa de salida será la responsable de dar una respuesta por cada una de las entradas recibidas (Rumelhart et al., 1986).

No existe un método definido que permita establecer el número más adecuado de capas en un modelo, por ello Güler et al. (2005) identificaban como método más adecuado el de prueba y error.

La arquitectura de una red neuronal puede ser definida como:

$$I - (H_1, H_2, H_3, \dots, H_N) - O \quad (5.1)$$

Donde I es la representación del número de nodos de entrada, mientras que H_n representa el número de neuronas en la capa oculta n y O la capa de salida.

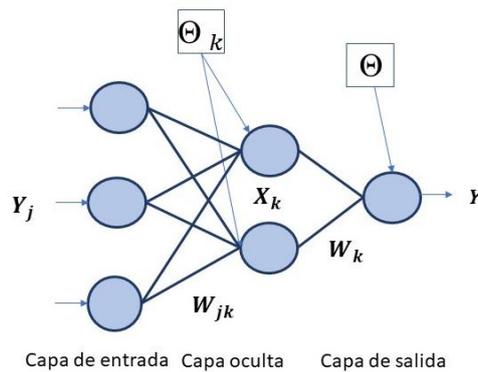
Fu (1994) realiza una descripción sobre cómo cada neurona en cada capa tiene una relación con las neuronas de la siguiente capa, y cómo estas relaciones tienen asociado un peso que es calculado a partir del resultado que se ha obtenido en la capa de salida y el valor real esperado. Este error es propagado a las capas anteriores, permitiendo un ajuste de los pesos entre las neuronas a lo largo del entrenamiento.

Para poder realizar este proceso es necesario disponer de un conjunto de datos que permita realizar la validación del modelo, y a partir de este evaluar la precisión de los resultados obtenidos en base al Error Cuadrático Medio (en inglés *Mean Squared Error*, MSE), con el principal objetivo de reducir este valor.

La Ilustración 13 representa la estructura de una red neuronal, en donde Y_i representa el vector de entrada $Y_j = \{y_1, y_2, y_3, \dots, y_n\}$; mientras que el vector de pesos de los i nodos de la capa de entrada a los k nodos de la primera capa oculta se define como W_{jk} ($j = 1, 2, 3, \dots, n; k = 1, 2, 3, \dots, m$); el vector de las k neuronas de las capas ocultas X_k ($k = 1, 2, 3, \dots, m$) se representa como $X_k = f(\sum_{j=1}^n W_{jk} Y_j + \Theta_k)$. El vector de la capa de salida con una única neurona Y es $Y = f(\sum_{k=1}^m W_k X_k + \Theta)$. En último lugar, el valor polarizado de los nodos de las capas ocultas se define como Θ_k ($k = 1, 2, 3, \dots, k$), siendo Θ el valor polarizado de la capa oculta.

La propuesta realizada por Hinton y Salakhutdinov (2006) implicaba realizar el entrenamiento de la red partiendo de una correcta inicialización de los pesos del modelo y un determinado número de capas ocultas, en lugar de la práctica común de adjudicar valores aleatorios. Para la realización de este proceso era necesario realizar el entrenamiento de forma no supervisada para posteriormente hacerlo de forma supervisada, estableciendo como valores iniciales los pesos recibidos en el proceso.

Ilustración 13. Arquitectura de una red neuronal



La inicialización de pesos de Xavier fue establecida por Glorot y Bengio (2010), quienes propone un proceso eficiente en el que se inicializan los pesos sin la necesidad de realizar el entrenamiento supervisado, convirtiendo este esquema en la metodología más ampliamente utilizada en el aprendizaje profundo, que ha demostrado ofrecer un mejor rendimiento y precisión gracias a la elección de una función de activación no lineal, en particular la Unidad Lineal Rectificada (ReLU) (Jarrett et al., 2009; Nair y Hinton, 2010).

5.2 ARQUITECTURA DEL MODELO BiLSTM-GCN

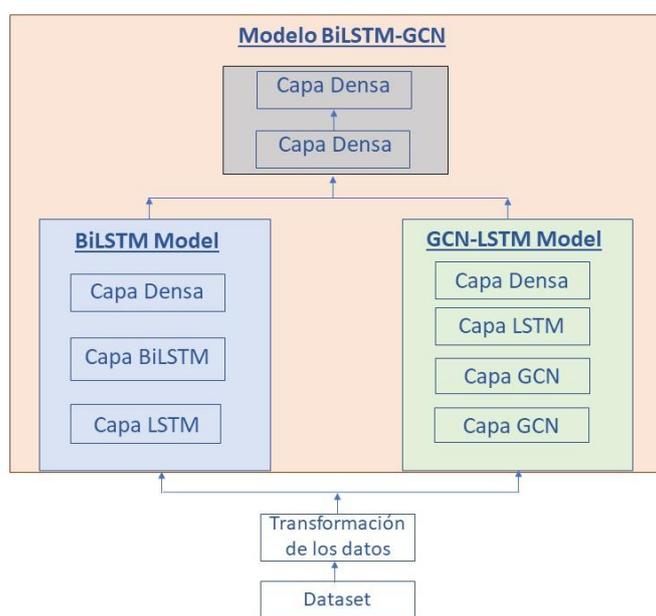
Para la predicción de valores a partir de series temporales de carácter económico, se distinguen diferentes etapas, entre las que destacan la recogida y preprocesamiento de la serie temporal, el entrenamiento del modelo de red neuronal y finalmente la recogida de resultados y su evaluación.

El enfoque propuesto en este trabajo y definido como BiLSTM-GCN, combina dos modelos, BiLSTM y GCN-LSTM, los cuales son pre-entrenados con la misma serie temporal obteniendo una salida de cada uno de ellos, correspondiente al modelo con los

hiperparámetros definidos. Las dos salidas obtenidas son recogidas y aportadas como entrada en el nuevo esquema además de la serie temporal original.

El modelo final se compone de las salidas de los modelos anteriores y dos capas densas, también llamadas capas completamente conectadas, en las que todas las neuronas se encuentran conectadas con todos los nodos de la capa anterior; que tras un nuevo entrenamiento con los datos arrojará como salida la predicción de la serie temporal como se refleja en la Ilustración 14.

Ilustración 14. Arquitectura del modelo BiLSTM-GCN



El modelo BiLSTM es la primera parte que conforma el esquema combinado propuesto. Este modelo consta de una capa de entrada, dos capas ocultas LSTM y BiLSTM respectivamente, y una capa densa como salida del modelo (ver Ilustración 15).

El segundo diseño empleado en el enfoque combinado propuesto está formado por una arquitectura GCN-LSTM inspirada en el modelo desarrollado por Zhao et al. (2019), la cual se estructura en dos partes definidas: un conjunto de capas convolucionales de grafos definidas por el usuario, y un conjunto de capas LSTM especificadas de nuevo por el usuario, a las cuales se une una capa *Dropout*, la cual se utiliza para realizar la desactivación de algunas neuronas y regularizar el modelo, y la capa de salida.

Las entradas que recibe este modelo son la serie temporal a tratar y la matriz de correlación (CM) que resulta tras generar el grafo correspondiente con los datos de la serie temporal. Esta arquitectura queda representada en la Ilustración 16.

Ilustración 15. Arquitectura del modelo BiLSTM

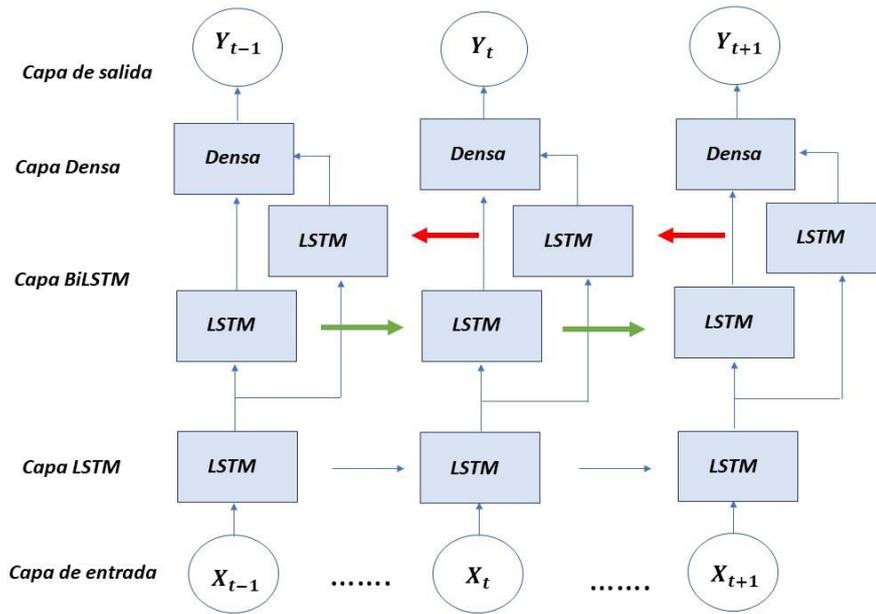
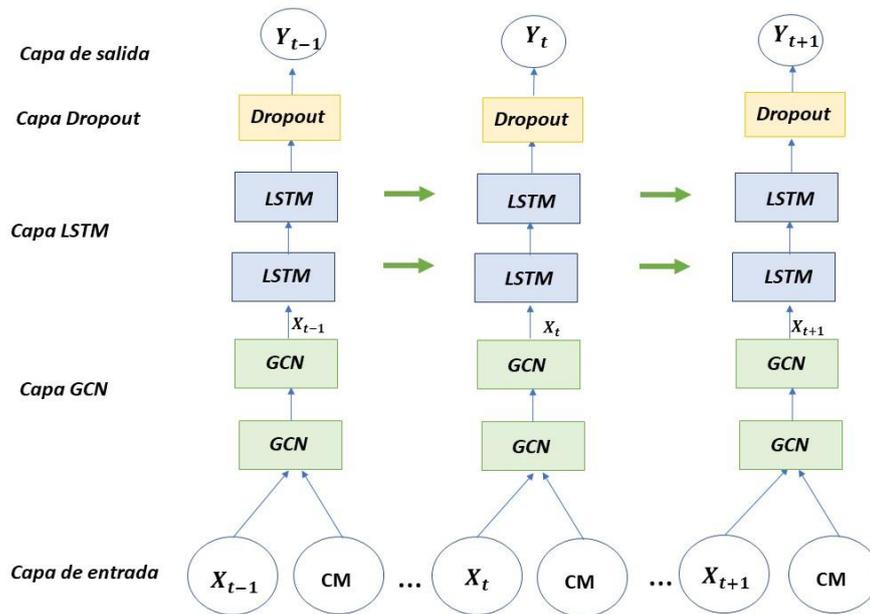


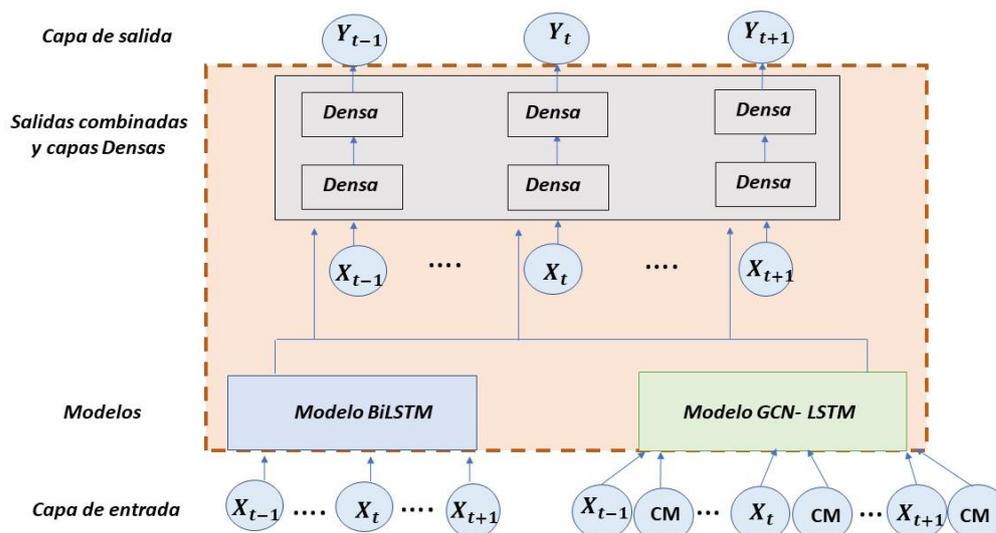
Ilustración 16. Arquitectura del modelo GCN-LSTM



El modelo combinado BiLSTM-GCN se compone de la concatenación de las salidas de los modelos BiLSTM y GCN-LSTM tras ser entrenados con la serie temporal correspondiente, unificando sus salidas en la nueva red la cual recibe de nuevo la serie temporal sobre la que realizar las predicciones (ver Ilustración 17). Este modelo además incluye dos capas densas que permiten mejorar el procesamiento de la red y una capa de

salida. La concatenación de los modelos DNN permite un máximo aprovechamiento de las capacidades de los dos tipos de redes y la obtención de predicciones más precisas (Li et al., 2019; Rajalakshmi y Ganesh Vaidyanathan, 2022). Este modelo combinado se denomina BiLSTM-GCN en este trabajo de investigación.

Ilustración 17. Arquitectura del modelo combinado BiLSTM-GCN



5.2.1 PARÁMETROS DEL MODELO BiLSTM-GCN

Para el correcto funcionamiento de una red neuronal artificial es necesario ajustar correctamente los parámetros que forman parte de la configuración de la red. Cuanto mayor sea una red neuronal más cantidad de parámetros deberán ser ajustados, por lo que el usuario deberá realizar los cambios necesarios para obtener los mejores resultados.

El mayor problema en la identificación de los valores de cada parámetro es el número de parámetros a definir y las diferentes combinaciones que pueden llevarse a cabo, además de los distintos valores que puede tomar un mismo parámetro. Bosan y Harris (1996) propusieron una matriz de datos en tres dimensiones para la representación de los datos, y así poder observar la relación entre cada uno de los parámetros y su rendimiento. Beck y Arnold (1977) afirmaban que la elección de los parámetros puede ser fácilmente estimada y diferenciada si los parámetros no son dependientes entre sí. Esta aproximación es posible cuando se comparan dos parámetros, ya que, en el caso de tratarse de más, el coste computacional aumentaría exponencialmente. Smith (2018) pone de manifiesto la

importancia de una elección adecuada de los hiperparámetros de una DNN para minimizar el error obtenido.

Con el fin de mejorar el rendimiento y minimizar el riesgo de sobreajuste, los hiperparámetros ajustados para la propuesta combinada BiLSTM-GCN son los siguientes:

- *Tasa de aprendizaje*: Hiperparámetro encargado de regular la velocidad a la que el modelo optimiza su función de coste. Un valor demasiado bajo puede hacer necesario aumentar el número de épocas y hacer que el entrenamiento sea más lento. Wilson y Martínez (2001) explican cómo la tasa de rendimiento tiene influencia directa en el tiempo de procesamiento aportando un punto de vista sobre la elección eficiente de este parámetro. En la misma línea Smith (2017) describe un nuevo método para la elección de la tasa de aprendizaje que elimina la necesidad de experimentar con diferentes valores para hallar el máximo rendimiento de la red.
- *Tamaño del lote*: de tamaño m sobre una muestra de tamaño N , de forma que $m < N$ y con la condición de que todos los lotes tengan el mismo tamaño y cumpla $n/m \in \mathbb{R}$. Devarakonda et al. (2017) desarrollaron un nuevo método mediante el cual el tamaño de los lotes aumentaba a medida que avanzaba el entrenamiento con el objetivo de mejorar la eficiencia de la red neuronal. Mientras que Masters y Luschi (2018) revisaban los supuestos comunes sobre la escala de la tasa de aprendizaje y la duración del entrenamiento, como base para una comparación experimental del rendimiento de la prueba para diferentes tamaños de mini lotes.
- *Época*: Intervalo comprendido entre la última actualización de las ponderaciones de un modelo de DNN respecto de un lote hasta la siguiente actualización sobre el mismo lote. El usuario hace una elección manual de las épocas de las que constará el modelo; en algunos modelos de DNN es posible especificar un umbral específico en lugar del número de épocas a ejecutar. Sinha et al. (2010) muestran el efecto de diferentes números de épocas en una red neuronal y se propone un método para determinar el número óptimo de épocas.
- *Capas ocultas*: El número de capas ocultas y el número de neuronas determinan en gran medida la complejidad del modelo y por tanto su potencial capacidad de aprendizaje. Para la selección del número de capas ocultas se experimentó con diferentes unidades, seleccionando el valor óptimo comparando las métricas de

evaluación. Uzair y Jamil (2020) revisan los diferentes impactos de las capas ocultas en la red, lo que brinda una descripción general del uso de tres números de capas ocultas que resultaron ser óptimos en términos de reducción de la complejidad del tiempo y obtención de la precisión calificada. Karsoliya (2012) realizó una encuesta para resolver el problema del número de neuronas en cada capa oculta y el número de capas ocultas requeridas.

- *Algoritmo de optimización*: La elección del algoritmo de optimización puede tener un impacto notable en el aprendizaje de los modelos. Actualizará los valores de los parámetros en función de la tasa de aprendizaje establecida. En este caso, se seleccionó Adam porque trata de combinar las ventajas de RMSProp (similar al descenso del gradiente), junto con las ventajas del descenso del gradiente con impulso (Sun, 2019). Sánchez y Velásquez (2011) describían la importancia del algoritmo de optimización en el entrenamiento de una red neuronal.

La elección de los parámetros se ha realizado siguiendo las recomendaciones de Goodfellow (2016), de realizar comparaciones múltiples utilizando un número fijo de ciclos, donde dicho número se determinará de acuerdo con las limitaciones computacionales y si el modelo llega a sobreajustarse. Los valores seleccionados para cada uno de estos parámetros se reflejan en la Tabla 1.

Tabla 1. Parámetros seleccionados

<i>Modelo</i>	<i>Tasa de aprendizaje</i>	<i>Tamaño del lote</i>	<i>Épocas</i>	<i>Capas ocultas</i>	<i>Neuronas</i>	<i>Optimizador</i>
BiLSTM	0,001	25	100	3	50	Adam
GCN-LSTM	0,001	32	50	2/1	16,10/100	Adam
BiLSTM-GCN	0,001	32	50	2	10	Adam

Para evaluar el rendimiento del modelo propuesto en base al número óptimo de capas, se llevaron a cabo distintos experimentos con el fin de obtener el máximo rendimiento. En la Ilustración 18 se observa la evolución de la raíz del error cuadrático medio (RMSE por sus siglas en inglés) en función del número de capas empleadas durante el entrenamiento y validación. Un mayor número de capas incrementa el RMSE como consecuencia de un *overfitting* derivado de la incapacidad de generalizar que ha adquirido la red por el elevado número de capas con respecto a la muestra utilizada. Mientras que en la

Ilustración 19 se muestra cómo el tiempo de ejecución se ve claramente incrementado con el aumento de capas, obteniendo una ejecución sumamente lenta cuando el número de observaciones es alto.

Ilustración 18. Comparativa RMSE respecto al número de capas

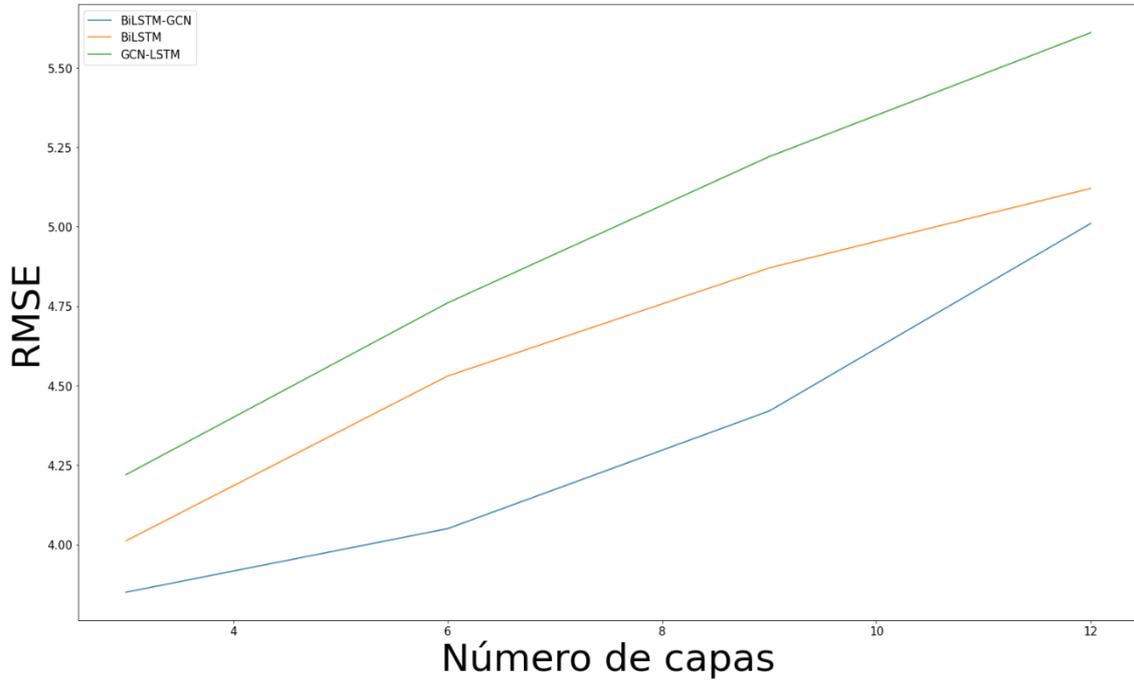
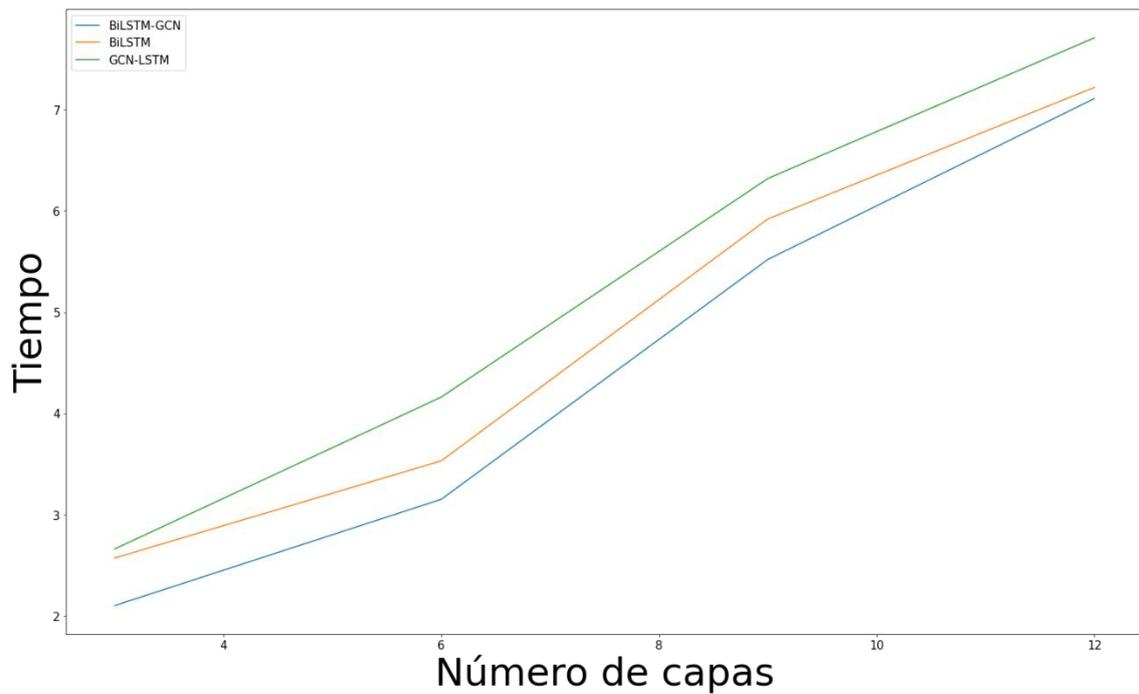


