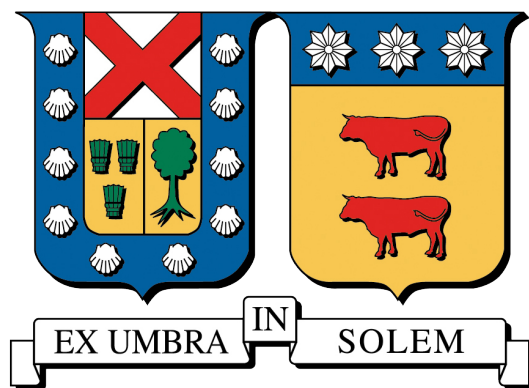


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INDUSTRIAS
VALPARAÍSO - CHILE



**PROPUESTA PARA LA PREDICCIÓN DE PRECIOS DE LOS
ÍNDICES BURSÁTILES UTILIZANDO REDES NEURONALES
PROFUNDAS CON UN ENFOQUE DE AUTO-SIMILITUD**

CARLOS BASTIÁN MENDOZA RIVEROS

TESIS PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL INDUSTRIAL

Y

AL GRADO DE
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA INDUSTRIAL

PROFESOR GUÍA : WERNER KRISTJANPOLLER
PROFESOR CORREFERENTE :

2022

*Quiero agradecer especialmente a mis padres Viviana y Juan Carlos
por su apoyo incondicional en este proceso y en todas
mis locuras que me he propuesto en la vida,
agradecer a toda mi familia que
de igual manera estuvieron presentes en mí.*

*Agradecer también a todos mis amigos que fui conociendo
en estos años de universidad, por esos momentos
de no solo estudio sino también de diversión, risas, apañes,
pláticas interesantes, consejos valiosos y deporte infaltable.*

*También, valorar todo el aporte
intelectual que generaron los profesores, ayudantes
y compañeros que me acompañaron hasta este momento.*

*Por último, pero no menos importante
agradecerme a mí, por creer en mí
y por poder llevar aquellos momentos difíciles
donde solo me quedó mirar con optimismo las cosas
para poder finalizar este y muchos otros procesos.*

Resumen

Los Índices Bursátiles son de vital importancia para entender el comportamiento de los mercados financieros y económicos de un determinado sector y/o país. En este trabajo se propone mejorar la predicción de precios diarios de los índices S&P 500, DAX, AEX y SMI, realizando un entrenamiento y validación de las redes neuronales profundas de arquitectura MLP, SimpleRNN y LSTM inspirándose en los comportamientos fractales y de Auto-Similitud que existen dentro de las series financieras. Se experimenta con los modelos base entrenados solo con datos diarios y a los modelos propuestos entrenados con datos diarios y re-entrenados con datos intradiarios (frecuencias cada: 60 min, 30 min y 15 min). Se utiliza el test de Diebold Mariano y el Model Confidence Set para verificar la superioridad predictiva de este método. Los resultados indican que para todos los índices y arquitecturas los pronósticos de los modelos propuestos son distintos y superiores a los modelos bases.

Abstract

Stock Indices are of vital importance to understand the behavior of the financial and economic markets of a certain sector and country. In this work, we proposed to improve the prediction of daily prices of the S&P 500, DAX, AEX and SMI indices, carrying out a training and validation of the deep neural networks of MLP, SimpleRNN and LSTM architecture inspired by the Fractal and Self-Similarity behaviors that exist within the networks. financial series. We experiment with the base models trained only with daily data and the proposed models trained with daily data and re-trained with intraday data (frequencies: 60 min, 30 min and 15 min). The Diebold Mariano test and the Model Confidence Set are used to verify the predictive superiority of this method. The results indicate that for all indices and architectures the forecasts of the proposed models are different and superior to the base models.

Resumen Ejecutivo

Los índices bursátiles son de vital importancia para entender el comportamiento de los mercados financieros y económicos de un determinado sector de la economía o país. Investigadores, inversionistas, entidades y organizaciones han estado constantemente realizando mejoras en el ámbito del machine learning y deep learning para la predicción de estas series. En esta investigación se propone mejorar la predicción de precios diarios de los índices S&P 500, DAX, AEX y SMI, realizando un entrenamiento y validación de las redes neuronales inspirándose en los comportamientos fractales y de Auto-Similitud que existen dentro de las series financieras. Se proponen la inclusión de las series minútales cada 60 min, 30 min y 15 min con las series de observaciones diarias (cada 1 día) como datos de entrada para los modelos con las arquitecturas MLP, SimpleRNN y LSTM. En específico se estudian los modelos D-Model, 60MD-Model, 30MD-Model y 15MD-Model para cada arquitectura e índice bursátil. Los resultados indican que para el S&P 500 los modelos propuestos superan a todos los modelos bases, el mejor modelo 60MD-MLP disminuye el error del modelo base D-MLP en un 21,46 %, para el índice DAX el mejor modelo propuesto 30MD-LSTM mejora el rendimiento en un 10,5 % al modelo base D-LSTM, en relación al índice AEX el mejor modelo 15MD-SimpleRNN disminuye el error en un 7,42 % del modelo base D-SimpleRNN y para el índice SMI, el mejor modelo 60MD-LSTM mejora el rendimiento en un 12,43 % del modelo base D-LSTM. Para comprobar la diferencia entre los pronósticos del modelo base contra el propuesto en la arquitectura correspondiente se utiliza el Test de Diebold-Mariano y para encontrar el modelo con mayor superioridad predictiva se utiliza el test The Model Confidence Set. En conclusión se demuestra que el enfoque de Auto-Similitud propuesto para entrenar redes neuronales es estadísticamente superior que los modelos bases para todos los índices y arquitecturas estudiadas en esta investigación.

Índice de Contenidos

1. Problema de Investigación	2
1.1. Introducción	2
1.2. Antecedentes	4
2. Objetivos	7
2.1. Objetivos Generales	7
2.2. Objetivos Específicos	7
3. Marco teórico	9
3.1. Índices bursátiles	9
3.2. Auto-similitud	10
3.3. Inteligencia Artificial	12
3.3.1. Machine Learning	13
3.3.2. Deep Learning	14
3.4. Arquitecturas de redes neuronales	16
3.4.1. MLP	16
3.4.2. SimpleRNN	17
3.4.3. LSTM	18
4. Metodología	21
4.1. Obtención de datos	21
4.1.1. Índices bursátiles	21
4.2. Estadísticas descriptivas	25
4.3. Estructura de los modelos	29
4.3.1. Configuración de los vectores de entrada	29
4.3.2. Configuración de los modelos	30
4.3.3. Enfoque de Auto-similitud	32
4.4. Experimentos	33
4.4.1. Comparando la capacidad y superioridad predictiva	34
5. Resultados	37
5.1. S&P 500	37
5.2. DAX	40
5.3. AEX	42
5.4. SMI	44
6. Conclusiones	46
Bibliografía	48

Índice de figuras

3.1. Helecho de Barnsley.	11
3.2. Precios diarios del Índice DAX.	12
3.3. Precios cada 1 hora del Índice DAX.	12
3.4. Inteligencia Artificial, Machine Learning y Deep Learning.	13
3.5. Diagrama de ejemplo de arquitectura de una MLP.	17
3.6. Unrolled diagram of a recurrent neural network architecture.	18
3.7. Una ilustración gráfica de una celda LSTM.	20
4.1. S&P500 - Currency in \$ USD.	23
4.2. DAX - Currency in \$EUR.	23
4.3. SMI Currency in \$CHF.	24
4.4. AEX - Currency in \$CHF.	24
4.5. Cantidad de observaciones por tipo de conjunto de datos.	25
4.6. Neural networks model structure diagram.	31
4.7. Time size for training, validation, and training data for each frequency data set.	31
4.8. Metodología con el enfoque de Auto-Similitud.	33
5.1. SP500 prices and best model proposed: 60MD-MLP.	39
5.2. DAX prices and best model proposed: 30MD-LSTM.	41
5.3. AEX prices and best model proposed: 15MD-SimpleRNN.	43
5.4. SMI prices and best model proposed: 60MD-LSTM.	45

Índice de tablas

4.1. Estadísticas descriptivas del Índice S&P 500.	26
4.2. Estadísticas descriptivas del Índice DAX.	27
4.3. Estadísticas descriptivas del Índice AEX.	27
4.4. Estadísticas descriptivas del Índice SMI.	28
4.5. Configuraciones presentes en la investigación.	34
5.1. Resultados S&P 500	39
5.2. Resultados Índice DAX	41
5.3. Resultados Índice AEX	43
5.4. Resultados Índice SMI	45

1 Problema de Investigación

1.1. Introducción

Los índices bursátiles son un valor que refleja el comportamiento conjunto de una serie de acciones mostrando el rendimiento global de un determinado mercado, estas sirven para representar la evolución financiera de un país, sector de la economía o algún tipo de conjunto de activos financieros permitiendo medir la rentabilidad y riesgo del mercado. Además de distintas aplicaciones, tales como la valorización de la rentabilidad de una empresa sobre un determinado mercado o la creación y análisis de una cartera de acciones indexadas a estas series. Este trabajo busca mejorar la precisión en la predicción de los índices accionarios inspirándose en los comportamientos que existen dentro de las series financieras de Auto-Similitud introducido por [Peters \(1994\)](#) quien plantea la Hipótesis de Mercados Fractales, declarando que las series de precios de acciones no siempre siguen un comportamiento aleatorio *Random Walk*, sino que en realidad tienen comportamientos fractales dado que muestran una estructura similar cuando están en diferentes intervalos de tiempo (Concepto estudiado por diversos autores, tales como, [Cheong \(2010\)](#), [Madan and Schoutens \(2020\)](#) y [Calvet and Fisher \(2002\)](#)). Se plantea una metodología que incorpora los precios y rentabilidades de estos índices en observaciones de distintas frecuencias de tiempo (1 día, 60 min, 30 min, 15 min) utilizando el concepto de Auto-Similitud en las series financieras para el entrenamiento y validación de los modelos de redes neuronales profundas. A su vez, tal como declaran los autores [Baek and Kim \(2018\)](#) al entrenar las Redes Neuronales Profundas DNNs con cantidad limitada de datos puede deteriorar la precisión de la predicción, resultando en un problema de overfitting¹. Este es un problema

¹Concepto asociado al sobre ajuste de las redes neuronales, en donde las redes tienen un mejor rendimiento en el entrenamiento que en la validación (Ver [Xue \(2019\)](#)).

que se presenta en las series financieras de frecuencia diaria debido a la poca cantidad de información que estas entregan² en comparación a la cantidad de información que necesita un modelo de red neuronal para conseguir un óptimo rendimiento. La metodología consiste en mezclar esta información a través del concepto de Auto-Similitud en el entrenamiento y validación de las redes neuronales, en una primera instancia entrenando un modelo con datos de frecuencia diaria (D-Model) y luego re-entrenandolo con datos de frecuencia intraday (MD-Model) a esta metodología se trabajará como el enfoque de Auto-Similitud. Es así que aumentando la cantidad de datos de entrada de la red neuronal, proporciona un mejor entrenamiento y validación y por ende una mejora en la precisión de la predicción de los precios de los índices accionarios. En particular, se propone estudiar los siguientes modelos:

- 60MD-Model: Consiste en re-entrenar el modelo de datos diarios (D-Model) con un conjunto de datos con observaciones cada 60min.
- 30MD-Model: Consiste en re-entrenar el modelo de datos diarios (D-Model) con un conjunto de datos con observaciones cada 30min. .
- 15MD-Model: Consiste en re-entrenar el modelo de datos diarios (D-Model) con un conjunto de datos con observaciones cada 15min.

Finalmente, se intenta comprobar a base de resultados experimentales en 4 índices bursátiles que la combinación de datos de frecuencia diaria con datos de frecuencia intradiaria intraday a través del método de Auto-Similitud pueda mejorar la eficiencia de los modelos de deep-learning utilizando distintas arquitecturas, tales como, Multilayer Perceptron MLP, Simple Recurrent Neural Network SimpleRNN y Long-Short Term Memory LSTM para realizar los experimentos.

La estructura de trabajo esta organizada por una introducción con el objetivo de investigación y la descripción de artículos relacionados a esta investigación. Una sección de metodología que incluye la descripción, obtención y pre-procesamiento del conjuntos de datos, explicación de los modelos, sus arquitecturas y la metodología de entrenamiento de

²10 años de información de precios diarios equivalen aproximadamente a 2500 observaciones.

los modelos propuestos con el enfoque de autosimilitud (i.e. entrenamiento con datos de frecuencia diaria e re-entrenamiento con datos de frecuencia intradiaria) y una explicación de los experimentos a analizar incluyendo una sección de análisis de resultados. Por ultimo, una sección de conclusiones y discusión.

1.2. Antecedentes

Estas series han sido estudiadas por muchos autores, organizaciones e inversores, que buscan entender el comportamiento de los precios, y así mejorar la toma de decisiones, desempeñando técnicas de predicción que se han basado en una primera instancia en métodos tradicionales con los modelos paramétricos estadísticos. En una segunda instancia, proponiendo modelos econométricos, tales como, modelos de autoregressive conditional heteroscedasticity (ARCH) [Engle \(1982\)](#) y generalized autoregressive conditional heteroscedasticity (GARCH) [Bollerslev \(1986\)](#). Últimamente, a causa del aumento de la tecnología se ha incursionado en técnicas de inteligencia artificial y machine learning para la predicción de los precios de los índices bursátiles, acciones de mercado y otras series financieras. El uso de machine learning ha demostrado en varias ocasiones ser más preciso en la predicción de series financieras que los modelos estadísticos y econométricos (e.g. [Siarni-Namini and Namin \(2018\)](#), [Jang and Lee \(2019\)](#), [Moshiri and Cameron \(2000\)](#) and [Adebiyi and Adewumi \(2014\)](#)). [Erkam, Kayukutlu Gulgun, and Tugrul \(2011\)](#) declaran que las redes neuronales son conocidas por ser dinámicas y efectivas en la predicción de acciones de mercado, en su investigación analizan la efectividad del uso del Deep-learning con las redes multi-layer perceptron (MLP), dynamic artificial network (DAN2) y un modelo híbrido de redes neuronales con modelos GARCH, demostrando a través de resultados experimentales la superioridad predictiva de estas técnicas sobre las técnicas estadísticas. [Fischer and Krauss \(2018\)](#) proponen una red neuronal Long-Short Term Memory (LSTM) para predecir la dirección de los movimientos de las acciones constituidas en el índice S&P 500 quienes declaran que tiene un rendimiento mejor en comparación de los métodos de memory-free classification, tales como, random forest (RAF), deep neural network (DNN) y logistic regression (LOG) demostrando la superioridad de las redes neuronales recurrentes (RNNs)

en la predicción de series de tiempo financieras.

