UNIVERSIDAD DE MURCIA

ESCUELA INTERNACIONAL DE DOCTORADO

TESIS DOCTORAL

Machine Learning Meets Asset Management

Aplicaciones de Aprendizaje Automático en la Gestión de Activos Financieros

D. Pedro Manuel Mirete Ferrer

2023

UNIVERSIDAD DE MURCIA

ESCUELA INTERNACIONAL DE DOCTORADO

TESIS DOCTORAL

Machine Learning Meets Asset Management

Aplicaciones de Aprendizaje Automático en la Gestión de Activos Financieros

Autor: D. Pedro Manuel Mirete Ferrer

Director/es: D.ª María Asunción Prats Albentosa, y

D. Juan Samuel Baixauli Soler

UNIVERSIDAD DE MURCIA

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD
DE LA TESIS PRESENTADA EN MODALIDAD DE COMPENDIO O ARTÍCULOS PARA OBTENER EL TÍTULO DE DOCTOR
Aprobado por la Comisión General de Doctorado el 19-10-2022

D./Dña. Pedro Manuel Mirete Ferrer

doctorando del Programa de Doctorado en

Economía Interuniversitario (DECiDE)

de la Escuela Internacional de Doctorado de la Universidad Murcia, como autor/a de la tesis presentada para la obtención del título de Doctor y titulada:

MACHINE LEARNING MEETS ASSET MANAGEMENT

y dirigida por,

D./Dña. María Asunción Prats Albentosa

D./Dña. Juan Samuel Baixauli Soler

D./Dña.

DECLARO QUE:

La tesis es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, de acuerdo con el ordenamiento jurídico vigente, en particular, la Ley de Propiedad Intelectual (R.D. legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, modificado por la Ley 2/2019, de 1 de marzo, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), en particular, las disposiciones referidas al derecho de cita, cuando se han utilizado sus resultados o publicaciones.

Además, al haber sido autorizada como compendio de publicaciones o, tal y como prevé el artículo 29.8 del reglamento, cuenta con:

- *La aceptación por escrito de los coautores de las publicaciones de que el doctorando las presente como parte de la tesis.*
- *En su caso, la renuncia por escrito de los coautores no doctores de dichos trabajos a presentarlos como parte de otras tesis doctorales en la Universidad de Murcia o en cualquier otra universidad.*

Del mismo modo, asumo ante la Universidad cualquier responsabilidad que pudiera derivarse de la autoría o falta de originalidad del contenido de la tesis presentada, en caso de plagio, de conformidad con el ordenamiento jurídico vigente.

En Murcia, a 24 de Septiembre de 2023

Fdo.: Pedro Manuel Mirete Ferrer

Esta DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD debe ser insertada en la primera página de la tesis presentada para la obtención del título de Doctor.

Información básica sobre protección de sus datos personales aportados	
Responsable:	Universidad de Murcia. Avenida teniente Flomesta, 5. Edificio de la Convalecencia. 30003; Murcia. Delegado de Protección de Datos: dpd@um.es
Legitimación:	La Universidad de Murcia se encuentra legitimada para el tratamiento de sus datos por ser necesario para el cumplimiento de una obligación legal aplicable al responsable del tratamiento. art. 6.1.c) del Reglamento General de Protección de Datos
Finalidad:	Gestionar su declaración de autoría y originalidad
Destinatarios:	No se prevén comunicaciones de datos
Derechos:	Los interesados pueden ejercer sus derechos de acceso, rectificación, cancelación, oposición, limitación del tratamiento, olvido y portabilidad a través del procedimiento establecido a tal efecto en el Registro Electrónico o mediante la presentación de la correspondiente solicitud en las Oficinas de Asistencia en Materia de Registro de la Universidad de Murcia

A mi esposa Carmen, mi compañera infatigable de viaje. Gracias por estar a mi lado compartiendo tanto las alegrías como los desafíos. Eres mi fuente de energía y mi inspiración

A mis hijas, Carolina y Carmen, mi razón de ser y mi mayor orgullo. Gracias por inspirarme para ser una mejor persona y luchar por alcanzar mis metas

A mis padres. Gracias por estar ahí, al principio del camino, un camino largo pero que ha pasado volando...

ACKNOWLEDGEMENTS

My most sincere gratitude to Dr. Alberto García of Google LLC, without whom it would have been impossible to reach the end of this long journey. His guidance has been truly enriching, and his unwavering support, indispensable. Thank you, my friend.

I'd like to extend my thanks to my directors, Dr. María Asunción Prats and Dr. Juan Samuel Baixauli, from the University of Murcia, for their collaboration and contributions to the writing process of this thesis. I'm also grateful to Dr. Fernando Sánchez from the University of Murcia and Dr. Josep Peris from the University of Alicante for their support and encouragement throughout this journey.

Finally, I can't resist to make an acknowledge in first person: thanks for going on and believing that it's never too late. Because the best opportunities are the ones we give ourselves.

Mi agradecimiento más sincero al Dr. D. Alberto García, de Google LLC, sin el cual habría sido imposible llegar al final de este largo camino. Su asesoramiento técnico ha sido muy enriquecedor y su apoyo constante, imprescindible. Gracias, amigo.

Gracias a mis directores, la Dra. Dª María Asunción Prats y el Dr. D. Juan Samuel Baixauli, de la Universidad de Murcia, por su colaboración y sus aportaciones en el proceso de redacción de esta tesis. Y al Dr. D. Fernando Sánchez, de la Universidad de Murcia, y al Dr. D. Josep Peris, de la Universidad de Alicante, por sus muestras de apoyo y ánimo a lo largo del proceso.

Y, por último, no puedo resistirme a realizar un reconocimiento en primera persona: gracias por seguir creyendo que nunca es tarde. Porque las mejores oportunidades son las que uno se da a sí mismo.

PUBLISHED AND SUBMITTED CONTENT

- [Mirete-Ferrer et al., 2022] (https://doi.org/10.3390/risks10040084) was published in April, 2022, and part of its content has been included as part of the thesis.
- Some of the original contents from Chapters 2, 3 and 5 were used for the completion of this article, so there is a partial overlap with its content.
- The material from this source included in this thesis is not singled out with typographic means and references.
- The co-authors of the mentioned article have given their consent for its content to be used as part of this thesis.

CONTENTS

LIST OF ACRONYMS

AB Adaptive Boosting

ADF Augmented Dickey-Fuller Test

AFA American Finance Association

AHC-DTW Agglomerative Hierarchical Clustering algorithm

AL Average Linkage

ANN Artificial Neural Network

APT Asset Pricing Theory

ARCH Autorregressive Conditional Heteroskedasticity

ARIMA Autorregressive Integrated Moving Average

ARL Association Rule Learning

BB Bollinger Band

BH Bayesian Hierarchical

BRT Boosted Regression Tree

CAGR Compounded Annual Growth Rate

CAPM Capital Asset Pricing Model

CCAPM Consumption-based Capital Asset Pricing Model

CEQ Certainty Equivalent Return

CL Complete Linkage

CLA Critical Line Algorithm

CMA Conservative minus Aggresive

CML Capital Market Line

CNN Convolutional Neural Network

CR Calmar Ratio

CRSP Center for Research in Security Prices

CV Cross Validation

CVaR Conditional Value at Risk

DA Directional Accuracy

DBHT Directed Bubble Hierarchical Tree

DBN Deep Belief Network

DBSCAN Density-based Spatial Clustering of Applications with Noise

DL Deep Learning

DMLP Deep Multi-layer Perceptron

DCNN Deep Convolutional Neural Network

DFNN Deep Feedforward Neural Network

DNN Deep Neural Network

DR Differential Return

DRIP Deep Responsible Investment Portfolio

DRL Deep Reinforcement Learning

DT Decision Tree

DTW Dynamic Time Warping

DWT Discrete Wavelet Transform

EBITDA Earnings before interests, taxes, depreciation and amortization

EGB eXtreme Gradient Boosting

EIIE Ensemble of Identical Independent Evaluators

EMD Empirical Mode Decomposition

EMD-IGRU Hybrid model with Empirical Mode Decomposition and Individual Gated Recurrent Unit Network

EMH Efficient Market Hypothesis

EN Elastic Net

EQ Equal-weighted modeling

ES Expected Shortfall

ESG Environmental, Social and Governance

ETF Exchange Traded Fund

FE Feature Expansion

FFN Feed Forward Network

GA Genetic Algorithm

GAN Generative Adversarial Network

GARCH Generalized Autorregressive Conditional Heteroskedasticity

GBRT Gradient Boosted Regression Tree

GDO Gradient Descent Optimization

GLM Generalized Linear Model

GP Gaussian Process

GPR Genetic Programming

GRU Gated Recurrent Unit Network

GS Grid Search

HFT High Frequency Trading

HML High minus Low

HRP Hierarchical Risk Parity

ICAPM Intertemporal Capital Asset Pricing Model

IMK-ELM Incremental Multiple Kernel Extreme Learning Machine

IMF Intrinsic Mode Functions

KNN K-Nearest Neighbor

LASSO Least Absolute Shrinkage and Selection Operator

LR Logistic Regression

LSTM Long-short Term Memory

MACD Moving Average Convergence and Divergence

MAE Mean Absolute Error

MAPE Mean Absolute Percentage Error

MCS Monte Carlo simulation

MCV Minimum Cross Validation

MDD Maximum Drawdown

ML Machine Learning

MLP Multi-layer Perceptron

MSE Mean Squared Error

MVO Mean-variance Optimization

NLP Natural Language Processing

NN Neural Network

NTN Neural Tensor Network

NV Naive Bayesian Classifier

OLS Ordinary Least Squares

OOS R2 Out-of-the-Sample R Squared

OWL Ordered and Weighted LASSO

PBR Performance-based Regularization

PCA Principal Components Analysis

PWLS Penalized Weighted Least Squares

RF Random Forest

RFE Recursive Feature Elimination

RL Reinforcement Learning

RMSE Root Mean Squared Error

RMW Robust minus Weak

RNN Recurrent Neural Network

RRL Recurrent Reinforcement Learning

SDDRRL Stacked Deep Dynamic Recurrent Reinforcement Learning

SDF Stochastic Discount Factor

SGD Stochastic Gradient Descent

SL Supervised Learning

SLI Simple Linkage

SMB Small minus Big

SMO Sequential Minimal Optimization

SNR Sortino Ratio

SR Sharpe Ratio

SRI Socially Responsible Investment

STR Sterling Ratio

SVM Support Vector Machine

SVR Support Vector Regression

TR Treynor Ratio

UL Unsupervised Learning

VAR Vector Autorregression model

VaR Value at Risk

WLS Weighted Least Squares

LIST OF FIGURES

LIST OF TABLES

RESUMEN

El propósito de esta tesis es proporcionar una revisión exhaustiva, desde una perspectiva tanto teórica como práctica, de la utilidad que el Aprendizaje Automático (ML) aporta a la disciplina financiera de la Gestión de Activos. En años recientes, el mundo de la Gestión de Activos ha presenciado una transformación notable con la integración de tecnologías de vanguardia. Entre estos avances, la aplicación de Machine Learning (ML) destaca como una herramienta que ha transformado la toma de decisiones de inversión, capacitando a los gestores de activos para tomar decisiones de inversión más informadas y basadas en datos. La tesis realiza contribuciones en dos áreas principales. En primer lugar, proporciona una revisión global exhaustiva del estado del arte en cuanto a la contribución del ML a la Gestión de Activos. Buscando un equilibrio entre Finanzas, Estadística y Computación, la revisión intenta mejorar el análisis de la literatura reciente, a fin de que investigadores y profesionales puedan explorar sin fisuras sus áreas de interés.

De acuerdo al análisis realizado, la economía financiera tradicional no tiene respuestas perfectas para todos los obstáculos y desafíos de la disciplina. En este contexto, las técnicas de ML han encontrado un excelente terreno fértil para desarrollar todas sus potencialidades. La teoría financiera, el comportamiento de los mercados, las fuentes de datos en constante aumento y la innovación computacional son necesarios para una buena predicción y fijación de precios. Al combinar el conjunto de herramientas más completo, se pueden crear modelos computacionales realistas. Este objetivo se puede lograr utilizando tanto la econometría financiera como las técnicas de ML. Según diversos autores, las herramientas de ML proporcionan la capacidad de hacer predicciones más precisas al ser capaces de modelizar datos que no se comportan de forma lineal, comprender la interacción compleja entre variables y permitir el uso de conjuntos ingentes de datos no estructurados. Las herramientas de econometría financiera, de forma complementaria, pueden seguir siendo fundamentales a la hora de responder preguntas relacionadas con la inferencia estadística, para describir las relaciones que existen entre las variables relevantes en finanzas; cuando se aplican correctamente, el papel de las herramientas tradicionales no disminuye con la introducción de ML.

De acuerdo a nuestra revisión, los algoritmos de ML se emplean muy usualmente para la predicción de precios y el diseño de estrategias de trading en los mercados financieros. Se pueden establecer, a grandes rasgos, tres tipos diferentes de aplicaciones. El primer tipo de aplicación pronostica los precios o rendimientos futuros de los activos. Generalmente, se utilizan algoritmos de *Support Vector Regression (SVR)* y *Neural Networks (NNs)* en este tipo de estrategias. El alto número de aplicaciones de modelos *Long-short Term Memory (LSTM)* es especialmente notable cuando entran en juego series temporales. La desventaja de esta estrategia es que tiene una alta tasa de error debido a la dificultad de predecir los valores futuros de los activos basados en datos erráticos de los

mercados financieros. El segundo tipo de aplicación utiliza algoritmos como *Support Vector Machine (SVM)* y *Decision Trees (DTs)*. Estos enfoques generalmente tienen una precisión significativa en el pronóstico, pero esto no siempre implica alta rentabilidad. Por ejemplo, un modelo puede anticipar pequeñas ganancias correctamente pero prever pérdidas masivas de manera incorrecta, lo que resulta en un riesgo sustancial a la baja. La optimización basada en reglas es el tercer tipo. Su objetivo es encontrar el mejor indicador de trading y combinaciones de parámetros (por ejemplo, indicadores técnicos, indicadores fundamentales e indicadores macroeconómicos). Los algoritmos de optimización que se han explorado incluyen *Gaussian Process (GP)* y *Reinforcement Learning (RL)*. En resumen, las conclusiones de la literatura financiera más reciente indica que los algoritmos de ML superan claramente los enfoques tradicionales de la econometría, sobre todo en el ajuste de la predicción. Tanto los métodos clásicos como modernos de ML se han aplicado con éxito. Sin embargo, en el otro lado de la balanza, se puede decir que el panorama es extremadamente diverso en términos de datos y métodos aplicados, lo que sugiere una falta de benchmarks, metodologías y marcos comunes.

Relacionado con lo anterior, en la tesis se han identificado los siguientes desafíos y oportunidades para la investigación futura: 1) Conjuntos de datos estándar: El campo de ML aplicado a Finanzas se caracteriza por una falta de conjuntos de datos estandarizados que puedan ser reutilizados por la comunidad. Por lo tanto, crear una base de datos estándar (completa y lo suficientemente amplia) que pueda ser reutilizada por la comunidad de investigación es una necesidad para futuros trabajos. 2) Reproducibilidad: No se ha establecido una metodología o marco común para el entrenamiento y la evaluación de métodos. Esto perjudica la reproducibilidad, ya que la mayoría de los métodos analizados son difíciles, si no imposibles, de comparar entre sí (a menos que se vuelvan a implementar específicamente para cada escenario). Además, casi ningún artículo incluye códigos o datos a los que puedan acceder otros investigadores. Establecer un marco de reproducibilidad para la investigación de ML en la gestión de activos es un ámbito de trabajo de alto impacto para mejorar la calidad y el ritmo de trabajo de la comunidad investigadora. 3) Datos multimodales: La mayoría de los métodos se centran en analizar datos financieros numéricos para generar predicciones. Analizar fuentes de información alternativas como noticias, redes sociales, sentimientos y contenido generado por usuarios puede proporcionar pistas útiles para las decisiones financieras. Actualmente, pocos trabajos utilizan esas fuentes de datos. El desafío de combinar todas esas fuentes multimodales y múltiples arquitecturas podría traer consigo alcanzar nuevos niveles de precisión en la predicción. 4) Arquitecturas heterogéneas: Debido al estado de inmadurez en el que se encuentra el ML financiero en la actualidad (en comparación con otras técnicas más establecidas como el procesamiento de imágenes o el *Natural Language Processing (NLP)*), aún no se han establecido arquitecturas claras para el procesamiento de datos financieros. Existe una amplia variedad de artículos que generan nuevos modelos y pocos que se basan en bases sólidas para mejorarlos. Encontrar los patrones comunes y unificar esas diversas arquitecturas podría tener un efecto beneficioso para la comunidad y su adopción generalizada en la industria (de manera similar a cómo otras redes, como UNet o ResNet, se han convertido en el estándar de facto para muchas aplicaciones de procesamiento de imágenes). 5) Trading algorítmico. Desde una perspectiva académica, esta área está relativamente desconectada del marco teórico sobre la fijación de precios de activos y la inversión en valor, que ha tenido un papel central en la economía financiera durante las últimas cinco décadas. Sin embargo, a cambio, y precisamente debido a esta característica (dependencia exclusiva de los datos de precios), es la disciplina financiera que puede maximizar las contribuciones de las aplicaciones de ML. Se puede encontrar un desafío futuro en la posibilidad de combinar ambos aspectos, profundizando en los algoritmos de trading con una mayor relevancia en el uso de los fundamentos financieros.

En la segunda parte de la tesis, se lleva a cabo una aplicación práctica de diversas herramientas de ML en la Gestión de Activos, buscando evidencia empírica de su utilidad para mejorar los resultados obtenidos por enfoques tradicionales como la econometría. La primera sección empírica aborda el problema más general dentro de la disciplina de Gestión de Activos, la predicción de precios. A pesar de las limitaciones que la Hipótesis de Eficiencia (EMH) impone en cuanto a la predicción de precios de activos financieros, se aplican los algoritmos de ML más innovadores, específicamente las Gated Recurrent Unit Networks (GRUs), a los últimos 33 años de precios diarios del índice S&P 500, previamente procesados con un innovador algoritmo de descomposición de señales para series no estacionarias (Empirical Mode Decomposition (EMD)). Los resultados son prometedores, ya que el modelo propuesto EMD-IGRU muestra un nivel notable de precisión en la predicción de movimientos direccionales de precios de acciones. La Directional Accuracy (DA) asciende al 75,15% cuando se prueba en escenarios reales de trading.

En el segundo ejercicio, la tesis aborda otra estrategia de predicción de precios muy conocida entre los gestores de activos, llamada estrategia de pares o arbitraje estadístico. El experimento examina la viabilidad de utilizar técnicas de aprenizaje no supervisado para encontrar posibles pares de acciones que incorporar a una cartera *long-short*. Además del uso de *autoencoders* para la reducción de dimensionalidad, se propone aplicar un enfoque específico de *clustering* para datos de series temporales, llamado k-means clustering con Dynamic Time Warping (DTW). Utilizando datos mensuales desde 2010 para los componentes individuales del índice S&P 500, el modelo propuesto logra un rendimiento total del 89% en 8 años, con una DA del 63% y un Sharpe Ratio (SR) de 2,658. En comparación con los dos benchmarks seleccionados, el índice S&P 500 y una estrategia *short-term reversal*, el modelo propuesto es el que ofrece un mejor rendimiento.

Finalmente, el tercer ejercicio se centra en los modelos de tres y cinco factores de Fama y French, quizás los modelos de valoración de activos más seguidos por la comunidad científica en los últimos años. En este caso, la propuesta es más metodológica que empírica, ya que la tesis sugiere un método de estimación alternativa de estos modelos a través del uso de algoritmos de ML. Para ser más específicos, se aplica una combinación de Support Vector Regression (SVR) y Grid Search (GS) con Cross Validation (CV) de 10 capas para pronosticar rendimientos mensuales en cinco industrias del índice S&P 500, utilizando los modelos de tres y cinco factores de Fama y French. El modelo propuesto

GS-CV-SVR arroja coeficientes de correlación fuera de muestra entre 92-97% para los diferentes sectores, lo que supone una ganancia de precisión notable en la predicción en comparación con el método de optimización OLS convencional.

Como líneas de investigación futura, la tesis propone las siguientes: 1) En la estrategia de predicción direccional de precios, la efectividad del modelo propuesto (EMD-IGRU) se podría mejorar seleccionando algoritmos alternativos de ML para las diversas *Intrinsic Mode Functions (IMF)* obtenidas de la descomposición de las series de precios. Podría ser una línea de investigación futura el uso de métodos alternativos de descomposición de señales como la Transformación de Fourier y *Discrete Wavelet Transform (DWT)*. Se puede explorar el efecto de diferentes conjuntos alternativos de indicadores técnicos para encontrar el mejor conjunto para la previsión de IMFs. Por último, se podría aplicar el modelo en mercados internacionales alternativos, para comprobar si los resultados obtenidos pueden generalizarse y/o si el nivel de eficiencia del mercado es un factor crucial para el rendimiento del modelo. Como modelos de referencia alternativos, se podrían usar modelos más avanzados como los ARCH o GARCH. 2) En la estrategia de trading de pares, una mejora lógica consistiría en utilizar algoritmos de *Supervised Learning (SL)* para prever los rendimientos futuros de los pares de acciones seleccionados por los métodos de clasificación y agrupamiento k-means y *Density-based Spatial Clustering of Applications with Noise (DBSCAN)* de la familia de UL. *Recurrent Neural Network (RNN)*, LSTM o GRUs pueden considerarse como técnicas potenciales para mejorar el procedimiento del sistema de trading. Se puede explorar el efecto de diferentes conjuntos alternativos de indicadores financieros. Se pueden utilizar técnicas de clasificación alternativas para encontrar mejores enfoques y un rendimiento más brillante. Podría utilizarse una técnica adicional de agrupación jerárquica, como Agglomerative Clustering. De manera similar a los ejercicios anteriores, sería interesante probar si los resultados obtenidos pueden generalizarse a otros mercados diferentes al norteamericano. 3) En la propuesta de estimación de los modelos de Fama y French por SVR, se podrían utilizar algoritmos de SL de ML alternativos para lograr mejores resultados de precisión en la predicción. Por ejemplo, NN, regresión LASSO, regresión DT, modelo *K-Nearest Neighbor (KNN)* y regresión de Ridge. Como en ejercicios anteriores, sería interesante probar si los resultados obtenidos pueden generalizarse a otros mercados internacionales distintos al S&P 500. La estimación de los modelos se ha realizado solo en su dimensión de series temporales. Sería interesante para investigaciones futuras aplicar esta propuesta a datos transversales, siguiendo los procedimientos originales de los autores.

Por último, y como propuestas futuras a nivel general, el autor de la tesis se compromete a realizar en el futuro contribuciones empíricas en una de las áreas más relevantes de la Gestión de Activos, la Gestión de Carteras. En una línea estrechamente relacionada, esta tesis se ha abstenido, por falta de tiempo y recursos, de realizar contribuciones con el uso de algoritmos del paradigma de Aprendizaje reforzado (RL). Teniendo en cuenta que el área de Gestión de Carteras es quizás donde las contribuciones de RL son particularmente adecuadas, ambas futuras líneas de investigación quedan perfectamente alineadas.

ABSTRACT

The purpose of this thesis is to provide a comprehensive review, from both theoretical and practical perspectives, of the usefulness that ML brings to the financial discipline of asset management. In recent years, the world of asset management has witnessed a remarkable transformation with the integration of cutting-edge technologies. Among these advancements, the application of ML stands out as a revolutionary tool that has revolutionized how investment decisions are made, empowering asset managers to make more informed and data-driven investment choices. The thesis makes significant contributions in two main areas. Firstly, it provides a comprehensive global review of the state of the art regarding the contribution of ML in the field of asset management. By striking a balance between finance, statistics, and computational fields, our review enhances the analysis of recent literature, ensuring that researchers and practitioners can explore their areas of interest without encountering gaps. Secondly, the thesis focuses on the practical application of ML tools to typical asset management topics. It seeks to find empirical evidence of the usefulness of ML techniques in enhancing the results of traditional econometrics approaches. The first empirical section addresses the most general problem within the discipline of asset management: price forecasting. Despite the limitations that the Efficient Market Hypothesis (EMH) imposes on financial markets regarding the prediction of financial assets prices, we apply the most innovative ML algorithms, specifically GRUs, to the last 33 years of S&P 500 daily prices, previously preprocessed with an innovative decomposition signal algorithm for non-stationary series (EMD). The results are quite promising, as our Hybrid model with Empirical Mode Decomposition and Individual Gated Recurrent Unit Network (EMD-IGRU), according to the statistical standards, exhibits a reasonable level of accuracy in predicting stock prices and directional movements. The Directional Accuracy (DA) rises to 75.15% in the case of the best model. Moreover, it also demonstrates to be effective in predicting actual profits in real-world scenarios. In the second empirical exercise, the thesis faces another price forecasting topic, very well-known between asset managers, called pairs trading strategy or statistical arbitrage. The second experiment of the thesis looks at the viability of using UL techniques of ML to find possible stock pairs for a long-short portfolio. Besides using autoencoders for dimensionality reduction, we propose to apply a specific clustering approach to time series data, called k-means clustering with Dynamic Time Warping (DTW). As an additional contribution, we suggest using financial data (firm characteristics) as complementary features in the pairs selection. Using monthly data since 2010 for the individual constituents of the US index S&P 500, the proposed model achieves a total return of 89% in 8 years, with a DA of 63%, a Sharpe Ratio (SR) of 2.658, and a Calmar Ratio (CR) of 0.678. In comparison with the two benchmarks selected, the market and the short-term reversal strategy, our model is the best in terms of total return, but also in terms of the different trading performance measures, mainly SR figures. Finally, the third exercise is focused on the

three- and five-factor models from Fama and French, maybe the most reproduced asset pricing models in recent years. In this case, the proposal is more methodological than empirical, because we suggest an alternative estimation method of these models through the use of ML algorithms. To be more specific, we apply a combination of Support Vector Regression (SVR) and mixed Grid Search (GS) with 10-fold Cross Validation (CV) optimization to forecast monthly returns in five industries of US S&P 500 using the three and five factors presented by Fama and French. The proposed hybrid GS-CV-SVR model produce out-of-sample (testing data sets) correlation coefficients between 92-97% for the different sectors, except for the health sector with a 81%. The findings show that the prediction accuracy gain is quite notable when compared to the conventional Ordinary Least Squares (OLS) optimization method used in the original work of Fama-French and the traditional approach of the literature globally inspired in them.

1. INTRODUCTION

During the last seventy years, financial economists over the world have dedicated great efforts to model and forecast the stock returns, trying to understand the patterns behind them. This has been the greatest challenge of the research focused on Asset Management over the last decades. As [Cochrane, 2011] brilliantly exposed, "...in the beginning, there was chaos; practitioners thought one only needed to be clever to earn high returns. Then came the CAPM. Every clever strategy to deliver high average returns ended up delivering high market betas as well. Then anomalies erupted, and there was chaos again. The “value effect” was the most prominent anomaly". In the last ten years, the incredibly huge amount of anomalies found have driven the academic professionals to call the phenomenon as "factor zoo".

In contrast to this ‘anomaly-challenging’ branch of literature, a growing amount of work indicates remarkable investment performance based on signals generated by various ML methods. With the recent advancement in financial technology (Fintech), there is an increasing trend of employing ML technologies to find new signals on price movements and build investment systems that can beat human fund managers from a practical investment management point of view. ML routines in academic research have been implicitly motivated by the American Finance Association (AFA) presidential address of [Cochrane, 2011]. This academically reputed author suggested that, in the presence of a large number of noisy and highly correlated return predictors, there is a need for other methods beyond cross-sectional regressions and portfolio sorts. Indeed, ML uses "regularization" approaches to choose models, moderate over-fitting biases, find complicated patterns and hidden linkages, and handle high-dimensional predictor sets and more flexible functional forms ([Gu et al., 2020b]).

Several studies have highlighted the benefits of integrating ML into asset management practices. For instance, a research paper by [Wang et al., 2020] demonstrates how ML algorithms can outperform traditional approaches in forecasting stock returns and identifying profitable trading opportunities. These algorithms can analyze vast amounts of historical market data and identify hidden patterns, allowing portfolio managers to make informed decisions and potentially generate higher returns. The flexibility and adaptability of ML models have been a focal point in academic research. As highlighted in a study by [Bao et al., 2017], ML algorithms can adjust their strategies in response to changing market conditions, enhancing their resilience and robustness. This adaptability is particularly crucial in volatile markets, where traditional models may struggle to capture complex patterns. Furthermore, the integration of ML in asset management has led to the development of personalized investment solutions. By incorporating individual investor preferences and risk profiles, ML algorithms can tailor investment portfolios to meet the specific needs of clients ([López de Prado, 2016]). This customization not only improves client satisfaction but also increases the potential for long-term wealth accumulation.

To illustrate the gap that still exists between academic finance and the financial industry, with regards to the interest of the academic finance community for ML techniques, [Huck, 2019] carries out a very interesting experiment using the academic papers database. Until 2017, the search for "machine learning" via the Ebsco database, produces no reference at all in "The Journal of Finance", the leading academic finance journal, and only one reference in "The Journal of Financial Economics". In the last four years, the interest of academic finance for ML techniques has grown, due to the movement initiated by econometricians and statiscians.

As [Heaton et al., 2017] points out, the main difference between the use of ML tools in finance and other areas of science is that, in finance, "the emphasis is not in replicating tasks that humans already do well. Unlike recognizing an image or responding appropriately to verbal requests, humans have no innate ability to, for example, select a stock that is likely to perform well in some future period". For this reason, the usefulness of ML tools for financial purposes should be searched somewhere else. Specifically, they are extremely powerful in selection problems since, at their basis, they are the best and most rapid way to compute any function mapping data, and that is what returns, prices, economic data, accounting data, etc. are.

1.1. Main contributions

The purpose of this thesis is to provide a comprehensive review, from both theoretical and practical perspectives, of the usefulness that ML brings to the financial discipline of asset management. As previously mentioned, in recent years, the world of asset management has witnessed a remarkable transformation with the integration of cutting-edge technologies. Among these advancements, the application of ML stands out as a revolutionary tool that has revolutionized how investment decisions are made. ML, a subset of artificial intelligence, has the ability to analyze vast amounts of data and extract valuable insights, empowering asset managers to make more informed and data-driven investment choices.

The thesis makes significant contributions in two main areas. Firstly, it provides a comprehensive global review of the state of the art regarding the contribution of ML in the field of asset management. This review addresses a relatively understated field in the existing literature, despite the growing relevance of this financial discipline in the banking industry in recent years. By striking a balance between finance, statistics, and computational fields, our review enhances the analysis of recent literature, ensuring that researchers and practitioners can explore their areas of interest without encountering gaps. The scope of our review spans all topics in asset management, encompassing price forecasting, algorithmic trading, asset pricing and portfolio management.

Secondly, the thesis focuses on the practical application of ML tools to typical asset management topics. It seeks to find empirical evidence of the usefulness of ML techniques in enhancing the results of traditional econometrics approaches. By applying ML tools to various asset management aspects, we aim to shed light on how these advanced techniques can optimize decision-making processes and improve investment outcomes. In this sense, the aim of the empirical section of this thesis is to present representative examples of how ML can help to better understand the behavior of certain markets, more accurately predict the behavior of their prices, better capture the structure of certain investment strategies, or significantly improve the predictive capacity of multi-factor models of financial asset valuation. Throughout the entire empirical process, we will try to bring our results closer to the financial world. Despite the fact that ML tools are primarily designed to improve model prediction capability, and to enhance data interpretability does not be considered as a goal by itself, we will make an effort to convey to the reader the economic and financial interpretation of the obtained outcomes. While predictive accuracy remains a primary goal, it is not the sole focus. In the realm of finance, being able to explain the economic and financial implications of predictions, along with their deviations, is often just as crucial as making accurate predictions.

Through these contributions, the thesis strives to bridge the gap between theoretical understanding and practical implementation in the domain of asset management. By exploring the potential of ML and its real-world applications, this work aims to provide valuable insights to both academia and industry, fostering a deeper understanding of the potential benefits of incorporating ML into traditional finance and investment practices. Last but not least, a suitable way to summarize the objective of this thesis is that it aims to be a work that, despite containing advanced technical tools typically employed by engineers or mathematicians, can also be beneficial to economists and finance specialists. The thesis seeks to help them grasp the immense relevance that the utilization of these new technologies can have in their work, which has traditionally been distant from the realm of big data and emerging technologies, or at most, only acquainted with traditional statistical and econometric tools.

1.2. Outline

The thesis is organized as follows. In Chapter 2, we expose the methodology and terminology to be used in the rest of the thesis, as well as a review of the basic theoretical background about asset management and ML techniques. In Chapter 3, we provide a detailed description of the current state of the art in the application of ML tools to asset management, using a double outlook to classify the recent literature: the financial field of application and the ML empirical approach used. In Chapter 4 we show the results of our methodological contribution in the application of ML tools to financial applications in the area of asset management. Chapter 5 will be dedicated to the final remarks, conclusions and future lines of research.

2. THEORETICAL BACKGROUND

2.1. Introduction

This section includes all the methodology and terminology to be used in the rest of the thesis. ML applications in finance is a field that can be included in the discipline of quantitative finance. On the one hand, it includes a review of the basic financial theoretical background, from the definitions of the different financial disciplines to the empirical asset pricing models, passing by the well-known theoretical asset pricing models as Capital Asset Pricing Model (CAPM), APT, and so forth. On the other hand, this section introduces the main theoretical principles about ML, mainly those connected with asset and portfolio management in a wide sense. Finally, it includes a review and description of the main indicators of performance used in the literature of ML applications to asset management, very related to reproducibility issues.

2.2. Financial Background

According to [Snow, 2020], the asset management discipline can be divided "into four streams: portfolio construction, risk management, capital management, infrastructure and deployment, and sales and marketing". We will focus our work on the first stream, portfolio construction, which is intimately tied with the investment process. Portfolio construction is divided, in turn, into four areas: price forecasting, event prediction, value investing and weight optimization. ML techniques have different approaches for each one of these streams.

The first three areas can be included in the field of trading strategies, but they differ in the type of data used and the outcome they are trying to predict. Price strategies include technical analysis, macro global and statistical arbitrage (also called pairs trading), and the price plays a starring and lonely role. Right in the opposite corner, we can place value investing strategies, where the relationship between value and price is what generates the investment opportunities. Examples of these types of strategies are risk parity, factor investing, and fundamental investing. Finally, event strategies can be considered as part of a relatively new stream: an event-driven strategy refers to an investment strategy in which an investor attempts to profit from a stock mispricing that may occur during or after a corporate event. The theoretical basis of all of them, one way or another, can be found in the Asset Pricing theory, although in the case of price strategies, the relevance of the concept of value decays notably.

The fourth area, weight optimization, can be considered independently from the other three areas. It comprises the use of mathematical or statistical techniques to solve optimization and simulation problems in finance, like optimal execution, position sizing, and portfolio optimization.

Due to the extreme complexity and variety of disciplines included in the term of asset management, we have decided to define our own structure of disciplines to face the gathering, processing, classification and analysis of recent literature regarding ML techniques. We will divide the asset management disciplines into three main streams: value investing and price forecasting, as trading strategies topics, and portfolio management as an independent area which involves optimization. Value investing will comprise all the disciplines which use asset pricing models to select the most valuable assets to invest in. The most relevant and recent example of this type of discipline is factor investing. Price forecasting, on the other hand, will comprise all the financial areas focused on the best prediction of asset prices. We will open a special category in Chapter 3 for algorithmic trading, the most relevant area of price strategies in recent literature. Lastly, following the proposal from [Snow, 2020], weight optimization will be defined as an independent discipline that, for our purposes, will be categorized as portfolio management. As the initiated reader will have guessed, price forecasting and portfolio management are intimately connected since the predictions obtained from the first one improve the optimization solutions of the second one.

In short, these will be the three financial disciplines that will be used in the rest of the thesis:

1. Value/factor investing. Investment strategies which use asset pricing models to select the most valuable assets to invest in.

2. Price forecasting. Investment strategies focused on the best prediction of asset prices. Algorithmic trading can be considered as a special case of price strategy.

3. Portfolio management. Mathematical and statistical techniques which solve optimization and simulation problems in investment management.

2.2.1. Value/Factor Investing

Over the past three decades, hundreds of financial research articles have been dedicated to the study of asset pricing models with a dual purpose: on the one hand, analyze the behavior of asset prices and, on the other hand, try to find variables that contain information about them. The advances in financial research about asset pricing and quantitative methods about factor investing have been crucial for the exceptional development of the asset management industry in general and the enhancement of the investment processes, specifically.

Throughout this section, we will analyze both theoretical and empirical models. This body of knowledge is usually applied to cross-sectional data, that is, data from different financial assets at the same period of time. When this occurs, we will use the sub-index i. When we apply the models to data which vary over time, that is, time series data, we will use the sub-index t. The coexistence of both dimensions, cross-section and time-series, is traditionally solved in Econometrics through panel data methodology but, specifically in factor investing, this double dimension is faced through the methodology of [Fama and Macbeth, 1973].

Stochastic Discount Factor (SDF) as general source of asset pricing models

Regardless of the type of asset pricing models we use, all of them can be deduced from the general Stochastic Discount Factor (SDF) Model, also known as the Euler Equation. The SDF, also denominated as the pricing kernel, allows one to relate the current price of an asset to its future payoffs. The roots of this type of representation are based on the Arrow–Debreu model of general equilibrium, and its application to option pricing ([Cox and Ross, 1976] and [Ross, 1978]), along with the Asset Pricing Theory (APT) of [Ross, 1976].

What we will see in the next sections is how every single model used in asset pricing can be derived from this SDF model. First of all, we will deduce the way how the risk premium of an asset can be explained in terms of the covariance with the SDF. Secondly, we will demonstrate how the SDF model may be transformed into a factor model, or a linear function of specific risk factors, which offers variable discounts on uncertain payoffs depending on different states of the world [Peñaranda and Sentana, 2010]. Finally, we will describe the various theoretical models that can derived from the seminal consumption-based model and, finally, the empirical model contributions that have been made in the last decades to try to explain the existence of certain anomalies in asset pricing.

This representation's foundations are found in the Arrow-Debreu general equilibrium model and its application to option pricing [Cox and Ross, 1976] [Ross, 1978], along with the APT of Ross [Ross, 1976]. In [Hansen and Richard, 1987], the authors developed the discrete-time approach, highlighting the distinction between conditional and unconditional expectations. Finally, [Cochrane, 2000] restated all the contributions to asset pricing theory within this framework.

Using the same notations as [Cochrane, 2000], X_{t+1} will represent an asset pay-off at date $t+1$, P_t the asset price at date t, and m_{t+1} the SDF at date $t+1$. The fundamental equality states that:

$$P_t = E_t(m_{t+1}X_{t+1}) \tag{2.1}$$

where:

$$m_{t+1} = f(data, parameters) \tag{2.2}$$

Equation 2.1 indicates that the asset price is equal to the conditional expected value of the future payoff multiplied by the SDF, which is a random variable whose realizations are always greater than zero. The expected price can be understood, hence, as a weighted average of future payoffs in different states of the economy, where each state has a probability of occurrence defined by its SDF. Consequently, the SDF is simply a discount factor that transforms expected payoffs tomorrow into value today, but in a world of uncertainty; if there is no uncertainty or if investors are risk-neutral, the SDF is merely a constant that transforms expected payoffs tomorrow into value today.[1]

In intuitive terms, therefore, if the SDF, m_{t+1}, is small in a certain state of the world, then investors in this state attribute little value to the payoff received, which will cause the price to fall.

The major contributions of the stochastic discount factor approach are its simplicity and universality. Instead of using three apparently different theories for different types of assets, for example bonds, stocks, and derivatives, now we can consider them as just special cases of the same theory.

If we rewrite the equation introducing the concept of return R_{t+1}, this mechanism will be arguably clearer:

$$R_{t+1} = \frac{X_{t+1} - P_t}{P_t} = \frac{X_{t+1}}{P_t} - 1 \tag{2.3}$$

Now, using Equation (2.1) we can introduce the statistical decomposition of covariance in Equation (2.3):

$$E_t[R_{t+1}] = \frac{1}{E_t[m_{t+1}]} - 1 - \frac{1}{E_t[m_{t+1}]} Cov[m_{t+1}, X_{t+1}] \tag{2.4}$$

If we assume that the expected value of m_{t+1} is positive, then the expected return of assets that covaries positively with SDF will be lower than the expected return of assets whose returns has a negative covariance. In the next chapter we will explain the economic meaning of this result in terms of wealth and intertemporal substitution of consumption.

If we consider the existence of a risk-free asset, that is, an asset whose payoff at date $t + 1$ is certainly known as of date t, Equation 2.4 can be expressed in an alternative form including the expected excess return. If we apply Equation 2.4 to the risk-free asset, the covariance does not apply, so that the certain rate of return, R_t^d, satisfies:

$$E_t[R_{t+1}] = R_t^d = \frac{1}{E_t[m_{t+1}]} - 1 \tag{2.5}$$

[1]The term "stochastic discount factor" is used because m generalizes standard discount factor ideas. If there is no uncertainty, we may use the conventional present value formula to describe prices:

$$P_t = \frac{1}{1 + R_t^d} X_{t+1}$$

where R_t^d is the risk-free rate, the return of a discount bond with an unique and riskless payoff of 1 in the period $t + 1$.

If we define the risk premium of an asset as the expected return minus the risk-free rate, that is, the excess of return that an investor is demanding to invest in an asset which is not free of risk, we can re-arrange Equation 2.4. Subtracting R_t^d from both sides of Equation 2.4, we get that the expected excess return of the asset, $\tilde{\mu}_t$, is:

$$\tilde{\mu}_t = E_t[R_{t+1}] - R_t^d = -\frac{Cov[m_{t+1}, R_{t+1}]}{E_t[m_{t+1}]} \tag{2.6}$$

This expression in Equation 2.6 can be applied to any asset. In short, it means that only when the expected returns are negatively correlated with the SDF, that asset will earn a positive risk premium. In the next section, by suggesting a natural interpretation for the stochastic discount factor in a consumption-based model, we can add more understandable interpretation to these results.

The SDF in a Consumption-Based Model

We can find several ways to reach Equations 2.1 or 2.4. Basically, it will depend on the interpretation that we make about the SDF.

The consumption-based asset pricing model, which enables direct translation of the SDF architecture, derives the SDF from the optimality conditions for a single agent. The very well-known Consumption-based Capital Asset Pricing Model (CCAPM) [Breeden, 1979] (see section 2.2.1) can be regarded as an equilibrium version with multiple investors of this seminal consumption-based model.

The primary hypothesis is that an investor chooses what asset portfolio to hold along with how much money to save and spend, according to [Cochrane, 2000]. The solution to the problem's first-order criteria is the simplest fundamental pricing equation, which specifies that the price should be equal to the predicted discounted return with the discount rate being the investor's marginal utility. The marginal utility gain of selling the investment and spending the proceeds at some point in the future should be equal to the marginal utility loss of consuming a little less today and investing the difference. The investor should increase their investment in the asset if the price does not match this requirement.

An investor's first order conditions give the basic consumption-based model, in which the pricing kernel or SDF can be expressed as:

$$m_{t+1} = \delta \frac{u'(c_{t+1})}{u'(c_t)} \tag{2.7}$$

where c_t denotes the level of consumption in period t, u the utility function and δ the elasticity of intertemporal substitution of consumption. The investor selects his consumption strategy to maximize expected utility while staying within a budget constraint, and the program's optimality condition asserts that at the optimum, Equation 2.4 holds with the SDF.

Using the definition of covariance $cov(m, X) = E(mX) - E(m)E(X)$, we can rewrite Equation 2.1 as:

$$P_t = E_t(m_{t+1})E_t(X_{t+1}) + cov(m_{t+1}, X_{t+1}) \tag{2.8}$$

Substituting Equation 2.5 for the risk-free asset, we obtain:

$$P_t = \frac{E_t(m_{t+1})}{1 + R_t^d} + cov(m_{t+1}, X_{t+1}) \tag{2.9}$$

The formula for discounted present value is the first term in Equation 2.9. In the absence of risk, this is how the price of an asset would be determined. A risk adjustment is the second component. In a manner substantially similar to that described in the preceding section, an asset's price is increased (and its expected return is decreased) when the payoff covaries positively with the discount factor, and vice versa.

Now it is time to introduce the concept of consumption. To understand the risk adjustment, let us substitute back for m_{t+1} in terms of consumption in Equation 2.7, to obtain:

$$P_t = \frac{E_t(X_{t+1})}{1 + R_t^d} + \frac{cov(\delta u'(c_{t+1}), X_{t+1})}{u'(c_t)} \tag{2.10}$$

Marginal utility $u'(c)$ declines as c rises. Thus, the price of any asset will be lowered if its payoff covaries positively with consumption, and vice versa. What is the reason for such a behaviour like that? Investors dislike consumption unpredictability. Your consumption stream will become more volatile if you invest in an asset whose payment covaries favorably with consumption, such as one that pays out well when you are already affluent and badly when you are already poor. You will demand a more appealing price (cheap price) in order to purchase such an asset. The scenario will be precisely the opposite if you purchase an item whose expected return is inversely correlated with consumption since it contributes to smoothing consumption and hence has a higher value than its expected return may suggest. A great example of this sort of product is insurance. When wealth and consumption would otherwise be low, insurance pays handsomely. Because of this, even if the cost of insurance is more than the projected payout discounted at the risk-free rate, you are delighted to keep insurance.

Similarly, we can rewrite Equation 2.6 to introduce consumption in the expression of the expected excess return of an asset:

$$\tilde{\mu}_t = E_t[R_{t+1}] - R_t^d = -\frac{Cov[u'(c_{t+1}), R_{t+1}]}{E_t[u'(c_{t+1})]} \tag{2.11}$$

The expected return on each asset is equal to the risk-free rate plus an additional risk adjustment. How does it run? Consumption is more volatile when an asset's returns are strongly correlated with it, hence it needs to be priced more attractively (or guarantee higher predicted returns) to make up for the increased risk. In contrast, investments like insurance that have a negative correlation to consumption can have expected returns that are lower than the risk-free rate or even negative (net) expected returns.