Ilustración 19. Comparativa tiempo respecto al número de capas



5.2.2 FUNCIÓN DE ACTIVACIÓN DEL MODELO BiLSTM-GCN

En la salida de una neurona se puede aplicar una función limitadora estableciendo así un valor mínimo que debe ser sobrepasado para que la información continúe a la siguiente neurona. Dicha función se conoce como función de activación. El principal objetivo de una red neuronal es la resolución de problemas complejos, por lo que generalmente las funciones de activación harán que los modelos sean no lineales. Las más utilizadas son:

- *Función identidad*: Es la combinación lineal que ocurre en el interior de una neurona, y se utiliza cuando la intención es captar patrones lineales, ya que la salida no es modificada.

$$g(z) = z \quad (5.2)$$

- *Función sigmoide*: Comúnmente utilizada como función de activación de la capa de salida en los modelos de clasificación binaria, transformará la salida numérica en un valor entre 0 y 1 que siga la distribución de Bernoulli buscando el pronóstico binario de dos categorías distintas.

$$g(z) = \frac{1}{1 + e^{-z}} \quad (5.3)$$

- *Función Tangente Hiperbólica (Tanh)*: Escala los valores de entrada a valores $[-1, 1]$, los valores más altos tenderán a 1 y los valores más bajos tienden de manera asintótica a -1 . Se trata de una función similar a la sigmoide, mata el gradiente y de uso en redes recurrentes.

$$g(z) = \frac{2}{1 + e^{-2z}} - 1 \quad (5.4)$$

- *Función Softmax*: Convierte las salidas de la red neuronal en probabilidades, el sumatorio de las salidas será igual a 1. La salida está acotada en valores entre 0 y 1.

$$g(z)_j = \frac{e^z}{\sum_{k=1}^K e^{z_k}} \quad (5.5)$$

- *Unidad Lineal Rectificada (ReLU)*: Es la función de activación más utilizada en las capas ocultas de las redes neuronales. Hahnloser et al. (2000) describieron este tipo de funciones para explicar el funcionamiento de circuitos de silicón. La función ReLU transformará en cero toda entrada que sea un número negativo. Zeiler et al. (2013) explican algunos de los motivos por los que seleccionar esta función en vez de la sigmoide, entre ellos la facilidad de optimización, rapidez de procesamiento y evitar el sobreajuste en el entrenamiento. Una desventaja de la función ReLU es la desactivación de neuronas, impidiendo el aprendizaje del resto de la red; por esta razón se diseñaron distintas alternativas como Leaky ReLU, PReLU y ELU, cuya descripción puede encontrarse en Pajares et al. (2021).

$$g(z) = \max \{0, z\} \quad (5.6)$$

Para el modelo propuesto en este trabajo, BiLSTM-GCN, se ha optado por la función *Tanh*.

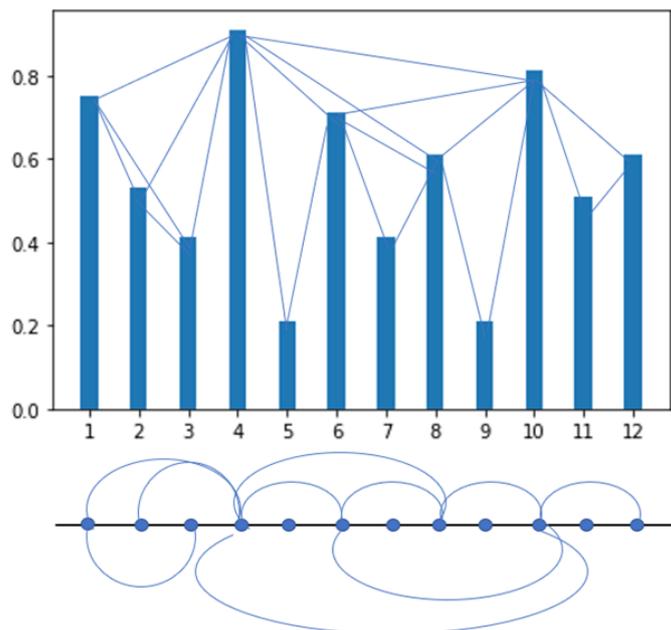
5.2.3 GRAFO DE VISIBILIDAD

El método utilizado para la transformación de las series temporales (que servirán como entrada al modelo) en grafos, es el *algoritmo de visibilidad* desarrollado por Lacasa et al. (2008), a partir del cual es posible transformar fácilmente una serie temporal en un grafo de visibilidad (del inglés *Visibility Graph*) con las características asociadas a la serie. Las series temporales periódicas darán como resultado grafos regulares, mientras que las series temporales aleatorias generarán grafos aleatorios. El método se puede definir formalmente como: dos datos arbitrarios (t_a, y_a) y (t_b, y_b) tendrán visibilidad y en consecuencia sus nodos estarán conectados en el grafo asociado si otro dato (t_c, y_c) ubicado entre ellos cumple la siguiente fórmula:

$$y_c < y_b + (y_a - y_b) \frac{t_b - t_c}{t_b - t_a} \quad (5.7)$$

La Ilustración 20 muestra los datos de una serie temporal representados en un grafo y las conexiones que se generan por el método de visibilidad y al aplicar la fórmula. El grafo se genera con las aristas correspondientes.

Ilustración 20. Ejemplo de una serie temporal con 12 datos y el grafo asociado mediante el algoritmo de visibilidad



A partir del grafo de visibilidad obtenido se calcula la matriz de correlación para ser introducida como entrada en el modelo. Dado que se trata de una serie univariada, la matriz resultante será igual a 1.

5.2.4 COMPLEJIDAD DEL MODELO BiLSTM-GCN

Para determinar la eficiencia de un algoritmo, es necesario tener en cuenta su complejidad computacional. La forma más común es usando la notación Big-O. Esta notación permite representar el límite superior de procesamiento de un algoritmo.

En las redes neuronales la complejidad viene determinada por las capas que la componen, por lo que la complejidad de la red será la suma de la complejidad de cada una de sus capas. La complejidad de los tres modelos de redes neuronales estudiados es la descrita en la Tabla 2.

Tabla 2. Complejidad de los modelos

Modelo	Complejidad del tiempo
BiLSTM	$O(2(4ih + 4h^2 + 3h + ho))$
GCN - LSTM	$O(LA0i + Lni^2) + O(W)$
BiLSTM - GCN	$O(ih + ho)$

L representa el número de capas, $A0$ el número de valores distintos de 0 en la matriz de adyacencia A , i el número de entradas y n el número de nodos en la capa.

Para la red tipo BiLSTM, la complejidad está definida por la determinada para redes LSTM duplicadas, esta es $W=(4dH + 4H^2 + 3H + Ho)$, i define el número de entradas, o el número de salidas y h el número de neuronas en la capa oculta.

En el modelo BiLSTM-GCN propuesto, los modelos se han entrenado previamente, por lo que la complejidad se define únicamente en términos de las capas densas incorporadas en los resultados.

En todos los casos se trata de una notación asintótica en la que el tiempo de cómputo aumentará linealmente en función de los datos ingresados; sin embargo, se evidencia una diferencia significativa en la complejidad del nuevo modelo propuesto, cuyo procesamiento es más sencillo.

6 EXPERIMENTACIÓN Y ANÁLISIS DE RESULTADOS

Este capítulo se organiza como sigue. En primer lugar, se muestran las herramientas empleadas: librerías, lenguaje y entorno de programación. Seguidamente, se detalla el preprocesamiento realizado sobre los datos, así como las distintas métricas utilizadas para evaluar el desempeño de la propuesta presentada y frente a otros modelos del estado del arte. Por último, se presentan los resultados obtenidos sobre datos procedentes de tres fuentes distintas: precios del petróleo en base al índice *West Texas Intermediate* (WTI), precios de las materias primas en base al índice de *Bloomberg Commodities Total Return* (BCTR) y precios de las materias primas denominadas *raras* sobre el índice *Rare Earth Elements Fund*.

6.1 HERRAMIENTAS EMPLEADAS

En la Tabla 3 se desglosan las herramientas software empleadas en el presente trabajo.

Tabla 3. Tabla de herramientas utilizadas en la investigación.

Librerías:	
Pandas 1.3.5	Manejo y análisis estructuras de datos
Matplotlib 1.21.6	Creación gráficos
Numpy 1.21.6	Cálculo numérico y análisis de datos
Sklearn 1.0.2	Análisis predictivo y clasificadores
Statsmodels 0.10.2	Funciones estadísticas
Keras 2.8.0	Librería redes neuronales

Seaborn 0.11.2	Librería elaboración gráficos
Scipy 1.7.3	Herramientas matemáticas
ts2vg 1.0.0	Generación de grafos a partir de series temporales
Networkx 2.6.3	Generación de grafos
Graphviz 0.10.1	Generación y visualización de grafos
Tensorflow 2.8.2	Computación numérica
Pmdarima 1.8.5	Pronósticos basados en el modelo ARIMA
Prophet 1.1	Herramienta para predicciones
Stellargraph 1.2.1	Librería predicción grafos
Lenguaje de programación:	
Python v 3.6.9	
Entorno de programación:	
Google Colab	

6.2 TRATAMIENTO DE LOS DATOS

6.2.1 DIVISIÓN DE LOS DATOS

En el desarrollo de los experimentos detallados en los siguientes apartados se ha llevado a cabo una división de los datos recogidos de cada una de las series temporales utilizadas en dos conjuntos: uno correspondiente a los datos que se utilizarán para el entrenamiento del modelo y que contendrá el 80 % del total de observaciones, y el otro 20 % para pruebas. El conjunto de entrenamiento se dividirá, a su vez, en dos subconjuntos: aplicando el 10% del total de datos a la validación del modelo, resultando una división para este conjunto del 70%-10% del total de datos utilizados en la investigación; lo que permitirá medir el aprendizaje del modelo y evaluar su desempeño.

Aunque generalmente la división del conjunto de datos se realiza en función de criterios arbitrarios, en este caso la división se realiza siguiendo el estudio realizado por Gholamy et al. (2018), en el que muestran que los mejores resultados se obtienen utilizando una división 80-20, evitando así un sobreajuste de la red por falta de datos de entrenamiento. Recientes estudios han demostrado la importancia de un buen criterio a la hora de realizar estas divisiones (Tokar y Johnson, 1999); en caso contrario no será posible evaluar el modelo de red neuronal (Maier y Dandy, 2000).

La necesidad de que los datos a utilizar en los conjuntos de entrenamiento y prueba cumplieran con las mismas propiedades estadísticas ha sido reconocida por varios autores en la literatura (Masters, 1993; Maier y Dandy, 2000). Otros estudios desarrollados por Braddock et al. (1998) y Tokar y Johnson (1999) empleaban métodos *ad-hoc* para garantizar que los datos de los dos conjuntos tuvieran las mismas propiedades. No fue hasta principios del siglo XXI cuando aparecieron enfoques sistemáticos para realizar esta división de los datos. Por ejemplo, Bowden et al. (2002) empleaba un algoritmo genético que permitía minimizar la diferencia entre las desviaciones estándar de los conjuntos, asegurando que las propiedades estadísticas de todos los conjuntos son similares, pero aun necesitando elegir la cantidad de datos perteneciente a los conjuntos de entrenamiento, prueba y validación.

6.2.2 NORMALIZACIÓN DE LOS DATOS

Las tareas para la predicción de series temporales económicas han supuesto un desafío al tratarse de datos de naturaleza no lineal y con un comportamiento altamente dinámico. Este tipo de predicciones tiene un importante impacto en el entorno socioeconómico por lo que es de vital importancia una buena definición del modelo de red neuronal a utilizar y un buen preprocesamiento de los datos.

La *normalización* es la técnica básica de preprocesamiento mediante la cual se ajustan los datos para hacer posible el aprendizaje; la precisión de las predicciones se debe en gran medida a la técnica de normalización empleada. Bhanja y Das (2018) analizaron el impacto de distintas técnicas de normalización en las predicciones llevadas a cabo por distintos modelos de redes neuronales, en la misma línea que la investigación realizada por Sola y Sevilla (1997), que también remarcaban la importancia de la normalización de los datos. Esta importancia viene dada por las posibles diferencias de escalas entre los datos de entrada, pudiendo aumentar la dificultad del problema a modelar. Un modelo que reciba valores grandes probablemente se convierta en un modelo inestable con un rendimiento deficiente durante el entrenamiento.

Una variable de salida con una gran variedad de valores puede dar como resultado valores de error de gradiente grandes, lo que conlleva que los pesos de las neuronas cambien drásticamente y convirtiendo el modelo en inestable. Una buena práctica consiste en

escalar las variables de entrada a valores pequeños, [0,1] o estandarizados con una media cero y una desviación estándar de uno (Lanzarini y Giusti, 2002).

Por todo ello, en este trabajo los datos se normalizaron al rango [0,1]. La normalización se realizó de la siguiente manera:

$$y = (x - \min) / (\max - \min) \quad (6.1)$$

Donde los valores mínimo y máximo pertenecen al valor x siendo este normalizado.

6.3 MÉTRICAS DE ERROR EMPLEADAS

Para la evaluación del desempeño de la propuesta combinada BiLSTM - GCN presentada en este trabajo, se han elegido las cuatro métricas de error más utilizadas: la *Raíz del Error Cuadrático Medio*, el *Error Cuadrático Medio*, el *Error Porcentual Absoluto Medio* y el *Coficiente de Determinación* o R^2 .

6.3.1 RAÍZ DEL ERROR CUADRÁTICO MEDIO

La Raíz del Error Cuadrático Medio (en inglés, *Root Mean Squared Error*, RMSE) es una medida de precisión de uso común entre los métodos de predicción, que estima la raíz del error cuadrático medio, siendo la diferencia entre los valores reales y los valores predichos. Siempre devolverá valores positivos y un resultado de 0 indicaría un ajuste perfecto del modelo, y puede ser interpretada como la varianza. Es utilizada para estimar los errores en las predicciones como en Valipour et al. (2013), donde se comparaba el rendimiento frente a modelos estadísticos clásicos, o en Almalaq y Edwards (2017) que evaluaban el rendimiento de distintos métodos de aprendizaje profundo a partir de esta métrica. Puede definirse como:

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (Y_t - \hat{Y}_t)^2} \quad (6.2)$$

6.3.2 ERROR CUADRÁTICO MEDIO

El Error Cuadrático Medio (en inglés *Mean Squared Error*, MSE) es el criterio de evaluación más utilizado en problemas de regresión. Calcula el promedio del cuadrado de los errores, esto es la diferencia cuadrática media entre el resultado obtenido y el esperado, siendo Y_t el valor real e \hat{Y}_t el valor obtenido. Medirá la calidad del modelo y será siempre un valor positivo. El error cuadrático medio disminuirá a medida que el error se acerque a cero. El resultado obtenido estará en la misma medida que se encuentre la serie temporal de entrada. Nakamura (2005) utilizaba el MSE obtenido en una red neuronal para la predicción de valores de inflación. Otros autores como Paletta y Lasenby (2020), Swanson y White (1997) y Zahra et al. (2014) buscaron además otros posibles estimadores además del MSE para evaluar el rendimiento de las redes neuronales. Matemáticamente se puede definir como:

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_t - \hat{Y}_t)^2 \quad (6.3)$$

6.3.3 ERROR PORCENTUAL ABSOLUTO MEDIO

El Error Porcentual Absoluto Medio (del inglés, *Mean Absolute Percentage Error*, MAPE) es un indicador de la precisión de las predicciones que mide el tamaño del error absoluto en términos porcentuales. Es un indicador de fácil interpretación incluso cuando no se conoce el volumen de la serie original. Frechtling (1996) afirmó que valores de MAPE entre 10% y 20% se consideran de buen pronóstico, entre 20% y 30% son aceptables, mientras que es difícil obtener menos del 10% debido a la naturaleza no lineal de las variables. MAPE se define como:

$$MAPE = \frac{\sum_{t=1}^n \frac{|A_t - F_t|}{|A_t|}}{n} \quad (6.4)$$

6.3.4 COEFICIENTE DE DETERMINACIÓN

El Coeficiente de Determinación (del inglés, *Coefficient of determination*), también denominado *R-cuadrado* (R^2) indica la aptitud del modelo. Explica la variabilidad entre variables y se mide en escala de menos infinito a 1, siendo 1 una predicción perfecta. El

valor de R^2 está estrechamente relacionado con MSE y también se utiliza para evaluar el rendimiento del modelo. Es la cantidad de variación en el atributo dependiente de la salida que es predecible a partir de las variables independientes de la salida. Cuanto más cerca de 1 sea este valor, mejor rendimiento tendrá la red. R^2 se define como:

$$R^2 = 1 - \frac{\sum_{i=1} (Y_t - \hat{Y}_t)^2}{\sum_{i=1} (Y_t - \bar{Y})^2} \quad (6.5)$$

6.4 PRECIOS DEL PETRÓLEO

En Estados Unidos, según Schurr et al. (1960), ha habido dos transiciones energéticas hasta que el petróleo crudo pasó a ser la mayor fuente de energía. La primera transición fue cuando el uso del carbón se impuso al de la madera como fuente de combustible en 1985 (con un 65% de carbón frente a un 30% de madera). La segunda transición ocurrió cuatro décadas y media después, relacionada con el petróleo y el gas, cuando el uso del carbón solo representaba el 28% y el petróleo y el gas el 65%.

Según la Agencia Internacional de la Energía, en el año 2019 el consumo de crudo en Estados Unidos fue de 20,54 millones de barriles por día (mbpd) incluyendo los que se corresponden a biocombustibles (1,1 mbpd), en 2020 se registró un descenso en el consumo debido a los confinamientos derivados de la pandemia de COVID-19, situándolo en los 17,4, mientras que los datos de 2021 lo sitúan en 18,6 mbpd. El consumo mundial es de 98,8 mbpd lo que ha convertido el petróleo en un activo clave para la mayoría de las economías mundiales.

Sin embargo, el cambio de mentalidad debido al calentamiento global, unido a la preocupación por la sostenibilidad de los recursos, ha supuesto una transformación en el sector energético que ha llevado a una notable disminución en el consumo de petróleo en los últimos años, generando entre las principales empresas petroleras una cierta preocupación que las ha llevado a estudiar cómo recuperar su posición competitiva (Heijnen et al., 2015).

6.4.1 TÉCNICAS DE PREDICCIÓN DE PRECIOS DEL PETRÓLEO

Debido al exponencial aumento del uso del petróleo como fuente para el desarrollo y crecimiento de las economías, un gran número de académicos han estudiado el comportamiento de los precios desde el punto de vista de la oferta y de la demanda (Monge et al, 2017a; Amano y Van Norden, 1998; Baumeister y Peersman, 2013). También hay muchos otros investigadores que han tratado de modelizar los precios del petróleo crudo (Bekiros y Diks, 2008; Bentzen, 2007) para poder entender su comportamiento y poder llevar a cabo predicciones.

Una de las primeras investigaciones sobre la predicción de precios del petróleo fue la realizada por Amano (1987), en la que llevaba a cabo técnicas econométricas clásicas para realizar pronósticos sobre el precio del petróleo. A finales del siglo XX, Gülen (1998) empleaba el índice WTI para realizar la predicción de los precios a partir de técnicas de cointegración. Con este mismo índice, Ye et al. (2002) adoptaban un modelo de regresión lineal simple para la predicción de los precios del petróleo. En el estudio llevado a cabo por Krane y Agerton (2015) explicaban cómo el tipo de pozo puede influir en la oferta del petróleo, tendiendo a ser más elástica cuando se trata de ciertos tipos de perforaciones. Siguiendo esa línea de investigación, Baffes et al. (2015) afirmaban que el notable aumento en la producción de petróleo en Estados Unidos y las nuevas técnicas para su extracción ha supuesto enormes cambios en la oferta de los últimos años. Baumeister y Kilian (2016) analizaban los distintos factores que pueden influir en la demanda de petróleo y en su precio, afirmando que existen diversos factores ajenos a la demanda que pueden influir en el precio del petróleo.

El índice *West Texas Intermediate* (WTI) se trata de un índice de referencia en EE.UU ya que se extrae en estados interiores del país, motivo por el cual existe un acceso restringido a los puertos de acceso. Al ser extraído y mezclado en EE.UU es necesario que los inversores presten especial atención a los cambios en el mercado estadounidense ya que la variación en la producción afectará directamente a este índice.

El índice WTI fue utilizado por Das et al. (2018) para llevar a cabo un estudio sobre la influencia de los precios del petróleo en el crecimiento económico de EE.UU. Monge y Gil-Alana (2015) analizaron cómo el precio del petróleo puede ejercer influencia en el mercado de fusiones y adquisiciones a partir del uso de técnicas de integración y

cointegración gradual. Además del estudio llevado a cabo por Monge et al. (2017b) en el que analizan el precio del petróleo en base al índice WTI utilizando técnicas *wavelet*.

Kaboudan (2001) y Rast (2001) fueron de los primeros investigadores en introducir el uso de redes neuronales artificiales (ANN) para la predicción de precios del petróleo. Mirmirani y Li (2004) aplicaron técnicas de vector autorregresivo (VAR) y de redes neuronales para hacer los pronósticos del precio del petróleo, obteniendo mejores resultados en la técnica de ANN mediante la comparación del RMSE. Xie et al. (2006) aplicaron el uso de SVM para su análisis de los precios WTI entre enero de 1970 y diciembre de 2003, realizando la comparación con los resultados obtenidos por el método ARIMA y obteniendo también un menor RMSE con la técnica SVM frente al modelo estadístico clásico. Yu et al. (2008) emplearon un modo empírico de descomposición basado en los paradigmas de redes neuronales con el objetivo de realizar la predicción de los precios WTI a partir de la descomposición de la serie temporal en una serie de funciones pequeñas. Mientras que Baruník y Malinska (2016) propusieron un modelo de regresión basado en redes neuronales para su predicción.