Del trabajo de [BeratSezer, UgurGudelekA, and MuratOzbayoglu \(2020\)](#) quienes realizaron una revisión sistemática de varias publicaciones en el ámbito del deep-learning para la predicción de series financieras se observa que para los índices bursátiles existen algunas corrientes de innovación e investigación. La primera esta relacionada con la investigación en técnicas de arquitectura para predecir estas series utilizando datos de precios OCHLV³ (Ver [Liu, Zeng, and Carrio \(2018\)](#), [Patel, Shah, Thakkar, and Kotecha \(2015a\)](#), [Liu, Gong, and Yang \(2020\)](#)). Un caso de combinación de arquitectura es el propuesto por [Chen and Ge \(2019\)](#) quienes exploran el funcionamiento del mecanismo de atención (attention mechanism) en las redes LSTM para la predicción de movimientos de dirección. Otro ejemplo, es el estudio realizado por [Lu, Li, and Wang \(2021\)](#) quienes declaran que debido a la volatilidad de los mercados los precios accionarios siguen con frecuencia un comportamiento no lineal aumentando la dificultad en la predicción. Ellos proponen el método CNN-BiLSTM-AM para predecir el precio de cierre del siguiente día. Compuesto por una red neuronal convolucional (CNN), una bi-directional long short-term Memory (BiLSTM) y un mecanismo de atención (AM) en donde los resultados experimentales indican que el modelo propuesto tiene un mejor rendimiento que la utilización de las redes CNN, LSTM y CNN-LSTM.

Otra corriente de investigación es sobre la propuesta de distintos conjuntos de características (feature set) en el vector de entrada, tales como, combinación de predicciones econométricas, indicadores técnicos, descomposiciones de series y otros. [Kim and Wo \(2018\)](#) realizan una comparación de los modelos GARCH, contra una multilayer perceptron MLP y LSTM con predicciones de los modelos GARCH para predecir la volatilidad del Índice Bursátil KOSPI. [Cao, Li, and Li \(2019\)](#) combinan 2 tipos de modos de descomposición empírica (EMD) y el modo *complete ensemble decomposition con adaptive noise* (CEEMDAN) como variables de entradas para la red LSTM, los resultados experimentales demuestran que al incluir esta información la precisión es mejor que los modelos de SVM y MLP. Tanto en [Hsieh, Hsiao, and Yeh \(2011\)](#) y [Bao, Yue, and Rao \(2017\)](#) proponen modelos híbridos con la descomposición de Wavelet, el primero propone utilizarlo junto a una RNN, optimizando

³Open price, Close price, High price, Low price y Volumen de la acción.

los pesos a través del algoritmo ABC⁴. En el segundo estudian 6 índices de futuros para examinar la metodología de combinar con Stacked autonecoders (SAEs) y LSTM para la predicción del precio de cierre del día siguiente. Los resultados experimentales indican que los modelos propuestos superan a otros modelos similares en la precisión de la predicción.

Actualmente, también se ha incursionado en estudios para la predicción de series con precios intraday. [Aévalo, Niño, and Sandoval \(2016\)](#) utilizan datos de alta frecuencia (High Frequency Data) para predecir las transacciones tick-by-tick de APPL utilizando una Deep Neural Network DNN. [Chong, Han, and Park \(2017\)](#) realizan un análisis sistemático en los parámetros de una DNN. [Li, Huang, Deng, and Zhu \(2014\)](#) utilizan datos intraday con la integración y/o fusión de información de noticias para predecir retornos intraday. [Frinjs and Margaritis \(2008\)](#) utilizan un método con una estrategia similar para predecir la volatilidad del final del día con la volatilidad intraday utilizando modelos GARCH. Sin embargo, no utiliza la combinación de estas frecuencias como datos para el entrenamiento de redes neuronales.

Las investigaciones presentadas buscan en general mejorar la arquitectura de las redes neuronales o aumentar la cantidad de variables de entradas. Si bien, se han incursionado en trabajos respecto a precios intradiarios, no se ha propuesto combinar estos precios con distintos horizontes de tiempos para mejorar la predicción de las series de índices accionarios.

⁴The Artificial Bee Colony algorithm.

2 Objetivos

2.1. Objetivos Generales

Mejorar la precisión de los pronósticos de los índices bursátiles de cuatro países a través del entrenamiento y validación de redes neuronales profundas utilizando la mezcla de precios intradiarios con precios diarios presentado en este trabajo como el enfoque de Auto-Similitud.

2.2. Objetivos Específicos

1. Presentar el enfoque de Auto-Similitud en que consiste la metodología que se aplica en el entrenamiento y validación de las redes neuronales explicando su funcionamiento y forma de utilizar en las redes neuronales profundas.
2. Entrenar los modelos propuestos (mezclar datos intradiarios de 60 min, 30 min y 15 min y diarios) y modelos base (datos diarios) en las redes neuronales con arquitectura MLP, SimpleRNN y LSTM.
3. Probar el enfoque de Auto-Similitud en las series bursátiles, S&P 500, DAX, AEX y SMI.
4. Comparar la función de pérdida Error Cuadrático Medio (MSE) de las redes neuronales para los modelos propuesto contra las redes bases que solo utilizan precios diarios.

5. Verificar estadísticamente que los modelos propuestos poseen mejor rendimiento que los modelos bases a través del Test de Diebold Mariano.
6. Encontrar el modelo con mayor capacidad predictiva dentro de los modelos estudiados en esta investigación.

3 Marco teórico

En el presente capítulo se busca explicar los distintos conceptos trabajados en esta investigación, partiendo por una definición de los índices bursátiles, el concepto de Auto-Similitud, inteligencia artificial, machine learning y redes neuronales.

3.1. Índices bursátiles

Los índices bursátiles son un indicador de un conjunto de acciones de un determinado espectro de la economía que se agrupan para analizar, entender y reflejar el comportamiento que estas van teniendo durante un determinado periodo de tiempo. Algunos sectores de la economía que representan estos índices son los sectores industriales y sectores tecnológicos, también, son conocidos por reflejar el mercado financiero de un país, algunos países agrupan según sus acciones según la capitalización bursátil o según la reputación de las empresas (Blue-Chip Stocks). Estos constituyen la principal referencia para comparar el rendimiento de distintos instrumentos financieros, son utilizados. Poseen distintas aplicaciones, tales como la valorización de la rentabilidad de una empresa sobre un determinado mercado o la creación y análisis de una cartera de acciones indexadas a estas series. Existen distintas maneras de agrupar o estimar estos índices:

1. **Índice de precios ponderados:** Se construye a través de la media aritmética simple de las acciones que lo componen. Tiene como ventaja la facilidad para obtener este indicador, sin embargo, las acciones con un precio más alto tienden a influir de mayor manera, independiente del impacto real que genera en la economía. El más clásico es el DJI y Nikkei.

2. **Índice de capitalización ponderada:** Se calcula a través de la capitalización bursátil de cada uno de los valores que forman el índice, entiendo la capitalización bursátil como la multiplicación entre la cantidad de acciones que existen en una acción por el precio de estas, siendo este el método más utilizado por los países para reflejar su sector financiero, algunos ejemplos son S&P 500 y S&P IPSA.
3. **Índices de igual ponderación:** Se obtiene como la media aritmética de los retornos de cada uno de las acciones que compone el índice. Esta forma no se utiliza con frecuencia en la actualidad debido a la cantidad de ajustes que hay que realizar y tienen mayor influencia los índices con menor capitalización bursatil, algunos ejemplos son, FT 30 y el Value Line Composite Average.

3.2. Auto-similitud

Para entender la Auto-similitud o Auto-Semejanza hay que primero entender que es un Fractal, termino acuñado por [Mandelbrot \(1967\)](#) en el artículo científico titulado "¿Cuanto mide la costa de Gran Bretaña?" en donde introduce la geometría fractal para explicar una metodología para medir la costa del país. El término fractal hace referencia a una figura que tiene la propiedad de Auto-similitud la cual explica que los objetos están conformados de copias más pequeñas de si misma, es decir, si observamos una figura se aprecia una réplica pequeña de si misma a cualquier escala de dimensión con una cierta especie de recursividad. En la **Figura 3.1** se muestra el Helecho de Barnsley que lo propuso para describir un ejemplo básico de la geometría fractal, como se puede observar las hojas del helecho tiene replicas pequeñas de si misma, compuesta por hojas de la misma estructura a lo largo de esta.

La Auto-similitud se puede presentar de 4 maneras:

- Auto-similitud exacta (estricta): Hace referencia cuando la figura se repite una o varias partes de manera exacta. Esta permita la auto-repetición sucesiva o infinita de la figura al realizar una aproximación o acercamiento.
- Auto-similitud aproximada: conformada por la repetición parcial de las figuras.



Figura 3.1: Helecho de Barnsley.

Usualmente se encuentran en la naturaleza y solo una parte de ellas se repite dentro del todo.

- Auto-similitud estadística: Como dice su nombre hace referencia cuando los objetos a distintas escalas tienen a repetir sus medidas estadísticas.
- Auto-afinidad: Se da cuando en las figuras en distintas escalas tienen una cierta afinidad. En las series de tiempo se dan con mayor frecuencia.

La Auto-Similitud en las series financieras introducida por [Peters \(1994\)](#) quien plantea la Hipótesis de Mercados Fractales declarando que las series de precios de acciones no siempre siguen un comportamiento aleatorio (*Random Walk*), sino que en realidad tienen comportamientos fractales dado que muestran una estructura similar (auto-afinidad) cuando están en diferentes intervalos de tiempo trabajando bajo la premisa que la historia de estos comportamientos se repiten. Estos comportamientos se pueden ver reflejados en la **Figura 3.2** que muestra los precios diarios del Índice DAX para un determinado periodo de tiempo y la **Figura 3.3** que muestra los precios cada 1 hora en un determinado periodo de tiempo, es fácil notar que existe un patrón con una auto-afinidad entre estas dos gráficas, o series de precios. Se observa que los precios se encuentran estables en subidas y bajadas controladas en un determinado periodo de tiempo y luego tienen una caída prominente con una recuperación del mismo nivel. Este comportamiento semejante se da en distintos

periodos de tiempo y hace evidencia del efecto de Auto-similitud presente en este Índice.



Figura 3.2: Precios diarios del Índice DAX.

Fuente: Web TradingView

<https://www.tradingview.com/chart/o7aD2Kie/?symbol=EURONEXT%3ADAX>



Figura 3.3: Precios cada 1 hora del Índice DAX.

Fuente: Web TradingView

<https://www.tradingview.com/chart/o7aD2Kie/?symbol=EURONEXT%3ADAX>

3.3. Inteligencia Artificial

La inteligencia artificial (IA) fue introducida en los 1950s cuando un puñado de pioneros de la ciencia de la computación partieron preguntándose si los computadores se les podría

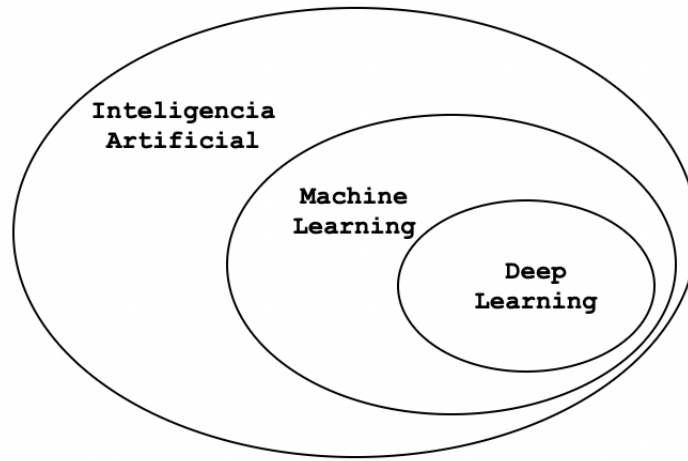


Figura 3.4: Inteligencia Artificial, Machine Learning y Deep Learning.

Fuente: Elaboración Propia.

hacer "pensar una pregunta que hasta el día de hoy nos hacemos. Una definición concisa para este campo es *El esfuerzo por automatizar las tareas intelectuales normalmente hechas por los humanos*. declarada por [Chollet \(2018\)](#). IA es un campo general que se compone del Machine Learning (ML) y el Deep Learning (DL), pero también incluye muchos otros enfoques que no involucran ningún aprendizaje. El caso de los programas de ajedrez, por ejemplo, solo envuelve una dura serie de códigos desarrollados por expertos en la materia que creen que la inteligencia artificial es explicitar todas las reglas que describen la inteligencia humana manipulando el conocimiento. Este enfoque se conoce como *Symbolic AI* y fue un paradigma que dominó entre los 1950s y 1980s. Si bien estos problemas son excelentes para representar problemas bien definidos como el ajedrez, para algunos otros problemas, tales como, *fuzzy problems*, clasificación de imagen, *speech recognition* y problemas de traducción de lenguaje. Para estos existe un nuevo enfoque llamado *machine learning* y *deep learning*. La **Figura 3.4** muestra una visión general y estructural de la relación existente entre estos conceptos.