Understanding how to convert this marginal utility for consumption into observable indicators is the key to understanding the various asset pricing models. Consumption may be a helpful signal since it decreases when marginal utility is high and increases when it is low, but there must be others. For instance, the CAPM [Sharpe, 1964], the all-time most common asset pricing model, offers the following argument: consumption is also low and marginal utility is high when the investor's other assets have performed badly. As a result, we may anticipate that prices for assets that correlate favorably with a significant index, such a market portfolio, would be cheap and that predicted returns will be high.

Finding meaningful indicators for marginal utility, items against which to compute a covariance in order to estimate the risk-adjustment for prices, is the goal of the remaining asset pricing models.

Risk premia and Systematic Risk

If we make a distinction between systematic and idiosyncratic risks, Equation 2.6 has an important implication. This "systematic component" of the risk [2] is defined, in the SDF model, as the orthogonal projection ($proj_m$) of the return on the SDF. In the SDF model, the "idiosyncratic component" (ϵ_{t+1}) [3] is defined as the residual of this projection in this general model, resulting in the following decomposition for the realized return:

$$R_{t+1} = proj_m[R_{t+1}] + \epsilon_{t+1} \tag{2.12}$$

with ϵ_{t+1} being uncorrelated from m_{t+1}, and the orthogonal projection being given by:

$$proj_m[R_{t+1}] = \frac{Cov_t[m_{t+1}, R_{t+1}]}{V_t[m_{t+1}]} m_{t+1} \tag{2.13}$$

Applying the last expression to Equation 2.6:

$$\tilde{\mu}_t = -Cov[m_{t+1}, proj_m[R_{t+1}]] \tag{2.14}$$

As a result, the idiosyncratic component ϵ_{t+1} does not contain any risk premium, and the risk premium only comes from the systematic component of the return that is precisely

[2] In financial terms, the systematic risk is the component of the total risk which is associated with the market and not the individual assets we want to price or invest in and, therefore, cannot be diversifiable.

[3] The idiosyncratic or non-systematic risk is the component of risk that can be associated with the specific factors or characteristics of each security, and is independent of the global factors of markets and economy. This type of risk can be diversified, reduced or controlled adequately through what is called in finance a "diversification strategy".

linked to the SDF. In other words, the risk premium is determined by the systematic risk of the asset as described with regard to the SDF, not the overall risk. In particular, regardless of the size of the overall risk, an asset whose return is uncorrelated from the SDF takes no risk premium.

Theoretical Factor Models

In this section we will demonstrate how the SDF model can be translated into a factor model, that is, a linear function of some risk factors, which discounts uncertain payoffs differently across different states of the world. Factor models correct the limitations of the unobservable SDF and the restrictions of a consumption-based approach bearing in mind that consumption is a macroeconomic aggregate that is provided with lags and is continuously revised.

The Equation 2.6 can be re-expressed, for the asset i as:

$$E_t[R_i] = R_t^d + \left(\frac{Cov[m, R_i]}{Var[m]}\right)\left(\frac{-Var[m]}{E_t[m]}\right) \tag{2.15}$$

or, what is equivalent,

$$E_t[R_i] = R_t^d + \beta_{i,m}\lambda_m \tag{2.16}$$

where $\beta_{i,m}$ is the regression coefficient of the return R_i on m. This is a beta pricing model, the most popular form of representation of asset pricing models in financial literature. It states that in a regression of returns on the discount factor, predicted returns on assets $i = 1, 2, ...N$ should be proportional to their betas. Keep in mind that while the coefficient $\beta_{i,m}$ changes from asset to asset, the coefficient λ_m is the same for all assets i. It is common to understand the λ_m as the risk premium and the $\beta_{i,m}$ as the amount of risk present in each asset.

Definition of a Factor Model. Equivalence between SDF representation and Beta representation A factor model suggests the existence of a scalar a and a Kx1 vector b for a Kx1 vector of factor values, such that the quantity described by:

$$m = a + b_i^T f_i \qquad i = 1, 2, ..., K \tag{2.17}$$

is an SDF. A factor f_k that has a non-zero loading b_k is known as "pricing factor".

In [Cochrane, 2000], the author demonstrated that formulating a factor model in terms of the SDF is equal to a "beta representation" of expected returns, which is more common across different factor model formulations.

Let us assume that f is Kx1 random vector. Thus, a scalar a and a Kx1 vector b exist such that Equation 2.17 prices all assets if, and only if, a scalar k and a K x 1 vector Λ exist such that for each asset i, the expected return is:

$$E[R_i] = \kappa + \Lambda_i^T \beta_i \qquad i = 1, 2, ..., K \tag{2.18}$$

where the Kx1 vector β_i is the vector of multivariate regression coefficients of R_i on f with a constant.

It can be proved that the equations that allow the translation between Equation 2.6 and Equation 2.18, that is, the equations which allow to switch between SDF models and beta models are:

$$\kappa = \frac{1}{E[m]} - 1 \tag{2.19}$$

$$\Lambda_i^T = -\frac{1}{E[m]} Cov[m, f_i] \tag{2.20}$$

$$a = \frac{1}{1+\kappa}(1 + \Lambda_i^T \Sigma_f^{-1} \mu_f) \tag{2.21}$$

$$b_i = \frac{-1}{1+\kappa} \Sigma_f^{-1} \Lambda_i \tag{2.22}$$

where Σ_f and μ_f are the vector of expected outcomes for the factors and the covariance matrix, respectively. Note that κ equals R_t^d if a risk-free asset exists. In the static CAPM, κ can be thought of as the return on a zero-beta portfolio.

As [Cochrane, 2000] explains in his book, "b_j (vector b) coefficients asks whether factor j helps to price assets given the other factors. b_j gives the multiple regression coefficient of m on f_j given the other factors. On the contrary, λ_j (components of vector Λ) asks whether factor j is priced, or whether its factor-mimicking portfolio carries a positive risk premium". λ_j gives the single regression coefficient of m on f_j. Therefore, when factors are correlated between them, one should test $b_j = 0$ to see whether to include a factor j given the other factors rather than test $\lambda_j = 0$. Multiple authors consider that, when we are able to find a factor whose risk premium is non zero ($\lambda_j = 0$), we can call it a "priced factor" or characteristic and, when we are able to find a factor that has a non-zero coefficient b_j, we can call it a "pricing factor".

Because it illustrates the magnitudes and signs of anticipated risk premiums for certain components, Equation 2.18 is useful. Understanding of Equation 2.6 is similar to this one in most cases. The model posits that the factor beta is associated with a positive risk premium if a risk factor f is negatively correlated with the SDF m. Since the largest payoffs occur when the value of payoffs is low, a component that is inversely correlated to marginal utility should have a positive premium. As a result, even if the actual value is low, the expected return is considerable. When there is a positive covariance, the opposite happens. Assets with a positive beta on the factor have a payout distribution that is "better" than risk free when the factor is high and payoffs are highly valued. As a result, the predicted returns on such assets might be lower than those of risk-free assets, and the expected return premium is negative.

When confronted with a factor model, a fundamental question arises: which are the genuine pricing factors? We will go over the two sorts of theoretical factor models that can be used to solve the prior question: equilibrium models (CAPM, Intertemporal Capital Asset Pricing Model (ICAPM), CCAPM) make assumptions about investors' preferences and derive a beta representation of expected returns of the form of Equation 2.17, while APT highlights the relevance of factors to explain returns from a statistical perspective.

Static CAPM The CAPM of [Sharpe, 1964] and [Lintner, 1965] is an equilibrium model in which the excess return on the market portfolio is the only pricing component. As a result, the model predicts that every asset's projected excess return is proportionate to its market beta.

The CAPM is built on [Markowitz, 1952] foundational work on mean-variance efficient portfolio creation. It essentially makes five different types of assumptions:

1. Investors are risk-averse.
2. Investors maximize the utility of terminal wealth.
3. Investors make their investment decisions on the basis of risk and return relationship.
4. Investors have similar expectations of risk and return.
5. Investors have identical time horizon.
6. Investors have free access to all available information.
7. Investors have identical time horizon.
8. There is risk-free asset and there is no restriction on borrowing and lending at the risk free rate.
9. Frictionless markets: there are no taxes and transaction costs.
10. There is total availability of assets and they are marketable and divisible.

Under these assumptions, the expected return on an asset is given by:

$$E[R_{i,t}] = R_t^d + \beta_i^m (E[R_t^m] - R_t^d) \tag{2.23}$$

R_t^d being the risk-free rate, R_t^m the return on the market portfolio and β_i^m the beta of the asset i with respect to the market.

If this model correctly reflects the reality, the excess return of a portfolio or asset which is not related to the market return and is persistent over time should be explained by randomness (ϵ) or investment skill (constant α not included in the equation above).

As we have explained before, this model implies the existence of an SDF given by:

$$m = a + b_i^T f_i \qquad i = 1, 2, ..., K \tag{2.24}$$

where the factor is the centered excess market return,

$$f = R_t^m - R_t^d - \tilde{\mu}_m$$

Intertemporal CAPM By allowing for various time horizons and preferences among investors, ICAPM [Merton, 1973] relaxes some assumptions of the static CAPM. Asset risk premia are linear functions of the market beta and other betas in terms of factors. As a result, the market factor is really not the exclusive determinant of pricing any longer.

The primary assumptions of the ICAPM mean some relaxation of the original framework in CAPM and may be summarized as follows:

1. Short sales and leverage, in particular, are permitted.
2. Investor expectations homogeneity: investors have similar ideas about the distribution of asset returns, but they have a wider range of perspectives and preferences.
3. Dynamic portfolios: Portfolios can be rebalanced on a regular basis (this assumption is linked to that of frictionless markets). This is the primary distinction between Static CAPM and Dynamic CAPM.

Under the previous assumptions, the expected excess return on asset i can be written as:

$$\mu_{i,t} - R_t^d = \beta_{i,t}^m(\mu_t^m - R_t^d) + \sum_{k=1}^{K-1} \beta_{i,t}^k(\mu_{X,t}^k - R_t^d) \tag{2.25}$$

where μ_t^m is the expected return on the market portfolio, $\mu_{X,t}^k$ is the expected return on the portfolio that maximizes the squared correlation with the k-th state variable.

These k portfolios are called hedging portfolios because they hedge each one of the k state variables. The ICAPM has the form of a factor model in which the K factors are the market premium in one hand, and a kind of $K - 1$ "mutual funds", in the other hand, according to Equation 2.25, committed to hedging against changes in investing prospects that are not beneficial. The predicted excess returns on these hedging strategies are the factor risk premia.

Consumption-based CAPM The ICAPM in its seminal form calls for determining the variables that influence the opportunity set's evolution. In [Breeden, 1979], the author introduces a Consumption-Based CAPM (CCAPM) that substitutes the multiple betas in the decomposition of expected returns by a single beta, which reflects changes in aggregate consumption. The assumptions are the same as in Merton's ICAPM.

[Breeden, 1979] shows that expected excess returns are given by:

$$\mu_{i,t} - R_t^d = \frac{\mu_t^C - R_t^d}{\beta_{C,t}^C} \beta_{i,t}^C \qquad i = 1, 2, ..., N \tag{2.26}$$

where $\beta_{i,t}^C$ is the beta of asset i with respect to aggregate consumption C, $\beta_{C,t}^C$ is the beta of the portfolio that maximizes the squared correlation with changes in aggregate consumption, and μ_t^C is the expected return on this portfolio.

Arbitrage Pricing Theory APT [Ross, 1976], along with CAPM is one of the two most influential theories on asset pricing. The APT varies from the acCAPM in that its assumptions are less restrictive. It offers an explanatory asset return model as opposed to a statistical one. Instead of the corresponding "market portfolio", it is suggested that each investor picks their own portfolio with their own set of betas. The securities market line indicates a single-factor model of asset price, with beta exposed to changes in market value, which in some ways makes the CAPM a special example of the APT.

In other words, the CAPM, but also the ICAPM, represent the SDF as an affine combination of factors. The APT, on the other hand, concentrates on return factor decomposition. A statistical description of asset returns as linear combinations of K common factors and a random disturbance serves as its foundation.

If x_i indicates the pay-off of an asset, then we have:

$$X_i = E[X_i] + \beta_i^T f + \epsilon_i, \qquad i = 1, 2, ..., N \tag{2.27}$$

where the idiosyncratic return is uncorrelated from the factors. Such a decomposition of factors is always satisfied since it is always possible to make a regression for a payoff on a given set of factors. Statistically, it is necessary to assume that there is not autocorrelation in the residuals ϵ_i across assets.

Empirical Factor Models

The model formulations explained above are able to express, under some conditions, the expected return of an asset as a linear function of the market return and a series of factors. When we have considered the marginal utility of consumption, we have supposed an approximation to the revisions in market expectations; when we have considered the absence of arbitrage, this has been implied a decomposition of the expected returns.

Given the CCAPM and the likelihood that consumption growth would be linked to macroeconomic fundamentals, it seems logical to look for the variables among macroeconomic data. According to APT, it seems convenient to look for asset pricing factors that explain a considerable amount of the time variation in individual returns, taking the premise that asset pricing factors should explain common variance in returns a step further.

Therefore, the expression derived from these theoretical models, in which the expected excess return of an asset is a linear function of the market return and a series of factors, regarding macroeconomic conditions, can be considered as the foundation for an entire empirical financial literature dedicated to explaining asset returns as linear combinations of factors, the well-known multifactorial asset valuation models.

Early work on the Sharpe-Lintner CAPM tended to be broadly supportive. In 1970s, the classic studies of [Black et al., 1972] and [Fama and Macbeth, 1973], found that, empirically, the high-beta stocks tended to have higher average returns than low-beta stocks and that the relation was roughly linear. During 1980s and 1990s, researchers began to look at other characteristics of stocks besides their betas. All those deviations from the original CAPM were called "anomalies", due to the fact that there did not exist any kind of theoretical model able to explain the existence of those kind of factors. For example, [Banz, 1981] reported the size effect, according to which small stocks had higher returns not explained by the CAPM. In [Fama and French, 1992], the authors drew further attention by classifying equities according to their size and beta, and demonstrating that high-beta stocks had no greater returns than low-beta stocks of the same size. [Basu, 1983] discovered a value effect, which is explained by the fact that market returns can be forecast by multiples of market price to accounting metrics like earnings or book value.

Empirically, all these anomalies may be explained more efficiently utilizing multi-factor models in which the factors are chosen based on empirical evidence rather than theoretical support. The core aim of factor models is to understand the drivers of asset prices. In the following lines, after introducing the usual procedure of estimation of all these models, we will make a brief review of the main contributions from the empirical factor models literature over the last three decades. But, contrary to the usual review across the different models and authors, we will make the revision using a different criteria: the main factors that has been progressively introduced in the literature and has been empirically tested as relevant across regions. We will focus our review on equities.

Two-pass Regression Procedure and Mimicking Portfolios As it was aforementioned, Equation 2.27, which represents the model that explains the expected returns of an asset as a linear combination of factors, needs to converge into an empirical estimation of all the parameters involved in the model. With N assets and K factors, there are NxK betas and K factor premia to estimate. The constant α is not always to be estimate: it will depend on the existence or not of the risk-free asset. If it exists, that α will be equivalent to the risk-free rate; otherwise, this constant must be considered as an unknown parameter.

The approach to obtain the NxK betas and the K factor premia was primarily introduced by [Fama and Macbeth, 1973]. This approach consists of running two series of regressions:

1. Run N time-series regressions of returns on factors to obtain NxK estimations for the β:

$$R_{i,t} = c_i + \sum_{k=1}^{K} \beta_{i,k} f_{k,t} + \epsilon_{i,t}, \qquad t = 1, 2, ..., T \tag{2.28}$$

2. Run a cross-sectional regression of expected returns on betas previously obtained (one average beta by each asset i) to get K estimates for the risk premia factors λ:

$$E[R_i] = \gamma + \sum_{k=1}^{K} \beta_{i,k} \lambda_k + \eta_i, \qquad i = 1, 2, ..., N \tag{2.29}$$

where γ is a constant and η_i is the error term associated to asset i.

From a statistical standpoint, there are some risks associated with the use of estimates of β instead of true betas. The main risk has to do with the bias we can find in the OLS estimates if the errors η are correlated with the regressors β.

The solution addressed by [Fama and Macbeth, 1973] has been embraced by hundreds of researchers since then. The proposal consists of grouping stocks in portfolios and estimating, instead of stock betas, portfolio betas. The idea behind this original suggestion is: because measurement mistakes tend to counterbalance across equities, portfolio estimates are more unbiased than individual estimates.

The other reason to use portfolios instead of individual stocks is that, most of times, factors are unobservable or immeasurable. For this reason, it appears the concept of "mimicking portfolios". A factor mimicking portfolio is a collection of assets designed to replicate the performance of a backdrop factor. When the factor's realizations are not returns, this design is typically favored over utilizing it directly. When we adopt this method to reduce noise in our asset pricing model (see for example [Vassalou et al., 2004] for generating replicating portfolios based on the macroeconomic factors), we only use the data recorded in the economic variables that is relevant for asset returns. A mimicking portfolio may also represent an unobservable factor when stock sensitivity to the factor are considered to be disclosed by certain business features. A particularly well-known illustration of this problem is given in the three-factor model by [Fama and French, 1992], where characteristics like company size or book-to-market ratio are meant to show how exposed a business is to various latent distress causes (coefficient β).

Suppose portfolio contains shares α_1, ..., α_j with $\sum \alpha_j^J = 1$. The sensitivity of this portfolio with respect to factor f_k is:

$$\gamma_k = \sum_j \alpha_j \beta_{jk}$$

The idiosyncratic risk of portfolio is $\nu = \sum_j \alpha_j \epsilon_j$, and hence:

$$\sigma^2(\nu) = \sum_j \alpha_j^2 \sigma(\epsilon_j),$$

A portfolio is only sensitive to factor k_0 (and idiosyncratic risks) if for each $k \neq k_0$,, $\gamma_k = \sum \alpha_j \beta_{jk} = 0$, and $\gamma_{k0} = \sum \alpha_j \beta_{jk_0} \neq 0$.

If a portfolio has the lowest idiosyncratic risk among portfolios that are just sensitive to k_0, it mimics that factor.

Size factor The empirical evidence that small-cap equities perform better than large-cap equities is known as size effect. [Van Dijk, 2011] made an extensive survey of 30 years of research in equity returns. As he recognizes, this additional factor in CAPM was primarily introduced by [Fama and French, 1992] with their 3-factor model, and since then, "there has been a vigorous debate on whether the size premium is a compensation for systematic risk." According to this research, there has been in recent years some divergence between the theoretical approach, which has been able to assess that size effect arises endogenously as a result of systematic risk, and the empirical approach, where recent studies assert that the size effect has progressively disappeared after the early 1980s.

Although there were several authors in 1980s who refer to this statistical anomaly (for example, [Banz, 1981]), it is not until 1992 with the seminal work from Fama and French that this effect is introduced in an empirical factor model. According to these authors, size is a stronger discriminating characteristic than market beta for predicted returns in the cross-section, and there is no substantial beta effect after correcting for size. It was, until that moment, the most serious threat to the CAPM. Making regressions in the Fama and MacBeth way, they demonstrated that, for the case of American stocks, the slope between returns and size was significantly negative over long periods and that the beta effect was not robust to a size control.

As in the rest of factors, we can find two types of explanation for the existence of the size effect: risk-based explanations and behavioural explanations. Small size companies, according to some researchers (for instance [Fama and French, 1992], outperform large size equities because there is a larger exposition to systematic risk. According to this outlook, small caps are more susceptible than large caps to have extensive periods of decreased earnings, which is a risk element that should be compensated. Finally, another line of thinking relates the size effect to firm default risk, since the excess return of small firms would be justified by their higher financial distress. In the second case, we can find the proposal by [Banz, 1981], who claims that the size impact is caused by a lack of knowledge on small companies. Other similar explanations involve the degree of confidence from investors or the lag in the formation of prices upon the arrival of new information about firms.

The size factor is represented as the excess return of small caps over large caps. [Fama and French, 1992] introduced a "Small-Minus-Big"(SMB) portfolio, which is a zero-investment portfolio built as the difference between the average return on three small-cap portfolios and that on three large-cap portfolios, which has been ordered, previously, according to the book-to-market ratio (Value, Neutral and Growth).

This previous filter is defining the third factor, "High-Minus-Low"(HML) of their model, which is another zero-investment portfolio built as the difference between the average return on two value stock portfolios and that on two growth stock portfolio, according to the size quantiles (Big and Small).

The 3-factor model by [Fama and French, 1992] was expressed as follows:

$$E[R_{i,t}] = R_t^d + \beta_i^m(E[R_t^m] - R_t^d) + \beta_i^{SMB} SMB_t + \beta_i^{HML} HML_t + \epsilon_{i,t} \tag{2.30}$$

R_t^d being the risk-free rate, R_t^m the return on the market portfolio, β_i^m the beta of the asset i with respect to the market factor, β_i^{SMB} the beta of the asset i with respect to the size factor, and β_i^{HML} the beta of the asset i with respect to the value factor that will be developed shortly.

The size impact is both a priced and a priced component, according to a significant number of academics who have looked for evidence over the past three decades (see section 2.2.1). One may assume that the size factor will be higher in "good" world states where the marginal utility of consuming is lower since the asset pricing theory based on the SDF requires that a factor bears a positive premium if and only if it has a negative correlation with the SDF.

Most of them, as in the case of [Vassalou et al., 2004] find the factor premium to be negative (the opposite sign to expected) but not significant [4], although the coefficient or loading [5] of the size factor in the SDF is significant. Therefore, it means that the size factor is a "pricing factor" because is relevant for the asset pricing, but not a "priced factor" because it does not command a significant premium.

More recently, some papers check how compatible is the SMB factor with the theoretical models. [Pukthuanthong and Roll, 2019] explain how the value factor is linked to the major components of the covariance matrix of returns, satisfying the APT model's requirements.

Value factor The concept of "value" and the methodology of value investing are extremely popular since the seminal works of [Graham and Dodd, 1934]. Generally speak-

[4] $\lambda_k = 0$ in Equation 2.16 and $SMB = 0$ in Equation 2.30

[5] β_i^k in Equation 2.25 and β_i^{SMB} in Equation 2.30

ing, a stock can be considered as "value" if it is undervalued according to fundamental analysis measures. The broadly accepted measure in factor investing to make a distinction between "value" stocks and their contrary, "growth" stocks, is the book-to-market ratio (BE/ME). When this measure is low, the stock can be considered as "inexpensive" from a fundamental point of view, and vice versa, for the case of "expensive" stocks.

Although there exists some previous empirical evidence about the relationship between expected returns and book-to-market ratios, as in [Stattman, 1980], it is not until the omnipresent work from [Fama and French, 1992] when this value effect is included in an asset pricing model in the CAPM way. In their study, the value effect associated with the factor HML defined in the previous section (Equation 2.28) turns out to be stronger than the size effect connected with the SMB factor.

Numerous publications published over the last thirty years have thoroughly examined the robustness of the value impact across different geographic areas and time intervals. For example, [Liew and Vassalou, 2000] discovered that value equities outperform growth stocks in several developed nations, but [Vassalou et al., 2004] discovered that this behavior is not uniformly strong in every segment of the stock market. Other authors have recommended using more sophisticated value measurements. For example, [Basu, 1977] demonstrates that in the 1960s, low P/E [6] portfolios outperformed high P/E portfolios, and [Basu, 1983] shows that this effect persists after controlling for business size.

As for the case of the size effect, there exist risk-based and behavioral explanations for the value effect. In the first case, the systematic risk factor associated to value stocks could be related to financial distress, as it is indicated in most of the articles about this issue. That is the case of [Vassalou et al., 2004]. With respect to the rational explanations, the justifications for a value premium have to do with the excess of optimism of investors regarding stocks that have performed well in the recent past, and vice versa. For instance, in [Lakonishok et al., 1994], the explanations arise from the rejection that investor feel for the least appealing stocks, which have done poorly in the past. That provokes a drop in their prices and increases, subsequently, their expected returns.

Finally, there are numerous research that deal with the existence of the value factor. The HML portfolio, like the value effect, is designed to be a mimicking portfolio for an unobservable factor, whose exposure is proxied by the BE/ME ratio. According to [Liew and Vassalou, 2000], the HML factor is favorably associated with GDP growth predictions, supporting the idea that this portfolio has a positive risk premium since it outperforms in good times. Recent additions by [Gerakos et al., 2013] raised doubts on the employment of the HML component, claiming that it satisfies both the conditions of a priced and an unpriced factor, resulting in the 3-factor model's illusory success.

[6] The price-to-earnings ratio (P/E ratio) is a valuation measure that compares the current share price of a company to its earnings per share (EPS). The price-to-earnings ratio, also known as the earnings multiple, is a ratio that compares a stock's price to its earnings and can be used to determine relative valuation. In general, and under certain assumptions, a low P/E ratio suggests stock undervaluation, and vice versa.

Momentum factor Beyond the size and value factors, the momentum factor is the most prevalent factor in the literature. Momentum can be defined as the rate of acceleration of a security's price, and simply, it refers to the inertia of a price trend to continue either rising or falling for a particular length of time. The trading strategies related to this effect, also called "trend following", seek to capitalize on momentum to enter a trend as it is picking up steam. In statistical terms, the momentum effect characterizes by the existence of serial autocorrelation.

The validity of the random walk hypothesis and, as a result, the unpredictability of asset returns (see next section 2.2.2) were early concerns for jobs that faced the existence of momentum effect in stock prices. An example of this kind of approach, which uses time series data, is [Lo and MacKinlay, 1988]. From the perspective of cross-sectional data, we can find [Jegadeesh, 1990], where authors examine the performance of stock selection techniques based on prior monthly returns. Despite the fact that this anomaly was discovered after the size and value effects, institutional investors have a long and reliable history with the momentum approach [Asness et al., 2014].

The studies by [Jegadeesh and Titman, 1993] can maybe considered as seminal in the finding of strong evidence in favor of the momentum effect for intermediate horizons. They introduced the term "relative strength strategies" for currency-neutral strategies that buy the best-performing stocks over the past 3 to 12 months and sell the worst-performing stocks.

The paper by [Carhart, 1997] can be considered as the study which, in the last years, has received more acknowledgement from the investment industry. The author provides evidence in this research, which focuses on the mutual fund business, that strong previous performance does not necessarily imply future returns, but that the contrary might be true (if the performance is based on loading up on specific risk factors). This paper presents a four-factor model with the 3 factors from [Fama and French, 1992] plus a new factor which represents the momentum effect:

$$E[R_{i,t}] = R_t^d + \beta_i^m(E[R_t^m] - R_t^d) + \beta_i^{SMB} SMB_t + \beta_i^{HML} HML_t + \beta_i^{WML} WML_t + \epsilon_{i,t} \quad (2.31)$$

β_i^{WML} being the beta of the asset i with respect to the momentum factor. The WML factor is defined as the excess return of an equally-weighted portfolio for the 30% past year winners over an identical portfolio of the 30% past year losers ("Winners-Minus-Losers".

After adjusting for the remainder of the risk variables, this article indicates that fund performance has very little consistency. Its findings do not support the presence of professional or knowledgeable mutual fund portfolio managers, since the only substantial persistence not explained is focused in substantial underperformance by the worst-return mutual funds.

Regarding the justification of the momentum anomaly, we can find again papers in the two usual ways: risk premium and behavioral explanations. In the first category, we can find [Jegadeesh and Titman, 1993], who discover a positive autocorrelation of the idiosyncratic component of returns that cannot be explained by size and value effects, and [Fama and French, 1996], who confess that the short-term persistence of returns is the "main embarrassment" for their three-factor model. In the second category, the main explanation used is the gradual diffusion of information in prices. For instance, [Hong and Stein, 1999] present a model that forecasts the phenomenon of over and under-reaction in the process of investment.

As for the size and the value factors, [Pukthuanthong and Roll, 2019] describe statistical tests to find the compatibility of momentum effect with the ICAPM and the APT models. The results suggest that, unlike size and value factors, the momentum factor does not seem a priced factor in the sense of ICAPM.

Profitability and Investment factors Following the release of the five-factor model [Fama and French, 2015], these two components have lately gained a lot of traction in stock investment techniques. This model was expressed as follows:

$$\begin{aligned} E[R_{i,t}] = R_t^d + \beta_i^m(E[R_t^m] - R_t^d) + \beta_i^{SMB} SMB_t + \\ + \beta_i^{HML} HML_t + \beta_i^{RMW} RMW_t + \beta_i^{CMA} CMA_t + \epsilon_{i,t} \end{aligned} \tag{2.32}$$

β_i^{RMW} being the beta of asset i with respect to the profitability factor, and β_i^{CMA} the beta of asset i with respect to the investment factor.

The procedure to estimate these factors is similar to previous factors in [Fama and French, 1992]. Stocks are first sorted according to a measure of profitability or investment. The profitability factor is the excess return of robust profitability stocks over weak profitability ones ("Robust-Minus-Weak" or RMW factor) while in the case of the investment factor, it is defined as the excess return of high investment stocks over low investment ones ("Conservative-Minus-Aggressive" or CMA factor). The authors choose as measures the operating profit after interest expenses and the growth of total assets, respectively.

Empirical evidence about these two types of effects can be found in earlier papers in 1990s. For instance, [Lakonishok et al., 1994] examined a company's historical profitability as a way to determine its "value" based on the book-to-market ratio, in the sense that stocks with a discrete path of sales growth can be considered as value stocks, unlike the stocks with a brilliant path of sales growth in the past, that can be considered as growth stocks. Bearing in mind that the former ones are less appealing, the intuition says that investors will pay less for them, which rises their subsequent expected returns. Exactly the opposite with the growth stocks, that will have lower expected returns. More recent papers, as in the case of [Novy-Marx, 2013], discover empirical evidence showing that lucrative companies perform better than less profitable ones (as determined by the gross profits-to-assets ratio). Other studies like [Hou and Zhang, 2015] show that Return on Equity (ROE) is a good measure to test the existence of the profitability factor, in the sense that a strategy that buys stocks with high ROE and sells stocks with low ROE generates significant extraordinary returns.

[Hou and Zhang, 2015] also describe how using several sorts in the definition of factors may help decrease correlation, and how a model that incorporates the market factor, a size factor, and profitability and investment factors can price a large cross section of test portfolios appropriately.

2.2.2. Price Forecasting

Introduction

Price forecasting is a financial discipline dedicated to predicting the future price of an asset using market data and its transformations. Examples of this type of strategy include trend trading strategies, which involve taking a position in an asset only after detecting changes in trends, as well as statistical arbitrage, where one looks for mispricing between assets by identifying relationships or potential anomalies that are expected to return to normal. Algorithmic Trading, which has become increasingly popular in recent years, can be considered part of trend-following methods and therefore connected with price forecasting approaches.

In this discipline, asset prices play a central and lonely role, while asset pricing models and economic/financial fundamentals stay in a very secondary term. Sometimes, the academic approach to price forecasting has not considered risk-related issues or, directly, has not been interested in finding explanatory variables of risk, except for the best prediction of prices. Shortly, the studies have been more interested in the predicting power of modeling than in the explanatory power and economic meaning of the variables included in the model. In the case of technical and trend strategies, the main theoretical topic has to do with the accomplishment of EMH that we will expose in the next few lines. Although statistical arbitrage strategies are also very related to EMH and autocorrelation issues, they are connected, to some extent, to theoretical issues previously explained in the Value Investing section, related to asset pricing and anomaly-seeking. In this sense, in the second section of this chapter, we will talk about the theoretical basis of one of the most followed strategies in statistical arbitrage, the pairs-trading strategy, which will play a central role in the empirical part of this thesis.

Regardless of the type of approach, time-series data analysis is prevalent in this financial discipline, unlike the previous one, value investing, in which cross-sectional data was clearly predominant.

Efficient Market Hypothesis (EMH)

The EMH states that asset prices always reflect all available information. A direct implication is that it is impossible to "beat the market" consistently on a risk-adjusted basis, since market prices should only react to new information, which can be considered as a crucial implication for the asset management industry, mainly in the case of price forecasting. The EMH is linked to the random walk theory, which defends that the best prediction for tomorrow's price is the current price.

The validity of the random walk hypothesis and, as a result, the unpredictability of asset returns were early concerns for jobs that faced the existence of the momentum effect in stock prices (see previous section). An example of this kind of approach, which uses time-series data, is [Lo and MacKinlay, 1988]. From the perspective of cross-sectional data, we can find [Jegadeesh, 1990], where authors examine the performance of stock selection techniques based on prior monthly returns. Despite the fact that this anomaly was discovered after the size and value effects, already exposed in previous section, institutional investors have a long and reliable history with the momentum approach [Asness et al., 2014].

The fundamental model of asset pricing, the SDF model, may explain how efficient markets are (or are not) connected to random walk theory. This model, as mentioned in the preceding section, makes mathematical predictions about a stock's price assuming that there is no arbitrage, that is, assuming that there is no risk-free way to successfully trade. Formally, if arbitrage is impossible, the model predicts that a stock's price will be the discounted value of its future price. We may rewrite the Equation 2.1 if we imagine we live in a world without dividends or, to be more limiting, if we assume we are functioning in the short term and no dividend is paid:

$$P_t = E_t(m_{t+1} P_{t+1}) \tag{2.33}$$

Note that this equation does not generally imply a random walk. However, if we assume the SDF is constant [7], we have:

$$P_t = mE_t(P_{t+1}) \tag{2.34}$$

Taking logs and assuming that Jensen's inequality term is negligible, we have:

$$\log P_t = \log m + E_t(\log P_{t+1}) \tag{2.35}$$

which implies that the log of stock prices follows a random walk (with a drift).

Regardless the type of asset pricing model we use, the EMH will be satisfied only under some restrictive assumptions. As [Cochrane, 2000] points out, "...if investors are

[7] In the consumption-based model already described, it means that investors are risk neutral, i.e., $u(c)$ is linear or there is no variation in consumption, and we are in short time horizons where δ is close to one.

risk-neutral, returns are unpredictable, and prices follow martingales (random walk process). In general, prices scaled by marginal utility are martingales, and returns can be predictable if investors are risk averse and if the conditional second moments of returns and discount factors vary over time. This is more plausible at long horizons".

As [Timmermann and Granger, 2004] points out, "The EMH is a backbreaker for forecasters", because in its crudest form it effectively says that returns from speculative assets are unforecastable. That may appear to be the conclusion of the narrative from an intellectual standpoint. Despite the strength of the argument, it does not appear to be fully compelling to many forecasters. The reason is that, despite its simplicity, the EMH is surprisingly difficult to test and considerable care has to be exercised in empirical tests.

EMH and stationarity of time series The EMH is based on the idea that market prices fully reflect all available information at any given time. If the EMH holds, then the price of an asset should reflect its true underlying value, making it impossible to consistently achieve above-average returns through trading. The EMH is usually tested using time series analysis techniques, and one of the key assumptions of these techniques is the stationarity of the data.

Stationarity is a key property of time series data that is often used in financial modeling. A stationary time series is one where the mean, variance, and autocorrelation structure do not change over time. This is important in finance because many models, such as the CAPM and the Black-Scholes option pricing model, assume that returns are stationary. However, the EMH suggests that stock prices follow a random walk process, which implies that they are non-stationary. As a result, researchers have used statistical tests such as the Augmented Dickey-Fuller Test (ADF) test to determine whether stock prices are stationary or not. If stock prices were stationary, it would suggest that the market is inefficient, and that there may be opportunities for investors to profit by exploiting predictable patterns in the data (see [Lee et al., 2010]).

Augmented Dickey-Fuller Test This test is particularly suitable for evaluating stationarity since it falls under the category of unit root tests. The Dickey-Fuller (DF) test involves a time series of ρ values calculated using Equation 2.36, where Y_t represents the value of the time series at time t, X_t^T represents exogenous variables, ϵ_t denotes white noise, and ρ and δ represent estimated parameters.

$$Y_t = \rho Y_{t-1} + X_t^T \delta + \epsilon_t, \qquad t = 1, 2, ..., T \tag{2.36}$$

The stationarity of a series Y can be determined by the value of ρ calculated by Equation 2.36. If $|\rho| > 1$, Y is a non-stationary series, whereas if $|\rho| < 1$, Y is a stationary series. Therefore, the stationarity hypothesis can be confirmed by verifying if the absolute value of the total ρ is less than 1.

The ADF test extends the DF test equation to incorporate higher-order autoregressive processes into the model. The ADF equation, which incorporates a higher-order regressive process into the model compared to the DF equation, can be computed as 2.37.

$$\Delta Y_t = \alpha Y_{t-1} + X_t^T \delta + \beta_1 \Delta Y_{t-1} + \beta_2 \Delta Y_{t-2} + ... + \beta_p \Delta Y_{t-p} + \upsilon_t, \qquad t = 1, 2, ..., T \quad (2.37)$$

After obtaining the t-ratio, it is used to test the null hypothesis (non-stationarity) as in the DF test. The null hypothesis assumes the presence of a unit root, that is, $\alpha = \rho$-1. If the calculated p-value from Equation 2.37 is greater than the significance level and the statistical test value is greater than the critical value, then the null hypothesis is rejected. Consequently, the series is considered non-stationary.

Pairs trading and the mean-reverting processes

Pairs trading is a popular quantitative technique utilized by hedge funds and investment banks to capitalize on financial markets that are out of balance. Although it may not seem as an investment strategy to be properly categorized under the heading of price forecasting, it fully falls under it. The difference is that, unlike pure price prediction strategies, pairs trading models the distance between the prices of two financial assets, considering that if they have maintained a close relationship in the past, any deviation that occurs will be temporary and exploitable in the future from an investment perspective. This type of investment strategy has traditionally been addressed through the concepts of cointegration and mean reversion.

During this strategy, two assets that typically move in the same direction and have a long-term relationship are identified. Assuming that the spread between the two assets is mean-reverting, deviations from the mean can be taken advantage of by taking a short position in the overvalued asset and a long position in the undervalued asset. If the spread eventually converges back to its mean, the investor can close both positions to make a profit.

Pairs trading is considered a financial markets anomaly that can challenge the weak efficiency hypothesis because it only utilizes past prices of stocks to achieve returns above the risk-adjusted return. In more recent research, it is already considered an anomaly in itself [Jacobs and Weber, 2015], but in previous studies it has been well framed within anomalies caused by the violation of the law of one price [Gatev et al., 2006] or with contrarian or mean-reverting strategies [Herlemont, 2004].

Two stocks that have the same expected returns in any state of nature should have the same price according to the law of one price. If not, a situation of arbitrage would arise, that is, the possibility of obtaining a return without assuming risk. The basic concept of riskless arbitrage implies the possibility of replicating the cash flows of a financial asset by a combination of financial assets, in such a way that the price of the combination is similar to that of the original financial asset [Burguess, 2003]. The pairs trading strategy assumes that the chosen pairs maintain a similar relative price. When a divergence occurs, it can be assumed that it is a violation of the law of one price and that this strategy is a way to test this violation [Gatev et al., 2006]. A similar argument is the assumption that the parallel movement of two stocks is based on the theory of arbitrage, which, as mentioned, implies that the stocks are exposed to the same risk factors and that the idiosyncratic factor is zero on average and that therefore they should maintain their relative price relationship. If, at some point, this relationship provided by the APT is broken, it is expected that it will be restored in the medium term [Vidyamurthy, 2004].

The mean reversion of expected stock returns has been studied for decades, starting in the 80s when it was argued that "incremental increases in expected stock returns are compensated for by immediate falls in current stock prices ... resulting in the presence of a short-term component of reversal" [Fama and French, 1988]. And further on: "Stock returns exhibit positive short-term autocorrelation and negative long-term autocorrelation" [Poterba and Summers, 1988].

In the strategy of pairs trading, it is assumed that the tendency of reversal to the average price or return of shares is not that of price or return of stocks, but the relationship between two stocks, measured as the difference or ratio of their normalized prices. It is considered, therefore, a contrarian strategy. The difference with strategies purely of value, in which one expects the price to revert to its intrinsic value according to the valuation model used, is that in pairs trading the values of the two stocks are compensated, allowing simultaneous over valuation or under valuation of both, but maintaining relative value between the titles.

The justification for this anomaly has been reasoned from Behavioral finance with the arguments of over/under reaction, slow learning, anchorages or irrational valuation, among others. From traditional finance, justification has also been given for the reversal to the average price of shares by the reversal to the average return [Spierdijk and Bikker, 2012].

[Krauss et al., 2017] classify pairs trading strategies into three groups. The most mainstream approach is the co-integration approach, which involves co-integration test from [Engle and Granger, 1987]. Co-integration is a concept closely related to stationarity in time series analysis. When two or more non-stationary time series have a long-term relationship, they are said to be co-integrated. In statistical terms, it means that both series have the same integration order or, what is similar, they need the same number of differentiations to become stationary. This implies that they are not independent and any shock to one of the series is transmitted to the others. The second method is the distance approach, which pinpoints pairs of co-moving individuals using a specific distance measure. [Gatev et al., 2006] utilize the Euclidean squared distance between the normalized prices of securities to identify pairs and form a long-short portfolio. The final approach to pairs trading is the time-series approach. [Elliott et al., 2005] propose another time-series approach for pairs trading based on the mean-reverting Gaussian Markov chain, but no backtesting is conducted on any market. More recently, ML techniques are being used to improve all the phases of the pairs trading strategies, as we will see in the empirical chapter of this thesis (chapter 4).

2.2.3. Portfolio Management

Portfolio management is the financial discipline where the basic goal is to establish the optimal weight of each asset by simultaneously maximizing expected return and minimizing risk. Better portfolio optimization models have a more efficient frontier, which can help investors get a greater expected return while taking the same risk. As a result, developing a more efficient portfolio optimization model has become a hot issue in the field of investment management. The main framework which can be considered as a foundation of portfolio management is the seminal work from [Markowitz, 1952], also known as the modern portfolio theory.

The modern portfolio theory (MPT), often known by its mathematical translation as mean-variance optimization (MVO), is a mathematical framework for constructing an asset portfolio that maximizes expected return for a given degree of risk. It is a formalization and extension of the concept of diversification in investment, which holds that having a variety of financial assets is less hazardous than owning only one. Its basic concept is that an asset's risk and return should not be assessed by itself, but rather on how it adds to the overall risk and return of a portfolio. The volatility of asset prices, measured in terms of standard deviation of returns, is used as a risk proxy.

For a portfolio consisting of m assets with expected returns μ_i, let w_i be the weight of the portfolio's value invested in asset i such that $\sum_{i=1}^{m} w_i = 1$, and let $w = (w_1, ..., w_m)^T$, $\mu = (\mu_1, ..., \mu_m)$, $\mathbf{1} = (1, ..., 1)^T$. The portfolio return has mean $w^T\mu$ and variance $w^T \sum w$, where $\sum$ is the covariance matrix of the asset returns; see [Lai and Xing, 2008]. Given a target value μ_* for the mean return of a portfolio, Markowitz characterizes an efficient portfolio by its weight vector w_{eff} that solves the optimization problem:

$$\begin{aligned} w_{eff} &= \arg\min_w w^T \sum w \\ \text{subject to} : w^T\mu &= \mu_*,\ w^T\mathbf{1} = 1,\ w \geqslant 0 \end{aligned} \tag{2.38}$$

If portfolio return variance, rather than standard deviation, were plotted horizontally, the inverse of the slope of the frontier would be q at the point on the frontier where the inverse of the slope of the frontier would be q. On q, the entire frontier is parametric. [Markowitz, 1952] developed a specific procedure for solving the above problem, called the critical line algorithm, that can handle additional linear constraints, upper and lower bounds on assets, and which has been proven to work with a semi-positive definite covariance matrix.

2.3. Machine Learning Background

As we have previously described, since the financial crisis of 2008, many quantitative asset pricing models have failed and, within the value investing discipline, many conventional factors have become unprofitable [Feng et al., 2019]. As a result, relevant market players are seeking for alternatives to standard asset pricing, price forecasting and stock-picking methods.

Many practitioners started building hand-crafted models that can dynamically learn from past data, as popular quantitative elements have become less credible. For many years, investors have relied on econometric approaches, but only a handful have had success using dynamic models based only on these techniques. This might be due to various factors: (1) inherent noise in financial data, (2) the fact that factors can be multicolinear, and (3) that connections between variables and returns can be changeable, non-linear, and contextual. These properties make estimating any dynamic connections between possible predictors and expected returns problematic for traditional models.

Given those limitations of hand-crafted methods, the use of ML techniques that automatically learn the best features from data has become widespread to avoid them. [Gu et al., 2020a] made the effort to concentrate all the elements involved in the ML techniques to give a context-specific definition of ML: “A diverse collection of high-dimensional models for statistical prediction, combined with so-called regularization methods for model selection and the mitigation of overfitting, and efficient algorithms for searching among a vast number of potential model specifications”.

These kind of methods have proven to be extremely successful in other disciplines (like image processing or NLP). Part of that success is due to their ability to generalize to unseen data, learn from noisy distributions, and automatically learn features for complex data. In other words, they excel in those areas where classical domain-specific approaches crafted by experts have found limitations. This meteoric rising of ML applications is mainly due to the fact that, over the last decade, a series of enhancements have enabled recent advances in practical ML and unlocked its utility: increased computing power, increased data availability, and novel optimization techniques and architectural breakthroughs.

The most promising ML applications in finance are on the buy-side, focusing on finding predictive signal among the noise and capturing alphas [8]. Some example applications are: time-series forecasting, market segmentation, regime-switching detection and, of course, asset management, which is the main issue of this thesis.

Nevertheless, as [Arnott et al., 2019] points out, it is important to bear in mind the very special characteristic of financial markets, since they reflect the actions of people,

[8] The alpha component is, according with the different factor models we have exposed, the independent term which is not associated with any factor of risk and, supposedly, can be associated with the skill of the investors to find extra returns in the securities they invest in.

which may be influenced by others' actions and by the findings of past research. Unlike other scientific disciplines, research can influence future actions of economic agents. In many respects, the problems that ML faces are just a continuation of the long-standing concerns that quantitative finance experts have always confronted. While investors must exercise caution, perhaps even more caution than in previous implementations of quantitative methods, these new tools have a wide range of financial applications.