Parte de esta literatura relacionada se encuentra recogida en la publicación de Hamdi y Aloui (2015), en la que llevaron a cabo un estudio de las publicaciones que analizan los precios del petróleo mediante el uso de ANN.

6.4.2 DATOS UTILIZADOS

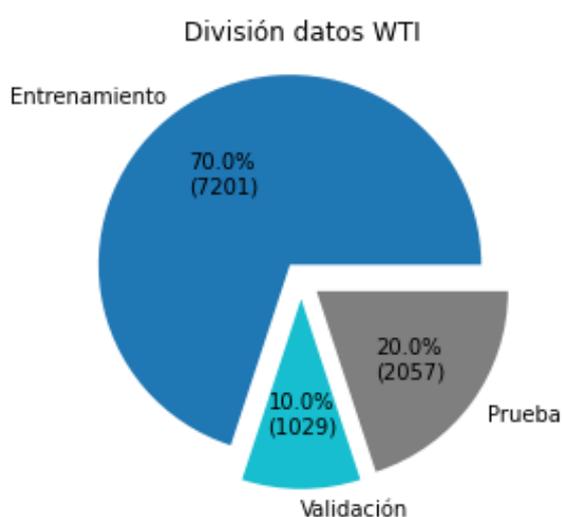
En este apartado se realizará una descripción de los datos utilizados en el presente experimento. En concreto, se trabajó con la serie temporal correspondiente al índice de precios del crudo *West Texas Intermediate* (WTI) obtenido de *Thomson Eikon Reuters* con una periodicidad diaria con datos del 10 de enero de 1983 al 15 de junio de 2022. WTI es el precio al contado del petróleo de grado intermedio de *West Texas*; es junto con el precio spot del Brent uno de los principales puntos de referencia para fijar el precio del petróleo.

El pronóstico se ha basado en las 10.289 observaciones resultantes en el periodo de tiempo especificado, y cargado en un *DataFrame* de Python generado a partir de la librería Pandas para poder llevar a cabo el procesamiento de los datos. En la Ilustración

21 se observa la distribución de los datos en los conjuntos de entrenamiento (70%), validación (10%) y prueba (20%).

Una vez estructurados, se ha procedido a la generación del grafo asociado utilizando el método del grafo de visibilidad desarrollado por Lacasa et al. (2008), y descrito en el capítulo 5 (apartado 5.2.3). Obtenido el grafo correspondiente a los datos, se realiza el cálculo de la matriz de correlación que será utilizada como entrada en el modelo *StellarGraph*. Al tratarse de una serie univariada, la matriz resultante será igual a 1.

Ilustración 21. División de los datos WTI en los conjuntos de entrenamiento, validación y prueba



6.4.2.1 ESTUDIO DE LA SERIE TEMPORAL WTI

Para un correcto procesamiento de los datos se realiza un análisis de la serie temporal objeto del pronóstico. Se realizará un estudio acerca de la estacionariedad y tendencia de los datos originales que serán utilizados para realizar las predicciones. La Ilustración 22 muestra cómo se representa la serie temporal procedente de los datos de WTI. El cálculo de la desviación estándar y la media móvil de 30 días, en contraposición con la serie original, se muestran en la Ilustración 23.

Para la comprobación de la estacionariedad de la serie, consistente en la ausencia de tendencias y patrones estacionales, se emplean el test aumentado de Dickey Fuller y el test y Phillips Perrón (descritos en el capítulo 3, sección 3.2.2) para verificar la estacionariedad de la serie temporal. Se generarán valores críticos y un p -valor que permitirá aceptar o rechazar la hipótesis nula, que establece que no existe estacionariedad.

Ilustración 22. Representación de la serie temporal WTI

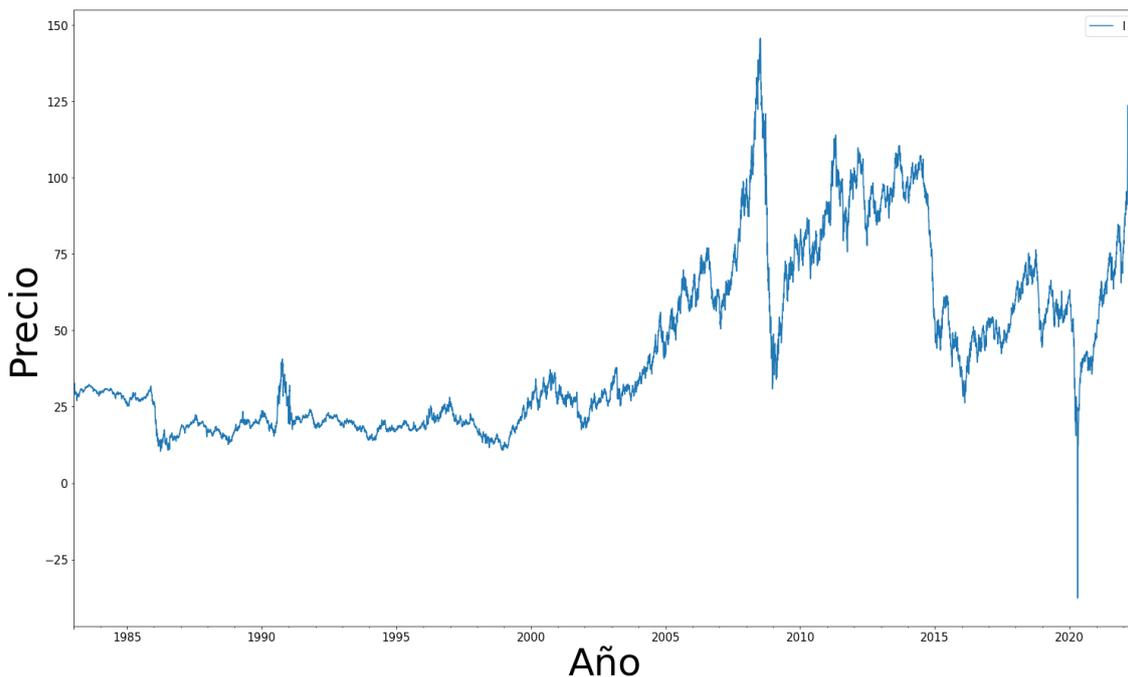
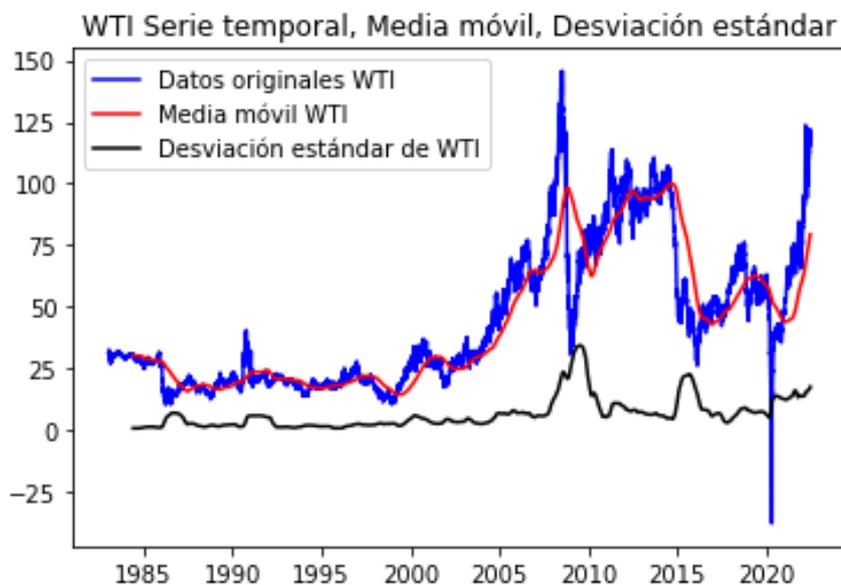


Ilustración 23. Representación de la media móvil y la desviación estándar WTI



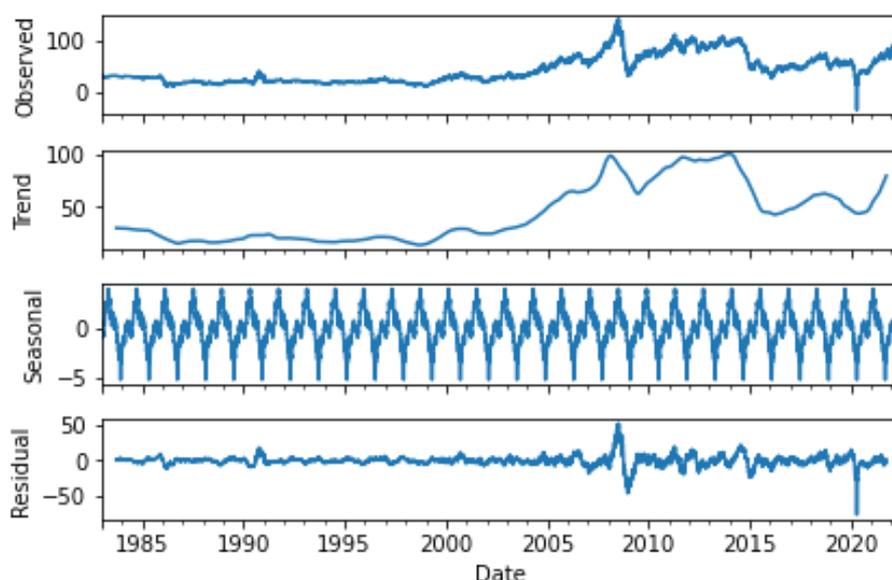
El criterio de información de Akaike, también descrito en el capítulo 3 (sección 3.2.3) será el empleado para la elección del mejor modelo ARIMA a emplear. Los resultados obtenidos se reflejan en la Tabla 4.

Tabla 4. Resultados Serie Temporal WTI

Valores ADF	Valores PP	Métrica
-1,753	-1,451	Test Estadísticos
0,404	0,558	<i>p</i> -valor
10248,000	10248,000	Número de observaciones
-2,862	-2,860	Valor crítico (5%)

La serie no es estacionaria dado que el *p*-valor es mayor al 5% y el valor estadístico de prueba es mayor que el valor crítico. Para confirmar gráficamente estos resultados se realiza la descomposición de la serie en la Ilustración 24.

Ilustración 24. Descomposición de la serie WTI



6.4.3 RESULTADOS

En la Tabla 5 se muestran los resultados obtenidos en la predicción de precios del petróleo en base al índice WTI con el modelo propuesto BiLSTM-GCN. Con fines comparativos, se ha evaluado también el rendimiento de los siguientes modelos, tanto clásicos como basados en ANN, y descritos en los capítulos 2 y 3: ARIMA, PROPHET, BiLSTM y GCN-LSTM inspirado en el modelo desarrollado por Zhao et al. (2019). Los parámetros ARIMA (p,d,q) utilizados en este estudio han sido (1,1,1).

El mejor rendimiento es el obtenido por el modelo BiLSTM-GCN en base a todas las métricas utilizadas (descritas en el apartado 6.3) y el tiempo, mostrando una clara efectividad para el pronóstico de los precios del petróleo del índice WTI.

Tabla 5. Resultados obtenidos por el modelo BiLSTM-GCN en la predicción de precios del petróleo WTI, frente a ARIMA, PROPHET, BiLSTM y GCN-LSTM.

	MSE	RMSE	R²	MAPE	Tiempo
ARIMA	81,219	9,012	0,716	22,500	5,23 s
PROPHET	134,050	11,580	0,834	14,380	6,43 s
BiLSTM	16,100	4,012	0,951	7,750	2,57 s
GCN-LSTM	17,829	4,220	0,947	7,630	2,66 s
BiLSTM-GCN	15,610	3,850	0,955	7,410	2,10 s

En la tabla se refleja además cómo los modelos basados en ANN tienen un mejor rendimiento y precisión para la predicción de esta serie temporal, que los basados en técnicas estadísticas tradicionales, obteniendo unos términos de error significativamente menores. Para la métrica de error RMSE, la propuesta presentada reduce el error en un 57,3% respecto a la técnica estadística ARIMA tradicional, y en un 65,5% respecto al modelo PROPHET. Mientras que para los modelos DNN, esto es BiLSTM y GCN-LSTM, es del 4,07% y del 8,7%, respectivamente.

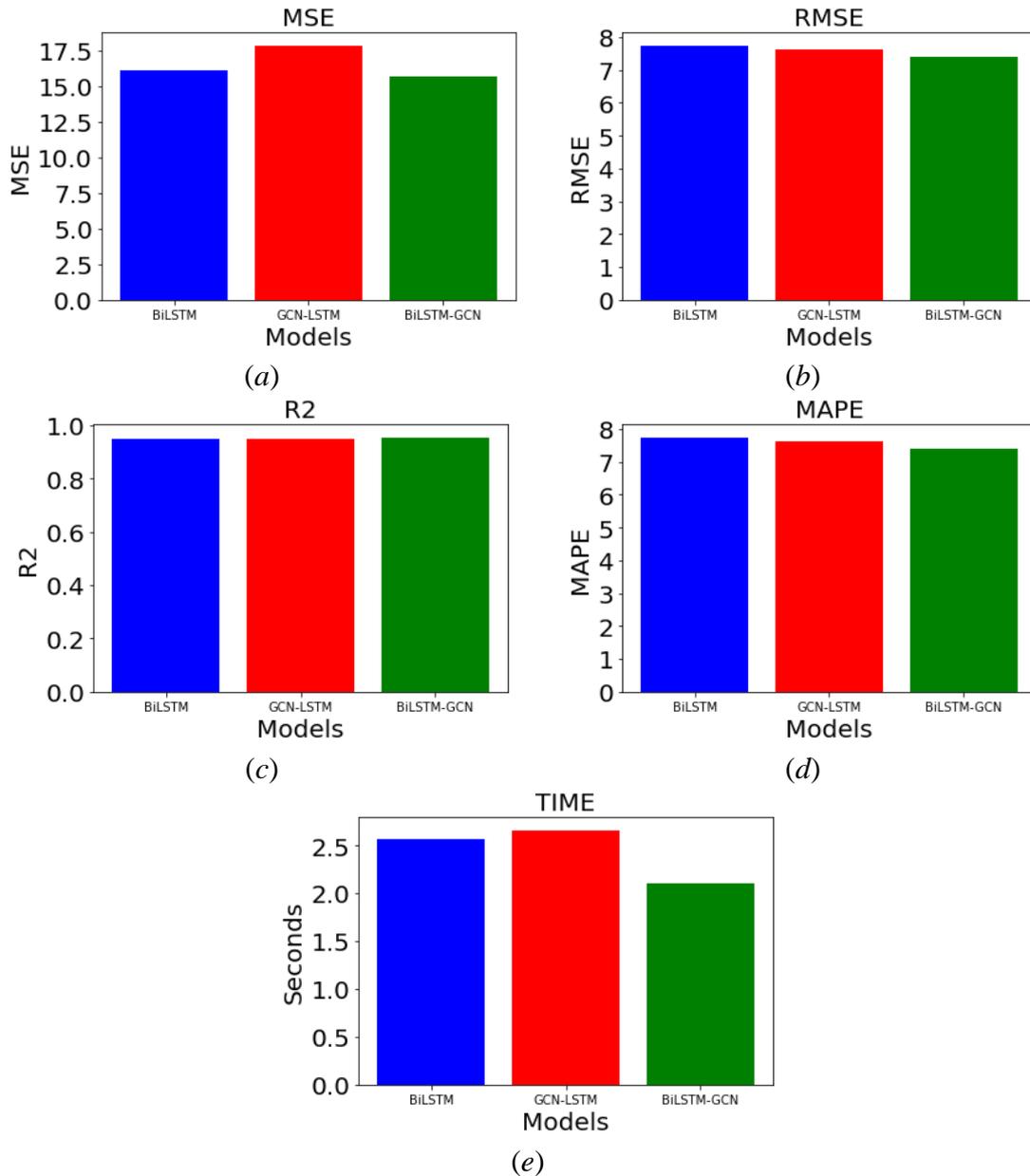
Estos resultados se deben principalmente a que métodos como ARIMA y PROPHET tienen dificultades en el tratamiento de grandes series temporales complejas y no estacionarias, incluso siendo diferenciadas para hacerlas estacionarias. Esto se debe a que cuando el número de observaciones pertenecientes a la serie es superior a 10.000 pueden derivarse problemas en los test estadísticos. Cuando se trata de un número elevado de muestras los p -valores disminuirán rápidamente a valores de 0, no aportando resultados fiables sobre las características estadísticas (Lin et al., 2013).

Por otro lado, las redes de tipo GCN-LSTM consideran las características espaciales, pero no tienen en cuenta las temporales, lo que explica el peor rendimiento de este modelo en comparación con el propuesto en este trabajo de investigación, que mediante la combinación de modelos aporta las ventajas subyacentes de ambos tipos de ANN permitiendo aumentar la precisión y minimizar el error de las predicciones. Esta mejora también se ve reflejada en los tiempos de ejecución debido al uso de dos modelos preentrenados y un menor número de capas, lo que permite un procesamiento más rápido incluso con una gran cantidad de datos.

La Ilustración 25 permiten comparar también de forma visual el rendimiento de los modelos utilizados en esta investigación. Los resultados obtenidos por los modelos

ARIMA y PROPHET generaban una distorsión en la gráfica por lo que se ha optado por no incluirlos.

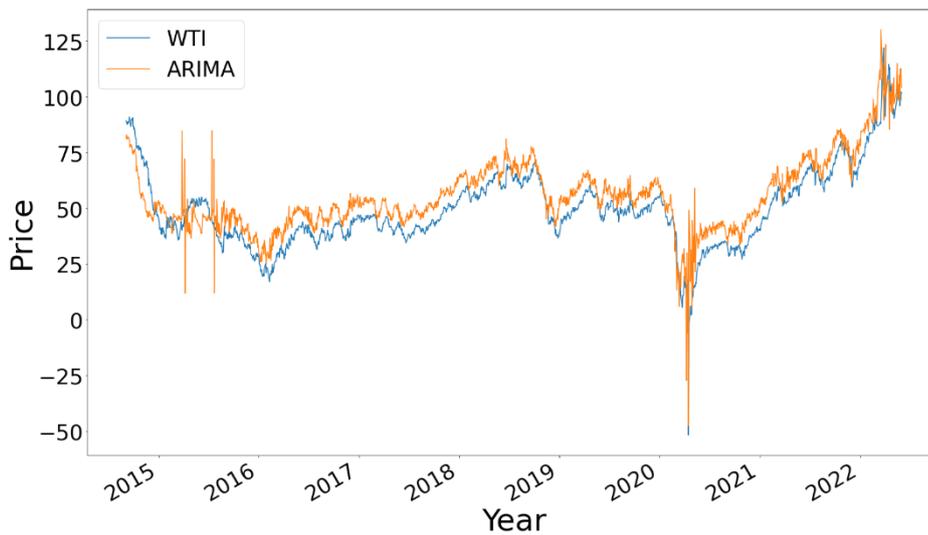
Ilustración 25. Métricas de resultados WTI: (a) MSE; (b) RMSE; (c) R2; (d) MAPE; y (e) Tiempo.



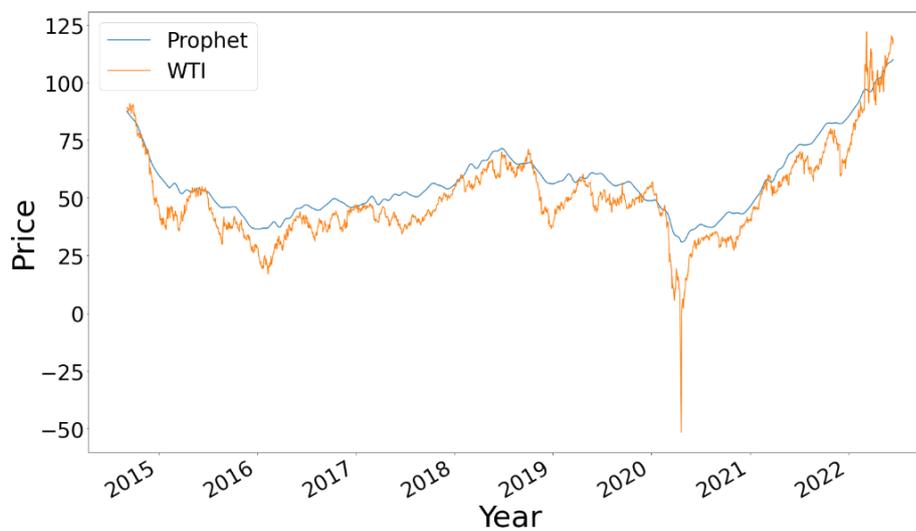
Con el objetivo de destacar la diferencia entre los resultados obtenidos por los diferentes modelos y mejorar la visualización de las predicciones realizadas, la Ilustración 26 muestra cómo se ajustan los modelos a los datos utilizados en la serie temporal a estudiar en el conjunto de prueba, con el 20% de los datos totales recogidos en la serie temporal. Los modelos basados en DNN muestran valores muy cercanos a los reales del índice WTI, por lo que para reflejar de forma más clara las diferencias entre los modelos BiLSTM, GCN-LSTM y BiLSTM-GCN en la Ilustración 27 se muestra la comparación de sus

resultados con los datos correspondientes al año 2020, en donde se pone de manifiesto el mejor rendimiento del modelo combinado propuesto, BiLSTM-GCN.

