UNIVERSIDAD
DE MURCIA

Facultad
de Informática

Predicción de volatilidad con modelos econométricos y modelos de Machine Learning

Grado en Ingeniería Informática

Trabajo de fin de grado

Autor: Jorge Lorenzo García
Tutor: Lorenzo Fernández Maimó
Junio de 2025

Resumen

El presente trabajo de fin de grado tiene como objetivo principal comparar las metodologías tradicionales de predicción de volatilidad, basadas en modelos econométricos de series temporales, con enfoques modernos sustentados en técnicas de aprendizaje automático (*Machine Learning*, ML). La motivación de este estudio radica en la importancia de la volatilidad como parámetro central en la valoración de derivados financieros, la medición de riesgos y la gestión de carteras, especialmente en un contexto económico global caracterizado por una alta incertidumbre.

El trabajo comienza con una contextualización sobre la relevancia de la volatilidad en el ámbito financiero. Se expone cómo, a raíz de eventos como la crisis financiera de 2008, la predicción precisa de la volatilidad se ha convertido en una necesidad regulatoria y operativa dentro de las entidades financieras. Asimismo, se presentan los conceptos fundamentales de estadística necesarios para comprender las técnicas de modelado: estadística descriptiva, inferencia estadística, regresión y análisis de series temporales.

Una sección clave del marco conceptual compara el enfoque estadístico clásico, denominado *data modeling*, con el enfoque algorítmico propio de la inteligencia artificial, o *algorithmic modeling*. El primero busca establecer relaciones explicativas entre variables mediante modelos con estructura funcional explícita, mientras que el segundo prioriza la capacidad predictiva, sin imponer suposiciones funcionales previas, utilizando modelos flexibles como redes neuronales profundas. Esta distinción metodológica fundamenta la estructura comparativa del trabajo.

Se introducen los fundamentos del mercado financiero, destacando el papel de instrumentos como las opciones y derivados. La volatilidad, definida como la desviación típica de los rendimientos de un activo, aparece como una variable crítica en el modelo de valoración de opciones Black-Scholes-Merton. Esta variable, sin embargo, no es directamente observable, y su predicción se torna un desafío técnico clave. Se presentan dos tipos de volatilidad: la histórica o realizada (calculada sobre retornos pasados) y la implícita (derivada del precio de mercado de opciones).

Adicionalmente, se analiza la volatilidad en su papel dentro de la medición del riesgo de mercado, utilizada para estimar las pérdidas potenciales bajo distintos niveles de confianza, de acuerdo con regulaciones internacionales como Basilea III y Solvencia II.

El estado del arte se divide en dos grandes bloques: modelos econométricos tradicionales y métodos modernos de aprendizaje automático. Dentro de los primeros, se revisan los modelos ARCH y GARCH, introducidos por Engle y Bollerslev respectivamente, y sus variantes como EGARCH y GJR-GARCH, que incorporan asimetrías. También se aborda el modelo HAR, que incorpora datos de alta frecuencia para mejorar la precisión de la predicción.

En cuanto al enfoque de *Machine Learning*, se presentan modelos como Random Forest, SVM y diversas arquitecturas de redes neuronales (ANN, LSTM, TCN), incluyendo aproxi-

maciones híbridas como ANN-GARCH y LSTM-GARCH. Estas combinan las propiedades estructurales de los modelos clásicos con el poder predictivo de las redes neuronales. Se destaca el uso de estos modelos en la predicción de métricas como el Valor en Riesgo (VaR).

La metodología parte de la definición formal de los conceptos de rentabilidad y volatilidad. Se utiliza el rendimiento logarítmico como variable de estudio por sus mejores propiedades estadísticas. La volatilidad se define como la desviación típica de los retornos logarítmicos y se analiza su estimación práctica a partir de precios de cierre diarios, incorporando aspectos como la elección del intervalo temporal y la anualización de la volatilidad.

Empíricamente, se destacan tres propiedades de la volatilidad: su variabilidad en el tiempo, la formación de clústeres de alta volatilidad, y la reversión a la media. Estas características motivan el uso de modelos con memoria, como los basados en series temporales, y justifican la incorporación de arquitecturas como LSTM.

En cuanto al análisis de series temporales, se ofrece una exposición concisa de los modelos estadísticos tradicionales más representativos, como los modelos autorregresivos (AR), de media móvil (MA), sus combinaciones (ARMA) y los modelos integrados (ARIMA), adecuados para series no estacionarias. Asimismo, se consideran los modelos de heterocedasticidad condicional, como los ARCH y GARCH, especialmente relevantes en contextos financieros por su capacidad para modelar la varianza condicional y capturar la presencia de clústeres de alta volatilidad. Estas herramientas, aunque fundamentales desde el punto de vista histórico y teórico, presentan limitaciones en la modelización de dinámicas no lineales o en presencia de comportamientos estructuralmente cambiantes.

Se introducen herramientas como el operador de retardo, la descomposición de Wold, las funciones de autocorrelación (f.a.s y f.a.p), y criterios de estacionariedad. Para los modelos de *Machine Learning*, se presentan tanto redes neuronales feedforward como recurrentes, y especialmente las Long Short-Term Memory (LSTM), que permiten capturar dependencias temporales complejas sin sobreajuste.

Por el contrario, los modelos de *Machine Learning* han cobrado protagonismo en los últimos años debido a su elevada capacidad predictiva y adaptabilidad a datos complejos. Dentro de este enfoque, se exploran distintos tipos de arquitecturas, destacando las redes neuronales artificiales (ANN) de tipo feedforward para tareas generales de regresión, así como las redes neuronales recurrentes (RNN), específicamente diseñadas para el tratamiento de secuencias temporales.

En particular, se profundiza en el uso de las redes Long Short-Term Memory (LSTM), una variante de las RNN que permite capturar dependencias de largo plazo en los datos. Gracias a su estructura de puertas (*input*, *output* y *forget gates*), las LSTM son especialmente adecuadas para capturar patrones persistentes en series financieras, como la memoria de eventos pasados o la persistencia de la volatilidad. Se analiza también el proceso de entrenamiento mediante algoritmos de optimización iterativa, la selección de hiperparámetros y la validación cruzada para evitar el sobreajuste.

En conjunto, las técnicas de *Machine Learning* no solo ofrecen mejoras en la precisión de las predicciones, sino que también abren la puerta al diseño de sistemas predictivos más flexibles, capaces de adaptarse en tiempo real a las condiciones cambiantes de los mercados financieros.

El núcleo experimental del trabajo consiste en un caso práctico de predicción de volatilidad sobre datos reales del mercado de valores. Se construye un conjunto de datos con precios históricos diarios y se calcula la volatilidad realizada. Se comparan diferentes modelos: modelos GARCH y redes LSTM.

Los resultados obtenidos muestran que los modelos de *Machine Learning*, especialmente las redes LSTM, superan en rendimiento predictivo a los modelos GARCH en la mayoría de los escenarios evaluados, lo que se traduce en una menor tasa de error y una mayor capacidad para capturar las dinámicas temporales de la volatilidad. No obstante, se ha evidenciado que la predicción de la volatilidad continúa siendo un desafío considerable, especialmente ante la aparición de eventos exógenos inesperados (shocks). Los modelos univariantes, al basarse exclusivamente en la dinámica histórica de la propia variable, carecen de la capacidad para anticipar estos eventos, limitándose a describir su evolución una vez que han tenido lugar. Esta limitación estructural pone de manifiesto la conveniencia de incorporar variables externas o información contextual en los modelos predictivos, lo que representa una prometedora línea de investigación futura.

Extended abstract

The main objective of this thesis is to carry out an exhaustive comparison between two major methodological paradigms used in the prediction of financial volatility: on the one hand, traditional econometric techniques based on time series analysis, and on the other, modern approaches based on Machine Learning (ML). The choice of this topic responds to the growing need, both regulatory and operational, for robust and accurate models to anticipate the volatile behavior of financial markets, especially in a global context characterized by strong economic interdependence and high levels of uncertainty. In particular, regulatory frameworks such as Basel III and the increasing complexity of financial instruments have pushed institutions to adopt more sophisticated predictive tools to comply with capital requirements and manage portfolio risk more effectively.

After critical events such as the 2008 financial crisis, it became clear that financial systems needed more effective tools to anticipate and mitigate systemic risk. In this framework, volatility emerges as a central variable. Defined as the standard deviation of the returns of a financial asset, it represents a quantitative measure of market uncertainty and is key both in the valuation of derivative products (as in the Black-Scholes-Merton model) and in the estimation of market risk through metrics such as Value at Risk (VaR). Volatility is not only a technical measure but also a practical input for decision-making in trading, asset allocation, and regulatory compliance, making its accurate forecasting a priority in financial research and practice.

This thesis therefore starts from a twofold hypothesis: first, that traditional econometric models offer a solid and proven theoretical framework for volatility prediction, but present certain limitations in highly nonlinear or structurally changing environments. Second, that Machine Learning models, especially deep neural networks, present significant potential in terms of their ability to model complex and adaptive relationships in financial data. These models, by leveraging large volumes of data and nonlinear functional forms, can uncover patterns that are otherwise invisible to classical approaches.

From a methodological perspective, this work is based on a fundamental distinction between two approaches to data modeling: the classical statistical approach or data modeling and the algorithmic approach or algorithmic modeling, as proposed by Leo Breiman. The former is based on the definition of an explicit functional structure that relates explanatory variables to a response variable, under a set of theoretical assumptions that allow statistical inferences to be made. This methodology has been the basis of econometrics and time series analysis since the mid-20th century, offering clarity, analytical tractability, and robustness in well-behaved systems.

In contrast, the algorithmic approach does not impose a rigid functional form, but uses highly flexible and adaptive models that directly optimize an error function on a data set, usually by iterative fitting processes such as gradient descent. This approach, common in the field of artificial intelligence and machine learning, does not seek a direct interpretation of the parameters, but rather seeks to maximize the predictive capability

of the model on unobserved data. This is particularly useful in financial markets, where unknown structural breaks and nonlinear dynamics can undermine traditional parametric modeling.

This difference is fundamental to understand the complementarity between both methodologies. While econometric models offer greater interpretability and theoretical consistency, ML models sacrifice this in exchange for a greater ability to capture complex and nonlinear patterns. This thesis sets out to explore this tension in the specific context of volatility forecasting, where the interplay between model structure and predictive accuracy becomes especially relevant.

In the analysis of econometric models, representative models such as ARCH, GARCH, and EGARCH have been reviewed and applied to capture the empirical properties of volatility, such as conditional heteroskedasticity, time persistence, and asymmetric effects. These models have been widely used in financial environments and enjoy a solid theoretical basis. Their widespread adoption stems from their capacity to model stylized facts observed in financial time series, including volatility clustering and leverage effects.

However, they have important structural limitations. For example, the imposition of symmetry in the treatment of positive and negative shocks (except in variants such as EGARCH), sensitivity to the choice of model orders, and rigidity in the representation of complex long-run dynamics. These drawbacks may hinder their performance in situations involving abrupt market shifts or regime transitions.

In the field of machine learning, work has been done with models such as neural networks, especially advanced architectures such as LSTM (Long Short-Term Memory). LSTMs, due to their ability to retain long-term information in temporal sequences through internal memory mechanisms, have proven to be particularly effective in the prediction of phenomena with high time dependence, such as financial volatility. Unlike traditional models with fixed lag structures, LSTMs can dynamically adjust the relevance of past observations over time, allowing them to capture more subtle temporal dependencies.

The analysis carried out throughout this thesis reveals that, although econometric and artificial intelligence-based approaches start from different assumptions, there are important points of convergence. Both approaches share the objective of predicting a future variable — in this case, volatility — from historical data. Also, many ML models use as input variables the past observations of the series, replicating the autoregressive logic of models such as AR(p). In both paradigms, the underlying idea is that past patterns may contain signals about future behavior, though they operationalize this idea differently.

However, the differences are profound in terms of implementation and flexibility. Econometric models require strong assumptions about the distribution of errors, stationarity and linearity of relationships, which sometimes restricts their applicability. These assumptions, while ensuring interpretability and theoretical consistency, can become a liability when markets exhibit non-stationary behavior or are subject to frequent regime changes. In contrast, ML models start with greater structural freedom, which allows them

to better adapt to highly dynamic environments with structural noise. This capability is especially valuable in modern financial markets, characterized by their complexity, interconnectedness, and propensity for abrupt changes that do not follow stable probabilistic laws.

One of the most remarkable aspects is that, despite the fact that LSTMs were not specifically designed for financial volatility prediction, they have achieved notoriously competitive results against models traditionally used in this field, such as GARCHs. This fact has profound implications: it reveals, on the one hand, the robustness and versatility of LSTMs; on the other hand, it highlights the possibility that generalist algorithmic models, although initially conceived for other tasks (e.g., natural language processing), can be effectively adapted to financial problems with the right configuration. Their recurrent structure and gating mechanisms allow them to capture long-range dependencies and non-linear relationships, which are often present but difficult to formalize in financial time series.

This generalization capacity of models such as LSTM, added to their architecture based on long-term memory mechanisms, allows them to model dependencies that escape the finite lag structure imposed by econometric models such as GARCH. Still, it is important to note that this predictive power is not automatic. It requires in-depth knowledge of the financial domain to select the relevant variables, tune the model architecture, and understand the underlying mechanisms affecting markets. Hyperparameter optimization, data preprocessing, and feature engineering all depend heavily on a nuanced understanding of financial structures, investor behavior, and institutional context.

Without a clear understanding of the financial fundamentals, there is a risk of building models that, while showing a good fit in the training phase, do not generalize well to real scenarios and do not allow for a meaningful interpretation of the results. In high-stakes applications such as portfolio management or risk monitoring, such limitations can lead to costly misjudgments.

Therefore, the importance of field knowledge in the success of ML modeling is highlighted here. A predictive model cannot be considered effective if it is not supported by an adequate understanding of the problem it addresses. In the financial domain, this implies knowing not only the statistical nature of the time series, but also the economic, regulatory, and behavioral factors that influence volatility. Such knowledge is essential for designing the model architecture, interpreting its predictions, and assessing its limits. In fact, much of the success of an ML model applied to finance comes from its hybridization with economic intuitions and domain expertise. Without this hybrid foundation, even the most sophisticated algorithm can fall short in delivering actionable insights.

It is worth noting that the recent evolution of ML models has come from computer engineering — driven by improvements in hardware, optimization algorithms and data availability — but it is equally true that much of the advances in deep neural networks are based on sophisticated mathematical concepts such as linear algebra, differential calculus, Bayesian statistics or information theory. It is this combination of mathematical rigor and

computational power that has enabled the explosion of ML applications in such demanding fields as quantitative finance.

However, it is computer science that has materialized these mathematical ideas in functional systems. The development of platforms such as TensorFlow or PyTorch, the appearance of specialized libraries such as arch for econometric models or scikeras for neural networks, and the possibility of performing distributed training on GPUs, have been key elements that have made the use of complex models viable in professional and academic practice.

In the case of volatility prediction, this evolution has been particularly remarkable. LSTM architectures, originally developed for natural language processing tasks, have proven to be effective in the treatment of financial series due to their ability to model long-range structures and avoid gradient loss in long series. Their adaptation to the financial field, however, has required significant work.

To validate the robustness of the models, they have been tested in two different markets and two time windows with different volatility characteristics have been defined —one including the COVID-19 crisis, and another more recent one— to validate the robustness of the models in different market environments.

The models evaluated include GARCH(1,1), EGARCH(1,1) and LSTM, trained on daily logarithmic returns. The prediction has been evaluated by rolling forecast, i.e., a one-day iterative forecast on a test set. The models have been evaluated according to mean squared error (MSE), among other metrics.

The results show that LSTM networks present, in most scenarios, superior performance in terms of predictive accuracy. This translates into a better ability to anticipate changes in volatility and adapt to changing dynamics. However, certain limitations are also observed: the need for hyperparameter calibration, sensitivity to data scaling, and greater difficulty in interpreting the internal mechanisms of the model.

This work has allowed us not only to empirically evaluate advanced models for volatility prediction, but also to reflect on the role that statistics and computer science play in modern quantitative analysis. It has become evident that volatility prediction remains a complex task, especially in the face of unexpected exogenous events (shocks). Since univariate models can only capture historical patterns, they are not able to anticipate these events, but only to model their propagation. This structural limitation highlights the need to incorporate exogenous or contextual variables, such as macroeconomic indicators or news flows, which represents a promising line of research.

In this sense, an ideal model would be one capable of integrating, in real time, financial data with contextual information from economic, political or social news, automatically detecting events likely to generate an abrupt change in volatility. This model would not be limited to capturing past price dynamics, but would incorporate natural language processing (NLP) mechanisms to interpret press headlines, macroeconomic reports, publications

on social networks or official communications, extracting from them anticipatory signals of possible shocks.

Such a hybrid model would combine sequential architectures such as LSTM or Transformers for time series and text analysis, together with econometric blocks to ensure structural consistency and statistical robustness. Training would be performed on a continuous data stream, allowing dynamic adjustments according to the current market context. This approach would allow a greater capacity for anticipation, overcoming the intrinsic limitation of univariate models and approaching the ideal of proactive prediction in complex financial scenarios.

Likewise, this work suggests that hybrid models —which combine the theoretical structure of econometric models with the adaptive capacity of machine learning— constitute a particularly fertile avenue for future research. Models such as ANN-GARCH or LSTM-GARCH have already shown promising results in this regard.

Finally, we conclude that research in quantitative finance should be nourished both by the theoretical rigor of statistics and by the flexible and scalable tools provided by computer science. The future of financial forecasting does not lie exclusively in one or the other approach, but in its intelligent and critical integration, in which knowledge of the financial domain remains the articulating axis guiding the choice of models, variables, and validation methodologies.

Índice

1 Introducción

En las últimas décadas, los mercados financieros han jugado un papel crucial en la economía debido a su importancia para transferir dinero entre sectores, países o personas. No es de extrañar que haya aumentado su importancia con el progreso económico global de los siglos XX y XXI. El trabajo que se realiza en el sector de las finanzas se ha visto sustancialmente modificado debido al auge de las tecnologías informáticas. Comenzando por la conexión global instantánea que provee el internet —que da la posibilidad de transferir dinero rápida y eficazmente—, como con la disponibilidad de gran cantidad de datos financieros y la capacidad de cómputo que ofrecen los ordenadores actuales. Todo esto no evitó la **crisis financiera del año 2008**, y desde entonces, las autoridades estatales, o supraestatales, de supervisión bancaria y financiera han puesto el foco en la protección de la economía frente a la **incertidumbre** inherente a las finanzas.