3.3.1. Machine Learning

Para entender el machine learning primero hay que comprender el concepto de *general-purpose computation* que significa usar de una forma las operaciones mecánicas para automatizar la computación desde el campo del análisis matemático, acuñando el nombre

de *Analytical Engine*. En 1843, Ada Lovelace remarcando en la invención, “La Ingeniería Analítica no tiene pretensiones de ninguna cosa para originar nada. Esta puede hacer lo que sea que se quiera como se le ordene a realizar... Su esquema es ayudarnos a poner a disposición lo que ya conocemos”. Esta cita fue luego mencionada por el pionero de la inteligencia artificial por **Turning (1950)** como “La objeción de Lady Lovelace’s”. En su trabajo “Computing Machinery and Intellince” introduciendo el concepto de *Turing test* también como conceptos que han formado a la inteligencia artificial. Machine Learning surge de las preguntas: ¿Puede un computador ir mas allá de lo "que conocemos de ordenar para realizar? aprender en si mismas como desarrollar tareas específicas?. Más allá de programar las reglas, ¿Podrán los computadores automáticamente aprender las reglas mirando la información?. Esta pregunta abre la puerta a un nuevo paradigma de la programación. En la programación clásica, el paradigma de *Symbolic AI* los humanos imputan las reglas(a program) y la información para ser procesada acorde a estas reglas y de ahí provienen las respuestas. Con el enfoque de machine learning, los humanos imputan la información y/o datos, así como también las respuestas esperadas a esos datos y de ahí se definen las reglas. Estas reglas pueden ser aplicadas a nueva información para producir nuevas respuestas. Un sistema de machine learning es entrenado en lugar de un programa explícito. Por ejemplo, si se quiere automatizar la tarea de clasificar las imágenes de tus vacaciones, se podría realizar un modelo de machine-learning con muchos ejemplos de imágenes ya clasificadas por humanos y el sistema aprendería reglas estadísticas/matemáticas para la asociar las imágenes a su respectiva clasificación.

3.3.2. Deep Learning

Para definir que es el deep learning y entender la diferencia con machine-learning, primero se necesita tener una idea de que hacen los algoritmos de ML. Hay tres cosas que los modelos de ML necesitan.

- Observaciones de entrada (Inputs): Por ejemplo para reconocimiento de imágenes, se necesitan un banco de imágenes. Para predicción de series de tiempo, se necesita la información histórica de estas series (precios, rentabilidades, indicadores).

- Resultados o predicciones esperadas (Outputs): En clasificación de imagen, podría ser el nombre o clase de la imagen. Para las series de tiempo podría ser el valor futuro del precio, rentabilidad o volatilidad.
- Una forma de medir si el algoritmo esta realizando un buen trabajo: Esto es necesario con el fin de determinar la distancia entre la respuesta del algoritmo y la respuesta esperada, estas son llamadas funciones de perdida. Además a este proceso dentro del machine-learning se le llama "aprendizaje".

Por lo tanto, un modelo de machine learning transforma los datos de entrada en datos significativos de salida. Este proceso se aprende a través de la exposición de conocidos ejemplos de entradas y salidas. Por ende, el problema central del machine-learning y el deep-learning es aprender útiles representaciones de la data de entrada a mano, para intentar entregar un dato cercano de salida a un requerimiento no conocido. Así es como la definición de *aprendizaje* en el contexto de machine learning, describe un búsqueda automatizada para realizar las mejores representaciones.

Ahora que se entiende a que se refiere con aprendizaje, se puede entender a que viene el termino "deep"(profundo) en este sub-campo del machine learning. En el machine learning tenemos una *capa* básica de aprendizaje que permite realizar la representación de estas reglas y tratamiento de la información. En el caso del deep learning se tiene una serie de capas complejas que se van apilando para ir procesando y representando la información. En el aprendizaje profundo estas representaciones de capas son en su mayoría por no decir siempre, modelos llamados *neural networks* que son estructuralmente capas apiladas una sobre otra.

3.4. Arquitecturas de redes neuronales

Existen distintos tipos de capas que poseen a su vez una serie de configuraciones específicas con lógicas estadísticas/matemáticas que se han ido representando para resolver distintos tipos de problemas, estas configuraciones son conocidas como los tipos de arquitectura presentes dentro del deep-learning. En esta investigación se utilizan las tres más mencionadas y trabaja dentro del ámbito de la inteligencia artificial para la predicción de series de tiempo.

3.4.1. MLP

Una Artificial Neural Network (ANN) es una de los algoritmos de machine learning diseñados para simular la estructura neuronal dentro del cerebro. Estas neuronas tienen la habilidad para conectar las neuronas artificiales unas con otras, adquiriendo la habilidad de problema-solución, ajustando sus conexiones a través de una tasa de aprendizaje. Una Multi-Layer Perceptron (MLP) es una red deep feedforward (DFN) como un modelo representativo del ámbito del deep learning. Estos modelos calculan la capa de salida desde una capa de entrada utilizando las salidas de las capas anteriores, conocidas como capas ocultas, como vector de entrada para la capa actual. Este algoritmo es conocido para distintas aplicaciones, tales como problema de clasificación y detección de patrones, así como también es conocida como en método prominente para la predicción de series financieras. (Aras and Kocakoç (2016); Patel, Shah, Thakkar, and Kotecha (2015b)). La Figura 3.5 muestra un ejemplo de ilustración de esta configuración descrita, donde tiene una Capa de Entrada, dos Capas Ocultas y una Capa de Salida con solo un perceptron o neurona. Las flechas representan los pesos que conectan estas neuronas unas con otras.

Estas relaciones son representadas por la siguiente ecuación matemática:

$$y_t = w_0 + \sum_{j=1}^q w_j f \left(w_{0j} + \sum_{i=1}^p w_{ij} y_{t-i} \right) + \epsilon_t \quad (3.1)$$

Donde y_t es la variable dependiente a predecir, p y q son los números de entradas y las cantidad de capas ocultas, $f()$ denota la función de activación, w_{ij} y w_j son los pesos de

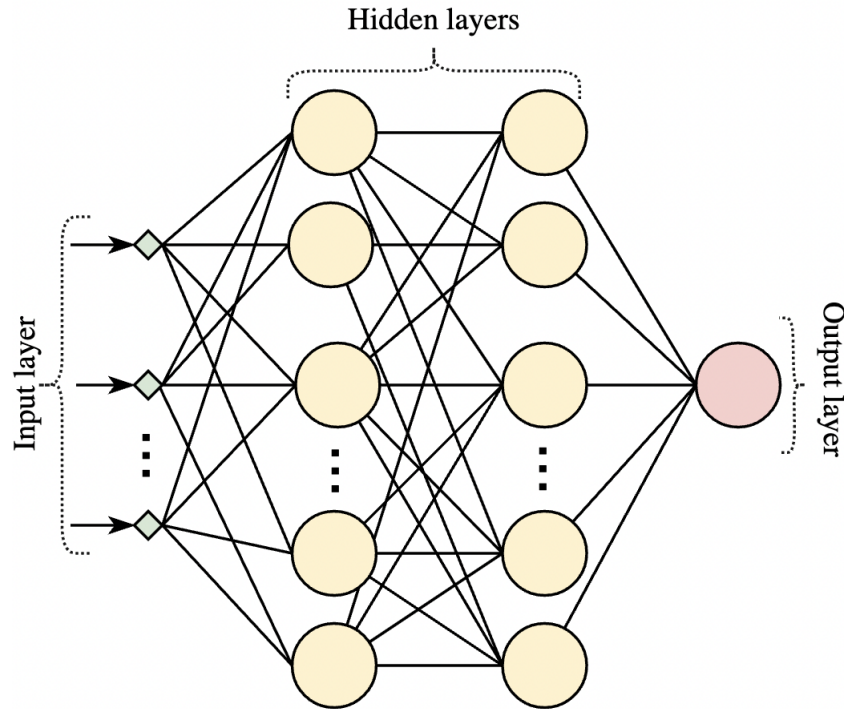


Figura 3.5: Diagrama de ejemplo de arquitectura de una MLP.

Fuente: Elaboración Propia.

conexión de las capas de entrada y capas ocultas.

3.4.2. SimpleRNN

Las Redes Neuronales Recurrentes (RNN) han sido ampliamente probadas que funcionan para problemas relacionados con una lógica de secuencia, tales como, secuencia de palabras en una oración o párrafo, secuencias de reconocimiento de voz y predicción de series de tiempo (e.g. Yamada, Murata, Arie, and Ogata (2017) y Abd El-Moneim, Nassar, Dessouky, Ismail, El-Fishawy, and Abd El-Samie (2020)). A diferencia de las redes neuronales Feed-forward que no tienen memoria de los valores de entradas que esta recibe las RNN resuelven este problema, recibiendo el pasado y el presente de la información utilizando un ciclo a través de un bucle . En concreto, las RNN usan la información de salida anterior para ser usadas como entrada mientras tiene estados ocultos. La Figura 3.6 explica un diagrama desenrollado de la estructura de una red neuronal recurrente donde es posible evidenciar la arquitectura de estas redes explicadas recientemente.

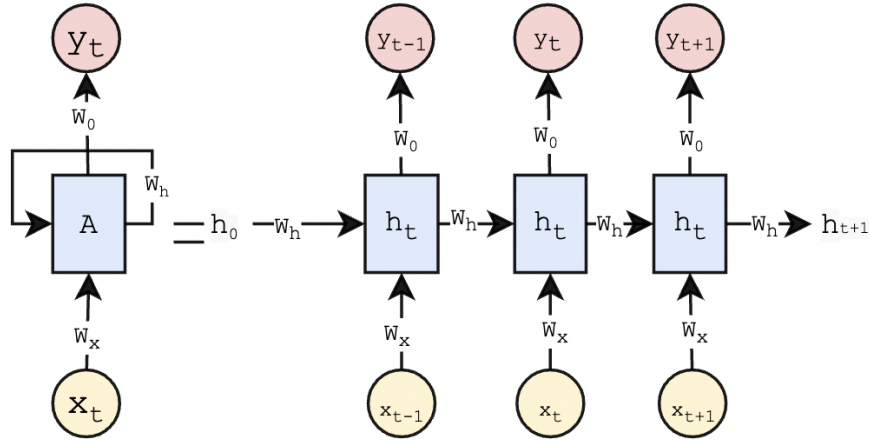


Figura 3.6: Unrolled diagram of a recurrent neural network architecture.

Fuente: Elaboración Propia.

Las siguientes ecuaciones matemáticas muestran la relación existente dentro de esta red:

$$h_t = f_1(W_h h_{t-1} + W_x x_t + b_h) \quad (3.2)$$

$$y_t = f_2(W_0 h_t + b_0) \quad (3.3)$$

Donde h_t es el estado en el tiempo t , y_t es el estado de salida en el tiempo t , f_1 y f_2 son las funciones de activación, W_h es el peso del estado h en la anterior capa oculta, W_x es el peso de la actual capa en el actual estado de entrada, W_0 es el peso en el estado de salida, b_h y b_0 son los coeficientes de temporalidad.

3.4.3. LSTM

Hochreiter and Schmidhuber (1997) introdujo las redes Long Short Term Memory (LSTM) como una solución al *vanishing gradient problem* (Hochreiter (1998)) en una red neuronal recurrente RNN. Las redes LSTM evita los problemas de dependencias a largo plazo agregando 3 puertas a la arquitectura RNN, una puerta de entrada, una puerta de olvido y una puerta de salida, usando una celda de memoria en lugar de las neuronas

explicadas en la **Figura 3.7** explicada principalmente por la puertas de entrada, salida, modulación y olvido. Las flechas son la ilustración del transporte de la información en esa dirección. El símbolo $\otimes$ muestra la multiplicación de cada vector de saldad. La Ecuación 3.4 explica la puerta de entrada i_t que determina qué información debe ingresar al estado de la celda, esta puerta es una función sigmoide y tiene un rango de $[0,1]$ simplemente agrega memoria y no puede eliminarse /olvidar la memoria. La Ecuación 3.5 explica que la puerta de modulación de entrada $\tilde{c}_t$ tiene una función de activación $\tanh$ debido a que el rango de esta función es $[-1,1]$ y permite que el estado de la celda olvide la memoria.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (3.4)$$

$$\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (3.5)$$

Donde t es la observación en ese momento, h_{t-1} es el último estado oculto, este valor de entrada y x_t que es el vector de entrada acceden a la puerta olvidada f_t explicada en Ecuación 3.6, esto se considera como el vector de recuerdo, la salida de esta puerta cuya información si es olvidada se multiplica por 0 o la información se mantiene en el estado de celda si es 1. Ecuación 3.7 explica la característica principal de la celda de memoria, que es el estado de celda c_t considerado como la memoria a largo plazo que tiene flechas en bucle que indican la naturaleza recursiva de la celda, esto permite almacenar la información anterior y mantener la lógica de secuencia de una RNN. Finalmente, toda esta información pasa al vector de enfoque generalmente llamado Output Gate o_t explicado en Ecuación 3.8 centrándose en qué información tiene que pasar al estado oculto h_t , donde este estado se multiplica por una función $\tanh$ para filtrar la última información para pasar a la siguiente celda de memoria mostrada en Ecuación 3.9.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (3.6)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \quad (3.7)$$

$$o_t = \sigma(W_o[h_t - 1, x_t] + b_o) \quad (3.8)$$

$$h_t = o_t \cdot \tanh(c_t) \quad (3.9)$$

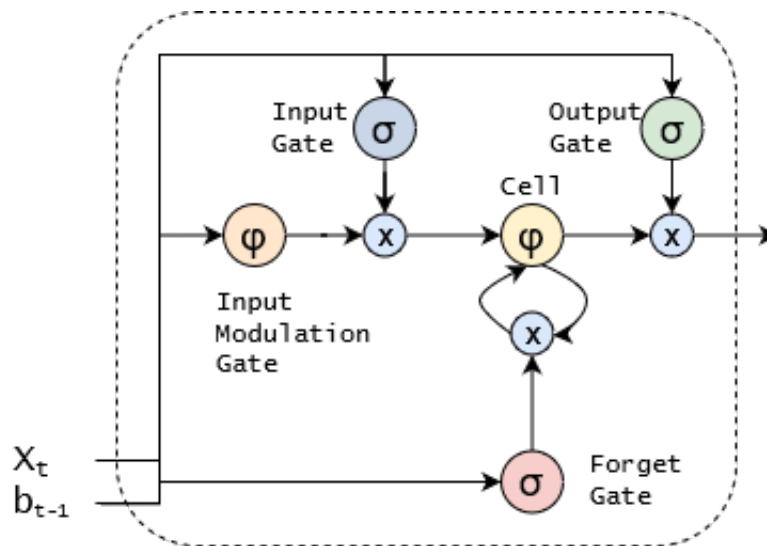


Figura 3.7: Una ilustración gráfica de una celda LSTM.
. **Fuente:** Elaboración Propia.