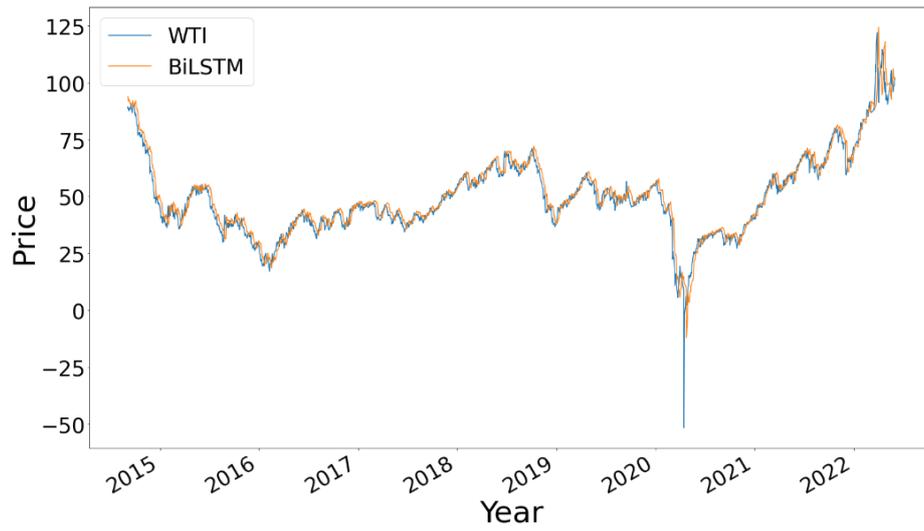
Ilustración 26. Ajuste de los resultados obtenidos frente al valor real del índice WTI por: (a) ARIMA; (b) PROPHET; (c) BiLSTM; (d) GCN-LSTM; y (e) BiLSTM-GCN



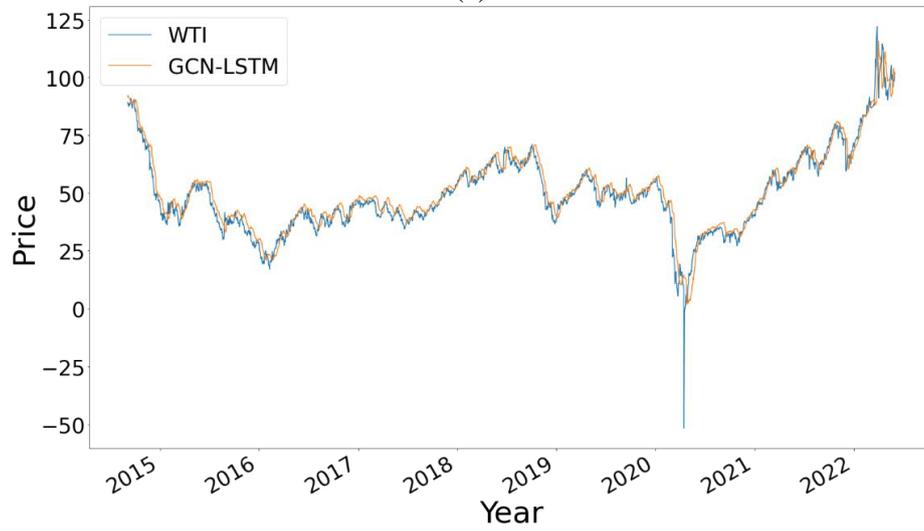
(a)



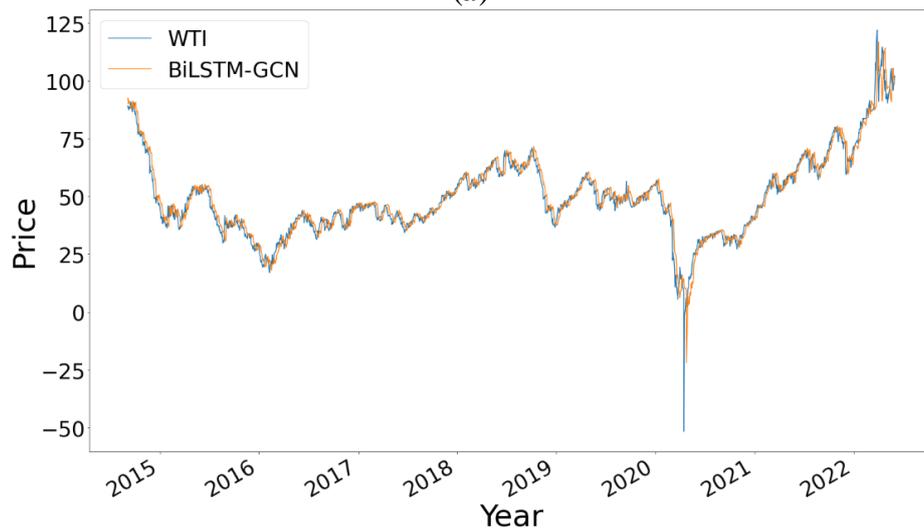
(b)



(c)

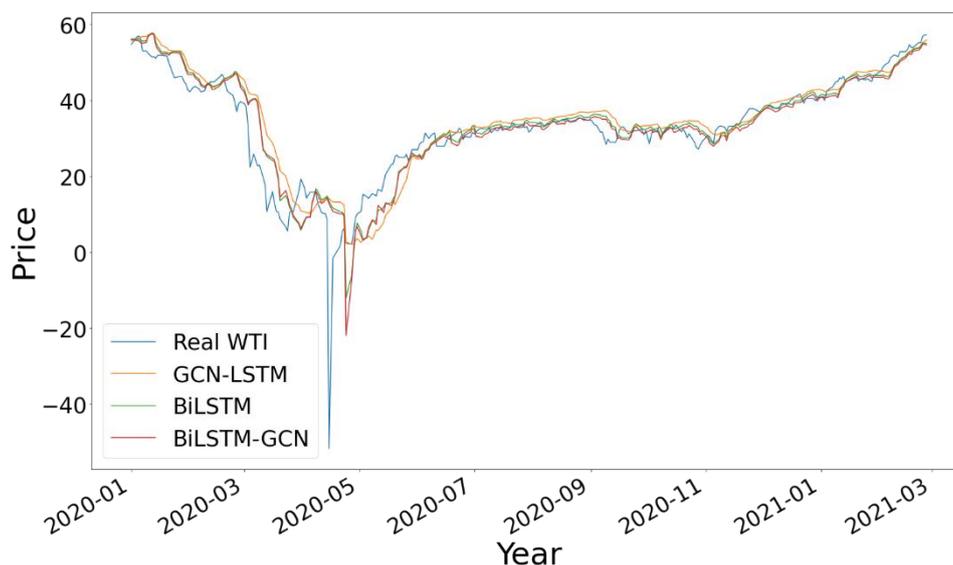


(d)



(e)

Ilustración 27. Ajuste de los resultados obtenidos por BiLSTM, GCN-LSTM y BiLSTM-GCN frente al valor real del índice WTI en el año 2020.



Para verificar si existe una diferencia estadísticamente significativa, y con ello corroborar la eficiencia de la red neuronal, se realiza la prueba de Friedman (1940), a través de la cual se puede confirmar si se rechaza la hipótesis nula, que establece que el promedio de cada una de las predicciones realizadas con los diferentes métodos es igual a las demás (ver Tabla 6), y que consiste en la comparación de las medianas de dos grupos de datos. Se aplica la prueba de Friedman a los resultados obtenidos por los modelos de redes neuronales, ya que estos son los resultados más precisos.

Tabla 6. Test de Friedman serie WTI

Valor estadístico	280,303
p-valor	$1,821 \times 10^{-60}$

Los resultados mostrados en la prueba, incluyendo todos los modelos utilizados en el experimento, reflejan un *p-valor* menor a 0,05, por lo que podemos rechazar la hipótesis nula, teniendo evidencia suficiente para concluir que las medias de las predicciones realizadas tienen diferencias significativas.

Dado que el resultado de la prueba de Friedman es significativo, podemos concluir que al menos dos de los grupos comparados son significativamente diferentes, pero no es posible saber cuáles. Para lo cual se aplica la prueba de Wilcoxon (Rosner et al., 2006)

entre cada par de grupos con los siguientes resultados, Tabla 7. Los resultados muestran como el único par de datos que nos permite rechazar la hipótesis nula es el correspondiente a los datos originales de la serie WTI con las predicciones obtenidas por el modelo BiLSTM-GCN, corroborando una mayor precisión en los resultados de este modelo.

Tabla 7. Test Wilcoxon serie WTI

Datos comparados	Valor estadístico	<i>p</i>-valor
WTI y BiLSMT	19441,0	0,024
WTI y GCN-LSTM	14162,0	$9,591 \times 10^{-9}$
WTI y BiLSTM-GCN	21371,0	0,322

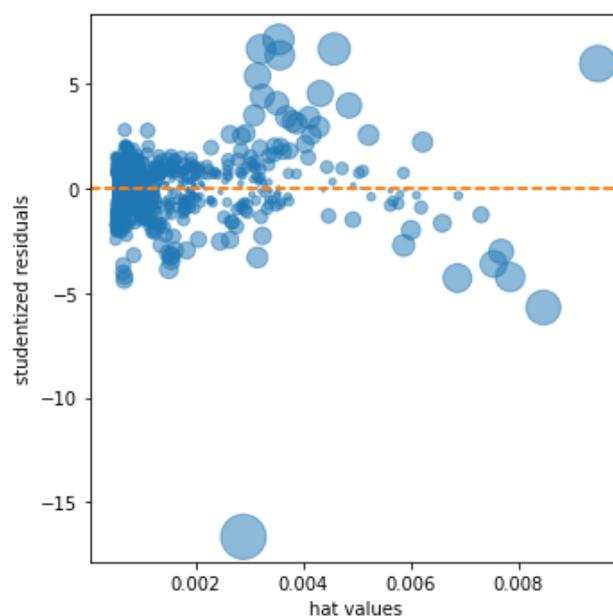
Para proporcionar un análisis detallado del error del modelo, se llevan a cabo una serie de pruebas estadísticas y representaciones visuales para confirmar la eficiencia del modelo. Se decide calcular el valor de F-Estadístico y su Prob (F-estadístico), cuya hipótesis nula establece que todos los parámetros utilizados en la regresión son 0 y que no ayuda a explicar la variable dependiente; en este caso no encontraría una relación entre el valor real de la serie temporal y la predicción. El valor resultante es muy pequeño, por lo que se rechaza la hipótesis nula, como se muestra en la Tabla 8.

Tabla 8. Valores estadísticos resultados WTI

Valores	Valor estadístico
Curtosis	47,72
F – Estadístico	81,14
Prob (F-Estadístico)	$1,03 \times 10^{-18}$

La medida de Curtosis resultante mide la concentración más alta o más baja de los datos alrededor de la media; un coeficiente positivo indicará una concentración más alta de los datos alrededor de la media, mientras que un coeficiente negativo reflejará una concentración más baja. El resultado del modelo propuesto refleja una alta concentración alrededor de la media; este resultado se sustenta gráficamente con el resultado de reflejar la gráfica de influencia de la Ilustración 28, en la que se muestran los residuos estudentizados junto con los valores *hat* de especial interés para detectar valores atípicos en las predicciones. El gráfico muestra una pequeña cantidad de valores atípicos en relación con la cantidad de observaciones estudiadas.

Ilustración 28. Valores atípicos



El error acumulado de las 10.289 observaciones, que consiste en la suma de los errores de todas las predicciones, tiene un resultado de 871,31, lo que lo sitúa en un error medio de 0,084 para cada observación.

6.5 PRECIOS DE LAS MATERIAS PRIMAS RARAS

Las materias primas raras (en inglés *rare-earth elements*, REE), son un grupo de metales que son críticos para una economía más verde, siendo las materias primas críticas (del inglés *Critical Raw Materials*, CRM) las creadoras de valor industrial, con un efecto significativo en los sectores intermedios. Son menos del 20% de todos los elementos que ocurren naturalmente en el medio ambiente (Klinger, 2015).

En concreto, lo forman el siguiente conjunto de 17 elementos químicos: Lantano (La), Cerio (Ce), Praseodimio (Pr), Neodimio (Nd), Prometio (Pm), Samario (Sm), Europio (Eu), Gadolinio (Gd), Terbio (Tb), Disprosio (Dy), Holmio (Ho), Erblio (Er), Tulio (Tm), Iterbio (Yb), Lutecio (Lu), Escandio (Sc) e Itrio (Y).

La búsqueda e identificación de elementos de tierras raras constituyó gran parte del desarrollo de la ciencia y la tecnología a finales del siglo XIX y principios del XX. La mayor dificultad se debió a que las propiedades físicas y químicas de los diferentes elementos son bastante similares, por lo que los métodos de separación y purificación

dependían de laboriosas técnicas de cristalización y precipitación fraccionada, siendo agravada por la falta de métodos para la identificación y evaluación de la pureza del elemento.

El cambio climático está propiciando el auge de energías renovables como la eólica, solar, hidroeléctrica y geotérmica. Sin embargo, la escasez de estos recursos lleva a la búsqueda de nuevas fuentes energéticas y con características menos contaminantes. Esta búsqueda de energías limpias ha propiciado el aumento de la demanda de materias primas raras, usadas en distintos ámbitos como el militar, la electrónica o la automoción.

El 97% de la producción de estos materiales se concentra en China (Massari y Ruberti, 2013) a pesar de su uso generalizado en todo el mundo, suponiendo un riesgo político y económico, a la vez que implica posibles problemas de suministro que impidan los objetivos de descarbonización.

Según la Organización para la Cooperación y el Desarrollo Económicos (OCDE) la demanda mundial del REE podría aumentar de los 79.000 millones de toneladas actuales a 167.000 millones de toneladas en 2060, generando una dependencia de este tipo de materiales. El bloqueo de REE llevado a cabo por China durante la pandemia producida por el COVID-19 ha supuesto un notable incremento de precios, que unido al aumento de la demanda para satisfacer la producción de vehículos eléctricos, podría provocar un incremento aún mayor en los productos finales y en la inestabilidad económica mundial.

Es difícil prever cómo podría responder el mercado global al exceso de oferta y precios más bajos de REE, dado que la producción de estos minerales es casi siempre complementaria. Si la demanda de REE continúa creciendo como se espera, aumentará la presión sobre las naciones para aumentar la producción nacional a expensas del medio ambiente.

6.5.1 TÉCNICAS DE PREDICCIÓN DE PRECIOS DE LAS MATERIAS PRIMAS RARAS

Es posible encontrar en la literatura recientes investigaciones acerca del comportamiento y predicciones de los precios de materias primas raras mediante el uso de técnicas estadísticas y predictivas. En el estudio llevado a cabo por García et al. (2018) aplicaban el modelo ARIMA para realizar la predicción de los REE con el objetivo de confirmar un

posible ciclo de precios de materias primas raras. Siguiendo esta línea de investigación, Proelss et al. (2020) realizaban una descripción de la volatilidad en los precios de los REE a partir de un profundo análisis en el que empleaban los modelos ARMA, ARFIMA y GARCH, con una precisión superior en el modelo ARFIMA.

El auge de las redes neuronales artificiales aplicadas a la predicción de precios ha provocado un aumento en el número de investigaciones, incluyendo los pronósticos de precios de materias primas. Lasheras et al. (2017) realizaron predicciones de cada una de las materias primas raras con aprendizaje profundo, aplicando un algoritmo híbrido y SVM para aumentar la precisión en sus predicciones. Más recientemente, Bian et al. (2021) emplearon un algoritmo de redes neuronales *backpropagation* para predecir los precios de los REE a lo largo de la pandemia por COVID-19.

6.5.2 DATOS UTILIZADOS

Para este estudio se ha utilizado el *Rare Earth Elements Fund* sobre los REE para realizar el análisis con datos desde el 14 de junio de 2004 hasta el 15 de junio de 2022 con periodicidad diaria. Los datos han sido extraídos de la base de datos Eikon de Thomson Reuters.

El pronóstico se realiza sobre las 4.699 observaciones recogidas en el intervalo señalado y son importadas a un *Dataframe* en Python generado con la librería Pandas, que permitirá la manipulación y transformación de los datos. En la Ilustración 29 se recoge la representación de la distribución de los datos en los conjuntos de entrenamiento (70%), validación (10%) y prueba (20%).

Una vez estructurados se ha procedido a la generación del grafo asociado utilizando el método del grafo de visibilidad desarrollado por Lacasa et al. (2008), y descrito en el capítulo 5 (apartado 5.2.3). Obtenido el grafo correspondiente a los datos, se lleva a cabo el cálculo de la matriz de correlación que será utilizada como entrada en el modelo *StellarGraph*. Al tratarse de una serie univariada la matriz resultante será igual a 1.

6.5.2.1 ESTUDIO DE LA SERIE TEMPORAL REE

Se realiza un análisis sobre la serie temporal procedente de los datos de REE para llevar a cabo el correcto preprocesamiento de la información y la valoración de la

estacionariedad y la tendencia de la serie temporal. La Ilustración 30 muestra los datos originales de la serie temporal.

Ilustración 29. División datos REE en los conjuntos de entrenamiento, validación y prueba.

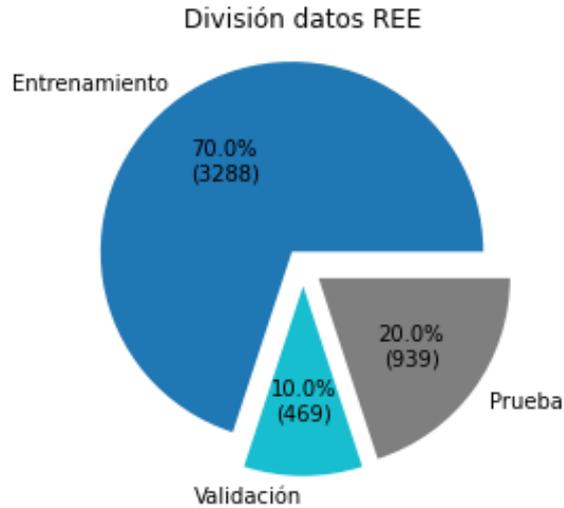
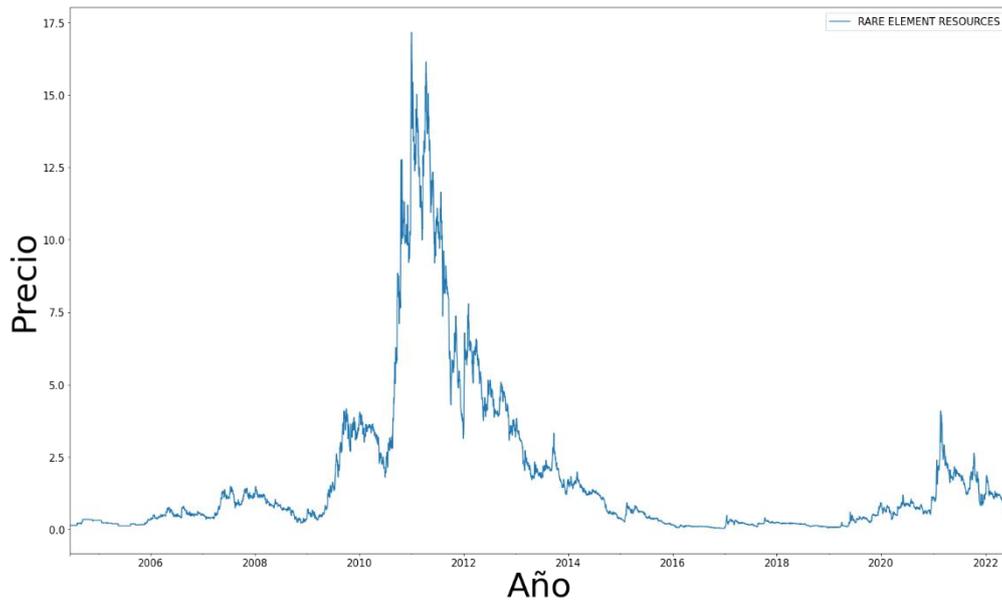
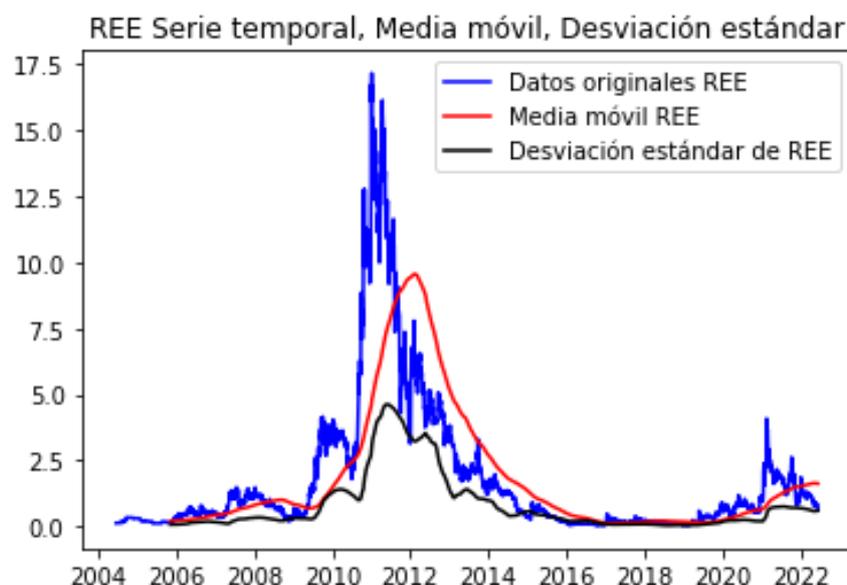


Ilustración 30. Representación de la serie temporal REE



La Ilustración 31 refleja el cálculo de la desviación estándar y la media móvil respecto a 30 días.

Ilustración 31. Representación de la media móvil y la desviación estándar REE



El test aumentado de Dickey Fuller y el test Phillips Perrón (descritos en el capítulo 3, sección 3.2.2) son empleados para comprobar la estacionariedad de la serie, si tiene tendencia o sigue patrones estacionales. Como resultado se obtendrán los valores críticos y el p -valor a partir del cual se podrá valorar si se acepta o rechaza la hipótesis nula. La Tabla 9 refleja los resultados obtenidos en el test.

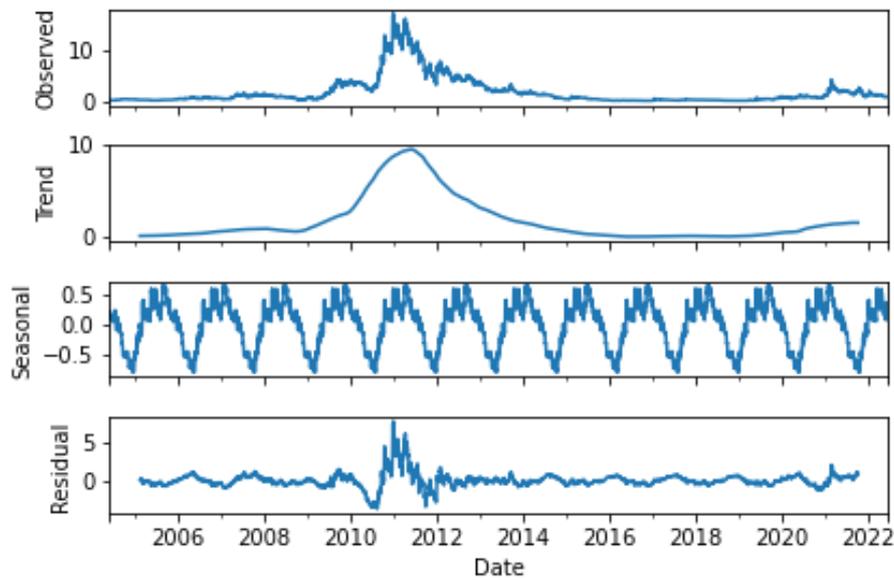
Tabla 9. Resultados Serie Temporal REE

Valores ADF	Valores PP	Métrica
-1,910	2,103	Test Estadístico
0,327	0,243	p -valor
4665,000	4665,000	Número de observaciones
-2,862	-2,861	Valor crítico (5%)

Se acepta la hipótesis nula ya que se trata de una serie no estacionaria, dado que el p -valor es mayor al 5% y el valor estadístico de prueba es mayor que el valor crítico. Para confirmar gráficamente estos resultados, en la Ilustración 32 se realiza la descomposición de la serie.

Para la elección del modelo ARIMA a emplear se recurre al criterio de información de Akaike, descrito en el capítulo 3, sección 3.2.3.