La incertidumbre es la base de las finanzas. En un préstamo, el dinero que debes pagar en concepto de intereses es relativo a la probabilidad de que no lo devuelvas. Y, como veremos más adelante, en las opciones financieras el principal parámetro del modelo es la **volatilidad**. La volatilidad es la forma en la que se expresa la incertidumbre en los mercados financieros. La volatilidad mide cuánto se mueve el precio de un activo financiero. En términos probabilísticos, es la desviación típica del cambio en el precio de un activo. Por tanto, es crucial conocer su valor para poder introducirla en los modelos de riesgos y de valoración de opciones. El principal inconveniente para ello es que la volatilidad no es estática, sino que su valor cambia con el tiempo. Por tanto, uno de los principales retos en los modelos financieros es conocer qué valores tomará la volatilidad. He aquí el objetivo de este trabajo, predecir la volatilidad y comparar los modelos tradicionales con los más contemporáneos, basados en **Machine Learning**. Resumidamente, nos gustaría que este trabajo sirviera para lo siguiente:

- Explicar y diferenciar las metodologías estadística e informática para la predicción de series temporales.
- Explorar las dificultades que presenta la predicción de la volatilidad.
- Probar sobre datos reales qué modelos predicen mejor la volatilidad de series financieras.

Más adelante retomaremos este objetivo con más profundidad, pero ahora vamos a contextualizar el trabajo. Por un lado, se presentará un marco estadístico general que emplearemos para la predicción de la volatilidad, así como sus diferencias y semejanzas con las técnicas más contemporáneas, basadas en Inteligencia Artificial. Por otro lado, motivaremos el uso de instrumentos financieros, como las opciones, con el fin de ilustrar la utilidad práctica de esta predicción.

1.1 'Big picture' de la estadística

Dado que la estadística será un tema central en este trabajo, conviene recordar brevemente los objetivos de sus principales ramas. Por un lado, tenemos la **estadística descriptiva**, cuyo propósito es resumir, organizar y presentar de forma clara la información contenida en un conjunto de datos. Esto se logra mediante medidas numéricas como la media, la mediana o la varianza, así como a través de representaciones gráficas como histogramas o diagramas de dispersión. Su función principal no es solo calcular ciertos valores, sino ofrecer una visión general del comportamiento de los datos, facilitando su interpretación y permitiendo detectar patrones, tendencias o posibles anomalías. Por otro lado, tenemos la estadística avanzada, que va más allá de la mera descripción y trata de modelizar la realidad para contrastar teorías y predecir futuros valores.

Dentro de esta se suele estudiar primeramente la **inferencia estadística**. La inferencia estadística intenta, en base a datos estadísticos, identificar valores y comprobar conjeturas sobre la población de la que provienen estos datos. Esta identificación de valores se hace mediante la estimación y los intervalos de confianza, y la comprobación, mediante el contraste de hipótesis. En general, el problema de la inferencia es: dada una población, saber su distribución y los parámetros que la describen. Normalmente, una distribución tiene ciertos parámetros asociados. Por ejemplo, la distribución normal tiene una forma que se adapta a la media y la varianza. Si suponemos que una población sigue una distribución normal, estimando su media y varianza la distribución que sigue dicha población queda totalmente determinada. Tras esto, solo nos faltaría contrastar la hipótesis de que, efectivamente, la población sigue una distribución normal.

Otra clase de conjeturas es suponer una cierta relación entre variables. Por ejemplo, cuanta más educación recibe una persona, más probabilidad de tener un mayor sueldo. Con este objetivo, se definen los **modelos de regresión**.

Y, por último, cuando una variable se explica en gran parte en función de los valores que ha ido tomando a lo largo del tiempo, se usan las llamadas **series temporales**, que serán el caso de estudio de este trabajo.

Se puede ver un resumen de estas ideas en la Figura 1.

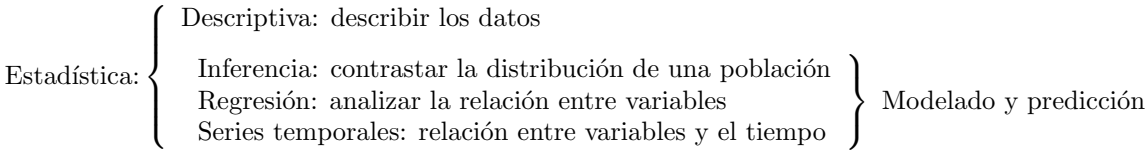


Figura 1: Las ramas de la estadística y su uso. Fuente: elaboración propia

En todos estos casos, la descripción de las poblaciones y la relación entre variables nos puede servir para predecir, pero esta predicción es siempre con incertidumbre. Estos modelos nos dan relaciones estadísticas que representan la realidad con aleatoriedad, no

por capricho, sino porque la naturaleza del problema es aleatoria o es imposible tener en cuenta todas las variables que afectan al resultado. En este último caso, normalmente se agrupa la incertidumbre que generan estas variables no controladas bajo la hipótesis de normalidad y cierta varianza. Esto se basa en la idea de que la suma de variables independientes y con poca influencia tiende a parecerse a una distribución normal. Esta idea está basada en el **Teorema Central del Límite**. Evidentemente, esta suposición es fuerte y hay que comprobarla con un test de hipótesis para confirmar que el modelo es explicativo de la realidad.

1.1.1 Diferencias entre técnicas estadísticas de predicción y técnicas basadas en IA

Siguiendo la taxonomía propuesta por Leo Breiman en [Bre01], la forma clásica de modelar los datos trata de imponer una forma funcional, dependiente de ciertos parámetros, a la relación entre las variables. Bajo este enfoque, se busca estimar dichos parámetros que hacen que esta forma funcional se adapte mejor a los datos. Esto se realiza, principalmente, mediante un problema de optimización que consiste en maximizar la función de verosimilitud del modelo. Finalmente, una vez tenemos los parámetros estimados, comprobamos si dicha forma funcional tiene sentido para los datos mediante tests de especificación. A esta aproximación, Breiman la denomina ***data modeling***.

Existe otro punto de vista, al que Breiman llama ***algorithmic modeling***, que consiste en no hacer asunciones sobre la forma funcional de los datos —dicho de otro modo, dotar al modelo de una elevada flexibilidad—, con el objetivo de reducir el error en la predicción. Dentro de esta categoría se encuentran las tecnologías basadas en Inteligencia Artificial. De forma simplificada, el proceso más habitual es tomar un modelo muy flexible, como puede ser una red neuronal profunda, y elegir una función de error a minimizar, usualmente basada en una medida de distancia. De tal manera que, en el entrenamiento, mediante un proceso iterativo —como un descenso del gradiente—, se trata de mover los parámetros del modelo, como el peso de las neuronas, para minimizar el error. Posteriormente, se prueba en un conjunto de validación o de test para probar si el modelo predice bien.

Tal como se ha presentado, ambos enfoques comparten una estructura básica: en primer lugar, se selecciona un modelo; en segundo lugar, se estiman sus parámetros; y, finalmente, se evalúa su desempeño. No obstante, la principal diferencia radica en el objetivo final: mientras que el data modeling está centrado en entender y validar la estructura funcional de los datos, el algorithmic modeling se enfoca en la eficacia de la predicción.

En la **elección del modelo**, el modelado de datos impone una forma funcional, mientras que el modelado algorítmico propone modelos muy flexibles y con muchos parámetros. Estos serán buenos adaptándose a muchas formas funcionales; por ello, decimos que no hacemos asunciones sobre los datos. A consecuencia de esto, será más difícil explicar cómo el modelo relaciona las variables para realizar predicciones.

Aspecto	Data Modeling	Algorithmic Modeling
Supuesto inicial	Se impone una forma funcional explícita para la relación entre variables.	No se hacen supuestos funcionales; se usan modelos muy flexibles.
Objetivo principal	Explicar la relación entre variables; interpretación.	Minimizar el error de predicción.
Modelo	Modelo con forma funcional fija (ej. regresión lineal).	Modelo altamente flexible (ej. redes neuronales profundas).
Estimación de parámetros	Mediante la minimización de la función de verosimilitud.	Mediante minimización de una función de error.
Optimización	Ajuste de parámetros dentro de una forma funcional fija.	Ajuste iterativo (ej. descenso del gradiente) que puede cambiar la forma funcional.
Evaluación del modelo	Pruebas de especificación para validar si el modelo representa bien los datos.	Evaluación en conjunto de test/validación para comprobar la capacidad predictiva.
Interpretabilidad	Alta; se puede explicar la influencia de cada variable.	Baja; difícil interpretar cómo se hacen las predicciones.

Tabla 1: Comparación entre Data Modeling y Algorithmic Modeling. Fuente: elaboración propia.

Para la **estimación de parámetros**, ambos resuelven un problema de optimización. El modelado de datos minimiza una función de verosimilitud que viene dada por el modelo elegido, mientras que el modelado algorítmico minimiza la función de error cambiando los parámetros. El detalle es que la modificación de estos parámetros cambia la forma funcional que relaciona las variables.

Finalmente, la **evaluación del modelo**, desde el punto de vista del modelado de datos consiste en comprobar que los datos encajan en el modelo. Es decir, se centra en haber acertado con la forma que hemos supuesto para las variables. Mientras que el modelado algorítmico se centra, simplemente, en evaluar si las predicciones son acertadas. En el caso de *algorithmic modeling*, este paso puede ser distinto si la forma funcional del modelo también tiene parámetros, conocidos como **hiperparámetros**, que se eligen previo al entrenamiento. En ese caso, este paso también sirve para elegir los hiperparámetros y, posteriormente, evaluar. Estas diferencias están expresadas en la Tabla 1.

Nuestro caso concreto se centra en el estudio y la predicción de la volatilidad. Para contextualizar este objetivo y motivar el desarrollo del trabajo, se incluye a continuación un pequeño epígrafe sobre el funcionamiento de los mercados y las opciones financieras.

1.2 Introducción al mercado financiero, valoración de opciones y medición de riesgos

El **mercado financiero** es el lugar, virtual o presencial, donde se compran y venden instrumentos financieros. Los instrumentos financieros son muy diversos y explicarlos todos excedería el propósito de este trabajo. Por ello, simplemente explicaremos qué son y motivaremos su utilidad, puesto que es importante para el sentido del trabajo.

Las empresas requieren de financiación para ejecutar sus tareas necesarias para vender productos o servicios. Esta financiación puede provenir de dos apartados bien diferenciados: fondos propios y deudas. Los fondos propios son el dinero que ponen los dueños o socios de la empresa.

La financiación mediante deuda consiste en pedir dinero por un tiempo determinado y devolverlo, una vez cumplido ese tiempo, con unos intereses al acreedor. Dos ejemplos de esto son un préstamo o crédito del banco, o los bonos de deuda que puede emitir una empresa o Estado.

Conseguir financiación mediante fondos propios consiste en vender parte de la propiedad de la empresa. Es el mecanismo que suelen utilizar las start-ups y, más popularmente, el que usan las empresas que cotizan en bolsa. Estas ofrecen parte de la propiedad de su empresa de forma que se pueda comprar y vender libremente por cualquiera. De la oferta y demanda que se genere en torno a estas participaciones se determina su valor y, por tanto, el de la empresa. Se puede ver una ilustración de esto en la Figura 2.

Formas de financiación:

$$\begin{cases} Vender\ fondos\ propios \rightarrow \begin{cases} Salir\ a\ bolsa \\ Vender\ porcentaje\ de\ la\ empresa\ a\ inversor \end{cases} \\ Aumentar\ deuda \rightarrow \begin{cases} Préstamo\ bancario \\ Emisión\ de\ bonos \end{cases} \end{cases}$$

Figura 2: Formas de financiación de una empresa. Fuente: elaboración propia.

De estas ideas básicas surgen la mayoría de instrumentos financieros. Una lista básica sería: bonos del Estado, bonos corporativos, hipotecas fijas o variables, titulizaciones, acciones y fondos indexados. Sin embargo, hay otro tipo de instrumentos que no son creados para proporcionar financiación sino para limitar o modificar riesgos. Se llaman *derivados* porque siempre se constituyen en base al valor de otro activo o instrumento llamado *subyacente*, normalmente uno de los de la lista anterior.

Ilustremos con un ejemplo la utilidad de estos derivados. Un agricultor que siembra en febrero para cosechar en septiembre puede querer eliminar la incertidumbre del precio por el que venderá su cosecha. El contrato derivado que le sirve para asegurarse un precio de venta para septiembre es un **futuro o forward**. También necesitará un derivado una empresa que venderá sus productos en Estados Unidos por dólares, pero prefiera asegurarse

ese dinero en euros. Un último caso: una empresa que posee un préstamo ligado al Euríbor pero quiere transformarlo a tipo fijo para eliminar la incertidumbre necesitará un **swap**. Un swap sirve para intercambiar pagos a un tipo fijo, por ejemplo, un 2%, frente a un tipo variable que se desconoce a inicio, como el Euríbor. De forma que, de cara a la empresa, reciben un tipo variable con el que pagan el préstamo y pagan un tipo fijo. De esta forma, la empresa se ha asegurado un pago a un tipo fijo eliminando así el riesgo por tipo de cambio.

Los derivados que más nos interesan a nosotros son las **opciones**. Los contratos de opciones te dan el derecho, pero no la obligación, de comprar o vender un subyacente al precio determinado en el contrato. Por ejemplo, una onza de oro en un año a 5000€. En un año, si el precio del oro es superior a los 5000€, el poseedor de la opción tendrá beneficios por el valor de la diferencia entre el precio y los 5000€. Si el precio del oro es inferior a 5000€, el poseedor de la opción no ejercerá su derecho. Si necesita oro, podrá ir al mercado y adquirirlo más barato.

Entre los *forwards* y las opciones hay una diferencia clave: los *forwards* son acuerdos bilaterales, mientras que las opciones se compran. Es decir, para los *forwards* hay un precio que hace que el contrato no tenga valor inicial, porque ambas partes están de acuerdo y consideran que se benefician. En cambio, las opciones las vende una entidad a un coberturista o un especulador, que solo tiene posibilidad de ganar dinero el día de vencimiento del contrato. Por tanto, las opciones no pueden valer 0.

Pero entonces, ¿qué precio tienen las opciones? Para responder a esta pregunta fue necesario desarrollar una teoría matemática que desembocó en un Premio Nóbel de Economía en el año 1997. Lo recibieron Scholes y Merton por el desarrollo del modelo Black-Scholes-Merton, cuya ecuación paradigmática la ecuación de Black-Scholes.

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0$$

En esta ecuación aparece σ, que representa la volatilidad. Esto es, cómo de sensible es el precio del subyacente. Una mayor volatilidad implica mayor incertidumbre, lo que conlleva más riesgo y, por tanto, más precio. La volatilidad es un elemento central en la valoración de opciones y se define como la desviación típica de los precios a lo largo del tiempo.

Dado que la predicción exacta del valor de un precio futuro es imposible—aunque existen intentos por lograrlo, si se conociera con certeza no existiría riesgo ni posibilidad de negocio alguno en su cobertura—, lo que se hace es intentar valorar cómo de cambiante, i.e. volátil, será dicho precio durante el periodo de vida de la opción.

Posteriormente, se desarrollaron diversos modelos, todos ellos con un elemento en común: la necesidad de asignar un valor a la volatilidad, o más precisamente, de estimar sus valores a lo largo del tiempo. Es en este punto donde se encuentra el propósito de nuestro trabajo: **intentar predecir la volatilidad**.

Otro ámbito financiero donde la volatilidad es relevante es en la **medición de riesgos**. El objetivo de la medición de riesgos es estimar las pérdidas potenciales que puede tener una institución. Existen diversos tipos de riesgos (de crédito, de liquidez, operacional, etc.), pero es en el **riesgo de mercado** donde la volatilidad juega un papel prioritario.

El riesgo de mercado se refiere a la posibilidad de incurrir en pérdidas debido a variaciones en los precios de los instrumentos financieros que conforman la cartera de una entidad. Su cuantificación implica el uso de métricas y metodologías de cálculo que tienen como objetivo la definición de una distribución probabilística de las posibles pérdidas y ganancias. Estos métodos, en lugar de predecir subidas o bajadas de los precios, modelan la distribución con la volatilidad como input.

Tras la crisis financiera del año 2008, estas mediciones han tomado especial relevancia por la entrada en vigor de reglamentos en instituciones financieras supervisadas por organismos estatales como los bancos centrales. En estos reglamentos, se exige a las entidades financieras hacer un cálculo de las pérdidas potenciales por distintos conceptos. Uno de estos conceptos es el riesgo de mercado, expresado como las máximas pérdidas potenciales por gestión de cartera de activos al 5% de confianza. En esta estimación probabilística, uno de los parámetros más relevantes, que elevan o disminuyen el resultado de la medición, es la volatilidad.

2 Estado del arte

La predicción de volatilidad comenzó con los modelos de estilo ARCH. Engle ([Eng82]) introdujo el modelo **ARCH**, definiendo la varianza condicional como función de errores pasados. Bollerslev ([Bol86]) generalizó ARCH al proponer **GARCH**, que incorpora también la propia varianza condicional en momentos previos. Estos modelos capturan el *cluster* de volatilidad característico de algunas series de datos económicos. Inicialmente fue aplicado para datos sobre inflación y poco después fue aplicado para datos financieros donde surgieron diversas variantes –por ejemplo **GJR-GARCH** ([GJR93]) o **EGARCH**([Nel91]) – que incorporan efectos asimétricos en la varianza. Con la mejora en la accesibilidad a datos del presente siglo se han visto mejoras en la predicción de la volatilidad realizada con datos intradía introducida por Bollerslev ([AB98]). El ejemplo paradigmático de estos modelos es el modelo **HAR** ([Cor09]) que goza de amplia aceptación. Estas formulaciones econométricas siguen siendo la base tradicional para modelar la volatilidad.

Con el auge de la inteligencia artificial, numerosas técnicas de machine learning se han aplicado a la predicción financiera. Se han empleado, entre otros, **random forest** ([LD18]), **redes neuronales** (ANN) y **máquinas de vector soporte** (SVM) ([TTS09]). Dentro de las redes neuronales se han probado múltiples arquitecturas como **redes temporales convolucionales** (TCN) ([ZLH+22]) y **Long Short-Term Memory** (LSTM) ([Liu19]). También, se han propuesto enfoques híbridos que incorporan los conocidos como *hechos simplificados* de los modelos clásicos, que describen formas de comportamiento de la volatilidad observadas históricamente, en arquitecturas basadas en redes neuronales como ANN-GARCH ([RP17]) o LSTM-GARCH ([KW18], [KJM22]).

Entre las aplicaciones actuales de la predicción de volatilidad se encuentra el cálculo de medidas de riesgo como el **VaR** ([KJM22], [GRLM04]), exigidas a las entidades financieras en los últimos reglamentos internacionales, como Basilea III ([SO13]) para bancos y Solvencia IIv([Bol15]) para aseguradoras y reaseguradoras.