4 Metodología

En esta sección se explica los datos utilizados, el marco teórico de los modelos propuestos, la metodología presentada de Auto-Similitud y los experimentos realizados para validar las hipótesis presentadas.

4.1. Obtención de datos

4.1.1. Índices bursátiles

El conjunto de datos consiste en los índices bursátiles de cuatro países, cada índice es considerado como un reflejo de cada mercado accionario del país.

S&P 500¹ para Estados Unidos, el índice contiene las principales 500 compañías en el mercado las cuales representan aproximadamente el 80 % de las capitalización del mercado disponible. En la Figura 4.1 se muestra el gráfico de los precios históricos que se utilizan en este estudio. **DAX**² es un índice bursátil *blue chip* el cual incluye las 30 principales compañías y abarca aproximadamente el 80 % de las capitalización de mercado disponible. En la Figura 4.2 se aprecia la variación histórica de los precios durante el año 2000 y 2020. **SMI**³ el índice *blue chip* para Suiza que cubre alrededor del 80 % del mercado de valores suizo e incluye las 20 compañías mas grandes y más liquidas. En la Figura 4.3 se observa la variación histórica de los precios de este índice del año 2000 al 2020. **AEX**⁴ es el índice bursatil de Amsterdam el cual ofrece la agrupación de las principales 25 compañías

¹<https://www.spglobal.com/spdji/en/indices/equity/sp-500/>

²<https://www.dax-indices.com/index-details?isin=DE0008469008>

³<https://www.six-group.com/en/products-services/financial-information/indices/swiss-indices/smi.html>

⁴<https://live.euronext.com/en/product/indices/NL0000000107-XAMS>

de mayor frecuencia de transacciones de compra/venta en el mercado Holandés. En la Figura 4.4 se aprecia la variación histórica de los precios desde el año 2000 al año 2020 utilizados en este estudio. Para cada índice mencionado se incluyen 4 tipos de conjuntos de datos con frecuencias distintas para los precios de transacción. Estos son

- I. Conjunto de datos con precios diarios, el cual contiene 5,284 días de transacción, las observaciones van del periodo de Enero del 2000 a diciembre del 2020.
- II. Conjunto de datos con precios cada hora (60 minutos) que contiene 17,475 observaciones que van desde Enero del 2017 a Diciembre del 2019.
- III. Conjunto de datos cada media hora (30 minutos) con 34,949 muestras desde Enero del 2017 a Diciembre del 2019.
- IV. Conjunto de datos cada 15 minutos que contiene 69,897 observaciones que van desde el Enero 2017 a Diciembre 2019.

El conjunto de datos de frecuencia diaria para todos los índices bursátiles es obtenido desde el sitio web de Yahoo Finance y los conjunto de datos de frecuencia intradiaria son obtenidos del sitio web de DukastCopy.

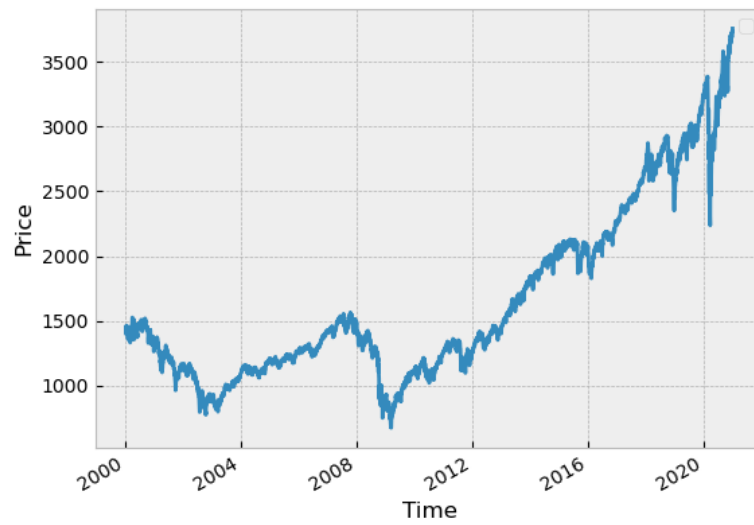


Figura 4.1: S&P500 - Currency in \$ USD.

Fuente: Elaboración Propia.

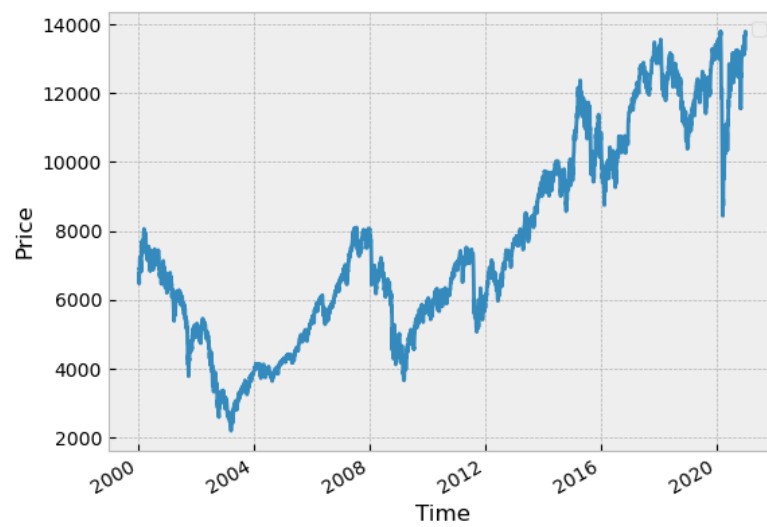


Figura 4.2: DAX - Currency in \$EUR.

Fuente: Elaboración Propia.

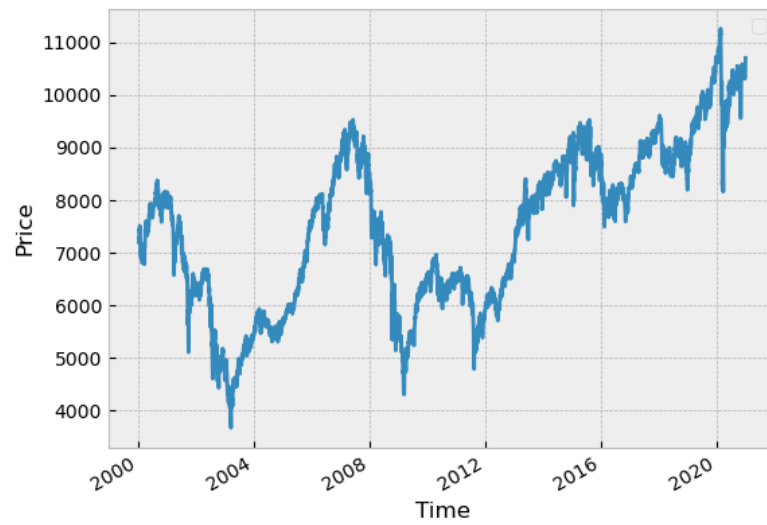


Figura 4.3: SMI Currency in \$CHF.

Fuente: Elaboración Propia.

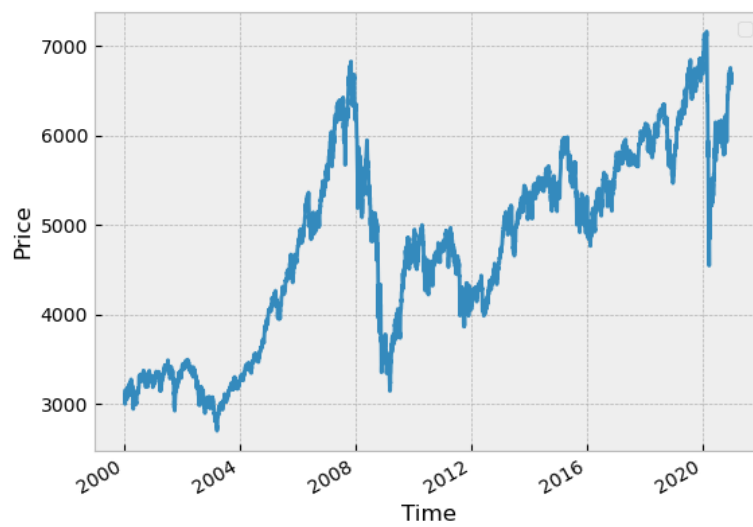


Figura 4.4: AEX - Currency in \$CHF.

Fuente: Elaboración Propia.

4.2. Estadísticas descriptivas

En esta sección se muestran las estadísticas de las series utilizadas en los modelos presentados para predecir el comportamiento de los 4 índices bursátiles estudiados en esta investigación, analizando los precios y rentabilidades en cada una de sus frecuencias (1 día, 60 minutos, 30 minutos, 15 minutos). En la **Figura 4.5** se observa la cantidad de datos por conjunto de datos utilizado para entrenar las redes, es considerable notar la gran diferencia en la cantidad de observaciones entre los datos diarios vs. los datos minútales, los datos diarios tienen 5,284, cada 1 hora son 17,475 observaciones, cada 30 minutos contiene 34,949 observaciones y cada 15 minutos son 69,897 observaciones. Este gráfico entrega la una comparación visual de la importancia y beneficios que puede traer usar los datos intradiarios para el entrenamiento de redes neuronales debido a la gran cantidad de información que se tiene de estos.

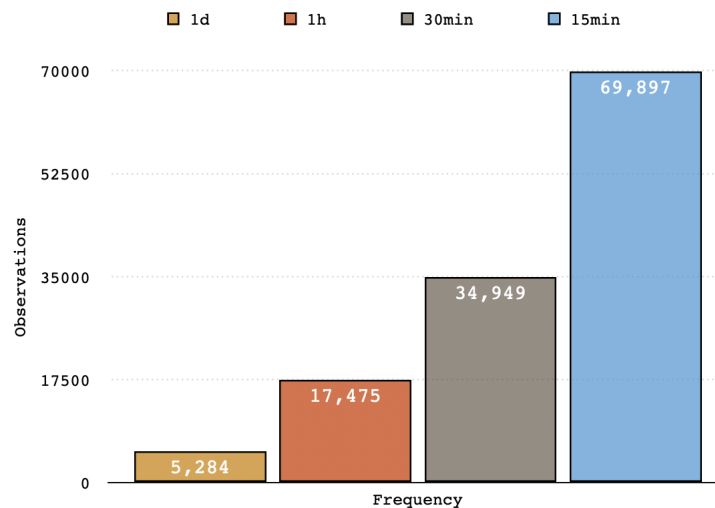


Figura 4.5: Cantidad de observaciones por tipo de conjunto de datos.

Fuente: Elaboración Propia.

Respecto a las estadísticas descriptivas mostradas en la **Tabla 4.1** del índice S&P 500, la serie de precio diarios posee una media de 1653,67 con una una distribución no normal, platicurtica y con tendencia a la derecha. Las series de precios de valores intradiarios tienen comportamientos parecidos para la media, desviación estándar, mediana, máximo y mínimo, así mismo para la curtosis y asimetría, dejando en evidencia que para el rango

de tiempo utilizado sus distribuciones no siguen un comportamiento normal según JB, de comportamiento platocurtica y con un distribución asimétrica positiva o hacia la derecha. Respecto a las rentabilidades diarias se tiene una media de 0,000179 con una distribución no normal, leptocurtica y asimétrica a la izquierda. Para las rentabilidades cada 15 min tiene una media de 0,000002 con distribución no normal, leptocurtica y asimétrica a la izquierda. Para las rentabilidades cada 30 min tiene una media de 0,000004 con distribución leptocurtica y asimetría a la izquierda y cada 60 min tienen una media de 0,000008 con distribución no normal por JB, leptocurtica y asimétrica a la izquierda.

Tabla 4.1: Estadísticas descriptivas del Índice S&P 500.

SP500	Price				Returns			
	1 day	15 min	30 min	60 min	1 day	15 min	30 min	60 min
mean	1653.67	2848.49	2848.50	2848.53	0.000179	0.000002	0.000004	0.000008
std	674.39	178.21	178.22	178.24	0.012553	0.003275	0.004629	0.006551
min	676.53	2325.47	2325.97	2326.47	-0.127652	-0.195002	-0.195002	-0.195002
25 %	1163.55	2724.27	2724.27	2724.47	-0.004766	-0.000103	-0.000165	-0.000241
50 %	1386.98	2830.77	2830.77	2830.77	0.000594	0.000000	0.000000	0.000000
75 %	2071.32	2945.77	2945.77	2945.77	0.005805	0.000148	0.000184	0.000344
max	3756.07	3336.75	3335.86	3335.86	0.109572	0.098720	0.098443	0.098616
Skewness	1.03	0.47	0.47	0.47	-0.390000	-17.870000	-12.610000	-8.880000
Kurtosis	0.07	0.23	0.23	0.23	10.94	1106.21	550.65	272.74
JB	940.89***	2720.84***	1360.24***	680.71***	26500.20***	3567590604.80***	442463660.44***	54393148.38***

Respecto a las estadísticas descriptivas mostradas en la **Tabla 4.2** del índice DAX, la serie de precio diarios posee una media de 7661,46 con una una distribución no normal, platocurtica y con tendencia a la izquierda. Las series de precios de valores intradiarios sus estadísticas descriptivas son de valores similares donde sus distribuciones no siguen un comportamiento normal según JB, de comportamiento platocurtica y con un distribución asimétrica negativa o hacia la izquierda. Respecto a las rentabilidades diarias se tiene una media de 0,000133 con una distribución no normal, leptocurtica y asimétrica a la izquierda. Para las rentabilidades cada 15 min tiene una media de 0,000001 con distribución no normal, leptocurtica y asimétrica a la izquierda. Para las rentabilidades cada 30 minutos tiene una media de 0,000001 con distribución leptocurtica y asimétrica a la izquierda y cada 60 minutos tienen una media de 0,000003 con distribución no normal por JB, leptocurtica y asimétrica a la izquierda.

