UNIVERSITY OF DEUSTO

ENHANCING FINANCIAL TIME SERIES PREDICTION AND ASSET ALLOCATION WITH MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

Doctoral thesis by
DANIEL GONZALEZ CORTES
within the program
ENGINEERING FOR THE INFORMATION SOCIETY
AND SUSTAINABLE DEVELOPMENT
Supervised by
ENRIQUE ONIEVA CARACUEL
IKER PASTOR LOPEZ

UNIVERSITY OF DEUSTO

ENHANCING FINANCIAL TIME SERIES PREDICTION AND ASSET ALLOCATION WITH MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

Doctoral thesis by
DANIEL GONZALEZ CORTES
within the program
ENGINEERING FOR THE INFORMATION SOCIETY
AND SUSTAINABLE DEVELOPMENT
Supervised by
ENRIQUE ONIEVA CARACUEL
IKER PASTOR LOPEZ

PhD Candidate

DANIEL ALEJANDRO GONZALEZ CORTES
Digitally signed by DANIEL ALEJANDRO GONZALEZ CORTES
Date: 2024.07.11 13:11:30 +02'00'

Advisor

ONIEVA CARACUEL ENRIQUE - 26973609Z
Firmado digitalmente por ONIEVA CARACUEL ENRIQUE - 26973609Z
Nombre de reconocimiento (DN): c=ES, serialNumber=IDCES-26973609Z, givenName=ENRIQUE, sn=ONIEVA CARACUEL, cn=ONIEVA CARACUEL ENRIQUE - 26973609Z
Fecha: 2024.07.12 07:05:30 +02'00'

Advisor

PASTOR LOPEZ, IKER (FIRMA)
Firmado digitalmente por PASTOR LOPEZ, IKER (FIRMA)
Fecha: 2024.07.14 12:36:04 +02'00'

Bilbao, July 2024

Enhancing Financial Time Series Prediction and Asset Allocation with Machine Learning and Artificial Intelligence

Author: Daniel González Cortés
Advisor: Enrique Onieva Caracuel
Co-advisor: Iker Pastor Lopez

Text printed in Bilbao
First edition, July 2024

We shall not cease from exploration. And the end of all our exploring.
Will be to arrive where we started. And know the place for the first time.
T.S. Eliot

Abstract

This thesis explored the integration of machine learning algorithms, deep learning, and reinforcement learning in the analysis of financial time series for prediction and portfolio construction. The main objectives were to enhance the accuracy of predictions, optimize asset selection to maximize risk-adjusted returns, improve the interpretability of market forecasts, and optimize portfolio management strategies.

This research began with a comprehensive review of traditional asset pricing models, regression algorithms, and neural network architectures. It then explored state-of-the-art machine learning applications in finance, highlighting trends and methodologies for financial time series forecasting and clustering.

A neural network-based approach for predicting financial market time series was proposed, demonstrating significant improvements in forecasting accuracy. Additionally, an autoencoder-enhanced clustering method was introduced to facilitate dimensionality reduction in financial datasets, leading to a more efficient data analysis.

The application of explainability in predicting digital asset prices was examined, emphasizing the importance of model transparency and interpretability. Various machine learning models were evaluated, and their performance was analyzed in terms of accuracy and explainability.

The thesis also presented a novel approach to portfolio construction using explainable reinforcement learning. This method leveraged reinforcement learning algorithms to optimize portfolio allocation while ensuring the model's decisions were interpretable and understandable to stakeholders.

The findings contributed to the field of financial machine learning by providing adequate and explainable methodologies for time series forecasting, clustering, and portfolio management.

Resumen

Esta tesis exploró la integración de algoritmos de aprendizaje automático, aprendizaje profundo y aprendizaje por refuerzo en el análisis de series temporales financieras para la predicción y la construcción de carteras de inversión. Los objetivos principales fueron mejorar la precisión de las predicciones, optimizar la selección de activos para maximizar los rendimientos ajustados al riesgo, mejorar la interpretabilidad de las previsiones del mercado y optimizar las estrategias de gestión de carteras.

Esta investigación comenzó con una revisión exhaustiva de los modelos tradicionales de fijación de precios de activos, algoritmos de regresión y arquitecturas de redes neuronales. Luego se adentró en las aplicaciones de vanguardia del aprendizaje automático en finanzas, destacando tendencias y metodologías para la predicción y la clusterización de series temporales financieras.

Se propuso un enfoque basado en redes neuronales para predecir las series temporales del mercado financiero, demostrando mejoras significativas en la precisión de las previsiones. Además, se introdujo un método de clusterización mejorado con autoencoders para facilitar la reducción de dimensionalidad en conjuntos de datos financieros, lo que condujo a un análisis de datos más eficiente e informativo.

Se examinó la aplicación de la explicabilidad en la predicción de precios de activos digitales, enfatizando la importancia de la transparencia e interpretabilidad del modelo. Se evaluaron varios modelos de aprendizaje automático y se analizó su rendimiento en términos de precisión y explicabilidad.

La tesis también presentó un enfoque novedoso para la construcción de carteras utilizando aprendizaje por refuerzo explicable. Este método aprovechó los algoritmos de aprendizaje por refuerzo para optimizar la asignación de carteras, asegurando que las decisiones del modelo fueran interpretables y comprensibles para las partes interesadas.

Los hallazgos contribuyeron al campo del aprendizaje automático financiero proporcionando metodologías adecuadas y explicables para la previsión de series temporales, la clusterización y la gestión de carteras.

Acknowledgments

Completing this thesis would not have been possible without the support and collaboration of numerous individuals and institutions to whom I would like to express my deepest gratitude.

First and foremost, I would like to thank my thesis advisor, Professor Enrique Onieva, for his invaluable guidance and for always believing in me since I was a bachelor's student. He is the one who encouraged me to start this path of research in the field of Artificial Intelligence. Also, I would like to thank my co-supervisor, Professor Iker Pastor, for their insights and feedback throughout my research journey.