3 Objetivos y metodología

El objetivo del presente trabajo es comparar técnicas tradicionales basadas en series temporales con las más recientes, basadas en aprendizaje automático, para predecir la volatilidad en los mercados financieros. En concreto, queremos ver hasta qué punto los modelos modernos, como redes neuronales o variantes de modelos secuenciales, son capaces de captar el comportamiento cambiante de la volatilidad en series temporales financieras.

Asimismo, trataremos los detalles de la predicción de volatilidad y discutiremos si la metodología estándar de predicción mediante aprendizaje automático es adecuada.

Para poder discutir estas cuestiones, es necesario comprender cómo funciona la predicción tanto con series temporales como con Machine Learning. Con este fin, desarrollaremos los siguientes apartados:

- Definición formal de **rentabilidad** y **volatilidad**. Debido al carácter, en ocasiones, confuso de la volatilidad, es importante tener claro qué es lo que estamos intentando predecir y cómo lo vamos a medir. Vamos a darle una base matemática y estadística, definiéndola con precisión a partir de los retornos de los precios.
- Introducción a las **series temporales**. Son el tipo de datos con el que vamos a trabajar. Realizaremos una introducción conceptual original de este trabajo.
- Aplicación de modelos de **aprendizaje automático** a problemas de series temporales. Explicaremos en qué sentido se parecen este enfoque y el basado en series temporales, e introduciremos los modelos más ampliamente usados, como las redes neuronales y las LSTM.
- Consideraciones específicas sobre la **predicción de la volatilidad**. Aquí discutiremos por qué este problema tiene ciertas particularidades frente a otras tareas de predicción, qué dificultades aparecen, así como por qué la metodología estándar de predicción puede inducir malos resultados.

3.1 Rentabilidad y volatilidad

En esta sección definiremos formalmente las nociones de **rentabilidad** y **volatilidad**. También mencionaremos algunas de sus propiedades empíricas. Las definiciones de este apartado están extraídas de [Tsa05].

Sea P_t el precio de un activo en tiempo t. Vamos a asumir que no paga dividendos. Entonces:

Definición 3.1 *La rentabilidad simple bruta de tener un activo de tiempo* $t-1$ *a* t *es* $1+R_t = \frac{P_t}{P_{t-1}}$ *o* $P_t = P_{t-1}\cdot(1+R_t)$ *. La rentabilidad simple neta es* $R_t = \frac{P_t}{P_{t-1}} - 1 = \frac{P_t - P_{t-1}}{P_{t-1}}$.

Como se puede observar, si conocemos el precio en tiempo $t-1$ y la rentabilidad del período, podemos calcular el precio del tiempo t. Esto es válido período a período y es a lo que nos referimos cuando afirmamos que la rentabilidad es un resumen completo del precio del activo, siempre que se conozca el precio inicial.

Por otro lado, una de las propiedades más útiles de los rendimientos puede deducirse mediante aritmética básica. La rentabilidad bruta de tener el activo durante k períodos, esto es, de $t-k$ a t se expresa como $1+R_t[k]=\frac{P_t}{P_{t-k}}=\frac{P_t}{P_{t-1}}\cdot\frac{P_{t-1}}{P_{t-2}}...\cdot\frac{P_{t-k+1}}{P_{t-k}}=\prod_{j=0}^{k-1}(1+R_{t-j})$. O sea, la rentabilidad de mantener el activo durante k períodos no es más que la composiciones de las rentabilidades individuales de cada período.

Para la rentabilidad neta, basta con restar 1 a ambos lados de la ecuación: $R_t[k]=\prod_{j=0}^{k-1}(1+R_{t-j})-1$. En la práctica, se desea comparar las rentabilidades en una misma escala de tiempo, que usualmente es el año.Por ejemplo, si obtenemos un 3% de rentabilidad en dos activos, pero en uno la conseguimos en un año y en el otro en dos años, resulta más justo comparar ambas en términos anuales. Es decir, debemos calcular qué rentabilidad anual equivale a lograr un 3% en dos años. Dado que las rentabilidades acumuladas en k períodos se obtienen mediante la composición multiplicativa de cada período, la rentabilidad anualizada correspondiente es, precisamente, la media geométrica.

Definición 3.2 *La rentabilidad neta anualizada de un activo mantenido durante k años es* $Anualizado\{R_t[k]\}=\left[\prod_{j=0}^{k-1}(1+R_{t-j})\right]^{1/k}-1$. *Pudiendo ser k una fracción de año.*

Esto se puede reescribir en términos de suma si introducimos la exponencial y el logaritmo en el productorio.

$$Anualizado\{R_t[k]\}=exp(\frac{1}{k}\cdot\left[\sum_{j=0}^{k-1}ln(1+R_{t-j})\right])-1$$

Definición 3.3 *El logaritmo de la rentabilidad neta de un activo de tiempo* $t-1$ *a* t *conocida como log rentabilidad o log-retorno es :* $r_t=ln(1+R_t)=ln(\frac{P_t}{P_{t-1}})=p_t-p_{t-1}$. *Siendo* $p_t=ln(P_t)$. *Y* R_t, *la rentabilidad neta de ese período.*

Una de las ventajas de usar **log-retornos** es que la rentabilidad logarítmica en k períodos es la suma de la rentabilidades logarítmicas en cada período: $r_t[k]=ln(1+R_t[k])=r_t+r_{t-1}+...+r_{t-k+1}$. Por esta razón, son utilizados en las modelizaciones matemáticas. Además, cuando R_t es cercano a 0, por la serie de Taylor del logaritmo, r_t se aproxima bastante a R_t. Por último, a nivel estadístico, los log-retornos tienen mejores propiedades. Por tanto, la variable de interés a estudiar a nivel estadístico, probabilístico y predictivo serán los log-retornos. Como veremos a continuación, realmente estudiaremos la volatilidad, o sea, su desviación típica.

Ahora vamos a tratar la definición de **volatilidad**, varias consideraciones clave y algunas de sus propiedades.

Definición 3.4 *Definimos la volatilidad de un activo como la desviación típica que sigue su serie de precios P_t.*

Una vez definida, surge la cuestión de cómo medirla. Al fin y al cabo, la desviación típica se aplica sobre valores que consideramos aleatorios. En este caso, además, estamos trabajando con una serie temporal de valores aleatorios. Realmente, si suponemos que la serie de precios P_t es una variable aleatoria (una serie de ellas en realidad), la volatilidad no es directamente observable, sino que es una propiedad de la distribución que sigue P_t. Por tanto, lo que debemos de hacer es estimar dicha probabilidad. Normalmente, se estima mediante la **cuasi-desviación típica** de los log-retornos entre precio de cierre de días. Por ejemplo, la bolsa española cierra a las 17:00 de la tarde. Por tanto, se tomaría como precios de referencia para el cálculo de los log-retornos de un día, el precio de cierre de ese día a las 17:00 y el precio del día hábil previo a las 17:00.

Aunque esta definición es estándar y operativa, presenta varios matices importantes:

1. En lugar de utilizar log-retornos diario se podría optar por otros saltos. Por un lado se podría tomar un intervalo temporal diferente. Por ejemplo, se podrían intervalos más cortos —cada hora, cada minuto—, o más largos —cada semana, cada mes—. Por otro lado, en lugar del cambio entre precios de cierre, podría tomarse, por ejemplo, el cambio entre el valor máximo y mínimo durante el día. Aunque, usualmente se usan saltos diarios por facilidad de acceso a información de precios de cierre diario.

2. Otra cuestión relevante es la ventana en la que medimos la volatilidad. La ventana es el número de datos a usar. Si el intervalo temporal es un día, la ventana sería el número de días sobre el que vamos a calcular esta volatilidad. Esto es relevante porque la volatilidad es cambiante, y cuanto más datos añadas, hay más posibilidad de incluir períodos con volatilidades distintas. Pero, si tomas pocos datos, la estimación será peor. Realmente la decisión es arbitraria y depende de para qué se vaya a utilizar esa medición de volatilidad.

3. Finalmente, para comparar volatilidades lo adecuado es tomar la misma base temporal. Si observamos saltos diarios, las volatilidades suelen estar entre el 0,5 % y el 4 %, mientras que para saltos anuales oscilan entre el 15 % y el 25 %. El estándar es hablar en términos anuales. La regla de paso de volatilidad diaria a anual es $\sigma_{anual} = \sigma_{diaria} \cdot \sqrt{252}$. 252 es el número de días que abre la bolsa en un año.

En la industria financiera, esta medida de volatilidad basada en la desviación típica se conoce como **volatilidad histórica o realizada**. Existe, sin embargo, otro concepto relevante: la **volatilidad implícita**. Es la volatilidad que debería tener el subyacente de un derivado para que el modelo de Black-Scholes reprodujera el precio observado en mercado. Esta última no será tratada en el presente trabajo; la mencionamos únicamente para evitar confusiones, dado que es ampliamente utilizada tanto en la industria como en la literatura académica.

La volatilidad presenta varias propiedades que se han observado empíricamente:

1. La volatilidad cambia en el tiempo. Por ejemplo, en situaciones normales de mercado hay volatilidades en torno al 15% o 20%. Mientras que en situaciones de estrés puede superar el 40%.

2. Los momentos de alta volatilidad están concentrados en **clústeres**. Es decir, los momentos de volatilidad alta suelen concentrarse en cortos momentos temporales, lo que indica dependencia temporal.

3. La volatilidad suele revertir a la media. Esto es, si hay momentos de alta o baja volatilidad, vuelve a estabilizarse en los valores promedios históricos.

Estas propiedades serán ilustradas empíricamente en el análisis exploratorio inicial que realizaremos sobre datos financieros reales.

Entre los usos de la volatilidad está la valoración de opciones, la medición de riesgos— especialmente riesgo de mercado— y la gestión de carteras.

Desde un punto de vista estadístico, nuestra predicción de volatilidad es una predicción temporal y univariante. El único predictor son los valores anteriores temporales de la misma variable. Esto, en ningún caso, va a predecir con anterioridad al momento del shock de volatilidad que inicia un clúster de alta volatilidad. Estos shocks suelen estar producidos por eventos sociales, políticos o económicos cuya detección mediante los valores de la volatilidad realizada previamente no es posible. Por tanto, la efectividad de los modelos que estudiaremos radica, principalmente, en cómo son de buenos captando los movimientos de la volatilidad en momentos(clusters) de alta volatilidad.

3.2 Análisis de Series Temporales

La comprensión y síntesis de la teoría sobre análisis de series temporales es un proceso largo y arduo, que requiere sólidas bases en estadística y, para una comprensión profunda, en cálculo y álgebra lineal. Asimismo, demanda una reflexión constante y prolongada en el tiempo con el fin de mantener siempre presente el objetivo práctico de su desarrollo y evitar caer en formulaciones matemáticas desvinculadas de la realidad empírica.

Por tanto, una exposición completa sobre esta teoría, incluso prescindiendo de divagaciones y ejemplos, requeriría una larga extensión que no podemos permitirnos en este trabajo.

En este apartado, se presenta una síntesis original de los resultados y justificaciones fundamentales de dicha teoría, basándome principalmente en los libros [Pe8], [Pe0] y [Pe2].

3.2.1 El objetivo

Queremos inferir un modelo que describa los datos en función del tiempo y que, de este modo, nos permita predecir su evolución futura. Evidentemente, un buen ajuste a los datos del pasado no garantiza una predicción certera en el futuro. No obstante, identificar la estructura de autocorrelaciones que mejor se ajuste(siempre que no sea un sobreajuste artificial) a los datos es un gran primer paso para predecir. Parte de la metodología consiste en elegir correctamente el modelo y, posteriormente, contrastarlo.

Además, en econometría, modelar es una buena forma de medir el efecto de una política, estudiando el cambio en la estructura temporal que esta provoca. Aunque no abordaremos esta posibilidad en el presente trabajo, nos centraremos en modelado y predicción.

Las herramientas más relevantes del análisis de series temporales se podrían resumir en:

1. **Modelos de descripción univariantes** para describir y predecir comportamientos futuros de una variable. Entre estos veremos los modelos **Autorregresivos**, los de **Media Móvil**, el que los aglutina, que es el modelo **ARIMA** y, para el caso de la volatilidad, el modelo **GARCH**.
2. **Métodos para encontrar la relación de dependencia dinámica** entre una serie de interés y diversas variables explicativas. Los modelos univariantes se pueden mejorar si encontramos variables explicativas. Aquí se encuentra la **regresión dinámica** y la **cointegración**.
3. **Modelos para representaciones conjuntas multivariantes** para obtener predicciones simultáneas. El más importante es el modelo VAR, **Vectorial Autorregresivo**.

En este trabajo nos centraremos en el primer bloque, ya que en él se fundamenta la teoría de series temporales.

3.2.2 Lo intuitivo

Al estudiar la forma en la que se estructura la sucesión de valores en el tiempo, surgen ciertas ideas que nos pueden resultar intuitivas. Estos conceptos intuitivos son los que vamos a tratar en esta sección, junto con una breve referencia a las teorías que se han desarrollado en torno a ellos.

También llamamos a esta sección lo intuitivo, en contraposición con las siguientes. Estas últimas, aunque serán básicas para el desarrollo de los modelos más generales que abarcan lo que aquí vemos como intuitivo, no resultan fáciles de relacionar a primera vista con los conceptos que presentamos en esta sección.

Modelos de tendencias deterministas

En una serie temporal es intuitivo querer modelar un crecimiento o un decrecimiento. Ya sean lineales, exponenciales o logarítmicos. De hecho, en las distintas ciencias sociales son comunes los gráficos donde se ven tendencias crecientes que muestran cambios a lo largo del tiempo.

Una forma de modelar estos cambios es con modelos del estilo:

$$z_t = \mu_t + a_t \tag{1}$$

Donde μ_t representa el cambio en el tiempo de forma determinista, es decir, no aleatorizada. Y el otro componente, a_t, recoge el efecto aleatorio del resto de componentes no modelados, llamado innovación. A las innovaciones se les suele suponer una estructura en su modelado: media cero, varianza constante, independencia y, en ocasiones, distribución normal. Esto lo veremos más adelante.

La tendencia determinista se conoce como el nivel de la serie y es una función conocida del tiempo y de unos parámetros, $\mu_t = f(t, \theta)$.

El proceso habitual consiste en suponer una forma funcional para μ_t y estimar los parámetros θ a partir de los datos observados.

El modelo más sencillo sería suponer que una serie es estable, es decir, que los valores siempre están en torno a la media. Esta serie podría representar un valor estancado y seguiría la siguiente estructura:

$$z_t = \mu + a_t \tag{2}$$

Otro ejemplo es suponer un nivel de la serie lineal:

$$z_t = \beta_0 + \beta_1 t + a_t \tag{3}$$

Estos modelos tienen dos grandes limitaciones:

- 1) Suponen que la función de tendencia es determinista en el tiempo.
- 2) Suponen que esta función es siempre la misma.

La segunda limitación puede abordarse mediante una representación por tramos en la que se ajusta una función distinta del nivel de la serie en distintos intervalos temporales. Sin embargo, este enfoque introduce un nuevo problema: al ajustar por tramos, se otorga igual importancia a todos los valores de la serie, independientemente de su cercanía temporal. Es decir, un valor presente dependería del pasado remoto con la misma ponderación que de los valores inmediatamente anteriores, lo cual resulta contraintuitivo y poco realista.

Un modelo en el que el valor de la serie en un momento dado depende de manera decreciente de los valores pasados, asignando mayor peso a los más recientes, es el **modelo de alisado exponencial**, que podemos ver en la siguiente ecuación:

$$z_{t+1} = (1-\theta)(z_t + \theta z_{t-1} + \theta^2 z_{t-2}...) \; con \; 0 < \theta < 1. \tag{4}$$

De esta forma, tendríamos que el próximo valor de la serie es la suma de los anteriores ponderados de forma que su influencia decrece cuanto más lejos estén. El $(1-\theta)$ es un término que hace que los valores no divergan, i.e. que no se hagan arbitrariamente grandes conforme pasa el tiempo.

Otra tendencia que nos gustaría poder modelar es la estacionalidad. Es decir, la parte del valor que corresponde con el momento de la semana, mes o año en que estemos. Por ejemplo, si quisiéramos modelar el número de viajeros a Grecia deberíamos tener en cuenta que el número de viajeros sube en verano y baja en invierno. Esto se podría modelar de la siguiente forma:

$$z_t = \mu_t + S_t + a_t \tag{5}$$

donde S_t representa la componente estacional, que puede modelarse como una función determinista sinusoidal que nos sirve para incluir subidas y bajqdas periódicas en la serie.

Si restamos S_t a los datos, obtenemos la serie desestacionalizada. Lo que nos ayuda a observar cómo avanza una serie en el tiempo de forma estructural. Por ejemplo, el desempleo es muy estacional y una subida en agosto no significa que haya una tendencia de subida generalizada.

3.2.3 La herramienta: el cálculo estocástico

El cálculo estocástico es una rama de la probabilidad que estudia colecciones de variables aleatorias (VVAA) relacionadas entre sí, lo cual resulta útil para el estudio temporal de fenómenos con componentes aleatorios.

Definición 3.5 *Un proceso estocástico es una sucesión de variables aleatorias* $z_0, z_1, z_2....$ *De forma más general,* z_t *con* $t \in [0, T]$.

Así pues, para modelar un proceso que cambia en el tiempo con una componente aleatoria, se puede utilizar el cálculo estocástico. Por tanto, nuestra herramienta para modelar las series temporales será el cálculo estocástico.s

3.2.4 La hipótesis principal: procesos estacionarios

En muchas ocasiones no es posible saber cuál es la distribución de probabilidad de un suceso. Por tanto, para poder estimar características de esta distribución es necesario hacer suposiciones sobre ella a lo largo del tiempo. Lo más sencillo es suponer que todos los valores de la serie siguen la misma distribución, independientemente de su momento en el tiempo. De ser así, los valores de la serie estarán alrededor de un valor constante a lo largo del tiempo, la media de la distribución que siguen todos los valores de la serie. La clave en la definición de los modelos se centrará en expresar cómo es la distribución condicionada a los valores en momentos anteriores del tiempo. De manera formal, para todo $t \in [0, T]$ z_t sigue una distribución X, que desconocemos y que no depende del tiempo t. Lo que modelaremos será la distribución condicionada a los valores previos, matemáticamente $z_t|z_{t-1}, z_{t-2}, ..., z_0$, i.e. conociendo dichos valores previos.

Hay dos definiciones de estacionariedad. Veremos la más débil, puesto que es la más útil.

Definición 3.6 *Un proceso estocástico se dice que es débilmente estacionario si, para todo* t*:*

- *1)* $\mu_t = \mu = cte.$
- *2)* $\sigma_t^2 = \sigma^2 = cte.$
- *3)* $\gamma(t, t-k) = E[(z_t - \mu)(z_{t-k} - \mu)] = \gamma_k$ *para* $k = 0, \pm 1, \pm 2, ...$

Las dos primeras propiedades exigen estabilidad en la media y la varianza del proceso. Mientras que la tercera pide que la covarianza entre variables en dos momentos del tiempo solo dependa de cuánto estén separados temporalmente. Veremos esto en profundidad un poco más adelante.