Ilustración 32. Descomposición de la serie REE



6.5.3 RESULTADOS

En la Tabla 10 están reflejados los resultados obtenidos tras realizar la predicción de precios del índice *Rare Earth Elements Fund* de REE con los mismos modelos utilizados en el estudio anterior: ARIMA, PROPHET, BiLSTM y GCN-LSTM (todos ellos descritos en los capítulos 2 y 3) frente al modelo BiLSTM-GCN propuesto en este trabajo. El modelo BiLSTM-GCN ha sido el que ha obtenido los mejores resultados en las métricas de error recogidas en la Tabla 10, mostrando unos resultados con mayor precisión que el resto de los modelos, bajo el conjunto de datos de prueba, para el índice de precios de REE. En este estudio los parámetros ARIMA (p,d,q) utilizados han sido $(5,2,7)$.

Tabla 10. Resultados obtenidos por el modelo BiLSTM-GCN en la predicción de precios de REE, frente a ARIMA, PROPHET, BiLSTM y GCN-LSTM.

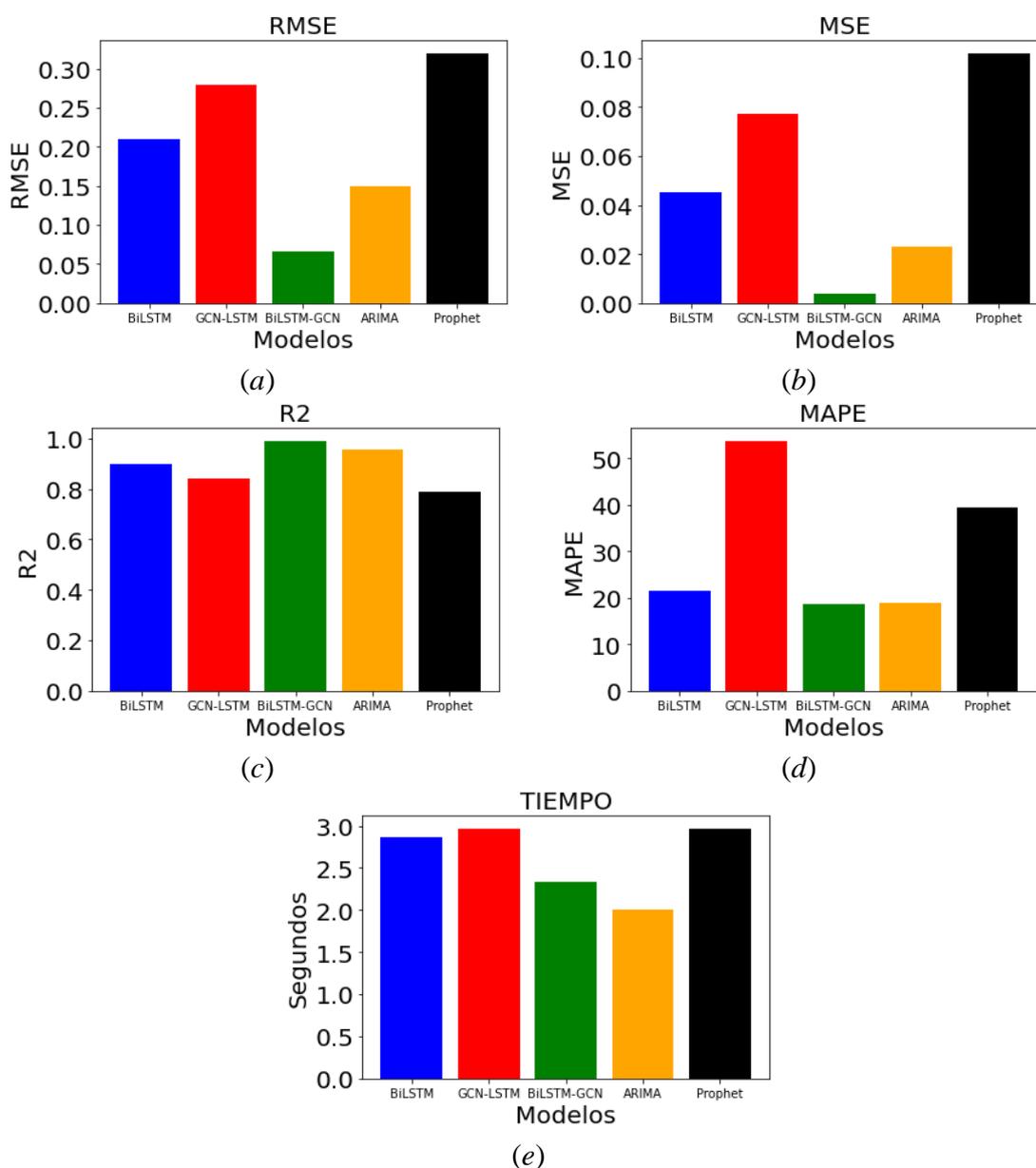
	MSE	RMSE	R²	MAPE	Tiempo
ARIMA	0,0233	0,152	0,954	18,812	2,00 s
PROPHET	0,102	0,320	0,794	39,430	2,96 s
BiLSTM	0,0452	0,211	0,901	21,423	2,86 s
GCN-LSTM	0,0770	0,284	0,840	53,832	2,97 s
BiLSTM-GCN	0,004	0,066	0,991	18,601	2,33 s

Los resultados ponen de manifiesto el mejor rendimiento del modelo híbrido propuesto en este trabajo de investigación, tan solo mostrando un resultado inferior en el tiempo de

ejecución frente al modelo ARIMA. Las métricas de error son significativamente menores en el modelo BiLSTM-GCN que en el resto de los modelos estudiados, mostrando que para el RMSE reduce el error en un 96% respecto al modelo PROPHET, en un 82% frente al modelo ARIMA y respecto a los modelos de redes neuronales, un 94,8% con el modelo GCN-LSTM y un 91% del modelo BiLSTM.

Estos resultados son acordes con la literatura existente, ya que con una cantidad de 5.000 observaciones el rendimiento del modelo ARIMA es mayor respecto a los modelos basados en redes neuronales, los cuales mejoran su eficiencia cuanto mayor es la cantidad de registros a estudiar.

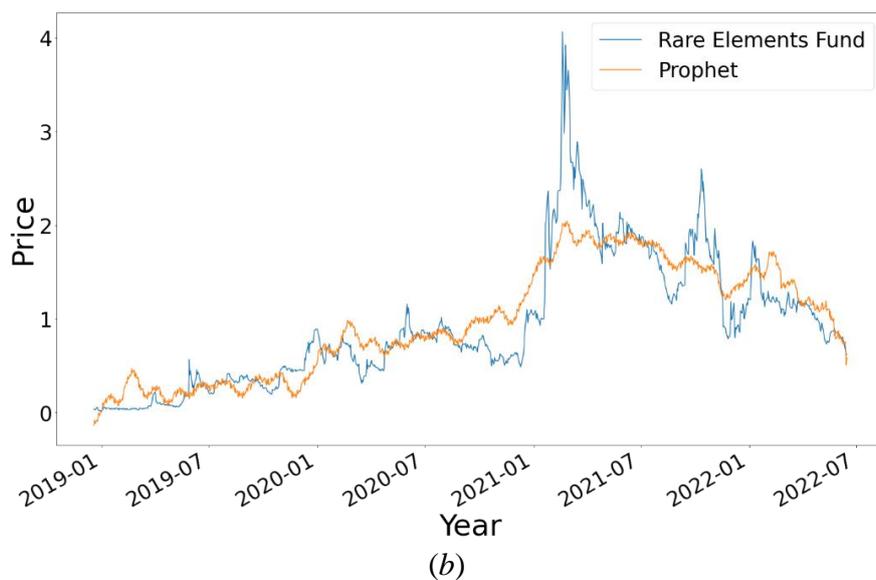
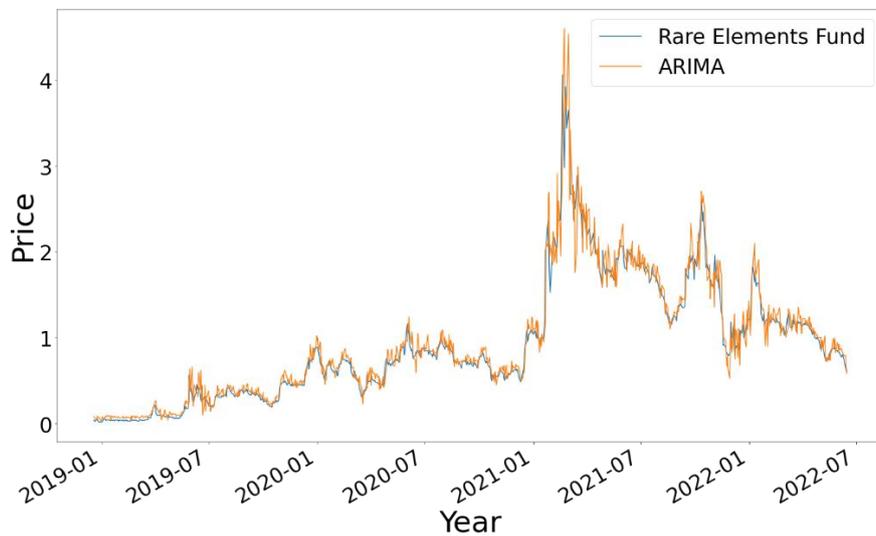
Ilustración 33. Métrica de resultados REE: (a) MSE; (b) RMSE; (c) R2; (d) MAPE; y (e) Tiempo.

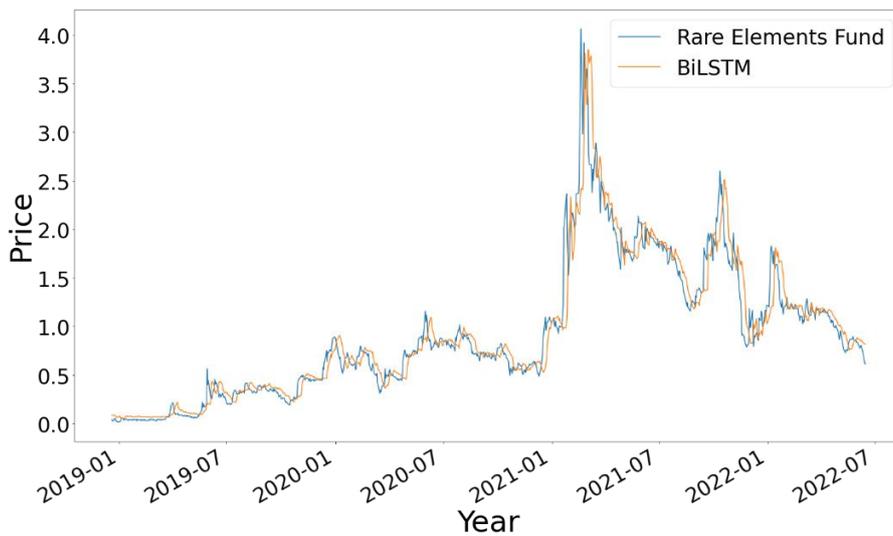


La Ilustración 33 permite comparar también de forma gráfica los rendimientos de los modelos estudiados en función de todas las métricas propuestas en este trabajo, con el objetivo de mostrar de forma visual los resultados obtenidos por cada uno de ellos.

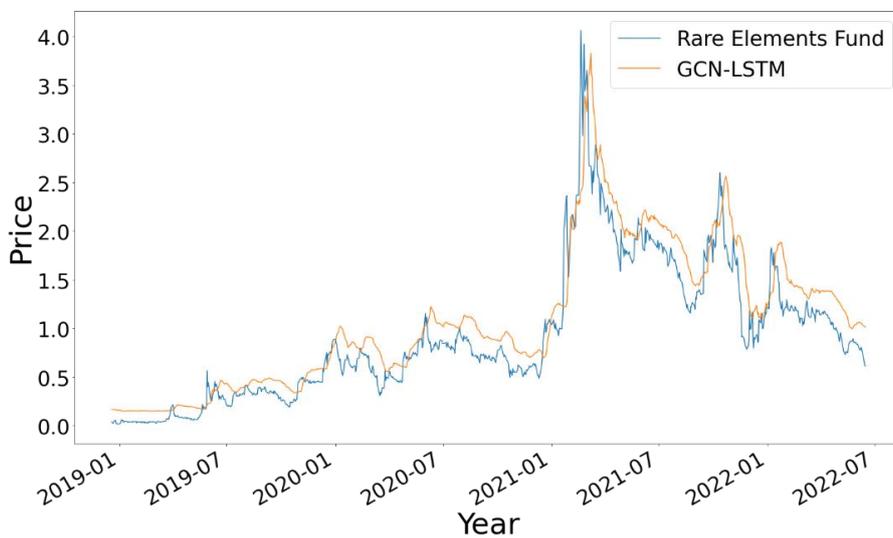
Para una mejor visualización de los resultados obtenidos por los modelos objeto de estudio y remarcar la diferencia de las predicciones, en la Ilustración 34 se pone de manifiesto el ajuste de cada uno de los modelos respecto a los datos pertenecientes al conjunto de prueba, correspondientes al 20% del total de la serie temporal del índice *Rare Earth Elements Fund* de REE.

Ilustración 34. Ajuste de los resultados obtenidos frente al valor real del índice de REE por: (a) ARIMA; (b) PROPHET; (c) BiLSTM; (d) GCN-LSTM; y (e) BiLSTM-GCN.

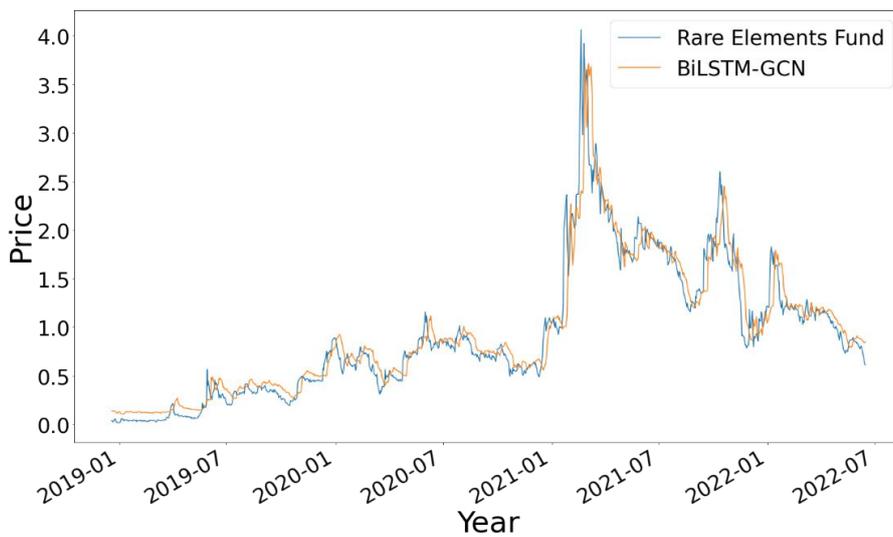




(c)



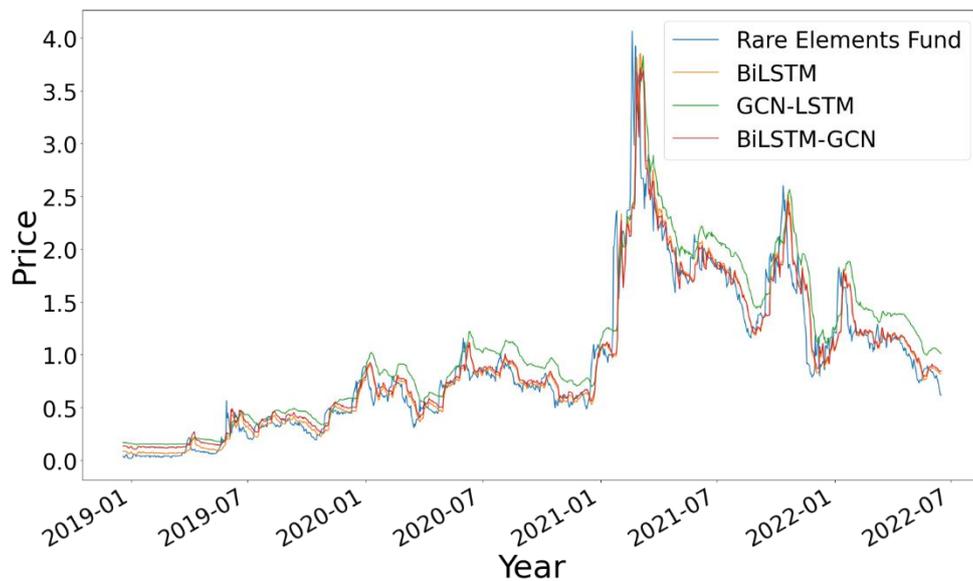
(d)



(e)

El modelo BiLSTM-GCN es el que refleja una mayor precisión en las predicciones con una gráfica más ajustada a los datos reales. En la Ilustración 35 se realiza la comparativa del ajuste de los tres modelos basados en DNN en la que se pone de manifiesto la superioridad del modelo propuesto.

Ilustración 35. Ajuste de los resultados obtenidos por BiLSTM, GCN-LSTM y BiLSTM-GCN frente al valor real del índice REE.



Con el objetivo de verificar la existencia de una diferencia estadísticamente significativa, y con ello corroborar la eficiencia del modelo híbrido, se realiza el test de Friedman (1940), mediante el cual es posible confirmar si se rechaza la hipótesis nula, que establece que la media de cada una de las predicciones realizadas con los diferentes métodos es igual a las demás (ver Tabla 11), y que se realiza comparando las medianas de dos grupos de datos. Se realiza el test de Friedman a los resultados arrojados por los modelos de redes neuronales, ya que estos son los resultados más precisos que los obtenidos por las técnicas tradicionales.

Tabla 11. Test Friedman serie REE

Valor estadístico	1378,244
<i>p</i>-valor	$1,548 \times 10^{-298}$

Los resultados arrojados por la prueba incluyendo todos los modelos utilizados en el experimento muestran un *p*-valor inferior a 0,05, por lo que es posible rechazar la hipótesis nula, siendo suficiente para concluir que las medias de las predicciones realizadas tienen diferencias significativas.

Dado que el resultado de la prueba de Friedman es significativo, es posible concluir que al menos dos de los grupos comparados son significativamente diferentes, por lo que es oportuno realizar más pruebas para identificar cuáles son los grupos. Se realizará la prueba de Wilcoxon (Rosner et al., 2006) entre cada par de grupos con los resultados que se muestran en la Tabla 12. Los resultados muestran que el par de datos que contiene el nuevo modelo con la serie original REE es el único que permite rechazar la hipótesis nula, corroborando una mayor precisión en los resultados de este modelo.

Tabla 12. Test Wilcoxon serie REE

Datos comparados	Valor estadístico	p-valor
REE y BiLSMT	164284,0	$4,597 \times 10^{-07}$
REE y GCN-LSTM	39715,0	$2,946 \times 10^{-99}$
REE y BiLSTM-GCN	128732,0	0,132

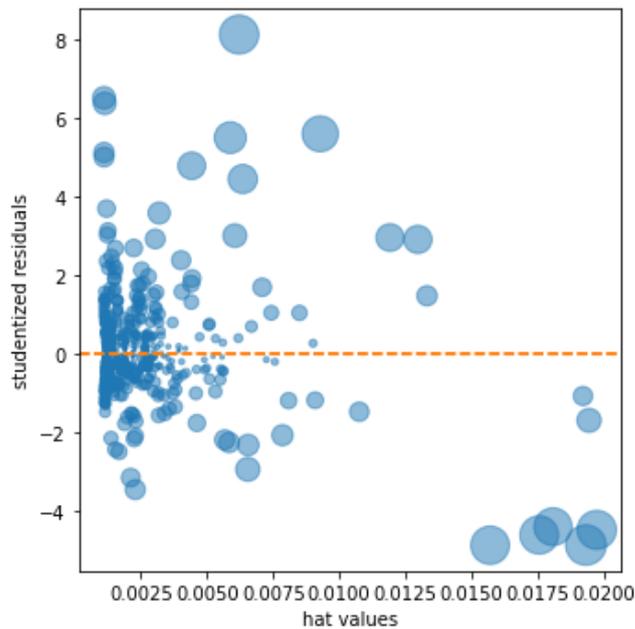
Para proporcionar un análisis detallado del error del modelo, se llevan a cabo una serie de pruebas estadísticas y representaciones visuales para confirmar la eficiencia del modelo. Se lleva a cabo el cálculo del valor de F-Estadístico y su Prob (F-estadístico), cuya hipótesis nula establecerá que todos los parámetros utilizados en la regresión son 0 y que no explican la variable dependiente, no encontrándose relación entre el valor real de la serie temporal y la predicción. El valor resultante es muy pequeño, por lo que se rechaza la hipótesis nula, como se muestra en la Tabla 13.

Tabla 13. Valores estadísticos resultados REE

Valores	Valor estadístico
Curtosis	17,805
F – Estadístico	95,78
Prob (F-Estadístico)	$1,56 \times 10^{-06}$

La medida de Curtosis es de utilidad para medir la concentración de los datos alrededor de la media; un coeficiente positivo significará una concentración más alta de los datos alrededor de la media, mientras que un coeficiente negativo reflejará una concentración más baja. El resultado del modelo propuesto muestra una alta concentración alrededor de la media; este resultado se confirma gráficamente con el resultado de reflejar la gráfica de influencia de la Ilustración 36 en la que se muestran los residuos estudentizados junto con los valores *hat* de especial interés para detectar valores atípicos en las predicciones. El gráfico muestra una pequeña cantidad de valores atípicos en relación con la cantidad de observaciones estudiadas.

Ilustración 36. Valores atípicos serie REE



El error acumulado de las 4.699 observaciones, que consiste en la suma de los errores de todas las predicciones, tiene un resultado de 25,197, lo que lo sitúa en un error medio de 0,005 para cada observación.

6.6 PRECIOS DE LAS MATERIAS PRIMAS

Tras la inestabilidad generada por la pandemia de COVID-19, la demanda de materias primas se ha recuperado rápidamente. Algunos analistas económicos (Poza y Monge, 2021; Monge y Gil-Alana, 2021) pronostican un ciclo de crecimiento de estos materiales debido al crecimiento económico posterior a la crisis. Se prevé un aumento en los precios de las materias primas derivado de las nuevas políticas llevadas a cabo por algunos gobiernos mundiales, con la intención de reformar infraestructuras, como la Administración de Estados Unidos o los fondos de recuperación europeos. Este aumento de precios ya puede observarse en distintas materias primas, como la madera. De hecho, en Estados Unidos ya se aprecia un fuerte crecimiento debido a la alta demandada de viviendas unida a la escasez de oferta. La subida de precios de la madera fue del 63,5% en 2022 y se sitúa en un 450% aproximadamente de los mínimos de 2020.

Dado este incremento en los precios de las materias primas, es necesario conocer la evolución de sus precios para poder analizar el impacto que esta tendencia puede generar

en las economías y los efectos derivados, como el aumento de los precios de los productos de primera necesidad repercutidos en el consumidor final, derivando en un aumento de la inflación.