I am grateful to NEOMA Business School for providing the necessary resources and financial support for my research. Special thanks to Professor Jian Wu for her support throughout my research journey and for always believing in me. I would also like to thank Professor Laura Trinchera for her invaluable insights, support, and recommendations.

A sincere thanks to Professor Monomita Nandy for her irreplaceable and esteemed help and Professor Suman Lodh and Professor Omar Bencharef for their camaraderie, support, and stimulating discussions. Their assistance and friendship have made this journey more enjoyable and rewarding.

Lastly, I am profoundly grateful to God and my parents for their persistent support and encouragement. To my parents and brother, thank you for your endless love and belief in me. To Anne and her family, your love, understanding ,and constant motivation have been invaluable.

Contents

List of Figures

List of Tables

1

Introduction

In this section, the factors that motivate the progress of this thesis will be explored. Additionally, discuss the factors that motivate the thesis and the challenges that must be addressed. It emphasizes the complexities and difficuluties in forecasting time series and creating portfolios in the financial markets. It examines how Artificial Neural Networks (ANN), Machine Learning (ML), Deep Learning (DL), Explainable Artificial Intelligence (XAI), and Reinforcement Learning (RL) can be instrumental in tackling these challenges. Subsequently, the thesis outlines its contributions, describes the research approach employed to achieve these objectives, details the research efforts undertaken, and explains the research structure.

1.1 Motivation and Scope

The irruption of ML into various industries represents a paradigm shift towards data-driven decision-making, where machines autonomously improve and adapt by extracting insights from vast datasets to predict and model outcomes with increasing accuracy. The financial sector, in particular, has witnessed a substantial transformation with the rise of ML, moving from heuristic-based decision processes to complex algorithmic strategies, accomplishing tasks that were only achieved with human expertise not long ago (López de Prado 2020). This trend can potentially enhance the precision and the scale on which financial markets can be studied and redefine asset management. Its application in asset pricing and portfolio construction is particularly significant, offering sophisticated tools for analyzing large datasets, identifying patterns, and making informed investment decisions (Goodell et al. 2021).

Asset pricing involves determining the fair value of financial instruments, like stocks or bonds, based on their risk, expected return, and underlying factors. Since ML can offer unparalleled capabilities in processing and interpreting vast arrays of financial data, it has become increasingly popular with a rising number of publications and citations in the last few years (Gu et al. 2020b). Additionally, by analyzing historical data ML algorithms can analyze different trends from price action, news articles, or social media (Xu et al. 2023) with superior performance than traditional methods and with the capability to uncover intricate patterns more effectively than conventional econometric models (Bagnara 2022).

On the other hand, portfolio construction involves strategically assembling various assets to meet specific investment goals, balancing risk against expected returns. ML algorithms enable the execution of simulations and predictive modeling to evaluate potential portfolio performances across different market scenarios, facilitating the development of optimized portfolios that align with specific investment goals and risk tolerances and also offering a level of customization and effectiveness not achievable with conventional methods. Additionally, data-driven portfolio management systems have the agility to adapt to market shifts rapidly, employing advanced algorithms, such as those based on RL, to enhance investment strategies (Betancourt and W.-H. Chen 2021) continuously.

In this context, the evolving domain of asset pricing and portfolio construction is increasingly influenced by data-driven techniques that contrast sharply with classical statistical and econometrics models (Bagnara 2022; López de Prado 2020). Classical asset pricing models rely on fixed factors and linear relationships, such as the Capital Asset Pricing Model (CAPM) (Sharpe 1964) or the Arbitrage Pricing Theory (Ross 1976), often overlooking the complexity of numerous variables. Classical research in asset pricing, especially in stock return prediction, has focused on a small selection of firm characteristics, using factor models that include only a few factors such as firm size, profitability, and equity ratios (Eugene F. Fama and K. R. French 2015a). However, these traditional approaches, with their high degree of selectivity, fail to capture the non-linear and complex dynamics of financial markets, neglecting a vast array of potentially relevant factors, overlooking the potential predictive power of a more extensive collection of variables, leading to an increasing recognition of the need for more complex models (Nagel 2021).

Furthermore, while conventional models offer an essential groundwork for comprehending market fundamentals, they are inadequate when dealing with the extensive, multidimensional, unstructured, and dynamic data typical of financial markets. This limitation limits effective data analysis and obstructs crucial analytical tasks like prediction and clustering (Njah et al. 2021). In this context, it is important also to consider classical diversification principles in portfolio construction, such as those proposed in the work by Harry Markowitz's modern portfolio theory (Markowitz 1952), which have been the cornerstone of efficient portfolio construction for over seventy years.

Therefore, a substantial body of evidence has accumulated in asset pricing and portfolio construction, affirming the limitations inherent in classical models. This empirical recognition is leading towards more sophisticated, data-oriented methodologies, such as ML and Artificial Intelligence (AI), with a different set of new benchmarks for adaptability, efficiency, and performance in the financial industry, challenging traditional paradigms and reshaping the financial industry and academia.

Comparatively, techniques underpinned by advanced algorithms such ML and DL can process and analyze data in a non-linear fashion, capturing dependencies and patterns beyond the grasp of classical models (Sezer et al. 2020). These methods offer enhanced accuracy in predictive analytics and a more dynamic approach to risk assessment and strategy formulation.

Additionally, the classical modern portfolio theory model often operates under the assumption of a static market environment and a normal distribution of returns. Advanced algorithms, on the other hand, introduce a dynamic component to portfolio management; for example, they can simulate numerous potential market conditions, adapting to real-time data, which allows for constructing robust portfolios under various future states and not just based on historical averages (Betancourt and W.-H. Chen 2021).

Nonetheless, deploying ML techniques has challenges and complications. These include the lack of fairness (Feuerriegel et al. 2020), issues like the opacity of algorithmic decision-making, commonly called 'black box' behavior (Barredo Arrieta et al. 2020), continuous questions about the propensity for overfitting models to specific data sets and the robustness and validity of the predictions (Roelofs et al. 2019), and the requirements and demand for high-quality and substantial volumes of data in the financial industry (Nguyen et al. 2023).