Este enfoque estacionario cobrará total sentido cuando lleguemos a los procesos integrados. Al igual que en los modelos de regresión lineal, las suposiciones de linealidad o normalidad entre las variables son muy fuertes, pero tienen grandes ventajas: el modelo se desarrolla con simplicidad y muchos casos que no cumplen con dichas hipótesis pueden ser adaptados fácilmente para que lo hagan. Muchas series que nos interesaran estudiar no

cumplen la hipótesis de estacionariedad, pero serán fácilmente adaptables con el desarrollo que veremos más adelante.

3.2.5 La descomposición de Wold

Para el caso de un proceso sin componente determinista, esto es, que no se pueda separar como una suma de una función determinista y no constante en el tiempo —por ejemplo, tendencia lineal o estacionalidad— y una variable aleatoria, podemos aplicar el **teorema de descomposición de Wold**. Este teorma aplicado a nuestro caso, nos dice lo siguiente:

Teorema 3.1 *Todo proceso estocástico débilmente estacionario, z_t, de media finita y que no contenga componentes deterministas puede escribirse como:*

$$z_t = \mu + a_t + \psi_1 a_{t-1} + \psi_2 a_{t-2} + ...$$

donde $E[a_t] = 0$, $Var(a_t) = \sigma^2$ *y* $E[a_t a_{t-k}] = 0$ *para todo* $k > 1$.

Esto nos será útil, porque ahora sabemos que cualquier proceso estacionario puede escribirse como suma infinita de innovaciones o ruido. Esto nos da ideas de formas en las que podemos intentar aproximar cualquier serie temporal estacionaria.

3.2.6 El operador de retardo

Introducimos la notación del operador de retardo, puesto que nos será útil en las próximas secciones. El operador de retardo, B, es un operador lineal aplicado a una función temporal que devuelve la misma función retardada en un período:

$$Bf(t) \equiv f(t-1)$$

Aplicado a series temporales:

$$Bz_t \equiv z_{t-1}$$

Si lo componemos linealmente, como haremos más adelante, obtenemos otro operador definido como $\phi_p(B) = 1 - xB$ donde $x \in \mathbb{R}$ es un escalar. Este operador se aplica a un proceso estocástico z_t de forma que, primero, se aplica el primer sumando y, luego, se resta el segundo sumando. De esta forma el 1, visto como operador es la identidad. Es decir, aplicar 1 a z_t te da como resultado z_t. A esto, se le restaría el otro sumando. En

el segundo sumando, x es un escalar que multiplica al resultado de aplicar B, el operador antes definido. De esta forma, $\phi_p(B) = 1 - xB$ es un operador que también se aplica a procesos estocásticos y devuelve otro proceso estocástico. Por ejemplo, aplicado a z_t nos da: $\phi_p(B)z_t = z_t - x \cdot z_{t-1}$

La ecuación característica del operador se define como $\phi_p(B) = 0$, donde B sería la incógnita a resolver para encontrar las **raíces** del operador. Esto nos será útil más adelante.

3.2.7 Un enfoque: las autocorrelaciones

La correlación entre dos variables aleatorias nos dice cómo de relacionadas están. Mide la proporción de variabilidad de las VVAA que sucede conjuntamente. Esto es:

$$\rho_{X,Y} = \frac{\sigma_{X,Y}}{\sqrt{Var(X)Var(Y)}} \tag{6}$$

, donde $\sigma_{X,Y} = Cov(X, Y)$, esto es, la covarianza entre ambas variables.

La autocorrelación es una medida usada en estadística para medir la correlación entre dos observaciones de una misma población, de ahí el prefijo *auto*. La definición para un proceso estocástico estacionario X_t con media constante μ y varianza σ^2 de autocorrelación de orden k es:

$$\rho_k = \frac{E[(X_i - \mu)(X_{i-k} - \mu)]}{\sigma^2} \tag{7}$$

Donde, $E[(X_i - \mu)(X_{i-k} - \mu)]$ es la covarianza entre X_i y X_{i-k}, o sea, γ_k que, para procesos estacionarios, hemos supuesto constante. Los ρ_k de un proceso estacionario se conocen también como función de autocorrelación simple, abreviado f.a.s. El orden k es el número de observaciones que están distanciadas en el tiempo. Por otro lado, existe también la función de autocorrelación parcial (f.a.p). Esta función es igual que la f.a.s pero elimina los efectos de la autocorrelación que van duran más de un período, de forma que se aisla mejor la influencia de cada período previo. Vamos a obviar su definición formal debido a su complejidad.

En un principio, el tratamiento de las autocorrelaciones se hace para estudiar si las observaciones de una muestra de una población no son independientes, ya que la independencia en las observaciones es una hipótesis básica en el modelado estadístico: tanto en la inferencia estadística como en regresión.

Precisamente, el problema de la autocorrelación surge cuando unas observaciones tienen capacidad predictiva sobre otras, y suele aparecer cuando los datos se recogen en distintos momentos del tiempo. Por ejemplo, en la Figura 3 se muestra una serie

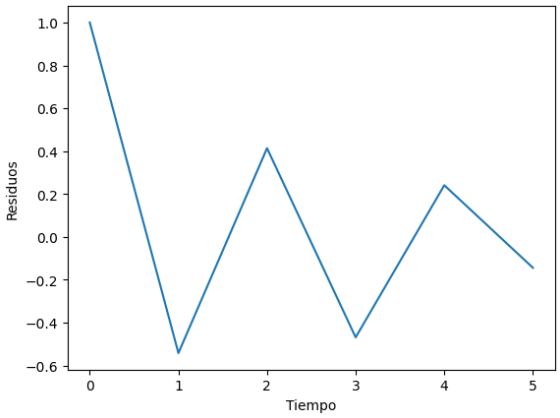


Figura 3: Ejemplo autocorrelación negativa de los residuos de un modelo. Fuente: elaboración propia.

temporal con autocorrelación negativa. En este caso, el valor de la serie está influido negativamente por el valor anterior, dando lugar a una secuencia en la que los signos de los valores se alternan en cada período, al tiempo que su magnitud disminuye progresivamente. Si la autocorrelación fuera positiva, la magnitud también se reduciría, pero los signos se mantendrían constantes.

Por tanto, las autocorrelaciones nos servirán para:

1. Crear distintos modelos con diversas estructuras en sus autocorrelaciones. Lo que llamaremos funciones de autocorrelación.
2. Identificar que modelo de series temporales se ajusta mejor a unos datos concretos mediante las autocorrelaciones observadas en estos.

3.2.8 Los procesos autorregresivos. AR(p)

Generalizan la idea de regresión para dos variables aleatorias. Pero en este caso, la regresión es de la variable consigo misma en instantes de tiempo anteriores. La p es el número de retardos (o sea, momentos anteriores en el tiempo) que tenemos en consideración para hacer la regresión.

AR(1)

Estudiamos primero la forma más simple de un proceso autorregresivo para ver de forma sencilla sus propiedades fundamentales. Luego generalizaremos este modelo:

$z_t = c + \phi z_{t-1} + a_t$ con c y ϕ constantes a determinar y las innovaciones, a_t, i.i.d. $N(0, \sigma^2)$ con σ^2 una constante. Escribiremos el primer valor de la serie $z_0 = h$.

Si extendemos la fórmula recursivamente para que dependa de momentos anteriores, nos sale:

$$z_t = c\sum_{i=0}^{t-1}\phi^i + \phi^t h + \sum_{i=0}^{t-1}\phi^i a_{t-i}$$

y al tomar esperanzas en ambos lados:

$$E[z_t] = \mu = c\sum_{i=0}^{t-1}\phi^i + \phi^t h.$$

Imponer media constante nos lleva a deducir que $|\phi|$ debe ser menor que 1 y $c = \frac{\mu}{1-\phi}$. Esto se debe a que si $|\phi| > 1$ la serie crecerá indefinidamente, haciendo imposible la media finita. Por otro lado, el límite de esta serie es $1-\phi$, por tanto, en el límite y despejando la ecuación $c = \frac{\mu}{1-\phi}$. Como esto se cumple para cualquier t, podemos asumir que $c = \frac{\mu}{1-\phi}$.

Una forma equivalente en notación de operador de retardo sería:

$$(1-\phi B)\tilde{z}_t = a_t$$

Y, también se puede escribir como suma de innovaciones, por ejemplo, pasando al otro lado de la ecuación el operador, invirtiéndolo queda así:

$$\tilde{z}_t = \sum_{i=0}^{\infty}\phi^i a_{t-i} = (1+\phi B+\phi^2 B^2+...)a_t$$

que, como se verá, es una forma particular de MA(∞), y una forma particular de la descomposición de Wold.

Su función de autocorrelación simple (f.a.s), $_k$ se consigue calculando covarianzas; nos saltamos el cálculo y queda como sigue:

$$\rho_k = \phi^k$$

La forma de identificar un proceso como autorregresivo será cuando sus autocorrelaciones sigan algo parecido a su f.a.s, es decir, decaimiento exponencial, algo parecido a la Figura 4

AR(p)

Este modelo es el mismo pero con un coeficiente distinto por cada retardo, introduciendo en él dependencia de hasta p retardos. En notación de operadores quedaría así:

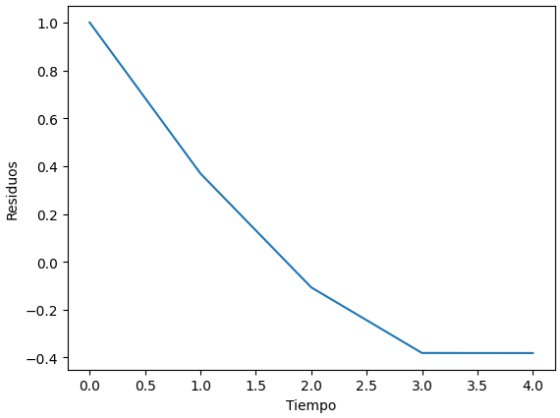


Figura 4: Residuos de un AR(1) perfecto con $\rho = 0.7$. Fuente: elaboración propia.

$$(1 - \phi_1 B - \phi_2 B^2 - ... - \phi_p B^p)\tilde{z}_t = \phi_p(B)\tilde{z}_t = a_t$$

Resolviendo el polinomio característico del operador $\phi_p(B)$ nos saldrían p raíces, no necesariamente distintas, que llamaremos G_i^{-1} con $i = 1, ..., p$. Las definimos así, puesto que es la inversa de estas raíces, G_i, el valor que nos interesa. No hay problema con suponer que tienen inversa dado que el 0 no puede ser solución de dicho polinomio, ya que $\phi_p(0) = 1$.

La condición de estacionariedad ahora es: $|G_i| < 1 \ \forall i = 1, ..., p$

Y la f.a.s sería:

$$\rho_k = \sum_{i=0}^{p} A_i G_i^k$$

donde los A_i son constantes a determinar según las condiciones iniciales de una ecuación en diferencias. Esta función decrecerá de forma exponencial y sinusoidal, ya que aquellas raíces del polinomio que sean complejas darán lugar a funciones sinusoidales.

Invirtiendo el polinomio volveríamos a tener la serie como suma de innovaciones. Faltaría estimar los valores de los parámetros y comprobar la validez del modelo, pero esto lo veremos con el modelo más general ARIMA.

3.2.9 Los procesos de media móvil. MA(q)

Los procesos AR son procesos con muchos coeficientes de autocorrelación distintos de cero que decrecen con el retardo. Captan bien algunos comportamientos de memoria larga que se desvanecen en el tiempo, pero no pueden representar series de memoria corta. Por ejemplo, cambios repentinos que solo tienen influencia en períodos cortos de

tiempo. Los procesos moving average (MA), o media móvil en español, sí captan estos comportamientos.Lo comprobaremos cuando calculemos su f.a.s.

MA(1)

Empezamos por el más simple, el de solo un retardo. Tiene la forma:

$$\tilde{z}_t = a_t - \theta a_{t-1} = (1 - \theta B)a_t = \theta_1(B)a_t \quad (*)$$

Donde a_t son i.i.d $N(0, \sigma^2)$, con σ constante.

Diremos que el proceso es invertible si $|\theta| < 1$. En caso contrario, las innovaciones tendrán más poder explicativo cuanto más atrás en el tiempo estén.

En este caso, vamos a calcular los coeficientes de autocorrelación para comprobar la propiedad que dijimos en la introducción. Si elevamos (*) al cuadrado y tomamos esperanzas a ambos lados:

$$Var(\tilde{z}_t) = E[\tilde{z}_t^2] = E[a_t^2] + \theta^2 E[a_{t-1}^2] - 2\theta E[a_t a_{t-1}]$$

Donde $E[a_t a_{t-1}] = 0$ porque los a_t son independientes y $E[a_t^2] = E[a_{t-1}^2] = Var(a_t) = \sigma^2$. Por tanto, $\gamma_0 = Var(\tilde{z}_t) = \sigma^2(1 + \theta^2)$.

Si multiplicamos (*) por $\tilde{z}_{t-1}$ y tomamos esperanzas a ambos lados:

$$E[\tilde{z}_t \tilde{z}_{t-1}] = E[a_t \tilde{z}_{t-1}] - \theta E[a_{t-1} \tilde{z}_{t-1}]$$

El término $E[a_t \tilde{z}_{t-1}]$ es 0, ya que $\tilde{z}_{t-1}$ no depende de innovaciones futuras y por tanto son independientes. Por otro lado, $E[a_{t-1}\tilde{z}_{t-1}] = E[a_{t-1}(a_{t-1} - \theta a_{t-2})] = Var(a_{t-1}) - \theta E(a_{t-1}a_{t-2}) = Var(a_{t-1)} = \sigma^2$. Ya que a_{t-1} y a_{t-2} son independientes. Por tanto, $\gamma_1 = -\theta\sigma^2$.

Mientras que, para $k > 1$:

$$\gamma_k = E[\tilde{z}_t \tilde{z}_{t-k}] = E[a_t \tilde{z}_{t-k}] - \theta E[a_{t-1} \tilde{z}_{t-k}]$$

donde ambos sumandos son cero por la independencia ya explicada y, entonces, $\gamma_k = 0 \ \forall k > 1$.

Finalmente, $\rho_1 = \frac{\gamma_1}{\gamma_0} = \frac{-\theta}{1+\theta^2}$, y $\rho_k = \frac{\gamma_k}{\gamma_0} = 0 \ \forall k > 1$.

Efectivamente, las autocorrelaciones son solo entre un valor y el inmediatamente anterior.

Si invertimos el operador, se nos queda la siguiente ecuación:

$$\theta_1(B)^{-1}\tilde{z}_t = a_t$$

que es un AR(∞).

MA(q)

En este caso es lo mismo pero dependiendo de las q anteriores innovaciones:

$$\tilde{z}_t = a_t - \theta_1 a_t - ... - \theta_q a_{t-q} = (1 - \theta_1 B - ... - \theta_q B^q)a_t = \theta_q(B)a_t$$

Será invertible si el polinomio $\theta_q(B)$ tiene raíces fuera del círculo unidad.

Y su f.a.s. tendrá la forma:

$$\rho_k = \begin{cases} \rho_k = \frac{\sum_{i=0}^{i=q} \theta_i \theta_{k+i}}{\sum_{i=0}^{i=q} \theta_i^2} & k = 1, ..., q \\ \rho_k = 0 & k > q \end{cases}$$

Donde, de nuevo, solo tienen autocorrelaciones aquellos q términos a los que hemos dado estructura expresamente.

Si invertimos el operador se nos queda de nuevo, un $AR(\infty)$:

$$\theta_q(B)^{-1}\tilde{z}_t = a_t$$

3.2.10 La dualidad AR-MA y los procesos ARMA

Si invertimos el operador de un AR(p) nos da un MA(∞) y si invertimos el operador de un MA(q) nos da un AR(∞). Por tanto, ambos modelos nos permiten representar de forma finita autocorrelaciones que con el otro no se podrían. A su vez, ambos son casos particulares de la descomposición de Wold. Por un lado, AR nos da infinitos coeficientes de la descomposición pero que tienen estructura exponencial. Y, por otro lado, MA nos permite dar la estructura que queramos a los coeficientes pero solo de forma finita. Combinándolos, tendremos los procesos ARMA. Estos son muy versátiles y servirán para representar muchos tipos de procesos estacionarios. Un ARMA(p,q) tiene la forma:

$$(1 - \phi_1 B - \phi_2 B^2 - .... - \phi_p B^p)\tilde{z}_t = (1 - \theta_1 B - ... - \theta_q B^q)a_t$$

Será estacionario si el polinomio $\phi_p(B)$ tiene raíces fuera del círculo unidad. Y será invertible si el otro polinomio $\theta_p(B)$ tiene sus raíces fuera del círculo unidad.

Su f.a.s tendrá la siguiente estructura:

- Tendrá q-p+1 coeficientes iniciales con una estructura determinada por los parámetros de AR y MA.
- Decrecerá a partir del q-p como una mezcla de exponenciales y sinusoidales determinada solo por la parte AR.

3.2.11 La gran genialidad: los procesos integrados y el modelo ARIMA

Ahora, volvemos la vista atrás. Después de estos modelos tan importantes surgen dos cuestiones claves. Primero, cómo modelamos las tendencias y demás conceptos que vimos en la parte intuitiva. Y segundo, ¿qué hacemos para modelar la serie si no es estacionaria?.

Pues aquí es donde viene la gran genialidad. El desarrollo que hemos hecho nos permite introducir los casos donde no hay tendencias constantes (y, por tanto, tampoco estacionariedad) como casos que al diferenciarlos resultan en procesos estacionarios. Estos casos incluyen tendencias lineales, polinomiales y muchos otros. De esta forma, muchos procesos no estacionarios pueden ser convertidos en estacionarios mediante la diferenciación:

$$\nabla z_t = z_t - z_{t-1}, \text{ donde } \nabla = 1 - B.$$

Este operador,∇, se puede aplicar repetidas veces hasta obtener un proceso estacionario.

Por tanto, el modelo ARIMA(p,d,q) quedaría de la siguiente forma:

$$(1 - \phi_1 B - \phi_2 B^2 - .... - \phi_p B^p)(1 - B)^d \tilde{z}_t = (1 - \theta_1 B - ... - \theta_q B^q) a_t$$

De forma más compacta:

$$\phi_p(B)\nabla^d \tilde{z}_t = \theta_q(B) a_t$$

Sobre cómo modelar la estacionalidad no vamos a entrar en profundidad. Pero básicamente es una variante de ARIMA llamada SARIMA$(P, D, Q)_s$x(p,d,q)que tiene la siguiente forma:

$$\Phi_P(B^s)\phi_p(B)\nabla_s^D \nabla^d \tilde{z}_t = \Theta_Q(B^s)\theta_q(B) a_t$$

donde s es el período estacional y $\nabla_s = (1 - B^s)$ es la diferenciación estacional.