ML has been successfully applied to virtually any existing scenario where data are available and useful information can be learned from it. However, different techniques must be applied depending on the problem at hand. There are three main paradigms which are characterized by the nature of the problem: SL, UL and RL, which will be analyzed in the three next sections. In the last section, we will introduce the basics of two techniques that can be considered as crucial in the data analysis and ML pipeline, aimed at preparing the data for effective analysis and model building: Data Preprocessing and EMD.

2.3.1. Supervised Learning

The dataset D contains samples x_i together with their expected outputs y_i, such as a dog image together with its label "dog", so $D = \{(x_1, y_1), ..., (x_n, y_n)\}$. Therefore, a function f maps each sample to its output $f(x_i) = y_i$. The goal of supervised learning is to find the best function approximation $g \approx f$ that meets two important criteria: (1) minimizes the difference between known samples $g(x_i) \sim f(x_i) = y_i$, and (2) generalizes properly to samples x_o which are not part of D.

The following are the the most prominent classical supervised learning methods employed in finance:

- Least Squares [Levenberg, 1944]: A method typically employed to find a linear regression by finding the best fit in the least squares framework, that is, minimizing the sum of squared residuals.
- LASSO [Tibshirani, 1996]: A form of linear regression that is characterized for using shrinkage. This means that LASSO performs L1 regularization to penalize the absolute value of the magnitude of the coefficients. As a result, typically a sparse set of coefficients is produced by helping reduce overfitting and model complexity. Ridge regression works in a similar fashion but by enforcing L2 penalties, which does not produce sparse models.
- Regression Trees [Elith et al., 2008]: A decision tree is an ML architecture that uses a flowchart-like structure to arrive to infer a result by taking tests over input variables. Each node of the tree is a test on an input variable and depending on the outcome, the flow continues in one branch of the tree or another until the flow reaches the leaves where final outputs are given. Regression trees are just an

extension of decision trees where the target value to predict takes the form of a continuous value.

- Random Forest [Breiman, 2001]: A classification or regression method that works by constructing multiple decision trees at training times. That multitude of trees constitutes an ensemble to produce a final prediction (e.g., in the case of classification by voting and in the case of regression by averaging the outputs of all trees).

- Support Vector Machines [Cortes and Vapnik, 1995]: Binary classifiers that map the training samples to points in another space to maximize the gap between the two categories. They can also perform non-linear classification using specific kernels which map those inputs to high-dimensional feature spaces where non-linear decision boundaries can be tackled. Intuitively, an SVM finds a hyperplane that optimally separates the decision boundary by maximizing the distance between one class and another. They can also be used for regression with the appropriate modifications (namely, SVRs).

Apart from the classical algorithms, we also briefly explain the modern architectures that are more popular mainly due to the advent of deep learning:

- Neural Networks (NNs) are the most basic architecture, usually composed of individual perceptrons which are arranged into multiple layers—usually with non-linear activation functions interleaved—of varying width. A perceptron [Rosenblatt, 1958] is a function f that maps an input x to generate an output z in the following way:

  $$z = f(x) = \sigma(wx + b), \tag{2.39}$$

  where w is a vector of weights, b is a bias, and σ is an activation function.

  In its most simple form, the activation function is just a threshold and the perceptron is just a binary classifier. Note that the bias simply shifts the decision boundary away from the origin. Single-layer perceptrons can be combined together to form a Multi-layer Perceptron (MLP). This architecture is usually composed of three layers: the input layer as before, a hidden layer, and an output one. The input layer remains as before, but the hidden and output ones can be composed by an arbitrary number of nodes (also named neurons). Each of those nodes is a single-layer perceptron that uses a non-linear activation function. Deep Neural Network (NN) (also called fully connected networks) often refer to MLPs with more hidden layers l. In this general case, the output of a certain neuron i of a layer l can be defined as follows:

  $$z_i^l = \sigma^l(w_i^l * z^{l-1} + b_i^l) \tag{2.40}$$

  Vanilla NNs are capable of learning any non-linear function (they are universal approximators) given enough network complexity, but they face a number of challenges: (1) due to their fully connected nature, they require a huge number of parameters, (2) are usually harder to train, (3) they lose spatial information of the input, and (4) there is no built-in mechanism for capturing sequential data.

- Convolutional Neural Networks (CNNs) [LeCun et al., 1998] uses learnable kernel filters to extract relevant features from the inputs by applying the convolution operation with them. They are especially useful with structured data and in those cases where spatial information is important. Typically, they are currently applied to process 2D images (although a convolution can be applied to any dimensionality). For instance, for the 1D case, we can formulate the output of a single neuron in a CNN as follows:

$$z_i^l = \sigma \sum_k w_k^l x_{i-k}^{l-1}, \tag{2.41}$$

  where w_k^l is a vector of weights, also named the kernel, with k elements. This kernel is convoluted over the adequate portion of the input x_{i-k} and passed through a non-linear activation function to compute the output of that neuron. As we can observe, a CNN shares this kernel across the whole layer and, by doing so, it does not fully connect each neuron from the previous layer to the next ones. Furthermore, each layer can have more than one kernel; each one is convoluted individually to produce a separate output. These outputs are often referred to as feature maps and are stacked to form a multi-channeled output.

  With regards to fully connected NNs, they sport some advantages: (1) as we mentioned, by convolving the input with filters of predefined size instead of being fully connected, they capture spatial features, and (2) by not being fully connected, but instead sharing kernel weights across the whole input, they require way less parameters and thus are easier to train and less prone to overfitting.

- Recurrent Neural Networks (RNNs) are specifically designed to deal with sequence data and learn from temporal information. Although internally they can be shaped either as traditional NNs or CNNs, they usually add recurrent connections in their layers, which helps take into account the state from previous sequence elements or temporal instants. They therefore can capture sequential information and share parameters across different timesteps (in a similar fashion as CNNs do spatially).

  Typically, the most general topology is a fully recurrent RNN where the outputs of all neurons are connected to the inputs of all for them. Each one multiplies the current inputs and previous outputs through an activation function. Other relevant topologies are Gated Recurrent Unit Network (GRU) and the widely spread Long-short Term Memory (LSTM).

  GRUs [Cho et al., 2014] features two gating mechanisms: update and reset. The update gate is responsible for determining the amount of previous information that will flow to the next step. The reset gate decides which information from the past timestep to neglect for the current state.

LSTMs [Hochreiter and Schmidhuber, 1997] features three gating mechanisms: input, output, and forget. This triple gate system allows the architecture to model long- and short-term dependencies properly.

As a matter of fact, all vanilla RNNs, GRUs, or LSTMs are able to model arbitrary time dependencies. The problem is, however, computational and numerical: due to the nature of the training process, the required gradients to learn can easily explode (turn to infinity) or vanish (go to zero) preventing any learning. GRUs are a step forward in comparison with vanilla RNNs and the additional gates from LSTMs help even more to control the information flow to avoid those problems.

2.3.2. Unsupervised Learning

The dataset D contains samples x_i without their expected outputs, such as, a set of images of dogs, cats, and mouses but without labels whatsoever, so $D = \{x_0, ..., x_n\}$. In this case, we do not know how f behaves, and therefore we cannot learn the output mapping $f(x_i) = y_i$. Given this setting, the goal of unsupervised learning is to learn a function g that finds patterns or trends in the dataset. For instance, g could be a function that clusters samples in D based on their similarity according to certain features.

Here, we describe the most common unsupervised learning methods, which are often employed as pre-processing steps:

- k-Means Clustering: Usually employed as a pre-processing technique to reduce the number of data points by summarizing them according to their mean expectations. In other words, it takes a number of samples (n) and aims to partition them in some sets (k, where $k < n$) so that the variance within each cluster is minimized. The most common algorithm is the iterative or naïve k-means [Lloyd, 1982]. In the second experiment of this thesis, focused on pairs trading, this will be one of the techniques employed to generate pairs of stocks.

- Principal Components Analysis (PCA) [Pearson, 1901]: Another common pre-processing technique to reduce the number of features while preserving their variance. It does so by computing the principal components of the input data and then using them to perform a change of basis.

It is important to remark Generative Adversarial Networks (GANs) [Creswell et al., 2018]: An architecture in which two networks (generator and discriminator) compete (adversarial); the goal of the generator is to produce samples able to fool the discriminator whilst the discriminator's role is to detect false examples from the generator. In other words, given a training set, this architecture is able to generate new data that statistically resembles the originally provided. Although initially proposed as a form of generative model for Unsupervised Learning, it has now impacted all paradigms.

2.3.3. Reinforcement Learning

The dataset D does not even exist beforehand, but new samples x_i arrive or are generated on the fly, such as, the current state of a chess board in a match after a movement has been performed. A function g which produces an output z_i given a sample and modifies the current state to produce another sample x_{i+1}. At any state or at certain times, we can measure how good this g is behaving according to a predefined criteria. The goal is to learn a g that will maximize (or minimize) such criteria.

RL can be typically subdivided into two main categories [Sutton and Barto, 2018]: model-based and model-free. The first builds an internal model of the possible states, transitions, and outcomes in the environment; the latter does not use any model but rather learns actions/transitions directly from experience at the expense of statistical efficiency.

As we will review later, the most common reinforcement learning algorithm for finance is Q-learning [Watkins and Dayan, 1992] and its deep counterpart Deep Q-learning [Hester et al., 2018]. Q-learning learns a so-called quality or action-value function, which describes how good is it to take a particular action in a determined state. To do so, a table of state-action pairs is kept; this table assigns a scalar reward that defines the quality of the action at a given state. During training, actions are performed either randomly or by looking at the best one in the table. By analyizing the reward after each action, the state-action table can be updated based on the old reward values and the new ones. Deep Q-learning keeps the same procedure, but makes use of deep neural networks to represent the state-action table; it is typically applied in problems in which the option space is so big that defining a state-action table would be too complex and computationally expensive.

Another successful trend of RL is Recurrent Reinforcement Learning (RRL) [Li et al., 2015]. RRL combines Supervised Learning with Reinforcement Learning typically by employing a RNN to learn the representation of hidden states for the RL algorithm, which is normally a deep Q-learning network to obtain the policy that maximizes the reward.

2.3.4. Preprocessing and Feature engineering

Preprocessing and feature engineering are crucial steps in the data analysis and ML pipeline, aimed at preparing the data for effective analysis and model building. These steps involve various techniques and methods to clean, transform, and extract relevant features from the raw data, which can significantly impact the performance and accuracy of the models.

Preprocessing involves cleaning and preparing the data before it is used for analysis or modeling. This step typically includes handling missing values, dealing with outliers, normalizing or scaling the data, and addressing any data quality issues. Missing values can be imputed or removed, outliers can be detected and treated, and data quality issues such as inconsistencies or errors can be addressed to ensure that the data is reliable and accurate for analysis. Proper preprocessing can help to reduce noise in the data and ensure that the data is in a suitable format for further analysis or modeling.

Feature engineering involves extracting relevant features or variables from the raw data that can be used as inputs for ML models. This step aims to identify the most informative and discriminative features that can capture the underlying patterns in the data. Feature engineering can involve various techniques, such as dimensionality reduction, feature selection, and feature extraction. Dimensionality reduction techniques, such as PCA, can reduce the number of features while retaining the most important information in the data, but may be also accomplished using autoencoders. While autoencoders, a class of NN design, need huge datasets and a lot of computer resources, PCA is a universal ML technique. Large and non-linear data are amenable to autoencoders. Feature selection techniques, such as Recursive Feature Elimination (RFE), can identify the most important features based on their contribution to the model's performance. Feature extraction techniques, such as extracting statistical or domain-specific features, can create new features from the existing data that may provide additional insights for modeling.

Effective preprocessing and feature engineering are essential for improving the performance and interpretability of ML models. Properly cleaned and transformed data with relevant features can help to reduce noise, improve model stability, and enhance the model's ability to capture underlying patterns in the data. Additionally, feature engineering can also lead to more interpretable models by selecting or creating features that are meaningful and interpretable in the context of the problem domain.

It's important to note that preprocessing and feature engineering are iterative processes that may require experimentation and fine-tuning. Different techniques and methods may be applied to different datasets or problem domains, and the effectiveness of these steps may depend on the specific characteristics of the data and the modeling objectives. Careful consideration of the data and domain knowledge is necessary to ensure that the preprocessing and feature engineering steps are appropriate for the specific problem at hand and can lead to meaningful and accurate results in the subsequent analysis or modeling stages.

Empirical Mode Decomposition (EMD)

EMD is a data-driven signal processing technique that was introduced by [Huang et al., 1998] as a method for analyzing non-linear and non-stationary data. EMD has gained significant attention in various fields, including finance, due to its ability to adaptively decompose signals into intrinsic mode functions (IMFs) that capture the underlying oscillatory patterns of the data.

The fundamental concept of EMD is to decompose a signal into a finite set of IMFs, which are functions that have specific time scales of oscillations. These IMFs are extracted directly from the data itself without relying on any predefined basis functions or mathematical assumptions. The decomposition process is performed in a data-driven manner, with IMFs extracted through a series of iterative sifting steps that isolate different frequency bands present in the data.

One of the key advantages of EMD is its ability to handle non-linear and non-stationary data, which are common characteristics of financial time series data. Financial markets are known to exhibit complex and dynamic patterns that change over time, making traditional linear methods inadequate. EMD can effectively capture the local oscillatory patterns and nonlinearities present in financial data, making it suitable for financial market forecasting.

EMD also has the property of providing a complete decomposition of a signal, meaning that the original signal can be exactly reconstructed by summing all the IMFs and a residual component. This property allows for a detailed analysis of the different frequency bands present in the data, which can provide insights into the underlying dynamics of financial markets.

Furthermore, EMD does not require any assumptions about the data distribution, making it a non-parametric method that can be applied to a wide range of financial data without the need for complex model assumptions. This flexibility makes EMD particularly useful for analyzing financial time series data, which often exhibit complex and changing dynamics that may not be fully captured by traditional parametric methods.

Despite its advantages, EMD also has some limitations, including the potential for mode mixing, which is the phenomenon where different frequency bands may be mixed together in a single IMF. This can occur when the data contains overlapping frequency bands that cannot be fully separated by the sifting process. Additionally, the EMD process can be computationally intensive and may require careful parameter selection to obtain accurate results.

In conclusion, while EMD is not classified as a ML algorithm, it can be combined with ML techniques to enhance predictive modeling or forecasting. For example, EMD can be used as a preprocessing step to decompose a financial time series data into IMFs, which can then be used as input features for ML models. This combination of EMD with ML can leverage the strengths of both approaches, with EMD extracting intrinsic oscillatory components from the data, and ML algorithms learning patterns from the decomposed IMFs for prediction or classification tasks.

2.4. Performance Criteria

As we will see in Section 4, about Methods, one of the most interesting areas of analysis has to do with the degree of heterogeneity of the performance criteria measures used by researchers in the last five years. The utilization of different measures of return has coexisted with diverse measures of dispersion or volatility and, in some other cases, with ratios of joint measure of risk adjusted return. This is one of the reasons why the reproducibility of this type of research might be questioned. In the next lines we will make a review and description of the main indicators of performance used in the literature of ML applications to asset management: returns, risk/returns ratios, goodness of fit/prediction, risk of loss measures, statistical significance, and accuracy of predictions.

2.4.1. Returns

Average returns appear as the first type of performance measure. Depending on the data frequency, we can find daily, monthly and annual returns as measures of profitability of the different investment strategies the authors propose using ML applications.

- As summary measure of return we can define the annualized rate of return as the Compounded Annual Growth Rate (CAGR) of the portfolio value between two periods separated n years:

$$\text{CAGR} = \left[\frac{P_{t+n}}{P_t}\right]^{(1/n)} - 1 \tag{2.42}$$

  P_t being the investment value in period t and n the number of years between the two periods we want to compare.

- Sometimes, the returns are measured in terms of Excess Returns. That means that the portfolio return is measured in terms of comparison with the risk-free asset or, in general, a benchmark asset that is used as reference. The arithmetic excess return can be expressed as follows:

$$R_A^E = \overline{R}_p - \overline{R}_b \tag{2.43}$$

  where $\overline{R}_p$ is the portfolio return and $\overline{R}_b$ the benchmark return. We can also define the excess return as a geometric measure:

$$R_G^E = \frac{\overline{R}_p + 1}{\overline{R}_b + 1} - 1 \tag{2.44}$$

2.4.2. Risk/Return Ratios

In many cases, the authors who propose new ML techniques in order to find a better fitting or prediction capacity in financial strategies have opted by considering the risk involved in achieving certain returns. Since the 1960s, investors and researchers have known how to quantify and measure risk with the variability of returns, basically using ratios, but no single measure actually looked at both risk and return together.

- The most popular ratio to measure portfolio performance is the SR. Conceived by Bill Sharpe, this measure closely follows his work on the CAPM and, by extension, uses total risk to compare portfolios to the Capital Market Line (CML). It compares the portfolio return with the risk involved in achieving this return, in form of total risk, measured through the return standard deviation, as follows:

$$\mathrm{SR} = \frac{\overline{R}_p - \overline{R}_b}{\sigma_p} \tag{2.45}$$

σ_p being the standard deviation of portfolio returns.

- Differential Return (DR), by contrast, results in an excess return for the portfolio manager that considers risk in the form of standard deviation (the variability of past returns). It is a sort of a modified Sharpe ratio. Here is the formula:

$$\mathrm{DR} = \overline{R}_p - \left[\frac{\overline{R}_b - \overline{\mathrm{RFR}}}{\sigma_{R_b}} * \sigma_p\right] - \overline{\mathrm{RFR}} \tag{2.46}$$

where RFR is the risk-free rate of return.

- When we use the systematic risk measured by the CAPM, instead of total risk, we are referring to Treynor Ratio (TR):

$$\mathrm{TR} = \frac{\overline{R}_p - \overline{R}_b}{\beta_p} \tag{2.47}$$

where β_p is the covariance between portfolio returns and market returns according to CAPM.

- We can find another very popular ratio, the CR, which can be defined as the ratio between the CAGR and its Maximum Drawdown (MDD) which, at the same time, measures the maximum observed loss from a peak to a trough of a portfolio, before a new peak is attained, and can be considered as an indicator of a downside risk over a specified time period.

$$\mathrm{CR} = \frac{\mathrm{CAGR}}{\mathrm{MDD}} \tag{2.48}$$

A very similar approach is achieved by the Sterling Ratio (STR) and the Sortino Ratio (SNR).

- Finally, the Certainty Equivalent Return (CEQ) considers the risk-free return for an investor with quadratic utility and risk aversion parameter λ compared to the risky portfolio and is given by the following equation:

$$\mathrm{CEQ} = (\mu - \mathrm{RFR}) - \frac{\lambda}{2}\sigma^2 \tag{2.49}$$

2.4.3. Goodness of Fit/Prediction

The statistical goodness of fit of a model defines how well it fits a collection of data. The disparity between actual values and predicted values under the model in issue is often summarized by goodness of fit measures. Very similarly, goodness of prediction refers to discrepancy between observed values and the values predicted by the model. Every goodness of fit statistic can be defined/used for prediction purposes, and vice versa.

- For linear regression models, R-squared is a goodness-of-fit metric. This statistic shows the percentage of variance in the dependent variable that the independent factors account for when taken jointly. The strength of the link between your model and the dependent variable is measured by R-squared, which is defined between 0 and 1. It can be calculated as follows:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{2.50}$$

  where y_i is the actual i-observation of dependent variable y, $\hat{y}_i$ its estimated value, and $\bar{y}$ its mean value. When the estimated values are substituted by the predicted ones, we are talking about Out-of-the-Sample R Squared (OOS R2), a measure of goodness of prediction.

- Mean Absolute Percentage Error (MAPE) is one of the most commonly used performance indicators to measure forecast accuracy. It can be defined as the sum of the individual absolute errors divided by the observed value (each period separately). It is the average of the percentage errors.

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{2.51}$$

  where n is the number of forecast periods. It is a quite well-known indicator among researchers, despite being a poor accuracy indicator. As it can be seen in the formula, MAPE divides each error individually by the observed value, so it is skewed.

- Mean Absolute Error (MAE) is a very useful performance indicator to measure forecast accuracy. As the name implies, it is the mean of the absolute error.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{2.52}$$

  It solves the problem of skewness of the previous indicator but, in return, it is not scaled, so it depends on the magnitude of the dependent variable.

- Root Mean Squared Error (RMSE) is a frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{2.53}$$

  The RMSE serves to aggregate the magnitudes of the errors in predictions for various data points into a single measure of predictive power. RMSE is a measure of accuracy, to compare forecasting errors of different models for a particular dataset and not between datasets, as it is scale-dependent. Actually, many algorithms (especially for ML) are based on the Mean Squared Error (MSE), which is directly related to RMSE.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{2.54}$$

2.4.4. Risk of Loss Measures

Risk of loss measures are typically used by firms and regulators in the financial industry to gauge the amount of assets needed to cover possible losses. But they are also very common in financial research to compare the risk associated with market and credit positions, mainly in the Risk Management discipline.

- Value at Risk (VaR) is a metric for calculating investment risk. It calculates how much a set of assets would lose (with a specified probability) in a specific time period, such as a day, under typical market conditions. According to [Abad et al., 2014], the VaR is thus a conditional quantile of the asset return loss distribution. Let $r_1, r_2, r_3, ..., r_n$ be identically distributed independent random variables representing the financial returns. Use $F(r)$ to denote the cumulative distribution function, $F(r) = Pr(r_t < r \mid \Omega_{t-1})$ conditionally on the information set Ω_{t-1} that is available at time $t-1$. Assume that r_t follows the stochastic process:

$$
\begin{aligned}
r_t &= \mu + \epsilon_t \\
\epsilon_t &= z_t \sigma_t \quad z_t \sim iid(0,1)
\end{aligned}
\tag{2.55}
$$

  where $\sigma_t^2 = E(z_t^2 \mid \Omega_{t-1})$ and z_t has the conditional distribution function $G(z)$, $G(z) = Pr(z_t < z \mid \Omega_{t-1})$. The VaR with a given probability $\alpha \in (0,1)$, denoted by VaR(α), is defined as the α quantile of the probability distribution of financial returns:

$$
\begin{aligned}
&F(\mathrm{VaR}(\alpha)) = Pr(r_t < \mathrm{VaR}(\alpha)) = \alpha \quad \text{or} \\
&\mathrm{VaR}(\alpha) = inf(v \mid Pr(r_t \leq v) = \alpha)
\end{aligned}
\tag{2.56}
$$

  This quantile can be estimated in two different ways: (1) inverting the distribution function of financial returns, $F(r)$ and (2) inverting the distribution function of innovations $G(z)$, in which case is also necessary to estimate σ_t^2.

$$
\mathrm{VaR}(\alpha) = F^{-1}(\alpha) = \mu + \sigma_t G^{-1}(\alpha) \tag{2.57}
$$

- Conditional Value at Risk (CVaR), also known as Expected Shortfall (ES), is a risk measure derived from the previous one. The ES at the $\alpha\%$ level is the expected return on the portfolio in the worst $\alpha\%$ of cases. ES is an alternative to VaR that is more sensitive to the shape of the tail of the loss distribution.

The estimation of risk measures has recently gained a lot of attention, partly because of the backtesting issues of VaR and CVaR related to elicitability. As [Pitera and Schmidt, 2018] mention, "once the parameters of a model need to be estimated, one has to take additional care when estimating risks". The typical estimations approaches, very often, introduce a bias which leads to a systematic underestimation of risk.

2.4.5. Statistical Significance

Many times, researchers are interested in checking whether the variables, factors or characteristics included in the models they propose are statistically significant. In order to achieve this result, they make the usual statistical hypothesis tests based in the t-student distribution. According to this, t-student statistic and p-value are the two most common measures to check the statistical significance.

A result has statistical significance in hypothesis testing when it is extremely improbable to have occurred given the null hypothesis. The significance level of the study rejecting the null hypothesis, represented by α, is the probability of the study rejecting the null hypothesis if this is true.

- Most times, hypothesis testing of statistical significance can be run using a t-student distribution. The t-statistic can be expressed this way:

$$t = \frac{\overline{X} - \mu}{\hat{\sigma}/\sqrt{n}} \tag{2.58}$$

  where $\overline{X}$ is the sample mean from a sample $x_1, x_2, \ldots, x_n$, of size n, $\hat{\sigma}$ is the estimate of the standard deviation of the population, and μ is the population mean. When the t-statistic value is higher, in absolute value, than the critical value of the t-student distribution given a significance level α, the null hypothesis can be rejected, and it can be affirmed than the coefficient or loading is statistically significant, and the variable, factor or characteristic can be considered as relevant.

- The p-value in hypothesis significance testing is the probability of getting test findings that are at least as extreme as the actual results, assuming that the null hypothesis is valid. A tiny p-value indicates that under the null hypothesis, such an extreme observed result would be very implausible. P-values of statistical tests are commonly reported in academic articles in a variety of quantitative domains.

$$\text{p-value} = Pr(T \geq t|H_0) \tag{2.59}$$

  for a one-sided left-tail test, being H_0 the null hypothesis. In a formal significance test, the null hypothesis H_0 is rejected if the p-value is less than a predefined threshold value α, which is referred to as the significance level. The meaning is equivalent to that in which the t-student statistic is higher than the critical value at a given significance level.

2.4.6. Accuracy of Predictions

In the case of price forecasting and algorithmic trading techniques, performance of the selected classifiers is evaluated using different evaluation metrics. Since the problem is a multi-class classification problem and the distribution of classes is not uniform, therefore it is very common to use accuracy primary classification metrics, namely, precision, recall, and the F-measure.

- Accuracy is a classification metric for evaluating classifiers and can be expressed as:

$$\text{Accuracy} = \frac{\text{\# correct predictions}}{\text{Total \# predictions}} \tag{2.60}$$

- Precision is the skill of the model to classify samples accurately and can be calculated as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.61}$$

  where TP is the true-positive rate and FP is the false-positive rate of the algorithm.

- Recall shows the skill of the model to classify the maximum possible samples, and is represented by the following equation:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.62}$$

  where FN is the false-negative rate of the algorithm.

- F-measure describes both precision and recall and can be represented as follows:

$$\text{F-measure} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.63}$$

3. RELATED WORKS

3.1. Methodology

The revision of literature has been made classifying the papers according to four financial areas. The three first areas are the financial disciplines we have used in previous sections: asset pricing/value investing, price forecasting and portfolio management. The fourth one, algorithmic trading, might be classified, from a conceptual point of view, as an intermediate discipline between portfolio management—since it can be considered as an special type of trading strategy and price forecasting—as this trading strategy has as priority goal to forecast the future direction of prices. Nevertheless, the growing relevance of this new discipline, not only at a practitioner level, but also between the researchers, has driven to consider it as an independent area. Given its special characteristics, as we will expose in Section 3.3, this application field can be considered as one of the most promising financial areas to be supported by ML applications.

However, the usual order we follow in the theoretical background chapter (Chapter 2) is going to be slightly altered in this chapter to adapt to the nature of the empirical testing exercises that will be shown in the Chapter of Methods (Chapter 4). Therefore, we will start with the review of papers related to Price Forecasting, with a special section dedicated to those focused on Pairs Trading strategy. Next comes Algorithmic Trading discipline, very related to this latter one. Following that, we will cover Value/Factor Investing, and Portfolio Management papers. Finally, we will include a very short section with a summary of review papers focused in the four disciplines previously analyzed.

To identify relevant journal articles dealing with ML applications in the four asset management disciplines mentioned above, we followed a search process in EBSCOhost, Google Scholar, Science Direct, SpringerLink and Wiley Online Library databases for the period of 2017–2022 using combinations of keywords "machine learning", "deep learning", "neural networks", "asset management", "portfolio management", "asset pricing", "asset returns", "stocks", "finance", "price forecasting", and "algorithmic trading". The range of our survey spanned not only journals and conferences, but also masters and PhD theses, book chapters and arXiv papers. Nevertheless, when the number of papers was sufficiently large in some of the financial areas under study, a filter of impact index was applied, and only journals in the first two quartiles were considered. Furthermore, we only chose the articles that were written in English. After searching through the databases, we reached a list of around 139 identified papers.

After this first preselection, each paper was assessed on quality. This was achieved by using a variety of quality indicators as the citation count and the impact factor of the journal. The arXiv and SSRN databases were also searched to ensure that the most up-to-date research papers were included in the sample. After this second filter we reached a list of 97 identified journals (see Table 3.1). Finally, and after the assessment of each one of the papers, we focused our research and commentaries in a final number of 79 articles, also summarized in the different tables throughout the chapter.

Table 3.1: Recurring themes and reference count from the literature review.

Themes	References
Price Forecasting	31
Algorithmic Trading	17
Value Investing	18
Portfolio Management	31
TOTAL	97

3.2. Price Forecasting

Return and price forecasting play a main role in modern portfolio theory, in asset pricing models and, from a practical point of view, in the asset management industry. As it was pointed out by [Gu et al., 2020b], because the primary objective of asset pricing is to understand the behavior of risk premiums, return prediction is economically significant. In academic finance, the terms expected return and risk premium have been usually interchanged.

As [Henrique et al., 2019] pointed out, the challenge of predicting asset prices and the search for models and profitable systems is still attractive for both the academia and the financial professionals despite the strong presence of the efficient market hypothesis (EMH) by [Malkiel and Fama, 1970], which defends that most of the financial asset prices follow, statistically, a random walk process, and therefore are almost unpredictable (see section 2.2.2).

The recent literature facing the use of ML techniques has been summarized in Table 3.2 and can be divided, roughly, into two levels. In the first level, the new ML technology has been applied to enhance forecasts made using traditional inputs, such as fundamental accounting data, macroeconomic data, or technical indicators. In other words, improving the results of econometric and time-series analysis using fundamental and technical approaches. In the second level, ML has been used to extract new inputs form alternative data, such as sentiments from news data.

Within the first category, we can cite [Kumar et al., 2018], where the authors analysed various SL techniques for stock market prediction. Specifically, they consider SVM, Random Forest (RF), KNN, Naive Bayesian Classifier (NV), and Softmax. Five models were developed and their performances compared in predicting stock market trends. According to their results, the RF algorithm performed the best for large datasets, while NV showed the best performance for small datasets. Moreover, they found that the reduction of technical indicators reduces the accuracy of each algorithm.

[Lee and Kang, 2020] proposed a novel method for training NNs to forecast the future prices of stock indexes. The main contribution of their work is to use only the data of individual companies -instead of index data- to obtain sufficient amount of data for training NN for the prediction of stock indexes. Their experiments, focused on S&P 500, show that NN trained this way outperform NN trained on stock index data. Specifically, they obtain a 5-16% annual return before transaction costs during the period 2006-18.

To evaluate the predictive capacity of certain popular technical indicators in the technological NASDAQ index, in [Cervelló-Royo and Guijarro, 2020] the authors compared the performance of four ML algorithms: RF, Deep Feedforward Neural Network (DFNN), Gradient Boosted Regression Tree (GBRT) and Generalized Linear Model (GLM). The results show that the RF algorithm beats the other ML algorithms, forecasting the market trend 10 days ahead with an average accuracy level of 80%.

[Nabipour et al., 2020] seeked the reduction of risk in trend prediction using ML and Deep Learning (DL) techniques, and applying 11 ML algorithms to data from the Tehran stock exchange. They used DT, RF, Adaptive Boosting (AB), eXtreme Gradient Boosting (EGB), SVM, NV, KNN, Logistic Regression (LR) and Artificial Neural Network (ANN) as ML algorithms and RNN and LSTM as DL ones. The analysis findings show that RNN and LSTM beat other prediction models for continuous data by a significant margin.

[Zhong and Enke, 2019] focused their analysis on daily stock market returns, specifically in the SPDR S&P ETF prices. To anticipate the daily direction of future stock market index returns, Deep Neural Networks (DNNs) and ANNs were applied to the full preprocessed but untransformed dataset, as well as two datasets transformed through principal component analysis (PCA). The simulation findings demonstrate that DNNs with two PCA-represented datasets, as well as numerous other hybrid ML methods, have considerably greater classification accuracy than those with the full untransformed dataset.

[Shen and Shafiq, 2020] propose a solution for the Chinese stock market prices prediction which consists of a feature engineering along with a fine-tuned system based on a LSTM model. The feature engineering applied are the Feature Expansion (FE) approaches with RFE, followed by PCA. This proposed solution outperforms the ML and DL-based models in similar previous works.

In [Nikou et al., 2019], the authors want to examine how well ML models can forecast the daily close price data of the iShares MSCI United Kingdom ETF. Four models of ML algorithms are used in the prediction process. The results indicate that the RNN and LSTM DL methods are better in prediction than the other ML methods, and the SVM method is in the next rank with respect to ANN and RF methods.

Table 3.2: Selection of papers for price forecasting. Mean Absolute Error (MAE), Mean Standard Error (MSE), Maximum Drawdown (MDD), Directional Accuracy (DA), Mean Absolute Percentage Error (MAPE).

Author	**Target Market**	**Method**	**Performance criteria**
[Kumar et al., 2018]	Selected Indian stocks	SVM, RF, KNN, NB, Softmax	Accuracy, F-measure
[Lee and Kang, 2020]	S&P 500 stocks	MLP, CNN	Total return, MDD
[Cervelló-Royo and Guijarro, 2020]	Nasdaq 100 stocks	GBM, RF, CNN	Average accuracy ratio
[Nabipour et al., 2020]	Selected Indian stocks	DT, RF, KNN, LR, ANN, RNN, LSTM	Accuracy, F-measure
[Zhong and Enke, 2019]	S&P 500 ETFs	DNN, FFNN	MSE
[Shen and Shafiq, 2020]	Selected Chinese stocks	CNN, LSTM	Overall accuracy
[Nikou et al., 2019]	iShares MSCI UK ETFs	ANN, SVM, RF, RNN, LSTM	MAE, MSE, RMSE
[Minh et al., 2018]	S&P 500 index	RNN, TGRU, LSTM	Overall accuracy
[Ding et al., 2015]	S&P 500 index	WB+CNN	Total return
[Nava et al., 2018]	S&P 500 index	EMD+SVR	MAE
[Akiyoshi, 2020]	S&P 500 index	EMD+ANN,SVM	MAPE, DA, Sharpe ratio, MDD
[Srijiranon et al., 2022]	Thailand index	PCA+EMD+LSTM	F1-Score, DA
[Shi and Zhuang, 2019]	Hong Kong index	SVM, ANN	MAPE
[Khan et al., 2020a]	Selected US stocks	GNB, SVM, LR, MLP, KNN, GBM, RF	Accuracy, precision, recall, F-measure
[Sarmento and Horta, 2020]	Commodity-linked ETFs	OPTICS, LSTM	Sharpe ratio, MDD
[Krauss, 2017]	S&P 500 constituents	DNN, GBRT, RF	Sharpe ratio, Calmar ratio, MDD, VaR
[Kim and Kim, 2019]	S&P 500 constituents	Deep Q-Network	Sharpe ratio, Calmar ratio, MDD
[Han et al., 2021]	Selected US stocks	k-means, DBSCAN, Agglomerative Clustering	Sharpe ratio, Calmar ratio, MDD

[Nava et al., 2018] provide a multistep-ahead forecasting approach that combines SVR with EMD. This approach is founded on the notion that the forecasting work is made simpler by using the time series that have been deconstructed using EMD as an input for SVR. The results of this technique are evaluated in comparison to benchmark models that are often cited in the literature. The findings show that the EMD and SVR combination can forecast the S&P 500 Index from 30 seconds to 25 minutes in advance, greatly outperforming benchmark models. Long-term horizons are better predicted by the low-frequency components, whereas short-term horizons are better predicted by the high-frequency components.

The S&P 500 Index prices were also predicted using ANN and SVM algorithms in the study by [Akiyoshi, 2020], and the predictive models were evaluated using three metrics: MAPE, DA, and trading simulation, which can be considered the real contribution of this work. The outcome revealed that SVM performed better than ANN in each of the five predictions. But, curiously, the trading simulation revealed that ANN and SVM did not reliably anticipate profits, which raises the possibility that they may not be appropriate for doing so in the real world.

An in-depth analysis of the most recent soft computing techniques is provided in the work by [Shi and Zhuang, 2019]. The samples are used to estimate the values of two representative indices and one stock on the Hong Kong market using a variety of ML models, including single and hybrid models. Different models' prediction abilities are assessed and contrasted. Investigated are the effects of the training sample size and stock patterns (namely momentum and mean reversion) on model prediction. The results show that models built on ANNs have the best prediction accuracy. It was also discovered that the pattern and volatility of equities should be considered while determining the ideal training sample size. When stocks show a transition between mean reversion and momentum trend, there might be significant forecast mistakes.

Within the second category, we can cite [Minh et al., 2018]. This study is focused on the financial news as potential factor which causes fluctuations in stock prices. The main contribution of the paper is to propose a novel framework to forecast directions of stock prices by using both financial news and sentiment dictionary, specifically a novel two-stream GRU and Stock2Vec, a sentiment word embedding trained on financial news dataset.

[Ding et al., 2015] suggested a DL method for event-driven stock market prediction. Events are first retrieved from news content and represented as dense vectors using an Neural Tensor Network (NTN). Second, a Deep Convolutional Neural Network (DCNN) is utilized to predict both short- and long-term effects of events on stock price fluctuations. When compared to state-of-the-art baseline approaches, experimental findings demonstrate that their model can enhance S&P 500 index prediction and individual stock prediction by nearly 6%.

[Khan et al., 2020a] utilized algorithms to examine the influence of social media and financial news data on stock market forecast accuracy over a ten-day period. To increase prediction performance and quality, feature selection and spam tweets reduction are carried out on the data sets. Finally, DL is implemented to achieve maximum prediction accuracy, and certain classifiers are ensembled. The highest forecast accuracies of 80.53% percent and 75.16%, respectively, were reached utilizing social media and financial news, according to their findings. RF classifier is found to be consistent and highest accuracy of 83.22% is achieved by its ensemble.

Similarly, [Khan et al., 2020b] sought to know whether public opinion and political environment in Pakistan on any given day may influence stock market patterns in individual firms or the whole market. Ten ML algorithms were applied to the final datasets to predict the stock market future trend. The experimental findings suggest that the sentiment feature increases ML algorithm prediction accuracy by 0–3%, whereas the political situation feature improves algorithm prediction accuracy by around 20%. The Sequential Minimal Optimization (SMO) algorithm was identified to have the best performance.

To anticipate the closing price of the Thai stock market one step ahead of time, a hybrid prediction model called PCA-EMD-LSTM that combines PCA, EMD, and LSTM was developed by [Srijiranon et al., 2022]. Based on financial and economic news utilizing FinBERT, news sentiment analysis was also used in this study to enhance the performance of the proposed framework. Data on the Thai stock market's price acquired between 2018 and 2022 was evaluated, and a number of statistical indicators served as the assessment criterion. The collected findings demonstrated that, when compared to standard approaches for forecasting stock market price, the suggested framework produced the best results. Adopting news sentiment analysis can also improve the performance of the initial LSTM model.

With the work by [Han et al., 2021] we initiate the review of Price Forecasting studies focused in the popular trading strategy of pairs trading, also known as statistical arbitrage. As we explained in Section 2.2.2, although it may not seem like an investment strategy that can be categorized under the heading of price forecasting, it fully falls under it. The difference is that, unlike pure price prediction strategies, pairs trading models the distance between the prices of two financial assets, considering that if they have maintained a close relationship in the past, any deviation that occurs will be temporary and exploitable in the future from an investment perspective. In their study, [Han et al., 2021] select stock pairs by combining firm characteristics as well as price information, in contrast to typical pairs trading algorithms that identify pairs based on return time series. Firm characteristics are made public in order to give crucial details for pair identification and significantly raise the effectiveness of the trading method for pairs. The long-short portfolio created using the agglomerative clustering generates a statistically significant annualized mean return of 24.8% and a Sharpe ratio of 2.69 when applied to the US stock market from January 1980 to December 2020. After taking transaction costs into consideration, the technique is still profitable.

In [Sarmento and Horta, 2020], the authors propose the use of UL algorithm called OPTICS in order to find profitable pairs. The findings show that the recommended strategy can perform better than the typical pairs search techniques, with an average portfolio Sharpe ratio of 3.79 as opposed to 3.58 and 2.59 achieved by conventional methods. They also present a forecasting-based trading approach that can reduce the times of portfolio drop by 75%. But doing so results in a decline in total profitability. An LSTM, an LSTM Encoder-Decoder, and an Autorregressive Integrated Moving Average (ARIMA) model are used to evaluate the suggested approach. The results of this study are simulated using 5-min price data from a collection of 208 commodity-linked ETFs during various time periods between January 2009 and December 2018, accounting for transaction expenses.

In the same context of statistical arbitrage, the paper by [Krauss, 2017] implements and evaluates the performance of DNNs, GBRTs, RFs, and various ensembles of these approaches. After removing survivor bias, each model is trained using lagged returns for all stocks in the S&P 500. Daily one-day-ahead trading signals were created from 1992 to 2015 based on the chance that a stock will outperform the overall market. The center of the ranking, which is less clear, is censored by converting the greatest k probability into long positions and the lowest k probabilities into short positions. Empirical results are encouraging. Out-of-sample returns greater than 0.45% are produced using a straightforward, equal-weighted ensemble (ENS1) composed of one DNN, one GBRT, and one RF.

Finally, we can refer to the work by [Kim and Kim, 2019], who investigated how Deep Reinforcement Learning (DRL) —particularly with the deep Q-network— may be used in a Pairs Trading scenario and found that the outcomes were good when compared to more conventional approaches. The S&P 500 Index stocks that integrate the pairs are chosen using a traditional cointegration test.

3.3. Algorithmic Trading

It is not easy to classify this discipline. Although it might be classified within one of the areas aforementioned, mainly price forecasting, we have thought that it has some characteristics that may justify to be defined into an independent category.

Algorithmic Trading can be defined as "buy-sell decisions made solely by algorithmic models", as cited in [Ozbayoglu et al., 2020]. These decisions can be based on some simple rules, mathematical models, optimized processes or, as in the case of ML and DL, highly complex function approximation techniques. Market making, inter-market spreading, arbitrage, and pure speculation such as trend following are examples of methods employed in algorithmic trading. Many fall into the category of high-frequency trading (HFT), which is characterized by high turnover and high order-to-trade ratios.

In the case of trend-following methods, many times algorithmic trading applications are connected to price forecasting methods. Consequently, most price forecasting models that generate buy-sell signals based on their forecast are likewise classified as algo-trading systems. Obviously, we will refrain from analyzing those papers focused on price forecasting again. However, most of times algo-trading is coupled with technical analysis, which means that, from a conceptual point of view, this discipline is poorly connected with finance theory and financial economics. Nonetheless, and given the great symbiosis between algorithmic systems and ML techniques, we will make a very brief review of the most interesting papers within this emerging and very popular field among financial market practitioners. This review has been condensed in Table 3.3.

[Sezer et al., 2017] proposed a stock trading system based on optimized technical analysis parameters for creating buy-sell points using genetic algorithms. The optimized parameters were then used to a Deep Multi-layer Perceptron (DMLP) for buy-sell-hold predictions. Daily prices of Dow 30 stocks were used. The results show that this method enhances the stock trading performance.

[Troiano et al., 2018] employed a LSTM based on market indicators, in particular, the Moving Average Convergence and Divergence (MACD) signals, to forecast the trend

Table 3.3: Selection of papers for algorithmic trading.

Author	Target Market	Method	Performance criteria
[Sezer et al., 2017]	Dow 30 stocks	MLP-ANN	Overall accuracy
[Troiano et al., 2018]	Dow 30 stocks	LSTM	Overall accuracy
[Sirignano and Cont, 2018]	Selected US stocks	LSTM	Overall accuracy
[Tsantekidis et al., 2017c]	Selected Finnish stocks	CNNs	Recall, precision, F1
[Sezer and Ozbayoglu, 2018]	World ETFs	CNNs	Annualized returns
[Niño et al., 2018]	Selected US stocks	CNNs	Directional accuracy
[Tsantekidis et al., 2017b]	Selected Finnish stocks	LSTM	Recall, precision, F1

of the Dow 30 stocks' daily prices. Using also LSTM, [Sirignano and Cont, 2018] proposed a novel method that used limit order book flow and history information for the determination of the stock movements. The same approach can be found in [Tsantekidis et al., 2017c].

Due to their effectiveness in image classification problems, several research papers have focused on using CNNs-based models. To do so, however, the financial input data have to be converted into images, which demands some creative preprocessing. It is the case of [Sezer and Ozbayoglu, 2018], who presented a new method for converting financial time-series data containing technical analysis indicator outputs to 2-dimensional images and classifying these images using CNNs to derive trading signals. Using the Limit Order Book Data and transaction data, [Niño et al., 2018] encoded financial time-series into an image-like representation, and get a very good performance in terms of directional accuracy. [Tsantekidis et al., 2017b] proposed a novel method that uses the last 100 entries form the limit order book to create a 2-dimensional image for the stock price prediction.

3.4. Value/Factor Investing

This financial area is very wide and can be considered as the starting point for the rest of interest areas within the asset management discipline. We will consider in this section papers and works regarding the search and location of factors, characteristics, patterns in securities prices which allow the investor to understand the drivers of asset prices, the way the expected returns compensate the assumption of risks, and how to outperform the market. The selection has been summarized in Table 3.4. The logic behind value/factor investing, in general, is that a firm's financial performance is influenced by fundamentals/factors, whether latent and unobservable or connected to fundamental characteristics.

Despite being a decades-old academic topic, value/factor investing has gained popularity in line with the emergence of equity traded funds (ETFs) as investment vectors.

In the decade of 2010, both gained traction. The mutually advantageous feedback loop between practical financial engineering and academic research has encouraged both sides, which is not surprising.

Nevertheless, in the realm of traditional quantitative techniques, researchers have developed in recent years more sophisticated approaches to organize the so-called "factor zoo" and, more crucially, to detect false anomalies and evaluate alternative asset pricing model specifications due to the ever-increasing number of factors and their relevance in asset management. Hundreds of possible candidates have emerged from the search for factors that explain the cross-section of expected stock returns, as noted by [Cochrane, 2011] and more recently by the references [Harvey and Liu, 2019], [David McLean and Pontiff, 2016], and [Hou et al., 2017].