Overcoming the opacity of many AI models presents a significant challenge, especially when it is difficult to identify the key variables influencing the prediction process. This lack of clarity often leads to reluctance in adopting ML for financial decision-making, as many algorithms, particularly in DL and RL, operate as 'black boxes'. Transparency is vital in scenarios prone to biases or exposed to substantial risks, necessitating rigorous auditing of these models.

In financial markets, the importance of transparency escalates in asset pricing and portfolio creation. These areas are especially susceptible to biases and significant risks. Thus, understanding and scrutinizing the decision-making processes of ML algorithms is essential. Ensuring that asset valuations and portfolio strategies are derived from fair and unbiased models is critical. They must accurately reflect the complex risk profiles inherent in financial markets, highlighting the need for thorough model auditing and transparent algorithmic operations. Therefore, explainability becomes a key factor in aiding decision-makers. It reveals how an AI model arrives at a decision and, when combined with domain knowledge, can foster trust among users (Dikmen and Burns 2022).

In asset pricing, biases can arise from over-reliance on particular data types or historical patterns, potentially leading to skewed valuations that do not accurately reflect market realities. Similarly, in portfolio construction, biases in algorithmic decision-making can result in inadequate exposure to certain asset classes, inadvertently magnifying risk instead of diversifying it. Moreover, the risks associated with opaque ML models are manifold. They range from financial losses due to poorly predicted market movements to regulatory repercussions if the models do not adhere to industry norms, governmental regulations, or policies (Nannini et al. 2023).

Additionally, there is the risk of reputational damage if stakeholders perceive the decision-making process as untrustworthy or ethically questionable. Therefore, the rigor in auditing these models is not just a technical necessity but also a safeguard against different sorts of risks (Bussmann et al. 2020), ensuring that decisions made in the financial markets, asset pricing, and portfolio management are robust, reliable, and aligned with the best interests of all stakeholders involved.

Another challenge is the quality and quantity of data required to function optimally for ML models. Financial markets generate vast data, but not everything is structured or clean. The preprocessing of data to remove noise and ensure its relevance for model training is a significant undertaking. Furthermore, the non-stationary nature of financial markets means that models must be continuously updated to remain relevant, posing a challenge regarding computational resources and the need for ongoing model validation. Therefore, there is a need for data cleaning and data compression techniques to avoid problems such as the dimensionality curse (Njah et al. 2021), where an excess of data can lead to overly complex models that are difficult to train and prone to overfitting. Effective data management strategies must be employed to balance the richness of data with computational feasibility and model accuracy.

Additionally, the risk of overfitting is a critical concern, where models excel with historical data but struggle to adapt to new, unseen market conditions. This issue can foster false confidence in the models' predictions, potentially resulting in substantial financial losses. To mitigate this, there is a growing need for sophisticated modeling techniques that can discern underlying relationships in data without being overly specific to past trends.

Therefore, integrating ML into existing financial systems requires a blend of domain expertise and technical knowledge. Financial professionals might find it necessary to enhance their skill set to understand and work alongside complex ML models, and there may be resistance to change from those accustomed to traditional methods.

Although ML offers interesting possibilities in asset pricing and portfolio construction, effectively dealing with its complexities requires careful consideration and a balanced combination of techniques and risk management strategies. As a result, this study was motivated by five key factors originating from market challenges and the limitations of existing methods in tackling them. The following points detail these challenges:

1. **Neural Network-Based Forecasting of Financial Market Time Series:** This segment explores the potential of ANN in forecasting financial market trends. While DL has shown promise in various domains, its application in financial time series forecasting at this thesis's beginning was not fully realized. The core aim is to develop and fine-tune neural network models to accurately predict market movements and complex time series data in financial markets.
2. **Dimensionality Reduction Strategies for Financial Time Series Clustering:** This research focuses on tackling the challenge of high-dimensional financial data. It aims to develop efficient dimensionality reduction methods using Autoencoders (AE) to simplify complex financial time series, enhancing clustering algorithms' ability to identify key market patterns and trends.
3. **Explainable AI for Predictive Analysis in Cryptocurrency Markets:** This research integrates XAI techniques with SHapley Additive exPlanations (SHAP) to forecast cryptocurrency market trends. Given the volatile nature of these markets, the study aims to develop AI models that are both accurate and transparent, providing clear insights into the most important variables in their predictive processes. The goal is to balance predictive power with understandability, ensuring reliable and transparent AI applications in the dynamic field of cryptocurrency trading and investment.
4. **Developing Transparent Reinforcement Learning Models for Portfolio Management:** This study focuses on creating RL models for portfolio management that are both effective and transparent. The goal is to develop strategies that optimize portfolios while remaining understandable to stakeholders, addressing the common issue of opaqueness in advanced AI techniques. The motivation is to enhance trust in AI-driven financial strategies by making their decision-making processes clear and accessible.

Given these challenges, this thesis explores advanced techniques and models to improve some practices and methods of asset pricing and portfolio construction within the financial markets.

1.2 Research Methodology

The research domain of this thesis is dynamically evolving, driven by technological advancements and a constant influx of new contributions in the fields of ML, RL, and DL. As a result, an iterative research methodology was adopted. The fundamental concept behind this cyclical process is that the insights gained during its initial phase serve as a foundation for designing progressively more promising techniques. These techniques are positioned to directly challenge existing methods, whether in terms of performance outcomes or conceptual innovations, thus offering original and notable contributions. Figure 1.1 illustrates the distinct phases of this research methodology, each of which is briefly outlined below.