Los precios de las materias primas es un tema ampliamente discutido en la literatura desde mediados del siglo XX. Stein (1961) afirmaba que los precios de las materias primas y los precios al contado están altamente relacionados. Garbade y Silber (1983) desarrollaron un modelo de dinámica de precios simultánea, mientras que Cox et al. (1981), Jarrow y Oldfield (1981) y Richard y Sundaresan (1981) argumentaron que un contrato de futuros puede considerarse como una serie de contratos al contado de un día, en los que se obtiene una ganancia o una pérdida cada día en una nueva fecha.

Desde finales del siglo XX los precios de las materias primas sufrieron un aumento en su valor hasta el año 2008, en el que los precios se triplicaron (Jacks, 2013), y donde se produjo un debate acerca de la especulación que provocó este aumento. Erdem y Unalmis (2016) analizaron cómo fue revertida la tendencia a la baja de los precios de las materias primas que se experimentó hasta el año 2000, triplicándose los precios antes de la crisis de 2008. En los años siguientes, propiciado por distintos *shocks* económicos, los precios de las materias primas experimentaron fluctuaciones, sin lograr recuperar los valores más altos del año 2012. Más recientemente, Erten y Ocampo (2021) estudiaron cómo, en 2020 tras el impacto inicial de la pandemia de COVID-19, el levantamiento paulatino de las restricciones a la movilidad a partir del segundo trimestre y los paquetes de estímulo económico impulsaron la recuperación de la actividad.

6.6.1 TÉCNICAS PREDICCIÓN DE PRECIOS MATERIAS PRIMAS

En la economía aplicada un tema ampliamente discutido ha sido el comportamiento a largo plazo de los precios de las materias primas, en concreto si se trata de series con comportamiento estacionario o por el contrario contienen una raíz unitaria. Algunos investigadores como Granger y Joyeux (1980) y Hosking (1981) emplearon métodos fraccionarios para el análisis de los precios de materias primas, capturando la dependencia a largo plazo de estos modelos y empleando métodos de regresión sugeridos por Geweke y Porter-Hudak (1983). Lien y Tse (1999) analizaron varios índices de materias primas a través de metodología ARFIMA-GARCH.

Haidar et al. (2008) basaron su investigación de predicción de distintos materiales, entre ellos materias primas, en un modelo de red neuronal de tres capas, donde las redes neuronales mostraron un gran poder de predicción con alta precisión. La superioridad de las redes neuronales en la predicción de diferentes valores de mercado también fue puesta de manifiesto con autores como Kulkarni y Haidar (2009) y Atsalakis y Valavanis (2009) entre otros, donde mostraban una clara superioridad en la predicción de precios de mercado usando redes neuronales, alcanzando precisiones muy superiores a las obtenidas por técnicas estadísticas clásicas. Entre estos estudios destaca el llevado a cabo por Lasheras et al. (2015), donde comparaba el modelo ARIMA frente a un modelo de redes neuronales para la predicción de precios al contado de cobre, con un rendimiento más eficiente y preciso de la red neuronal empleada.

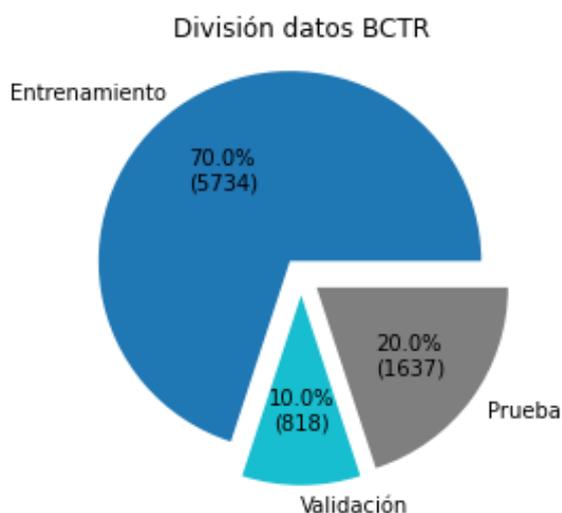
6.6.2 DATOS UTILIZADOS

Para este estudio se ha utilizado el índice de materias primas de *Bloomberg Commodities Total Return* (BCTR) en frecuencia diaria, desde el 2 de enero de 1991 hasta el 25 de mayo de 2022. Los datos se obtuvieron de la base de datos Eikon de Thomson Reuters. El índice representa ciertas materias primas relacionadas con la energía, la ganadería, los productos blandos, los metales industriales, los metales preciosos y los granos.

El pronóstico se realiza sobre las 8.192 observaciones recogidas en el intervalo señalado y son importadas a un *Dataframe* en Python generado con la librería Pandas, que permitirá la manipulación y transformación de los datos. En la Ilustración 37 se representa la división de los datos en los conjuntos de entrenamiento (70%), validación (10%) y prueba (20%).

Una vez estructurados se ha procedido a la generación del grafo asociado utilizando el método del grafo de visibilidad desarrollado por Lacasa et al. (2008), y descrito en el capítulo 5 (apartado 5.2.3). Obtenido el grafo correspondiente a los datos, se lleva a cabo el cálculo de la matriz de correlación que será utilizada como entrada en el modelo *StellarGraph*. Al tratarse de una serie univariada la matriz resultante será igual a 1.

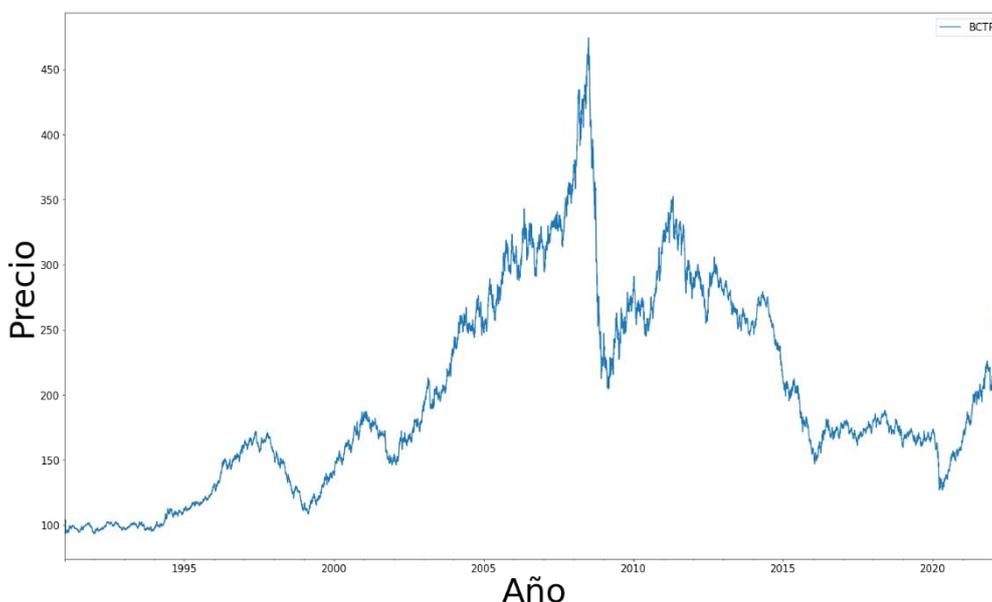
Ilustración 37. Serie temporal BCTR en los conjuntos de entrenamiento, validación y prueba.



6.6.2.1 ESTUDIO DE LA SERIE TEMPORAL BCTR

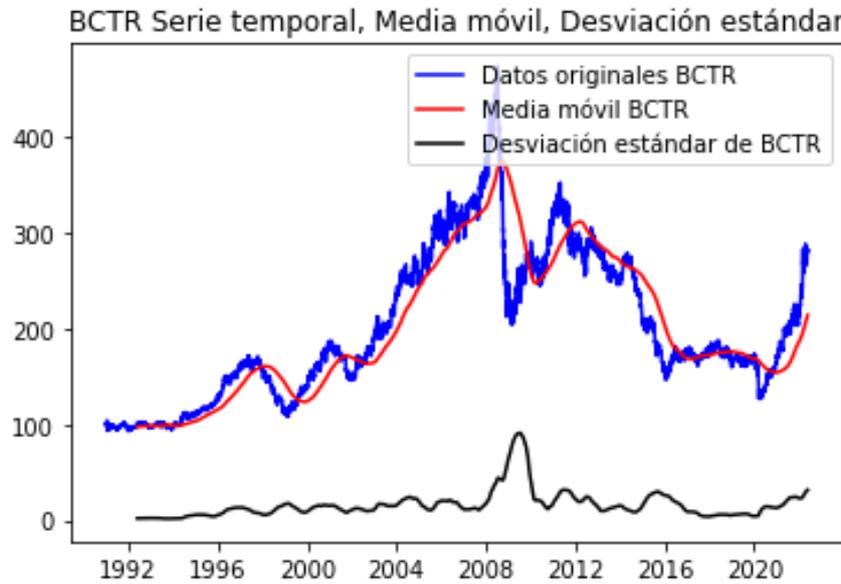
La serie temporal correspondiente a los datos BCTR es analizada con el objetivo de estudiar la estacionariedad y tendencia de los datos para realizar las posteriores predicciones en base a estos datos. La Ilustración 38 muestra los datos originales de la serie temporal.

Ilustración 38. Representación de la serie temporal BCTR



El cálculo de la desviación estándar y la media móvil a 30 días se muestra en la Ilustración 39.

Ilustración 39. Representación de la media móvil y la desviación estándar BCTR



Se realiza el test aumentado de Dickey Fuller y Phillips Perrón (descritos en el capítulo 3, sección 3.2.2) sobre los datos para la comprobación de la estacionariedad de la serie, los valores críticos y el p -valor permitirá aceptar o rechazar la hipótesis nula. La tabla 14 refleja los resultados obtenidos en el test.

Tabla 14. Resultados Serie Temporal BCTR

Valores ADF	Valores PP	Métrica
-1,593	-0,1519	Test Estadísticos
0,487	0,524	p -valor
8160,000	8160.000	Número de observaciones
-2,862	-2,860	Valor crítico (5%)

El criterio de información de Akaike, también descrito en el capítulo 3 (sección 3.2.3) será el empleado para la elección del mejor modelo ARIMA a emplear.

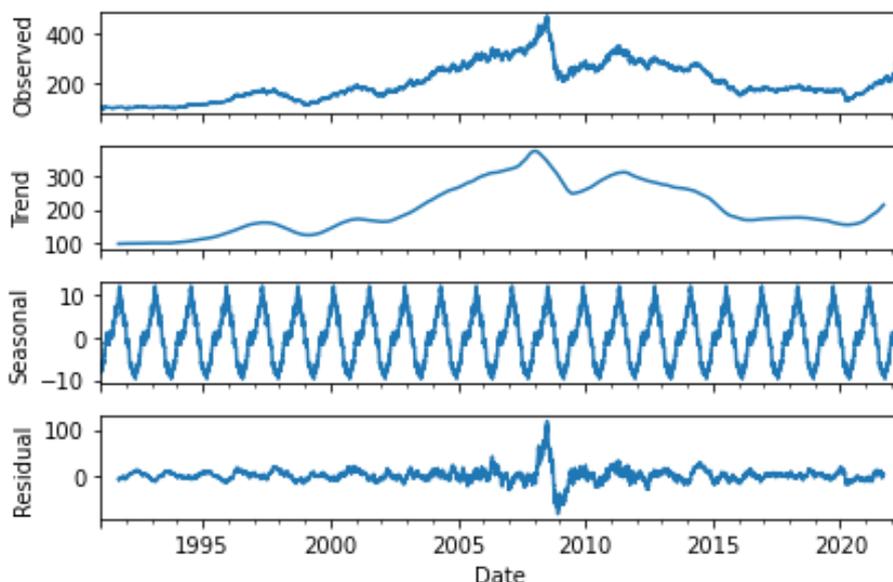
El p -valor inferior al 5% y el test estadístico mayor que el valor crítico refleja la no estacionariedad de la serie, lo que confirma la hipótesis nula. Estos resultados son mostrados gráficamente en la descomposición de la serie de la Ilustración 40.

6.6.3 RESULTADOS

Los resultados obtenidos tras realizar la predicción de precios del índice de materias primas de BCTR con los modelos ARIMA, Prophet, BiLSTM y GCN-LSTM (todos ellos

descritos en los capítulos 2 y 3) frente al modelo BiLSTM-GCN propuesto en este trabajo se muestran en la Tabla 15. Los resultados obtenidos por el modelo BiLSTM-GCN demuestran una mayor precisión en las predicciones y menor error en los pronósticos. En este estudio, los parámetros ARIMA (p,d,q) utilizados han sido (0,1,0).

Ilustración 40. Descomposición de la serie BCTR



Los resultados recogidos muestran la superioridad del modelo combinado propuesto con unas métricas notablemente mejores que las obtenidas por el resto de los modelos sometidos a estudio. La tasa de RMSE del modelo BiLSTM-GCN es un 94,6% inferior que el modelo ARIMA, un 73,7% mejor que el modelo PROPHET y muestra una reducción del 20,8% y del 23,5% con respecto a los modelos BiLSTM y GCN-LSTM, respectivamente.

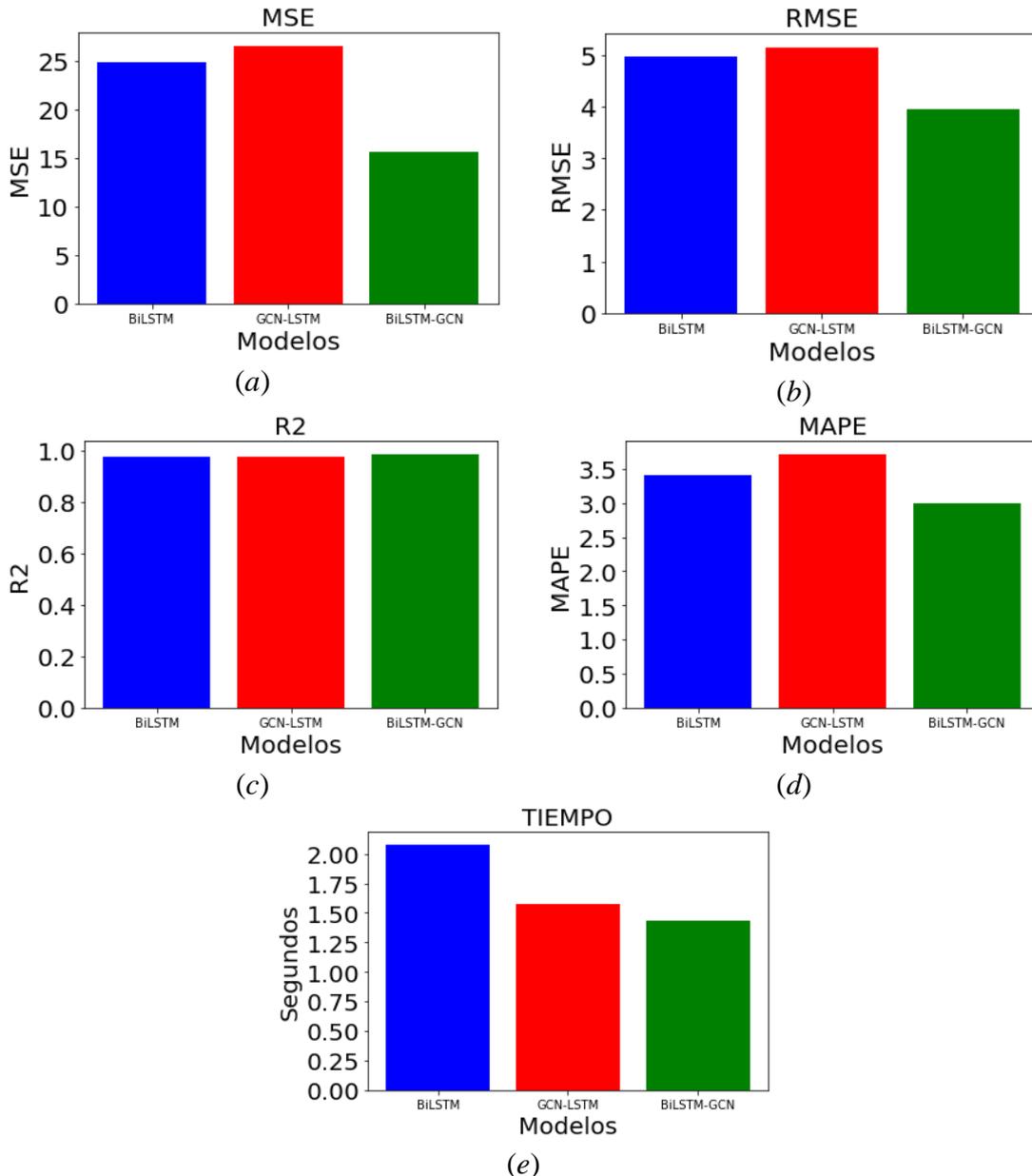
Tabla 15. Resultados obtenidos por el modelo BiLSTM-GCN en la predicción de precios de materias BCTR, frente a ARIMA, PROPHET, BiLSTM y GCN-LSTM.

	MSE	RMSE	R²	MAPE	Tiempo
ARIMA	5372,381	73,291	-3,974	76,662	2,50 s
PROPHET	225,133	15,004	0,674	7,980	2,25 s
BiLSTM	24,802	4,980	0,9772	3,412	2,08 s
GCN-LSTM	26,533	5,152	0,976	3,723	1,57 s
BiLSTM-GCN	15,557	3,941	0,986	3,001	1,43 s

La gran cantidad de observaciones empleadas en este estudio afectan negativamente a los modelos ARIMA y PROPHET en términos de un peor rendimiento, mejorando

significativamente los resultados en los modelos basados en redes neuronales. Debido a los resultados obtenidos por los dos modelos anteriores, no han sido incluidos en la evaluación de los rendimientos mostrada en la Ilustración 41, favoreciendo así una mejor representación del error del modelo combinado propuesto.

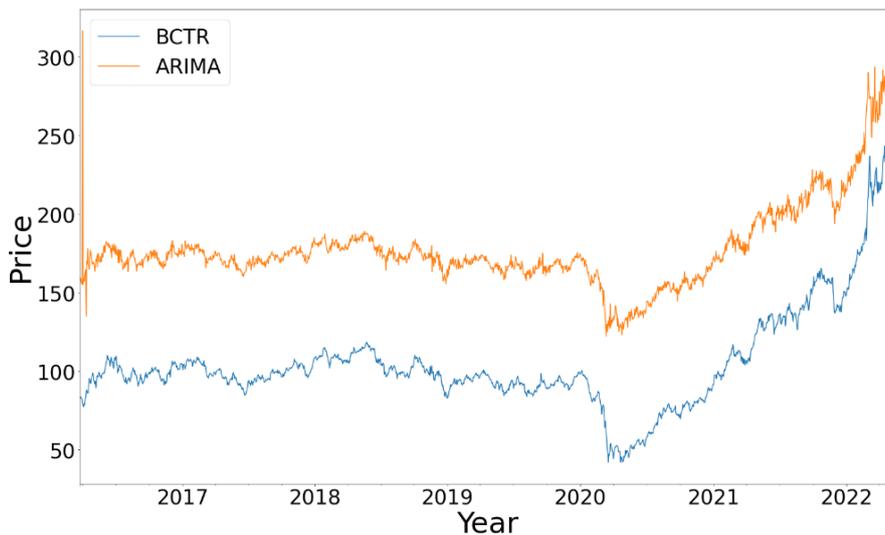
Ilustración 41. Métrica de resultados BCTR: (a) MSE; (b) RMSE; (c) R2; (d) MAPE; y (e) Tiempo.



Con el objetivo de una mejor visualización de los resultados obtenidos por los distintos modelos empleados en este análisis, la Ilustración 42 representa los resultados obtenidos por cada uno de los modelos frente a los datos reales del conjunto de prueba, con un 20% del total de observaciones de la serie temporal perteneciente al índice de materias primas BCTR.

La Ilustración 43 compara los resultados obtenidos por los tres modelos basados en DNN, en el periodo comprendido entre los años 2021 y 2022, y en donde el modelo BiLSTM-GCN muestra un mayor ajuste con los datos reales.

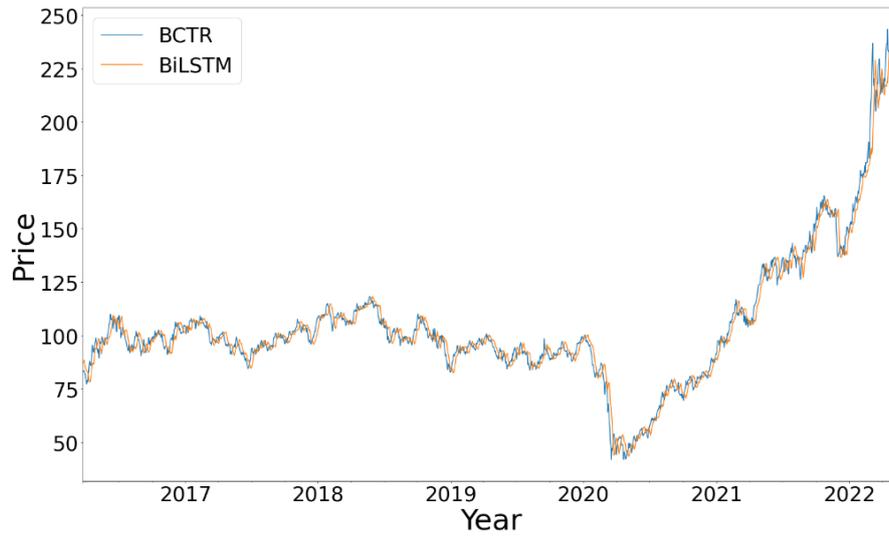
Ilustración 42. Ajuste de los resultados obtenidos frente al valor real del índice BCTR por: (a) ARIMA; (b) PROPHET; (c) BiLSTM; (d) GCN-LSTM; y (e) BiLSTM-GCN.