Respecto a las estadísticas descriptivas mostradas en la **Tabla 4.3** del índice AEX, la serie de precio diarios posee una media de 435,65 con una una distribución no normal,

Tabla 4.2: Estadísticas descriptivas del Índice DAX.

	Price				Returns			
DAX	1 day	15 min	30 min	60 min	1 day	15 min	30 min	60 min
mean	7661.46	12201.57	12201.52	12201.39	0.000133	0.000001	0.000001	0.000003
std	3006.77	685.45	685.33	685.13	0.014921	0.003660	0.005174	0.007311
min	2202.96	10298.71	10298.71	10298.71	-0.130549	-0.138037	-0.138037	-0.138037
25 %	5365.42	11637.71	11637.71	11637.71	-0.006661	-0.000087	-0.000122	-0.000211
50 %	6971.07	12306.90	12306.71	12305.58	0.000759	0.000000	0.000000	0.000000
75 %	10166.04	12698.99	12698.83	12698.90	0.007460	0.000104	0.000175	0.000301
max	13790.29	13586.28	13584.48	13584.48	0.107975	0.164649	0.164649	0.164649
Skewness	-0.38	-0.47	-0.47	-0.47	-0.170000	-1.200000	0.880000	0.650000
Kurtosis	-1.02	-0.44	-0.44	-0.44	5.74	711.90	354.19	175.65
JB	363.24***	3208.47***	1603.26***	800.82***	7348.41***	1476731200.53***	182835116.65***	22502141.45***

platicurtica y con tendencia a la derecha. Las series de precios de valores intradiarios sus estadísticas descriptivas son de valores similares donde sus distribuciones no siguen un comportamiento normal según JB, de comportamiento platicurtica y con una distribución asimétrica negativa o hacia la izquierda. Respecto a las rentabilidades diarias se tiene una media de $-0,000015$ con una distribución no normal, leptocurtica y asimétrica a la derecha. Para las rentabilidades cada 15 minutos tiene una media de $0,000002$ con distribución no normal, leptocurtica y asimétrica a la derecha. Para las rentabilidades cada 30 minutos tiene una media de $0,000004$ con distribución leptocurtica y asimétrica a la derecha y cada 60 minutos tienen una media de $0,000008$ con distribución no normal por JB, leptocurtica y asimétrica a la derecha.

Tabla 4.3: Estadísticas descriptivas del Índice AEX.

	Price				Returns			
AEX	1 day	15 min	30 min	60 min	1 day	15 min	30 min	60 min
mean	435.65	496.57	496.57	496.57	-0.000015	0.000002	0.000004	0.000008
std	108.93	54.86	54.86	54.86	0.014002	0.004612	0.006520	0.009211
min	199.25	433.37	433.37	433.37	-0.113758	-0.233457	-0.233457	-0.233457
25 %	340.22	433.37	433.37	433.37	-0.006082	0.000000	0.000000	0.000000
50 %	431.51	516.43	516.43	516.43	0.000510	0.000000	0.000000	0.000000
75 %	522.56	548.74	548.74	548.74	0.006561	0.000000	0.000000	0.000000
max	701.56	576.70	576.70	576.70	0.100283	0.233457	0.233457	0.233457
Skewness	0.26	-0.12	-0.12	-0.12	-0.217500	1.871000	1.311000	0.858000
Kurtosis	0.78	-1.73	-1.73	-1.73	7.32	1846.19	922.30	459.46
JB	197.98***	8902.29***	4451.29***	2225.73*	12018.23*	9926487769.20***	1238688304.76***	153704944.01***

Respecto a las estadísticas descriptivas mostradas en la **Tabla 4.4** del índice SMI, la serie de precio diarios posee una media de $7466,25$ con una una distribución no normal, platicurtica y con tendencia a la izquierda. Las series de precios de valores intradiarios sus estadísticas descriptivas son de valores similares donde sus distribuciones no siguen un

comportamiento normal según JB, de comportamiento platocurtica y con una distribución asimétrica negativa o hacia la izquierda. Respecto a las rentabilidades diarias se tiene una media de 0,000073 con una distribución no normal, leptocurtica y asimétrica a la izquierda. Para las rentabilidades cada 15 minutos tiene una media de 0,000001 con distribución no normal, leptocurtica y asimétrica a la izquierda. Para las rentabilidades cada 30 minutos tiene una media de 0,000002 con distribución leptocurtica y asimétrica a la izquierda y cada 60 minutos tienen una media de 0,000004 con distribución no normal por JB, leptocurtica y asimétrica a la izquierda.

Tabla 4.4: Estadísticas descriptivas del Índice SMI.

SMI	Price				Returns			
	1 day	15 min	30 min	60 min	1 day	15 min	30 min	60 min
mean	7466.25	8907.56	8907.56	8907.55	0.000073	0.000001	0.000002	0.000004
std	1540.92	291.52	291.50	291.50	0.011622	0.002568	0.003631	0.005136
min	3675.40	8143.97	8147.97	8147.97	-0.101339	-0.101988	-0.101988	-0.101988
25 %	6214.13	8683.97	8683.97	8683.97	-0.005319	0.000000	0.000000	0.000000
50 %	7641.02	8934.97	8934.97	8933.97	0.000492	0.000000	0.000000	0.000000
75 %	8734.32	9082.97	9082.97	9082.97	0.005848	0.000000	0.000000	0.000000
max	11263.01	9609.97	9607.97	9607.97	0.107876	0.119145	0.119145	0.119145
Skewness	-0.02	-0.04	-0.04	-0.04	-0.261810	-2.701410	-1.927660	-1.369800
Kurtosis	-0.92	-0.37	-0.37	-0.37	7.65	611.11	304.43	150.71
JB	188.04***	424.93***	212.09***	106.00***	12902.97***	1091474805.35***	135441971.58***	16600208.67***

4.3. Estructura de los modelos

Se definen 3 modelos bases utilizando las arquitecturas D-MLP, D-SimpleRNN y D-LSTM entreandas solo con precios diarios. Luego se presentan 3 modelos propuestos con el enfoque de Auto-similitud. El primer modelo es mezclando datos diarios con datos cada una hora llamado 60MD-Model, el segundo es mezclando la data diaria con data cada media hora llamado 30MD-Model y mezclando datos diarios con datos cada 15 minutos llamado 15MD-Model, donde Model es la arquitectura mencionada anteriormente, por lo que para cada arquitectura se va a presentar y estudiar estos 3 modelos propuestos.

4.3.1. Configuración de los vectores de entrada

Para comparar los modelos estudiados en esta investigación, se usan las mismas variables de entrada para cada red neuronal analizada (MLP, Simple RNN y LSTM)

Para las redes neuronales RNN y LSTM los vectores de entradas formados por una capa de entrada donde **Ecuación 4.1** es la matriz F como las características de entradas usadas para entrenar los modelos incluyendo solo 4 series: Los precios de cierre P_t en el periodo t , el rezago de P_t en el periodo

$$F = \begin{bmatrix} P_t & P_{t-1} & R_t & R_{t-1} \\ P_{t-2} & P_{t-3} & R_{t-2} & R_{t-3} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ P_{t-s} & P_{t-s-1} & R_{t-s} & R_{t-s-1} \end{bmatrix} \quad (4.1)$$

$$R_t = \ln(P_t) - \ln(P_{t-1}) \quad (4.2)$$

Donde R_t is el retorno logaritmico en el tiempo t dado por la Ecuación 4.2 y P_t es el precio en el momento t .

Para el modelo MLP debido al tipo de arquitectura posee una restricción en la capa de entrada por lo que se propone “aplanar” la forma de la matriz F . Sin embargo la cantidad de información es la misma que en cantidad de variables y rango que para la red LSTM.

Las observaciones de los modelos son transformadas entre $[0,1]$ usando la función MinMaxScaler explicada en la Ecuación 4.3.

$$x_{scaled} = \frac{x - \min(X)}{\max(X) - \min(X)} \quad (4.3)$$

Donde x es la observación y X es el vector completo de información utilizado en la red neuronal.

4.3.2. Configuración de los modelos

Para los modelos analizados la **Figura 4.6** muestra la configuración general confeccionada por capas, la cual está dada por una capa de Entrada usando el vector de entrada de matriz F , seguida de una Capa de Modelo que puede ser arquitectura MLP, SimpleRNN o LSTM dependiendo del modelo presentado, después de cada capa del modelo, se usa una capa *Dropout* al 20 % para evitar el sobreajuste (Srivastava, Hinton, Krizhevsky, Sutskever, and Salakhutdinov (2014)). La tercera parte del modelo incluye una Capa Densa con 10 unidades de celdas. Finalmente, todo el modelo se configura con un tamaño de lote de 128, optimizado con Adam Optimizer (Kingma and Ba (2015)) con una tasa de aprendizaje de 0,001 con 100 épocas.

En concreto, la estructura de los modelos MLP está dada por 1 o 2 capas densas ocultas con diferente cantidad de nodos para la capa Modelo. La función de activación es una función *ReLU* para capas densas. Los modelos SimpleRNN están dados por 1 o 2 capas SimpleRNN con diferente configuración en los números de celda para la capa del modelo, con *tanh* como función de activación. Los modelos LSTM tienen 1 o 2 capas LSTM con diferente configuración en el número de celdas, estas capas tienen una función de activación de *tanh*.

La **Figura 4.7** muestra el tamaño de tiempo del conjunto de datos de entrenamiento,

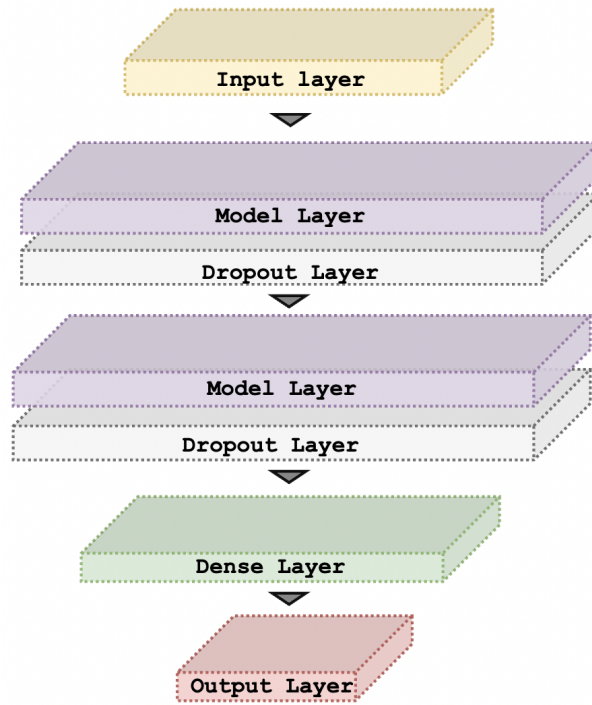


Figura 4.6: Neural networks model structure diagram.

Fuente: Elaboración Propia.

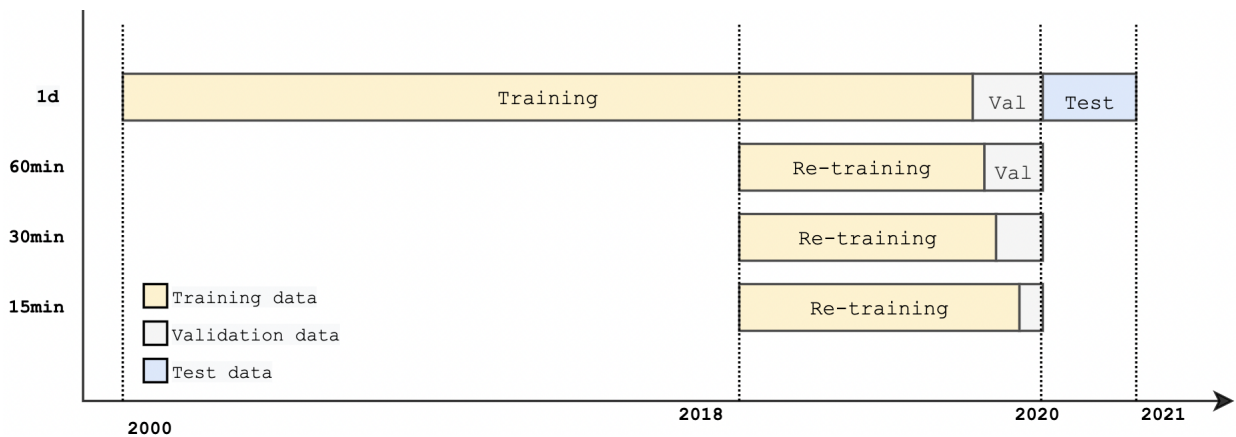


Figura 4.7: Time size for training, validation, and training data for each frequency data set.

Fuente: Elaboración Propia.

re-entrenamiento, validación y prueba para cada tipo de frecuencia. El conjunto de entrenamiento se da desde el año 2000 al 2020 con precios diarios, la validación está dada por 700 observaciones en cada conjunto de datos de frecuencia. El re-entrenamiento viene dado por datos intradiarios (60 min, 30 min, 15 min) de 2018 a 2020. El conjunto de prueba

solo está configurado con precios diarios según el objetivo de este estudio

4.3.3. Enfoque de Auto-similitud

El enfoque presentado en esta investigación utiliza un re-entrenamiento basado en los datos de frecuencia minutal, de tal manera que los datos de entrada tienen la misma configuración que los datos diarios. Este se puede clarificar de mejor manera con un ejemplo, si se tiene un vector de entrada $\{X_i | i \in I : [1, 2, 3, \dots, 10]\}$ y un valor de salida de x_{11} donde I hace referencia a observaciones de frecuencia diaria, aquí el modelo es entrenado con los 10 días anteriores para predecir el valor del día 11. El modelo con el enfoque de Auto-Similitud se va a re-entrenar con un vector de entrada $\{X_j | j \in J : [1, 2, 3, \dots, 10]\}$ y un valor de salida x_{11} , donde J hace referencia a observaciones cada 1 hora, por lo tanto, el modelo sería re-entrenado con las 10 horas anteriores para predecir el valor de la hora 11. Cuando se hace referencia a un re-entrenamiento este consiste en mantener los pesos obtenidos de la red neuronal entrada con datos diarios y sobre esta realizar un re-entrenamiento que ajuste de mejor manera los pesos y así la predicción de la red. Cabe destacar, que se podría pensar que los precios diarios con los precios intradiarios no tienen una relación y no es válido utilizarlos para aumentar la cantidad de información para entrenar los modelos. Sin embargo, según la hipótesis de mercados fractales, existe una auto-afinidad que nos permite realizar decir que los comportamientos de subidas y bajadas de precios son similares dentro de horizontes de tiempos distintos y afirmar este supuesto y también así eventualmente decir que estas series servirán de ejemplo para las predicciones de datos fuera de la muestra.