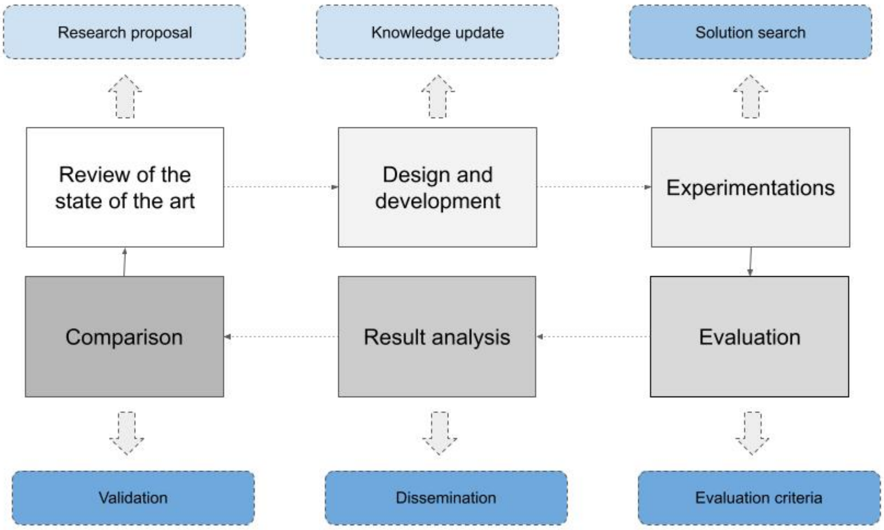


Figure 1.1: Followed research methodology.

1. **Review of the current state of the art:** In this phase, the principal aim is to comprehensively examine the current state of the art pertinent to the field of study, specifically identifying prevalent issues and challenges. This is accomplished through an extensive review of relevant literature in Quantitative Finance and different topics on ML, RL, and DL, including articles published in esteemed academic journals and papers presented at international conferences. Such a meticulous survey of the scientific community's contributions is

crucial for a deep understanding of the field. The knowledge acquired in this phase is anticipated to lead to the development of a solid research proposal that is intended to be a solution that adeptly addresses the challenges that have been identified. This proposal should be both innovative and feasible and integrate current knowledge and new possibilities for further research.

2. **Design and development:** This phase focuses on applying updated knowledge to create a solution that reflects the latest state of the art and tries to improve it. This process involves planning and executing coding strategies with the use of appropriate programming tools and frameworks, emphasizing accuracy and clearness.

   The development process is cyclical, involving repeated coding, testing, and adjustment rounds. This approach ensures that every aspect of the project is executed with precision and aligns with the intended objectives. Techniques such as continuous integration and continuous deployment are employed throughout the process to guarantee consistently high-quality code.

3. **Experimentations:** The emphasis lies in conducting a series of experiments to test the practicality and effectiveness of the developed solution. This phase is characterized by a systematic approach to validating hypotheses and assessing the solution's performance under diverse conditions. The experiments are designed to be robust and replicable, ensuring the reliability of the results.

4. **Evaluation:** The attention turns towards a comprehensive evaluation of the solution's overall impact and effectiveness. This requires thoroughly examining the results against established evaluation criteria and benchmarks. This phase is crucial for assessing the solution's efficacy in achieving its objectives and deriving valuable insights.

5. **Result analysis:** The emphasis is placed on conducting a thorough analysis and interpretation of the data gathered from the experiments. It helps to disseminate the information that is created. This phase allows for analyzing the results to understand better the underlying patterns, correlations, and possible causations. This phase is essential for evaluating the developed

solution's strengths and weaknesses and establishing a solid foundation for future research.

6. **Comparison:** The focus is on comparing the solution to established standards, other solutions in the field, or previous research findings in research journals. Conducting this comparative analysis to situate the solution into the broader field context is crucial. The evaluation process focuses on analyzing the performance of the solution in comparison to others, considering key metrics. This phase emphasizes the importance of showcasing the distinctive contributions and benefits of the solutions compared to the state of the art, validating the current research.

1.3 Research Hypotheses

In this thesis, three hypotheses are put forth for validation, as outlined below:

- **Hypothesis 1:** By using ANN and DL in forecasting financial market trends could result in markedly improved accuracy in predicting market movements. With the utilization of advanced data processing capabilities, this approach is expected to outperform conventional methods in detecting and understanding complex market patterns.
- **Hypothesis 2:** By developing efficient dimensionality reduction methods using AE can substantially simplify the analysis of high-dimensional financial data. These methods are anticipated to significantly enhance clustering algorithms' performance by effectively reducing the complexity of financial time series data. This improvement in data processing is expected to lead to more accurate identification of key market patterns and trends.
- **Hypothesis 3:** This hypothesis asserts that using XAI techniques, particularly SHAP, in cryptocurrency market forecasting will enhance transparency of predictions. Integrating an explainable method is expected to unravel the AI's decision-making process, balancing advanced predictive power with user understandability and trust. This enhancement facilitates a clearer understanding of the key variables influencing decision-making.

- **Hypothesis 4:** This hypothesis suggests that utilizing RL models for portfolio management with explainable components can enhance transparency and significantly improve portfolio optimization compared to traditional methods. The RL model can adjust to intricate and evolving environments and is expected to outperform conventional portfolio management strategies by effectively identifying investment opportunities. By combining effectiveness and explainability, this RL models are expected to build trust with stakeholders, gaining a deeper insight into the relevant variables that impact decision-making.

In order to validate these hypotheses, four significant contributions have been developed as part of this thesis. The contributions involved creating a practical and hands-on environment for achieving these outcomes.

1.4 Contributions

The research carried out in this study has led to several key contributions to the field, as outlined below:

- **Advancement in Financial Market Forecasting:** The research contributes to the advancement of financial market forecasting by exploring the potential of ANN and DL for predicting market trends. Developing and fine-tuning neural network models aims to improve the accuracy of market movement predictions, harnessing the power of DL in handling complex time series data.
- **Efficient Dimensionality Reduction for Market Analysis:** The study addresses the challenge of high-dimensional financial data by developing efficient dimensionality reduction methods using AE. This contribution simplifies complex financial time series data, making it easier for clustering algorithms to identify key market patterns and trends. It enhances the overall understanding of financial data.
- **Enhancing Transparency and Accuracy in Cryptocurrency Market Predictions:** Advanced the use of XAI techniques, particularly integrating SHAP, for predictive analysis in cryptocurrency markets. This research led to the

development of ML models that are accurate and transparent, offering clear insights into the predictive variables. This innovation balances predictive strength with explainability, fostering reliable and transparent applications in the dynamic and volatile cryptocurrency markets.