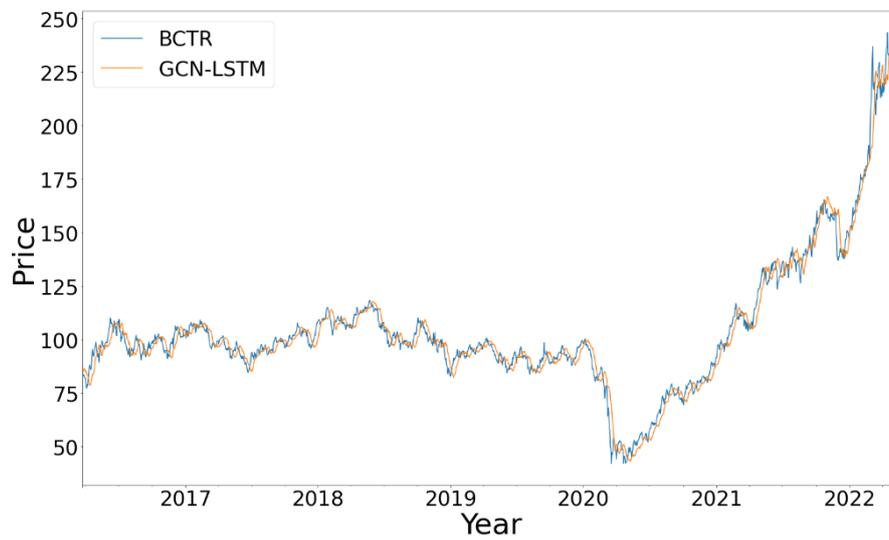
(a)



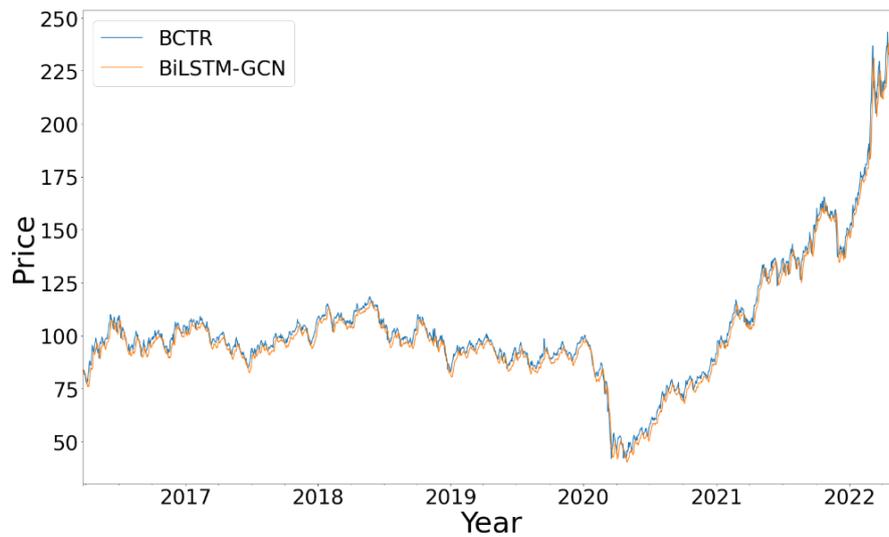
(b)



(c)

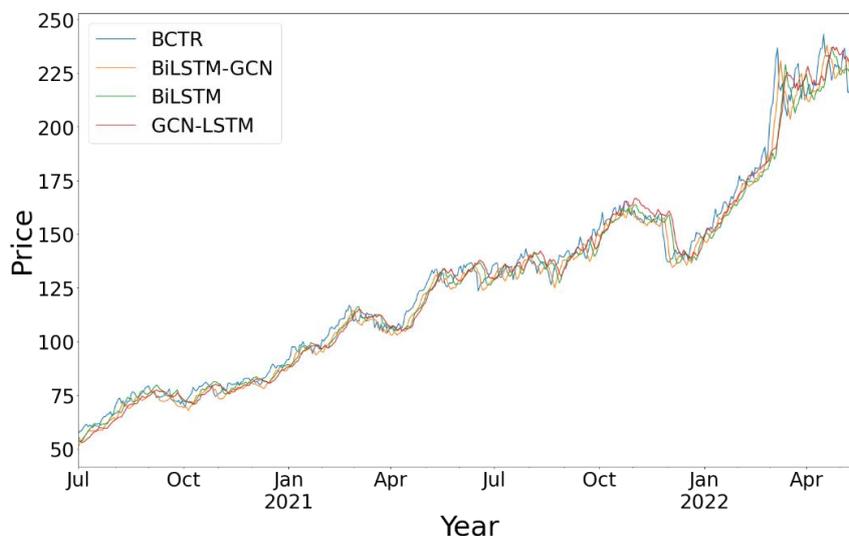


(d)



(e)

Ilustración 43. Ajuste de los resultados obtenidos por BiLSTM, GCN-LSTM y BiLSTM-GCN frente al valor real del índice BCTR.



Para poder llevar a cabo la verificación de la existencia de una diferencia estadísticamente significativa entre los resultados, y con ello corroborar la eficiencia del nuevo modelo, se realiza el test de Friedman (1940), que permitirá determinar si se rechaza la hipótesis nula, que establece que la promedio de cada una de las predicciones realizadas con los diferentes métodos es igual a las demás como se refleja en la tabla 16, se realiza mediante la comparación de las medianas de dos grupos de datos. Se aplica la prueba de Friedman a los resultados obtenidos por los modelos de redes neuronales, al tratarse de unos resultados más precisos que los obtenidos por los otros modelos.

Tabla 16. Test Friedman serie BCTR

Valor estadístico	203,085
p-valor	$9,088 \times 10^{-44}$

A la luz de los resultados obtenidos en la prueba incluyendo todos los modelos utilizados en el experimento reflejan un p -valor menor a 0,05, por lo que podemos rechazar la hipótesis nula, teniendo evidencia suficiente para concluir que las medias de las predicciones realizadas tienen diferencias significativas.

Dado que el resultado de la prueba de Friedman es significativo, es posible afirmar que al menos dos de los grupos comparados son significativamente diferentes, sin una evidencia de cuáles son esos grupos significativamente diferentes; para lo cual se lleva a

cabo la prueba de Wilcoxon (Rosner et al., 2006) entre cada par de grupos con los resultados mostrados en la Tabla 17. Los resultados muestran cómo, aunque todos los pares de datos nos permiten rechazar la hipótesis nula, es el correspondiente a los datos originales de la Serie BCTR con las predicciones obtenidas por el modelo BiLSTM-GCN el que obtiene un p -valor menor, corroborando una mayor precisión en los resultados de este modelo.

Tabla 17. Test Wilcoxon serie BCTR

Datos comparados	Valor estadístico	p -valor
BCTR y BiLSMT	35408,0	$1,276 \times 10^{-15}$
BCTR y GCN-LSTM	43283,0	$3,779 \times 10^{-08}$
BCTR y BiLSTM-GCN	15895,0	$1,195 \times 10^{-45}$

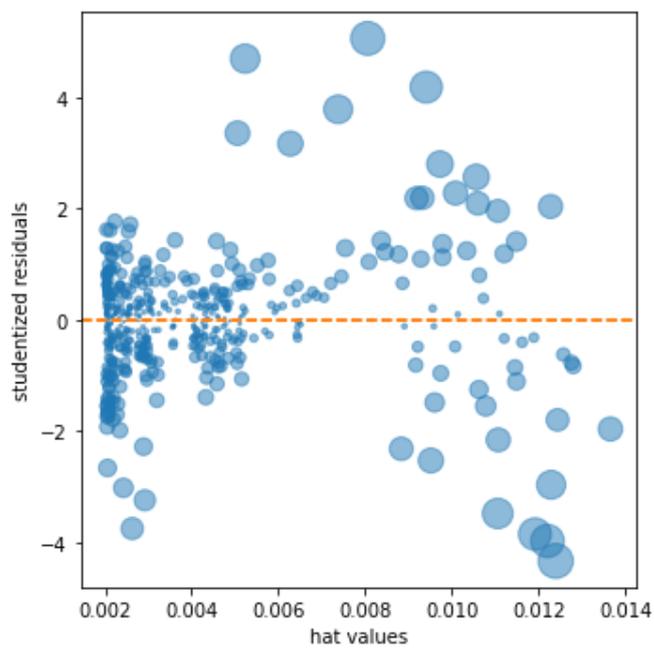
Con el objetivo de proporcionar un análisis detallado del error del modelo, se llevan a cabo una serie de pruebas estadísticas y representaciones visuales para confirmar la eficiencia del modelo. Se realiza el cálculo del valor de F-Estadístico y su Prob (F-estadístico), cuya hipótesis nula establece que todos los parámetros utilizados en la regresión son 0 y que no ayuda a explicar la variable dependiente, lo que indica una falta de relación entre el valor real de la serie temporal y la predicción. El valor resultante es muy pequeño, por lo que se rechaza la hipótesis nula, como se muestra en la Tabla 18.

Tabla 18. Valores estadísticos resultados serie BCTR

Valores	Valor estadístico
Curtosis	8,173
F – Estadístico	55,99
Prob (F-Estadístico)	$0,23 \times 10^{-06}$

La medida de Curtosis permite medir la concentración de los datos alrededor de la media; un coeficiente positivo indicará una concentración más alta de los datos alrededor de la media, mientras que un coeficiente negativo reflejará una concentración más baja. El resultado del modelo propuesto refleja una concentración elevada alrededor de la media; siendo un resultado coherente con el que refleja la gráfica de influencia de la Ilustración 44, en la que se muestran los residuos estudentizados junto con los valores *hat* de especial interés para detectar valores atípicos en las predicciones. El gráfico muestra una pequeña cantidad de valores atípicos en relación con la cantidad de observaciones estudiadas.

Ilustración 44. Valores atípicos serie BCTR



El error acumulado de las 8.192 observaciones, que consiste en la suma de los errores de todas las predicciones, tiene un resultado de 1427,16, lo que lo sitúa en un error medio de 0,174 para cada observación.

7 CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS

El modelo combinado propuesto en este trabajo de tesis consiste en una técnica novedosa para el pronóstico de series temporales, que uniendo las características de las redes de tipo BiLSTM y GCN, es capaz de ajustar las predicciones mejorando los resultados de ambos modelos por separado. La red de tipo GCN aplicada a la predicción de series temporales consigue capturar las características de los datos mediante la representación gráfica de estos, que combinado con el potencial de las redes recurrentes de tipo BiLSTM, permite la creación del modelo propuesto.

La combinación de las salidas de los dos tipos de DNN en el modelo híbrido propuesto genera una red sólida que mejora los resultados logrados por otros modelos, tanto clásicos (ARIMA, PROPHET) como basados en ANN (BiLSTM, GCN-LSTM), y en donde las métricas seleccionadas para la evaluación del rendimiento (RMSE, MSE, MAPE y R^2), ponen de manifiesto un menor error en las predicciones y un menor tiempo de ejecución en la mayoría de los casos, demostrando cómo los problemas presentados por modelos clásicos como ARIMA y PROPHET ante una gran cantidad de observaciones, no suponen un factor determinante en el caso del nuevo modelo estudiado.

Gracias a los resultados de esta investigación, se confirma la capacidad de combinar diferentes tipos de redes neuronales concebidas para diferentes propósitos (Abbasimehr et al., 2022), para la predicción de series temporales económicas con resultados que mejoran la literatura existente (Ensafi et al., 2022) y los obtenidos por diferentes pronósticos tradicionales técnicas.

Entre los distintos objetivos marcados al inicio y a lo largo del desarrollo de esta investigación, se logró realizar una revisión de la literatura en lo que a series temporales se refiere, permitiendo modelizar aquellas en las que se centraban los experimentos. El análisis del estado del arte sobre la predicción de series temporales con redes neuronales permitió establecer la necesidad existente de un modelo que lograra capturar las características de la serie temporal desde varias perspectivas, minimizando los errores en las predicciones.

El modelo BiLSTM-GCN logra capturar con éxito las características espaciales y temporales de los datos relacionados con series temporales del ámbito económico, principal objetivo de esta tesis doctoral. Concretamente, procedentes de precios del petróleo en base al índice WTI, de materias primas en base al índice BCTR y de materias primas raras sobre el índice *Rare Earth Elements Fund*. Las tres series temporales estudiadas en esta tesis muestran notables diferencias en sus características de estacionalidad y tendencia, así como en el número de observaciones, lo que confirma la robustez de la red BiLSTM-GCN para modelar las relaciones entre los datos.

Entre las futuras líneas de trabajo derivadas de esta tesis doctoral, se propone analizar el comportamiento del modelo mediante el uso de distintas series temporales, incluyendo la observación de series de distintos ámbitos. También se profundizará en las características especiales del modelo, como la variación de ciertos hiperparámetros y el comportamiento de la red en función de las modificaciones.

BIBLIOGRAFÍA

- Abbasimehr, H.; Paki, R.; Bahrini, A. (2022). A novel approach based on combining deep learning models with statistical methods for COVID-19 time series forecasting. *Neural Computer Applications*, 34, 3135–3149.
- Akaike, H. (1970). Statistical predictor identification. *Annals of the institute of Statistical Mathematics*, 22(1), 203-217.
- Almalaq, A., & Edwards, G. (2017). A review of deep learning methods applied on load forecasting. In 2017 16th IEEE international conference on machine learning and applications (ICMLA) (pp. 511-516). IEEE.
- Amano, A. (1987). A small forecasting model of the world oil market. *Journal of Policy Modeling*, 9(4), 615-635.
- Amano, R. A., & Van Norden, S. (1998). Oil prices and the rise and fall of the US real exchange rate. *Journal of international Money and finance*, 17(2), 299-316.
- Atsalakis, G. S. & Valavanis, K. P. (2009) Surveying stock market forecasting techniques– part II: soft computing methods, *Expert Systems with Applications*, 36, 5932-5941.
- Baffes, J., Kose, M. A., Ohnsorge, F. & Stocker, M. (2015). The great plunge in oil prices: Causes, consequences, and policy responses. Policy Research Note No.1, World Bank.
- Balcilar, M., Renton, G., Héroux, P., Gauzere, B., Adam, S., & Honeine, P. (2020). Bridging the gap between spectral and spatial domains in graph neural networks. arXiv preprint arXiv:2003.11702.
- Baruník, J., & Malinska, B. (2016). Forecasting the term structure of crude oil futures prices with neural networks. *Applied energy*, 164, 366-379.
- Baumeister, C., & Peersman, G. (2013). The role of time-varying price elasticities in accounting for volatility changes in the crude oil market. *Journal of Applied Econometrics*, 28(7), 1087-1109.
- Baumeister, C. & Kilian, L. (2016). Understanding the decline in the price of oil since June 2014. *Journal of the Association of Environmental and Resource Economists*, 3, 131- 158.

- Beck, J. V., & Arnold, K. J. (1977). *Parameter estimation in engineering and science*. James Beck.
- Bekiros, S. D., & Diks, C. G. (2008). The relationship between crude oil spot and futures prices: Cointegration, linear and nonlinear causality. *Energy Economics*, 30(5), 2673-2685.
- Bengio, S., Fessant, F., & Collobert, D. (1995). A connectionist system for medium-term horizon time series prediction. In *Proc. Intl. Workshop Application Neural Networks to Telecoms* (pp. 308-315).
- Bentzen, J. (2007). Does OPEC influence crude oil prices? Testing for co-movements and causality between regional crude oil prices. *Applied Economics*, 39(11), 1375-1385.
- Bhanja, S., & Das, A. (2018). Impact of data normalization on deep neural network for time series forecasting. *arXiv preprint arXiv:1812.05519*.
- Bhardwaj, G., & Swanson, N. R. (2006). An empirical investigation of the usefulness of ARFIMA models for predicting macroeconomic and financial time series. *Journal of econometrics*, 131(1-2), 539-578.
- Bian, L., Liu, Z., Zhang, P., & Dong, J. J. (2021). Rare Earth Price Fluctuation and Forecasting Methods under the COVID-19 Pandemic. *CONVERTER*, 2021(7), 276-287.
- Bosan, S., & Harris, T. R. (1996). Graphical lung analysis and simulation environment. *Computer methods and programs in biomedicine*, 49(3), 211-228.
- Borovykh, A., Oosterlee, C. W., & Bohté, S. M. (2019). Generalization in fully-connected neural networks for time series forecasting. *Journal of Computational Science*, 36, 101020.
- Bowden, G. J., Maier, H. R., & Dandy, G. C. (2002). Optimal division of data for neural network models in water resources applications. *Water resources research*, 38(2), 2-1.
- Box, G. E. P. & Jenkins, G. M. (1970) *Time-series Analysis, Forecasting and Control*. San Francisco: Holden -Day.
- Braddock, R. D., Kremmer, M. L., & Sanzogni, L. (1998). Feed-forward artificial neural network model for forecasting rainfall run-off. *Environmetrics: The official journal of the International Environmetrics Society*, 9(4), 419-432.
- Bruna, J., Zaremba, W., Szlam, A., & LeCun, Y. (2013). Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*.
- Cao, P., Zhu, Z., Wang, Z., Zhu, Y & Niu, Q. (2022). Applications of graph convolutional networks in computer vision. *Neural Comput & Applic* 34, 13387–13405.
- Chen, J., Ma, T., & Xiao, C. (2018). Fastgc: fast learning with graph convolutional networks via importance sampling. *arXiv preprint arXiv:1801.10247*.

- Chung, R. C., Ip, W. H., & Chan, S. L. (2009). An ARIMA-intervention analysis model for the financial crisis in China's manufacturing industry. *International Journal of Engineering Business Management*, 1, 5.
- Cox, J., Ingersoll, J. & Ross, S. (1981). The relation between forward prices and futures prices, *Journal of Financial Economics*, 9 (4), 321-346.
- Danel, T., Spurek, P., Tabor, J., Śmieja, M., Struski, Ł., Słowik, A., & Maziarka, Ł. (2020). Spatial graph convolutional networks. In *International Conference on Neural Information Processing* (pp. 668-675).
- Das, D., Kumar, S. B., Tiwari, A. K., Shahbaz, M., & Hasim, H. M. (2018). On the relationship of gold, crude oil, stocks with financial stress: A causality-in-quantiles approach. *Finance Research Letters*, 27, 169-174.
- De Lillo, A., & Meraviglia, C. (1998). The role of social determinants on men's and women's mobility in Italy. A comparison of discriminant analysis and artificial neural networks. *Substance use & misuse*, 33(3), 751-764.
- Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* (pp. 3844–3852).
- Deng, A., & Hooi, B. (2021). Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence* 5 (35), 4027-4035.
- Devarakonda, A., Naumov, M., & Garland, M. (2017). Adabatch: Adaptive batch sizes for training deep neural networks. arXiv preprint arXiv:1712.02029.
- Dickey, D. A. & Fuller, W. A. (1979). Distributions of the estimators for autoregressive time series with a unit root, *Journal of American Statistical Association*, 74 (366), 427-481.
- Diebold, F., & Mariano, R. (1995). Comparing predictive accuracy. *Journal of Business and Economic Statistics*, v. 13.
- Ding, Z., Granger, C., & Engle, R. (1993). A long memory property of stock returns and a new model. *Journal of Empirical Finance*, 1, 83106.
- Eğrioğlu, E., Aladağ, Ç. H., & Günay, S. (2008). A new model selection strategy in artificial neural networks. *Applied Mathematics and Computation*, 195(2), 591-597.
- Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2), 179-211.
- Ensafi, Y.; Amin, S.H.; Zhang, G.; Shah, B. (2022). Time-series forecasting of seasonal items sales using machine learning—A comparative analysis. *Int. J. Inf. Manag. Data Insights*, 2, 100058.
- Erdem, F., & Unalmis, I. (2016). Revisiting super-cycles in commodity prices, *Central Bank Review*, 16(4), 137-142.

- Erten, B., & Ocampo, J.A. (2021). The future of commodity prices and the pandemic-driven global recession: Evidence from 150 years of data. *World development*, 137, 105164.
- Frank, R. J., Davey, N., & Hunt, S. P. (2001). Time series prediction and neural networks. *Journal of intelligent and robotic systems*, 31(1), 91-103.
- Frechtling, D. C. (1996). *Practical Tourism Forecasting*. Oxford, UK: Butterworth-Heinemann.
- Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1), 86-92.
- Fu, L. (1994). Rule generation from neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(8), 1114-1124.
- Fuller, W. A. (1976). *Introduction to Statistical Time Series*, New York: JohnWiley. Fuller Introduction to Statistical Time Series 1976.
- Gallicchio, C., & Micheli, A. (2010). Graph echo state networks. In *The 2010 international joint conference on neural networks (IJCNN)* (pp. 1-8). IEEE.
- Garbade, K. & Silber, W.L. (1983) Price Movement and Price Discovery in Futures and Cash Markets. *The Review of Economics and Statistics*, 65, 289-297.
- García, M. V. R., Krzemień, A., del Campo, M. Á. M., García-Miranda, C. E., & Lasheras, F. S. (2018). Rare earth elements price forecasting by means of transgenic time series developed with ARIMA models. *Resources Policy*, 59, 95-102.
- García Martínez, R., Servente, M., & Pasquín, D. (2003). *Sistemas Inteligentes*, Capítulo 1: "Aprendizaje automático", Capítulo 2: "redes Neuronales Artificiales". Buenos Aires, Argentina: Nueva Librería. ISBN: 987-1104-05-7.
- Gasparin, A., Lukovic, S., & Alippi, C. (2022). Deep learning for time series forecasting: The electric load case. *CAAI Transactions on Intelligence Technology*, 7(1), 1-25.
- Geweke, J. & Porter-Hudak, S. (1983). The estimation and application of long memory time series models. *Journal of Time Series Analysis*, 4 (4), 221-238.
- Gheyas, I. A., & Smith, L. S. (2009). A neural network approach to time series forecasting. In *Proceedings of the World Congress on Engineering* (2), 1-3.
- Gholamy, A., Kreinovich, V., & Kosheleva, O. (2018). Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation. Available online: https://scholarworks.utep.edu/cs_techrep/1209/ (accedido el 15 Julio 2022).
- Goodfellow, I. (2016). Nips 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160.

- Gori, M., Monfardini, G., & Scarselli, F. (2005). A new model for learning in graph domains. In *Proceedings. 2005 IEEE international joint conference on neural networks, 2005* (2), 729-734.
- Granger, C. W., & Morris, M. J. (1976). Time series modelling and interpretation. *Journal of the Royal Statistical Society: Series A (General)*, 139(2), 246-257.
- Granger, C. W. (1980). Long memory relationships and the aggregation of dynamic models. *Journal of econometrics*, 14(2), 227-238.
- Granger, C. W., & Joyeux, R. (1980). An introduction to long-memory time series models and fractional differencing. *Journal of time series analysis*, 1(1), 15-29.
- Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249-256). *JMLR Workshop and Conference Proceedings*.
- Glorot, X., Bordes, A., & Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*.
- Gujarati, D. N., & Porter, D. C. (2011). *Econometría Básica*. McGraw Hill, 5^a ed. ISBN: 9788580550511.
- Gülen, S. G. (1998). Efficiency in the crude oil futures market. *Journal of Energy Finance & Development*, 3(1), 13-21.
- Güler, N. F., Übeyli, E. D., & Güler, I. (2005). Recurrent neural networks employing Lyapunov exponents for EEG signals classification. *Expert systems with applications*, 29(3), 506-514.
- Haidar, S. K. & Pan, H. (2008). Forecasting model for crude oil prices based on artificial neural networks. *International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 103-108.
- Hahnloser, R. H., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J., & Seung, H. S. (2000). Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *nature*, 405(6789), 947-951.
- Hamilton, J. D. (2020). *Time series analysis*. Princeton university press.
- Hamdi, M., & Aloui, C. (2015). Forecasting crude oil price using artificial neural networks: a literature survey. *Econ Bull*, 3(2), 1339-1359.
- Han, Y., Karunasekera, S., & Leckie, C. (2020). Graph neural networks with continual learning for fake news detection from social media. *arXiv preprint arXiv:2007.03316*.
- Haykin, S. (1994). *Neural Network: A comprehensive foundation*. Neural networks, Macmillan, ISBN 0-02-352781-7.