En la **Figura 4.8** se representa el Enfoque de Auto-Similitud para obtener los modelos propuestos. Como se observa lo primero que hay que realizar es un pre-procesamiento de la información, dejando ambas bases de datos en las configuraciones correspondientes. Luego obtener las características de entradas que son los Precios de cierre y retornos que generan los precios. Una vez establecidos estos dos pasos, se realiza el entrenamiento y validación del primero modelo con datos diarios obteniendo el D-Model, este ya no sirve para obtener la función de pérdida de este modelo base. Luego viene la fase de aplicar el enfoque de Auto-Similitud que parte del D-Model con los pesos guardados y obtenidos

en su entramiento y sirven como base para el re-entrenamiento y re-validación con los datos intradiarios, obteniendo el MD-Model y finalmente se testea con los datos diarios obteniendo la función de perdida correspondiente.

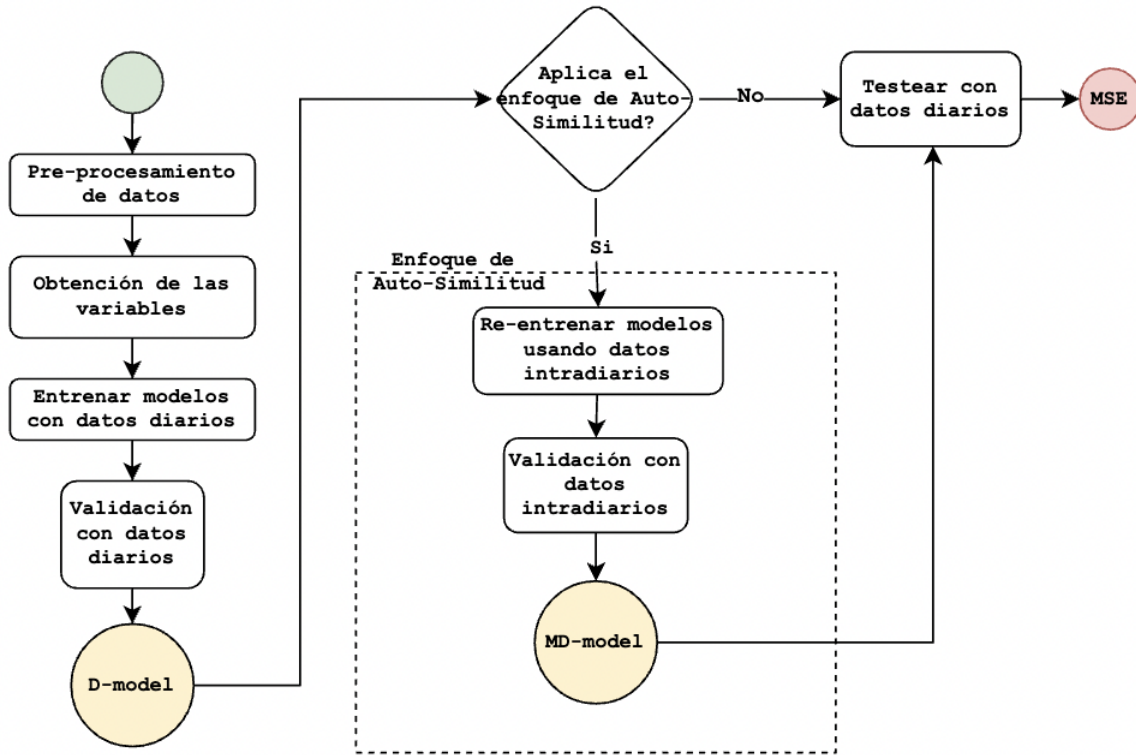


Figura 4.8: Metodología con el enfoque de Auto-Similitud.
Fuente: Elaboración Propia.

4.4. Experimentos

Los experimentos consisten en buscar la mejor configuración utilizando diferentes parámetros para entrenar, validar y testear los modelos y así poder estudiar la robustez de estos, los experimentos se centran en analizar el rendimiento de estos modelos propuestos en comparación con los modelos base. Los experimentos se constituyen en dos secciones. La primero consiste en obtener la configuración de los parámetros con la mayor precisión para cada modelo en cada índice bursátil para predecir con 1 día de antelación los precios de cierre diarios.

Se propuso realizar experimentos con distintas cantidades pasos hacia atrás, distintas cantidades de celdas o nodos y distintas cantidades de capas del modelo. La **Tabla 4.5** muestra todos los parámetros del modelo establecidos para experimentar.

Tabla 4.5: Configuraciones presentes en la investigación.

Nombre	Conjunto
Index	SP500, DAX, AEX, SMI.
Models	D-MLP, 60MD-MLP, 30MD-MLP, 15MD-MLP D-RNN, 60MD-RNN, 30MD-RNN, 15MD-RNN D-LSTM, 60MD-LSTM, 30MD-LSTM, 15MD-LSTM
Steps	5, 22, 66, 88, 126.
Cells	10, 25, 50, 100, 150, 200.
Layers	1, 2.

Para entender la distnacia entre los pronosticos se usa la función de perdida Mean Square Errors MSE (explicada en Ecuación 4.4) donde es la métrica utilizada para comparar los resultados de estos experimentos. De acuerdo a la investigación presentada por [BeratSezer et al. \(2020\)](#), la mayoría de los artículos científicos en la literatura respecto a predicción de índices bursátiles con redes neuronales utilizan las métricas MSE y RSME para medir y comparar los resultados.

$$MSE = \frac{1}{|T|} \sum_{t=0}^T (Y_t - \hat{Y}_t)^2 = \frac{1}{|T|} \sum_{t=0}^T (P_t - \hat{P}_t)^2 \quad (4.4)$$

Donde $t \in T$ es el conjunto de periodos para la validación y testeo de los datos y pronosticos. Y_t es el valor a medir, en este caso el valor real del precio de cierre de la serie financiera y $\hat{Y}$ es el pronostico del precio de cierre para medir y comparar los resultados.

4.4.1. Comparando la capacidad y superioridad predictiva

La segunda sección consiste en comparar la precisión predictiva usando Diebold-Mariano Test (DM) de los modelos base contra los modelos propuestos y la precisión de superioridad entre todos los modelos estudiados usando The Model Confidence Set (MCS) para encontrar el mejor modelo.

El Test de Diebold-Mariano (DM) presentado por [Diebold and Mariano \(1995\)](#) que comprueba significativamente si dos pronosticos fuera de la muestra (out-of-sample) son

distintos. Donde la hipotesis nula menciona que dos o mas pronosticos tienen la misma exactitud $H_0 : E(d_t) = 0 \quad \forall t$, se mencionan los errores como $\epsilon_{i,t} = \hat{y}_{i,t} - y_t$, para $i = 1, 2$ donde y_t es el valor real de la serie de financiera y $\hat{y}_{i,t}$ es el pronostico de de la serie y_t . $d_t = g(\epsilon_{1,t}) - g(\epsilon_{2,t})$ es el diferencial de las funciones de perdida $g(\cdot)$. El estadístico DM esta dado por las siguientes ecuaciones:

$$DM = \frac{\bar{d}}{\sqrt{2\pi\hat{f}_d(0)/T}} \quad (4.5)$$

$$\bar{d} = \frac{1}{T} \sum_{t=1}^T g(\epsilon_{1,t}) - g(\epsilon_{2,t}) \quad (4.6)$$

Donde $\hat{f}_d(0)$ es un estimador consistente de la función de densidad del diferencial de perdida en la frecuencia 0. El estimador esta considerado para utilizarlo en pronósticos de 1 paso hacia adelante siguiendo una distribución t Student con $T - 1$ grados de libertad.

El Model Confidence Set (MCS) propuesto por [Hansen, Lunde, and Nason \(2011\)](#) permite ordenar los modelos de acuerdo a su capacidad de predicción utilizando una secuencia de pruebas donde la hipótesis nula de igual capacidad predictiva (EPA) no se rechaza en un nivel de confianza especificado. La estadística de prueba EPA puede usar cualquier función de pérdida, en este caso de estudio, se usa la función MSE. El procedimiento MCS es un procedimiento de prueba secuencial, que consiste en eliminar el peor modelo en cada paso hasta que se acepta la hipótesis de igual capacidad predictiva para todos los modelos pertenecientes a un conjunto de modelos superiores. La selección del peor modelo está determinada por una regla de eliminación que es consistente con el estadístico de prueba.

$$e_{R,M} = \arg \max_{\rho \in M} \left\{ \sup_{\xi \in M} \frac{\bar{d}_{\rho\xi}}{\sqrt{\widehat{\text{Var}}(\bar{d}_{\rho\xi})}} \right\} \quad (4.7)$$

$$e_{\max,M} = \arg \max_{\rho \in M} \frac{\bar{d}_{\rho}}{\widehat{\text{Var}}(\bar{d}_{\rho})} \quad (4.8)$$

La representación de las reglas de eliminación está dada por $e_{R,M}$ la cuál es la expresión que considera la diferencia entre el modelo ρ y ξ , y $e_{\text{máx},M}$ es la expresión de la diferencia entre ρ y ξ sobre ξ . Donde ρ y ξ pertenecen al conjunto del modelo M . $\bar{d}$ es la función de pérdida entre dos modelos y $\widehat{\text{Var}}$ es la varianza de las estimaciones reforzadas.

Finalmente, todos los cálculos realizados en esta investigación y experimentos realizados para redes neuronales profundas se implementan utilizando Keras Environment desarrollado por [Chollet et al. \(2015\)](#) en Python 3.7.

5 Resultados

Para cada índice se realiza una tabla de comparación de los resultados obtenidos en los experimentos mostrando los mejores resultados para cada uno de ellos, el MSE con la mejor configuración, el delta de diferencia entre el modelo propuesto y el modelo base, la respectiva configuración, el Test de Diebold-Mariano para comprar las predicciones del modelo propuesto contra las del modelo base, y el Test de Model Confidence Set para comparar cuales de todos los modelos estudiados es el que tiene mayor capacidad predictiva.

En general, los mejores resultados se dan para el índice S&P500 donde presentan mejoras de al menos un 17% en los modelos propuestos. También se observa que para todos los índices la capacidad predictiva de los modelos bases es superada por algunos y/o todos los modelos propuestos, además cabe destacar de todos los experimentos el 90% de las veces los modelos propuestos superan a los modelos bases, siendo solo el 10% de las veces en que alguno de los modelos propuestos no los supera. Por otro lado, queda demostrado que los modelos propuestos con el enfoque de Auto-Similitud tienen mayor capacidad predictiva comprado con todos los modelos.

5.1. S&P 500

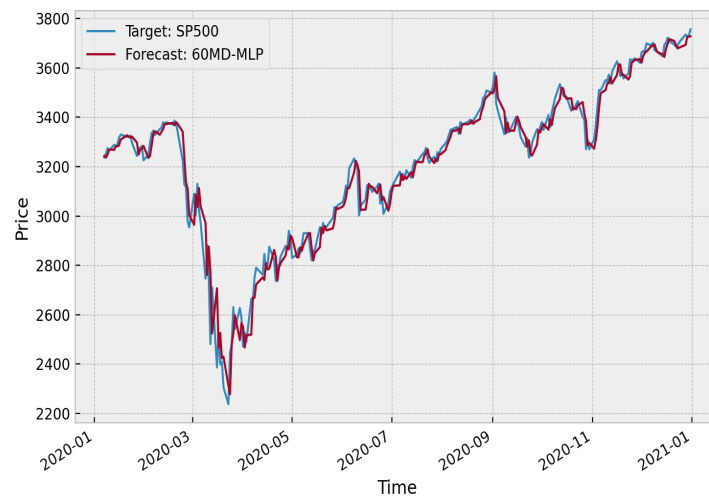
La **Tabla 5.1** muestra los resultados para el índice bursátil S&P500, se observa que para los modelos con arquitectura MLP los modelos propuestos con el enfoque de Auto-Similitud 60MD-MLP, 30MD-MLP, 15MD-MLP genera una reducción respecto al modelo base de 21,46%, 18,51% y 18,17% respectivamente, donde los pronósticos generados por estos modelos son estadísticamente distintos según el Test de DM. Para la arquitectura SimpleRNN los modelos propuestos 60MD-SimpleRNN y 15MD-SimpleRNN son estadísticamente diferentes en un 17,47% y 18,78% respectivamente reduciendo el error cuadrático medio en

esas cantidades, el modelo 30MD-SimpleRNN si bien muestra una mejoría en un 11,96 % este no es estadísticamente diferente según DM. Por otro lado, los modelos de arquitectura LSTM, los tres modelos propuestos 60MD-LSTM, 30MD-LSTM y 15MD-LSTM reducen en un 23,14 %, 21,06 % y 23,14 % respectivamente el MSE, según el test de DM los pronósticos de los modelos 30MD-LSTM y 15MD-LSTM son estadísticamente distintos. Respecto a las configuraciones obtenidas, se observa que para el caso del modelo MLP los mejores modelos son con 5 back steps (i.e. una semana de información) siendo esta la configuración con menos información entregada en los experimentos, respecto a los modelos SimpleRNN solo un modelo no tiene la mejor configuración con 22 steps (un mes de información), también se observa que la configuración 1 capa, 100 celdas y 22 steps se repite para los modelos D-SimpleRNN, 30MD-SimpleRNN y 15MD-SimpleRNN, mostrando que incluso para el mejor modelo base los modelos propuestos en la misma configuración tienen rendimientos mejores. Para el caso de la arquitectura LSTM los mejores modelos como se observa en la **Tabla 5.1** son con 50 celdas LSTM para cada capa probando ser la mejor configuración para estos modelos, también se puede observar que la configuración de 2 capas, 50 celdas y 88 steps (4 meses de información) se repiten en los modelos 60MD-LSTM y 30MD-LSTM. Respecto al mejor modelo según The Model Confidence Set, el modelo propuesto 60MD-MLP es el que posee una mejor capacidad predictiva siendo superior a su modelo base D-MLP y a los modelos bases D-SimpleRNN y D-LSTM así como también a los modelos propuestos en cada una de las arquitecturas, el pronóstico del mejor modelo se observa en la **Figura 5.1** donde se muestra el valor real del S&P500 y los pronósticos del modelo 60MD-MLP para el periodo fuera de la muestra, siendo el año 2020 completo. En general según los resultados mencionados los modelos MLP presentan los mejores MSE, seguidos por los modelos LSTM y luego por los modelos SimpleRNN, siendo los modelos propuestos MLP los mejores para pronosticar este índice.