- **Enhancing Portfolio Management through Transparent RL Models:** Developed innovative and transparent RL models for portfolio management, which effectively optimize investment strategies while maintaining clarity and accessibility. This approach addresses the challenge of opaqueness in advanced AI techniques, enhancing stakeholder trust in AI-driven financial decisions.

1.5 Publications

Throughout the course of this thesis research, multiple articles were published in international peer-reviewed journals and presented at conferences to disseminate the results obtained. These publications are listed in Table 1.1.

TITLE Time Series Forecasting Using Artificial Neural Networks:
A Model for the IBEX 35 Index
AUTHORS Daniel González-Cortés, Enrique Onieva, Iker Pastor, Jian Wu
CONFERENCE Conference on Hybrid Artificial Intelligence Systems, 2022
STATUS Published. Vol. 13469, pp. 249-260, 2022

TITLE The application of artificial neural networks to forecast financial time series
AUTHORS Daniel González-Cortés, Enrique Onieva, Iker Pastor, Jian Wu
JOURNAL Logic Journal of the IGPL
STATUS Published. ISSN 1367-0751, 2024

TITLE Autoencoder-Enhanced Clustering: A Dimensionality Reduction Approach to Financial Time Series
AUTHORS Daniel González-Cortés, Enrique Onieva, Iker Pastor, Laura Trinchera, Jian Wu
JOURNAL IEEE Access
STATUS Published. vol. 12, pp. 16999-17009, 2024

TITLE Portfolio Construction Using Explainable Reinforcement Learning
AUTHORS Daniel González-Cortés, Enrique Onieva, Iker Pastor, Laura Trinchera, Jian Wu
JOURNAL Expert Systems
STATUS Published, 2024

TITLE Application of Explainable Artificial Intelligence in predicting digital asset prices
AUTHORS Daniel González-Cortés, PK Senyo, Monomita Nandy, Suman Lodh, Jian Wu, Enrique Onieva
JOURNAL International Journal of Electronic Commerce
STATUS Submitted

Table 1.1: Publications conducted during this thesis.

1.6 Outline

This dissertation is divided into eight chapters. Next, a summary of each chapter is presented:

- **Chapter 2: Background.**

  This chapter investigates the impact of AI on asset pricing and portfolio construction, highlighting its role in creating intelligent models. It explores into how different models integrate into the financial sector to enhance predictive accuracy, optimize portfolio allocation, and improve data-driven financial decision-making. By bridging computational methods with traditional financial theories, this chapter reviews traditional asset pricing models, portfolio techniques, ML, ANN, clustering methods, dimensionality reduction, and the application of RL and explainability in ML models.

- **Chapter 3: State of the art.**

  This chapter explores into the most recent advancements and applications of AI, ML, DL, and RL within the financial sector, with a special emphasis on asset pricing and portfolio construction. It highlights the field's latest developments, showcasing how innovative algorithms enhance traditional financial models and strategies.

- **Chapter 4: Neural Network-Based Forecasting of Financial Market Time Series.**

  This chapter investigates into the use of ANN and DL for forecasting financial market trends. It explores how these advanced technologies can process complex time series data to predict market movements more accurately, highlighting their potential over traditional forecasting methods.

- **Chapter 5: Dimensionality Reduction Strategies for Financial Time Series Clustering.**

  This chapter focuses on developing efficient dimensionality reduction methods using AE to simplify high-dimensional financial data. It discusses how these methods enhance the ability of clustering algorithms to identify crucial market patterns and trends in complex financial time series.

- **Chapter 6: Explainable AI for Predictive Analysis in Cryptocurrency Markets.**

  This chapter presents the integration of XAI techniques with SHAP for forecasting cryptocurrency market trends. It emphasizes the development of AI models that are accurate and transparent, offering insights into the predictive processes and balancing predictive power with understandability in the volatile cryptocurrency market.

- **Chapter 7: Developing Transparent Reinforcement Learning Models for Portfolio Management.**

  This chapter discusses the creation of RL models for portfolio management, focusing on their effectiveness and transparency. It addresses the challenge of opaqueness in AI techniques, aiming to develop strategies that optimize portfolios while being understandable to stakeholders, thereby enhancing trust in AI-driven financial strategies.

- **Chapter 8: Conclusions and Future Research.**

  This concluding chapter synthesizes the key findings from the preceding chapters, highlighting the significant contributions and advancements made in financial market analysis using AI and ML techniques. It reflects on the challenges encountered, the solutions developed, and their impact on current and future financial strategies. The chapter also outlines potential areas for future research, suggesting directions for further enhancing and refining AI applications in financial forecasting, portfolio management, and investment strategy development, focusing on transparency, accuracy, and innovation.

2

Background

AI is a domain of computer science that aims to create intelligent machines that can perform tasks that traditionally require human intelligence. AI covers a spectrum of functions from learning to problem-solving, including different cognitive abilities such as comprehension of human language and vision. In a more constrained manner, ML is a computational approach that enables a computer to adapt to new situations and detect and extrapolate data patterns (Russell and Norvig 2010). It involves the construction of algorithms that can acquire knowledge from input data and use it to generate specific predictions. These capacities enable the computer to have decision-making capabilities without being explicitly programmed for each task.

This thesis is dedicated to examining the integration of AI and its various approaches—including ML, DL, and RL—within the domain of asset pricing and portfolio construction. The objective is to bridge the gap between cutting-edge computational methods and conventional financial theories. The exploration aims to understand how these contemporary technologies can bolster predictive accuracy, refine portfolio allocation strategies, and enhance financial decision-making processes in a complex, data-centric market environment. The chapters are methodically structured to provide theoretical and practical insights into these advanced computational techniques and their synergistic application with established financial models.