3.2.12 Predicción, elección de modelo, estimación y diagnosis

En este apartado tratatremos brevemente cuestiones metodológicas sobre el uso práctico de estos modelos.

Predicción

Bajo las hipótesiss usuales del modelo se puede demostrar que el predictor óptimo (definido como el que minimiza los errores cuadráticos) es la esperanza condicionada respecto al momento del que poseemos información:

$$\hat{z}(k) = \mathbb{E}[Z_{T+k}|Z_T] \tag{8}$$

Sobre este predictor podemos decir que las predicciones a corto plazo las determinan tanto los coeficientes AR y MA mientras que a largo plazo lo hacen las diferenciaciones y la media.

La identificación, selección, estimación y diagnosis en modelos ARIMA es tratada en el anexo A.

3.2.13 Modelos de heterocedasticidad condicional

Los modelos que vamos a presentar en esta sección surgieron para modelar series temporales con varianza de predicción no constante, recordemos que la varianza constante es una de las hipótesis de estacionariedad débil. Como escribió Engle en el paper original([Eng82]): " el pronóstico del valor de hoy basado en la información pasada, bajo suposiciones estándar, es simplemente $E(y_t|y_{t-1})$, que depende del valor de la variable Y_{t-1}. La varianza de este pronóstico a un período está dada por $Var(y_t|y_{t-1})$. Tal expresión reconoce que la varianza condicional del pronóstico depende de la información pasada y, por lo tanto, puede ser una variable aleatoria. Sin embargo, para los modelos econométricos convencionales, la varianza condicional no depende de Y_{t-1}".

Este modelo serviría para modelar propiedades que presentan con frecuencia las series financieras. Las rentabilidades tienen poca estructura en la media y parecen seguir paseos aleatorios, pero no tienen estructura de normalidad en los residuos, ya que presentan alta curtosis y, además, aunque estén incorreladas tienen otras estructuras de dependencia, com por ejemplo, entre sus cuadrados. Además, su varianza no es constante, ya que se presenta en forma de clusters. Una alta varianza en un momento da lugar a más desviaciones de la media en momentos siguientes. Profundizaremos más sobre estas propiedades y presentaremos resultados empíricos en la sección sobre Rentabilidad y volatilidad.

Estos modelos suponen varianza constante para tener procesos estacionarios (lo cual

encaja con la evidencia empírica de media constante cercana a cero en los retornos), pero su varianza condicionada no será constante, esto nos permitirá modelar propiedades que alejan los residuos de la normalidad.

ARCH(1)

Vamos a ver un detalle de lo que hemos hablado. Suponiendo un proceso de ruido blanco, e_t, si estas son normales, la incorrelación implica independencia. Pero si no son normales podemos tener estructuras de dependencia no lineales y sin correlación.

Los modelos ARCH tienen la forma $e_t = \sigma_t \cdot \epsilon_t$, donde ϵ_t es un proceso de ruido blanco normal estandarizado. O, lo que es lo mismo, $\epsilon_t \sim N(0,1)$ i.i.d. Ahora, lo que vamos a modelar es σ_t , esto es un proceso estacionario, con media 0 y varianza constante σ^2, pero con estructura dinámica. Este proceso es independiente de ϵ_t y modelará la varianza condicional del proceso e_t en el momento t. Ya que $Var(e_t) = Var(\epsilon_t)Var(\sigma_t) = \sigma^2$, pero la varianza condicionada $Var(e_t|e_{t-1}) = Var(\epsilon_t|e_{t-1})Var(\sigma_t|e_{t-1}) = Var(e_t) \cdot E[\sigma_t^2|e_{t-1}] = \sigma_t^2$. La hipótesis de que ϵ_t tiene media cero y varianza constante σ_t^2, junto con su independencia temporal, garantiza que el proceso e_t tenga media constante igual a cero y que sus valores estén incorrelacionados. Por tanto, e_t será estacionario.

Los modelos ARCH modelan σ_t dependiente dinámicamente de sus valores anteriores de la siguiente forma: $E[e_t|e_{t-1}] = \sigma_t^2 = \alpha_0 + \alpha_1\sigma_{t-1}^2$, con $\alpha_0 > 0$ y $\alpha_1 \geq 0$ para asegurar varianza positiva en todo momento.

La varianza marginal del proceso σ será, usando el teorema de la doble esperanza, $\sigma^2 = E(E[e_t^2|e_{t-1}]) = \alpha_0 + \alpha_1 E[e_{t-1}^2] = \alpha_0 + \alpha_1\sigma^2$, que, despejando queda $\sigma^2 = \frac{\alpha_0}{1-\alpha_1}$, por tanto $0 \leq \alpha_1 < 1$.

Se puede demostrar que la curtosis del proceso viene dada por $K = 3\frac{1-\alpha_1^2}{1-\alpha_1^3}$, por tanto, como $\alpha_1 > 0$ tendremos que $K \geq 3$, es decir colas más pesadas que la distribución normal. Para tener curtosis finitas necesitaremos $\alpha_1^2 < \frac{1}{3}$, lo cual limita el modelo.

ARCH(r)

De la misma forma se puede generalizar para que la varianza condicional dependa de varios valores pasados, r en este caso:

$$\sigma_t^2 = \alpha_0 + \alpha_1 e_{t-1}^2 + ... + \alpha_r e_{t-r}^2 \tag{9}$$

La varianza marginal será $Var(e_t) = \frac{\alpha_0}{1-\alpha_1-...-\alpha_2}$ lo que nos lleva a la restricción $\sum_{i>0}\alpha_i < 1$ para que la serie sea estacionaria. Puede comprobarse que, en este caso, su curtosis también es mayor que 3 siempre.

GARCH(r,s)

En la práctica los modelos ARCH llevan a modelos con órdenes altos ([Tsa05], página 106). En el año 1986 Bollerslev describió un nuevo método con un término adicional de media móvil en la varianza ([AB98]).

Suponemos, al igual que antes, el siguiente proceso: $e_t = \epsilon_t \sigma_t$, donde ϵ_t y σ_t son procesos estacionarios independientes entre sí. Con ϵ_t son i.i.d $N(0,1)$.

Y, suponemos, que las varianzas condicionales siguen el proceso $\sigma_t^2 = \alpha_0 + \sum_{i=1}^{r} \alpha_i e_{t-i}^2 + \sum_{j=1}^{s} \beta_j \sigma_{t-j}^2$. La restricción de varianza positiva nos lleva a que $\alpha_0 > 0$ y $\alpha_i, \beta_j \geq 0$. En este caso, un valor alto en la serie da lugar a varianzas altas, y el término e media móvil hace que estas varianzas se mantengan durante un mayor tiempo.

Considérese el caso en que se observa un valor extremadamente alto en la serie, seguido por uno bajo, simplemente como resultado de la aleatoriedad inherente al proceso (es decir, debido a una alta varianza condicional, pero no de forma determinista). En un modelo ARCH(1), dicho comportamiento abrupto implicaría el fin inmediato del período de alta volatilidad, ya que la varianza condicional en el siguiente período dependería únicamente del último valor observado.

Esto pone de manifiesto una de las principales limitaciones de los modelos ARCH de orden bajo: su incapacidad para generar clústeres persistentes de volatilidad. Para superar esta limitación sería necesario aumentar el orden del modelo, lo que complica la estimación y puede afectar la estabilidad.

Por el contrario, un modelo GARCH(1,1) introduce un componente autorregresivo en la varianza condicional a través del término β_1, lo que permite que la volatilidad decrezca gradualmente después de un choque. Esta estructura reproduce de forma más realista los clústeres de alta volatilidad observados en muchas series financieras, ya que la varianza depende tanto de valores pasados de la serie como de su propia dinámica reciente. Esto se ilustra claramente en la Figura 5.

Igual que antes se calcula la varianza marginal y sale: $Var(e_t) = \frac{1}{1-\sum_{i=1}^{p} \alpha_i - \sum_{j=1}^{q} \beta_j}$, de ahí la restricción $\sum_{i=1}^{p} \alpha_i + \sum_{j=1}^{q} \beta_j < 1$ para que el proceso sea estacionario. En muchas ocasiones esta suma es cercana a uno, lo que quiere decir que es posible que el proceso de varianza condicional esté integrado, simplemente se añadiría una restricción a uno de los coeficientes $\alpha_p = 1 - \sum_{i=1}^{p-1} \alpha_i + \sum_{j=1}^{q} \beta_j$ y tendríamos un proceso IGARCH(p,q).

Se puede demostrar que los procesos GARCH tienen siempre curtosis $K > 3$.

3.2.14 Debilidades de los modelos ARCH y GARCH

Estos modelos ARCH y GARCH sirven para modelar procesos con residuos no normales y con alta curtosis, así como dependencias temporales no lineales (o sea, dependencia sin

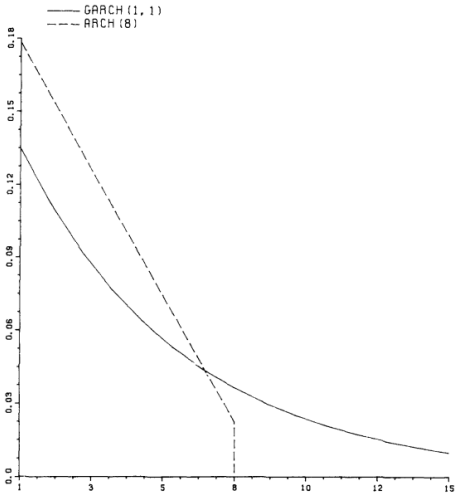


Figura 5: Decaimiento de volatilidad en ARCH y GARCH. Fuente: [Bol86]

correlación). Esto es especialmente útil en el modelado de series temporales financieras, ya que tienen dependencia en los cuadrados de los residuos, pero estos son incorrelados en la práctica. Sin embargo, estos modelos fallan en lo siguiente:

- 1) El modelo asume que los shocks negativos y positivos tienen los mismos efectos en la varianza. Ya que esta depende de los cuadrados se los shocks (y varianzas en caso de GARCH) previas. Pero en la práctica se sabe que las volatilidades responden distinto a las subidas que a las bajadas de precios.
- 2) Estos modelos son restrictivos en los parámetros como hemos podido ver. Esto limita las posibilidades de captar el exceso de curtosis de la serie.

Para superar el primer problema se han propuesto modelos como el EGARCH, un GARCH exponencial, de forma que reacciona de forma distinta a los shocks negativos y a los positivos.

3.2.15 Identificación, estimación y diagnosis

La identificación de un modelo ARCH o GARCH se suele realizar tras intentar modelar con ARMA, lo que nos dará unos residuos sin dependencia en la media pero que no cumplen las hipótesis del modelo ARMA, ya que no tienen varianza constante en las distribuciones condicionadas, ni se distribuyen como una normal. Esto también se hará notar en las autocorrelaciones de los residuos. Existen contrastes de dependencia en los cuadrados que se podrían realizar.

La estimación se puede hacer utilizando un algoritmo de optimización no lineal sobre la función de verosimilitud.

Para la diagnosis. Si llamamos e_t a los residuos del modelo ARIMA y los estandarizamos con $e_t/\hat{\sigma}_T$. Estos deberían seguir, aproximadamente, un proceso de ruido blanco. Además, los cuadrados no deberían mostrar dependencia.

Una serie de tipo ARCH puede presentar muchos atípicos si se considera de varianza constante. Es importante no confundirlos con la heterocedasticidad ni eliminar demasiados. Para ello, se propone considerar outliers aquellos valores que se desvíen en más de 7σ y construir la serie GARCH. Después, analizar los residuos estandarizados y, si se detectan atípicos, eliminarlos y repetir hasta converger.

3.3 Predicción de series temporales con ML

A continuación, pasamos a examinar los métodos de predicción que se han desarrollado desde la Ingeniería Informática que resultan útiles para estas tareas de predicción. Estos modelos se enmarcan dentro del ámbito de la **Inteligencia Artificial**, el **Aprendizaje Automático** (*Machine Learning*) y el **Aprendizaje Profundo** (*Deep Learning*). Este apartado está basado en el libro [Gé23].

Antes de empezar a aplicarlo a las series temporales, presentaremos las definiciones comunes. El campo más genérico es la Inteligencia Artificial, que se centra en desarrollar sistemas capaces de realizar tareas que, normalmente, requieren la inteligencia humana. Desde el punto de vista de la informática, esto involucra tanto la programación como los elementos externos a un ordenador que nos permiten interactuar con él, como por ejemplo las pantallas, altavoces o brazos robóticos.

Dentro de la Inteligencia Artificial se encuentra el Aprendizaje Automático. Este campo de la informática crea programas que aprenden a realizar estas tareas a partir de la experiencia (expresada en datos interpretables por el programa) sin ser programados explícitamente para realizar dichas tareas.

A su vez, dentro del Aprendizaje Automático, encontramos el Aprendizaje Profundo. En este caso, el modelo de aprendizaje se basa en una red neuronal profunda, una arquitectura compuesta por múltiples capas de nodos (o neuronas artificiales) que permiten representar y aprender patrones complejos en los datos.

3.3.1 Predicción con ML

Tal y como hemos explicado en la introducción, las tareas de predicción tanto en la estadística como en el aprendizaje supervisado presentan la siguiente metodología:

1. Elegimos datos observables en los que basaremos la predicción, en la estadística se suelen conocer como variables explicativas y, en el campo del ML, como features o características.
2. Tomamos un conjunto de datos que relacionen valores de la variable a predecir, conocida como variable objetivo, con las variables explicativas.
3. Formulamos un modelo que relacionará los valores de las variables explicativas con la variable objetivo.
4. El modelo elegido es la forma funcional de la relación, así que debemos adaptar el modelo a los datos que tenemos. En estadística se suele conocer como estimación de parámetros y, en ML, entrenamiento del modelo.
5. Evaluamos el modelo en el conjunto de datos. En estadística suele ser un test de especificación mientras que en ML se conoce como validación.

Es importante remarcar una diferencia adicional. En ML, usualmente, los modelos tienen parámetros que no se estiman durante el entrenamiento, sino que son elegidos previamente, y se conocen como **hiperparámetro**. En caso de que tengamos estos hiperparámetros, la validación puede servir como paso intermedio para elegir entre modelos con distintos hiperparámetros. Tras esto, la evaluación final consistiría en realizar el **test** con el modelo elegido en la validación.

La predicción de un valor numérico se suele llamar regresión, mientras que, si la variable objetivo es una categoría, se suele llamar clasificación. Bajo este contexto se han desarrollado numerosos modelos, tanto desde el enfoque estadístico, como desde el enfoque del ML. En algunos modelos, la diferencia entre ambos enfoques es sutil.

En las siguientes secciones vamos a tratar algunos modelos de ML útiles para la predicción de series temporales. En principio, cualquier modelo de Machine Learning es aplicable a series temporales. Simplemente, para cada valor x_t de la serie, sus features serían los momentos anteriores $x_{t-1}, x_{t-2}, \ldots$. De hecho, este es el enfoque de los modelos AR. Un AR(p) es una regresión lineal donde la variable objetivo para cada t es x_t y las variables explicativas son $x_{t-1}, x_{t-2}, ..., x_{t-p}$.

Bajo este pretexto vamos a basarnos en el trabajo de [K+22], donde analizan los distintos modelos de Machine Learning más usados para la predicción en el mercado de valores, para introducir y, más adelante, probar algunos de dichos modelos.

Modelos que veremos basados en redes neuronales:

- Redes neuronales recurrentes (RNN)
- Long Short-Term Memory (LSTM)

3.3.2 Redes Neuronales

Vamos a ver una breve introducción a las redes neuronales, dado que muchos de los modelos que veremos se basan en estas. Nos basaremos en [Gé23].

En aprendizaje automático, una neurona se define como un nodo dentro de una red que recibe un conjunto de entradas, las transforma mediante una combinación lineal y, posteriormente, aplica una función —conocida como función de activación— para generar una salida no lineal. Matemáticamente, el output se puede escribir como $o = f(\mathbf{w}^T\mathbf{x} + b)$, donde $\mathbf{x}$ es el vector de inputs, b es el sesgo de la neurona, $\mathbf{w}$ el vector de pesos de cada input y f la función de activación.

Estas neuronas se agrupan para formar capas, estas se agrupan de forma que cada capa recibe las entradas de la capa anterior y comunica sus salidas a la posterior. El típico esquema de una red con una capa de entrada, una capa oculta y una capa de salida se puede ver en la Figura 6. Donde cada flecha entre neuronas representa que el output de

la neurona de salida llega como input a la neurona de entrada.

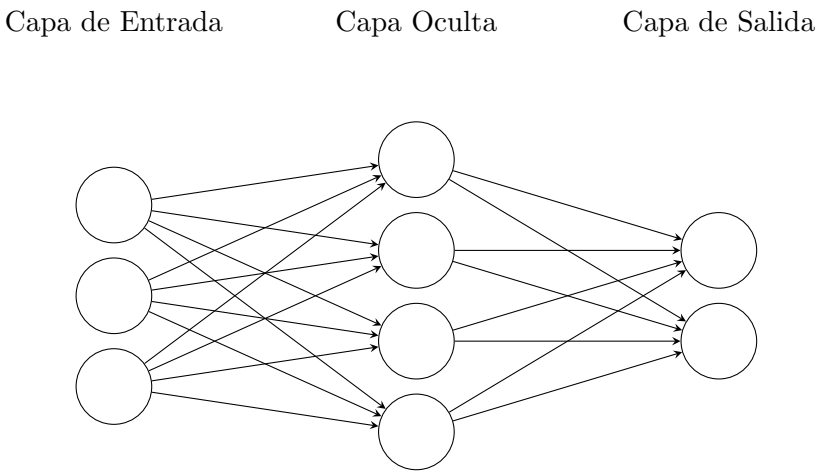


Figura 6: Red neuronal simple. Fuente: elaboración propia.

Los parámetros de una red neuronal de n capas son:

- El vector de sesgos de cada neurona $\mathbf{b_i}$ para cada capa $i = 1, .., n$.
- La matriz de pesos, $\mathbf{W_i}$ para cada capa $i = 1, .., n$, donde cada fila de la matriz son los pesos de una neurona.