Tabla 5.1: Resultados S&P 500

Valor del MSE, variación respecto al D-Model, configuración (layers, cells, steps) , Test DM y su valor-p en parentesis y el p-value del test MCS

Model	MSE	$\Delta \% r'$ D-Model	Configuration	DM	MCS
D-MLP	4403	-	(1;200;5)	-	0.000
60MD-MLP	3458	-21.46 %	(1;150;5)	3.20(0.0015)	1.000
30MD-MLP	3588	-18.51 %	(1;100;5)	2.45(0.0151)	0.004
15MD-MLP	3603	-18.17 %	(2;100;5)	1.98(0.0501)	0.000
D-SimpleRNN	4734		(1;100;22)		0.000
60MD-SimpleRNN	3902	-17.57 %	(1;25;88)	1.90(0.0583)	0.000
30MD-SimpleRNN	4168	-11.96 %	(1;100;22)	1.15(0.2476)	0.000
15MD-SimpleRNN	3845	-18.78 %	(1;100;22)	2.01(0.0451)	0.000
D-LSTM	4905	-	(1;50;22)		0.000
60MD-LSTM	3775	-23.14 %	(2;50;88)	1.14(0.2533)	0.000
30MD-LSTM	3872	-21.06 %	(2;50;88)	1.68(0.0927)	0.000
15MD-LSTM	3770	-23.14 %	(2;50;5)	2.70(0.0073)	0.000

**Figura 5.1:** SP500 prices and best model proposed: 60MD-MLP.

Fuente: Elaboración Propia.

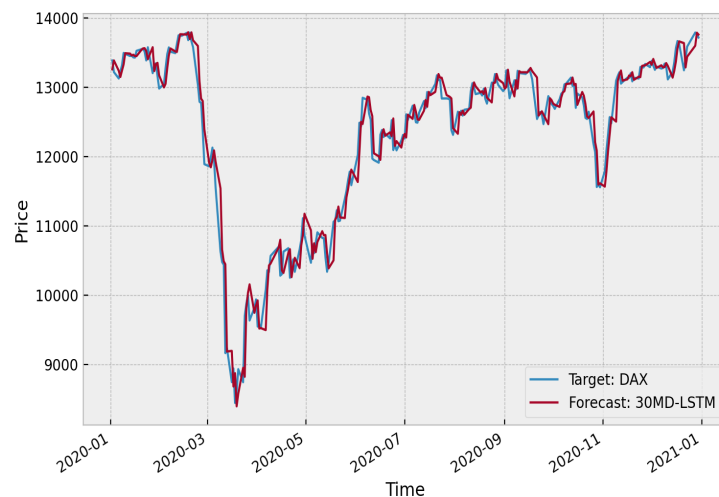
5.2. DAX

La **Tabla 5.2** muestra los resultados de los experimentos realizados para el índice bursátil DAX donde se evidencia que para la arquitectura MLP, los modelos propuestos 60MD-MLP , 30MD-MLP y 15MD-MLP reducen el error cuadrático medio en un 11,26 %, 9,77 % y 9,27 % respectivamente, según el test de DM tanto el modelo 60MD como el modelo 30MD los pronósticos son estadísticamente diferentes al modelo base D-MLP, a diferencia del modelo 15MD-MLP que no rechaza su hipótesis nula y no se puede afirmar la diferencia en los pronósticos. A su vez en los modelos SimpleRNN se genera una reducción menor de los modelos propuesto contra los modelos base en comparación a los modelos MLP, sin embargo, siguen siendo significativas tanto para el modelo 30MD-SimpleRNN y 15MD-SimpleRNN con una reducción del 9,38 % y 8,76 % respectivamente. Por otro lado, los modelos con arquitectura LSTM tienen una reducción del MSE de 10,15 %, 10,50 % y 10,10 % para los modelos 60MD, 30MD y 15MD respectivamente, según el Test DM los tres pronósticos de los modelos propuestos son estadísticamente significativos, demostrando la superioridad en la predicción respecto al modelo base. En relación a las configuraciones de los mejores modelos, se observa que para los modelos MLP con 2 capas y 5 back steps se dan las mejores configuraciones, también es de notar que se repite la configuración 2 capas, 150 celdas y 5 steps para el modelo 60MD-MLP y 15MD-MLP las capas. Para los modelos de SimpleRNN solo se observa que las mejores configuraciones se dan con una capa. Considerando todos los modelos según el test MCS el que posee mayor superioridad predictiva es el modelo 30MD-LSTM, a su vez se evidencia que los modelos propuestos LSTM tienen los mejores rendimientos, seguidos de los modelos propuestos con arquitectura SimpleRNN y luego por la arquitectura MLP. **Figura 5.2** muestra el valor real del índice DAX y los pronósticos del mejor modelo 30MD-LSTM para los datos fuera de la muestra (out-of-sample).

Tabla 5.2: Resultados Índice DAX

Valor del MSE, variación respecto al D-Model, configuración (layers, cells, steps) , Test DM y su valor-p en parentesis y el p-value del test MCS

Model	MSE	$\Delta \% r'$ D-Model	Configuration	DM	MCS
D-MLP	55221	-	(2;200;5)	-	0.000
60MD-MLP	49003	-11.26 %	(2;150;5)	1.89(0.0598)	0.000
30MD-MLP	49825	-9.77 %	(2;25;5)	1.72(0.0867)	0.000
15MD-MLP	50091	-9.29 %	(2;150;5)	0.01(0.9918)	0.000
D-SimpleRNN	52402	-	(1;150;5)	-	0.000
60MD-SimpleRNN	48851	-6.78 %	(1;150;88)	1.41(0.1604)	0.000
30MD-SimpleRNN	47250	-9.83 %	(1;25;5)	1.68(0.0941)	0.010
15MD-SimpleRNN	47813	-8.76 %	(1;100;22)	2.09(0.0367)	0.009
D-LSTM	52585	-	(1;100;88)	-	0.000
60MD-LSTM	47247	-10.15 %	(2;50;5)	3.03(0.0026)	0.011
30MD-LSTM	47062	-10.50 %	(2;50;66)	2.64(0.0087)	1.000
15MD-LSTM	47273	-10.10 %	(2;50;126)	2.99(0.0029)	0.015

**Figura 5.2:** DAX prices and best model proposed: 30MD-LSTM.

Fuente: Elaboración Propia.

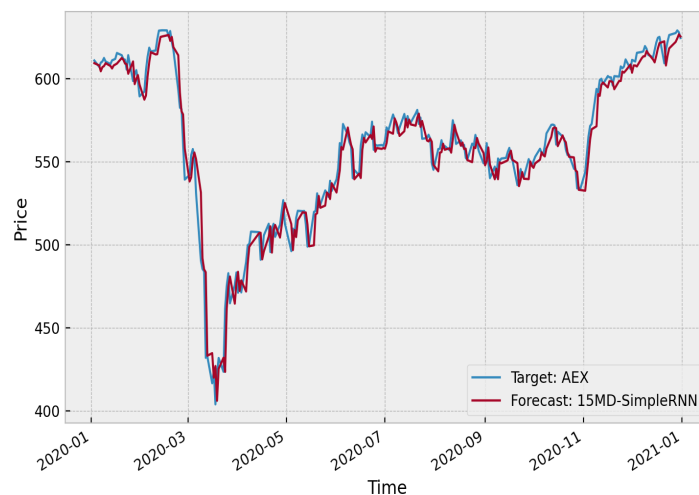
5.3. AEX

La **Tabla 5.3** muestra los resultados para el índice bursátil S&P500, se observa que para los modelos con arquitectura MLP los modelos propuestos con el enfoque de Auto-Similitud 60MD-MLP, 30MD-MLP, 15MD-MLP genera una reducción respecto al modelo base de 15,42 %, 19,79 % y 20,76 % respectivamente, donde los pronósticos generados por estos modelos son estadísticamente distintos según el Test de DM. Para la arquitectura SimpleRNN los modelos propuestos 60MD-SimpleRNN y 15MD-SimpleRNN son estadísticamente diferentes en un 8,53 % y 7,42 % respectivamente reduciendo el error cuadrático medio en esas cantidades, el modelo 30MD-SimpleRNN si bien muestra una mejoría en un 3,91 % este no es suficiente para ser estadísticamente diferente según DM. Por otro lado, los modelos de arquitectura LSTM propuestos 60MD-LSTM, 30MD-LSTM y 15MD-LSTM reducen en un 17,99 %, 19,81 % y 17,16 % respectivamente el MSE, según el test de DM los pronósticos de los modelos 60MD-LSTM, 30MD-LSTM y 15MD-LSTM son significativamente distintos. Respecto a las configuraciones obtenidas, se observa que para el caso del modelo MLP los mejores modelos son con 5 back steps, también se observa que la configuración de 2 capas, 200 celdas y 5 steps es la mejor para el modelo base y dos propuestos (60MD, 30MD), evidenciando que los modelos propuestos generan una mejora en la precisión de la predicción con la misma configuración del mejor modelo base. Respecto a los modelos SimpleRNN no siguen una lógica o estructura dada, sin embargo se observa que los steps destacados son 22 y 66. Para el caso de la arquitectura LSTM los mejores modelos como se observa en la **Tabla 5.1** son con 88 back steps para tres modelos y 22 para otro, también se puede observar que la configuración de 2 capas, 50 celdas y 88 steps(4 meses de información) se repiten en los modelos 30MD-LSTM y 15MD-LSTM. Respecto al mejor modelo según The Model Confidence Set, el modelo propuesto 15MD-SimpleRNN es el que posee una mejor capacidad predictiva siendo superior a su modelo base D-SimpleRNN y a los modelos bases D-MLP y D-LSTM así como también a los modelos propuestos en cada una de las arquitecturas, el pronóstico del mejor modelo se observa en la **Figura 5.3** donde se muestra el valor real del índice y los pronósticos del modelo 15MD-SimpleRNN para el periodo fuera de la muestra.

Tabla 5.3: Resultados Índice AEX

Valor del MSE, variación respecto al D-Model, configuración (layers, cells, steps) , Test DM y su valor-p en parentesis y el p-value del test MCS

Model	MSE	$\Delta \% r'$ D-Model	Configuration	DM	MCS
D-MLP	107.66		(2;200;5)		0.00
60MD-MLP	91.06	-15.42 %	(2;200;5)	3.39(0.0007)	0.00
30MD-MLP	86.35	-19.79 %	(2;200;5)	5.29(0.0000)	0.00
15MD-MLP	85.31	-20.76 %	(2;50;5)	4.17(0.0000)	0.00
D-SimpleRNN	93.05		(1;50;66)		0.00
60MD-SimpleRNN	85.11	-8.53 %	(1;100;22)	1.88(0.0608)	0.00
30MD-SimpleRNN	89.41	-3.91 %	(1;50;66)	1.15(0.2515)	0.00
15MD-SimpleRNN	86.15	-7.42 %	(1;100;22)	1.70(0.0892)	1.00
D-LSTM	102.15	-	(2;100;88)		0.00
60MD-LSTM	83.77	-17.99 %	(2;150;22)	2.34(0.0202)	0.00
30MD-LSTM	81.91	-19.81 %	(2;50;88)	2.61(0.0093)	0.00
15MD-LSTM	84.62	-17.16 %	(2;50;88)	2.31(0.0217)	0.00

**Figura 5.3:** AEX prices and best model proposed: 15MD-SimpleRNN.

Fuente: Elaboración Propia.