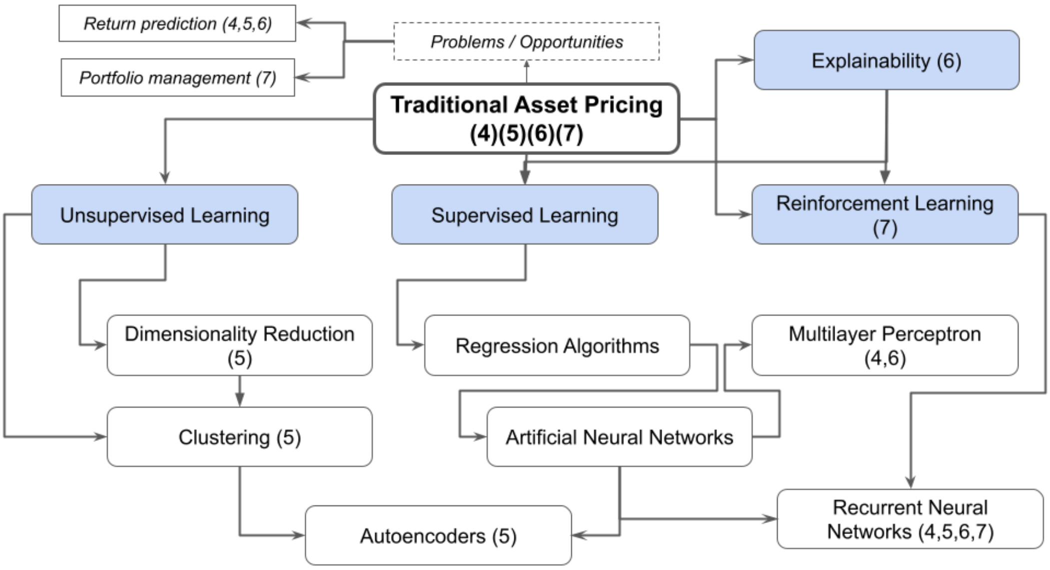


Figure 2.1: Integration of Machine Learning in Asset Pricing.

The provided scheme in Figure 2.1 delineates a framework for augmenting traditional asset pricing models with contemporary ML methodologies. It integrates unsupervised learning, supervised learning, and RL, alongside with explainability. Specific applications are categorized under return prediction and portfolio management, indicated by citations to Chapters 4, 5, 6, and 7 respectively.

In the following sections, the discussion will be expanded to elucidate the individual constituents of this framework. The chapter initiates with a detailed exploration of traditional asset pricing models and portfolio construction methods in Section 2.1, including essential technical analysis indicators. This foundational analysis sets the stage for advancing into sophisticated ML regression algorithms, which are elaborately discussed in Section 2.2. Then, Section 2.3 delves into ANN, examining the architecture and applications of complex data modeling. In Section 2.4, the discussion is dedicated to clustering techniques, delving into how these techniques categorize data into subsets by similarity, aiding in revealing underlying structures within the dataset. Section 2.5 introduces dimensionality reduction techniques. Following this with an overview of RL in Section 2.6, which clarifies the strategy and implementation of this subject, it then moves on to Section 2.7 that discusses the fundamental ideas of explainability in ML models.

2.1 Traditional Asset Pricing Models and Portfolio Construction Techniques

This chapter provides an in-depth analysis of the foundational principles of classical asset pricing and portfolio construction. It provides a comprehensive examination of their theoretical foundations and practical implementations. It begins with an overview of key asset pricing models, such as the CAPM and Arbitrage Pricing Theory (APT). Then, it transitions into the principles of modern portfolio construction, including diversification strategies and risk management. Furthermore, the chapter includes a technical analysis, emphasizing its significance in investment strategy and the process of making informed decisions. Integrating these elements provides a comprehensive understanding of effective portfolio management in financial markets.

2.1.1 Efficient Markets

The main concern of many economic agents is forecasting the financial markets' future trends to make better decisions. The methods used and the time frames to predict are diverse.

Securities are financial instruments that hold significant importance within the financial markets. These instruments represent various forms of financial value and serve as crucial mechanisms in modern economies. Their primary function is to facilitate the exchange of these instruments within stock markets, where their prices are subject to frequent fluctuations. These market variations are often considered chaotic and non-stationary, making predicting future prices complex. However, historical empirical evidence (Lo and MacKinlay 1988) indicates that stock returns may have some predictable components. This evidence challenges the random walk hypothesis, which posits that stock prices move randomly and are unpredictable. The existence of predictable elements in stock returns opens up avenues for investors to use various analytical methods to forecast market movements and make informed investment decisions.

All the economic agents need to be aware of the implications that the stock market has on different levels of the economy, as seen in the global financial crisis of 2007–2009 when a financial contagion occurred affecting different sectors of the real economy such as consumer goods, industrials, telecommunications, and technology (D. Baur 2012). Forecasting future stock prices and trends can be critical for any economic agent to make better business decisions. However, this is not an easy task because the nature of the stock market is intrinsically nonlinear, non-parametric, and chaotic due to the interactions of many variables that make the market price move in one direction or another.

The prediction process has been approached through fundamental and technical analysis. The first is based on the asset's intrinsic value valuation by using the company's current and future earnings to evaluate the fair value (Wafi et al. 2015) and compare it with the market value indexed in the stock exchange. This methodology used the company's financial statements as the primary source of information; instead, technical analysis did not merely rely on this data but used historical asset prices to make predictions. Many investors use both methodologies to make buying or selling decisions, and 87 % of fund managers use technical analysis (Menkhoff 2010).

In this realm of forecasting asset prices, the Efficient Market Hypothesis (EMH) presents itself as a pivotal theoretical framework widely regarded as a fundamental principle in contemporary financial theory, which asserts that the prices of assets in financial markets accurately incorporate all relevant information available at any given time. EMH, primarily, suggests that it is impossible to achieve returns over the average market consistently on a risk-adjusted basis, considering the premise that market prices are supposed to respond solely to new information (Eugene F Fama 1970). EMH is categorized into weak, semi-strong, and strong. The weak form asserts that historical trading data and prices are entirely incorporated into stock prices, thus making technical analysis ineffective. The semi-strong form extends this to all publicly available information, diminishing the fundamental analysis's utility. The strong form of the EMH posits that market prices reflect all public and private information, making it impossible for even insiders to gain an advantage. While influential, this hypothesis is subject to debate, with critics pointing to anomalies like market bubbles and behavioral economics insights that suggest market participants do not always act rationally or have equal access to information.