For instance, [Harvey and Liu, 2019] used bootstrap on orthogonalized factors in their regressions to solve the problem of correlations among predictors. [Fama and French, 2018] compared asset pricing models through squared maximum Sharpe ratios, whilst [Giglio and Xiu, 2019] estimated factor risk premia using a three-pass method based on principal component analysis. It is obvious that there does not exist an infallible method, but the majority of new contributions in the field are interested in the search for robustness.

In all the previous cases, the decomposition of returns has been made using linear factor models, of course because of its simple interpretation. Nevertheless, beyond the problem of robustness of the estimations, there has been an eternal debate about whether firms returns are explained by their exposure to macroeconomic factors or simply by their intrinsic characteristics. Characteristics, rather than factor loadings, explain a higher share of variation in predicted returns, according to [Chordia et al., 2019]. On the other hand, adopting a theoretical model in which certain agents' needs are sentiment-driven, [Kozak et al., 2018] reconciles factor-based theories of risk premia.

In all this immense sea of different approaches to the factor investing discipline, and given the exponential increase in data availability, ML techniques make their appearance with the aim to help avoid the mentioned limitations of classical approaches. As we mentioned in Chapter 2, the work from [Gu et al., 2020b] provided a detailed description of ML tools for empirical asset pricing and give their justification for the growing role of ML in financial research. They perform a comparative analysis of ML methods for "the canonical problem of empirical asset pricing: measuring asset risk premiums". The methods they compare are, between others, generalized linear models, dimension reduction tools, Boosted Regression Trees (BRTs), and RFs. In comparison with standard forecasting approaches, they discover that ML tools increase the description of predicted returns. They also highlight that all ML techniques agree on a limited set of main predictive signals, which include variants on momentum, liquidity, and volatility, and that BRT and NN are the top performing techniques. These findings suggest that enhanced risk premium measurement by ML can simplify the examination of asset pricing economic mechanisms and ML is a viable technique for new financial technology.

Table 3.4: Selection of papers for value/factor investing. Mean Absolute Percentage Error (MAPE), Out-of-the-Sample (OOS), Mean Standard Error (MSE), and Maximum Drawdown (MDD).

Author	**Target Market**	**Method**	**Performance criteria**
[Tobek and Hronec, 2020]	NYSE, Amex and NASDAQ common stocks	WLS, PWLS, RF, GBRT, NN	Average return, Sharpe ratio, MDD
[Giglio and Xiu, 2019]	US stocks, T-bonds, C-Bonds and currencies	PCA	R Squared, p-value
[Kelly et al., 2018]	World stocks	IPCA	R Squared, p-value
[Moritz and Zimmermann, 2016]	US stocks	DT	Excess returns, R Squared, MSE
[Kozak et al., 2019]	US stocks	Bayesian and Lasso Regressions	OOS R2, Sharpe ratio
[Messmer, 2017]	US stocks	DFNN	Sharpe ratio
[Feng et al., 2018a]	NYSE, Amex and NASDAQ common stocks	DFNN	Sharpe ratio
[Chen et al., 2020]	US stocks	DFNN, LSTM, GAN	Sharpe ratio
[Feng et al., 2018b]	NYSE, Amex and NASDAQ common stocks	TensorFlow, SGD, AD	MSE, R Squared
[Simonian et al., 2019]	US stocks	RF, ARL	R Squared, Annual return, Sharpe ratio
[Sun, 2020]	NYSE common stocks	Ordered-Weighted LASSO	SR, Mean returns
[Freyberger et al., 2020]	NYSE, Amex and NASDAQ common stocks	LASSO	Sharpe ratio
[Lu et al., 2019]	Chinese stocks	NN, MLP	Average return, Sharpe ratio
[Feng and He, 2019]	US stocks	Bayesian Hierarchical	OOS R2
[Feng et al., 2020]	US stocks	DS LASSO	SR, Mean returns, t-stat
[Sugitomo and Minami, 2018]	TOPIX 500 stocks	SVM, GBRT and NN	Average return, Sharpe ratio, RMSE
[Avramov et al., 2020]	US stocks	NN3, FFN, LSTM, GAN	Average return, Sharpe ratio
[Aw et al., 2019]	US stocks	NNs	Average return, Sharpe ratio
[Gogas et al., 2018]	NYSE, Amex and NASDAQ common stocks	SVR	R Squared, MAPE

The review of specific papers about application of ML techniques to Asset Pricing could start with the contribution mentioned above of [Giglio and Xiu, 2019]. The authors use PCA to solve the problem of bias in the estimation of linear asset pricing models when some priced factors are omitted. They show that in a linear factor model, the risk premium of a factor may be identified independently of the rotation of the other control factors as long as they all cover the actual factor space.

In a similar way, [Kelly et al., 2018] presented the Instrumented PCA, a new cross-sectional modeling technique for equity returns that accounts for latent factors and time-varying loadings by incorporating observable features that instrument for the unobservable dynamic loadings. In this sense, if IPCA tool identifies the corresponding latent factors, that will mean that the relationship between characteristics and expected returns is due to risk compensation. The other way round, if no such factors exist, the characteristic effect will be compensation without risk or "anomaly".

The closest work in methodology and application of ML techniques to [Gu et al., 2020b] comes from [Tobek and Hronec, 2020]. The authors look at out-of-sample returns on over a hundred equities anomalies that have been recorded in scholarly literature. They then demonstrate that ML approaches that combine all anomalies into a single mispricing signal are valuable all around the world and can persist in a liquid universe of stocks. Among others, the techniques used are Weighted Least Squares (WLS), Penalized Weighted Least Squares (PWLS), RFs, GBRTs and NNs.

[Moritz and Zimmermann, 2016] also used an ML approach to look at the cross-section of stock returns. In the context of portfolio sorting, they utilize tree-based models to link information from previous returns to future returns. The authors demonstrate that the traditional linear Fama–MacBeth framework does not take use of all of the data's significant information, and that their ML approach is more robust.

There are two aforementioned contributions which deserve additional analysis. In [Kozak et al., 2018], the authors contribute with their study to the everlasting fight between factors and characteristics, on the one hand, and risk and behavioral explanations to mispricing, on the other hand. They point out that traditional factor models' efforts to summarize the cross-section of stock returns using a sparse number of characteristic-based factors was futile. Moreover, there is just not enough redundancy across the large variety of potential predictors for such a basic model to price the cross-section appropriately. As a result, a SDF model requires a large number of characteristic-based factors to be loaded. ML techniques, and more specifically, the unsupervised statistical technique PCA, helps in this process, which might be useful in future study on the economic interpretation of the SDF. In [Kozak et al., 2019], the authors' method achieves robust out-of-sample performance by imposing an economically motivated prior on SDF coefficients that shrinks contributions of low-variance principal components of the candidate characteristics-based factors. In other words, if the characteristics based models doesn't work well with a very low number of factors, a SDF formed from a small number of principal components performs well.

Without a doubt, the most frequent technique of ML in the literature are NNs. A first example of this kind of approach can be found in [Messmer, 2017]. Based on a very large set of firm characteristics, they use DL techniques to predict the US cross-section of stock returns. Specifically, they train a deep NN and, after applying a network optimization strategy, he finds that deep NN learned long-short portfolios can generate attractive risk-adjusted returns in comparison with a linear model. This result highlights the relevance of non-linearities in the relationship between firm characteristics and expected returns. In the same line of study using DL techniques, [Feng et al., 2018b] designed a deep NN with the aim to minimize pricing errors. As inputs they use firm characteristics, they generate risk factors as intermediate features, and finally fit the cross-sectional returns as outputs. Another example of deep NN can be found in [Chen et al., 2020], where the authors combine three different deep neural network structures in a novel way: a NN to capture non-linearities, a recurrent LSTM network to find a small set of economic state processes, and a GAN to identify the portfolio strategies with the most unexplained pricing information estimate. The primary contributions of this study include the use of the fundamental non-arbitrage condition as a criteria function, the use of an adversarial technique to design the most informative test assets, and the extraction of economic states from a large number of macroeconomic time-series.

The procedure of sorting securities, based on firm characteristics, very usual in factor investing literature, is the starting point of the work by [Feng et al., 2018a], which uses multi-layer deep networks to augment traditional long-short factor models.

Another notable architectures are RFs, which are one of the most used classical techniques of ML in recent years. For instance, [Simonian et al., 2019] showed how to use RFs to produce factor frameworks that improve upon more traditional models in terms of their ability to account for non-linearities and interaction effects among variables, as well as their higher explanatory power. In combination with Association Rule Learning (ARL), they are able to produce viable trading strategies.

[Sun, 2020] proposed a new ML method, the Ordered and Weighted LASSO (OWL), which circumvents complications from correlations between the different factors in the traditional approach. This method can identify and group correlated factors while shrinking off redundant ones. Using Monte Carlo simulations, he shows that OWL outperforms Least Absolute Shrinkage and Selection Operator (LASSO), specially when factors are highly correlated.

Very similarly, [Freyberger et al., 2020] suggested a non-parametric approach for determining which features give incremental information for the cross-section of expected returns. They select features and evaluate how they impact expected returns non-parametrically using the adaptive group LASSO. This technique can manage a high number of factors, has a flexible form, and is not affected by outliers.

[Lu et al., 2019] tried to extract factors according to the definition from Barra team in MSCI company. They utilize Smart Beta Index technique to construct factor indexes to reflect performance and style on the market they analyze, and they bring NNs into the work of cross-section factor integration. Doing so, their experimental results show that the index that compiled based on factors integration by NNs, specifically with MLP, exhibits better profitability and stability.

In [Feng and He, 2019], we can find a Bayesian Hierarchical (BH) approach. This market-timing method uses heterogeneous time-varying coefficients driven by lagging fundamental factors to jointly estimate conditional expected returns and residual covariance matrix, allowing for estimation risk in portfolio analysis. The BH approach also allows to model different assets separately while sharing information across assets. According to the authors' conclusions, the BH approach outperforms alternative methods in terms of prediction for the US market. At the same time, they were able to identify the most important factors in the past decade: size, investment and short-term reversal.

The authors of [Feng et al., 2020] offered a selection methodology to systematically assess each new factor's contribution to asset pricing beyond what a high-dimensional collection of current factors explains. To evaluate the contribution of a component to explaining asset prices in a high-dimensional context, they offer combined cross-sectional asset pricing regressions with the double-selection LASSO of [Belloni et al., 2014]). This model selection phase closely resembles the existing literature's strategy to dealing with the proliferation of asset price factors (e.g., [Kozak et al., 2018]): to take a large set of factors, to apply some dimension-reduction method (LASSO, Elastic Net (EN), PCA, etc.), and to interpret the resulting low-dimensional model as the SDF.

[Sugitomo and Minami, 2018] used a multi-factor model of Fama–French type as a starting point to test if the ML techniques are able to enhance portfolio performance. Specifically, they used a typical method, consisting of SVM, GBRT and NN, and verified the effectiveness and applicability of nonlinear methods in practical operation by comparing it with conventional linear models.

[Avramov et al., 2020] investigated if ML techniques can remove acceptable economic constraints in empirical finance, which is a largely uncharted field. They investigated whether signals generated by ML procedures can withstand economic constraints both in the cross-section and the time-series. For instance, in the cross-section, they remove microcaps and distressed forms, and in the time-series, they look at the sensitivity of investment payoffs to market conditions with less arbitrage opportunities. They concentrate on two DL approaches that perform well with financial data in order to do this job. They first implement NNs with three hidden layers, and then, they incorporate non-arbitrage conditions into multiple connected NNs, including vanilla NNs, LSTMs, and GANs.

A ML factor model using NNs is developed by [Aw et al., 2019]. This model delivers a superior in-sample performance, but a mediocre out-of-sample performance versus a conventional factor model. The reason they point out for this underperformance is that the market noise during the training period overwhelmed the non-linear association uncovered in the ML process. Nevertheless, they defend that the rationality behind investor behaviour explains the ultimate success of new ML techniques in asset management.

In [Gogas et al., 2018], the methodological approach used is SVR, a direct extension of SVM and the objective, to evaluate the effectiveness of four of the most popular models in asset pricing theory, the CAPM, the APT and the three- and five-factor models from Fama and French. They observe large improvements in comparison to the traditional linear regression in terms of the main measures of goodness of fit: R-squared-adjusted and MAPE.

3.5. Portfolio Management

Making decisions about asset allocation is a key component of portfolio management because it allows you to build a portfolio with specified risk and return characteristics. By doing basic analysis using quantitative or textual data analysis and developing new investing strategies, ML approaches can be used to improve this process. ML can assist in addressing the drawbacks of conventional portfolio development methods. Particularly, ML can produce more precise estimations of asset returns and risks and solve portfolio optimization issues with complicated restrictions, resulting in portfolios that perform better outside of samples than traditional methods.

In portfolio management, the concept of diversification is extremely relevant, as we can find in [Markowitz, 1952]. Additionally, very related to diversification, we find the concept of assets correlation. [Goetzmann and Kumar, 2008] argues that while investors are aware of the benefits of diversity, they build portfolios without properly taking the correlations into account. This is the fundamental reason why, while recent and sophisticated portfolio optimization approaches perform well in-sample, they typically perform poorly out-of-sample. For instance, [DeMiguel et al., 2009] proves that equal-weighted allocation, which assigns equal weight to each asset, outperforms the whole range of frequently used portfolio optimization strategies. In the end, every optimization model requires the inversion of a positive-definite covariance matrix, which results in errors of such size that the benefits of diversification are completely neutralized. At this point, ML has an important role to play with regard to the simplification of the problem. The selection of papers on ML for portfolio management has been summarized in Table 3.5.

[Ban et al., 2018] adapted two ML methods with regularization for portfolio optimization. The objective of this technique, known with the acronym Performance-based Regularization (PBR), is to restrict the sample variances of the estimated portfolio risk and return, guiding the solution towards one with less estimation error in the performance. The results show how this technique outperforms all other benchmarks in a proportion of two out of three using Fama–French datasets.

[Rasekhschaffe and Jones, 2019] explains some of the fundamental ideas behind ML and offer a simple example of how investors may use ML approaches to estimate cross-section stock returns while avoiding overfitting. Moreover, in order to demonstrate the benefits of ML techniques to make accurate forecasts, they emphasize the importance of mixing forecasts from several algorithms and training periods for diversification. GBRT, SVM, AB, DNN are some examples of the algorithms used. They demonstrate that, with sensible feature engineering and forecast combinations, ML algorithms can produce results that dramatically exceed those derived from simple linear techniques, such as OLS.

Very similarly, [Huck, 2019] presents a summary of the main techniques that can be implemented to manage a long-short portfolio. He uses three different types of ML tools: Deep Belief Networks (DBNs), RF y EN regression, because they are all able to perform classification tasks as demanded by the trading system he designs. After developing several independent statistical arbitrage strategies based on these three ML methods, the article describes how adding predictors is not a guarantee to increase the performance of the portfolios. Among the tools considered, the RF seems to generate the best performance portfolios.

In [Huotari et al., 2020], the main goal was to look at how modern ML analytics can help portfolio management, specifically by using an ANN-based system to automatically detect market anomalies using technical analysis and exploiting them to maximize portfolio returns by realizing excess returns. They used the Ensemble of Identical Independent Evaluators (EIIE) architecture described by [Jiang et al., 2017] on a sample of 415 stocks from the S&P 500 Index and incorporated selected performance indicators for stock performance in the analysis. They used RL to create an ANN-based deep-learning (multi-layer ANN) agent model for portfolio management (trading model) for this study. A reward function drove the agent model, and the objective was to maximize predicted rewards over time.

[Krauss et al., 2017] implemented and analyzed the effectiveness of several ML methods in the context of statistical arbitrage. Specifically, they used DNNs, GBRTs, RFs and, finally, a combination of them all. Each model was trained on lagged returns of all stocks in the S&P 500, after elimination of survivor bias. The database is comprised of daily data. The empirical findings show that a simple ensemble of the three techniques produces a significant excess of out-of-sample returns.

[Park et al., 2020] proposed a novel long-only portfolio trading strategy in which an intelligent agent is trained to identify an optimal trading action by using Deep Q-learning, on of the most popular DRL methods. Compared with the stochastic programming-based models (Monte Carlo simulations) and heuristic methods (technical analysis), the authors defend that the proposed model, using daily data for two different portfolio cases which comprises Exchange Traded Funds (ETFs) from the US stock market, is a superior trading strategy relative to benchmark strategies.

[Heaton et al., 2017] presented a four-step algorithm for model construction and validation with special emphasis on building deep portfolios. In particular, they introduced DL hierarchical decision models and provided a smart indexing example by auto-encoding the IBB biotechnology index.

[Yun et al., 2020] proposed a two-stage DL framework for portfolio management, which uses LSTM for the prediction model, in addition to a cost function that addresses both absolute and relative return. The proposed methods are evaluated with an ETF dataset, and the empirical results show that the DL two-stage methods outperform ordinary DL models.

Table 3.5: Selection of papers for portfolio management. Value at Risk (VaR), Conditional Value at Risk (CVaR), Maximum Drawdown (MDD), Certainty Equivalent Return (CEQ), Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), Out-of-the-Sample (OOS), Mean Standard Error (MSE), Maximum Drawdown (MDD).

Author	**Target Market**	**Method**	**Performance criteria**
[Ban et al., 2018]	NYSE, Amex and NASDAQ common stocks	PBR	Sharpe ratio, Turnover
[Rasekhschaffe and Jones, 2019]	Stocks from 22 countries	GBRT, SVM, AB, DNN	Excess return, Alpha
[Huck, 2019]	US Stocks	DFN, RF, EN	VaR, Sharpe ratio, Max. Drawdown
[Huotari et al., 2020]	S&P 500 stocks	ANN, EIIE	Sharpe ratio, p-value
[Krauss et al., 2017]	S&P 500 stocks	DNN, GBRT, RF	Return distribution, VaR, Calmar ratio
[Park et al., 2020]	US and Korean ETFs	LSTM, DNN, Q-Learning	Cum. return, Sharpe ratio, Turnover
[Heaton et al., 2017]	IBB Index	Autoencoders	Validation error
[López de Prado, 2016]	Monte Carlo simulations	HRP	OOS variance
[Yun et al., 2020]	World ETFs	PCA, LSTM	IR, MDD, VaR, CVaR
[Raffinot, 2017]	S&P 500 stocks	HRP	IR, Sharpe ratio, MDD
[Jain and Jain, 2019]	NIFTY 50 index	HRP	CVaR, Sharpe Ratio
[Tristan and Chin Sin, 2021]	Singapore Index	AHC-DTW clustering	Cum. Return, Sharpe ratio
[Konstantinov et al., 2020]	World Assets	NN, LASSO regressions	Sharpe ratio, MDD, CEQ
[Xue et al., 2018]	Shanghai ETFs	FFN, IMK-ELN	MAP, MDD, Sharpe ratio
[Wang et al., 2020]	UK Stock Exchange 100 Index	LSTM+MVO	MSE, RMSE, MAPE, MAE, R2
[Ta et al., 2020]	S&P 500 stocks	LSTM+MVO	Sharpe ratio
[Lee et al., 2019]	World equity indices	SVM	Directional accuracy
[Song et al., 2017]	Selected US Stocks	ListNet and RankNet (NN)	Sharpe ratio
[Vo et al., 2019]	S&P 500 stocks	LSTM+MVO	MAE, RMSE
[Ma et al., 2020]	China Securities 100 Index	DMLP, LSTM, CNN	MAE, MSE, MDD
[Ma et al., 2021]	China Securities 100 Index	DMLP, LSTM, CNN, SVR, RF	MAE, MSE, MDD
[Almahdi and Yang, 2017]	US and World ETFs	RRL	Sharpe ratio, Calmar ratio, Sterling ratio
[Aboussalah and Lee, 2020]	Selected US Stocks	SDDRRL	Total return
[Paiva et al., 2019]	Ibovespa stocks	SVM	Average return, st.deviation

In [López de Prado, 2016], we can find a very complete definition and explanation of hierarchical methods, that address the main pitfalls of the Critical Line Algorithm (CLA), the classical quadratic optimization procedure specifically designed by Markowitz in 1954 for inequality-constrained portfolio optimization problems, which was, at the time, a brilliant solution to the generic-purpose quadratic programming models that did not guarantee a correct solution after a known number of iterations. In particular, the author presented the Hierarchical Risk Parity (HRP) method, which is based on graph theory and ML techniques, and used the information in the covariance matrix without requiring its inversion or positive definitiveness. The reason is that this new approach replaces the covariance structure with a tree structure. By using Monte Carlo simulations, the author demonstrates that, despite the CLA method delivering the minimum-variance portfolio, the HRP produces lower out-of-sample variance portfolios.

[Raffinot, 2017] proposes a hierarchical clustering-based asset allocation method, which uses graph theory and ML techniques, in a very similar way to [López de Prado, 2016]. Complete Linkage (CL), Average Linkage (AL) and Directed Bubble Hierarchical Tree (DBHT) are among the hierarchical clustering approaches described and tested, using three empirical datasets from US Stock Market. AL and DBHT prove to be the best clustering methods, and the clustered portfolios to achieve statistically better risk-adjusted performance than commonly used portfolio optimization techniques.

In [Jain and Jain, 2019] we can find research which is also focused on the out-of-sample performance of the portfolios, because it aims to test if there are any covariance matrix forecasting techniques that outperform both traditional risk-based and ML-based portfolios (such as HRP introduced by [López de Prado, 2016]) empirically. According to their results, HRP is less sensitive to bad specifications of covariance than minimum variance or maximum diversification portfolios, while it is less robust than an inverse volatility weighted portfolio. The authors used daily prices from the individual stocks comprising the NIFTY 50 from the Indian stock market.

The approach is very similar in [Tristan and Chin Sin, 2021], because it aimed to use unsupervised time-series clustering-based ML techniques to diversify portfolios and overcome the varied outcomes of industry diversification. Specifically, they used shape-based clustering approach for diversification, the Agglomerative Hierarchical Clustering algorithm (AHC-DTW), applied to the daily prices of the top 82 stocks listed in the Singapore equity market, and was demonstrated to clearly outperform industry diversification.

Another example of the use of hierarchical-based techniques is the work by [Konstantinov et al., 2020], that might be placed in an intermediate point between the financial areas of factor investing and portfolio management. The aim of their work is to approach and compare factor and asset allocation portfolios using both traditional and alternative allocation techniques, considering centrality and hierarchical-based networks, specifically LASSO. The monthly data used comes from the US stock market.

[Xue et al., 2018] developed Incremental Multiple Kernel Extreme Learning Machine (IMK-ELM), which aims to enhance the efficiency of previous algorithms to make classification tasks in robo-advisors services. Specifically, the novel algorithm is able to handle heterogeneous customer information sets. The empirical results, reached through simulation, show that IMK-ELM outperforms other generic classification methods.

[Wang et al., 2020] suggested a hybrid approach consisting of LSTM networks and a Mean-variance Optimization (MVO) model for optimum portfolio construction in combination with asset preselection, in which long-term dependencies of financial time-series data may be represented. In this sense, the empirical results show how LSTM clearly outperforms other ML techniques, such as SVM, RF and common DNNs. In the second stage, after selecting asset with higher returns according to that ML technique, the MVO model is applied for portfolio optimization. The monthly data used comes from the UK Stock Exchange 100 Index.

[Ta et al., 2020] implemented ML techniques at a double level. First, they use LSTM, an special type of RNN, to forecast stock direction based on historical data. In the second level, and in order to build an efficient portfolio, they make use of multiple optimization techniques, including Equal-weighted modeling (EQ), Monte Carlo simulation (MCS) and MVO. The results show that the LSTM prediction model works efficiently by obtaining high accuracy from stock prediction, and generating portfolios which outperform those obtained using alternative techniques as LR or SVM. The data used in this work are the 10-year daily historical stock prices of 500 large-cap stocks listed on the America Stock Exchange S&P 500.

The approach in [Lee et al., 2019] is very similar to those aforementioned in the sense of using a double-scale framework. In this case, the first level of the trading strategy is based on the effect and usefulness of networks indicators. Using a Vector Autorregression model (VAR) model, they forecasted global and regional stock markets' directions. Once these trend predictions were defined, they were used as inputs for determining portfolio strategies via several ML techniques, such as LR, SVM, and RFs. The research data are daily stock index prices from 10 different countries over a period of 22 years. The empirical results show that the prediction accuracy and profit performances are enhanced with network indicators, and that the SVM approach displays the best performances.

[Song et al., 2017] focused their attention on the area of investors' sentiment. In particular, they showed that learning-to-rank algorithms are effective in producing reliable rankings of the best and worst performing stocks and, according to them, they are able to implement outperforming portfolio strategies which produce risk-adjusted returns superior to the benchmark. The algorithms used with weekly prices and financial news from US Stock market are called RankNet and ListNet, which are SL approaches that rely on NNs and Gradient Descent Optimization (GDO) techniques.

The work by [Vo et al., 2019] enters into the emerging field of Socially Responsible Investments (SRIs). The authors defend that traditional optimization methods for portfolio management are inadequate for this kind of investments, so they propose a new model called Deep Responsible Investment Portfolio (DRIP) that contains a multivariate bidirectional LSTM neural network to predict stock returns for the construction of a SRI portfolio using the MVO model. For the empirical application, they used daily closing prices of all individual stocks contained in the S&P 500 from the past 30 years. The portfolios obtained using this method had a high degree of accuracy and achieved much higher Environmental, Social and Governance (ESG) ratings compared with standard MVO models.

[Ma et al., 2020] used the most common DL techniques to build prediction-based portfolio optimization models. These models start by using DNNs to forecast each stock's future performance. The risk of each stock is then calculated using DNNs predictive errors. Following that, portfolio optimization models are constructed by combining predicted returns and semi-absolute deviation of prediction errors. These models are contrasted against three equal-weighted portfolios, with stocks picked by DMLP, LSTM, and CNN, respectively. Additionally, two SVR-based portfolio models are utilized as benchmarks. As experimental data, this article uses component stocks of the China Securities 100 index in the Chinese stock market. The prediction-based portfolio model based on DMLP performs the best among these models.

The approach is very similar in [Ma et al., 2021], where the authors propose two ML models, specifically RF and SVR, and three DL models, in particular DMLP, LSTM, and CNN, for stock preselection before portfolio construction. Therefore, they incorporates those results in advancing MVO models. As benchmarks, they utilize portfolio models with ARIMA predictions. Once evaluated the models with daily data from the Chinese Stock market index, the results show that portfolio models with RF predictions are the best among the set of models used.

[Almahdi and Yang, 2017] applied RRL techniques to build an optimal variable weight portfolio allocation. For this purpose, they propose a RRL with a coherent risk adjusted performance objective function, based on the expected maximum drawdown, the Calmar ratio. They use as dataset five asset portfolios built with five of the most commonly traded ETFs. The maximized function using this method yields superior return performance than other techniques proposed in the existing literature.

Similarly, using RRL techniques, [Aboussalah and Lee, 2020] aimed to enter into the field of continuous action and multi-dimensional state spaces, and, hence, they propose the called Stacked Deep Dynamic Recurrent Reinforcement Learning (SDDRRL) to build a real-time optimal portfolio. As performance metric, they use the Sharpe ratio. The model was trained and tested with daily data from the S&P 500 index, and the results showed that their model outperforms three different benchmarks, including MVO model.

In [Paiva et al., 2019], again, the two-pass methodology is applied to get a portfolio selection model. First, they apply a ML technique to make stock price predictions, the SVR method. After that, they use the traditional scope of MVO for portfolio construction. They compare the results of this method with benchmarks of the Ibovespa index, and the results are favourable for the proposed technique.

3.6. Review papers

In this section, we will review the recent literature that, in form of survey papers, analyse the recent contributions regarding ML applications to finance in general (General

Surveys) and, later, to specific areas of finance regarding asset management: price forecasting, value/factor investing and portfolio management.

Finance is an extremely diverse field in Economics, that includes so diverse disciplines as Asset and Portfolio Management, Risk Assessment, Fraud Detection or Financial Regulation, among many others. The use of ML techniques in all these fields, in the recent years, has been increasingly relevant.

In the sample selected in this work, spanning the last five years, we have been able to find fifteen papers, as it is shown in Table 3.6. From general to specific, the compilation papers adopting the scope of general applications of ML to finance, in its widest extension, is relatively popular, with four papers in the last five years. Among the specific review papers, which are focused on some of the application areas of finance, the discipline related to price forecasting and time-series prediction counts for more than a half of the total, with eight papers. We can also find one general review about asset pricing and value investing. And finally, the asset and portfolio management discipline and its applications has a main role in two papers of revision, one of them exclusively focused on online portfolio selection. In most cases (12 out of 15), as it can be seen in the second column, the publishing journals were computer science outlets.

Table 3.6: List of review articles in the last five years, both general and specific financial areas. General Survey (GS), Price Forecasting (PF), Asset Pricing (AP), Portfolio Management (PM), Stock Market (SM), Exchange Rates forecasting (FX), Interest Rate forecasting (IR), Criptocurrencies (CC), Commodity Prices (CP), Derivatives (DV), Real Estate (RE). Applied Soft Computing Journal (1), Expert Systems with Applications (2), Artificial Intelligence Review (3), Frontiers of Business Research in China (4), International Journal of Electricity and Computer Engineering (5), International Journal on Emerging Technologies (6), Financial Markets and Portfolio Management (7), ACM Computing Surveys (8), ICEFR 2019 (9). (∗) for financial journals or conferences.

Author	Journal/Venue	Period	Citations	References	Area	Market
[Ozbayoglu et al., 2020]	1	2014-2018	44	196	GS	SM, CC, DV
[Huang et al., 2020]	4 *	2014-2018	13	51	GS	SM, FX, CP
[Tkáč and Verner, 2016]	1	1994-2015	184	425	GS	SM, FX, IR, DV
[Cavalcante et al., 2016]	2	2009-2015	282	144	GS	SM, FX, DV
[Sezer et al., 2020]	1	2005-2019	162	216	PF	SM, FX, CP
[Henrique et al., 2019]	2	1991-2017	101	98	PF	SM
[Bustos and Pomares-Quimbaya, 2020]	2	2014-2018	24	87	PF	SM
[Kamley et al., 2016]	5	2000-2015	13	68	PF	SM
[Jiang, 2021]	2	2017-2019	39	234	PF	SM
[Nti et al., 2019]	3	2017-2019	36	207	PF	SM
[Xing et al., 2018]	3	1998-2016	157	153	PF	SM
[Durairaj and Mohan, 2019]	6	1999-2019	15	46	PF	SM, FX, IR, DV
[Weigand, 2019]	7 *	1994-2018	8	49	AP	SM, IR, DV, RE
[Emerson et al., 2019]	9 *	2015-2018	1	81	PM	SM
[Li and Hoi, 2014]	8	1991-2013	137	246	PM	SM

3.7. Datasets

According to [Arnott et al., 2019], one crucial limitation of ML applications in finance involves data availability. On the one hand, it has been hard to find standardized data sources for finance. On the other hand, deep ML methods usually require large datasets where complex patterns can be extracted while avoiding overfitting [LeCun et al., 2015]; those kind of datasets are hard to generate in finance. Therefore, data plays a key role in any ML application to asset management.

The work on asset management papers related to ML techniques may be categorized based on the kind of inputs used. A significant number of the articles examined employ structured type inputs, for which processing techniques already exist and the relevance of which has been thoroughly researched. The more current ones permit the usage of unstructured data, which is more difficult to analyse and extract valuable data from. In this section, we review both sources of data.

3.7.1. Structured Data

When we talk about structured data we are referring to data which is organized and fits tidily into spreadsheets and relational databases. This kind of data is quantitative and is often displayed as numbers, dates, values, and strings. It is usually stored in rows and columns.

Most of the articles revised in this thesis use this type of structure in their information, which is usually open and prepared for API programming interfaces. The most common is the time series of historical prices, with different frequencies. The preferred periodicity is monthly, but in other cases we can find yearly or daily data. In the case of algorithmic trading, most of High Frequency Trading (HFT) models use intraday data.

Stock Values

Generally, this information is public and free and can be downloaded from the pages of the stock markets. Besides, some companies like Bloomberg and Reuters provide paid services with additional information related to stock prices. In some articles, daily stock information is used, which consists of opening price, closing price, maximum and minimum, as well as the volume of transactions or negotiation. In the case of intraday information, it is very usual to find data related to bid-ask spread.

Macroeconomic Indicators and Financial Information

Taking into account that a large list of papers are focused on asset pricing and factor investing, it is very usual to use corporate information from financial reports to estimate the fair value of the stock. These financial reports are the balance sheet, the income statement and the cash flow statement. Related to them, we can find calculated indicators called financial ratios, which summarize the company's financial and economic situation numerically. Some of these ratios are Debt to Equity, Price to Earnings or Enterprise

Value to Operating Earnings (EBITDA). This information is usually provided by private companies like Bloomberg, Factset or Thomson Reuters.

Many times, fundamental analysis uses macroeconomic indicators to understand how the stock prices are correlated to changes in variables outside the company. Firstly, because depending on the health of a country's economy, one can estimate the earnings growth of a company, and secondly, because the consumption is on the basis of the majority of asset pricing models. These economic indicators are usually free and published by governments and public institutions. However, they have an important pitfall: their low frequency and the relevant lag with regard to the current prices.

Technical Indicators

This category of data is very focused on algorithmic trading, since they are useful for predicting the stock market direction. Technical indicators are not based on economic or financial models. Traders that utilize technical analysis use heuristic or pattern-based signals generated by the price, volume, and/or open interest of a security or contract. Technical analysts utilize indicators to forecast future price changes by evaluating previous data.

There are two types of technical indicators: trend indicators and oscillators. The former ones are focused on identifying movement directions, and the latter ones on finding the turning points in the time-series.

All of then can be calculated using the information of prices obtained in the first section, or directly downloaded from the same information sources. Bloomberg is the main source for this type of data.

3.7.2. Unstructured Data

Unstructured data, unlike structured ones, have no predefined construction or systematization. It comes in text form, audio, images, videos and can be challenging to analyze.

The utilization of unstructured data increases the complexity of stock market forecasting, but at the same time opens a whole new world of possibilities. In order to be utilized as an input for the model, this data must be preprocessed and transformed to categorical or numerical data. Text mining algorithms, which extract news segments or views from social networks and may create numerical representations, are required for textual unstructured inputs.

The news analysis is usually taken from media sources, but sometimes from the same company. In the case of social networks, we are talking about a very new, challenging and complex world. In this case, the main problem is the enormous volume of information as well as the computational challenges.

News feeds, social media, earnings call transcripts, multiple CRM platforms, email, call notes, and other unstructured data sources are common in finance. The attractiveness and added value come from a substantially better information base, which includes unstructured data for decision-making that is both relevant and timely.

3.7.3. Analysis

In order to get a more complete perspective about the type of datasets which are used in the research about ML applied to finance, we have selected a small sample of articles within the three areas of research we have used in previous sections. To obtain this sample, we have applied a double filter: firstly, to be published in a Q1-Q2 journal, and, secondly, to be classified in the first quartile in terms of number of citations.

Value/Factor Investing

In this particular field of financial research, it is very common to use public and shared datasets, many times constructed, updated and fed by the own authors. The most usual data frequency is monthly.

- [Kozak et al., 2018]. The methodology used is a very good example of how this type of datasets can be a good way to generate results that can be globally interpreted. In this work, the authors use, firstly, the 5×5 size and book-to-market (B/M) sorted portfolios of [Fama and French, 1993]. Secondly, they use 15 anomaly long-short strategies defined as in [Novy-Marx and Velikov, 2016] and the underlying 30 portfolios from the long and short sides of these strategies. These two datasets are available in two websites fed by the own authors, with the aim to contribute to future and reproducible research. In the first case, datasets are provided by the Kenneth French's website, which provides downloadable and updated files of Fama/French factors from 1926. In the second case, the author shares the datasets used in his 2016 article. The French's webpage can be considered as one of the best examples of publicly available databases that has become as meeting point of asset pricing researchers.

- [Feng et al., 2020]. In this paper we can find another excellent example of how using publicly available datasets from other authors. In their article, the authors firstly download all workhorse factors in the U.S. equity market from Ken French's data library. Then they add several published factors directly from the authors' websites, including liquidity from [Pastor and Stambaugh, 2003] (Stambaugh's website), the q-factors from [Hou and Zhang, 2015], and the intermediary asset pricing factors from [He et al., 2016]. In addition to these 15 publicly available factors, they follow [Fama and French, 1993] to construct 135 long-short value-weighted portfolios as factor proxies, using firm characteristics surveyed in [Hou et al., 2017] and [Green et al., 2016].

- [Gu et al., 2020a]. Another example of academic database, but in this case accessible by subscription, is the Center for Research in Security Prices (CRSP) of US Stock Databases, an affiliate of University of Chicago, which contain daily and monthly market and corporate action data for over 32,000 active and inactive securities with primary listings on the NYSE, NYSE American and NASDAQ. The research-quality data created by this transformational project spawned a vast amount of scholarly research from several generations of academics. Today, nearly 500 leading academic institutions in 35 countries rely on CRSP data for academic research and to support classroom instructions. The CRSP value-weighted index is one of the most usual equity market benchmarks used in financial research.

Portfolio Management

After analyzing the most cited papers within this discipline, we can conclude that, in most of cases, the datasets are composed exclusively by price data. Just in some cases financial data or news can be found in the articles researched. Sometimes, the closing price is joined by other indicators of prices, as maximum, minimum or opening price. Most times, the assets utilized are stock indices constituents and, eventually, ETF prices. The data sources are Datastream, Bloomberg and, in some specific cases, Ken French's website.

- [Heaton et al., 2017]. Weekly returns data for the components of the biotechnology IBB index in the period 2012–2016. They train the learner without knowledge of the actual component weights. Their goal is to find a selection of investments that outperforms the official index.

- [Krauss et al., 2017]. Monthly and daily returns data for the components of S&P 500 in the period 1989–2015. The data source is the Thomson Reuters Datastream. The goal was to build portfolios following a statistical arbitrage strategy with better performance than the benchmark index.

- [Ban et al., 2018]. Ken French's website mentioned in the section of asset pricing. They collect monthly excess returns for three different data sets, composed of 5, 10 and 49 industry portfolios, in the period 1994–2013.

- [Almahdi and Yang, 2017]. A five-asset portfolio using five of the most commonly traded ETFs from different asset categories. They extract the weekly closing prices for each of the five assets from Yahoo Finance website, for the period 2011–2015.

- [Lee et al., 2019]. Data over a period of 22 years from 1995 to 2016, sourced from Thomson Reuters Datastream database. The dataset is composed by weekly closing prices of 10 global equity indices.

- [Paiva et al., 2019]. Opening, closing, maximum and minimum daily prices of the components of the Brazilian index Ibovespa, from 2001 to 2016. They were sourced from the Bloomberg terminal.

Price Forecasting

In this third discipline, the datasets used in the most cited articles are quite diverse, depending on the methodology implemented.

- [Nikou et al., 2019]. In some cases, historical closing prices are the only reference, where the data used include the daily closing price of iShares MSCI UK ETF, also collected from the Yahoo Finance site.
- [Zhong and Enke, 2019]. In other cases we can find financial and economic factors -as in asset pricing models-. In this paper, the dataset includes the daily direction (up or down) of the closing price of the SPDR S&P 500 ETF as the output, along with 60 financial and economic factors as input features. The daily data is collected from 2518 trading days from June 2003. The data sources are public and free (e.g., finance.yahoo.com).
- [Khan et al., 2020a]. Lastly, we can find some examples of financial news and social media data, perfect examples of unstructured data. The source of stock historical daily prices is the same, Yahoo Finance, but the downloaded data have seven features, from date to closing price, passing by traded volume. Given the methodology used in the article, financial news data are also needed, as well as social media data. In the first case, the authors have used Business Insider because it contains a collection of stock market related news from the most famous world news websites, such as Reuters, Financial Times, and so forth. In the second case, they have utilized Twitter API, implemented in Python, to download desired tweets.

4. METHODS

4.1. Proposal

Despite the extensive use of ML tools, its application in financial markets remains restrained due to specific issues. Primarily, data from financial markets has a low signal-to-noise ratio, making it difficult for ML to differentiate signal from noise. Consequently, models are prone to overfitting. To address this, ML algorithms must be finely tuned with methods such as cross-validation.

In addition, financial markets are not static. Rather, they are in a constant state of evolution. As an example, two decades ago, most stock exchanges in the U.S. were operated by human traders in trading floors; now, most exchanges are electronic. Thus, market conditions of past decades are not a reliable representation of current markets. Consequently, from a data science point of view, the historical data available for predicting future market dynamics is restricted to the recent past.

Lastly, a challenge that is specific to financial markets is the self-correcting mechanism. As practitioners who make predictions in the markets are usually market participants themselves, they can capitalize on predictable signals and thereby reduce or completely eliminate them. This is termed "alpha decay." As a result, financial markets are highly efficient, thus making it hard to predict future prices and anomalies are generally very subtle and hard to detect.

Despite these challenges, there are still areas in finance where ML is applied consistently. In the preceding sections, we have thoroughly examined ML techniques applied, mainly from the academic perspective, to each discipline involved in asset management. The objective of this empirical chapter is to show some illustrative examples of ML applications in relevant areas of asset management, to present representative cases of how ML can help to better understand the behavior of certain markets, more accurately predict the behavior of their prices, better capture the structure of certain investment strategies, or significantly improve the predictive capacity of multi-factor models of financial asset valuation.

In this sense, the first empirical section will address the most general problem within the discipline of asset management, price forecasting. Despite the limitations that the EMH imposes on financial markets regarding the prediction of first-order moments in the prices of financial assets, related to the preeminence of a random walk in their generation process, a significant part of the existing literature on ML applied to finance rests on price forecasting. We will try to check the veracity of this hypothesis in the most developed and potentially most efficient financial market, the US stock market. We will apply the most innovative ML algorithms, from the family of SL, to the daily data of the S&P 500, with the aim of verifying the goodness of prediction through various alternative adjustment measures.

In the second section, we will address the empirical testing of one of the most successful investment strategies in financial markets in recent years, the pairs trading strategy. This strategy can be included within the discipline of price forecasting, specifically within the so-called statistical arbitrage. Unlike pure price prediction strategies, the pairs trading models the distance between the prices of two financial assets, considering that if they have kept a close relationship in the past, any deviation that occurs will be temporary and susceptible to be exploited from the point of view of investment. In this testing section we will use ML algorithms, but in a different way. Specifically, we will carry out specific algorithms from the family of UL techniques, to identify and build pairs of values. Unlike most contributions, which opt for technical indicators as the only way of classification, we will opt for introducing classification tools that take into account financial characteristics of the companies, with the idea that these fundamental characteristics can amplify and guarantee greater consistency in the co-integration of the selected values, and greater success in whatever trading strategy can be selected later. The sample will be monthly data from US stock market.

Connected to the above, the third example of empirical testing will be focused on factor investing models. As we have demonstrated in previous sections, factor models have supposed a real revolution in recent years in terms of explaining which factors or characteristics explain the differences in returns between different assets, different investment portfolios. Among all of them, the three- ([Fama and French, 1993]) and five-factor ([Fama and French, 2015]) models are the most widely used, and have served as a starting point for an entire line of research in this branch of asset management. Far from entering into the established battle in recent years to find new explanatory factors, in our data experiment we will try to demonstrate how the use of ML algorithms can consistently improve the predictive capacity of Fama and French models, approached from a traditional econometric perspective. For this purpose, we will use monthly data from five different sectors of the US stock market, and the already known 5 factors of the Fama and French models, published in the Kenneth French website.

4.2. Empirical Results

4.2.1. Price Forecasting

Introduction

The stock market is a lucrative investment arena, where predicting stock movements can lead to arbitrage and profit opportunities. Price forecasting is the primary goal of every single investor or asset manager. While the Efficient Market Hypothesis (EMH) suggests that stock prices reflect all available information and are unforeseeable (in other words, the behavior of stock prices can be described as a random walk, implying that their movements are inherently unpredictable), evidence from various studies shows that the market is not completely efficient, especially with the use of computational intelligence technologies. Those techniques, collectively known as soft computing ([Devendra, 2008]), such as fuzzy systems, NN, ML, and probabilistic reasoning, utilize the concept of accommodating imprecision, uncertainty, and partial truth in order to systematically and flexibly address real-world problems through progressive and adaptive approaches. Additionally, they have been successfully applied in various fields and can be used to exploit market inefficiencies. Specifically, the ML architecture used in this first empirical approach has been Gated Recurrent Unit Network (GRU) Networks.

Traditional econometric models, such as ARIMA models and other more sophisticated, as the Autorregressive Conditional Heteroskedasticity (ARCH) models, may not be suitable for predicting stock prices as they are designed for linear and stationary processes, while stock prices are inherently nonlinear, non-stationary, and volatile. ML approaches, which are part of soft computing, have gained popularity as they can automatically adapt to data without prior statistical assumptions. However, the misuse of ML can lead to disappointment, and back-testing protocols have been proposed to ensure proper application in quantitative finance.

Another challenge in stock price prediction is multi-scaling, where statistical properties of time-series change with time horizons. Signal decomposition tools, such as Discrete Wavelet Transform (DWT) ([Huang et al., 1998]), borrowed from electrical engineering, have been used to address this challenge. In recent years, Empirical Mode Decomposition (EMD) (see section 2.3.4) has been recognized as an effective method for enhancing financial market forecasting ([Nava et al., 2018]). EMD decomposes the signal into a finite set of nearly orthogonal oscillating components, called IMFs, which are locally stationary and associated with characteristic time-scales and -horizons. This allows for a granular detrending procedure and improved forecasting accuracy.

Technical indicators are frequently used as features in ML algorithms to forecast stock prices. For instance, [Tsantekidis et al., 2017a] show that the use of technical indicators, in addition to historical price data, improves the accuracy of stock price forecasts. Technical indicators are derived from the price and volume of the asset being analyzed and are used to identify trends, momentum, and other patterns that can help predict future price movements. Examples of technical indicators commonly used in financial forecasting include moving averages, relative strength index (RSI), and stochastic oscillators. The use of technical indicators has been shown to improve the accuracy of stock price forecasts compared to models based solely on historical price data.