De forma que la salida de cada capa es un vector que depende de la salida de la capa anterior, $O_i = f(O_{i-1}W_i + \mathbf{b_i})$ para $i = 1, ...n$. Con O_0 los valores de las features. Realmente, las funciones de activación pueden ser distintas entre capas. Por tanto, se puede expresar matemáticamente la salida en base a las features como una composición de funciones. Por ejemplo, para dos capas: $O_2 = f(f(O_0W_1 + \mathbf{b_1})W_2 + \mathbf{b_2})$. Como podemos ver, en esencia, es lo mismo que elegir una forma funcional como hace clásicamente la estadística. De hecho, una regresión lineal es un caso particular de una red neuronal con una única neurona cuya función de activación es la constante $f \equiv 1$. Con un detalle: estos modelos son mucho más flexibles y necesitan mayor cantidad de datos para que la estimación, o entrenamiento, genere un modelo que prediga bien. De ahí que se haya dado su proliferación en el contexto actual, con mayor capacidad de cómputo, y que su uso generalizado comenzara en el campo de la Informática.

Como ilustración de esta propiedad de flexibilidad, está demostrado que una red neuronal feed-forward (como la que hemos explicado) es capaz de adoptar la forma de cualquier función continua en un intervalo cerrado y acotado, incrementando el número de nodos en las capas ocultas. Véase [Hor89].

Sin embargo, esta flexibilidad es un arma de doble filo: si bien hace a las redes neuronales útiles para multitud de problemas y capaces de captar relaciones entre los datos que los modelos clásicos no pueden, también hace que sean complicadas de entrenar. Mientras que las bases teóricas para las redes neuronales fueron sentadas en los años 50 y 60 del siglo pasado, no fue hasta el año 1986 cuando se propuso un algoritmo de entrenamiento eficiente en [RHW86].

Entrenamiento de redes neuronales

Dicho algoritmo para el entrenamiento de redes neuronales es el afamado **backpropagation**. Este algoritmo consiste, como el resto de modelos estudiados, en modificar los parámetros de los nodos para minimizar una función de error.

Primero, vamos a explicar la función de error con precisión. La función de error mide cuán diferente es el valor predicho por la red, $\hat{\mathbf{y}}$, del valor real $\mathbf{y}$. Por ejemplo, el error cuadrático se calcularía como $\mathcal{L} = (\hat{\mathbf{y}} - \mathbf{y})^2$. Escribimos $\hat{\mathbf{y}} - \mathbf{y}$ por simplicidad, pero si son vectores lo que debemos hacer es calcular la distancia entre puntos basados en alguna métrica, como podría ser la distancia euclídea. Desde el punto de vista del entrenamiento, tenemos una red con L capas $i = 1, 2, ..., L$ con nodos determinados por pesos $\mathbf{W_i}$ y sesgos $\mathbf{b_i}$ con $i \in [1, .., L]$. Estos parámetros de los nodos son lo que queremos estimar. Por tanto, son las variables de la función de error $\mathcal{L}(\{\mathbf{W_i}\}_{i\in[1,..,L]}, \{\mathbf{b_i}\}_{i\in[1,..,L]})$. En resumen, dada una instancia de los datos $(\mathbf{x}, \mathbf{y})$ y una red, tenemos una predicción $\hat{\mathbf{y}}$, que con $L = 2$ resulta en $\hat{\mathbf{y}} = f(f(\mathbf{x}W_1 + \mathbf{b_1})W_2 + \mathbf{b_2})$. A modo de ejemplo, con dos capas y función de error cuadrático, para una única instancia de los datos, la función de error sería $\mathcal{L}(\{\mathbf{W_i}\}_{i\in[1,..,L]}, \{\mathbf{b_i}\}_{i\in[1,..,L]}; (\mathbf{x}, \mathbf{y})) = (f(f(\mathbf{x}W_1 + \mathbf{b_1})W_2 + \mathbf{b_2}) - \mathbf{y})^2$.

En realidad, lo que queremos minimizar es el error en el conjunto completo, no en una única instancia. Por tanto, dado un conjunto de datos $(\mathbf{x}_j, \mathbf{y}_j)$ con $j \in [1, ..., N]$, queremos minimizar el error cuadrático medio en el conjunto:

$$\frac{1}{N} \cdot \sum_{j=1}^{N} \mathcal{L}(\{\mathbf{W_i}\}_{i\in[1,..,L]}, \{\mathbf{b_i}\}_{i\in[1,..,L]}; (\mathbf{x}_j, \mathbf{y}_j))$$

Hay dos observaciones que se pueden hacer. Por un lado, tenemos muchas variables, por tanto, necesitaremos muchos datos para entrenar el modelo. Además, el hecho de tener muchas variables hace que sea complicado encontrar puntos de minimización de la función, ya que tendremos muchas dimensiones. Por otro lado, si tenemos muchos datos, necesitaremos mucho tiempo para calcular esa función de error. ¿Cómo superamos estas limitaciones? La respuesta está en un algoritmo iterativo basado en el **descenso del gradiente** de la función de error por ***batches***.

Batch es como se le conoce a un subconjunto del conjunto de entrenamiento. Formalmente, podemos escribir $\{(\mathbf{x}_j, \mathbf{y}_j)\}_{j\in J}$, donde J es un subconjunto de $[1, .., N]$. Es decir, la idea es aplicar un algoritmo iterativo que va reduciendo el error en sucesivas iteraciones,

conocidas como **epochs**, pero aplicando la red y el cálculo del error en cada iteración solo sobre un subconjunto de los datos, de forma que en cada *epoch* se emplee menos tiempo. Hay diversas formas de escoger estos batches, pero la más usada es tomarlos de forma aleatoria. Al descenso del gradiente con *batches* escogidos aleatoriamente se le conoce como **descenso del gradiente estocástico**. Veamos cómo funciona.

El **gradiente** de una función de varias variables $\mathcal{L} : \mathbb{R}^m \to \mathbb{R}$ es un operador vectorial que se define en base a las derivadas parciales de dicha función:

$$\nabla \mathcal{L}(\mathbf{x}) = \left(\frac{\partial \mathcal{L}}{\partial x_1}(\mathbf{x}), \frac{\partial \mathcal{L}}{\partial x_2}(\mathbf{x}), \ldots, \frac{\partial \mathcal{L}}{\partial x_m}(\mathbf{x}) \right)^T$$

El **gradiente** en un punto $\mathbf{x} \in \mathbb{R}^n$ tiene la propiedad de ser el vector que apunta en la dirección de **mayor crecimiento** de la función en ese punto. Y la dirección hacia donde más se reduce el valor de la función es la dirección contraria al gradiente $-\nabla \mathcal{L}(\mathbf{x})$. En la función de error $\mathcal{L}$ las variables son los pesos de las neuronas. Por tanto, el gradiente nos dice hacia dónde debemos mover estos pesos para reducir la función de error. De forma que, en una iteración i, dados unos pesos y sesgos, que, por simplificar, escribimos vectorialmente como $\mathbf{x}_i$, y un gradiente $\nabla \mathcal{L}(\mathbf{x}_i)$, los pesos y sesgos de la siguiente iteración los moveremos hacia la dirección donde se reduce el error $\mathbf{x}_{i+1} = \mathbf{x}_i - \eta \nabla \mathcal{L}(\mathbf{x}_i)$. Donde η es el **learning rate**, e indica cuánto vamos a mover los parámetros. Un mayor *learning rate* puede dar lugar a una convergencia más rápida o provocar una inestabilidad en el algoritmo que haga que no converja. Por tanto, el *learning rate* es un hiperparámetro de las redes neuronales.

Sin embargo, esta idea existía décadas antes del año 1986, como se puede ver en [Ama67], pero tenía un problema a la hora de ser aplicado. El problema era la dificultad para calcular el gradiente en redes neuronales con varias capas. Ya que, como hemos visto, cada capa aplica la función de activación a las salidas de las anteriores. Lo que hizo especial al *backpropagation* fue la forma eficiente de calcular el error correspondiente a cada parámetro de cada nodo y de aplicar el ajuste correspondiente por descenso del gradiente.

El algoritmo funciona de la siguiente manera (traducción de [Gé23] con ecuaciones de [Nie15]). En cada *epoch* t:

1. Realizamos la predicción de la red neuronal para cada dato guardando los valores de salida de cada nodo.

2. Medimos el error de la salida con la función de error escogida $\mathcal{L}$ y calculamos la derivada parcial de cada peso para todos los nodos de la última capa aplicando la regla de la cadena: $\frac{\partial \mathcal{L}}{\partial w_{ij}^L} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_j^L} \cdot \frac{\partial z_j^L}{\partial w_{ij}^L}$. Donde w_{ij} es el peso que se le aplica a la salida de la neurona i de la capa anterior $L-1$ en la neurona j de la última capa L y z_j^L es el valor de salida de la neurona $z_j^L = f(\sum_{i=1}^n w_{ij}^L \cdot a_i^{L-1} + b_j^l)$, con a_i^{L-1} la salida de la neurona i de la capa anterior. De forma que $w_{ij}^L(t+1) = w_{ij}^L(t) - \eta \cdot \frac{\partial \mathcal{L}}{\partial w_{ij\,L}}$.

Para los sesgos el gradiente quedaría: $\frac{\partial \mathcal{L}}{\partial b_j^L} = \frac{\partial \mathcal{L}}{\partial \hat{y}_j} \cdot \frac{\partial \hat{y}_j}{\partial z_j^L} \cdot \frac{\partial z_j^L}{\partial b_j^L}$. Dado que $\frac{\partial z_j}{\partial b_j^L} = 1$, $\frac{\partial \mathcal{L}}{\partial b_j^L} = \frac{\partial \mathcal{L}}{\partial \hat{y}_j} \cdot \frac{\partial \hat{y}_j}{\partial z_j^L}$.

3. Una vez actualizados los pesos en la última capa L debemos propagar los errores hacia las capas anteriores $0 < l < L$ y repetir el estos cálculos. La clave en este paso es cómo saber los errores correspondientes a la capa anterior. El cálculo usual se realiza con álgebra matricial y se calculan como $\delta^l = \left(W^{l+1}(t)\right)^\top \delta^{l+1} \odot f'(z^l)$, donde $W^{l+1}(t)$ son los pesos de la capa posterior antes de actualizarlos, δ^{l+1} los errores de la capa posterior, z^l los outputs de la capa l y $\odot$ la multiplicación elemento a elemento. De esta forma, los gradientes quedan como: $\frac{\partial \mathcal{L}}{\partial b_j^l} = \delta_j^l$ para los sesgos y $\frac{\partial \mathcal{L}}{\partial W_{jk}^l} = a^{l-1}\delta_j^l$ para los pesos.

Este algoritmo, aunque revolucionario, tuvo que enfrentarse a dos problemas. Uno de los problemas fue la dificultad que, en ocasiones, tenía para converger debido al crecimiento explosivo del gradiente en algunas neuronas y al desvanecimiento del gradiente en otras neuronas. El otro de los problemas fue la gran cantidad de tiempo que requería para el entrenamiento. Parte del primer problema fue identificado como consecuencia del uso extendido como función de activación que tenía la función sigmoide, que saturaba los resultados de las neuronas en 0 y 1, de forma que las derivadas tendían a cero rápidamente y, por tanto, también tendía a cero el gradiente. Ambos problemas fueron mejorados notablemente con el uso de la función $ReLU(z) = \max(z, 0)$. Esta función no satura por arriba y tanto la propia función, como su derivada $\text{ReLU}'(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{si } x \leq 0 \end{cases}$, son muy rápidas de calcular computacionalmente.

La función ReLU sufre de otro problema. Muchas neuronas caen en valores negativos y hacen que esa neurona produzca el valor 0 constantemente. Se han propuesto numerosas variantes como ELU, SELU, GELU, etcétera. Aunque, en general, es útil la normalización, esto es, centrar los valores de las *features* en 0 con varianza 1. De forma que no haya variables con valores mucho más grandes que otras ni valores sesgados hacia abajo provocando esta muerte de las neuronas. Para hacer aún más incisión en esto, se puede forzar la normalización de los valores que entran o salen de cada neurona en cada paso mediante la **Batch Normalization**, aunque esto ralentiza el entrenamiento.

Otra técnica que puede ser útil para evitar este problema es el uso de **dropout**. Esta técnica consiste en apagar un porcentaje de las neuronas de la red en cada *epoch*. Esto fuerza que todas las neuronas tengan una salida relevante durante algún momento del entrenamiento, evitando así que caigan en la irrelevancia en algún momento y se queden en un cero constante.

3.3.3 Redes neuronales recurrentes

Las redes neuronales recurrentes (RNN) son una variante de las redes neuronales artificiales (ANN). Las ANN, como la que hemos explicado, se conocen como *feed-forward* ya que las salidas de las neuronas de una capa van hacia la neurona siguiente y reciben los datos de las capas anteriores. Tal y como se ve en la figura 7, las neuronas reciben las entradas como en una ANN *feed-forward*. Sin embargo, la salida de cada neurona también es entrada, junto con el el siguiente dato de la serie, de las neuronas en la siguiente iteración. De esta forma, las neuronas tienen pesos para los datos de $\mathbf{x}_t$ y otros pesos para los datos de la salida anterior $\hat{\mathbf{y}}_{t-1}$.

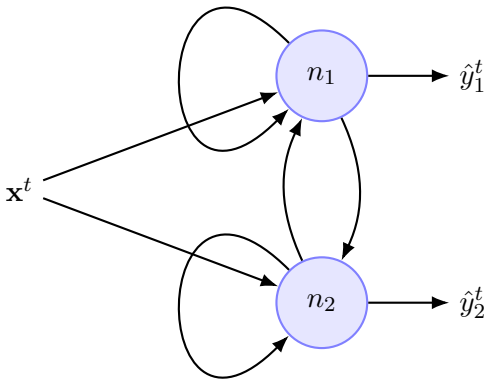


Figura 7: Red neuronal recurrente de dos neuronas. Fuente: elaboración propia.

La aplicación de estas redes es secuencial, dato tras dato. Se puede ver en la figura 8 este proceso. Vemos que en h_0 tiene como entrada el primer dato, lo que provoca una salida de predicción y_0 que, además, se comunica a las neuronas para la siguiente iteración junto con el siguiente dato de la serie. En general, la salida de predicción de la red y_i no tiene por qué ser el mismo valor que el que se le pasa a las neuronas para la siguiente iteración h_i. Pueden tener un tratamiento distinto dentro de n_i. Por tanto, se habla de **celdas de memoria** en lugar de redes neuronales o capas, dado que estas son solo una parte de la celda.

Hay distintas formas de usar e interpretar esta arquitectura:

- **Sequence-to-sequence**: La entrada x_i es el valor temporal previo al que queremos predecir, lo que nos produce y_i. Este es el uso común para predicción de series temporales.
- **Sequence-to-vector**: Las entradas son son datos secuenciales, como un texto, pero

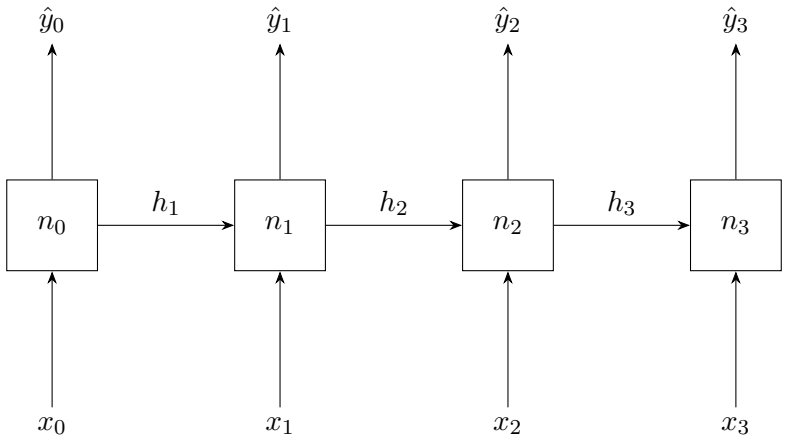


Figura 8: Desarrollo temporal de una RNN. Fuente: elaboración propia.

solo nos interesa el resultado final donde se ha interpretado toda la información secuencialmente.

- **Vector-to-sequence**: La entrada es siempre el mismo dato pero queremos generar una salida secuencial, como un texto.
- **Encoder-Decoder**: Unimos un *sequence-to-vector* , del cual ignoramos sus salidas, con un *vector-to-sequence* con entrada constante 0 de forma que la primera red solo se informa por el h final del *vector-to-sequence* y las siguientes del h anterior.

Para el entrenamiento de una RNN usamos una variación del *backpropagation* conocida como **backpropagation through time** (BPTT).

La idea es tomar el desarrollo temporal de la RNN y, para cada iteración, aplicar la idea del *backpropagation*. Esto sería, calcular el error de la predicción y la derivada parcial de cada peso de cada neurona para modificar los parámetros y repetir en la siguiente iteración. Una ilustración de esto la tenemos en la figura 9.

Las redes neuronales recurrentes también sufren problemas con la inestabilidad del gradiente durante el entrenamiento. Sin embargo, en estas es preferible una función de activación saturante como la taangente hiperbólica (tanh). Aunque sí se pueden utilizar técnicas de inicialización y de *dropout*.

Un problema nuevo que surge con las RNN es el **problema de la memoria**. En cada iteración se destruye algo de información debido a las transformaciones y, tras unas cuantas iteraciones, ya no queda información de los primeros datos. La solución a esto la veremos en el siguiente modelo.

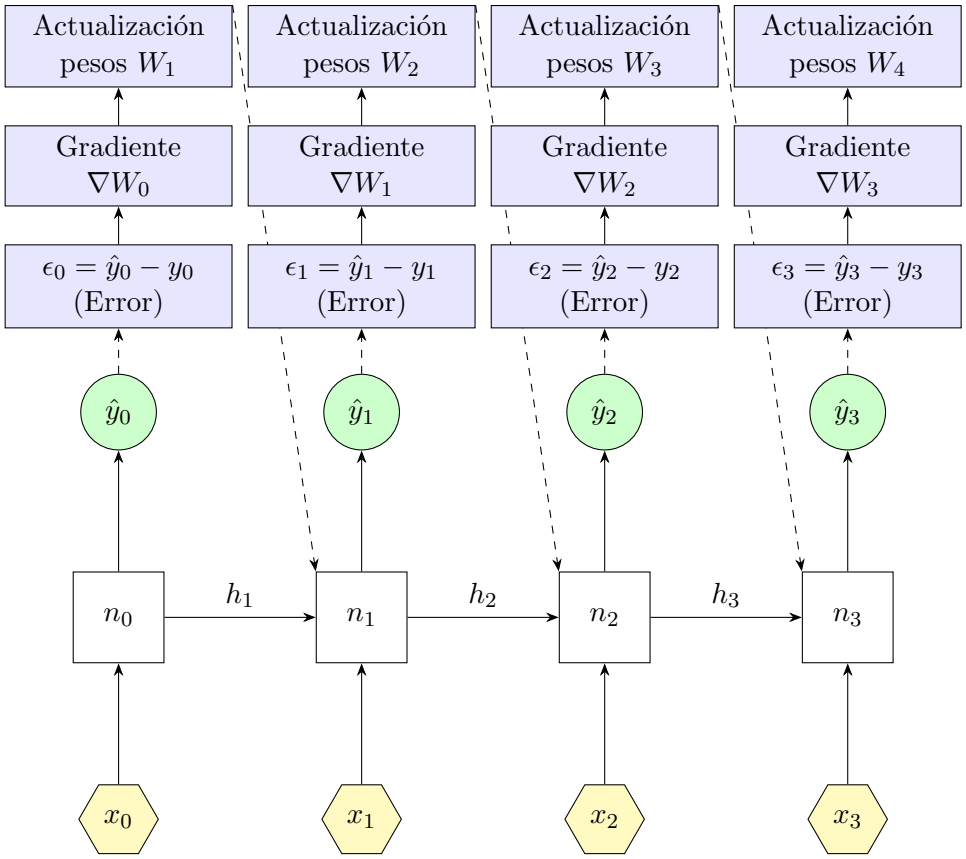


Figura 9: Diagrama del algoritmo BPTT. Fuente: elaboración propia.