The EMH is a crucial principle underpinning classical asset pricing theory because it provides a foundational understanding of how information is reflected in asset prices. This theory revolves around a trade-off between the risk and returns of holding assets, where the expected return on an asset is directly proportional to its risk.

2.1.2 Theoretical Foundations of Portfolio Theory

This risk-return trade-off is a central principle of modern portfolio theory, developed by Markowitz (1952). This theory asserts that the fundamental aspect of investment decision-making lies in balancing risk against expected returns, and diversification is a key strategy for achieving this balance.

The core concept of diversification suggests that by combining various types of assets, investors can construct a portfolio that mitigates overall risk while maintaining or enhancing expected returns. Hence, investors must consider the prospective returns of their investment selections and the inherent risks linked to these returns. The notion of portfolio diversification significantly transformed the comprehension of asset allocation and portfolio construction, shifting the emphasis from individual asset selection to adopting a broad perspective on portfolio performance that considers all assets' collective performance.

Furthermore, this method acknowledges the important role of covariance between asset returns when optimizing portfolios, creating an efficient frontier that graphically represents the set of optimal portfolios that, given a specific level of risk, yield the highest expected return.

Building upon the established principles of modern portfolio theory, the capital asset pricing model CAPM further refines the understanding of risk, return, and asset pricing. Developed independently in the 1960s based on the unpublished manuscripts of Jack Treynor (C. W. French 2003), William Sharpe, John Lintner, and Jan Mossin (Lintner 1965a; Lintner 1965b; Mossin 1966; Sharpe 1964) introduced the concept of systematic and unsystematic risk with profound implications for asset pricing and portfolio management.

Systematic risk, alternatively referred to as market risk, impacts the entire market and is unavoidable when maintaining a diversified portfolio. Broad economic factors, including interest rates, inflation, and economic growth, are commonly linked to this risk category. Unsystematic risk, conversely, is a risk unique to a specific asset or company and is also referred to as idiosyncratic risk.

This approach plays a key role in asset allocation by quantifying the relationship between risk and expected return, allowing investors to estimate the expected return for a given degree of market risk. Furthermore, it serves as a benchmark for evaluating the performance of assets and portfolios, facilitating the evaluation of whether increased returns result from wise investment decisions or simply assuming more risk.

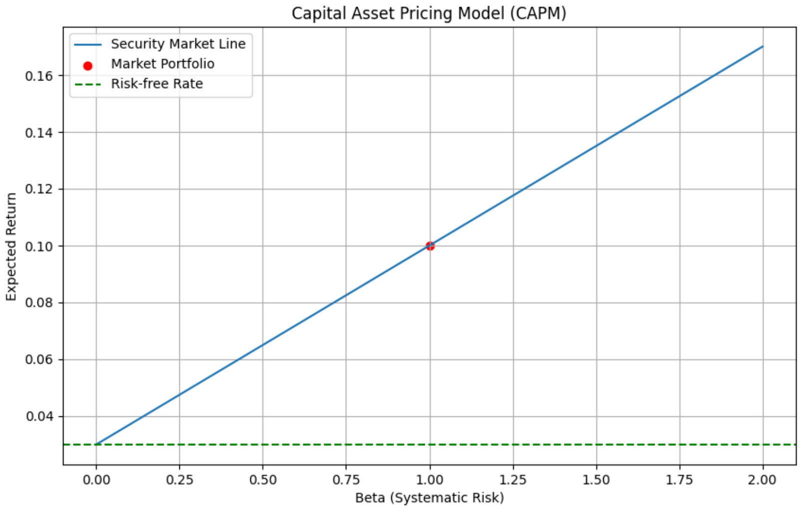
According to CAPM, ss shown in Equation 2.1, the expected return on a security or portfolio equals the risk-free rate plus the risk premium, which is the product of the asset's beta and the expected market risk premium.

$$E(R_i) = R_f + \beta_i \times (E(R_m) - R_f) \tag{2.1}$$

Where,

- $E(R_i)$ represents the expected return on the investment.
- R_f represents the rate of return on an investment that is considered to have no risk.
- β_i is the investment's beta, indicating its relative volatility.
- $E(R_m)$ is the expected market return.
- $E(R_m) - R_f$ is the market risk premium, the excess return anticipated from investing in a risky market portfolio compared to a risk-free asset.

CAPM significantly enhances financial theory by quantifying an asset's risk exposure to market fluctuations, the beta, and the additional return demanded for bearing market risk, the market risk premium; it is possible to establish a direct correlation between risk and expected return.



.

Figure 2.2: Security Market Line as per the CAPM

Figure 2.2 illustrates the linear relationship between the beta asset's systematic risk, also known as beta, and its expected return. Where beta equals one, the red dot represents the market portfolio, and the green dashed line represents the risk-free rate. The slope of the security market line corresponds to the market risk premium, which is the additional return associated with assuming a more significant risk.

The APT, developed in the mid-1970s (Ross 1976), is a multifactor asset pricing model that extends beyond the single-factor approach of the CAPM. APT suggests that the expected return of a financial asset can be modeled as a linear function of various macroeconomic factors, each associated with a specific risk premium. Unlike CAPM, which primarily focuses on systematic risk and represents it using a single beta coefficient, APT recognizes that multiple factors influence the risk and return of an asset. These factors include inflation, interest rates, market indices, and other economic variables.

The theory suggests that arbitrage opportunities help maintain prices closely aligned with their expected values, as determined by relevant factors. Furthermore, APT does not require the market portfolio to be efficient, a critical assumption in CAPM. This makes APT more flexible and applicable in diverse market scenarios.