The proposed research introduces a hybrid framework that combines EMD and GRU Networks to forecast the closing price of two stock market indices one step ahead. We have chosen S&P 500 as the main exponent of efficient financial markets and a representative of a less potentially efficient market as the Spanish Ibex 35, in order to check if the

ML proposed architecture has different degrees of accuracy depending on the levels of efficiency and stationarity. The framework consists of sequential experiments. Initially, GRU is applied to the original series of S&P 500 and Ibex 35 in order to get familiar with them. Progressively, several improvements are introduced hoping to enhance the prediction accuracy. Firstly, EMD is utilized to eliminate the non-stationarity of the series and decompose the closing price of the stock market into several Intrinsic Mode Functions (IMF). GRU is then employed to forecast them as a whole, in a second experiment, and individually, in the third one. Afterwards, ten selected technical indicators are introduced as additional features in order to check if this supplementary information improves the prediction accuracy of the model. In the last three experiments, the prediction values of each IMF are combined to obtain the final closing price of the stock market. The prediction model is then evaluated using different metrics (including and innovative model of trading simulation) and compared to some benchmark univariate models as ARIMA (see [Shi and Zhuang, 2019]).

Main contributions

- Hybrid framekork. A hybrid framework that combines EMD and GRU Networks to forecast the closing price of stock market indices. EMD is utilized to eliminate the non-stationarity of the series and GRUs to forecast efficiently the closing prices.
- Technical indicators. Its use has been shown to improve the accuracy of stock price forecasts compared to models based solely on historical price data.
- Optimized for different time windows. Thanks to the power of data mining and the versatility of ML algorithms, the training and validation windows have been optimized, aiming to find the best temporal prediction window while controlling the possibility of overfitting.
- Multiple accuracy measures. The prediction model is evaluated using different metrics and compared to several benchmark univariate models as ARIMA. After running all the scheduled experiments, the best model of all, using classical performance measures, will be finally tested using a trading simulation model.

Research Data

This research utilized 33 years of daily data on the S&P 500 and Ibex 35 stock indices from March 1990 to March 2023, specifically data of closing, high and low prices by session. The data was obtained from Factset and Yahoo Finance, two of the most relevant financial data providers.

Table 4.1 displays the statistical analysis of the closing price of the two analyzed stock markets, presenting information such as the amount of data included in the closing index, as well as the minimum, maximum, mean, standard deviation, and p-value resulting from the ADF test. The ADF test has been computed using the R Extensions module for SPSS Statistics. The results show a notable difference between the maximum and minimum values for both indices, and the closing prices exhibited high volatility, as evidenced by the large standard deviation. The proposed framework utilizes the ADF test to determine the stationary or nonstationary nature of time series data. As shown in Table 4.1, if the ADF test's p-value exceeds the 0.05 threshold and fails to reject the null hypothesis, it suggests that the dataset is highly volatile and non-stationary. Therefore, both datasets are suitable for use with the EMD method, which is an effective technique for analyzing non-linear and non-stationary time series.

Table 4.1: Descriptive statistics of the closing prices. ADF is the Augmented Dickey-Fuller Test for Stationarity of time series.

Statistic Indicators	**S&P 500**	**Ibex 35**
Count	8,334	8,334
Average	1,520.71	8,157.60
Minimum	295.46	1,873.58
Maximum	4,796.56	15,945.70
Standard deviation	1,012.23	3,119,97
ADF Test (p-value)	0.810	0.541

Table 4.2: Technical indicators and their formulas. C_t is the closing price, H_t and L_t are the high and low prices at time t, HH_{14} and LL_{14} mean highest high and lowest low in the last 14 days, respectively, RS is average gain of last 14 trading days divided by average loss of last 14 trading days, EMA is Exponential Moving Average, SMA is Simple Moving Average, and $Mt : Ht + Lt + Ct/3$, being D_t the mean deviation.

Name of Indicators	**Formulas**
Simple 10-day moving average (SMA10)	$\frac{C_t+C_{t-1}+...+C_{t-10}}{10}$
Weighted 10-day moving average	$\frac{((n)\times C_t+(n-1)\times C_{t-1}+...+C_{10})}{(n+(n-1)+...+1)}$
10-day Momentum	$\frac{C_t-C_{t-10}}{C_{t-10}}$
Stochastic K%	$\frac{C_t-LL_{14}}{HH_{14}-LL_{14}} \times 100$
Stochastic D%	$\frac{K_t\%+K_{t-1}\%+K_{t-2}\%}{3}$
RSI (Relative Strength Index)	$100 - \frac{100}{1+\text{RS}}$
MACD	$\text{EMA}_{12} - \text{EMA}_{26}$
Larry William's R%	$\frac{HH_{14}-C_t}{HH_{14}-LL_{14}} \times 100$
A/D Oscillator	$\frac{\text{A}}{\text{D}_{t-1}} + \frac{(C_t-L_t)-(H_t-C_t))}{H_t-L_t}$
CCI (Commodity Channel Index)	$\frac{M_t-\text{SMA}_{20}}{0.015D_t}$

Additionally, the collected data was pre-processed to generate ten technical financial indicators that were used as input into some of the ML models. These technical indicators were computed using the formulas given in Table 4.2. Finally, it is worth noting that 80% and 20% of the data was allocated for training and testing, respectively.

Evaluation methods

This project assesses prediction models using three evaluation metrics inspired by the work of [Xing et al., 2018], including closeness, accuracy, and trading simulation. The introduction of this third evaluation metric can be considered as one of the main contributions of this empirical approach, since this measure can provide a more realistic and comprehensive evaluation of a a price forecasting tool in the trading strategy's performance.

Additionally, since the time series used in the project are seemingly non-stationary, it is introduced the ADF test. The ADF test is a statistical significance test used to determine whether a time series is stationary or non-stationary (see section 2.2.2).

Closeness In section 2.4.3 the main measures of Goodness of Fit (synonim to Closeness) were presented. In this project the measure selected has been the MAPE, defined as the mean of absolute percentage errors showed in Equation 4.1.

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\text{Actual} - \text{Forecast}}{\text{Actual}}\right| \tag{4.1}$$

Directional accuracy The second metric assessed is the DA, which gauges the performance of prediction models by determining if the chosen stock price will increase or decrease. In the section 2.4.6 the main metrics related to the accuracy of predictions were presented. The directional accuracy presented here is evaluated using four criteria: true positive (TP), false positive (FP), true negative (TN), and false negative (FN). TP denotes the instances where a prediction model correctly guesses a positive movement in the stock price, while TN refers to the situations where the model accurately forecasts a negative change. Conversely, FP indicates situations where the model incorrectly forecasts an increase in the stock price, when it actually drops. FN is when the model incorrectly predicts a decrease in the stock price, but it goes up instead. The directional accuracy is computed using Equation 4.2.

$$\text{Dir. Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{4.2}$$

Trading simulation The third measurement was not presented in the theoretical background in Chapter 2 because it cannot be considered as a truly performance measure from an academic point of view. This is the reason why it will not be used in the central empirical phases of this project, and only as an extension of the experiments driven in this research. Specifically, it is based in trading simulation results by stock predictive models. Although an algorithm may be highly accurate in its predictions, this does not mean that it will be successful in generating profits. To know for sure whether a model is viable in a real-world situation, it is necessary to run simulations and tests on the model to evaluate its efficacy. Based on that, after running all the scheduled experiments, the best model of all, using classical performance measures, will be finally tested using a trading simulation model.

System architecture

This section describes the two algorithms used in this project and the proposed integrated prediction model that combines them to improve prediction effectiveness and reduce computational complexity. These two algorithms are: firstly, EMD, which is a pre-processing algorithm that decomposes the signal into a finite set of nearly orthogonal oscillating components, called IMFs, which are locally stationary and associated with characteristic time-scales and -horizons. And, secondly, GRU, a ML algorithm which is a type of RNN that is designed to address the vanishing gradient problem in traditional RNNs. Similar to other RNNs, GRUs use feedback loops to process sequential data, but they incorporate gating mechanisms that allow them to selectively retain or forget information over time.

Empirical Mode Decomposition (EMD) Originally proposed by [Huang et al., 1998], EMD is a signal processing technique which breaks down a signal into IMFs and a residual component while still operating in the time domain. This makes it a good substitute for other analytical techniques like the Fourier Transformation and the DWT. EMD is frequently used in non-linear and non-stationary scenarios and is particularly effective when evaluating natural signals.

The complexity of the original signal is divided into a series of IMFs with amplitude and a residual difference during the decomposition phase of EMD. The IMFs meets the following two criteria:

- They only have one extreme between zero crossings, which means there can only be a difference of 1 between maxima and minima.
- the mean of the wave of IMFs is zero.

This decomposition technique helps reveal the underlying structure of the signal, making it easier to analyze and predict.

The EMD algorithm performs signal decomposition by sifting the signal into a series of IMFs. This process can be illustrated using the following algorithm, as shown in Figure

4.1, taken from [Srijiranon et al., 2022]. First, a data set $x(t)$ is decomposed into IMFs $x_n(t)$ and a residual $r(t)$, which can be expressed as a sum of IMFs and the residual, as shown in Equation 4.3.

$$x(t) = \sum_n x_n(t) + r(t) \tag{4.3}$$

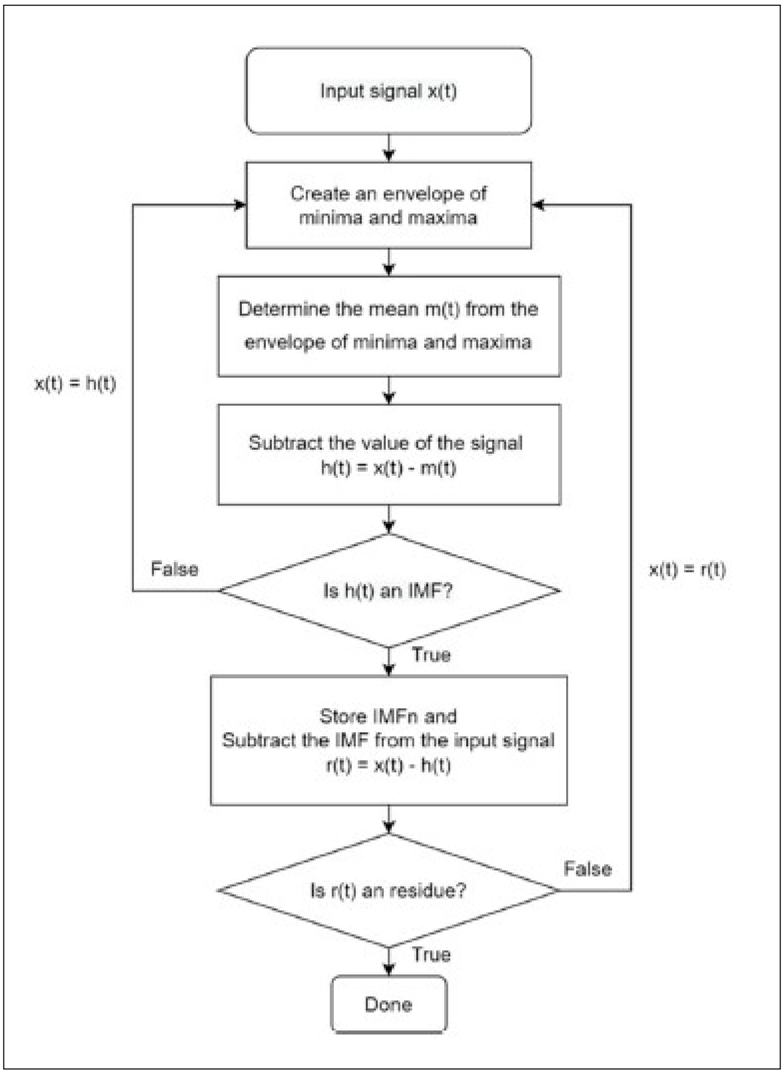

Figure 4.1: A flowchart of shifting processes.

Gated Recurrent Unit Network (GRU) GRUs were introduced in Section 2.3.1 as a special type of RNN. The outstanding new algorithm from the DL family, GRUs [Cho et al., 2014], has two gating mechanisms: update and reset. The quantity of prior information that will flow to the following phase is decided by the update gate. The reset gate determines which prior timestep information to ignore for the present state. The GRU is like a LSTM with a forget gate, but has fewer parameters than LSTM, as it lacks an output gate. This results in a generally faster computing architecture. It was discovered that GRUs performed similarly to LSTM on a few polyphonic music modeling, speech signal modeling, and NLP tasks. The GRUs have demonstrated that gating is effective overall, although it is impossible to say for sure which of the two gating units was superior.

A GRU, which is derived from the LSTM architecture, incorporates an update gate and a reset gate, combining the functionalities of the input gate and forget gate in LSTM (see Figure 4.2). Update gates (z_t), reset gates (r_t), and current memory contents ($\hat{h}_t$) make up an GRU unit. The GRU's last memory is where the output (h_t) is kept. The percentage of the input (x_t) and preceding output (h_{t-1}) that is sent to the following cell is determined by the update gate, which is controlled by the weight (W_z). The amount of prior information to be forgotten is set by the reset gate. The weight (W), which determines the next iteration, guarantees that only pertinent information is transmitted to that iteration. The following equations regulate the primary operations of an GRU:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{4.4}$$

$$r_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{4.5}$$

$$\hat{h}_t = \tanh(W[r_t \cdot h_{t-1}, x_t]) \tag{4.6}$$

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot h_t \tag{4.7}$$

where z_t and r_t are intermediate values obtained from the update and reset gates, respectively; *tanh* is hyperbolic tangent function; σ is the sigmoid function.

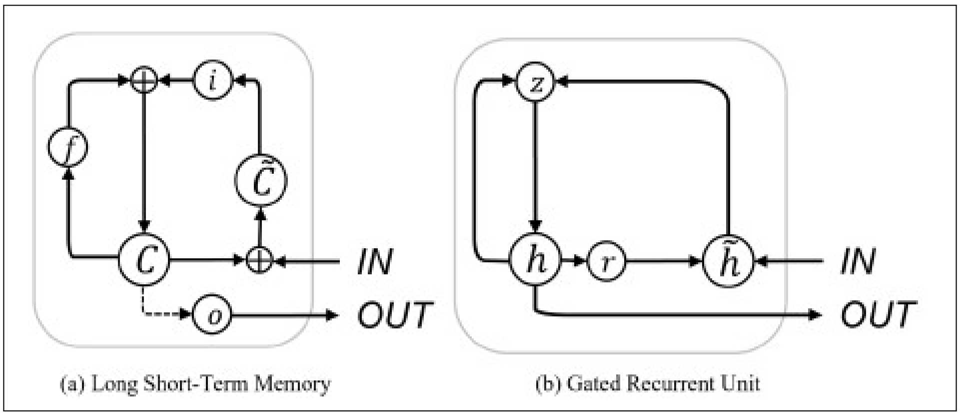

Figure 4.2: Simplified LSTM and GRU diagrams.

Prediction Model In order to understand the framework architecture, Figure 4.3 depicts one of the possible integrated prediction models that combines EMD and GRU, as well as technical indicators as additional features, to improve prediction effectiveness and reduce computational complexity. This model has named as Experiment 4 in the sequential framework defined in this project. The model comprises three main steps:

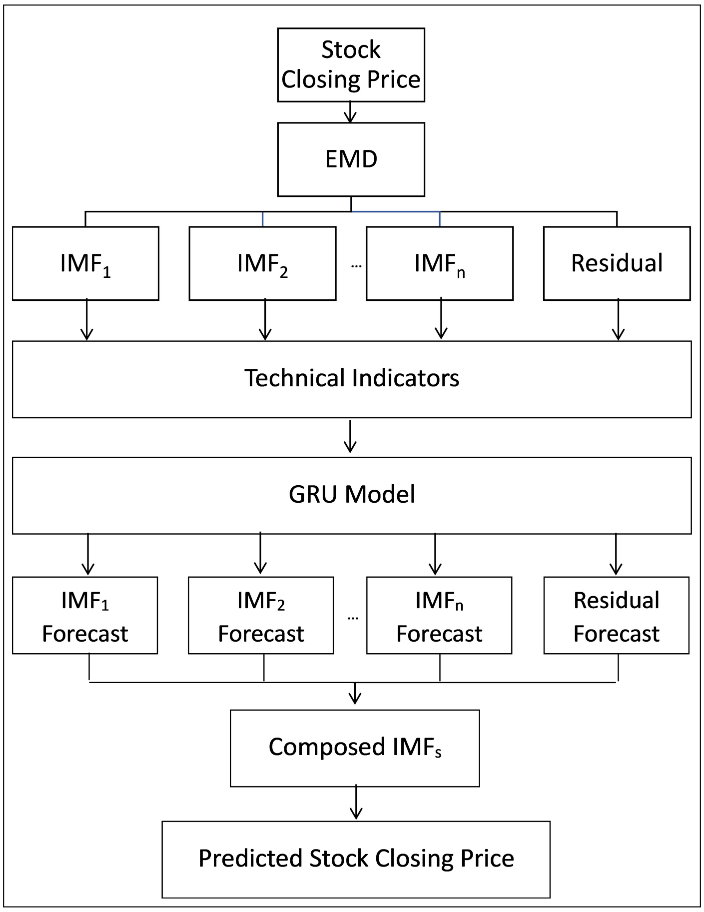

Figure 4.3: System architecture. Experiment 4. The process of the prediction model.

- First, the original stock closing price time series is divided into numerous independent intrinsic mode function (IMFs) components and one residual component using the EMD method. This phase makes it easier to analyze each IMF component using GRU as a prediction model.

- Secondly, the technical indicators are integrated into the model as input features. The GRU model is then employed as the prediction tool for each IMF component. Notably, the GRU is trained separately for each IMF component, with special tuning of network parameters, epochs, and batch sizes for each IMF. The architecture itself is also optimized. This feature distinguishes the hybrid EMD-GRU model from a single GRU model.

- Thirdly, each predicted IMF component is combined to obtain the final predicted stock closing price after obtaining the predicted results of the IMFs. Finally, the performance metrics are used to compare the results of the proposed model with those of other models. Overall, the proposed integrated prediction model that combines EMD and GRU models offers promising results in terms of prediction effectiveness and computational complexity reduction.

Experimental Methods and Results

The effectiveness of the proposed framework for predicting the closing price of the stock market in US and Spain was evaluated in this section, which is divided into different sub-sections, according to the sequential experiments carried out. The first subsection presents the results of the simplest model, a GRU applied directly to the data of S&P 500 and Ibex 35. In next sub-sections, several improvements will be introduced sequentially, in order to find the best prediction model, as the EMD to avoid absence of stationarity in the series, optimized GRUs for each IMF of the decomposed series, or technical indicators as additional information features. The last sub-section presents the comparison results between the different models presented.

The framework consisted of multiple processes in both feature engineering and prediction modeling, so a set of sensitivity experiments was conducted to validate the proposed model's efficacy in each process. The goal is to assess whether the suggested model can significantly outperform cutting-edge stock price time series modeling approaches in terms of prediction outputs. The proposed model was validated using various combinations of EMD, GRU, and financial variables from a number of perspectives. Table 4.3 provides the experimental design. For simplicity reasons, and once that it has been proved in the first two experiments that S&P 500 prices is the series with the greatest issues in terms of stationarity and, therefore, modeling, the remaining experiments have been focused on American stock prices. The time window which has been used by the GRU algorithms has been 10 days back and 1-day ahead. This window will be optimized in Experiment 5.

- Experiment 1 applies GRU algorithm to the original series of S&P 500 and Ibex 35, in order to get familiar with both financial series of prices.
- Experiment 2 aims to solve the non-stationarity problems of series of prices, using EMD technique of pre-processing. A General GRU model will be applied to every single IMF to check if this stationarity pre-processing tool improve the results.
- Experiment 3 tries to improve previous results applying a different GRU model to each one of the IMFs. Taking into account that each IMF shows a very different pattern of stationarity, it might be interesting to model each IMF separately.
- Experiment 4 focuses on comparing the effects of incorporating technical indicators into the EMD-GRU model. This experiment adds the original technical indicators as additional input feature to the proposed model and evaluates whether this information can enhance the prediction performance of EMD-GRU.
- Experiment 5 evaluates the impact in performance of using different time windows (look-backs and look-aheads). It is worth noting that basic window used in experiments 1-4 has been 10 days back and 1-day ahead.

- Experiment 6 evaluates the best model using a different evaluation measure. After running all the scheduled experiments, the best model of all, using classical performance measures, will be finally tested using a trading simulation model.

Table 4.3: Details of the model and input features in each experiment. GGRU: General GRU model for every IMF. IGRU: Individual GRU model for each individual IMF.

Experiment	Model Name	Input Features
1	GRU	Closing price
2	EMD-GGRU	Closing Price
3	EMD-IGRU	Closing Price
4	EMD-TIs-IGRU	Closing Price + Tech. Ind.

Experiment 1: GRU basic model We explored the use of GRUs for prediction in the Ibex 35 and S&P 500 indices, in both cases considering a ten-day input window and predicting one day ahead. Table 4.4 and Figure 4.4 exhibit the performance metrics and the predictive fitting of this model for Spanish Ibex 35.

The model GRU 32-1 shows the best results in terms of MAPE. Nevertheless, the results in terms of DA are quite disappointing, since they are just slightly above the minimum 50%.

Table 4.5 and Figure 4.5 exhibit the performance metrics and the predictive fitting of this model for US S&P 500.

Table 4.4: Performance metrics for Ibex 35. Experiment 1. First component of GRU refers to number of neurons, and second component to layers. MAPE: Mean Absolute Percentage Error, DA: Directional Accuracy. Bold figures for best result. All models were trained for 100 epochs with early stopping using ADAM as the optimizer and a learning rate of 0.01 with a batch size of 128. In all cases, a one-day prediction window was used, considering the previous 10 days. A 20% of the data was reserved for validation.

	Training		**Validation**	
Model	**MAPE(%)**	**DA(%)**	**MAPE(%)**	**DA(%)**
GRU 8-1	1.05	51.35	0.83	50.18
GRU 8-2	1.28	**51.39**	0.83	49.94
GRU 8-3	1.36	51.41	0.83	50.30
GRU 16-1	1.11	51.17	0.83	50.18
GRU 16-2	1.06	51.12	0.84	50.12
GRU 16-3	1.12	51.21	0.83	50.00
GRU 32-1	**1.04**	51.14	**0.83**	**50.54**
GRU 32-2	1.05	51.03	0.831	50.24
GRU 32-3	1.26	50.84	0.85	50.48

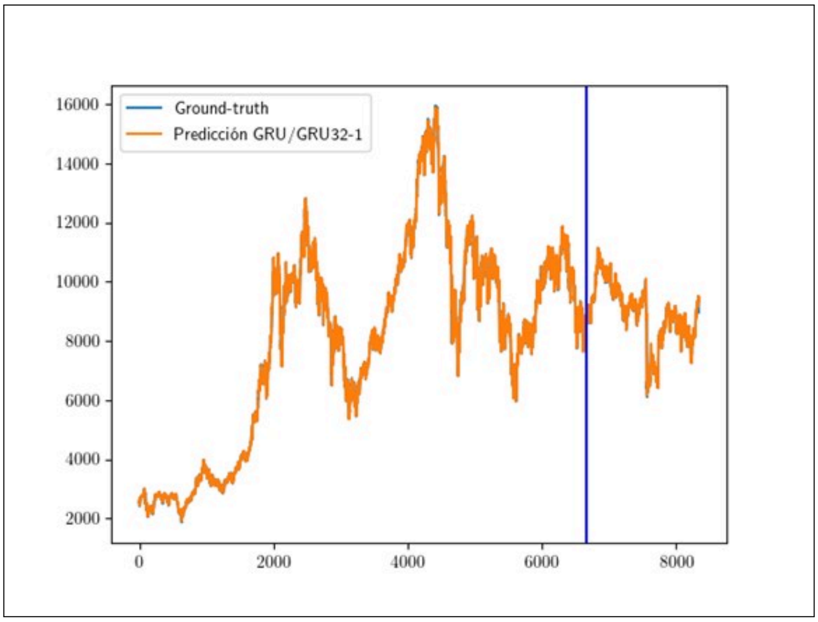

Figure 4.4: Results in Ibex 35 using the selected best model GRU 32-1. Experiment 1. The blue vertical line separates the data into training (left) and validation (right) sets. Ground truth: actual data.

In the case of the S&P 500, once again the model that provides the best closeness in validation is the GRU 32-1, although with slightly worse metrics in the training set and directional accuracy than Ibex 35. It is worth noting that, in this case, the DA does not even reach 50%.

Therefore, it can be concluded that, using only the ML algorithm, it does not seem that we can successfully predict stock market prices, especially in the case of the S&P 500.

Table 4.5: Performance metrics for S&P 500. Experiment 1. First component of GRU refers to number of neurons, and second component to layers. MAPE: Mean Absolute Percentage Error, DA: Directional Accuracy. Bold figures for best result. All models were trained for 100 epochs with early stopping using ADAM as the optimizer and a learning rate of 0.01 with a batch size of 128. In all cases, a one-day prediction window was used, considering the previous 10 days. A 20% of the data was reserved for validation.

	Training		**Validation**	
Model	**MAPE(%)**	**DA(%)**	**MAPE(%)**	**DA(%)**
GRU 8-1	**0.81**	49.02	7.21	48.73
GRU 8-2	0.86	**49.08**	12.97	48.73
GRU 8-3	0.92	48.81	14.50	**49.34**
GRU 16-1	1.10	48.79	5.28	48.61
GRU 16-2	1.09	48.84	8.08	48.37
GRU 16-3	1.69	48.37	12.45	48.43
GRU 32-1	1.20	48.84	**2.78**	48.55
GRU 32-2	2.07	48.31	3.82	48.37
GRU 32-3	2.44	48.31	6.60	48.49

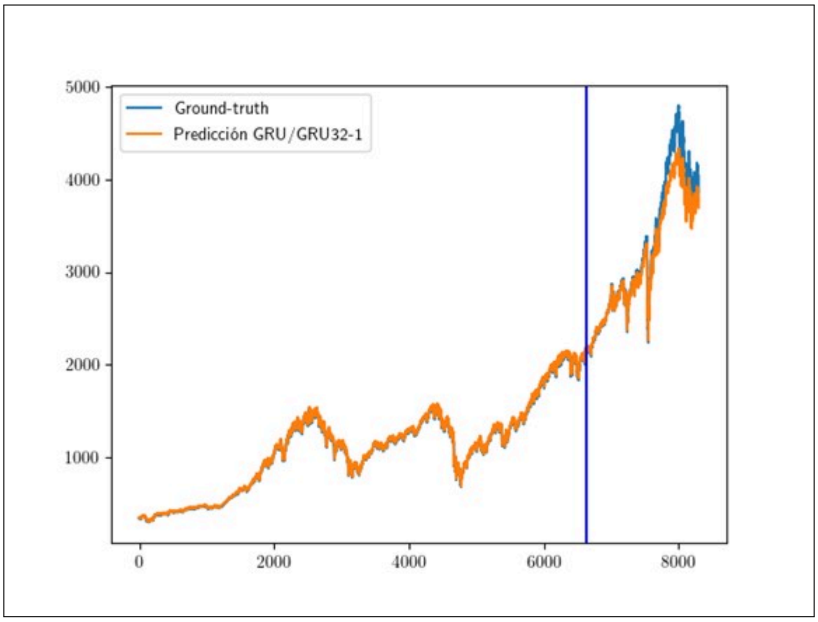

Figure 4.5: Results in S&P 500 using the selected best model GRU 32-1. Experiment 1. The blue vertical line separates the data into training (left) and validation (right) sets. Ground truth: actual data.

The EMH appears to be relentless, preventing any hint of inefficiency in price formation process and bringing it closer to a random walk.

Experiment 2: EMD model + General GRU As we have explained in previous sections, EMD has demonstrated to be particularly useful in analyzing non-stationary situations. Therefore, in this second experiment the closing price of the stock data has been transformed into new data using the EMD method. The experiment depicted in Figure 4.6 demonstrates the decomposition of the original closing price sequence into ten IMFs for the S&P 500 and eleven IMFs for the Ibex 35, of varying frequency scales, using EMD. Therefore, the number of IMFs generated varied depending on the raw data and the market analyzed. The EMD process was repeated until there was only one global maxima and minima value on IMF 10 and 11, respectively, as shown in Figure 4.6. The number of IMFs generated would differ based on the raw data, but applying EMD to the same data would result in the same number of IMFs.

The findings demonstrate that the signal can be categorized into three groups based on their frequency components. The first group comprises high-frequency components with a large amount of noise, represented by the initial IMFs. The second group consists of middle-frequency components with medium noise, represented by the central IMFs. The final group comprises low-frequency components with minimal noise, represented by the latter IMFs. Furthermore, it is assumed that the GRU model is capable of accurately predicting low-frequency IMFs while struggling with high-frequency IMFs. To optimize prediction efficiency, the GRU model is trained separately for each IMF, and the hyperparameters, number of hidden layers, and weights differ for each GRU. This distinction is a key factor that allows the hybrid EMD-GRU model to outperform a single GRU model

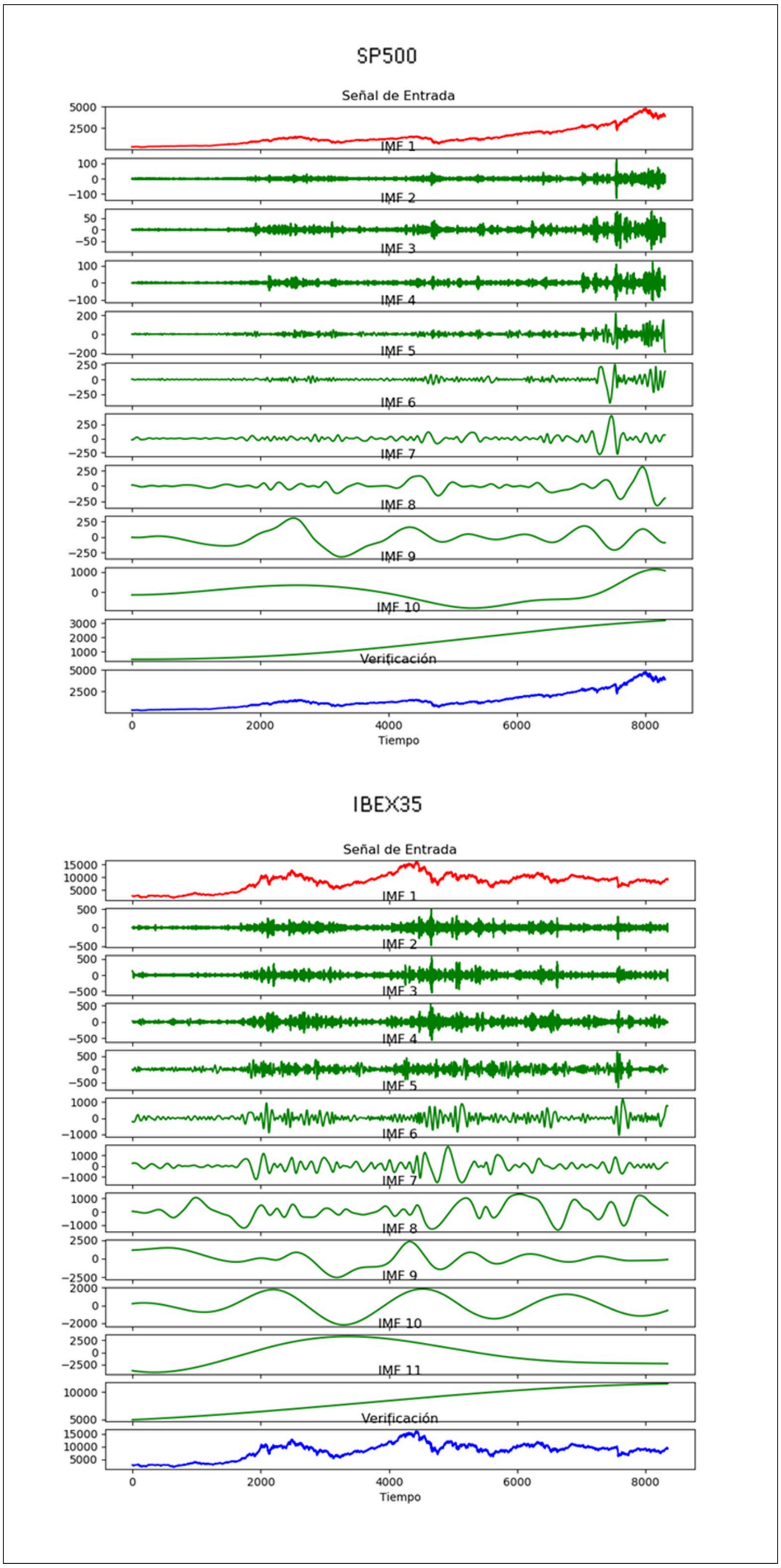

Figure 4.6: The original closing price for SP500 and Ibex 35 and respective decomposed IMFs. Experiment 2.

Table 4.6: Performance metrics for S&P 500. Experiment 2. First component of GRU refers to number of neurons, and second component to layers. MAPE: Mean Absolute Percentage Error, DA: Directional Accuracy. Bold figures for best result. All models were trained for 100 epochs with early stopping using ADAM as the optimizer and a learning rate of 0.01 with a batch size of 128. In all cases, a one-day prediction window was used, considering the previous 10 days. A 20% of the data was reserved for validation.

	Training		**Validation**	
Model	**MAPE(%)**	**DA(%)**	**MAPE(%)**	**DA(%)**
GRU 8-1	1.46	62.39	13.30	62.09
GRU 8-2	**0.88**	63.49	12.45	54.85
GRU 8-3	1.91	45.32	20.92	46.59
GRU 16-1	2.02	69.28	**6.18**	61.78
GRU 16-2	0.92	73.11	10.29	65.16
GRU 16-3	2.66	71.58	17.00	66.52
GRU 32-1	0.90	71.55	6.41	63.53
GRU 32-2	3.32	71.66	10.78	**70.40**
GRU 32-3	1.38	**75.06**	15.20	66.00

(Experiment 1), which is applied directly to the original closing price series, exhibiting noise and volatility.

By subtracting from the original closing price, the IMFs were obtained, and as a result, the summation of all IMFs becomes exactly the same as the original closing price. Hence, it can be inferred that the addition of the prediction outcomes of all IMFs can be regarded as the prediction outcome for the original closing price.

Once we have decomposed the initial non-stationary series in a sum of IMFs, we can apply the GRU algorithm to the separated IMFs. In this Experiment 2, we used a general GRU model to predict all the IMFs that are combined for the final price prediction. As the main problems with the non-stationarity of the series were shown in the S&P 500, we have focused our analysis from this point on the American index. Table 4.6 and Figure 4.7 exhibit the performance metrics and the predictive fitting of this model for American S&P 500.

As it can be observed, the performance metrics improve significantly after the application of the EMD transformation. The DA figures, just around 50% in the previous experiment, now reach remarkable levels around 70%. MAPE figures keep similar levels after EMD decomposition. Therefore, it seems that treating the non-stationarity of the series with EMD has greatly improved the model's ability to capture the price behavior and eliminated the bias in the prediction that the previous model without EMD showed. The fact that the MAPE is similar in both experiments means that, on average, the model stays close to the series being predicted. However, the significantly better DA indicates that the ML algorithm has been able to understand the series much better, being much more accurate in anticipating changes in price direction.

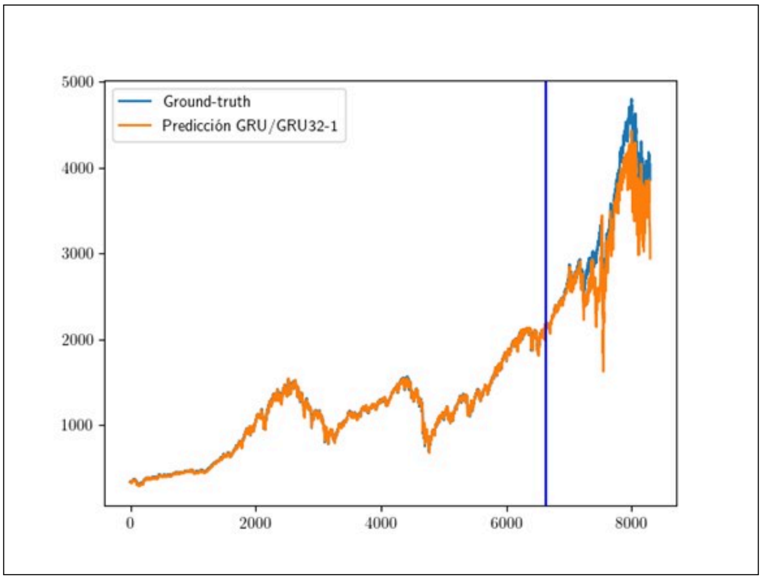

Figure 4.7: Results in S&P 500 using the selected best model GRU 32-1. Experiment 2. The blue vertical line separates the data into training (left) and validation (right) sets. Ground truth: actual data.

Experiment 3: EMD model + Individual GRU This time we use a different GRU model to predict each IMF. Taking into account that the level of complexity, noise and stationarity is clearly different between IMFs, it might have sense to apply different GRU algorithms to each one of the components. Therefore, we adapt the complexity of the GRU algorithms to that of the IMFs, bearing in mind that the first group of IMFs have high frequency and complexity, unlike the last group of IMFs, that shows much easier profiles. This fact can be seen reflected in Figure 4.8, where the selected GRU 8-1 model has been applied to every single IMF. The errors associated to the first group of IMFs is clearly higher to those of latest IMFs.

It can be seen in Table 4.7 that this model shows the best results until now. The DA figures are clearly above 75% both in training and validation sets. The prediction errors are remarkably low, between 0.5% and 1.0% for both datasets. Therefore, the improvement showed after EMD application has carried on with the use of individual GRU algorithms to each of its components.

Figure 4.9 exhibit the predictive fitting of this model for S&P 500 (bottom half) versus the model used in Experiment 2 (top half), focusing the analysis in the validation sets. It is more than evident, comparing both graphs, the improvement in prediction capacity introduced by using an individual GRU for each IMF.

Table 4.7: Performance metrics for S&P 500. Experiment 3. Best model figures. MAPE: Mean Absolute Percentage Error, DA: Directional Accuracy.

Training		**Validation**	
MAPE(%)	**DA(%)**	**MAPE(%)**	**DA(%)**
1.07	78.88	0.56	75.15

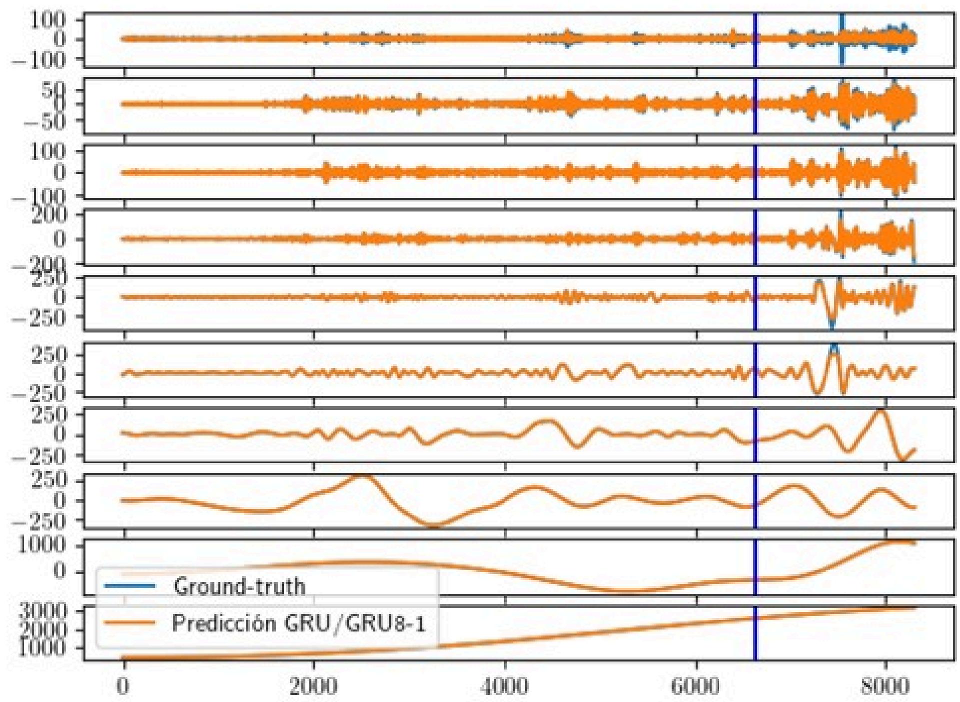

Figure 4.8: Results in S&P 500 using model GRU 8-1. Experiment 3. Prediction errors for individual IMFs. Ground truth: actual data.

Table 4.8: Performance metrics for S&P 500. Experiment 4. Best model figures. MAPE: Mean Absolute Percentage Error, DA: Directional Accuracy.

Training		**Validation**	
MAPE(%)	**DA(%)**	**MAPE(%)**	**DA(%)**
2.00	74.4688	0.82	73.84

Experiment 4: EMD model + Individual GRU + Technical Indicators This experiment has a similar structure to the previous one. Nevertheless, in addition to using each IMF as input for each model, we also use the 10 technical indicators of our dataset (see Table 4.2) as additional inputs for each one.

It can be seen in Table 4.8 that the addition of this new features has not improved the results of previous experiments, using only closing prices. It is not a bad result, but MAPE is slightly higher and DA slightly poorer. This result partially contradicts the findings obtained in other empirical studies.

Experiment 5: Time window optimization of EMD model + Individual GRU Once that different combinations of EMD and GRU have been tested, the only remaining step

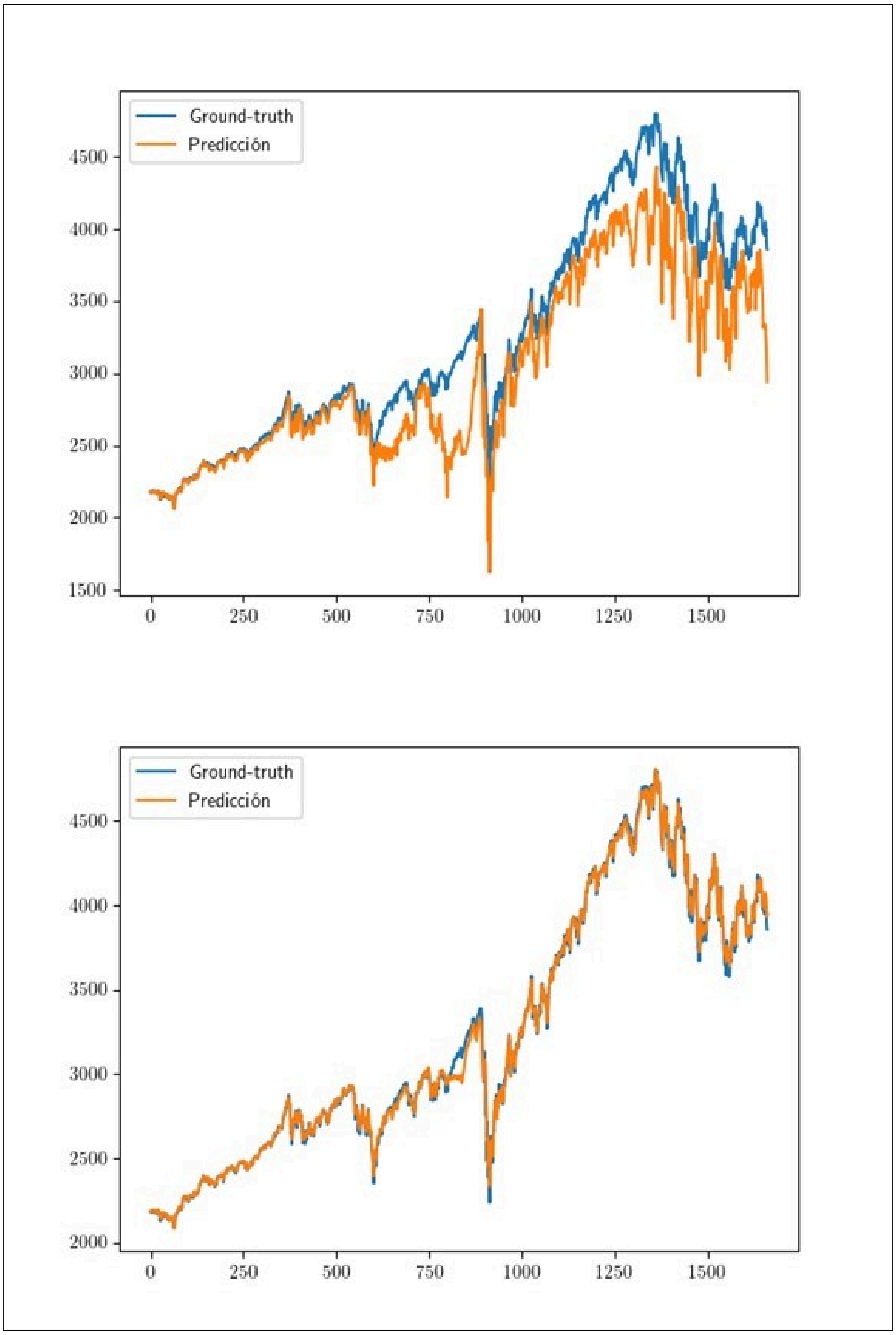

Figure 4.9: Results in S&P 500 using best model. Experiment 2 versus Experiment 3. Validation set prediction errors. Ground truth: actual data.

Table 4.9: Impact in Closeness and Directional Accuracy for S&P 500. Best model figures. Experiment 5. DA: Directional Accuracy. Look-back: Number of days taken for modeling. Look-ahead: Number of days taken for forecast.

	Closeness				**Directional Accuracy**			
Lookback/Lookahead	**1**	**5**	**10**	**20**	**1**	**5**	**10**	**20**
1	3.15	3.62	3.84	4.36	68.92	57.89	56.11	53.56
5	1.16	1.15	1.77	2.74	76.27	61.78	57.54	56.76
10	**0.62**	1.14	1.71	2.58	74.74	61.01	56.91	54.80
20	0.68	1.12	1.42	2.25	**76.40**	62.14	56.31	54.93

is to optimize the time windows for modeling. It is worth noting that the results presented so far have used a time window of 10 days backwards and one day forward.