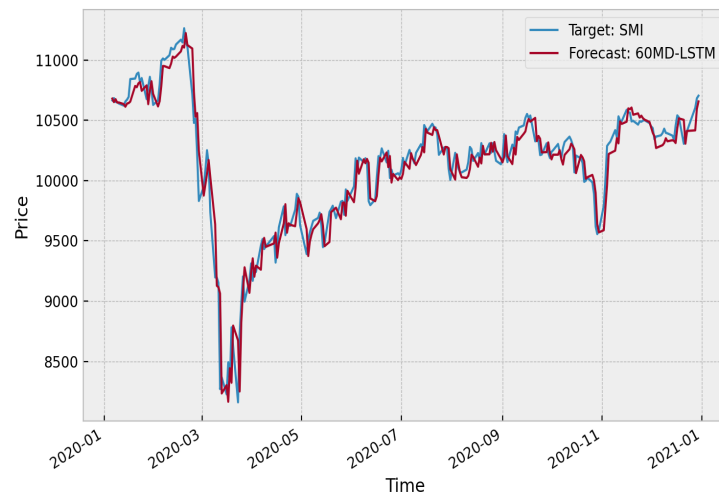
5.4. SMI

La **Tabla 5.4** muestra los resultados para el índice bursátil SMI, se observa que para los modelos con arquitectura MLP los modelos propuestos 60MD-MLP, 30MD-MLP, 15MD-MLP genera una reducción respecto al modelo base de 10,94 %, 10,02 % y 7,55 % respectivamente, donde los pronósticos generados por estos modelos son estadísticamente distintos según el Test de DM. Para la arquitectura Simple-RNN los modelos propuestos 60MD- SimpleRNN, 30MD-SimpleRNN y 15MD-SimpleRNN son estadísticamente diferentes en un 11,39 % y 13,38 % y 11,63 % respectivamente con respecto al modelo base reduciendo el error cuadrático medio. Por otro lado, los modelos de arquitectura LSTM propuestos 60MD-LSTM, 30MD-LSTM y 15MD-LSTM reducen en un 12,43 %, 10,14 % y 12,40 % respectivamente el MSE, según el test de DM los pronósticos de los modelos 30MD-LSTM y 15MD-LSTM son significativamente distintos. Respecto a las configuraciones obtenidas, se observa que para el caso del modelo MLP los mejores modelos son con 5 back steps siendo esta la configuración con menos información entrenada en los experimentos. Respecto a los modelos SimpleRNN los mejores modelos son con una configuración de 88 back steps, a excepción del modelo 15MD que tiene 5 back steps, también se observa que la configuración de 1 capa, 100 celdas y 88 steps, tiene un mejor desempeño en los modelos propuestos que en el modelo base. Para el caso de la arquitectura LSTM las mejores configuraciones de los modelos no sigue un patrón o estructura en común. Respecto al mejor modelo según The Model Confidence Set, el modelo propuesto 60MD-LSTM es el que posee una mejor capacidad predictiva siendo superior a su modelo base D-LSTM y a los modelos bases D-SimpleRNN y D-LSTM así como también a los modelos propuestos en cada una de las arquitecturas, el pronóstico del mejor modelo se observa en la **Figura 5.4** donde se muestra el valor real del índice SMI y los pronósticos del modelo 60MD-LSTM para el periodo fuera de la muestra.

Tabla 5.4: Resultados Índice SMI

Valor del MSE, variación respecto al D-Model, configuración (layers, cells, steps) , Test DM y su valor-p en parentesis y el p-value del test MCS

Model	MSE	$\Delta \% r' D$	Configuration	DM	MCS
D-MLP	23313	-	(1;150;5)	-	0.0000
60MD-MLP	20762	-10.94 %	(2;25;5)	2.47(0.0139)	0.0140
30MD-MLP	20976	-10.02 %	(1;25;5)	2.63(0.0088)	0.0098
15MD-MLP	21553	-7.55 %	(2;50;5)	1.91(0.0565)	0.0008
D-SimpleRNN	24645		(1;100;88)	-	0.0000
60MD-SimpleRNN	21838	-11.39 %	(1;100;88)	2.12(0.0348)	0.0002
30MD-SimpleRNN	21348	-13.38 %	(2;100;88)	2.19(0.0291)	0.0096
15MD-SimpleRNN	21780	-11.63 %	(1;25;5)	2.85(0.0047)	0.0001
D-LSTM	23461	-	(1;150;5)	-	0.0002
60MD-LSTM	20544	-12.43 %	(2;100;66)	1.47(0.1406)	1.0000
30MD-LSTM	21082	-10.14 %	(2;100;5)	2.26(0.0244)	0.0105
15MD-LSTM	20552	-12.40 %	(2;150;126)	2.06(0.0405)	0.0255

**Figura 5.4:** SMI prices and best model proposed: 60MD-LSTM.

Fuente: Elaboración Propia.

6 Conclusiones

En este trabajo se propone una metodología de entrenamiento de redes neuronales profundas enfocada en los comportamientos de Auto-Similitud que existen dentro de las series financieras. Realizando un re-entrenamiento con los datos intradiarios al modelo entrenado con datos diarios para predecir los precios diarios de los índices bursátiles S&P500, DAX, AEX y SMI. Esta innovación es de importancia para los distintos inversores, investigadores, entidades y organizaciones que están constantemente analizando estas bolsas y utilizando técnicas de machine learning e inteligencia artificial para predecir el precio y comportamiento de las series financieras de su interés.

Dado los resultados experimentales queda demostrado que el enfoque de Auto-Similitud, dado los comportamientos fractales mejora el rendimiento en la capacidad predictiva de los modelos de redes neuronales profundas generando mejoras significativas de hasta un 23 % para el S&P 500, de un 11.26 % para el índice DAX, de un 21 % para el índice AEX y un 12 % para el índice SMI. Además, se evidencia que existe una mejora significativa del modelo propuesto contra el modelo base en el 90 % de todas las configuraciones que se realizaron en este estudio. En las configuraciones presentadas para los distintas arquitecturas, se observa que para las arquitecturas MLP los modelos con mayor capacidad predictiva tienden a ser de menor complejidad o profundidad, tanto en las capas como en la cantidad de información utilizada. No así con las redes LSTM que exigen una mayor cantidad de recursos e información para tener resultados con mayor capacidad predictiva, se concluye que mientras el vector de entrada sea grande y la red neuronal tenga un cierto grado de profundidad mejores son los pronósticos.

Respecto a la metodología, se concluye que la inclusión de datos intradiarios para aumentar la cantidad de información que necesita una red neuronal para ser entrenada de forma

óptima y realizar mejores predicciones evitando el overfitting es efectiva demostrando que funciona y genera una mejoría, esto se explica debido a que la red al tener esta información de series con frecuencia minútales tiene comportamientos similares debido a la Auto-Similitud presente permitiendo generar un mejor pronóstico captando información que no había sido entregada por los datos diarios.

Cabe destacar que la presente metodología es un pie de inicio para lo que pueden ser una serie de futuras investigaciones relacionadas con el ámbito del deep learning y series de tiempo en donde se tenga información para distintos horizontes de tiempo, tales como Acciones, Futuros, Forex, ETFs, etc que son series que tienen condiciones idénticas para que la metodología sea replicada. Por otro lado, en este trabajo solo se mezclan datos cada 60 min, 30 min y 15 min por separado, sin embargo se podría hacer la inclusión de combinaciones para mejorar la predicción. También se podría realizar un enfoque a predecir los precios intradiarios y realizar una técnica de fine-tuning utilizada en el ámbito del machine learning, donde la red pre-entrenada es de datos diarios y se le hace un enfoque con datos minútales para predecir esa serie con esa frecuencia de tiempo. En conclusión, las aplicaciones que pueden ser derivadas de esta técnica son amplias y prometedoras basta con la curiosidad de los investigadores y/o inversores para ir avanzando en los desafíos para la predicción de series financieras y de tiempo.

Bibliografía

- Edgar E Peters. *Fractal market analysis : applying chaos theory to investment and economics*. New York : J. Wiley & Sons, 1994.
- Chin Weng Cheong. Self-similarity and multifractality in financial asset returns. *Mathematical and Computer Modelling*, 52:459–471, 2010. 10.1016/j.mcm.2010.03.031.
- DB. Madan and W. Schoutens. Self-similarity in long-horizon returns. *Mathematical Finance*, 30:1368–1391, 2020. 10.1111/mafi.12269.
- L. Calvet and A. Fisher. Multifractality in asset returns: Theory and evidence. *The Review of Economics and Statistics*, 84:381–406, 2002. 10.1162/003465302320259420.
- Y. Baek and Y. Kim. Modaugnet: A new forecasting framework for stock market index value with an overfitting prevention lstm module and a prediction lstm module. *Expert System With Applications*, 113:457–480, 2018. 10.1016/j.eswa.2018.07.019.
- Ying Xue. An overview of overfitting and its solutions. *Journal of Physics Conference Series*, 1168, 2019. 10.1088/1742-6596/1168/2/022022.
- R. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50:987–1007, 1982. <https://doi.org/10.2307/1912773>.
- T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31:307–327, 1986. 10.1016/0304-4076(86)90063-1.
- S. Siami-Namini and A. Namin. Forecasting economics and financial time series: Arima. *preprint*, 2018. arXiv:1803.06386.
- H. Jang and J. Lee. Machine learning versus econometric jump models in predictability and domain adaptability of index options. *Physica A-Statistical mechanics and its applications*, 513:74–86, 2019. 10.1016/j.physa.2018.08.091.

- S. Moshiri and N. Cameron. Neural network versus econometric models in forecasting inØation. *Journal of Forecasting*, 19:201–217, 2000. [https://doi.org/10.1002/\(SICI\)1099-131X\(200004\)19:3<201::AID-FOR753>3E3.0.CO;2-4](https://doi.org/10.1002/(SICI)1099-131X(200004)19:3<201::AID-FOR753>3E3.0.CO;2-4).
- AA. Adebiyi and CK. Adewumi, AO. Ayo. Comparison of arima and artificial neural networks models for stock price prediction. *JOURNAL OF APPLIED MATHEMATICS*, 2014. 10.1155/2014/614342.
- G. Erkam, K. Kayukutlu Gulgun, and D. Tugrul. Using artificial neural network models in stock market index prediction. *Expert System With Applications*, 38:10389–10397, 2011. 10.1016/j.eswa.2011.02.068.
- T. Fischer and C. Krauss. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270:654–669, 2018. 10.1016/j.ejor.2017.11.054.
- O. BeratSezer, M. UgurGudelekA, and H. MuratOzbayoglu. Financial time series forecasting with deep learning : A systematic literature review: 2005–2019. *Applied Soft Computing*, 90:106181, 2020. 10.1016/j.asoc.2020.106181.
- Y. Liu, Q. Zeng, and A. Carrio. Stock price movement prediction from financial news with deep learning and knowledge graph embedding. *Knowledge Management and Acquisition for Intelligent Systems*, 11016:102–113, 2018. 10.1007/978-3-319-97289-3_8.
- J. Patel, S. Shah, P. Thakkar, and K. Kotecha. Predicting stock market index using fusion of machine learning techniques. *Expert System With Applications*, 42:2162–2172, 2015a. 10.1016/j.eswa.2014.10.031.
- YQ. Liu, CY. Gong, and L. Yang. Dstp-rnn: A dual-stage two-phase attention-based recurrent neural network for long-term and multivariate time series prediction. *Expert System With Applications*, 143, 2020. 10.1016/j.eswa.2019.113082.
- Shun Chen and Lei Ge. Exploring the attention mechanism in lstm-based hong kong stock price movement prediction. *Quantitative Finance*, 2019. 10.1080/14697688.2019.1622287.
- W. Lu, J. Li, and J. Wang. A cnn-bilstm-am method for stock price prediction. *Neural Computing and Applications*, 33:4741–4753, 2021. <https://doi.org/10.1007/s00521>

- 020-05532-z(0123456789().,-volV)(0123456789().,-volV).
- Y. Kim and C. Wo. Forecasting the volatility of stock price index: A hybrid model integrating lstm with multiple garch-type models. *Expert System With Applications*, 103:25–37, 2018. doi:10.1016/j.eswa.2018.03.002.
- J. Cao, Z. Li, and J. Li. Financial time series forecasting model based on ceemdan and lstm. *Physica A: Statistical Mechanics and its Applications*, 519:127–139, 2019. 10.1016/j.physa.2018.11.061.
- T. Hsieh, H. Hsiao, and W. Yeh. Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm. *Applied Soft Computing*, 11:2510–2525, 2011. 10.1016/j.asoc.2010.09.007.
- W. Bao, J. Yue, and YL. Rao. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *Plos One*, 12, 2017. 10.1371/journal.pone.0180944.
- A. Aévalo, G. Niño, J. Hernández, and J. Sandoval. High-frequency trading strategy based on deep neural networks. *Lecture Notes in Computer Science Book series*, 9773, 2016. 10.1007/978-3-319-42297-8_40.
- E. Chong, C. Han, and F.C. Park. Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert System With Applications*, 83:187–205, 2017. 10.1016/j.eswa.2017.04.030.
- X. Li, X. Huang, X. Deng, and S. Zhu. Enhancing quantitative intra-day stock return prediction by integrating both market news and stock prices information. *Neurocomputing*, 132:228–238, 2014. 10.1016/j.neucom.2014.04.043.
- B. Frinjs and D. Margaritis. Forecasting daily volatility with intraday data. *The European Journal of Finance*, 14:523–540, 2008. 10.1080/13518470802187644.
- Benoît. Mandelbrot. How long is the coast of britain? statistical self-similarity and fractional dimension. *Science*, 156:636–638, 1967.
- Francois Chollet. *Deep Learning with Python*. MANNING SHELTER ISLAND, 2018.
- A.M. Turning. Computing machinery and intelligence. *Mind*, 236:433–460, 1950.
- S. Aras and I. Kocakoç. A new model selection strategy in time series forecasting with artificial neural networks: Ihts. *Neurocomputing*, 174:974–987, 2016. <https://doi.org/>

- 10.1016/j.neucom.2015.10.036.
- J. Patel, S. Shah, P. Thakkar, and K. Kotecha. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert System With Applications*, 42:259–268, 2015b. 10.1016/j.eswa.2014.07.040.
- T. Yamada, S. Murata, H. Arie, and T. Ogata. Representation learning of logic words by an rnn: From word sequences to robot actions. *Frontiers in neurorobotics*, 11, 2017. 10.3389/fnbot.2017.00070.
- S. Abd El-Moneim, MA. Nassar, MI. Dessouky, NA. Ismail, AS. El-Fishawy, and FE. Abd El-Samie. Text-independent speaker recognition using lstm-rnn and speech enhancement. *Multimedia tools and applications*, 79:24013–24028, 2020. 10.1007/s11042-019-08293-7.
- Sepp Hochreiter and Jürgen Schmidhuber. Long short term memory. *Neural Computation*, 9:1735–1780, 1997. <https://doi.org/10.1162/neco.1997.9.8.1735>.
- S. Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems*, 6:107–116, 1998. 10.1142/S0218488598000094.
- N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. <https://jmlr.org/papers/v15/srivastava14a.html>.
- D.P Kingma and J. Ba. Adam: A method for stochastic optimization. *Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego*, 2015. arXiv:1412.6980.
- F. Diebold and R. Mariano. Comparing predictive accuracy. *Journal of Business & Economic Statistics*, 13:253–63, 1995.
- P. Hansen, A. Lunde, and J. Nason. The model confidence set. *Econometrica*, 79:453–497, 2011. <https://www.jstor.org/stable/41057463>.
- F. Chollet et al. Keras. 2015. URL <https://github.com/fchollet/keras>.