- Hebb, D. O. (1949). The first stage of perception: growth of the assembly. *The Organization of Behavior*, 4, 60-78.
- Heijnen, L., R. Rijkers, & R.G. Ohmann (2015). Management of Geological and Drilling Risks of Geothermal Projects in the Netherlands. In *Proceedings of World Geothermal Congress 2015*. Melbourne, Australia: International Geothermal Association (IGA).
- Herrera, P.J., Montes, F. & Pajares, G. (2016). Stereovision matching based on combining neural networks for outdoor images. *Conference on Future Trends in Robotics (RoboCity16)*, pp. 153-160. ISBN: 978-84-608-8452-1.
- Herrera, P.J., Pajares, G., Guijarro, M., Ruz, J.J., & Cruz, J.M. (2011). A stereovision matching strategy for images captured with fish-eye lenses in forest environments. *Sensors* 11 (2), 1756-1783.
- Hill, T., O'Connor, M., & Remus, W. (1996). Neural network models for time series forecasts. *Management science*, 42(7), 1082-1092.
- Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786), 504-507.
- Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition (Vol. 1, pp. 278-282)*. IEEE.
- Hochreiter, S. (1991) Investigations on dynamic neural networks. PhD thesis, Technische Universität München.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.
- Holt, C.C. (1957). Forecasting trends and seasonals by exponentially weighted averages, *Carnegie Institute of Technology, Pittsburgh* ONR memorandum no. 52.
- Hooker, R.H. (1901). Correlation of the marriage-rate with trade. *Journal of the Royal Statistical Society*, 64(3), 485-492.
- Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79 (8): 2554–2558.
- Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5), 359-366.
- Hosking, J.R.M. (1981). Fractional differencing, *Biometrika* 68, 165-176.
- International Energy Administration, IEA (2020). *Global Energy Review 2020*, IEA, Paris.
- Jacks, D.S. (2013) *From Boom to Bust: A Typology of Real Commodity Prices in the Long Run*.

- Jarrett, K., Kavukcuoglu, K., Ranzato, M. A., & LeCun, Y. (2009). What is the best multi-stage architecture for object recognition?. In 2009 IEEE 12th international conference on computer vision, IEEE, 2146-2153.
- Jarrow, R. & Oldfield, G. (1981), Forward contracts and futures contracts, *Journal of Financial Economics*, 9, (4), 373-382.
- Jiang, W. (2021). Applications of deep learning in stock market prediction: Recent progress. *Expert Systems with Applications*, 184, 115537.
- Jiang, W., & Luo, J. (2022). Graph neural network for traffic forecasting: A survey. *Expert Systems with Applications*, 207, 117921.
- Junior, P. R., Salomon, F. L. R., & de Oliveira Pamplona, E. (2014). ARIMA: An applied time series forecasting model for the Bovespa stock index. *Applied Mathematics*, 5(21), 3383.
- Kaboudan, M. A. (2001). Compumetric forecasting of crude oil prices. In *Proceedings of the 2001 congress on evolutionary computation*. IEEE, 01TH8546, (1), 283-287.
- Karsoliya, S. (2012). Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture. *International Journal of Engineering Trends and Technology*, 3(6), 714-717.
- Keskes, O., & Noumeir, R. (2021). Vision-Based Fall Detection Using ST-GCN. *IEEE Access*, 9, 28224-28236.
- Klinger, J. M. (2015). A historical geography of rare earth elements: From discovery to the atomic age. *The Extractive Industries and Society*, 2(3), 572-580.
- Khashei, M., & Bijari, M. (2010). An artificial neural network (p, d, q) model for time series forecasting. *Expert Systems with applications*, 37(1), 479-489.
- Kim, Y. J., & Chi, M. (2018). Temporal Belief Memory: Imputing Missing Data during RNN Training. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI-2018)*.
- Kim, J., & Moon, N. (2019). BiLSTM model based on multivariate time series data in multiple field for forecasting trading area. *Journal of Ambient Intelligence and Humanized Computing*, 1-10.
- Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR '17)*.
- Kohonen, T. (1982). Self-Organized Formation of Topologically Correct Feature Maps. *Biological Cybernetics*, 43 (1), 59–69
- Kolmogorov, A. (1957). Théorie générale des systèmes dynamiques de la mécanique classique. *Séminaire Janet. Mécanique analytique et mécanique céleste*, 1, 1-20.

- Krane, J. & Agerton, M. (2015). Effects of low oil prices on U.S. shale production: OPEC calls the tune and shale swings. Baker Institute Research Paper.
- Kullback, S., & Leibler, R.A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 22, 79-86.
- Kulkarni, S., & Haidar, I. (2009). Forecasting model for crude oil price using artificial neural networks and commodity futures prices. arXiv preprint arXiv:0906.4838.
- Kwiatkowski, D., Phillips, P. C., Schmidt, P., & Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?. *Journal of econometrics*, 54(1-3), 159-178.
- Lacasa, L., Luque, B., Ballesteros, F., Luque, J., & Nuno, J. C. (2008). From time series to complex networks: The visibility graph. *Proceedings of the National Academy of Sciences*, 105(13), 4972-4975.
- Maçaira, P. M., Thomé, A. M. T., Oliveira, F. L. C., & Ferrer, A. L. C. (2018). Time series analysis with explanatory variables: A systematic literature review. *Environmental Modelling & Software*, 107, 199-209.
- Lansangan, J. R. G., & Barrios, E. B. (2009). Principal components analysis of nonstationary time series data. *Statistics and Computing*, 19(2), 173-187.
- Lanzarini, L. C., & De Giusti, A. E. (2002). Redes neuronales aplicadas al reconocimiento de patrones. In *IV Workshop de Investigadores en Ciencias de la Computación*.
- Lasheras, F. S., de Cos Juez, F.J, Suárez Sánchez, A., Krzemień, A. & Riesgo Fernández, P. (2015) Forecasting the COMEX copper spot price by means of neural networks and ARIMA models, *Resources Policy*, 45, 37-43.
- Lasheras, F. S., Gómez, S. L. S., García, M. V. R., Krzemień, A., & Sánchez, A. S. (2017). Time series and artificial intelligence with a genetic algorithm hybrid approach for rare earth price prediction. *ITISE 2017*.
- Lazcano, A., Herrera, P.J., & Monge, M. (2023). A Combined Model Based on Recurrent Neural Networks and Graph Convolutional Networks for Financial Time Series Forecasting. *Mathematics*, 11, 224.
- Le, X. H., Ho, H. V., Lee, G., & Jung, S. (2019). Application of long short-term memory (LSTM) neural network for flood forecasting. *Water*, 11(7), 1387.
- LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., & Jackel, L. (1989). Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2.
- LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10).

- Leybourne, S. J., & McCabe, B. P. (1994). A consistent test for a unit root. *Journal of Business & Economic Statistics*, 12(2), 157-166.
- Li, Q., Tricaud, C., Sun, R., & Chen, Y. (2007). Great Salt Lake surface level forecasting using FIGARCH model. In *International design engineering technical conferences and computers and information in engineering conference 4806*, 1361-1370.
- Li, Z., Xiong, G., Chen, Y., Lv, Y., Hu, B., Zhu, F., & Wang, F. Y. (2019). A hybrid deep learning approach with GCN and LSTM for traffic flow prediction. In *2019 IEEE intelligent transportation systems conference (ITSC)* 1929-1933.
- Lien, D. D. & Tse, Y. K. (1999) Fractional cointegration and futures hedging, *Journal of Futures Markets*, 19 (4), 457-474.
- Lin, M., Lucas, H. C. & Shmueli, G. (2013). Too big to fail: Large samples and the p-value problem. *Information Systems Research*, 24(4), 906-917
- Lin, H. W. & Tegmark, M. (2016). Criticality in formal languages and statistical physics. arXiv preprint arXiv:1606.06737.
- Lins, A. P. S., & Ludermir, T. B. (2005). Hybrid optimization algorithm for the definition of mlp neural network architectures and weights. In *Fifth International Conference on Hybrid Intelligent Systems (HIS'05)* (6).
- MacKinnon, J. G. (1994). Approximate asymptotic distribution functions for unit-root and cointegration tests. *Journal of Business & Economic Statistics*, 12(2), 167-176.
- Magee, J.F. (1958). *Production Planning and Inventory Control*, McGraw-Hill.
- Maier, H. R., & Dandy, G. C. (2000). Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. *Environmental modelling & software*, 15(1), 101-124.
- Makridakis, S., Anderson, A., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., Newton, J., Parzen, E., & Winkler, R. (1982). La exactitud de los métodos de extrapolación (series de tiempo): resultados de una competencia de pronósticos. *J. Forecasting* 1 111–153.
- Makridakis, S., & Hibon, M. (1997). ARMA models and the Box–Jenkins methodology. *Journal of forecasting*, 16(3), 147-163.
- Makridakis, S., Wheelwright, S., & Hyndman, R. (1998). *Forecasting: Methods and Applications*. John Wiley & Sons, New York, USA, third edition.
- Massari, S., & Ruberti, M. (2013). Rare earth elements as critical raw materials: Focus on international markets and future strategies. *Resources Policy*, 38(1), 36-43.
- Masters, T. (1993). *Practical neural network recipes in C++*. Academic Press Professional, Inc.
- Masters, D., & Luschi, C. (2018). Revisiting small batch training for deep neural networks. arXiv preprint arXiv:1804.07612.

- McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 115-133.
- Minsky, M., & Papert, S. (1969). *An introduction to computational geometry*. Cambridge tiass., HIT, 479, 480.
- Mirmirani, S., & Li, H. C. (2004). A comparison of VAR and neural networks with genetic algorithm in forecasting price of oil. In *Applications of artificial intelligence in finance and economics*. Emerald Group Publishing Limited.
- Monge, M. & Gil-Alana, L. A. (2015). Fractional Integration and Cointegration in Merger and Acquisitions in the US Petroleum Industry. *Applied Economics Letters*, 23-10.
- Monge, M., Gil-Alana, L. A., & de Gracia, F. P. (2017a). Crude oil price behaviour before and after military conflicts and geopolitical events. *Energy*, 120, 79-91.
- Monge, M., Gil-Alana, L. A., & de Gracia, F. P. (2017b). US shale oil production and WTI prices behavior. *Energy*, 141, 12-19.
- Monge, M., & Gil-Alana, L. A. (2021). Lithium industry and the US crude oil prices. A fractional cointegration VAR and a Continuous Wavelet Transform analysis. *Resources Policy*, 72, 102040.
- Monge, M., & Lazcano, A. (2022). Commodity Prices after COVID-19: Persistence and Time Trends. *Risks*, 10(6), 128.
- Mukherjee, S., Osuna, E., & Girosi, F. (1997). Nonlinear prediction of chaotic time series using support vector machines. In *Neural Networks for Signal Processing VII. Proceedings of the 1997 IEEE Signal Processing Society Workshop* (pp. 511-520). IEEE.
- Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Icml*.
- Nakamura, E. (2005). Inflation forecasting using a neural network. *Economics Letters*, 86(3), 373-378.
- Namasudra, S., Dhamodharavadhani, S., & Rathipriya, R. (2021). Nonlinear neural network based forecasting model for predicting COVID-19 cases. *Neural processing letters*, 1-21.
- Navratil, M., & Kolkova, A. (2019). Decomposition and forecasting time series in the business economy using prophet forecasting model. *Central European Business Review*, 8(4), 26.
- Nazir, M. S., Alturise, F., Alshmrany, S., Nazir, H. M. J., Bilal, M., Abdalla, A. N., Sanjeevikumar. P., & M. Ali, Z. (2020). Wind generation forecasting methods and proliferation of artificial neural network: A review of five years research trend. *Sustainability*, 12(9), 3778.

- Pajares, G., Guijarro, M., Herrera, P.J., Ruz, J.J., & de la Cruz, J.M. (2010). Fuzzy Cognitive Maps Applied to Computer Vision Tasks. In: Glykas, M. (eds) Fuzzy Cognitive Maps. Studies in Fuzziness and Soft Computing, vol 247.
- Pajares, G., Herrera, P.J., & Besada, E. (2021). Aprendizaje Profundo. RC Libros, Madrid. ISBN: 9788412106985.
- Paletta, Q., & Lasenby, J. (2020). Convolutional Neural Networks applied to sky images for short-term solar irradiance forecasting. arXiv preprint arXiv:2005.11246.
- Palmer, P. A., & Montaña-Moreno, J. J. (1999). ¿Qué son las redes neuronales artificiales? Aplicaciones realizadas en el ámbito de las adiciones. Adicciones, 11(3), 243-255.
- Phillips, P. C., & Perron, P. (1988). Testing for a unit root in time series regression. Biometrika, 75(2), 335-346.
- Playfair, W. (1786), The Commercial and Political Atlas; Representing, by Means of Stained Copper-Plate Charts, the Exports, Imports, and General Trade of England, at a Single View. To Which are Added, Charts of the Revenue and Debts of Ireland, Done in the Same Manner by James Corry, London: Debrett; Robinson; and Sewell.
- Playfair, W. (2005). Playfair's commercial and political atlas and statistical breviary. Cambridge University Press.
- Poynting, J. H. (1884). A comparison of the fluctuations in the price of wheat and in cotton and silk imports into Great Britain. Journal of the Royal Statistical Society, 47, 34–74.
- Poza, C., & Monge, M. (2021). Forecasting Spanish economic activity in times of COVID-19 by means of the RT-LEI and machine learning techniques. Applied Economics Letters, 1-6.
- Proelss, J., Schweizer, D., & Seiler, V. (2020). The economic importance of rare earth elements volatility forecasts. International Review of Financial Analysis, 71, 101316.
- Rajalakshmi, V., & Ganesh Vaidyanathan, S. (2022). Hybrid CNN-LSTM for Traffic Flow Forecasting. In Proceedings of 2nd International Conference on Artificial Intelligence: Advances and Applications (pp. 407-414).
- Rast, M. (2001). Fuzzy neural networks for modelling commodity markets “ The Proceedings of the Conference of the North American Fuzzy Information Processing Society” (NAFIPS’2001), 952-955.
- Rhif, M., Ben Abbes, A., Farah, I. R., Martínez, B., & Sang, Y. (2019). Wavelet transform application for/in non-stationary time-series analysis: a review. Applied Sciences, 9(7), 1345.
- Richard, S. F. & Sundaresan, M. (1981). A continuous time equilibrium model of forward prices and futures prices in a multigood economy. Journal of Financial Economics, Elsevier, vol. 9(4), 347-371.
- Rojas, R. (1996). Neural Networks: A Systematic Introduction. Springer-Verlag, Berlin.

- Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6), 386.
- Rosner, B., Glynn, R. J., & Lee, M. L. T. (2006). The Wilcoxon signed rank test for paired comparisons of clustered data. *Biometrics*, 62(1), 185-192.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533-536.
- Sánchez, P., & Velásquez, J. (2011). El rol del algoritmo de entrenamiento en la selección de modelos de redes neuronales. *Revista UDCA Actualidad & Divulgación Científica*, 14(1), 149-156.
- Sánchez-González, A., Heess, N., Springenberg, J. T., Merel, J., Riedmiller, M., Hadsell, R., & Battaglia, P. (2018). Graph networks as learnable physics engines for inference and control. In *International Conference on Machine Learning* (pp. 4470-4479). PMLR.
- Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2008). The graph neural network model. *IEEE transactions on neural networks*, 20(1), 61-80.
- Schurr, S. H., Netschert, B. C., Eliasberg, V. F., Lerner, J., & Landsberg, H. H. (1960). Energy in the American economy, 1850-1975.
- Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11), 2673–2681.
- Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, (pp. 461-464).
- Siarni-Namini, S., Tavakoli, N., & Namin, A. S. (2019). The performance of LSTM and BiLSTM in forecasting time series. In *2019 IEEE International Conference on Big Data (Big Data)* (pp. 3285-3292). IEEE.
- Sinha, S., Singh, T. N., Singh, V. K., & Verma, A. K. (2010). Epoch determination for neural network by self-organized map (SOM). *Computational Geosciences*, 14(1), 199-206.
- Smith, L. N. (2017). Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)* (pp. 464-472). IEEE.
- Smith, L. N. (2018). A disciplined approach to neural network hyper-parameters: Part 1--learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*.
- Smola, A. J., Schölkopf, B., & Müller, K. R. (1998). The connection between regularization operators and support vector kernels. *Neural networks*, 11(4), 637-649.
- Sola, J., & Sevilla, J. (1997). Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Transactions on nuclear science*, 44(3), 1464-1468.

- Spencer, J. (1904). On the graduation of the rates of sickness and mortality presented by the experience of the Manchester Unity of Oddfellows during the period 1893–97. *Journal of the Institute of Actuaries*, 38(4), 334-343.
- Sperduti, A., & Starita, A. (1997). Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3), 714-735.
- Stein, J. L. (1961). The simultaneous determination of spot and future prices. *American Economic Review* 51, 1012-1025.
- Sun, R., Chen, Y., & Li, Q. (2007). Modeling and prediction of Great Salt Lake elevation time series based on ARFIMA. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 4806, 1349-1359.
- Sun, R. (2019). Optimization for deep learning: theory and algorithms. arXiv preprint arXiv:1912.08957.
- Sutskever, I., Martens, J., & Hinton, G. E. (2011). Generating text with recurrent neural networks. In *ICML*.
- Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In *International conference on machine learning* (pp. 1139-1147). PMLR.
- Swanson, N. R., & White, H. (1997). A model selection approach to real-time macroeconomic forecasting using linear models and artificial neural networks. *Review of Economics and Statistics*, 79(4), 540-550.
- Takahashi, K., Hayasawa, H., & Tomita, M. (1999). A predictive model for affect of atopic dermatitis in infancy by neural network and multiple logistic regression. *Arerugi=[Allergy]*, 48(11), 1222-1229.
- Tang, Y.; Song, Z.; Zhu, Y.; Yuan, H.; Hou, M.; Ji, J.; Tang, C.; Li, J. (2022) A survey on machine learning models for financial time series forecasting. *Neurocomputing*, 512, 363–380.
- Taylor, S. J., & Letham, B. (2018). Forecasting at scale. *The American Statistician*, 72(1), 37-45.
- Tealab, A. (2018). Time series forecasting using artificial neural networks methodologies: A systematic review. *Future Computing and Informatics Journal*, 3(2), 334-340.
- Tokar, A. S., & Johnson, P. A. (1999). Rainfall-runoff modeling using artificial neural networks. *Journal of Hydrologic Engineering*, 4(3), 232-239.
- Uzair, M., & Jamil, N. (2020). Effects of hidden layers on the efficiency of neural networks. In *2020 IEEE 23rd international multitopic conference (INMIC)* (pp. 1-6). IEEE.

- Valipour, M., Banihabib, M. E., & Behbahani, S. M. R. (2013). Comparison of the ARMA, ARIMA, and the autoregressive artificial neural network models in forecasting the monthly inflow of Dez dam reservoir. *Journal of hydrology*, 476, 433-441.
- Vapnik, V. (1979). *Estimation of dependences based an empirical data*. Moscow: Nauka. English translation, Springer, New York.
- Villavicencio, J. (2010). *Introducción a series de tiempo*. Puerto Rico.
- Wanchen, L. (2020). Analysis on the weight initialization problem in fully-connected multi-layer perceptron neural network. In *2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE)* (pp. 150-153). IEEE.
- Wang, J., Zhang, S., Xiao, Y., Song, R. (2021). A review on graph neural network methods in financial applications, arXiv:2111.15367.
- Watkins, N. W. (2019). Mandelbrot's stochastic time series models. *Earth and Space Science*, 6(11), 2044-2056.
- Widrow, B., & Hoff, M. E. (1960). *Adaptive switching circuits*. Stanford Univ Ca Stanford Electronics Labs.
- Wilson, D. R., & Martínez, T. R. (2001). The need for small learning rates on large problems. In *IJCNN'01. International Joint Conference on Neural Networks Proceedings*, 01CH37222 (1), 115-119.
- Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., & Zhang, C. (2020). Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 753-763).
- Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2021). A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1), 4-24.
- Xie, W., Yu, L., Xu, S., & Wang, S. (2006). A new method for crude oil price forecasting based on support vector machines. In *International conference on computational science* (pp. 444-451).
- Yang, W., Jin, L., & Liu, M. (2015). Chinese character-level writer identification using path signature feature, DropStroke and deep CNN. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)* (pp. 546-550). IEEE.
- Yang, M., & Wang, J. (2022). Adaptability of Financial Time Series Prediction Based on BiLSTM. *Procedia Computer Science*, 199, 18-25.
- Yao, L., Mao, C., & Luo, Y. (2019). Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, 33(01), 7370-7377.

- Ye, M., Zyren, J., & Shore, J. (2002). Forecasting crude oil spot price using OECD petroleum inventory levels. *International Advances in Economic Research*, 8(4), 324-333.
- Yolcu, U., Egrioglu, E., & Aladag, C. H. (2013). A new linear & nonlinear artificial neural network model for time series forecasting. *Decision support systems*, 54(3), 1340-1347.
- Young, M. R. (1996). Robust seasonal adjustment by Bayesian modelling. *Journal of Forecasting*, 15(5), 355-367.
- Yu, L., Wang, S., & Lai, K. K. (2008). Forecasting crude oil price with an EMD-based neural network ensemble learning paradigm. *Energy economics*, 30(5), 2623-2635.
- Yule, G. U. (1927). VII. On a method of investigating periodicities disturbed series, with special reference to Wolfer's sunspot numbers. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 226(636-646), 267-298.
- Zahra, M. M., Essai, M. H., & Abd Ellah, A. R. (2014). Performance functions alternatives of MSE for neural networks learning. *International Journal of Engineering Research & Technology (IJERT)*, 3(1), 967-970.
- Zeiler, M. D., Ranzato, M., Monga, R., Mao, M., Yang, K., Le, Q. V. & Hinton, G. E. (2013). On rectified linear units for speech processing. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 3517-3521). IEEE.
- Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International journal of forecasting*, 14(1), 35-62.
- Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 50, 159-175.
- Zhang, G. P., & Kline, D. M. (2007). Quarterly time-series forecasting with neural networks. *IEEE transactions on neural networks*, 18(6), 1800-1814.
- Zhang, Z., Li, J., Zhu, P., Zhao, H., & Liu, G. (2018). Modeling multi-turn conversation with deep utterance aggregation. *arXiv preprint arXiv:1806.09102*.
- Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into Deep Learning. *arXiv:2106.11342*.
- Zhao, B., Lu, H., Chen, S., Liu, J., & Wu, D. (2017a). Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics*, 28(1), 162-169.
- Zhao, Z., Chen, W., Wu, X., Chen, P. C. Y., & Liu, J. (2017b). LSTM network: A deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, 11(2), 68-75.
- Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., & Li, H. (2019). T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9), 3848-3858.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., & Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, 1, 57-81.