3.3.4 Long Short-Term Memory

Esta arquitectura se ha popularizado en los últimos años debido a su solución para el problema de la memoria. La idea básica es, en cada iteración, transmitir una memoria a largo plazo y la salida de la celda como memoria a corto plazo. De forma que, el proceso consiste en decidir qué datos conservar para la memoria a largo plazo, cuáles añadir y cuáles olvidar. Estas decisiones son tomadas por redes neuronales. Además, hay una red neuronal que, dados los datos nuevos y los conservados, decide el dato de salida, que también funciona como memoria a corto plazo.

En la figura 10 podemos ver una representación de esta memoria. A la izquierda podemos ver en hexágonos las entradas. Los cuadrados con el texto "FC" representan redes neuronales *fully connected*. De arriba abajo, estas redes tienen las siguientes funciones:

1. La primera de ellas sirve para elegir qué datos se deben eliminar de la memoria a largo plazo.

2. La segunda indica qué datos deben ser guardados en la memoria a largo plazo.

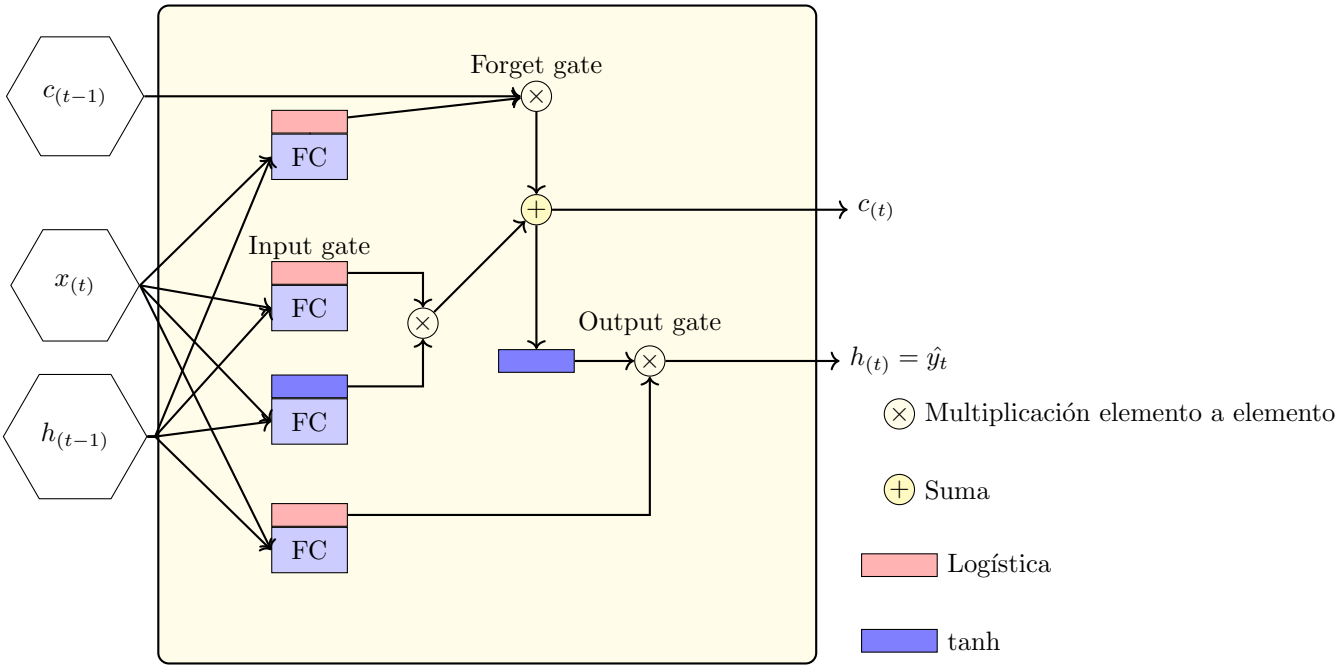


Figura 10: Representación de celda LSTM. Fuente: elaboración propia inspirado en [Gé23].

3. La tercera genera los datos que serán guardados a largo plazo si así lo indica la segunda.

4. La cuarta realiza cálculos que servirán para generar el output junto con la memoria a largo plazo actualizada.

3.3.5 Detalles metodológicos en la predicción de series temporales

Las diferencias entre la metodología de una predicción estándar y la de una serie temporal radican en que, en el segundo caso, importa el orden de los valores.

Esto nos lleva a las siguientes variaciones:

- Las variables explicativas o *features* en el caso de series temporales pueden ser los valores anteriores o datos basados en ellos. Por ejemplo, *rolling windows* de medias o desviaciones típicas. Esto es, para cada dato a predecir tomar la media o desviación típica de los valores inmediatamente anteriores. Esto lleva también a elegir con cuidado cuántos valores anteriores queremos como *features*, ya que pocos valores pueden ser insuficientes, mientras que muchos valores pueden llenar el modelo de ruido.

- La división de los datos, ya sea **train-test** o bien con conjuntos de validación, debe hacerse respetando el orden de los valores. Esto implica que no se puede aplicar muestreo aleatorio. En nuestro caso, utilizaremos un K-fold adaptado, de forma que, para cada entrenamiento y validación, va ampliando el número de datos usados como se ve en la Figura 11.

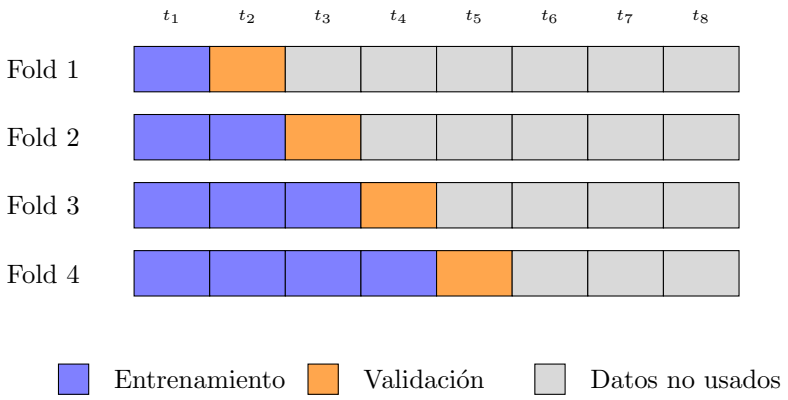


Figura 11: K-Fold para series temporales. Fuente: elaboración propia.

3.3.6 Detalles sobre la predicción de volatilidad

En esta sección vamos a ver unos matices sobre la predicción de volatilidad previos a la implementación práctica. Por un lado, debemos preguntarnos para cuándo vamos a realizar la predicción. Podríamos querer predecir para un mes después del último dato que dispongamos, para un año después o para pocos días después. Dado que es una predicción temporal univariante que, prácticamente, tiene media cero de forma estacionaria y sin tendencia ni estacionalidad. Y, como ya hemos dicho, la mayoría de eventos relevantes en el cambio de volatilidad son externos a la serie, la predicción se reduce a captar los clústeres de volatilidad que duran pocos días. Por tanto, en nuestro caso, la predicción será a un día vista.

Por otro lado, debemos pensar en cómo medir la volatilidad de un día para evaluar la predicción. Ya hemos dicho que esta volatilidad no es directamente observable, sino una propiedad de la distribución que sigan los datos. Dos opciones:

1. Usar datos intradía para estimar volatilidad realizada. Esto es lo propuesto en [AB98]. El problema de esta opción es podemos no tener esos datos o que ni siquiera existan. Puede ser porque sean mercados poco líquidos(es decir, con pocas operaciones) o porque no son datos sobre el que queremos predecir la varianza.
2. Usar un proxy de la volatilidad, en nuestro caso un estimador de volatilidad asumiendo media 0. En este caso la varianza estimada de ese día será el retorno de ese día al cuadrado. O sea, $\sigma_i^2 = r_i^2$.

La segunda opción no está exenta de problemas. En general, este proxy no es un buen estimador (de hecho, por eso los modelos ARCH necesitaban un orden alto para captar los clúster de alta volatilidad). En el anexo B, se incluye un breve estudio que se ha realizado donde se prueba por simulación Montecarlo que tener un buen Error Cuadrático Medio, usada comúnmente en este campo(véase [EBK11]), no significa que se esté haciendo una buena predicción. Y, cómo esto puede llevar a elegir modelos malos.

4 Caso Práctico: Predicción de volatilidad en el mercado de valores

Con el objetivo de evaluar y comparar el rendimiento de modelos de redes neuronales LSTM frente a modelos econométricos tradicionales (GARCH y EGARCH) en la predicción de volatilidad, se ha diseñado una serie de experimentos basados en datos financieros históricos.

4.1 Dataset

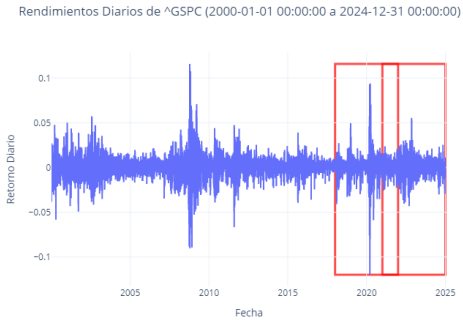
Los datos financieros básicos son los precios de cierre de dos índices: el IBEX y el SP500. Estos datos son proporcionados mediante la librería pública yfinance ([Ac25]) , que obtiene los datos del portal de noticias financieras Yahoo! Finance ([Yah25]).

Para dar generalidad al trabajo se han realizado las pruebas sobre dos ventanas temporales solapadas pero con características distintas. La primera ventana (2018–2021) incluye episodios de alta volatilidad como la pandemia de COVID-19 y la recuperación posterior, mientras que la segunda (2021–2024) presenta un entorno post-COVID más estable, pero con eventos como subidas de tipos de interés y tensiones geopolíticas. Esta segmentación permite evaluar el desempeño de los modelos bajo distintos regímenes de volatilidad.

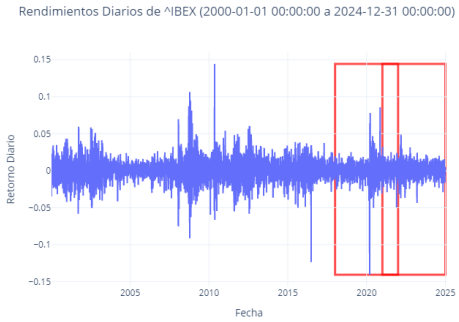
En el anexo C se indica dónde encontrar los datos y el script utilizado para extraerlos.

El dato importante para nosotros es el rendimiento logarítmico diario, pues vamos a estudiar su volatilidad. Las series de estos datos se pueden ver en la figura 12. Se enmarcan las ventanas de estudio en rectángulos rojos.

En ocasiones, las cotizaciones de cierre del mercado bursátil son modificadas artificialmente para tener en cuenta ajustes en el valor de la empresa por operaciones corporativas como reparto de dividendos, splits, etcétera. Estos ajustes son tenidos en cuenta en la serie para no afectar a sus retornos diarios.



(a) Rendimientos del 2000 al 2024 del GSPC



(b) Rendimientos del 2000 al 2024 del IBEX

Figura 12: Evolución de los rendimientos logarítmicos diarios

Diseño

Se ha experimentado con modelos econométricos clásicos, GARCH(1,1) y EGARCH(1,1), y, por parte de modelos de aprendizaje automático, con modelos LSTM. Los modelos econométricos se han implementado con la librería de python `arch` ([Sc25]), mientras que LSTM se ha implementado con la librería `tensorflow` ([Ten15]).

Se seleccionaron GARCH(1,1) y EGARCH(1,1) por su uso extendido y su capacidad de predicción probada con solo dos parámetros.

Ambos modelos se han entrenado y validado, en el caso de LSTM, con el 80% de la ventana temporal. En concreto, la validación y el entrenamiento del modelo LSTM se ha realizado con la clase `TimeSeriesSplit` de `tensorflow`.

El otro 20% de la ventana restante se ha utilizado para ejecutar el test en forma de *rolling prediction*. *Rolling prediction* es una metodología de predicción para simular la predicción a un día vista durante un período de tiempo. Esto es, se calibra hasta cierto día, se predice el siguiente , y, con el dato real de ese día añadido como input, se realiza la predicción del posterior.

Modelado con LSTM

El modelo LSTM se entrena utilizando secuencias de longitud fija (*sequence length*), que definen el número de observaciones pasadas utilizadas para predecir la siguiente varianza condicional. Dado que el modelo requiere una secuencia completa para realizar la primera predicción, el conjunto de test se ve efectivamente reducido en los primeros ℓ pasos (siendo ℓ la longitud de la secuencia). Además, los datos de entrada son escalados con MinMax Scaler.

Para ajustar los hiperparámetros del modelo, se emplea una *Grid Search* con validación

cruzada, utilizando el *wrapper* `KerasRegressor` de `scikeras`. Los hiperparámetros utilizados en la búsqueda son la función de activación, el número de neuronas, el porcentaje de *recurrent dropout* y el *learning rate*. Los modelos se evalúan utilizando el error cuadrático medio (MSE) sobre el conjunto de validación.

Modelado con GARCH y EGARCH

Los modelos GARCH(1,1) y EGARCH(1,1) se ajustan únicamente con los datos del conjunto de entrenamiento. Para garantizar una comparación justa con el LSTM, se descartan las primeras ℓ observaciones del conjunto de test en las predicciones de GARCH y EGARCH, igualando el horizonte temporal sobre el cual se computan las métricas.

Evaluación y Comparación

Las predicciones de cada modelo se comparan frente a la volatilidad empírica (calculada como el cuadrado del retorno logarítmico diario) mediante el error cuadrático medio (MSE). Además, se generan visualizaciones comparativas y se almacenan los resultados detallados en archivos Excel, facilitando su análisis posterior.

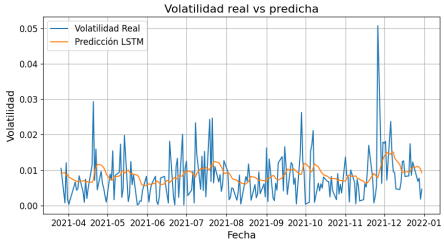
4.2 Resultados

En la tabla 13 podemos ver los MSE de la predicción con cada modelo. En la última fila podemos ver cuánto mejora el MSE del modelo LSTM respecto de los modelos GARCH y EGARCH.

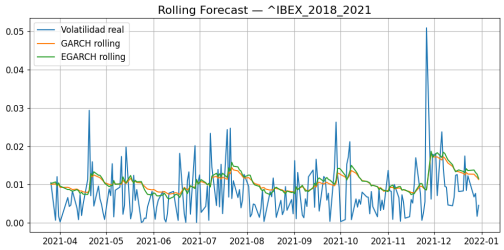
		IBEX		GSPC	
		2018-2021	2021-2024	2018-2021	2021-2024
Modelos econométricos	GARCH	5.1287E-05	3.5534E-05	2.663E-05	3.6053E-05
	EGARCH	5.3551E-05	3.5954E-05	2.9486E-05	3.5772E-05
LSTM	LSTM	4.4058E-05	2.8615E-05	2.3411E-05	3.2927E-05
	Mejora	14.10%	19.47%	12.09%	7.95%

Figura 13: Errores MSE para los distintos modelos en cada ventana temporal e índice

Gráficamente podemos ver la predicción de volatilidad comparada con el *proxy* de la volatilidad realizada en las figuras 14, 15, 16 y 17.

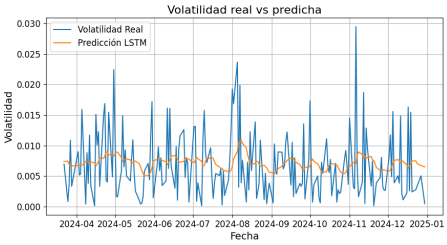


(a) Predicción con LSTM

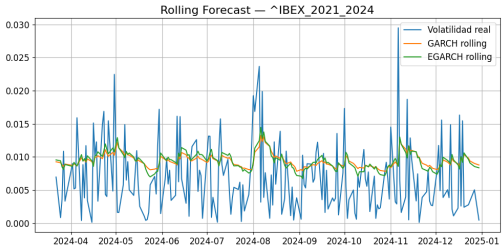


(b) Predicción con GARCH y EGARCH

Figura 14: IBEX en ventana temporal de 2018 a 2021

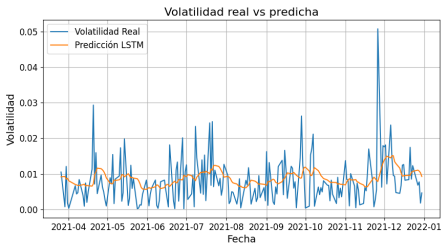


(a) Predicción con LSTM

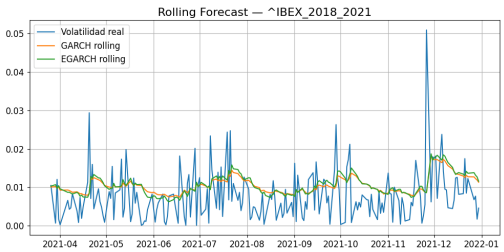


(b) Predicción con GARCH y EGARCH

Figura 15: IBEX en ventana temporal de 2021 a 2024

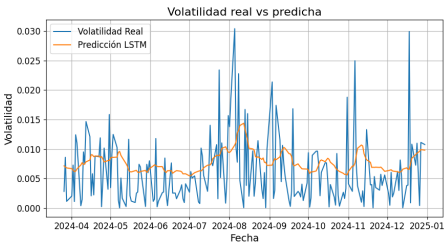


(a) Predicción con LSTM

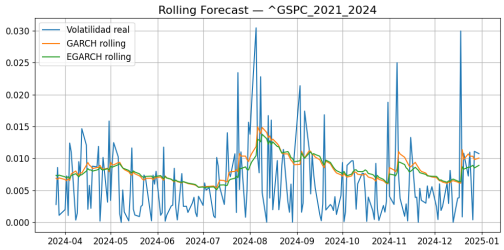


(b) Predicción con GARCH y EGARCH

Figura 16: GSPC en ventana temporal de 2018 a 2021



(a) Predicción con LSTM



(b) Predicción con GARCH y EGARCH

Figura 17: GSPC en ventana temporal de 2021 a 2024

5 Conclusiones

Este trabajo ha tenido como objetivo principal el estudio comparativo entre modelos econométricos tradicionales y técnicas modernas de aprendizaje automático (Machine Learning) para la predicción de la volatilidad en series temporales financieras. A lo largo de este estudio, se ha desarrollado un marco teórico riguroso, acompañado de un caso práctico aplicado a datos reales del mercado de valores, lo que ha permitido extraer conclusiones relevantes tanto desde una perspectiva técnica como desde una aplicación práctica en el ámbito financiero.