As can be seen in Table 4.9, the impact on Closeness and Accuracy of expanding and reducing the time windows is clear. The optimal closeness is achieved with a time window of 10 days in the past and 1 day in the future. These have been the windows used in previous experiments. The results are fully intuitive, as they worsen as we use fewer days for modeling and move further into the future for prediction. A similar pattern is observed with the Accuracy, where the maximum is achieved with the 20/1 window and the minimum with the 1/20 window. The 5-day look-back also yields very interesting results in terms of Accuracy.

Experiment 6: Trading Simulation of best Model Unlike previous experiments, where the goodness of the model was measured based on the proximity of the estimated prices to the actual prices, in this experiment we want to be much more practical. While an algorithm's directional accuracy can be impressive, it does not guarantee profitability for an investment strategy based on it.

Hence, evaluating predictive models through trading simulations offers a more comprehensive perspective on their performance in real-world scenarios. This last exercise employs the Bollinger Band (BB) trading technique to make decisions on investing in the S&P 500 Index. The BB strategy was created by John Bollinger in the 1980s and involves two boundaries – an upper and lower band – which are placed two standard deviations away from a 20-day simple moving average. The upper band indicates a buy signal and the lower band indicates a sell signal.

The upper and lower bands have served as indicators for when to buy or sell the S&P 500 Index. If the predicted price, using the best model (EMD-IGRU) is higher than both the upper band and the closing price, it has been interpreted as a buying signal, since the predicted price is two standard deviations above the average. On the other hand, if the predicted price is lower than both the lower band and the closing price, it has been considered as a selling signal, since the predicted price is two standard deviations below the average. In theory, if prices are accurately predicted, the trading signal generated from these conditions should never result in a loss. By utilizing them, this study can accurately measure the profit-making capabilities of the predictive models.

The trading simulation works as follows: When the predicted price surpasses both the upper band and the actual closing price, showing that the predicted price is significantly higher (two standard deviations) than the normal price, a long (buy) signal is sent. Conversely, it indicates a short (sell) signal when the forecast price is significantly below the average price (by two standard deviations) and drops below both the lower band and the actual closing price. Theoretically, if prices were forecast precisely, trading signals developed based on these criteria would never result in losses. Since we have translated the prediction models into actual trading strategies, we will measure the goodness of prediction of the ML models using the standard statistics of any trading system. Hence, conducting trading simulations offers a more practical approach to assessing the effectiveness of predictive models in real-life scenarios. These statistics primarily focus on checking the size of the obtained returns, their volatility, and the presence or absence of significant draw-downs in the cumulative performance.

Figure 4.10 illustrates the trading signals generated by the BBs using the EMD-IGRU model's 20-day advance predictions of the S&P 500 Index prices. The 20-day closing price forecast is shown by the green line. The blue and red lines, respectively, represent the top and lower bands of the BBs. The 20-day forecast price points that are above the upper band imply long signals, while the 20-day anticipated price points that are below the lower band suggest short signals (shown by dashed vertical red lines). The trading system assumes that long and short positions are closed after a 20-day trading period. This means that if the system buys the index today, it will automatically be sold 20 days later. To avoid repeated signals resulting from minor price fluctuations around the BBs, repeated long and short signals within a 20-day time-frame were disregarded.

Figure 4.11 shows the trading signals using the best model, and the trading simulation using the BBs, for the window of 10-day ahead. As it can be seen, there are 25 trading signals, of which 13 are buy (long) signals and 12 are sell (short) signals.

In Table 4.10 there have been presented the main statistics of the trading simulation experiments, using the best prediction model, for the four different temporal windows considered, and taking into account transactions costs. Specifically, it has been assumed a negative return of 0.05% for every single situation in which there is a position change. This cost includes both brokerage fees and the bid-ask spread.

It is quite remarkable that, taking into account the DA as unique prediction accuracy measure, the best ML model works well for the four temporal windows considered. Every single figure is clearly above desirable 50%. Moreover, 10-day and 20-day ahead simulations seem the best. Nevertheless, in the case of the longest window, the trading simulation results, in terms of accumulated return, Sharpe ratio or MDD are quite disappointing. Hence, it becomes evident that, although an algorithm may be highly accurate in its predictions, using traditional accuracy measures, this does not mean that it will be

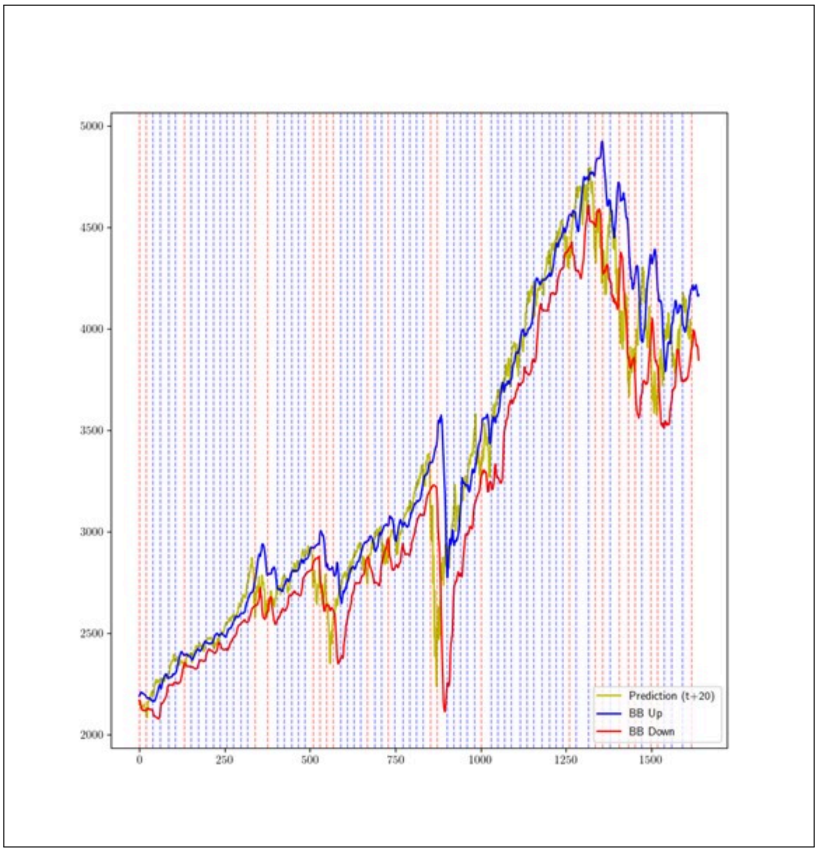

Figure 4.10: Trading simulation EMD-IGRU Model 20-day ahead predictions.

day	bb_up	bb_down	price_today	price_t10_predicted	price_t10	type
36	2.166,32	2.121,41	2.126,41	2.118,37	2.164,45	short
87	2.281,86	2.244,30	2.270,44	2.282,89	2.294,69	long
105	2.304,02	2.258,79	2.294,67	2.307,08	2.363,81	long
155	2.370,76	2.333,21	2.348,69	2.373,17	2.399,29	long
212	2.452,58	2.412,14	2.447,83	2.454,70	2.475,42	long
329	2.703,29	2.622,79	2.687,54	2.703,98	2.786,24	long
339	2.780,30	2.634,49	2.786,24	2.780,94	2.853,53	long
491	2.871,66	2.803,01	2.862,96	2.874,17	2.888,60	long
517	2.932,89	2.868,83	2.914,00	2.868,43	2.728,37	short
661	2.946,28	2.846,26	2.939,88	2.845,73	2.881,40	short
671	2.959,33	2.869,10	2.881,40	2.866,91	2.826,06	short
747	2.961,30	2.828,10	2.887,94	2.973,92	3.009,57	long
757	3.020,69	2.830,16	3.009,57	3.030,39	2.977,62	long
805	3.140,19	3.000,31	3.120,18	2.994,99	3.112,76	short
815	3.150,79	3.069,02	3.112,76	3.014,78	3.191,14	short
829	3.238,67	3.082,82	3.223,38	3.246,95	3.274,70	long
839	3.283,97	3.152,43	3.274,70	3.381,66	3.295,47	long
849	3.343,48	3.203,70	3.295,47	3.460,13	3.327,71	long
859	3.359,23	3.230,07	3.327,71	3.386,37	3.225,89	long
869	3.427,48	3.211,91	3.225,89	3.128,97	2.746,56	short
925	2.954,79	2.753,54	2.820,00	2.749,45	3.029,73	short
1016	3.572,98	3.252,92	3.315,57	3.238,06	3.360,95	short
1264	4.558,51	4.420,39	4.473,75	4.405,93	4.357,04	short
1514	4.265,55	3.823,60	3.901,35	3.815,46	3.640,47	short
1585	4.109,07	3.763,50	3.829,25	3.759,65	3.969,61	short

Figure 4.11: Trading signals 10 day-ahead using EMD-IGRU best model.

Table 4.10: Descriptive statistics of the trading simulation using best model (EMD-IGRU). 1, 5, 10 and 20-day ahead predictions. Return calculated using interest capitalization and transaction costs. TP: True Positive. TN: True Negative. FP: False Positive. FN: False Negative. DA: Directional Accuracy. MDD: Maximum Draw-Down.

Statistic Indicators	**1-day**	**5-day**	**10-day**	**20-day**
Total Net Return (%)	40.24	36.50	39.04	-0.13
Average Return (%)	0.22	0.56	1.42	0.18
StDev Return (%)	1.63	2.23	4.18	5.18
# Trades	173	61	25	24
TP	33	21	11	8
TN	59	15	7	8
FP	27	7	2	1
FN	54	18	5	7
DA (%)	53.18	59.02	72.00	66.67
Sharpe Ratio	1.76	1.94	1.70	0.15
MDD (%)	11.94	13.65	12.57	28.48

successful in generating profits.

Except in this case, the trading results can be considered as truly interesting, according to trading standards, as can be remarked taking a look to the accumulated return performance charts in Figure 4.12. The temporal windows of 5- and 10-day show a more than remarkable results in terms of return, volatility, accuracy and stability. Specifically, the 10-day model offers a total return of 39.04% in only 25 trades, a DA of 72.00%, a SR of 1.70 and a MDD of 12.57%.

Therefore, it can be concluded that our best ML model, the EMD-IGRU, passes such a demanding test as exhibiting good performance in a simulated trading test, even after considering transaction costs. This is a crucial contribution comparing with the results showed by similar papers in the literature.

Discussion

Table 4.11 summarizes the results of the different models carried out, in terms of prediction accuracy. In terms of MAPE, only the model with individual GRUs for each one of the IMFs is able to beat the figures of a plain vanilla Box-Jenkins model (ARIMA(1,1,1)). The results are very similar for the model with technical indicators as additional features.

In terms of DA the results are quite different. Except in the case of the initial GRU model, without the use of EMD, the remaining models predict better the S&P 500 closing prices than ARIMA best model. Thus, the combination of EMD pre-processing technique and GRU enhances substantially the results from traditional univariate models, usually considered as the better approach to financial series as S&P 500 stock prices. The DA of

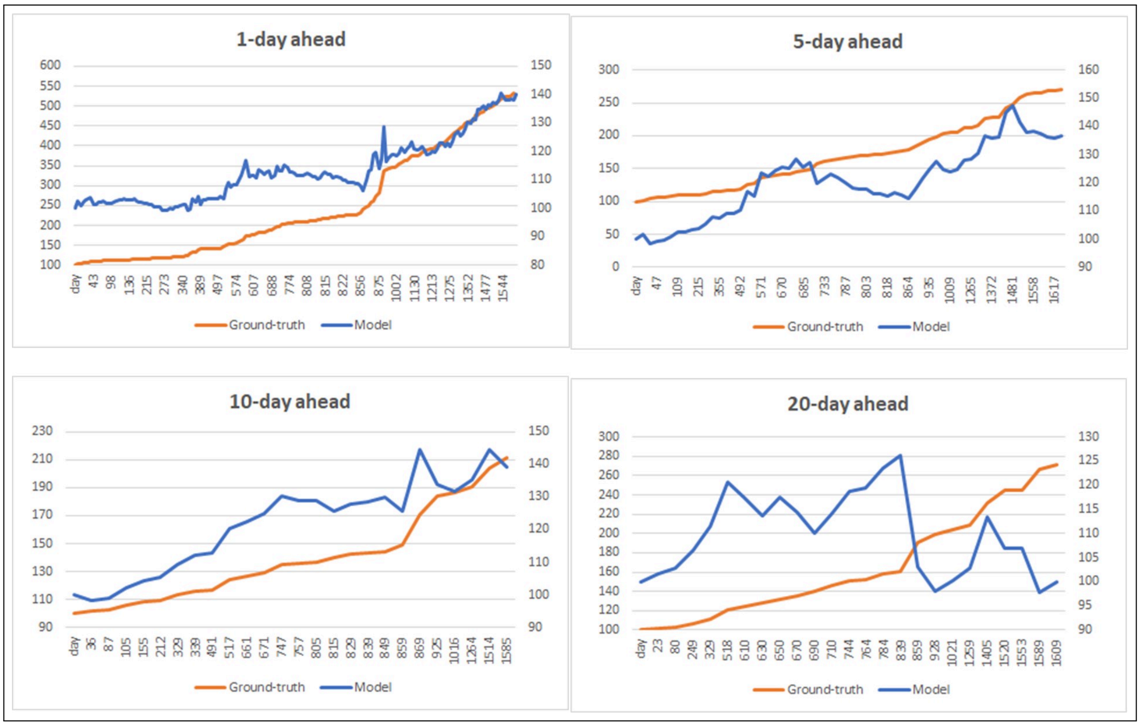

Figure 4.12: Trading simulation performance. Cumulative return. Different windows. Best prediction model. Ground truth: actual data (left axis).

ML approaches oscillates between 63.53% in the worst case and 75.15% in the case of the best model (Experiment 3). They are also above recent results in the literature, which use some other approaches of ML or DL to model and predict S&P 500 stock prices. In most of cases, the DA figures do not exceed 70%. Good examples of that can be found in [Akiyoshi, 2020], [Nava et al., 2018], [Shi and Zhuang, 2019] or [Srijiranon et al., 2022].

Finally, as we have seen in the previous section with the Experiment 6, the best model of all, the EMD-IGRU, has been exposed to a real-life stress test, a trading simulation. The model's response has been more than satisfactory, and according to the statistical standards employed in market trading systems, the model delivers commendable results in

Table 4.11: Summary of prediction accuracy figures for best model in each experiment (validation set). S&P 500. MAPE: Mean Absolute Percentage Error, DA: Directional Accuracy. ARIMA: ARIMA model (1,1,1). GRU: GRU model without EMD. GGRU: General GRU model for every IMF. IGRU: Individual GRU model for each IMF. TI: Technical Indicators.

Experiment	Model Name	MAPE (%)	DA (%)
0	ARIMA	0.76	51.41
1	GRU	2.78	48.55
2	EMD-UGRU	6.41	63.53
3	EMD-IGRU	**0.56**	**75.15**
4	EMD-TIs-IGRU	0.82	73.84

practically all prediction windows. This result suggests that the GRU predictive models, enhanced with the previous application of the EMD transformation for non-stationary series, exhibit a reasonable level of accuracy in predicting stock prices and directional movements. Moreover, it has also demonstrated to be effective in predicting actual profits in real-world scenarios.

4.2.2. Pairs Trading

Introduction

The goal of investing is to minimize risk while maximizing returns. To this end, the 'pairs-trading' strategy, sometimes called statistical-arbitrage, has been used increasingly in modern hedge funds due to its simplicity and market-neutrality. This technique entails monitoring the correlation between two stocks that are known to be connected, and opening a long position on the one that rises and a short position on the one that falls. The underlying assumption is that, despite short-term divergence, the two stocks will converge in the long-run, thus allowing the trader to benefit from the pair regardless of the market situation.

Trading techniques for pairs often use cointegration or some other time series-related statistic to find pairs ([Gatev et al., 2006], [Vidyamurthy, 2004]). For other authors ([Han et al., 2021]), using solely historical price data, however, might lead to the misleading identification of spuriously linked pairs that might not move jointly in the future. It might seem that it is possible to find more successfully comparable equities using some kind type of techniques which take into account both previous price movements and additional firm characteristics associated to the idiosyncratic risk of both stocks. Stocks with similar firm characteristics would be seen to move together more likely, and this would be a good way to find pairs that can work successfully in a pairs trading strategy.

This research looks at the viability of using unsupervised learning ML techniques to find possible stock pairs for a long-short portfolio. Nevertheless, considering that we are working with time-series data, each data point is an ordered sequence, and because of that, clustering various time series into comparable groups can be considered a difficult operation. By flattening the time series into a table with a column for each time index (or aggregate of the series), the most popular way for clustering time series is to directly employ basic clustering algorithms like k-means. In other words, the most recurrent method of applying UL techniques of classification for time series is to eliminate the time dependence of the data or, what is the same, to avoid the temporal dimension of the data.

We propose a framework for classifying high-dimensional financial data into clusters and, for this aim, we propose a triple-way contribution. First, we suggest to apply two different clustering algorithms to build pair portfolios for statistical arbitrage. Specifically, we pick k-means clustering and Density-based Spatial Clustering of Applications with Noise (DBSCAN). Second, we propose to apply a very modern clustering approach to time series data, without losing the temporal dimension, using the k-means clustering with Dynamic Time Warping (DTW). Third, we propose to use as additional features in the classification problem the firm characteristics associated to the idiosyncratic risk of stocks.

After clustering, we use the previously chosen pairs to create an equally-weighted long-short portfolio. Two benchmarks —the S&P 500 index and the short-term reversal portfolio— will be used to compare these clustering-based long-short portfolios.

Main contributions

- Hybrid framework. A hybrid framework that combines dimensionality reduction through auto-encoders and UL techniques of clustering as k-means and DBSCAN.

- Time series approach. We propose to apply a very recent clustering approach to time series data, without losing the temporal dimension, using the k-means clustering with DTW.

- Firm characteristics. We propose to use as additional features in the classification problem the firm characteristics associated to the idiosyncratic risk of the stocks, trying to add financial valuable information to find pairs of values which can perform together in the long term.

- Multiple accuracy measures. The prediction model is evaluated using traditional metrics and it will be finally tested using a trading simulation model compared to several benchmarks as S&500 and short-term reversal portfolio.

Research Data and Feature Generation

The dataset includes monthly stock price data from S&P 500 Index constituents and quarterly fundamentals, which have been provided by Factset. Those stocks with a remarkable lack of information about quarterly fundamentals (above 50% of missing values) have been eliminated from the dataset. Therefore, the feature set finally consists of stock prices about 392 firms and 15 firm characteristics generated every month for the sample period from January 2010 to January 2023. Our sample period for training strategies was set from 2010-01-01 to 2014-12-31 (5 years), and we used the sample period of 2015-01-01 to 2023-01-01 for validation.

The feature set is split into two parts. The first set is composed of 36 price momentum values, ranging from 1 to 36 months. The second set of features includes 15 firm characteristics, with monthly frequency, selected using our own criteria. Using the features available at the end of month $t-1$, stocks are grouped together, and the prior one-month return is used to pinpoint undervalued and overvalued stocks.

The momentum of stock prices can be used to predict future stock movements. However, differences in firm characteristics may cause two stocks that had similar movements in the past to vary in the future. To take this into account, we introduce a clustering method that uses both momentum and firm characteristics. This is relatively unusual in the literature about pairs trading, which usually rely solely on price or return information.

Momentum features The price momentum at the end of month $t-1$ is defined as the cumulative return from month $t-i$ to $t-2$ for $i > 1$, and as the prior one-month return for $i = 1$:

$$
\begin{aligned}
mom_i &= r_{t-1}, \quad i = 1, \\
mom_i &= \prod_{j=t-i}^{t-2} (r_j + 1) - 1, \quad i \in 2, ..., 36
\end{aligned}
\tag{4.8}
$$

where r_j denotes the return in month j.

Firm characteristics The 15 firm characteristics have been chosen following our own criteria, trying to be as general as possible to define the financial and economic reality of the firms. There are characteristics associated to price, profitability, solvency, liquidity, among others, as can be seen in Table 4.12.

System Architecture

In this section, we will be taking a deep dive into the two types of algorithms we will use in this experiment. In the one hand, a dimensionality reduction algorithm brought from the NNs field, the auto-encoders. In the other hand, the two clustering algorithms to be used: k-means clustering and DBSCAN. These two algorithms represent partition-based and density-based clustering, respectively. As opposed to SL methods, clustering algorithms are unsupervised, meaning they do not require a set of labels to target. All they need is the input data, and they can find patterns that may not be easily visible to the human eye. We will pay special attention to time series k-means clustering with DTW, which is the UL classification algorithm developed to be applied specifically to time series data as ours.

In contrast to SL methods, such as DL or gradient boosting, which generally require a great number of hyperparameters, clustering algorithms require only a limited number of them and leave little opportunity for data snooping. We assess the effects of various hyperparameter values on the operation of the pairs trading strategy by experimenting with them.

Table 4.12: Firm characteristics used in UL clustering methods.

Characteristics	**Formula**
Price to Book ratio	$\frac{\text{Price per share}}{\text{Book value per share}}$
Return on Equity	$\frac{\text{Net Income}}{\text{Total Equity}}$
Asset Turnover	$\frac{\text{Total Sales}}{\text{Total Assets}}$
Total Debt % Equity	$\frac{\text{Total Debt}}{\text{Total Equity}} \times 100$
Net Cash Flow % Total Debt	$\frac{\text{Net Operating Cash Flow}}{\text{Total Debt}} \times 100$
Free Cash Flow Margin	$\frac{\text{Free Cash Flow to Equity}}{\text{Total Sales}}$
Free Cash Flow to Equity	$\frac{\text{Free Cash Flow}}{\text{Total Equity}}$
Long Term Debt % Total Capital	$\frac{\text{Long Term Debt}}{\text{Total Capital incl.ST Debt}}$
Net Income Margin	$\frac{\text{Net Income}}{\text{Net Sales}} \times 100$
Price Earnings ratio	$\frac{\text{Price per share}}{\text{Earnings per share}}$
Return on Average Assets	$\frac{\text{Net Income}}{\text{Average Total Assets}} \times 100$
Return on Invested Capital	$\frac{\text{Net Income}}{\text{Invested Capital}} \times 100$
Earning Assets to Total Available Funds	$\frac{\text{Earning Assets}}{\text{Total Capital incl.ST Debt}} \times 100$
Cash Flow Return on Invested Capital	$\frac{\text{Net Cash Flow}}{\text{Invested Capital}} \times 100$
Price to Free Cash Flow	$\frac{\text{Price per share}}{\text{Free Cash Flow}}$

Dimensionality reduction: auto-encoders It is a typical practice to preprocess data by scaling it and using PCA before carrying out an unsupervised clustering process (for instance, [Sarmento and Horta, 2020]. To avoid skewed clustering, it is very common to apply normalization to features so that all of them are given equal weighting in distance calculation. This normalization is done by computing the cross-sectional mean and standard deviation of the features, and then subtracting the mean from each feature and dividing it by the standard deviation. This ensures that all features have the same weighting in clustering, and that the resulting clusters are not skewed.

Nevertheless, there are some techniques, coming from the SL field of ML algorithms, which are more effective when there are some doubts about the linearity of the data structure. This is the case of the auto-encoders.

By stacking several non-linear transformations (layers), auto-encoders are NNs that may be used to reduce the data into a low dimensional latent space. Their architecture is encoder-decoder. The input is translated into latent space by the encoder, and it is reconstructed by the decoder. For appropriate input reconstruction, they get back propagation training. Auto-encoders can be used for dimensionality reduction when the latent space has fewer dimensions than the input. These low-dimensional latent variables can reconstruct the input, therefore it seems make sense that they would encode the majority of its key properties.

An auto-encoder is a unique and challenging three-layered NN in which we set the output $h_{W,b}(x) = (\tilde{x_1}; \tilde{x_2}; ...; \tilde{x_n})^T$ equal to the input $x = (x_1; x_2; ...; x_n)^T$. Suppose the original input x belongs to n-dimensional space and the new representation y belongs to m-dimensional space. The reconstruction error is J. It employs the back propagation technique for training and is an UL algorithm.

$$h_{W,b}(x) = g(f(x) \approx x \tag{4.9}$$
$$J(W, b; x, y) = \frac{1}{2}||h_{W,b}(x) - y||^2$$

The structural description of an auto-encoder is shown in Figure 4.13. An encoder f goes from the first to the second layer, while a decoder g goes from the second to the third layer. By modifying the encoder and decoder's settings, we may then obtain the code while minimizing the reconstruction error J.

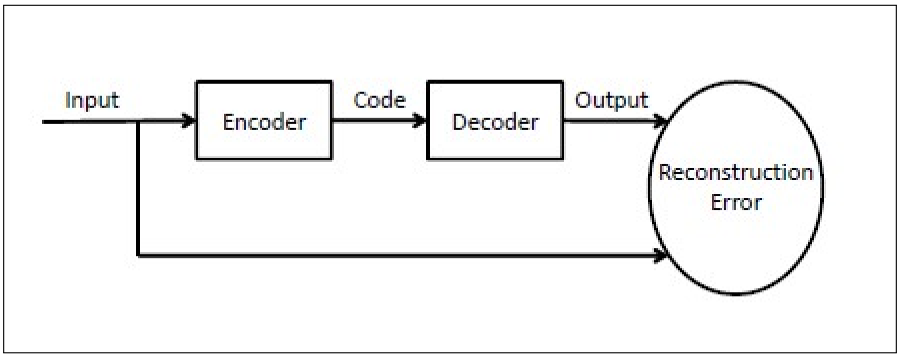

Figure 4.13: The visualization description of auto-encoder.

K-means clustering The most used clustering technique is probably k-means clustering ([MacQueen, 1967]). It initially requires the specification of K, the number of clusters. By using the Euclidean distance, the within-cluster sum of squares (WCSS) between the data points and their individual centroids is minimized, K centroids are then identified, and all of the data points are then grouped into one of these clusters. The following provides the objective function:

$$W = WCSS = \sum_{i=1}^{N}\sum_{k=1}^{K} w_{ik}||x^i - \mu_k||^2 \tag{4.10}$$

where x^i refers to i-th data point, μ_k the centroid of cluster k, w_{ik}=1 if x^i belongs to cluster k, and N is the total number of data points. $||.||$ denotes l_2 norm.

The minimization issue for the k-means clustering involves two steps. In order to update the assignment of data points to clusters, W is first reduced with regard to w_{ik}

while keeping μ_k fixed (Equation 4.11). Then, while retaining w_{ik} fixed to recompute the centroids, W is minimized with respect to μ_k (Equation 4.12). Until W is minimized, the aforementioned actions are repeated.

$$w_{ik} = \begin{cases} 1, & \text{if } k = \text{argmin}_j \|x^i \mu_j\|^2. \\ 0, & \text{otherwise.} \end{cases} \tag{4.11}$$

$$\frac{\partial W}{\partial \mu_k} = 2 \sum_{i=1}^{N} w_{ik}(x^i - \mu_k) = 0 \tag{4.12}$$

K-means clustering integrated with Dynamic Time Warping (DTW) [Niennattrakul and Ratanamahatana, 2007] integrated DTW in their work, replacing the Euclidean distance with a K-medoids clustering approach. Due to its greater sequence-alignment flexibility, this relatively new distance measure has replaced the more widely used Euclidean distance as a similarity measurement for different data mining tasks. In other words, Euclidean distance is less unsuited for time series since it will incorrectly assess the distance between two strongly correlated time series if one is altered by even one time step.

According to [Ratanamahatana and Keogh, 2004], DTW overcomes the restriction of one-to-one alignment of other distance measures and permits non-equal-length time series. Using a distance matrix with each element representing the cumulative distance of the minimum of the three immediate neighbors, it uses dynamic programming to search every feasible path before choosing the one that results in the smallest distance between the two time series. Assume we have two time series, $Q = q_1, q_2, ..., q_i, ..., q_n$ and $C = c_1, c_2, ..., c_j, ..., c_m$. The first step is to generate a nxm matrix, where each (i, j) element represents the sum of the distances at (i, j) and the minimum of the three elements that are immediately next to the (i, j) element, where $0 \leq i \leq n$ and $0 \leq j \leq m$.

The (i, j) element can be described as:

$$e_{ij} = d_{ij} + \min\{e_{(i-1)(j-1)}, e_{(i-1)j}, e_{i(j-1)}\} \tag{4.13}$$

where e_{ij} is the (i, j) matrix element and d_{ij} is equal to $(c_i + q_i)^2$, the total of the squared distances between (i, j), as well as the minimum cumulative distance between the three elements surrounding the (i, j) element. The path that delivers the smallest cumulative distance at (n, m) is the one to pick next in order to find the best course of action. The measurement of distance is:

$$D_{DTW}(Q, C) = \min_{\forall w \in P} \left\{ \sqrt{\sum_{k=1}^{K} d_{w_k}} \right\} \tag{4.14}$$

where P is a set of all possible warping paths, and w_k is (i, j) at k-th element of a warping path and K is the length of the warping path.

DBSCAN In a high dimensional data space, the DBSCAN identifies areas of high density divided by regions of low density ([Ester et al., 1996]). The minimal number of data points per cluster, $MinPts$, and the maximum distance between data points that must be deemed to be in the same cluster, ϵ, are its two parameters.

DBSCAN first chooses a random data point to serve as a core point. For the core point, only data points with at least $MinPts$ neighbors within ϵ are taken into consideration. In a technique known as direct density accessible, every neighboring point inside ϵ of a core point is grouped with the core point. These neighboring points' neighbors are likewise included in the same cluster. Density reachable is the term for this procedure. Border points are non-core points in a cluster, while points inside the same cluster are referred to as being densely linked. According to [Schubert et al., 2017], outliers or noise are places that cannot be densely reached from any of the core points and are not a part of any cluster. This process is repeated until all data points are visited.

Portfolio Formation and Trading Strategy

At the conclusion of each month throughout the research period, we perform the dimensionality reduction using auto-encoders, and then we group stocks together using one of the already described clustering algorithms. Clustering enables us to assign stocks to a group or recognize them as outliers. The clustering algorithm has been applied to identical time windows of five years (sixty observations), and the portfolio formation has been started in February 2015. From this moment on, the clustering optimization has been carried out repeatedly every month.

According to price trends and fim characteristics up to month $t - 1$, we will assume in the prediction that stocks that were grouped together in the same cluster will move similarly in month t. To detect pairs of stocks, we analyze their one-month returns (mom_1) prior to month t. Stocks with a lower mom_1 value are likely to experience an upswing in the following month -or longer-, while stocks with a higher mom_1 are likely to experience a decrease in price.

We arrange the stocks in each cluster according to their mom_1 values, forming pairs of stocks with the highest and lowest mom_1 values, the second-highest and second-lowest mom_1 values, and so on. We then assemble an equally weighted long-short portfolio that is composed of pairs whose mom_1 difference is greater than the average variance of the mom_1 differences of all the pairs. If the mom_1 differences of the pairs in a cluster are not wide enough, none of them will be eligible for trading. The positions are held for a month, and at the conclusion of each month, the portfolio is rebalanced. Up to the conclusion of the sample period, this cycle is repeated.

In order to check the robustness of the model, we have tested the trading system using the alternative periods of three, six and twelve months, and rebalanced the portfolio at the end of each month until the end of the sample period. Obviously, in the experiments

Table 4.13: Clustering statistics summary. Features used in the different clustering algorithms. Price: Prices and momentums. Price+FI: Prices, momentums and financial indicators. Price+DTW: Prices as only features, integrated with DTW.

Statistic	k-means			DBSCAN	
	price	price+FI	price+DTW	price	price+FI
# clusters	20	20	20	20	20
# stocks in total	391	391	391	391	391
# average stocks in the biggest cluster	38	34	63	42	37
# average stocks in the 2nd biggest cluster	32	30	51	35	31
# median stocks in all clusters	19	20	14	18	19

longer than one month, several portfolios will be held at the same time, although only one portfolio will be formed each month. In this way, we will be able to test the average period in which the temporary differences in behavior that may be revealed between two stocks that, due to their characteristics, should perform similarly, tend to go away. In other words, what is the average period to expect to "return to normal."

The performance of the clustering-based long-short portfolios is compared with two benchmarks; the S&P 500 index (including dividends available from Factset) and the short-term reversal portfolio (which involves sorting all stocks based on their mom_1, and initiating long positions on the stocks in the first decile and short positions on the stocks in the bottom decile).

Results

This section provides a summary of each clustering method's grouping characteristics, as well as an evaluation of the financial outcomes of pairs trading approaches from various perspectives. Several robustness tests have been completed to make sure that the results are not caused by data mining.

Clustering Characteristics We provide an overview of the clustering statistics of the two clustering methods, with their subsequent sub-methods, that is, with the use or not of the financial characteristics as additional features, and in the case of k-means clustering, considering the use of DTW. The results of this analysis are outlined in Table 4.13.

It can be seen that the number of clusters and their characteristics are very homogeneous throughout methods. The number of clusters has been optimized at 20, being the median number of stocks approximately 15-20 stocks, and the two biggest clusters conformed by between 40 and 60 stocks.

Strategy Performance Table 4.14 reports the performances of the long-short portfolios constructed via the two clustering methods used in this study. In the same table we have represented the same figures for the two models used as benchmarck: the S&P 500 index and the short-term reversal portfolio calculated over the same index.

As it can be observed, five models have been revised. In the case of k-means clustering, we have estimated three different algorithms, depending on the features considered and the use or not of the DTW: 1) only stock prices and their momentums without DTW, 2) stock prices, momentums and financial indicators without DTW, and 3) only stock prices but using DTW. In the case of DBSCAN, two algorithms have been used: 1) only stock prices and their momentums, and 2) stock prices, momentums and financial indicators.

The model using the k-means algorithm, but in its version adapted to time series, i.e., the k-means clustering integrated with DTW, shows the best results. Without capitalization of the monthly returns, it achieves a total return of 89% in 8 years, with a DA of 63%, a SR of 2.658, and a CR of 0.678. The second best model is obtained using the second algorithm, DBSCAN, without any additional features, only stock prices. It achieves a total return of 67% in 8 years, with a DA of 60%, a SR of 2.148, and a CR of 0.638.

In Table 4.15 we have represented a summary of these two best models, in comparison with the two benchmarks previously introduced. As it can be seen, the k-means clustering integrated with DTW keeps its privilege place in terms of total return, but also in terms of the different trading performance measures, mainly SR figures. The results form the DBSCAN model are not so brilliant in terms of total return, as they keep below benchmark figures, but it is able to stay ahead in terms of SR and CR.

In Figure 4.14, this outperformance of both clustering algorithms over the two benchmarks can be visually confirmed. Both benchmarks go through situations of higher stress, with more pronounced ups and downs, while the two clustering algorithms evolve in a smoother manner, without major disturbances.

Table 4.14: Performance of the equally-weighted pairs trading portfolios constructed via k-means and DBSCAN (for all versions). The monthly returns are not capitalized. The validation sample period is 2015.01 - 2023-01. Features used in the different clustering algorithms. Price: Prices and momentums. Price+FI: Prices, momentums and financial indicators. Price+DTW: Prices as only features, integrated with DTW.

Statistic	k-means			DBSCAN	
	price	**price+FI**	**price+DTW**	**price**	**price+FI**
Total return (%)	44.675	46.518	89.067	67.157	35.188
Monthly return (%)	0.470	0.490	0.938	0.707	0.370
Minimum return (%)	-12.857	-15.538	-14.784	-12.315	-16.721
Maximum return (%)	12.165	12.709	13.563	11.629	12.905
Directional accuracy (%)	50.526	57.895	63.158	60.000	53.684
Sharpe ratio	1.547	1.418	2.658	2.148	1.027
Maximum drawdown (%)	-14.204	-16.391	-16.582	-13.303	-19.628
Calmar ratio	0.397	0.358	0.678	0.638	0.231
Skewness	0.045	-0.360	-0.187	-0.090	-0.502
Kourtosis	5.818	6.072	5.570	3.167	6.450

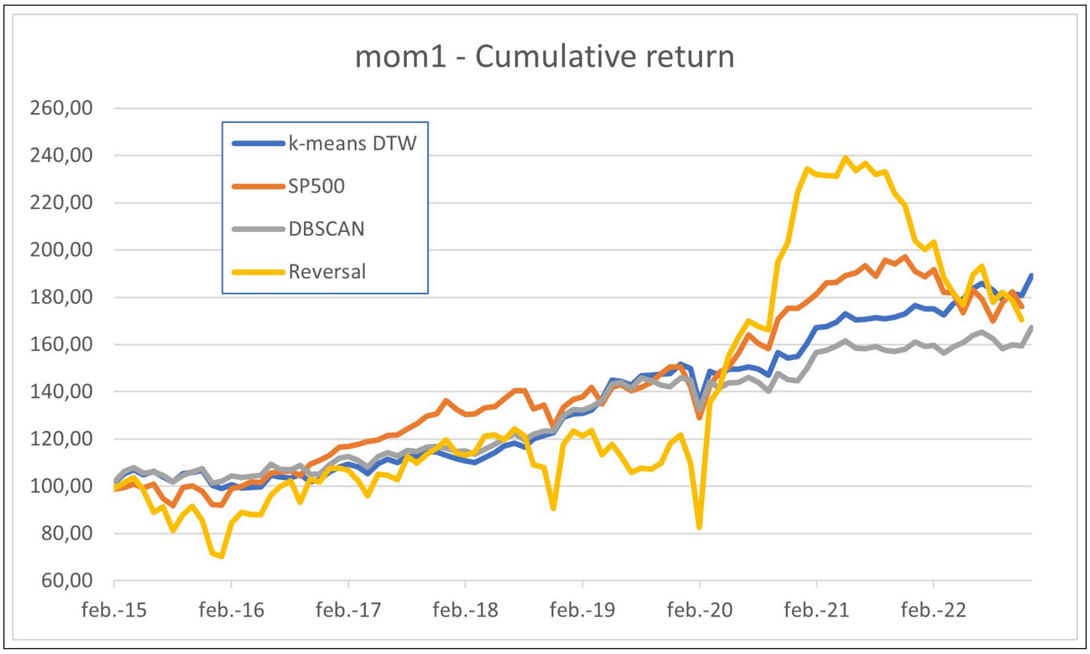

Figure 4.14: Performance of two best clustering methods trading simulations versus benchmarks.

Effects of Firm Characteristics We initially hypothesized that the firm characteristics might be advantageous for clustering stocks and improving pairs trading strategies. As we have explained in previous section, the data does not seem to confirm our hypothesis. The results are not bad, but they do not improve the achievements from the k-means clustering with DTW and DBSCAN, in both cases without financial indicators of firm characteristics as features. It seems that this kind of financial information is not useful for enhancing the classification of stocks in pairs and, thus, for improving the performance of the trading

Table 4.15: Performance of the equally-weighted pairs trading portfolios constructed via k-means and DBSCAN (in their best versions). The monthly returns are not capitalized. The validation sample period is 2015.01 - 2023-01.

Statistic	**k-means**	**DBSCAN**	**S&P 500**	**Reversal**
Total return (%)	89.067	67.157	75.926	70.320
Monthly return (%)	0.938	0.707	0.808	0.748
Minimum return (%)	-14.784	-12.315	-13.375	-27.090
Maximum return (%)	13.563	11.629	13.589	52.850
Directional accuracy (%)	63.158	60.000	64.286	57.317
Sharpe ratio	2.658	2.148	1.642	0.711
Maximum drawdown (%)	-16.582	-13.303	-27.259	-68.670
Calmar ratio	0.678	0.638	0.356	0.131
Skewness	-0.187	-0.090	-0.341	1.518
Kourtosis	5.570	3.167	0.811	7.342

system build with them.

All the clustering techniques produce similar levels of returns, but the returns and other risk-adjusted performance metrics decline when the firm characteristics are considered in the equation.

Factor Regression We use three factor models to investigate if systematic risk factors can account for the returns of long-short portfolios created through the most effective clustering algorithm used in this exercise. We chose k-means clustering integrated with DTW as it showed the most satisfactory results compared to other clustering algorithms.

The three models that we consider are the Fama and French ([Fama and French, 1996]) three-factor model (FF3), the Fama and French ([Fama and French, 2015]) five-factor model, and the [Hou et al., 2020] q^5 factor model. Data for the Fama-French factors are obtained from Kenneth French's website, and the q^5 factors are taken from the global-q website.

As shown in Table 4.16, the long-short portfolio constructed from k-means exhibits an economically and statistically meaningful monthly alpha of 0.56 (t = 1.84) when regressed on FF3. The market beta is also high and positive (0.35), showing that the trading strategy is not risk-neutral but rather dollar-neutral. Positive and weakly significant factor loading on High minus Low (HML) suggests that it favors companies with larger book-to-market ratios. The size factor does not seem to be significant in the returns of our pairs trading strategy. The adjusted R^2 from FF3 is 0.27.

The explanatory power of the FF5 model is comparable to that of the FF3, and the alpha (0.642) is still statistically and economically significant, much as the market beta (0.311). Compared to the FF3 model, the factor loading on HML is once more significant (0.249) and positive, what means that the book-to-market ratio remains as an explanatory factor of our returns. Another factor loading, the investment factor Conservative minus

Aggresive (CMA), also presents statistical significance (-0.284), denoting that our pairs trading model favours firms with an aggressive level of investment. The returns of our strategy do not seem to be explained by the factors of size and profitability.

Three of the factors in q^5 are also unable to explain the returns, suggesting that the significant return of the strategy is not a result of excessive risk taking in this kind of firm characteristics, but the two remaining factors appear to be economically and statistically significant. As in the two previous cases, the alpha (0.846) remains high, and the market beta (0.274) and expected growth factor associated to investment (-0.577) also present significant coefficients. These results seem to demonstrate that our model excess returns might be explained by the selection of firms with lower level of expected growth.

Table 4.16: Factor regression results of the equally-weighted pairs trading portfolio constructed via k-means integrated with DTW. FF3, FF5 and q^5 respectively denote the Fama-French three factors, Fama-French five factors and Hou-Mo-Xue-Zhang q^5 factors. The sample period is from 2015.01 to 2023.01. t-statistic values in parentheses.

Variable	**FF3**	**FF5**	q^5
Intercept	0.564^{*}	0.642^{**}	0.846^{***}
	(1.841)	(2.071)	(2.856)
Market-Rf	0.350^{***}	0.311^{***}	0.274^{***}
	(5.310)	(4.326)	(4.079)
SMB3 - Size factor	0.089		
	(0.760)		
HML3 - Value factor	0.148^{*}		
	(1.895)		
SMB5 - Size factor		0.089	
		(0.665)	
HML5 - Value factor		0.249^{**}	
		(2.208)	
RMW5 - Profitability factor		-0.002	
		(-0.015)	
CMA5 - Investment factor		-0.284^{*}	
		(-1.648)	
R-ME - Size factor			-0.189
			(-1.470)
R-IA- Investment factor			-0.135
			(-1.017)
R-ROE - Profitability factor			0.016
			(0.113)
R-EG - Growth factor			-0.577^{***}
			(-3.207)
R^2	0.294	0.319	0.387
$Adj.R^2$	0.271	0.281	0.352

* p < 0.10, ** p < 0.05, *** p < 0.01

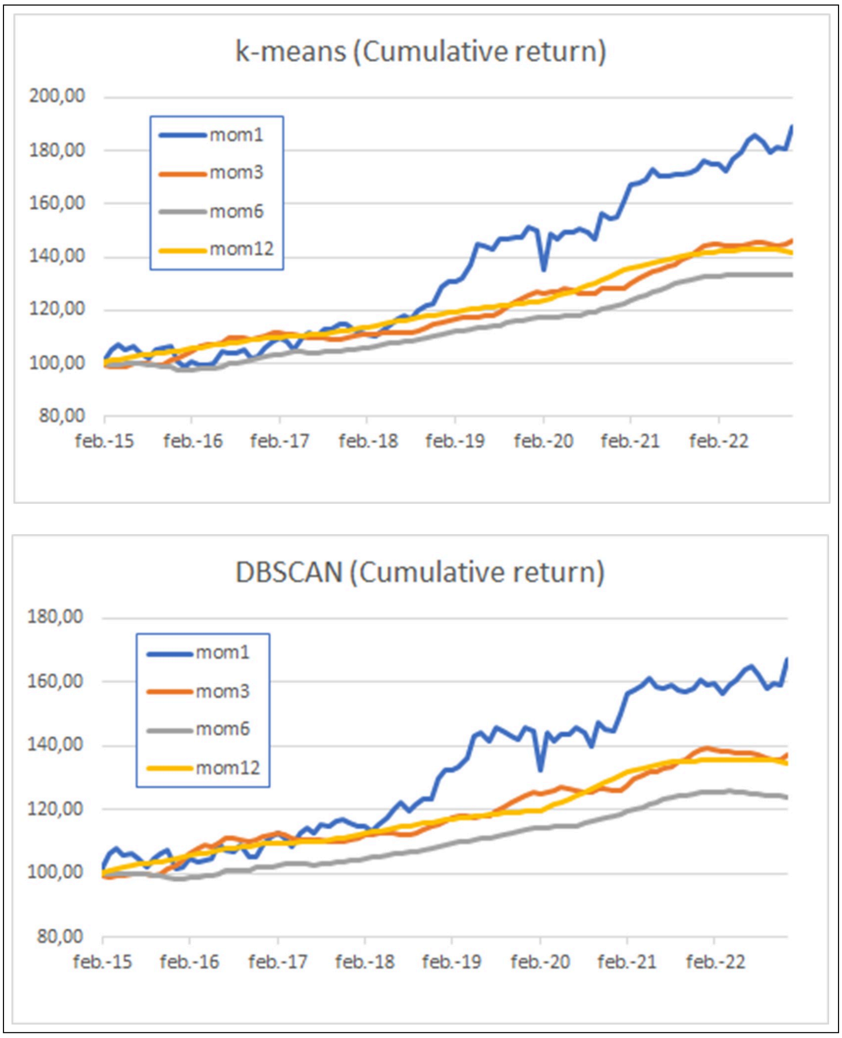

Figure 4.15: Performance two best models. Time windows of 1, 3, 6 and 12 months.

Robustness Check In order to check the robustness of the model, we have tested the trading system using the alternative periods of three, six and twelve months, and rebalanced the portfolio at the end of each month until the end of the sample period. Obviously, in the experiments longer than one month, several portfolios will be held at the same time, although only one portfolio will be formed each month. In this way, we will be able to test the average period in which the temporary differences in behavior that may be revealed between two stocks that, due to their characteristics, should perform similarly, tend to go away. In other words, what is the average period to expect to *return to normal*.