En primer lugar, se ha confirmado que la volatilidad es una magnitud clave en finanzas, al representar cuantitativamente la incertidumbre en el comportamiento de los precios de los activos. Su correcta estimación y predicción es fundamental para la valoración de derivados financieros, como las opciones, así como para el cálculo de métricas de riesgo ampliamente utilizadas por reguladores y entidades financieras, como el Valor en Riesgo (VaR). La importancia de la volatilidad queda reflejada tanto en su papel como input en modelos como Black-Scholes-Merton, como en su relevancia para determinar requisitos de capital regulatorio.

Desde el punto de vista metodológico, el análisis comparativo ha evidenciado las diferencias fundamentales entre el enfoque estadístico clásico (data modeling) y el enfoque basado en aprendizaje automático (algorithmic modeling). Mientras que los modelos econométricos, como ARIMA o GARCH, imponen estructuras funcionales rígidas pero interpretables, los modelos de aprendizaje automático, especialmente las redes neuronales recurrentes como las LSTM, ofrecen una mayor flexibilidad y capacidad para capturar patrones complejos en los datos, a costa de una menor interpretabilidad.

El trabajo ha demostrado que los modelos tradicionales, en particular los modelos GARCH, continúan siendo herramientas válidas para la predicción de volatilidad. Sin embargo, los modelos de Machine Learning han mostrado una mayor capacidad para adaptarse a los datos, especialmente en la presencia de relaciones no lineales, cambios de régimen o ruido estructurado. Las redes LSTM, en particular, han mostrado ser eficaces en la captura de dependencias temporales de largo plazo, superando en términos de error de predicción a los modelos tradicionales en la mayoría de los experimentos realizados. No obstante, esta superioridad predictiva viene acompañada de una serie de desafíos metodológicos, como la necesidad de un mayor volumen de datos, una cuidadosa selección de hiperparámetros y una mayor complejidad computacional.

En cuanto al caso práctico desarrollado, se ha constatado que la predicción de la volatilidad sigue siendo una tarea altamente compleja, especialmente en presencia de eventos exógenos inesperados (shocks). Dado que los modelos univariantes sólo pueden capturar los patrones históricos de la volatilidad, no son capaces de anticipar los shocks, sino únicamente de modelar su propagación una vez ocurren. Esta limitación intrínseca hace evidente la necesidad de incorporar información exógena en los modelos, lo cual abre una interesante línea de investigación futura.

Como conclusión general, se puede afirmar que la predicción de volatilidad en mercados financieros requiere un enfoque multidisciplinar que combine teoría estadística, conocimientos financieros y capacidades computacionales. Ni los modelos clásicos ni los modernos son, por sí solos, soluciones universales al problema de la predicción. En cambio, su combinación estratégica puede ofrecer soluciones más robustas y ajustadas a los desafíos actuales de los mercados financieros.

Finalmente, cabe destacar que el proceso de desarrollo de este trabajo ha aportado una comprensión profunda no solo de los modelos matemáticos y computacionales involucrados, sino también de los retos prácticos que conlleva el análisis y la predicción de fenómenos financieros reales.

A Identificación, selección, estimación y diagnosis

Identificación de un modelo ARIMA

Antiguamente se utilizaba la metodología Box-Jenkins, resumida en el sigiuente esquema:

- 1) Transformación de las variables para conseguir estacionariedad. Por un lado, usar una función de la familia Box-Cox para conseguir varianza constante y, por otro, diferenciar para conseguir media constante. Esto último se puede apoyar de contrastes de raíces unitarias.
- 2) Determinar p,q mediante la f.a.s. y la f.a.p.
- 3) Si es estacional, determinar P, Q del modelo SARIMA.

En la actualidad, la capacidad de cálculo de los computadores permite tomar varios modelos que puedan ser posibles, estimarlos y seleccionar entre ellos.

Selección y estimación

El criterio de selección suele ser una fórmula que pondere la varianza intrínseca del modelo con el número de parámetros. Como el criterio AIC: $AIC = T \cdot ln(\hat{\sigma}_{MV}) + 2k$, donde T es el tamaño muestral, k el número de parámetros y MV el valor de la función de verosimilitud del modelo. Se escoge el modelo de menor valor AIC. Existen otros criterios, pero la idea básica es la misma.

Para la estimación se utilizan algoritmos de optimización no lineal que intentan maximizar la función de máxima verosimilitud. También, algoritmos recursivos como el filtro de Kalman.

Diagnosis

El proceso de diagnosis consiste en comprobar si se cumplen las hipótesis básicas del modelo. Estas hipótesis se resumen en los siguientes contrastes sobre los residuos:

- 1) Contraste de media cero. Si la media es 0 se debe cumplir $\frac{\bar{a}}{\hat{\sigma}/\sqrt{T}} N \sim (0,1)$. En este contraste hay que tener especial cuidado con los outliers e investigar de dónde proceden.
- 2) Contraste de varianza constante. Suponiendo que hay tramos con varianzas distintas, digamos s_i^2 con n_i observaciones, entonces $ln(\hat{\sigma}^2) - \sum_{i=1}^{n} n_i \cdot ln(s_i^2) \sim \chi^2_{n-1}$.

- 3) Contraste de incorrelación en cualquier retardo. El contraste Ljung-Box para varios retardos comprueba esta hipótesis.
- 4) Contraste de normalidad. Cualquier contraste de bondad de ajuste como el Jarque-Bera nos serviría. También serían útiles contrastes para los coeficientes de asimetría y curtosis.

También es importante el estudio de outliers o atípicos, aquellos datos que se alejan de la media en 2 o 3 desviaciones típicas. Además, sería útil añadir algún coeficiente a la parte autorregresiva o a la MA y ver si mejora el valor del criterio de selección.

Por último, la 2), 3) y 4) se deben cumplir también para las distribuciones condicionadas. El siguiente apartado servirá para modelar casos con varianza condicional variable. Conocido en econometría como heterocedasticidad condicional.

El diagrama de flujo de la Figura 18 resume el proceso:

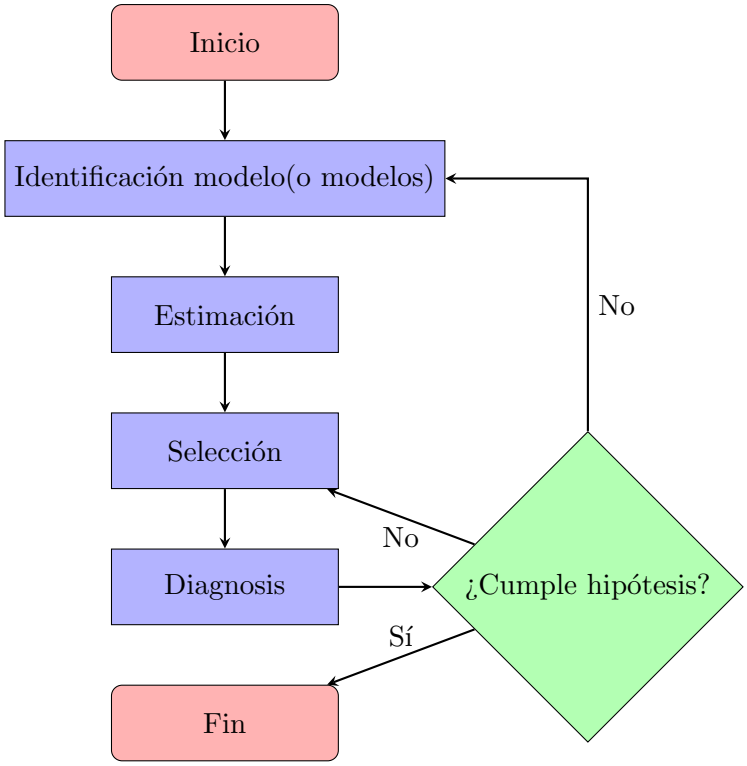


Figura 18: Diagrama modelado con ARIMA. Fuente: elaboración propia.

B Breve estudio sobre predicción de volatilidad

Supongamos una serie temporal $Y_t = \varepsilon_t$ donde $\varepsilon_t \sim N(0, \sigma_t)$. Suponemos también que sabemos la serie temporal de las varianzas σ_t^2. Esto no significa que sepamos los valores Y_t, pero sí sabemos su distribución. ¿Cómo podríamos comprobar que conocemos la distribución de Y_t? Además, en este caso, con conocer la varianza ya determinamos la distribución.

Una forma simple es mirar la serie de percentiles. Esto es, dado un Y_t vemos cuál es su percentil en $N(0, \sigma_t)$, que lo llamaremos p_t. Pues, bajo el supuesto de que $Y_t \sim N(0, \sigma_t)$ para todo t, $p_t/100 \sim U(0, 1)$, una uniforme de 0 a 1. La comprobación de que $p_t/100 \sim U(0, 1)$ se puede hacer con un test de bondad de ajuste como Kolgomorov-Smirnov, o un test de permutaciones basado en este KS.

Bajo estos supuestos, dada una serie de σ_t vamos a hacer simulaciones de Y_t, i.e. general valores aleatorios de la serie, y mediremos su función de error usual:

$$MSE(Y_t, \hat{\sigma}_t) = (Y_t^2 - \sigma_t^2)^2$$

Vamos a comprobar empíricamente que, en general, dadas varias realizaciones del modelo, la que menor MSE o QL tiene, suele ser una de las que peor se ajusta su serie de percentiles a una $U(0, 1)$. Esto lo veremos a través del p-valor del test KS.

Aplicando esto, vemos una gráfica generada a partir de 50 series de volatilidad de 100 pasos. Cada serie de volatilidad genera 10 simulaciones, sobre estas calculamos el Error Cuadrático Medio del proxy simulado. Por otro lado, calculamos la serie de percentiles de la simulación y el p-valor de un test de bondad de ajuste de estos percentiles a una distribución uniforme $U(0, 1)$. Finalmente, en la Figura 19 mostramos un histograma sobre qué p-valor tiene en cada serie la simulación con menor MSE frente el p-valor resto de series. Como vemos, no hay una relación directa entre tener un error bajo y generar series de percentiles parecidas a una U(0,1).

También se observa que, aunque aplicamos el test a un modelo que es exactamente el que genera la serie, lo normal es que la serie de percentiles no se parezca a una uniforme (0,1), lo que también muestra lo difícil que es comprobar que las volatilidades son correctas.

De la misma forma, si tenemos varios modelos de estimación de volatilidades y unos datos fijos, podría resultar que el modelo que menor función de error tiene (o un modelo con poco error) realmente no esté produciendo mejores predicciones de varianza, sino que, por casualidad, esté acertando la varianza σ_t^2 que hace que $Y_t^2 = \sigma_t^2$.

Por tanto, sería interesante para estos casos buscar una forma de evitar que modelos que se adapten mal a la distribución que deberían tomar fueran excluidos de la validación. Por ejemplo, aquellos con sobrerrepresentación en algunos percentiles.

Otro enfoque sería utilizar en la validación test de hipótesis pensados para comparar la capacidad predictiva de los modelos entrenados, como el test Diebold-Mariano, aunque

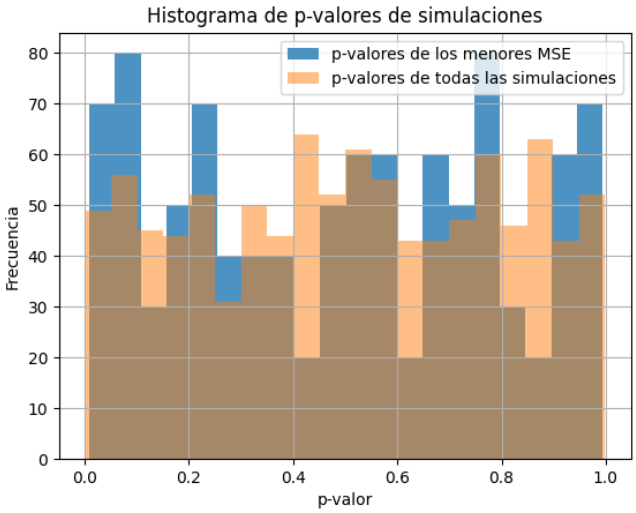


Figura 19

siguen teniendo la limitación de usar el proxy de la varianza.

Por otro lado, en los casos de volatilidad en mercados financieros, la serie no sigue una distribución normal, sino que suele tener curtosis alta. Por lo que, en estos casos, quizá sería también interesante entrenar el modelo con una función que corrija por curtosis.

C Repositorio con programas y datos

En la URL https://github.com/jorgelorenz/tfg-informatica se encuentran los programas y datos utilizados para el experimento. Se encuentran las siguientes carpetas y archivos:

- Archivo `data_extraction.py`: Este código es el usado para sacar los datos de retornos financieros.
- Carpeta **datasets**: Aquí se encuentran los datos descargados en formato csv.
- Archivo `models.py`: Este código es el usado para el experimento. Se calibran y prueban los modelos LSTM, GARCH y EGARCH.
- Carpeta **outputs**: Contiene los excel con la salida de cada modelo para cada ventana temporal. En concreto, la medición de test final, los datos predichos y reales y los datos de los modelos LSTM entrenados para la validación.
- Carpeta **images**: Contiene las imágenes de la predicción generadas.

References

[AB98] Torben Andersen and Tim Bollerslev. Answering the skeptics: Yes, standard volatility models do provide accurate forecasts. *International Economic Review*, 39(4):885–905, 1998.

[Ac25] Ran Aroussi and contributors. yfinance: Yahoo! finance market data downloader, 2025. Version 0.2.27, software library, accessed 2025-06-30.

[Ama67] Shunichi Amari. A theory of adaptive pattern classifiers. *IEEE TRANSACTIONS ON ELECTRONIC COMPUTERS*, 1967.

[Bol86] Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, 1986.

[Bol15] Boletín Oficial del Estado. Ley 20/2015, de 14 de julio, de ordenación, supervisión y solvencia de las entidades aseguradoras y reaseguradoras. https://www.boe.es/buscar/act.php?id=BOE-A-2015-7897, 2015. BOE n.º 168, 15 de julio de 2015, págs. 58455–58611.

[Bre01] Leo Breiman. Statistical modeling: The two cultures. *Statistical Science*, 16(3):199–231, 2001.

[Cor09] Flavio Corsi. A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics*, 7(2):174–196, 2009.

[EBK11] Robert Engle, Christian Brownlees, and Bryan Kelly. A practical guide to volatility forecasting through calm and storm. *The Journal of Risk*, 14(2):3–26, 2011.

[Eng82] Robert Engle. Autoregressive conditional heteroskedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50(4):987–1007, 1982.

[GJR93] Lawrence R. Glosten, Ravi Jagannathan, and David E. Runkle. On the relation between the expected value and the volatility of the nominal excess return on stocks. *The Journal of Finance*, 48(5):1779–1801, 1993.

[GRLM04] Gloria González-Rivera, Tae-Hwy Lee, and Santosh Mishra. Forecasting volatility: A reality check based on option pricing, utility function, value-at-risk, and predictive likelihood. *International Journal of Forecasting*, 20(4):649–663, 2004.

[Gé23] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, 3 edition, 2023.

[Hor89] Kurt Hornik. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[K+22] Mahinda Mailagaha Kumbure et al. Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications*, 200:117046, 2022.

[KJM22] Kshitij Kakade, Ishan Jain, and Aswini Kumar Mishra. Value-at-risk forecasting: A hybrid ensemble learning garch-lstm based approach. *Resources Policy*, 76:102571, 2022.

[KW18] Ha Young Kim and Chang Hyun Won. Forecasting the volatility of stock price index: A hybrid model integrating lstm with multiple garch-type models. *Expert Systems with Applications*, 93:417–426, 2018.

[LD18] Chuong Luong and Nikolai Dokuchaev. Forecasting of realised volatility with the random forests algorithm. *Journal of Risk and Financial Management*, 11(4):56, 2018.

[Liu19] Yang Liu. Novel volatility forecasting using deep learning – long short term memory recurrent neural networks. *Expert Systems with Applications*, 124:292–304, 2019.

[Nel91] Daniel B. Nelson. Conditional heteroskedasticity in asset returns: A new approach. *Econometrica*, 59(2):347–370, 1991.

[Nie15] Michael A. Nielsen. *Neural Networks and Deep Learning*. 2015.

[Pe8] Daniel Peña. *Fundamentos de estadística*. Alianza Editorial, 1 edition, 2008.

[Pe0] Daniel Peña. *Regresión y diseño de experimentos*. Alianza Editorial, 1 edition, 2010.

[Pe2] Daniel Peña. *Análisis de series temporales*. 2012.

[RHW86] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, pages 318–362. MIT Press, 1986.

[RP17] Werner Kristjanpoller R. and Esteban Hernández P. Volatility of main metals forecasted by a hybrid ann-garch model with regressors. *Expert Systems with Applications*, 88:376–386, 2017.

[Sc25] Kevin Sheppard and contributors. arch: Autoregressive conditional heteroskedasticity models, 2025. Version 5.3.1, software library, accessed 2025-06-30.

[SO13] Jesús P. Ibáñez Sandoval and Beatriz Domingo Ortuño. La transposición de basilea iii a la legislación europea. *Estabilidad Financiera*, 22:35–50, 2013.

[Ten15] TensorFlow Developers. Tensorflow: An end-to-end open source machine learning platform, 2015. Software library.

[Tsa05] Ruey Tsay. *Analysis of Financial Time Series*. Wiley, 2 edition, 2005.

[TTS09] Ling-Bing Tang, Ling-Xiao Tang, and Huan-Ye Sheng. Forecasting volatility based on wavelet support vector machine. *Expert Systems with Applications*, 36(2):1207–1214, 2009.

[Yah25] Yahoo Finance. Yahoo finance, 2025. Accedido el 30 de junio de 2025.

[ZLH+22] Chun-Xia Zhang, Jun Li, Xing-Fang Huang, Jiang-She Zhang, and Hua-Chuan Huang. Forecasting stock volatility and value-at-risk based on temporal convolutional networks. *Expert Systems with Applications*, 206:117778, 2022.