As can be observed in Table 4.17, there is a smoothing of the results. Although the outcomes are less outstanding in terms of total return, the models benefit from greater stability as the poor results of one trade are offset by the good results of other trades that coincide in time. This smoothing effect increases as we extend the trade duration period from 1 to 3, 6, and 12 months, which is evident in significantly higher Sharpe ratios and virtually non-existent draw-downs. These results, also represented in Figure 4.15, confirm the strong robustness of the models used in this exercise.

Table 4.17: Performance of the equally-weighted pairs trading portfolios constructed via k-means and DBSCAN (best versions) for the alternative periods of 3, 6 and 12 months. The monthly returns are not capitalized. The validation sample period is 2015.01 - 2023-01.

Statistic	k-means			DBSCAN		
	3 months	**6 months**	**12 months**	**3 months**	**6 months**	**12 months**
Total return (%)	46.463	33.557	41.367	37.014	24.078	34.435
Monthly return (%)	0.489	0.353	0.435	0.390	0.253	0.362
Minimum return (%)	-0.969	-0.674	-0.838	-0.821	-0.534	-0.826
Maximum return (%)	2.348	1.143	1.187	2.050	0.921	1.18588
Directional accuracy (%)	70.526	82.105	95.789	61.053	78.947	94.737
Maximum Drawdown (%)	2.685	2.478	1.549	3.714	1.856	1.406
Sharpe ratio	5.979	9.056	13.079	5.073	7.529	10.628

Discussion

In this study, we used UL to create pairs trading strategies. We employed, additionally to price data, data from firm characteristics and historical returns, and utilized two clustering techniques, k-means clustering and DBSCAN.

We tested our tactics on the US stock market from January 2010 to January 2023, and we found that both of them significantly beat the market as well as the benchmark for short-term reversals. With an annualized return of 11.3% and an annualised Sharpe ratio of 2.66, the k-means clustering-based approach, integrated with DTW, seemed to be the best performing one. This return was obtained without considering return capitalization or using any level of leverage. It is also true that, on the other hand, transaction costs did not been introduced in this exercise for considering them not so relevant.

During the 2020 market collapse, the trading system exhibited strong performance. Surprisingly, it turns out that firm characteristics are not a significant source of information for recognizing couples. At least, it seems evident, according to the data used, that firm characteristics do not help to find better pairs to trade. The volatility and downside risk of the methods are not significantly reduced, and performance is not enhanced.

We have been interested in finding out whether some kind of risk factors can be behind the extra returns achievement. Therefore, we used three factor models to investigate if systematic risk factors can account for the returns of long-short portfolios created through the most effective clustering algorithm used in this exercise, the k-means with temporal dimension DTW. First of all, the explanatory power of the models is not so brilliant, around 30-40% in the three cases. In terms of statistical significance, factors as size, profitability, or investment factor are not significant in none of the three models, and only market beta factor or growth factor seems to show some level of significance. Therefore, it can be concluded that the extra returns achieved with our ML model and trading system can not be explained by excessive risk taking.

Finally, in order to check the robustness of the model, we have tested the trading system using the alternative periods of three, six and twelve months, and rebalanced the portfolio at the end of each month until the end of the sample period. The models gain from increased stability since the strong results of some transactions that overlap in time outweigh the negative results of other trades, despite the outcomes being less exceptional in terms of overall return. The results achieved confirm the robustness of the models and strategy used in the exercise.

4.2.3. Factor Investing

Introduction

[Fama and French, 1992] included size and book-to-equity as the two new variables to the original CAPM model. As we explain in a detailed way in Chapter 2, these authors discovered that when these two factors were combined with the initial risk factor, they effectively reflected the cross-sectional variations in stock returns. Then, as an extension of their 1992 model, the Fama-French three-factor model ([Fama and French, 1993]) was released. It identifies three stock market variables, including an overall market component and factors related to business size and book-to-market equity. They suggested modifying the baseline CAPM model by adding High Minus Low (HML) and Small Minus Big (SMB), respectively. High Minus Low (HML) is the average return on the two high-value portfolios less the average return on the two growth portfolios. Small Minus Big (SMB) is the average return on the three smallest portfolios reduced by the average return on the three largest portfolios. Their model did better than the original CAPM, which could only account for over 90% of portfolio returns with data from the United States on average.

In spite of this compelling new data, some scholars questioned the ability of the Fama-French three-factor model to forecast future returns. For instance, [Daniel and Titman, 1997] were unable to confirm Fama and French's findings using a collection of monthly U.S. data from the NYSE, AMEX, and NASDAQ covering the years 1963–1993. [Griffin and Lemmon, 2002] also discovered that the Fama-French three-factor model was country-specific and needed to integrate local characteristics to more accurately forecast the returns of diversified portfolios.

More recently, a five-factor model was presented by [Fama and French, 2015], who added two extra components to the three-factor model to account for profitability and investment. The elements Robust minus Weak (RMW) is determined by the difference in returns between businesses with strong (high) and weak (low) operational profitability, as well as the investment component. There was developed a concept known as Conservative minus Aggresive (CMA), which measures the difference between the returns from companies that invest prudently and firms that invest proactively.

In this research, we suggest an alternative estimation method for the Fama-French three and five-factor models through the use of ML algorithms. To be more specific, we apply a combination of Support Vector Regression (SVR) and mixed Grid Search (GS) with 10-fold Cross Validation (CV) optimization to forecast portfolio returns for the

United States using the three and five factors presented by Fama and French. In line with the principle of structural risk minimization, SVR is a statistical ML toolkit that aids in model prediction. Cross-validation (CV) is used to evaluate the predictive power of those models, and comprehensive grid search (GS) is used to study all possible combinations of the hyperparameters of SVR models. We can simply map both linear and non-linear relationships between our dependent variable (output) and the variables (inputs) by using the two models.

We will try to demonstrate that the SVR models have been found to be quite successful when applied to the analysis of portfolio returns in the United States from July 1926 to November 2022 for five industries. This is especially so when taking into consideration the presence of collinearity among the factors, an issue that ML approaches are well-suited to address.

Main contributions

- Hybrid framework. We apply a combination of SVR and mixed GS with 10-fold CV optimization to improve the prediction accuracy of traditional linear models (OLS). The ML technique enables us to address problems with factor collinearity that have been documented in the finance literature.

- Traditional (OLS) versus ML approach comparison. In the same study, we can see the compared performance analysis of two of the most famous financial asset pricing models, using econometric and ML approaches. For this purpose, we will use a battery of different performance measures.

Research Data

We employ monthly data taken from the Kenneth French Data Library from the period of July 1926 to November 2022 to re-evaluate the Fama-French three-factor model. With regard to the Fama and French 3-factor (FF-3), we use three measurements in particular: Small minus Big (SMB) (average return of the three smallest portfolios minus the average return of the three biggest portfolios), HML (average return of the two highest-value portfolios minus the average return of the two growth portfolios), and (Rm-Rf) (excess return on the market), weighted in terms of return of all CRSP companies with US registration, listed on the NYSE, AMEX, or NASDAQ, with price data at the start of t and good return data for t less the 1-month Treasury bill rate (from Ibbotson Associates), and with a CRSP share code of 10 or 11 at the start of the month. As explanatory variables, these factors are employed. We will utilize the additional returns from the five industries that make up the S&P 500 as our dependent variables. These industries are: consumption, manufacturing, high technology, health, and others (mining, construction, transportation, hotels, entertainment, and finance).

Table 4.18: Summary statistics. SMB: Small Minus Big; HML: High Minus Low; RMW: Robust Minus Weak; CMA: Conservative Minus Aggressive. CRF: Consumption Industry; MANUF: Manufactury Industry; HITEC: High Technology Industry; HLTH: Healthcare Industry; OTHER: Other Industries.

Variables	**Obs.**	**Mean**	**Std. Dev.**	**Max.**	**Min.**
SMB	1,157	0.19	3.17	36.56	-17.23
HML	1,157	0.36	3.56	35.61	-13.97
RMW	713	0.28	2.22	13.09	-18.73
CMA	713	0.30	2.05	9.05	-6.94
$(R_m - R_f)$	1,157	0.67	5.35	38.85	-29.13
$(R_i - R_f)$ CRF	1,157	0.74	5.29	43.57	-28.61
$(R_i - R_f)$ MANUF	1,157	0.70	5.53	43.43	-30.89
$(R_i - R_f)$ HITEC	1,157	0.71	5.60	33.81	-26.81
$(R_i - R_f)$ HLTH	1,157	0.82	5.53	37.03	-34.14
$(R_i - R_f)$ OTHER	1,157	0.65	6.41	58.64	-30.08

In the case of the 5-factor from Fama and French (FF-5), we consider two additional measurements (see section 2.2.1): RMW and CMA, which are calculated, respectively, as the gap between the returns on portfolios of stocks of companies with strong and poor profitability, as well as the difference between the returns on portfolios of stocks of businesses with high and low investment levels. In the case of these two last factors, the available sample is clearly shorter. It comprises from July 1963 to November 2022.

The Kenneth French Data Library is also the source of the data for SMB, HML, RMW and CMA factors. Statistics are summarized in Table 4.18. The average values for SMB and HML factors are 0.2 and 0.4, respectively, with standard deviations of 3.2 and 3.6. RMW and CMA factors have averages of 0.3 and 0.3, and standard deviations of 2.2 and 2.1. The lowest figures for SMB, HML, RMW and CMA factors are -17.2, -14.0, -18.7 and -6.9, respectively; and the highest values for them are 36.6, 35.6, 13.1 and 9.0, respectively. The calculation of RMW and CMA factors involves the difference in returns between portfolios of stocks from companies with high and low profitability, as well as the gap between the returns on portfolios of stocks of firms with high and low investment levels.

Estimation models

Support Vector Regression (SVR) [Cortes and Vapnik, 1995] developed the supervised ML technique known as SVR. The idea of structural risk reduction later established by [Vapnik et al., 1996] serves as the foundation for the model. SVR is a powerful statistical technique with several benefits for model predictions. It first implements the notion of structural risk reduction by decreasing the upper bound of the anticipated risk, boosting the predictive capability of the SVM. This is done when the number of observations and inputs (independent variables) is constrained. Second, mapping using SVR is feasible for both linear and non-linear relationships between the inputs and outputs (dependent variables), making it one of the most effective prediction strategies.

SVR is a modeling and prediction-related application of SVM. The fundamental idea of SVRs is to display a single output variable, y, as a function of n input variables x using a function $f(x)$ that, for the whole training dataset, has an ϵ maximum deviation. The simplest version of such a function is presented as a linear connection in Equation 4.15:

$$f(x_i) = \omega^T \phi(x_i) + b; \quad \omega \in \mathbb{R}^n, b \in \mathbb{R} \tag{4.15}$$

where $f(x_i)$ is the output, ω is a weight vector, $\phi(x_i)$ is a non-linear mapping function, and b is a constant called intercept vector. To determine the minimum error or deviation, ϵ, the lowest weight vector could be obtained by minimizing the Eucledian norm $||\omega||^2$. The problem becomes:

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}||\omega||^2 \\ \text{subject to} \quad & y_i - \omega^T \phi(x_i) - b \leq \epsilon \\ & \omega^T \phi(x_i) + b - y_i \leq \epsilon \end{aligned} \tag{4.16}$$

Equation 4.16, however, can only be accurate to a certain extent when f exists for all pairings (x_i, y_i) with a specific precision ϵ. There can be included certain flaws in the equation known as slack variables (ξ_i and ξ_i^*), which allow for flexibility and go against these restrictions. Slack variables are added to the minimization process when the restrictions in Equation 4.16 are broken. Likewise, in the non-separable situation. The slack variables also guarantee the existence of a solution and permit input from sources other than the loss function being trained. The current issue is:

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}||\omega||^2 + C \textstyle\sum_i(\xi_i + \xi_i^*) \\ \text{subject to} \quad & y_i - \omega^T \phi(x_i) - b \leq \epsilon + \xi_i \\ & \omega^T \phi(x_i) + b - y_i \leq \epsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0, i = 1, 2, ..., n \end{aligned} \tag{4.17}$$

where the newly included parameter C, also known as the regularization parameter, is a positive constant that handles data points that are outside of the ϵ-sensitive loss margin. The trade-off between model complexity and the precision of the predictions made using training and test data is also considered by this penalty.

Through the Lagrangian method, we can solve Equation 4.17 as follows:

$$
\begin{aligned}
L = \frac{1}{2}||\omega||^2 + C\sum_i(\xi_i + \xi_i^*) \\
- \sum_i \alpha_i(\epsilon + \xi_i - y_i + (\omega, x_i) + b) \\
- \sum_i \alpha_i^*(\epsilon + \xi_i^* + y_i - (\omega, x_i) \\
- \sum_i(\eta_i\xi_i + \eta_i^*\xi_i^*)
\end{aligned} \tag{4.18}
$$

where α_i, α_i^*, η_i and η_i^* are the Lagrangian multipliers which are positive numbers. Using the first order conditions and eliminating the Lagrangian multipliers allow us to find the value of the weight vector ω given by:

$$
\omega = \sum_i(\alpha_i^* - \alpha_i)K(x_i, x) + b \tag{4.19}
$$

where $K(x_i, x)$ is the kernel parameter that allow our model to transform into a higher dimensional space. Some kernel functions used in the literature are the following:

- 1. linear model, with the kernel function $K(x_i, x_j) = (x_i, x_j)$,
- 2. Gaussian kernel also called the radial basis function (RBF), with $K(x_i, x_j) = exp(\frac{-||x_i - x||^2}{2\sigma^2})$ or $K(x_i, x_j) = exp(-\gamma||x_i - x||^2)$, where $\gamma = \frac{1}{2\sigma^2}$
- 3. polynomial kernel $K(x, y) = ((x_i, x_j) + 1)^d$

In order to estimate the SVR model in an efficient manner, three parameters (namely the cost parameter C, the error ϵ and the kernel parameter γ) must be chosen accurately. They are also named hyperparameters, because unlike parameters, hyperparameters are set before training a ML model. The penalty parameter C should not be either too little or too high since if it is, the model will not penalize the training data enough or, if it is, will penalize the training data too much. The loss function ϵ emphasizes the quantity of support vectors and also affects how well the SVR model performs. To assess the capacity of the model to handle non-linear instances, the kernel parameter γ transfers linear functions into higher-dimensional space.

Grid Search and Cross-validation optimization To find efficient hyperparameters of our SVR models, namely C, ϵ, and γ, we use the combined GS and Minimum Cross Validation (MCV) method.

The GS algorithm offers a technique for determining a model's optimal hyperparameters. Contrary to parameters, hyperparameters are not present in training data. In order to identify the right hyperparameters, we develop a model for each group of them. Since we are essentially forcing all potential combinations, GS is regarded as a fairly conventional hyperparameter optimization technique. The models are then tested against one another. Naturally, the model with the highest accuracy is regarded as the best.

CV is a method for assessing the ability of a model to generalize to different datasets. To avoid over-fitting, S-fold CV is advised rather than leave-one-out CV. In the training of an S-fold CV, N groups of data are randomly created from the data, and they have identically sized divisions, which are called folds. Then, one group (independent dataset) is utilized as the data to test the model built using the information from the other $S-1$ groups (training dataset). This process is carried out S times such that the information of the validation data is utilized once for each of the S groups. Finally, it is possible to acquire predicted values of y rather than computed values. In this exercise, S is set to 10, which is a standard value.

Finding the SVR regression model with the best generalization is our aim. Each SVR regression is determined using training data D and a set of hyperparameters. To accomplish our objective, we must identify the optimal collection of hyperparameters, and to do so, we require validation data V that is unrelated to D. Since data size is frequently severely constrained, we use the cross-validation approach.

During the cross-validation process, given data D is randomly divided into S separate segments $\{G_s, s = 1, ..., S\}$. $S-1$ segments are used for training, while the remaining segment is used for testing. By altering the remaining section, this procedure is carried out N times, and the generalization performance is assessed using the MSE stated in Equation 4.20:

$$\mathrm{MSE}_{CV} = \frac{1}{N}\sum_{s=1}^{S}\sum_{\mu\in G_s}(y^{\mu} - y(x^{\mu}|\hat{\theta}_s))^2 \tag{4.20}$$

where G_s denotes the s-th segment for the test, and $\hat{\theta}_s$ denotes the optimal parameter vector obtained by using $D - G_s$ for training. The extreme case of $S = N$ is known as the *leave-one-out* (LOO) method, often used for a small size of data.

By using CV, the hyperparameters are optimized so that the CV error MSE_{CV} is minimized. The MCV method, which learns a different penalty factor for each weight, functions well when NNs are regularized. The following approach may be used to generalize the MCV process (see [Ito and Nakano, 2003]): let λ be a hyperparameter vector, and a target function for optimizing λ is MSE_{CV}. Let θ and $J(\theta|\lambda)$ be a parameter vector of a model and a target function to train θ under the given hyperparameters λ respectively. The MCV method employs the coordinate descent; that is, iterates the following two steps until convergence:

- step 1: Given λ, optimize $\{G_s, s = 1, ..., S\}$.
- step 2: Given $\{G_s, s = 1, ..., S\}$, improve λ.

Table 4.19: Factor regression results of the 3- and 5-factor Fama-French models using Ordinary Least Squares (OLS). The sample period is from 1926.07 to 2022.11. Training set represents 70% of the total dataset. The value in parenthesis are the t-statistics.

OLS Models										
Variable	**Consumption**		**Manufacturing**		**High Tech**		**Health**		**Other**	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Fama-French 3-factors										
Intercept (α)	0.139**	0.138	-0.014	0.069	0.121*	0.092	0.314***	0.382**	−0.226***	−0.176**
	(2.253)	(1.380)	(-0.368)	(0.621)	(1.883)	(0.832)	(2.762)	(2.370)	(-3.385)	(-2.155)
Market-Rf	0.967***	0.832***	1.017***	0.883***	0.915***	1.198***	0.941***	0.697***	1.039***	1.109***
	(81.308)	(36.911)	(135.206)	(35.511)	(73.451)	(48.171)	(42.802)	(19.242)	(80.729)	(19.646)
SMB3	0.083***	−0.107***	−0.087***	-0.012	-0.019	0.063*	-0.019	−0.176***	0.173***	−0.111***
	(3.957)	(-3.291)	(-6.571)	(-0.337)	(-0.879)	(1.768)	(-0.487)	(-3.387)	(7.648)	(-4.237)
HML3	−0.164***	0.089***	0.097***	0.344***	−0.174***	−0.499***	−0.287***	−0.084*	0.306***	0.489***
	(-6.469)	(2.904)	(8.561)	(10.216)	(-9.314)	(-14.791)	(-8.678)	(-1.716)	(15.810)	(19.646)
R^2	0.908	0.801	0.966	0.800	0.884	0.894	0.715	0.525	0.925	0.921
$Adj.R^2$	0.908	0.800	0.966	0.798	0.883	0.893	0.714	0.520	0.925	0.920
$F-Test$	2668***	462.3***	7560***	457.8***	2041***	961.1***	674.7***	126.1***	3310***	1328***
Fama-French 5-factors										
Intercept (α)	−0.237***	0.048	−0.166*	-0.096	0.398***	0.130	0.242	0.284	−0.209***	-0.083
	(-3.041)	(0.477)	(-2.412)	(-0.685)	(3.961)	(1.369)	(1.735)	(1.669)	(-2.701)	(-0.901)
Market-Rf	1.044***	0.906***	0.970***	0.962***	0.931***	1.089***	0.899***	0.761***	1.157***	1.069***
	(54.449)	(37.858)	(57.401)	(29.177)	(37.611)	(48.297)	(26.152)	(18.878)	(60.619)	(48.843)
SMB5	0.172***	0.021	-0.029	0.110*	−0.076**	-0.020	−0.144***	-0.027	0.125***	-0.068
	(6.847)	(0.474)	(-1.294)	(1.777)	(-2.369)	(-0.464)	(-3.213)	(-0.358)	(5.006)	(-1.652)
HML5	0.142***	−0.187***	0.127***	0.132**	−0.195***	−0.254***	−0.533***	−0.301***	0.421***	0.551***
	(3.671)	(-4.648)	(3.720)	(2.372)	(-3.897)	(-6.669)	(-7.678)	(-4.417)	(10.907)	(14.909)
RMW5	0.482***	0.203***	0.247***	0.260***	−0.518***	0.013	0.447***	−0.208**	0.215***	−0.235***
	(13.898)	(3.595)	(8.081)	(3.328)	(-11.551)	(0.253)	(7.191)	(-2.186)	(6.219)	(-4.539)
CMA5	0.158***	0.126*	0.242***	0.197**	−0.402***	-0.102	0.426***	0.326***	−0.129**	−0.228***
	(2.912)	(1.872)	(5.021)	(2.128)	(-5.701)	(-1.615)	(4.355)	(2.878)	(-2.373)	(-3.714)
R^2	0.881	0.890	0.882	0.846	0.859	0.932	0.670	0.669	0.903	0.948
$Adj.R^2$	0.880	0.887	0.881	0.842	0.857	0.931	0.667	0.661	0.902	0.946
$F-Test$	732.3***	336.7***	740.8***	228.2***	597.7***	572.3***	200.2***	84.09***	920.6***	748.8***

* p < 0.10, ** p < 0.05, *** p < 0.01

Results

OLS models Table 4.19 shows the estimation results of the 3- and 5-factor Fama-French models using the traditional optimization method of OLS.

The first point to highlight is the notable difference in goodness of fit between the training and testing datasets, except for the high-tech sector in the 5-factor model, where the results are very similar in both datasets. In any case, the significance of the variables is clearly higher in the training datasets in a general sense, as reflected in the F-Test values. This suggests that, over the years, this type of framework to explain return differences among sectors and assets no longer works as it did in the past. Among sectors, the health sector exhibits the worst fit and overall statistical significance, with a noticeable difference compared to the other four, which show very high levels in both aspects.

Table 4.20: Factor regression results of the 3- and 5-factor Fama-French models using Support Vector Regression (SVR) algorithm, optimized using 10-fold cross validation optimization. The sample period is from 1926.07 to 2022.11. Training set represents 70% of the total dataset.

SVR Models										
Weight	**Consumption**		**Manufacturing**		**High Tech**		**Health**		**Other**	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Fama-French 3-factors										
Intercept (b)	-1.178	0.994	-1.811	0.648	-0.985	0.519	-0.852	-0.479	-3.231	0.282
ω_Market	198.9	26.458	201.4	50.0	160.4	66.9	185	47.8	202.8	40.27
ω_SMB3	59.6	0.524	23.0	-17.0	56.5	34.8	65.8	-10.8	34.5	-29.71
ω_HML3	129.2	10.241	33.8	25.5	95.3	-14.3	93.7	-21.3	61.2	5.27
bR^2	0.578	0.617	0.596	0.600	0.578	0.723	0.355	0.335	0.522	0.733
Fama-French 5-factors										
Intercept (b)	0.220	0.204	-0.101	0.325	0.480	0.069	0.256	0.495	-0.289	0.260
ω_Market	102.96	77.746	87.7	52.3	121.8	53.16	67.47	51.32	88.1	60.23
ω_SMB5	33.25	12.257	-11.23	13.4	40.5	9.81	14.34	16.87	-15.2	20.12
ω_HML5	6.97	-0.489	-17.43	2.3	-34.9	23.71	-50.77	-3.22	-15.4	36.81
ω_RMW5	3.78	-12.818	5.91	17.7	-57.1	-18.5	9.24	14.15	24.3	-3.18
ω_CMA5	-16.43	-19.313	13.95	12.2	-59.3	-9.24	-36.61	-11.75	-26.8	-3.13
bR^2	0.672	0.673	0.673	0.600	0.650	0.771	0.476	0.480	0.743	0.734

Regarding the comparison between the 3-factor and 5-factor models, the 3-factor model shows more accurate fit and significance figures. In this latter case, only in the manufacturing sector some of the factors result in non-significance when working with the testing dataset.

For the 5-factor case, concerning the testing dataset, the opposite occurs: only in the manufacturing sector, all five factors of the model remain statistically significant. From the point of view of economic significance, except in the case of the market factor, there is not an unanimous answer for the different factors. Although positive sign is predominant, there exist some factors with negative loadings, both in 3- and 5-factor models.

SVR models In Table 4.20 we can observe the global results using SVR models. Specifically, the results shown for each industry are those obtained after the application of the combined hyperparameter optimization method of Grid Search (GS) and Minimum Cross Validation (MCV) (GS-CV-SVR model). Three parameters, namely the cost parameter C, the error parameter ϵ, and the kernel parameter γ, have been precisely selected using this combined model in order to estimate the optimum SVR model in an effective manner.

In most of cases, the kernel function used in this optimization has been the Gaussian kernel, also called the radial basis function (RBF). By industries, the health sector shows the worst figures of fit, both in 3- and 5-factor models, showing the remaining sectors very similar metrics in terms of bR^2.

Table 4.21: Performance statistics results of the 3- and 5-factor Fama-French models using both OLS estimation and optimized SVR algorithm. The sample period is from 1926.07 to 2022.11. Training set represents 70% of the total dataset. RMSE: Root of Mean Squared Errors, MAE: Mean of Absolute Errors.

Fama-French 3-factors										
OLS										
Measure	Consumption		Manufacturing		High Tech		Health		Other	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
RMSE	1.729	1.832	1.094	2.020	1.811	2.022	3.197	2.942	1.871	1.491
MAE	1.265	1.388	0.795	1.571	1.368	1.508	2.333	2.238	1.307	1.113
SVR										
RMSE	1.526	1.567	1.026	1.752	1.644	1.769	2.835	2.515	1.639	1.236
MAE	1.127	1.133	0.757	1.287	1.231	1.239	2.054	1.873	1.221	0.921
Fama-French 5-factors										
OLS										
Measure	Consumption		Manufacturing		High Tech		Health		Other	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
RMSE	1.631	1.394	1.438	1.921	2.107	1.314	2.924	2.349	1.623	1.275
MAE	1.237	1.080	1.060	1.477	1.632	1.069	2.244	1.837	1.213	0.961
SVR										
RMSE	1.243	0.761	0.982	1.256	1.460	0.822	2.128	1.639	1.028	0.948
MAE	0.908	0.606	0.693	0.875	1.087	0.638	1.537	1.117	0.772	0.716

Table 4.22: Correlations for 3-Factor and 5-Factor models for the testing (i.e. out-of-sample) and training (in-sample) datasets. The sample period is from 1926.07 to 2022.11. Training set represents 70% of the total dataset.

Model	Consumption		Manufacturing		High Tech		Health		Other	
Fama-French 3-factors										
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
OLS	0.953	0.895	0.983	0.894	0.940	0.945	0.846	0.724	0.962	0.960
SVR	0.964	0.925	0.985	0.922	0.950	0.959	0.881	0.808	0.971	0.972
Fama-French 5-factors										
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
OLS	0.938	0.943	0.939	0.920	0.926	0.966	0.819	0.818	0.950	0.973
SVR	0.965	0.983	0.972	0.967	0.965	0.987	0.908	0.916	0.980	0.990

Performance analysis The performance comparison between the two estimation methods can be evaluated in Table 4.21. The reported statistics consist of the RMSE and the MAE. The most relevant conclusion in the light of this report is that the optimized SVR model outperforms OLS model in every single industry, both in 3- and 5-factor Fama-French models, and also for both the training and testing datasets.

For testing and training datasets of both models, the manufacturing industry has the lowest RMSE and MAE values. The health sector, on the other hand, has the highest values, supporting the conclusions drawn from prior estimation statistics, such as OLS or SVR models, in that it appears to have the lowest goodness of fit and the greatest modeling difficulties.

We report in Table 4.22 the correlation coefficients between actual and predicted values, for the 3- and 5-factor models across industries, for the training and testing datasets, using the traditional OLS model and, alternatively, the GS-CV-SVR model. For portfolio returns in the consumption industry, the correlations between experimental (i.e., observed) and predicted (i.e. SVR) values are 96% and 93%, respectively; for manufacturing, the correlations are 99% and 92%, for high-tech, they are 95% and 96%, for the health sector, they are 88% and 81%, and for other industries, they are 97% and 97%. The Fama-French 3-factor model may produce great predictions when used with our ML algorithm of SVR. We can vouch for the fact that the correlation coefficients for this model range from 81% for the health sector to 99% for the manufacturing sector. The Fama-French 5-factor model has correlation coefficients that are considerably better, ranging from 91% (for the health business) to 99.9% (for the high-tech industry). As can be seen, the SVR 5-factor model performs better than the SVR 3-factor model.

Anyway, GS-CV-SVR outperforms clearly the prediction performance from the OLS models. This is specially remarkable in the case of the health industry, the sector which has shown more resistance to be modeled and fitted in previous tests. The discrete correlation coefficients from OLS, specifically 72% and 85% for training and testing datasets, climb to 81% and 88%, in the case of the 3-factor model, and from 82% and 82% to 92% and 91%, respectively, which can be considered as brilliant. The improvement, something more modest, is present in each one of the remaining industries, both in 3- and 5-factor versions.

This prediction performance enhancement is also evident when we observe Figures 4.16 and 4.17, where we show the XY-plots between actual and predicted values across the five industries, for OLS (left side) and SVR (right side). In the case of industries as Consumption or High-Tech, the correlation reaches 99% for the 5-factor model.

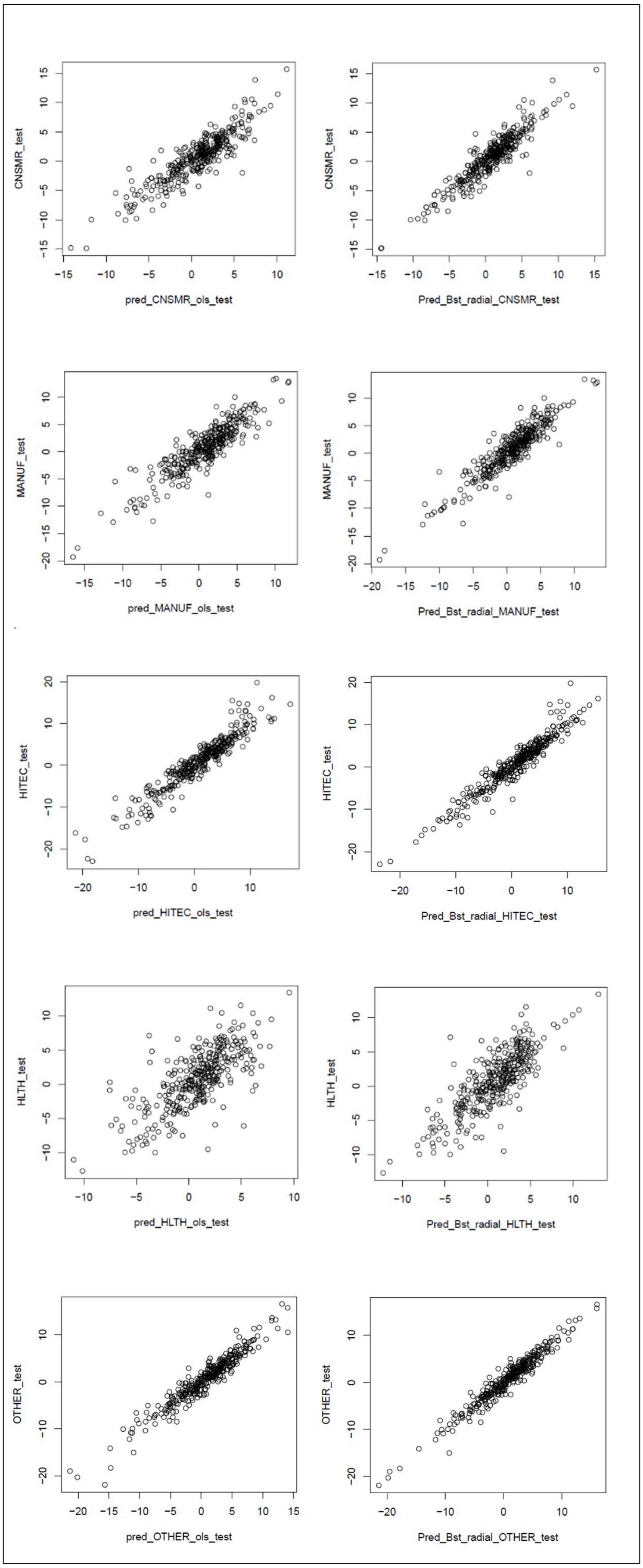

Figure 4.16: Actual versus predicted values in testing datasets. 3-factor Fama-French model.

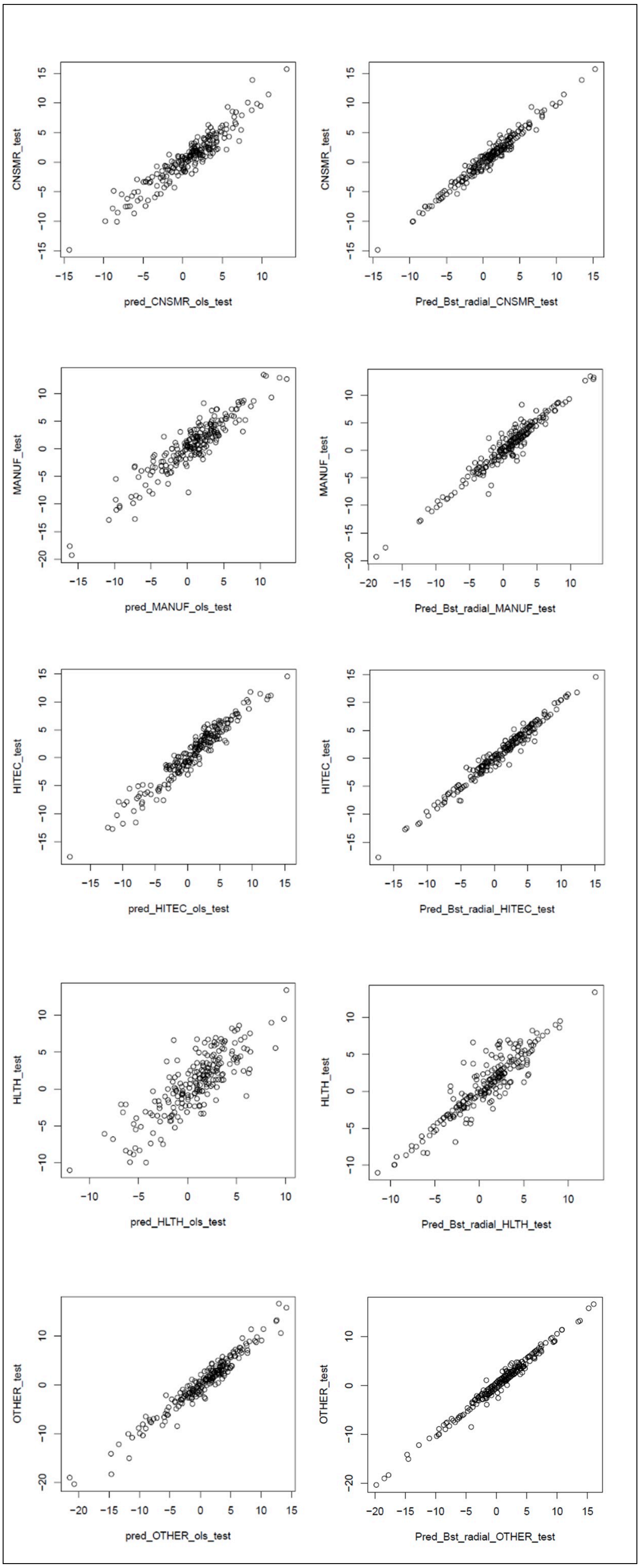

Figure 4.17: Actual versus predicted values in testing datasets. 5-factor Fama-French model.

Discussion

This study have effectively used ML to improve the estimation procedure for the three- and five-factor Fama French models. To be more exact, we have applied a combination of SVR and mixed GS with 10-fold CV optimization to forecast monthly portfolio returns for the US stock market sectors using the three and five factors presented by [Fama and French, 1993], [Fama and French, 2015]. We have discovered the following findings for a dataset of portfolio returns across different industries in the United States from July 1926 to January 2019.

First, our GS-CV-SVR three-factor estimations produced out-of-sample (testing datasets) correlation coefficients of 97% for portfolio returns for the mining, construction, transportation, hotels, entertainment, and finance sectors, 96% for the high-tech sector, 92% for manufacturing and consumption sectors, and 81% between predicted and experimental (i.e. observed) values of portfolio returns for the health sector. In comparison with the traditional approach of OLS optimization, the results suggest that the prediction accuracy improvement is quite remarkable. Very similar conclusions can be reached if the comparison criteria to be used are traditional predictive accuracy statistics as RMSE or MAE.

Second, the GS-CV-SVR Fama-French five-factor model performs even better than the GS-CV-SVR Fama-French three-factor model, with correlation coefficients ranging from 92% to 99% across industries. This conclusion differs from a significant portion of literature about this topic (for instance, [Diallo et al., 2019]).

5. CONCLUSION

5.1. Methodological conclusions

This first part of this chapter about conclusions will be focused on the first part of the thesis. After reviewing the recent literature, the selected datasets, methods, and performance criteria, we will take a step back to analyze them at a higher level of abstraction. This discussion will comprise three aspects: firstly, we will provide an overview of the state of the art of ML for asset management; second, we will highlight the most successful data, methods, and criteria for each financial discipline we reviewed; at last, we will lay down the existing challenges which might motivate further research.

5.1.1. Overview

The traditional approach (statistics and econometrics) to solving asset management issues has been the focus of academics and practitioners alike throughout the last 50 years. However, it has also exhibited many drawbacks. Next, we will describe shortly the main pitfalls and challenges that this discipline is currently addressing:

- Researchers are sometimes compelled to present incomplete results that are often refuted by additional studies due to the publication bias towards successful results (see [Harvey, 2017]). As a result, replication is critical, and many academic findings have a very short expiration date, especially if transaction costs are considered [Cakici and Zaremba, 2021].
- One of the main pitfalls of the traditional econometric approach has to do with the p-hacking. As it was demonstrated by [Chen, 2019], p-hacking alone cannot account for all the anomalies documented in the literature. One way to reduce the risk of spurious detection is to increase the hurdles (often, the t-statistics) but the debate whose title might be "the factor zoo" is still ongoing [Harvey and Liu, 2019].
- Because of its easy understanding, the decomposition of returns into linear factor models is extremely useful. Nonetheless, there is an eternal dispute in the academic literature as to whether business returns are explained by exposure to macroeconomic variables or merely by firm characteristics. Until the new century the factor-based explanation for risk premium was the favorite, but after the seminal work by [Daniel and Titman, 1997], the characteristics-based explanation has become a great competitor of the traditional outlook. The second experiment of the thesis (section 4.2.2) is focused on the pairs trading strategy, but uses as additional feature a set of firm characteristics. The third empirical exercise (section 4.2.3) focuses

on a ML alternative to estimate and forecast the three- and five-factor models from Fama and French.

- Some researchers have observed fading anomalies as a result of publication: once an anomaly is made public, agents invest in it, driving up prices and causing the anomaly to vanish. [David McLean and Pontiff, 2016] documents this impact in the United States, while [Jacobs and Mülle, 2020] finds that post-publication factor returns are sustained in other relevant markets. Herding may be destroying factor premia [Krkoska and Schenk-Hoppé, 2019], and the democratization of so-called smart-beta products (particularly the ETFs) that enable investors to actively invest in specific styles (value, low volatility, etc.) may speed the process up.

- Researchers have developed more sophisticated techniques to organize the so-called factor zoo and, more significantly, to detect false anomalies. [Feng et al., 2020], for example, use LASSO selection and Fama–MacBeth regressions to see if new factor models are worthwhile. They calculate the benefit of adding one new factor to a set of preset factors, demonstrating that many of the factors described in papers published in the 2010 decade do not provide much extra value.

- There is no such thing as a flawless approach, but the sheer volume of contributions in the field emphasizes the importance of robustness. The notion that factors are likely to change over time is a key obstacle for short-term strategies. We refer for instance to [Cooper and Maio, 2019].

- As we have seen in the Section 3.2 about price forecasting, the difficulty to test consistently, using traditional approaches, the EMH, leaves a huge space to alternative techniques. We have tested two in this thesis: a hybrid model EMD+GRU to forecast daily prices in the US Equity market (section 4.2.1), and another hybrid model Autoencoders+k-means+DTW to create a pairs trading system (section 4.2.2).

- In the case of MVO, as [Cochrane, 2011] points out, even though it is not a particularly useful guide to computation, classic one-period mean-variance analysis is a brilliantly useful characterization of an optimal portfolio, useful for final investors to understand and think hard about risk allocations. Even when investors are considering highly non-normal payoffs, traditional mean-variance analysis continues to dominate portfolio applications. Nevertheless, many researchers have tried to improve the suitability of this model from different perspectives.

In conclusion, traditional financial economics has no perfect answers to all these pitfalls and challenges described. On the other hand, ML techniques have found an excellent breeding ground to develop all its potentialities. As [Cerniglia and Fabozzi, 2020] point out, financial theory, market behavior, ever-increasing data sources, and computational innovation are all required for good forecasting and pricing. By putting together the most comprehensive toolbox, you can create realistic computational models. This goal may be achieved using both financial econometrics and ML techniques. According to these authors, ”ML tools provide the ability to make more accurate predictions by accommodating nonlinearities in data, understanding complex interaction among variables, and allowing the use of large, unstructured datasets. The tools of financial econometrics remain critical in answering questions related to inference among the variables describing economic relationships in finance; when properly applied, their role has not diminished with the introduction of ML”.

As we have reviewed, and according to [Song et al., 2017], ML algorithms are commonly employed for financial market forecast and trading strategies. There are three different sorts of applications. The first sort of application forecasts asset prices or returns in the future. Generally, SVR and NN algorithms are used in this type of strategies. The high number of applications of LSTM models is especially remarkable where time-series come into play. The drawback with this strategy is that it has a high error rate owing to the difficulty in predicting future asset values based on erratic financial market data. The second type uses classification algorithms to anticipate price movement directions, such as SVM and DTs. These approaches generally have significant forecast accuracy, but this does not always imply high profitability. For example, a model can anticipate small gains properly but massive losses wrongly, resulting in a substantial downside risk. Rule-based optimization is the third type. Its goal is to find the best trading indicator and parameter combinations (for example, technical indicators, fundamental indicators, and macroeconomic indicators). Optimization algorithms that have been explored include GP and RL.

To sum up, the trend indicates that ML algorithms clearly outperform traditional econometrics approaches. The empirical part of our thesis also seems to confirm this trend. Both classical and modern ML methods have been applied successfully. However, the main pitfall is that the landscape is extremely diverse in terms of data and applied methods, which suggests a lack of common benchmarks, methodologies, and frameworks.

5.1.2. Discipline Focus

After providing an overview for the field as a whole, we will increase our focus to discuss each one of the reviewed disciplines (including the extra algorithmic trading) from the ML perspective.

- Price Forecasting: It is the most heterogeneous discipline, where all sorts of ML methods have been applied to either refine the output of other algorithms, to generate predictions on its own or even as a technique to process alternative data sources. A few works make use of social media, financial news, and sentiment analysis to increase prediction accuracy. There is no dominant paradigm in this discipline.

- Algorithmic Trading: In this case, most reviewed papers make use of SL to train architectures more typically suited to target other domains. For instance, CNNs (which are common in image processing scenarios) are applied to specially pre-processed financial data with success. Oftentimes, they are also coupled with RNNs techniques to model time dependencies, usually applying LSTM.

- Value/Factor investing: The landscape is not specifically dominated by any particular technique. PCA is successful in most works as a pre-processing technique whilst other classical ML methods like RFs, SVMs, or shallow NNs are present in almost all the analyzed works. RNNs does not have much presence in this discipline. The paradigms are mainly SL and UL.

- Portfolio Management: In this discipline, we observed a trend of favoring RNNs architectures to model long-term dependencies of financial time-series data. In particular, most of the reviewed methods make use of variations of LSTMs (usually combining them with other techniques like MVO). RL methods appear in this discipline coupled with RNNs in the form of LSTMs in the most recent works. The dominant paradigms are SL and RL.

5.1.3. Challenges and Opportunities

We have identified the following main challenges and opportunities:

- Standard Datasets: The whole field is characterized by a lack of curated datasets to be reused by the community. Although some datasets are built upon the portfolio database of [Fama and French, 1993], most of them deviate from this standard. Furthermore, even those which reuse that database end up diverging in terms of the final data available for investigation. Therefore, creating a standard database (complete and broad enough) to be reused by the research community is a need for further works.

- Reproducibility: No common methodology or framework for method training and benchmarking has been established. This hurts reproducibility since most of the analyzed methods are difficult, if not impossible, to compare against each other (unless reimplemented specifically for each scenario). In addition, almost no paper includes codes or data to be accessed by other researchers. Establishing a reproducibility framework for asset management ML research is a high-impact workstream for improving the quality of life and pace of the research community.

- Multimodal Data: Most methods are focused on analyzing numerical financial data to generate predictions. Analyzing alternative sources of information like news, social media, sentiment, and user-generated content can provide useful cues for financial decisions. Few works make use of those data sources at the moment. The challenge of combining all those multimodal sources and multiple architectures might unlock new levels of prediction accuracy.

- Heterogeneous Architectures: Arguably due to the state of immaturity in which financial ML sits nowadays (with regard to other more established synergies like image processing or NLP), no clear architectures for processing financial data have been established yet. There is a broad range of papers that spawn new models, and few that build upon solid groundwork to improve them. Finding the common patterns and unifying those diverse architectures could have a beneficial effect to the community for broad adoption in industry (in a similar way as other networks, such as UNet or ResNet, have done for image processing by becoming the de facto standard for many applications).

- Algorithmic trading: This application field is characterized by a very interesting trade-off. From an academic perspective, this area is relatively disconnected from the theoretical background about asset pricing and value investing, which has had a central role in financial economics during the last five decades. However, in return, and precisely because of this characteristics—exclusive dependence on price data—it is the financial discipline that can maximize the contributions of ML applications. We can find a future challenge in the possibility of combining both issues, deepening trading algorithms with a higher relevance of financial fundamentals.

5.2. Empirical conclusions

After this global analysis of the methodological conclusions, in this new section we will try to extract the most relevant conclusions about the empirical section of this thesis. As we commented in the introduction of Chapter 4, the empirical part of this thesis has been dedicated to present representative examples of how ML can help to better understand the behavior of certain markets, more accurately predict the behavior of their prices, better capture the structure of certain investment strategies, or significantly improve the predictive capacity of multi-factor models of financial asset valuation.

5.2.1. Price Forecasting

The main objective of every single investor or asset manager is forecasting prices accurately. The EMH claims that stock prices reflect all available information and are unforeseeable (i.e., the way they behave can be compared to that of a random walk, indicating that its movements are inevitably unpredictable), but evidence from numerous studies shows that the market is not completely efficient, especially with the use of machine learning technologies. Traditional econometric models like the ARIMA models and other sophisticated models, like the ARCH models, may not be suitable for forecasting stock values since stock prices are intrinsically nonlinear, non-stationary, and volatile.

With this aim we face the first empirical exercise of the thesis. We started the experiments using both US and Spain stock market indices, but we quickly realized that the American one exhibits the deepest problems to be modelled, mainly due to problems of non-stationarity. Nothing different from what happens in traditional approaches like ARIMA or ARCH. The non-stationarity of price series poses an initial obstacle that must be addressed before moving on to subsequent stages of estimation.

One of the main contributions of this empirical exercise is the use of the EMD transformation to avoid the problems of stationarity. EMD has demonstrated to be particularly useful in analyzing non-stationary situations, according to some previous findings. We get the same result. Once applied to our series, the results demonstrate that the initial signal of prices can be categorized into three groups based on their frequency components. The first group comprises high-frequency components with a large amount of noise, represented by the initial IMFs. The second group consists of middle-frequency components with medium noise, represented by the central IMFs. The final group comprises low-frequency components with minimal noise, represented by the latter IMFs.

Once we have decomposed the initial non-stationary series in a sum of IMFs, we can apply the ML algorithm to estimate and forecast the separated IMFs. We selected the Gated Recurrent Unit Networks (GRUs), a ML algorithm which is a type of RNN that has been designed to address the vanishing gradient problem in traditional RNNs. Similar to other RNNs, GRUs use feedback loops to process sequential data, but they incorporate gating mechanisms that allow them to selectively retain or forget information over time. We carried out some different experiments in order to check what is the best approach of all. One of those experiments was to include technical indicators, usually employed in financial markets, as additional features in the prediction model.

Finally, we concluded that the improvement showed after EMD application carried on with the use of individual GRU algorithms (EMD-IGRU model) to each of its components. For this purpose, we used two traditional performance measures, MAPE and DA. In terms of MAPE, the model with individual GRUs for each one of the IMFs was able to beat the figures of a plain vanilla Box-Jenkins model (ARIMA(1,1,1)). The results were very similar for the model with technical indicators as additional features. Nevertheless, in terms of DA, the results showed that almost every single model using a combination of EMD and GRU was better than ARIMA to predict future prices of S&P 500. Thus, the combination of EMD pre-processing technique and GRU enhances substantially the results from traditional univariate models, usually considered as the better approach to financial series as S&P 500 stock prices. The DA of ML approaches oscillates between 63.53% in the worst case and 75.15% in the case of the best model (Experiment 3). They are also above recent results in the literature, which uses some other approaches of ML or DL to model and predict S&P 500 stock prices. In most of cases, the DA figures do not exceed 70% in the best references. Good examples of that can be found in [Akiyoshi, 2020], [Nava et al., 2018], [Shi and Zhuang, 2019] or [Srijiranon et al., 2022].

The EMD-IGRU, the greatest model of all, has finally undergone a trading simulation stress test that mimics real-world conditions. The reaction of the model has been more than adequate, and by the usual statistical criteria used in trading systems, the model performed admirably in almost all forecast windows. This finding implies that the GRU predictive models have a respectable degree of accuracy in predicting stock prices and directional movements after being strengthened by the prior application of the EMD transformation for non-stationarity series. Additionally, it has proven successful in predicting actual profits in real-world settings.

5.2.2. Pairs Trading

Investments should aim to reduce risk while boosting returns. Due to its simplicity and market neutrality, the "pairs-trading" strategy —also known as statistical arbitrage— has been utilized more frequently in contemporary hedge funds. This strategy can be categorized as a "price forecasting" strategy, but the difference with the pure asset price prediction is that, in this case, the variable to estimate and forecast is the price difference between two stocks or assets. With this strategy, you establish a long position on the stock that increases its price and a short position on the stock that decreases it while keeping an eye on the correlation between two stocks that are known to be related. The core idea is that despite short-term divergence, the two stocks would eventually converge, allowing the trader to profit from the pair regardless of the state of the market.

Trading techniques for pairs often use cointegration or some other time series-related statistic to find pairs. This second experiment of the thesis looks at the viability of using unsupervised learning ML techniques to find possible stock pairs for a long-short portfolio. Nevertheless, considering that we are working with time-series data, each data point is an ordered sequence, and because of that, clustering various time series into comparable groups is a difficult operation. For this reason, besides using autoencoders for dimensionality reduction and additional clustering techniques as DBSCAN, we propose to apply an specific clustering approach to time series data (specifically, monthly data from 2010 for individual constituents of the US index S&P 500), without losing the temporal dimension, called k-means clustering with DTW. As an additional contribution, we suggest to use financial data (firm characteristics) as additional features in the pairs selection.

The results suggest that the number of clusters and their characteristics are very homogeneous throughout methods. The number of clusters has been optimized at 20, being the median number of stocks approximately 15-20 stocks, and the two biggest clusters conformed by between 40 and 60 stocks. Once stock pairs have been selected, a very simple trading strategy is defined, basically taking into account the one-month divergences in prices. Five different models of clustering and classification are revised once they are materialized in the trading strategy previously presented. The model using the k-means algorithm, but in its version adapted to time series, i.e., the k-means clustering integrated with DTW, shows the best results. Without capitalization of the monthly returns, it achieves a total return of 89% in 8 years, with a DA of 63%, a SR of 2.658, and a CR of 0.678. In comparison with the two benchmarks selected, the market and the short-term reversal strategy, this model keeps its privilege place in terms of total return, but also in terms of the different trading performance measures, mainly SR figures. The results are not improved when firm characteristics are used as additional features for clustering, what differs from some other academic references.

The final analysis of this second exercise is intimately connected with the third one, because we will use three factor models to check if systematic risk factors can account for the returns of long-short portfolios created through the most effective clustering algorithm used in this exercise. The three models that we consider are the Fama and French ([Fama and French, 1996]) three-factor model (FF3), the Fama and French ([Fama and French, 2015]) five-factor model, and the [Hou et al., 2020] q^5 factor model. The explanatory power of the models does not result so brilliant, around 30-40% in the three cases. In terms of statistical significance, factors as size, profitability, or investment factor are not significant in none of the three models, and only market beta factor or growth factor seem to show some level of significance. Therefore, it can be concluded that the extra returns achieved with our ML model and trading system can not be explained by excessive risk taking.

5.2.3. Factor Investing

As we have intensively explained throughout this thesis, both in chapters 2 and 3, the three- and five-factor models from Fama and French, can be considered as the most famous, productive, applied, interpreted and admired asset pricing models in history, maybe only behind the model they are inspired by, and they tried to improve as well, at least from an empirical point of view, the CAPM model.

After some years of strong criticism against the canonical model from [Sharpe, 1964], [Fama and French, 1993] suggested adding High Minus Low (HML) and Small Minus Big (SMB) as additional explanatory variables to explain the excess returns in the US stock market. Their model outperformed the original CAPM, which could only explain an average of 70% of portfolio returns with U.S. data, by being able to explain over 90% of portfolio returns. Some years later, the same authors ([Fama and French, 2015]), introduced two additional variables to answer some criticisms about the lack of consistency of their models to new data and new markets.

In the third exercise of the thesis, we take a some different way from the previous two exercises. The proposal is more methodological than empirical, because we suggest an alternative estimation method for the Fama-French three and five-factor models through the use of ML algorithms. To be more specific, we apply a combination of SVR and mixed GS with 10-fold CV optimization to forecast monthly returns in five industries of US S&P 500 using the three and five factors presented by Fama and French. This hybrid model is called GS-CV-SVR Model. SVR is a powerful statistical technique with several benefits for model predictions. It first implements the notion of structural risk reduction by decreasing the upper bound of the anticipated risk, boosting the predictive capability of the SVM. This is done when the number of observations and inputs (independent variables) is constrained. Second, mapping using SVR is feasible for both linear and non-linear relationships between the inputs and outputs (dependent variables), making it one of the most effective prediction strategies.

Our GS-CV-SVR three-factor model estimations produced out-of-sample (testing data sets) correlation coefficients of 97% for the mining, construction, transportation, hotels, entertainment, and finance sectors' portfolio returns, 96% for the high-tech sector, 92% for the manufacturing, and 81% for the health sector's predicted and experimental (i.e. observed) values of portfolio returns. The findings show that the prediction accuracy gain is rather notable when compared to the conventional OLS optimization method used in the original work of Fama-French and the traditional approach of the literature globally inspired in them. Very similar findings may be drawn if classic predicted accuracy measures like RMSE or MAE are employed as comparison criteria. The GS-CV-SVR Fama-French five-factor model performs even better than the three-factor version, with correlation coefficients ranging from 92% to 99% across industries. This conclusion differs from a significant share of literature about this topic.

5.2.4. Future lines of research

As potential and futures lines of research, we have identified the following ones:

Price forecasting

- The effectiveness of the EMD-IGRU model can be improved by further research. After breaking down the initial data, for instance, alternative ML algorithms may be adaptively chosen for the various IMFs. This procedure may enhance each IMFs prediction outcomes and forecasts for stock market closing prices.
- Related to the previous point, it could be a future line of research the use of alternative methods of signal decomposition as Fourier Transformation and DWT.
- Although in our study the technical indicators have proved to be useless, the effect of different or alternative sets of technical indicators can be explored to find the best set for IMF forecast.
- We have tested our price forecasting models, initially to Spain and US markets, and finally to the American one. It is obvious that one way of future research drives to the application to alternative world markets, in order to check whether the results obtained can be generalized and/or the efficiency level of the market is a crucial factor for the performance of the model.
- As benchmark traditional models, we have used a plain-vanilla univariate model as ARIMA(1,1,1). Alternative and more advanced models can be used as benchmark;

for example, ARCH, Generalized Autorregressive Conditional Heteroskedasticity (GARCH) and GARCH-M models.

Pairs trading

- Since our aim was to suggest some contribution in the field of UL, the trading system/strategy implemented in our study was extremely simple, although it proved to be efficient enough to improve alternative traditional approaches. Bearing in mind that our proposals were driven to the use of ML algorithms to improve the portfolio risk-return binomial, it seems a logical improvement to use SL algorithms to forecast future returns of the stock pairs selected by k-means and DBSCAN methods from UL classification and clustering family. RNN, LSTM or GRUs can be considered as potential techniques to be used in enhancing the trading procedure.

- Although in our study the firm characteristics have proved to be useless, the effect of different or alternative sets of firm characteristics can be explored to find the best set for k-means classification.

- Obviously, alternative classification techniques can be used in order to find better approaches and more brilliant performance. We have used in our study two techniques of partitioning and density-based clustering, respectively. It could be used an additional technique of hierarchical clustering. For instance, agglomerative clustering.

- In a similar way to previous exercise, we have tested our pairs trading model in the most potentially efficient stock market of all, the US market. It would be interesting to test whether the results achieved can be generalized to other markets.

- Transaction costs have not been considered in the return calculations of our trading strategy. Although we believe they are not crucial in this case, the introduction of this kind of costs in the calculations could be an interesting fine-tuning step.

- Co-integration analysis and some other time series-related statistic methods related to mean-reversion processes could be used as supplementary criteria to choose stock pairs, in order to enhance the results of the clustering ML techniques used in this thesis.

Factor Investing

- Alternative ML supervised learning algorithms could be used in order to achieve better prediction accuracy results. For instance, NN, LASSO regression, DT regression, KNN model, and Ridge Regression.

- In a similar way to previous exercises, we have tested our GS-CV-SRV model in the most potentially efficient stock market of all, the US market. It would be interesting to test whether the results achieved can be generalized to other markets or, in other words, apply this technique to cross-sectional data across countries in order to investigate if our new method is country-specific.

- The estimation of Fama-French models has been made only in its time-series dimension. It could be interesting for further research to apply this proposal to cross-sectional data, as Fama-French original procedures ([Fama and Macbeth, 1973]).

Others

- Due to time and available resources constraints, this thesis has not made any empirical contribution in one of the most important areas of interest in asset management, the portfolio management. It is a personal commitment of the author to address this discipline through some ML contribution in future work.

- In a closely related vein, this thesis has also refrained from undertaking any empirical contribution utilizing algorithms from the reinforcement learning paradigm. Notably, the area of portfolio management is, maybe, where the contributions of the third ML paradigm may be particularly pertinent. This aligns with the commitment of the author to engage in empirical contributions through the utilization of RL tools.

- As we have mentioned in section 5.1.3, there are relevant possibilities to enhance the field of application of ML techniques to asset management. All of them could be interesting ways to improve futures research about this topic: standard datasets, reproducibility issues, multimodal data, and heterogeneous architectures.

BIBLIOGRAPHY

[Abad et al., 2014] Abad, P., Benito, S., and López, C. (2014). A comprehensive review of value at risk methodologies. *The Spanish Review of Financial Economics*, 12:15–32.

[Aboussalah and Lee, 2020] Aboussalah, A. M. and Lee, C. G. (2020). Continuous control with stacked deep dynamic recurrent reinforcement learning for portfolio optimization. *Expert Systems with Applications*, 140.

[Akiyoshi, 2020] Akiyoshi, D. (2020). Evaluating machine learning algorithms for stock market prediction. *SSRN papers*.

[Almahdi and Yang, 2017] Almahdi, S. and Yang, S. Y. (2017). An adaptive portfolio trading system: A risk-return portfolio optimization using recurrent reinforcement learning with expected maximum drawdown. *Expert Systems with Applications*, 87:267–279.

[Arnott et al., 2019] Arnott, R., Harvey, C. R., and Markowitz, H. (2019). A backtesting protocol in the era of machine learning. *The Journal of Financial Data Science*, 1:64–74.

[Asness et al., 2014] Asness, C., Frazzini, A., Israel, R., and Moskowitz, T. (2014). Fact, fiction, and momentum investing. *The Journal of Portfolio Management*, 40:75–92.

[Avramov et al., 2020] Avramov, D., Cheng, S., and Metzker, L. (2020). Machine learning versus economic restrictions: Evidence from stock return predictability. *Mutual Funds*.

[Aw et al., 2019] Aw, E. N., Jiang, J., and Jiang, J. Q. (2019). Rise of the machines: Factor investing with artificial neural networks and the cross–section of expected stock returns. *The Journal of Investing*, 29:6–17.

[Ban et al., 2018] Ban, G. Y., Karoui, N. E., and Lim, A. E. (2018). Machine learning and portfolio optimization. *Management Science*, 64:1136–1154.

[Banz, 1981] Banz, R. W. (1981). The relationship between return and market value of common stocks. *Journal of Financial Economics*.

[Bao et al., 2017] Bao, W., Yue, J., and Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one*, 12(7):e0180944.

[Basu, 1977] Basu, S. (1977). Investment performance of common stocks in relation to their price-earnings ratios: A test of the efficient market hypothesis. *Journal of Finance*, 32:663–82.

[Basu, 1983] Basu, S. (1983). The relationship between earnings' yield, market value and return for nyse common stocks: further evidence. *Journal of Financial Economics*, 12:129–156.

[Belloni et al., 2014] Belloni, A., Chernozhukov, V., and Hansen, C. (2014). Inference on treatment effects after selection among high-dimensional controls. *The Review of Economic Studies*, 81:608–650.

[Black et al., 1972] Black, F., Jensen, M. C., Business, H., Edu, S. M., and Scholes, M. (1972). The capital asset pricing model: Some empirical tests. *Studies in the Theory of Capital Markets*.

[Breeden, 1979] Breeden, D. T. (1979). An intertemporal asset pricing model with stochastic consumption and investment opportunities. *Journal of Financial Economics*, 7:265–296.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

[Burguess, 2003] Burguess, A. N. (2003). Using cointegration to hedge and trade international equities. *Applied Quantitative Methods for Trading and Investment*, pages 41–70.

[Bustos and Pomares-Quimbaya, 2020] Bustos, O. and Pomares-Quimbaya, A. (2020). Stock market movement forecast: A systematic review. *Expert Systems with Applications*, 156.

[Cakici and Zaremba, 2021] Cakici, N. and Zaremba, A. (2021). Size, value, profitability, and investment effects in international stock returns: Are they really there? *Journal of Investing*.

[Carhart, 1997] Carhart, M. M. (1997). On persistence in mutual fund performance. *The Journal of Finance*, 52:57–82.

[Cavalcante et al., 2016] Cavalcante, R. C., Brasileiro, R. C., Souza, V. L., Nobrega, J. P., and Oliveira, A. L. (2016). Computational intelligence and financial markets: A survey and future directions. *Expert Systems with Applications*, 55:194–211.

[Cerniglia and Fabozzi, 2020] Cerniglia, J. A. and Fabozzi, F. J. (2020). Selecting computational models for asset management: Financial econometrics versus machine learning— is there a conflict? *The Journal of Portfolio Management*, 47.

[Cervelló-Royo and Guijarro, 2020] Cervelló-Royo, R. and Guijarro, F. (2020). Forecasting stock market trend: a comparison of machine learning algorithms. *Finance, Markets and Valuation*, 6:37–49.

[Chen, 2019] Chen, A. Y. (2019). The limits of p-hacking: A thought experiment. SSRN Working paper 3272572.

[Chen et al., 2020] Chen, L., Pelger, M., and Zhu, J. (2020). Deep learning in asset pricing. Working paper.

[Cho et al., 2014] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

[Chordia et al., 2019] Chordia, T., Goyal, A., and Shanken, J. (2019). Cross-sectional asset pricing with individual stocks: Betas versus characteristics. *SSRN Working Paper*.

[Cochrane, 2000] Cochrane, J. H. (2000). *Asset Pricing*. Princeton University Press.

[Cochrane, 2011] Cochrane, J. H. (2011). Discount rates. Working paper.

[Cooper and Maio, 2019] Cooper, I. and Maio, P. F. (2019). New evidence on conditional factor models. *Journal of Financial and Quantitative Analysis*, 54:1975–2016.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

[Cox and Ross, 1976] Cox, J. C. and Ross, S. A. (1976). The valuation of oprions for alternative stochastic processes. *Journal of Financial Economics*, 3:145–166.

[Creswell et al., 2018] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., and Bharath, A. A. (2018). Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65.

[Daniel and Titman, 1997] Daniel, K. and Titman, S. (1997). Evidence on the characteristics of cross sectional variation in stock returns. *Journal of Finance*, 52:1–33.

[David McLean and Pontiff, 2016] David McLean, R. and Pontiff, J. (2016). Does academic research destroy stock return predictability? *Journal of Finance*, 71:5–32.

[DeMiguel et al., 2009] DeMiguel, V., Garlappi, L., and Uppal, R. (2009). Optimal versus naive diversification: How inefficient is the 1/n portfolio strategy? *The Review of Financial Studies*, 22:1915–1953.

[Devendra, 2008] Devendra, K. (2008). *Soft Computing: Techniques and Its Applications in Electrical Engineering*. Springer: Berlin,Germany.

[Diallo et al., 2019] Diallo, B., Bagudu, A., and Zhang, Q. (2019). A machine learning approach to the fama-french three- and five-factor models. *SSRN papers*.

[Ding et al., 2015] Ding, X., Zhang, Y., Liu, T., and Duan, J. (2015). Deep learning for event-driven stock prediction. In *Twenty-fourth international joint conference on artificial intelligence*.

[Durairaj and Mohan, 2019] Durairaj, M. and Mohan, B. K. (2019). A review of two decades of deep learning hybrids for financial time series prediction. *International Journal on Emerging Technologies*, 10:324–331.

[Elith et al., 2008] Elith, J., Leathwick, J. R., and Hastie, T. (2008). A working guide to boosted regression trees. *Journal of animal ecology*, 77(4):802–813.

[Elliott et al., 2005] Elliott, R., Hoek, J. V. D., and Malcolm, W. (2005). Pairs trading. *Quantitative Finance*, 5:271–276.

[Emerson et al., 2019] Emerson, S., Kennedy, R., O'Shea, L., and O'Brien, J. (2019). Trends and applications of machine learning in quantitative finance. ICEFR 2019.

[Engle and Granger, 1987] Engle, R. and Granger, C. (1987). Co-integration and error correction: representation, estimation and testing. *Econometrica*, pages 251–276.

[Ester et al., 1996] Ester, M., Kriegel, H., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, AAAI Press*.

[Fama and French, 1992] Fama, E. and French, K. (1992). The cross section of expected stock returns. *The Journal of Finance*, XLVII:427–465.

[Fama and French, 1993] Fama, E. and French, K. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33:3–56.

[Fama and French, 1988] Fama, E. F. and French, K. (1988). Dividend yields and expected stock returns. *Journal of Financial Economics*, 22:3–25.

[Fama and French, 1996] Fama, E. F. and French, K. R. (1996). Multifactor explanations of asset pricing anomalies. *The Journal of Finance*, 51:55–84.

[Fama and French, 2015] Fama, E. F. and French, K. R. (2015). A five-factor asset pricing model. *Journal of Financial Economics*.

[Fama and French, 2018] Fama, E. F. and French, K. R. (2018). Choosing factors. *Journal of Financial Economics*, 128:234–52.

[Fama and Macbeth, 1973] Fama, E. F. and Macbeth, J. D. (1973). Risk, return, and equilibrium: Empirical tests. *Journal of Political Economy*, 81:607–636.

[Feng et al., 2020] Feng, G., Giglio, S., and Xiu, D. (2020). Taming the factor zoo: A test of new factors. *The Journal of Finance*, 75:1327–70.

[Feng and He, 2019] Feng, G. and He, J. (2019). Factor investing: Hierarchical ensemble learning. *SSRN Electronic Journal*.

[Feng et al., 2018a] Feng, G., Polson, N. G., and Xu, J. (2018a). Deep factor alpha. Working paper.

[Feng et al., 2018b] Feng, G., Polson, N. G., and Xu, J. (2018b). Deep learning in characteristics-sorted factor models. *Portfolio Theory eJournal*.

[Feng et al., 2019] Feng, G., Polson, N. G., and Xu, J. (2019). Deep learning in asset pricing. SSRN: https://ssrn.com/abstract=3350138.

[Freyberger et al., 2020] Freyberger, J., Neuhierl, A., and Weber, M. (2020). Dissecting characteristics nonparametrically. *Review of Financial Studies*, 33:2326–2377.

[Gatev et al., 2006] Gatev, W., Goetzmann, W. N., and Rouwenhorst, K. (2006). Pairs trading: Performance of a relative value arbitrage rule. *Review of Financial Studies*, 19:797–827.

[Gerakos et al., 2013] Gerakos, J., Linnainmaa, J. T., and Daniel, K. (2013). Confounded factors. *Chicago Booth Paper*.

[Giglio and Xiu, 2019] Giglio, S. and Xiu, D. (2019). Asset pricing with omitted factors. *SSRN Working Paper*.

[Goetzmann and Kumar, 2008] Goetzmann, W. N. and Kumar, A. (2008). Equity portfolio diversification. *Review of Finance*, 12:433–463.

[Gogas et al., 2018] Gogas, P., Papadimitriou, T., and Karagkiozis, D. (2018). The fama 3 and fama 5 factor models under a machine learning framework. Working paper.

[Graham and Dodd, 1934] Graham, B. and Dodd, D. L. (1934). *Security Analysis*. Whittlesey House, McGraw-Hill Book.

[Green et al., 2016] Green, J., Hand, J. R. M., and Zhang, F. (2016). The characteristics that provide independent information about average u.s. monthly stock returns. *Capital Markets: Market Efficiency eJournal*.

[Griffin and Lemmon, 2002] Griffin, J. and Lemmon, M. (2002). Book-to-market equity, distress risk, and stock returns. *Journal of Finance*, 57:2317–2336.

[Gu et al., 2020a] Gu, S., Kelly, B., and Xiu, D. (2020a). Autoencoder asset pricing models. *Journal of Econometrics*.

[Gu et al., 2020b] Gu, S., Kelly, B., and Xiu, D. (2020b). Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33:2223–2273.

[Han et al., 2021] Han, C., He, Z., and Toh, A. (2021). Pairs trading via unsupervised learning. *SSRN papers*.

[Hansen and Richard, 1987] Hansen, L. P. and Richard, S. F. (1987). The role of conditioning information in deducing testable restrictions implied by dynamic asset pricing models. *Econometrica*, 55:587–613.

[Harvey and Liu, 2019] Harvey, C. and Liu, Y. (2019). Lucky factors. *SSRN Working Paper*.

[Harvey, 2017] Harvey, C. R. (2017). Presidential address: The scientific outlook in financial economics. *Journal of Finance*, 72:1399–1440.

[He et al., 2016] He, Z., Kelly, B. T., and Manela, A. (2016). Intermediary asset pricing: New evidence from many asset classes. *Journal of Financial Economics*, 126:1–35.

[Heaton et al., 2017] Heaton, J. B., Polson, N. G., and Witte, J. H. (2017). Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*, 33:3–12.

[Henrique et al., 2019] Henrique, B. M., Sobreiro, V. A., and Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications*, 124:226–251.

[Herlemont, 2004] Herlemont, D. (2004). Pairs trading, convergence trading, cointegration. *YATS Finances and Technologies*.

[Hester et al., 2018] Hester, T., Vecerik, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., Horgan, D., Quan, J., Sendonaris, A., Osband, I., et al. (2018). Deep q-learning from demonstrations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Lstm can solve hard long time lag problems. *Advances in neural information processing systems*, 9:473–479.

[Hong and Stein, 1999] Hong, H. and Stein, J. C. (1999). A unified theory of underreaction, momentum trading, and overreaction in asset markets. *The Journal of Finance*, 54:2143—2184.

[Hou et al., 2020] Hou, K., Mo, H., Xue, C., and Zhang, L. (2020). An augmented q-factor model with expected growth. *Review of Finance*.

[Hou et al., 2017] Hou, K., Xue, C., and Zhang, L. (2017). Replicating anomalies. Working paper.

[Hou and Zhang, 2015] Hou, K. C. X. and Zhang, L. (2015). Digesting anomalies: An investment approach. *Review of Financial Studies*, 28:650–705.

[Huang et al., 2020] Huang, J., Chai, J., and Cho, S. (2020). Deep learning in finance and banking: A literature review and classification. *Frontiers of Business Research in China*, 14.

[Huang et al., 1998] Huang, N. E., Shen, Z., Long, S. R., C.Wu, M., Shih, H. H., Zheng, Q., Yen, N.-C., Tung, C. C., and Liu, H. H. (1998). The empirical mode decomposition and the hilbert spectrum for non-linear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, pages 903–95.

[Huck, 2019] Huck, N. (2019). Large data sets and machine learning: Applications to statistical arbitrage. *European Journal of Operational Research*, 278:330–342.

[Huotari et al., 2020] Huotari, T., Savolainen, J., and Collan, M. (2020). Deep reinforcement learning agent for s&p 500 stock selection. *Axioms*, 9:1–15.

[Ito and Nakano, 2003] Ito, K. and Nakano, R. (2003). Optimizing support vector regression hyperparameters based on cross-validation. *Proceedings of the International Joint Conference on Neural Networks*, 3:2077 – 2082 vol.3.

[Jacobs and Mülle, 2020] Jacobs, H. and Mülle, S. (2020). Anomalies across the globe: Once public, no longer existent? *Journal of Financial Economics*, 135:213–30.

[Jacobs and Weber, 2015] Jacobs, H. and Weber, M. (2015). On the determinants of pairs trading profitability. *Journal of Financial Markets*, 23:75–97.

[Jain and Jain, 2019] Jain, P. and Jain, S. (2019). Can machine learning-based portfolios outperform traditional risk-based portfolios? the need to account for covariance misspecification. *Risks*, 7.

[Jegadeesh, 1990] Jegadeesh, N. (1990). Evidence of predictable behavior of security returns. *The Journal of Finance*, 45:881—898.

[Jegadeesh and Titman, 1993] Jegadeesh, N. and Titman, S. (1993). Returns to buying winners and selling losers: Implications for stock market efficiency. *The Journal of Finance*, 48:65–91.

[Jiang, 2021] Jiang, W. (2021). Applications of deep learning in stock market prediction: recent progress. *Expert Systems with applications*, 184:115537.

[Jiang et al., 2017] Jiang, Z., Xu, D., and Liang, J. (2017). A deep reinforcement learning framework for the financial portfolio management problem. Arxiv.

[Kamley et al., 2016] Kamley, S., Jaloree, S., and Thakur, R. S. (2016). Performance forecasting of share market using machine learning techniques: A review. *International Journal of Electrical and Computer Engineering*, 6:3196–3204.

[Kelly et al., 2018] Kelly, B. T., Pruitt, S., and Su, Y. (2018). Characteristics are covariances: A unified model of risk and return. Working paper.

[Khan et al., 2020a] Khan, W., Ghazanfar, M. A., Azam, M., Karami, A., Alyoubi, K. H., and Alfakeeh, A. S. (2020a). Stock market prediction using machine learning classifiers and social media, news. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–24.

[Khan et al., 2020b] Khan, W., Malik, U., Ghazanfar, M. A., Azam, M. A., Alyoubi, K. H., and Alfakeeh, A. S. (2020b). Predicting stock market trends using machine learning algorithms via public sentiment and political situation analysis. *Soft Computing*, 24:11019–11043.

[Kim and Kim, 2019] Kim, T. and Kim, H. (2019). Optimizing the pairs-trading strategy using deep reinforcement learning with trading and stop-loss boundaries. *Complexity*, 2019:1–20.

[Konstantinov et al., 2020] Konstantinov, G. S., Chorus, A., and Rebmann, J. (2020). A network and machine learning approach to factor, asset, and blended allocation. *The Journal of Portfolio Management*.

[Kozak et al., 2018] Kozak, S., Nagel, S., and Santosh, S. (2018). Interpreting factor models. *Journal of Finance*, 73:1183–223.

[Kozak et al., 2019] Kozak, S., Nagel, S., and Santosh, S. (2019). Shrinking the cross-section. *Journal of Financial Economics*, 135.

[Krauss, 2017] Krauss, C. (2017). Statistical arbitrage pairs trading strategies: Review and outlook. *Journal of Economic Surveys*, 31:513–545.

[Krauss et al., 2017] Krauss, C., Do, X. A., and Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research*, 259(2):689–702. Online Oct. 2016 Indicateurs 2016.

[Krkoska and Schenk-Hoppé, 2019] Krkoska, E. and Schenk-Hoppé, K. R. (2019). Herding in smart-beta investment products. *Journal of Risk and Financial Management*, 12:47.

[Kumar et al., 2018] Kumar, I., Dogra, K., Utreja, C., and Yadav, P. (2018). A comparative study of supervised machine learning algorithms for stock market trend prediction. Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2018.

[Lai and Xing, 2008] Lai, T. and Xing, H. (2008). *Statistical Models and Methods for Financial Markets*. Springer, New York.

[Lakonishok et al., 1994] Lakonishok, J., Shleifer, A., and Vishny, R. (1994). Contrarian investment, extrapolation, and risk. *Journal of Finance*, 49:1541–78.

[LeCun et al., 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

[LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[Lee et al., 2010] Lee, C.-C., Lee, J.-D., and Lee, C.-C. (2010). Stock prices and the efficient market hypothesis: Evidence from a panel stationary test with structural breaks. *Japan and the World Economy*, 22:49–58.

[Lee and Kang, 2020] Lee, J. and Kang, J. (2020). Effectively training neural networks for stock index prediction: Predicting the s&p 500 index without using its index data. *PLoS ONE*, 15.

[Lee et al., 2019] Lee, T. K., Cho, J. H., Kwon, D. S., and Sohn, S. Y. (2019). Global stock market investment strategies based on financial network indicators using machine learning techniques. *Expert Systems with Applications*, 117:228–242.

[Levenberg, 1944] Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168.

[Li and Hoi, 2014] Li, B. and Hoi, S. C. (2014). Online portfolio selection: A survey. *ACM Computing Surveys*, 46.

[Li et al., 2015] Li, X., Li, L., Gao, J., He, X., Chen, J., Deng, L., and He, J. (2015). Recurrent reinforcement learning: a hybrid approach. arXiv:1509.03044.

[Liew and Vassalou, 2000] Liew, J. and Vassalou, M. (2000). Can book-to-market, size and momentum be risk factors that predict economic growth? *Journal of Financial Economics*, 57:221–245.

[Lintner, 1965] Lintner, J. (1965). The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets. *The Review of Economics and Statistics*.

[Lloyd, 1982] Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.

[Lo and MacKinlay, 1988] Lo, A. W. and MacKinlay, A. C. (1988). Stock market prices do not follow random walks: Evidence from a simple specification test. *Review of Financial Studies*, 1:41—66.

[Lu et al., 2019] Lu, Z., Long, W., Zhang, J., and Tian, Y. (2019). Factor integration based on neural networks for factor investing. In *Computational Science – ICCS 2019, 19th International Conference*, volume 11538 LNCS, pages 286–292. Springer Verlag.

[López de Prado, 2016] López de Prado, M. (2016). Building diversified portfolios that outperform out of sample. *The Journal of Portfolio Management*, 42:59–69.

[Ma et al., 2020] Ma, Y., Han, R., and Wang, W. (2020). Prediction-based portfolio optimization models using deep neural networks. *IEEE Access*, 8:115393–115405.

[Ma et al., 2021] Ma, Y., Han, R., and Wang, W. (2021). Portfolio optimization with return prediction using deep learning and machine learning. *Expert Systems with Applications*, 165.

[MacQueen, 1967] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Oakland, CA, USA*.

[Malkiel and Fama, 1970] Malkiel, B. and Fama, E. (1970). Efficient capital markets: A review of theory and empirical work. *Journal of Finance*, pages 383–417.

[Markowitz, 1952] Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, 7:77–91.

[Merton, 1973] Merton, R. C. (1973). An intertemporal capital asset pricing model. *Econometrica*.

[Messmer, 2017] Messmer, M. (2017). Deep learning and the cross-section of expected returns. *SSRN Electronic Journal*.

[Minh et al., 2018] Minh, D. L., Sadeghi-Niaraki, A., Huy, H. D., Min, K., and Moon, H. (2018). Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network. *Ieee Access*, 6:55392–55404.

[Mirete-Ferrer et al., 2022] Mirete-Ferrer, P. M., Garcia-Garcia, A., Baixauli-Soler, J. S., and Prats, M. A. (2022). A Review on Machine Learning for Asset Management. *Risks*, 10(4):1–46.

[Moritz and Zimmermann, 2016] Moritz, B. and Zimmermann, T. (2016). Tree-based conditional portfolio sorts: The relation between past and future stock returns. Working paper.

[Nabipour et al., 2020] Nabipour, M., Nayyeri, P., Jabani, H., S., S., and Mosavi, A. (2020). Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis. *IEEE Access*, 8:150199–150212.

[Nava et al., 2018] Nava, N., Matteo, T. D., and Aste, T. (2018). Financial time series forecasting using empirical mode decomposition and support vector regression. *Risks*, 6.

[Niennattrakul and Ratanamahatana, 2007] Niennattrakul, V. and Ratanamahatana, C. (2007). On clustering multimedia time series data using k-means and dynamic time warping. *Multimedia and Ubiquitous Engineering International Conference*, pages 733–738.

[Nikou et al., 2019] Nikou, M., Mansourfar, G., and Bagherzadeh, J. (2019). Stock price prediction using deep learning algorithm and its comparison with machine learning algorithms. *Intelligent Systems in Accounting, Finance and Management*, 26:164–174.

[Niño et al., 2018] Niño, J., Hernández, G., Arévalo, A., and León, D. (2018). Cnn with limit order book data for stock price prediction. In *Proceedings of the Future Technologies Conference (FTC) 2018*. Institute of Electrical and Electronics Engineers (IEEE).

[Novy-Marx, 2013] Novy-Marx, R. (2013). The other side of value: The gross profitability premium. *Journal of Financial Economics*, 108:1–28.

[Novy-Marx and Velikov, 2016] Novy-Marx, R. and Velikov, M. (2016). A taxonomy of anomalies and their trading costs. *Review of Financial Studies*, 29:104–147.

[Nti et al., 2019] Nti, I. K., Adekoya, A. F., and Weyori, B. (2019). A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review*, 53:3007–3057.

[Ozbayoglu et al., 2020] Ozbayoglu, A. M., Gudelek, M. U., and Sezer, O. B. (2020). Deep learning for financial applications: A survey. *Applied Soft Computing Journal*, 93.

[Paiva et al., 2019] Paiva, F. D., Cardoso, R. T. N., Hanaoka, G. P., and Duarte, W. M. (2019). Decision-making for financial trading: A fusion approach of machine learning and portfolio selection. *Expert Systems with Applications*, 115:635–655.

[Park et al., 2020] Park, H., Sim, M. K., and Choi, D. G. (2020). An intelligent financial portfolio trading strategy using deep q-learning. *Expert Systems with Applications*, 158.

[Pastor and Stambaugh, 2003] Pastor, L. and Stambaugh, R. F. (2003). Liquidity risk and expected stock returns. *Journal of Political Economy*, 111:642—685.

[Pearson, 1901] Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572.

[Peñaranda and Sentana, 2010] Peñaranda, F. and Sentana, E. (2010). A unifying approach to the empirical evaluation of asset pricing models. CEMFI Working paper.

[Pitera and Schmidt, 2018] Pitera, M. and Schmidt, T. (2018). Unbiased estimation of risk. *Journal of Banking and Finance*, 91:133–145.

[Poterba and Summers, 1988] Poterba, J. M. and Summers, L. (1988). Mean reversion in stock prices: Evidence and implications. *Journal of Financial Economics*, 22:27–59.

[Pukthuanthong and Roll, 2019] Pukthuanthong, K. and Roll, R. (2019). A protocol for factor identification. *Review of Financial Studies*, 32:1573–1607.

[Raffinot, 2017] Raffinot, T. (2017). Hierarchical clustering-based asset allocation. *Journal of Portfolio Management*, 44:89–99.

[Rasekhschaffe and Jones, 2019] Rasekhschaffe, K. C. and Jones, R. C. (2019). Machine learning for stock selection. *Financial Analysts Journal*, 75:70–88.

[Ratanamahatana and Keogh, 2004] Ratanamahatana, C. and Keogh, E. (2004). Making time-series classification more accurate using learned constraints. *2007 International Conference on Multimedia and Ubiquitous Engineering*.

[Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

[Ross, 1976] Ross, S. A. (1976). The arbitrage theory of capital asset pricing. *Journal of Economic Theory*, 13:341–360.

[Ross, 1978] Ross, S. A. (1978). A simple approach to the valuation of risky streams. *The Journal of Business*, 51:453–475.

[Sarmento and Horta, 2020] Sarmento, S. and Horta, N. (2020). Enhancing a pairs trading strategy with the application of machine learning. *Expert Systems with Applications*, 158:113490.

[Schubert et al., 2017] Schubert, E., Sander, J., Ester, M., Kriegel, H., and Xu, X. (2017). Dbscan revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42:10823.

[Sezer et al., 2020] Sezer, O. B., Gudelek, M. U., and Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005-2019. *Applied Soft Computing*, 90:106181.

[Sezer and Ozbayoglu, 2018] Sezer, O. B. and Ozbayoglu, A. M. (2018). Algorithmic financial trading with deep convolutional neural networks: time series to image conversion approach. *Applied Soft Computing*, 70:525–538.

[Sezer et al., 2017] Sezer, O. B., Ozbayoglu, M., and Dogdu, E. (2017). A deep neural-network based stock trading system based on evolutionary optimized technical analysis parameters. *Procedia Computer Science*, 114:473–480.

[Sharpe, 1964] Sharpe, W. F. (1964). American finance association capital asset prices: A theory of market equilibrium under conditions of risk. *The Journal of Finance*, 19:425–442.

[Shen and Shafiq, 2020] Shen, J. and Shafiq, M. O. (2020). Short-term stock market price trend prediction using a comprehensive deep learning system. *Journal of Big Data*, 7.

[Shi and Zhuang, 2019] Shi, C. and Zhuang, X. (2019). A study concerning soft computing approaches for stock price forecasting. *Axioms*, 8.

[Simonian et al., 2019] Simonian, J., Wu, C., Itano, D., and Narayanam, V. (2019). A machine learning approach to risk factors: A case study using the fama–french–carhart model. *The Journal of Financial Data Science*, 1.

[Sirignano and Cont, 2018] Sirignano, J. A. and Cont, R. (2018). Universal features of price formation in financial markets: Perspectives from deep learning. *General eJournal*.

[Snow, 2020] Snow, D. (2020). Machine learning in asset management-part 1: Portfolio construction-trading strategies. *The Journal of Financial Data Science*.

[Song et al., 2017] Song, Q., Liu, A., and Yang, S. Y. (2017). Stock portfolio selection using learning-to-rank algorithms with news sentiment. *Neurocomputing*, 264:20–28.

[Spierdijk and Bikker, 2012] Spierdijk, L. and Bikker, J. (2012). Mean reversion in stock prices: Implications for long-term investors. *Working Papers, De Nederlandsche Bank*.

[Srijiranon et al., 2022] Srijiranon, K., Lertratanakham, Y., and Tanantong, T. A. (2022). Hybrid framework using pca, emd and lstm methods for stock market. *Applied Sciences*, 12:10823.

[Stattman, 1980] Stattman, D. (1980). Book value and stock returns. *The Chicago MBA: A Journal of Selected Papers*, 4:25–45.

[Sugitomo and Minami, 2018] Sugitomo, S. and Minami, S. (2018). Fundamental factor models using machine learning. *Journal of Mathematical Finance*, 08:111–118.

[Sun, 2020] Sun, C. (2020). Dissecting the factor zoo: A correlation-robust machine learning approach. *Market Efficiency eJournal*.

[Sutton and Barto, 2018] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

[Ta et al., 2020] Ta, V. D., Liu, C. M., and Tadesse, D. A. (2020). Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading. *Applied Sciences (Switzerland)*, 10.

[Tibshirani, 1996] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

[Timmermann and Granger, 2004] Timmermann, A. and Granger, C. W. (2004). Efficient market hypothesis and forecasting. *International Journal of forecasting*, 20(1):15–27.

[Tkáč and Verner, 2016] Tkáč, M. and Verner, R. (2016). Artificial neural networks in business: Two decades of research. *Applied Soft Computing Journal*, 38:788–804.

[Tobek and Hronec, 2020] Tobek, O. and Hronec, M. (2020). Does it pay to follow anomalies research? machine learning approach with international evidence. *Journal of Financial Markets*.

[Tristan and Chin Sin, 2021] Tristan, L. and Chin Sin, O. (2021). Portfolio management: A financial application of unsupervised shape-based clustering-driven machine learning method. *International Journal of Computing and Digital Systems*, 10.

[Troiano et al., 2018] Troiano, L., Villa, E. M., and Loia, V. (2018). Replicating a trading strategy by means of lstm for financial industry applications. *IEEE Transactions on Industrial Informatics*, 14:3226–3234.

[Tsantekidis et al., 2017a] Tsantekidis, A., Passalis, N., Tefas, A., and Kanniainen, J. (2017a). Using deep learning and technical analysis to predict stock prices. *Expert Systems with Applications*, 73:1–9.

[Tsantekidis et al., 2017b] Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., and Iosifidis, A. (2017b). Forecasting stock prices from thelimit order book using convolutional neural networks. In *2017 IEEE 19thConference on Business Informatics (CBI)*. Institute of Electrical and Electronics Engineers (IEEE).

[Tsantekidis et al., 2017c] Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., and Iosifidis, A. (2017c). Using deep learning to detect price change indications in financial markets. In *2017 25th European Signal Processing Conference (EUSIPCO)*. Institute of Electrical and Electronics Engineers (IEEE).

[Van Dijk, 2011] Van Dijk, M. A. (2011). Is size dead? a review of the size effect in equity returns. *Journal of Banking and Finance*.

[Vapnik et al., 1996] Vapnik, V. N., Golowich, S. E., and Smola, A. (1996). Support vector method for function approximation, regression estimation and signal processing. In *NIPS*.

[Vassalou et al., 2004] Vassalou, M., Xing, Y., French, K., Hirshleifer, D., Jagannathan, R., Lando, D., Nielsen, L. T., Shanken, J., Stein, J., and Ritter, J. (2004). Default risk in equity returns. *Journal of Finance*, 59:831–868.

[Vidyamurthy, 2004] Vidyamurthy, G. (2004). Pairs trading. *Quantitative Methods and Analysis*.

[Vo et al., 2019] Vo, N. N., He, X., Liu, S., and Xu, G. (2019). Deep learning for decision making and the optimization of socially responsible investments and portfolio. *Decision Support Systems*, 124.

[Wang et al., 2020] Wang, W., Li, W., Zhang, N., and Liu, K. (2020). Portfolio formation with preselection using deep learning from long-term financial data. *Expert Systems with Applications*, 143.

[Watkins and Dayan, 1992] Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine learning*, 8(3):279–292.

[Weigand, 2019] Weigand, A. (2019). Machine learning in empirical asset pricing. *Financial Markets and Portfolio Management*, 33:93–104.

[Xing et al., 2018] Xing, F. Z., Cambria, E., and Welsch, R. E. (2018). Natural language based financial forecasting: a survey. *Artificial Intelligence Review*, 50:49–73.

[Xue et al., 2018] Xue, J., Liu, Q., Li, M., Liu, X., Ye, Y., Wang, S., and Yin, J. (2018). Incremental multiple kernel extreme learning machine and its application in robo-advisors. *Soft Computing*, 22.

[Yun et al., 2020] Yun, H., Lee, M., Kang, Y. S., and Seok, J. (2020). Portfolio management via two-stage deep learning with a joint cost. *Expert Systems with Applications*, 143.

[Zhong and Enke, 2019] Zhong, X. and Enke, D. (2019). Predicting the daily return direction of the stock market using hybrid machine learning algorithms. *Financial Innovation*, 5.