However, APT is also more complex in its implementation, requiring identifying and analyzing multiple risk factors and their respective sensitivities for each asset.

The Fama-French model, introduced in the early 1990s (Eugene F. Fama and K. R. French 1993), expands the field of asset pricing models by incorporating numerous factors that empirical research has shown to better explain asset returns than the CAPM. There are several Fama-French models; however, the first model includes, in addition to the market risk factor inherent in the CAPM, size, and value factors. These factors capture the excess returns of small-cap stocks over large-cap stocks and value stocks over growth stocks.

Subsequent versions of their model have incorporated additional factors and identified some risk factors in the returns on stocks and bonds (Eugene F. Fama and K. R. French 2015a) among others, including profitability and investment patterns (Eugene F. Fama and K. R. French 2015b), indicating a more detailed comprehension of the factors influencing asset returns. Unlike APT, which suggests a potentially unlimited number of factors, the Fama-French model specifies certain key factors that have consistently explained returns across various periods and markets.

2.1.3 Return prediction

Return prediction is a fundamental aspect of financial analysis and portfolio management. It involves using methodologies and models to estimate the future performance of assets. Accurate return forecasting is crucial for formulating investment strategies, asset allocation, and, ultimately, achieving portfolio optimization.

Accurate return forecasting is essential for developing investment strategies, allocating assets, and optimizing portfolio performance. Statistical methods utilize historical data to infer future returns using models such as Auto-Regressive Integrated Moving Average (ARIMA) and Generalized Autoregressive Conditional Heteroskedasticity (GARCH). These models typically assume that historical behavior can provide valuable insights into future performance, considering the model's structure and the inherent unpredictability of financial markets.

In contrast, fundamental analysis considers multiple variables such as economic indicators, financial statements, and market conditions to forecast returns. This approach is based on the premise that a security's intrinsic value can be determined and that market inefficiencies will eventually be resolved, leading to prices aligning with this intrinsic value.

Like statistical methods, technical analysis uses historical behaviors to predict the past. This methodology is based on the belief that market trends and patterns, as observed in asset price movements and trading volumes, can indicate future performance. Technical analysts scrutinize charts and use various indicators to predict price trajectories.

In technical analysis, trend indicators such as moving averages that can be computed as simple, exponential, or weighted help smooth out price data to reveal underlying trends.

The Simple Moving Average (SMA), as defined in Equation 2.2, calculates the average price over specific periods. It is represented as:

$$SMA = \frac{P_1 + P_2 + \cdots + P_n}{n} \tag{2.2}$$

Where $P_1 + P_2 + ... + P_n$ are prices over n periods. SMA gives equal weight to each price within the period.

The Exponential Moving Average (EMA), detailed in Equation 7.14, delivers a more dynamic alternative to the SMA by giving more weight to recent prices. It is calculated as:

$$EMA_t = \left(\frac{P_t}{1+\alpha}\right) + \left(\frac{\alpha \times EMA_{t-1}}{1+\alpha}\right) \tag{2.3}$$

Where P_t is the current price EMA α is a smoothing factor, and EMA_{t-1} is the EMA of the previous period. Emphasis on current prices makes EMA more responsive to recent market movements.

The Weighted Moving Average (WMA), outlined in Equation 2.4, extends this idea by assigning different weights to each price in the series and assigns greater importance to recent prices. It is calculated as:

$$WMA = \frac{\sum_{i=1}^{n} w_i \times P_i}{\sum_{i=1}^{n} w_i} \tag{2.4}$$

Another trend indicator widely used in technical analysis is the Moving Average Convergence Divergence (MACD). This indicator is designed to reveal changes in the strength, and direction of a trend in a stock's price by combining the use of multiple moving averages.

$$MACD = EMA_{12}(Price) - EMA_{26}(Price) \tag{2.5}$$

$$Signal = EMA_9(MACD) \tag{2.6}$$

The MACD, as defined in Equation 2.5, is calculated by subtracting the 26-period EMA from the 12-period EMA of the price, even though these periods do not necessarily need to remind fixed. As a result of this subtraction, the MACD line is produced, which reflects the relationship between these two EMAs and, as a result, an indicator of the general trend in the market.

Equation 2.6 introduces the Signal Line, computed as the 9-period EMA of the MACD. This line triggers buy and sell signals, providing a smoother representation of the MACD line. Trading signals are generated when the MACD crosses this Signal Line.

The components of the MACD provide a versatile tool for analyzing market trends using convergence, divergence, and crossing to highlight momentum shifts and potential trend reversals.

Another indicator is the Relative Strength Index (RSI), a momentum oscillator that measures price changes in a given period. This indicator is calculated as,

$$RSI = 100 - \frac{100}{1 - RS_p} \tag{2.7}$$

Where RS_p is the relationship between the upward price averages and the average downward price change in period p.

An additional signal is the Stochastic Oscillator (SO), a momentum indicator used in technical analysis that signals overbought and oversold market conditions. It is calculated using the following formula:

$$SO = 100 \times \frac{C - L_p}{H_p - L_p} \tag{2.8}$$

where C, is the closing price and H_p is the highest value and L_p is the lowest value in period p. The SO oscillates between 0 and 100, with readings above 80 typically considered overbought and below 20 considered oversold. These levels suggest possible turning points in the market's direction, suggesting the asset is headed toward a price correction.

The On-Balance Volume (OBV) is a technical indicator that combines price and volume, and it is defined as,

$$\begin{cases} C_p = C_{p-1} & \Rightarrow OBV(p) = OBV(p-1) \\ C_p < C_{p-1} & \Rightarrow OBV(p) = OBV(p-1) - V_p \\ C_p > C_{p-1} & \Rightarrow OBV(p) = OBV(p-1) + V_p \end{cases} \tag{2.9}$$

where C_p and V_p are the closings and the volume at time p respectively. This formula for OBV demonstrates the calculation of the indicator, which is determined by the closing price movement between consecutive periods, integrating both price movement and volume to provide insights about market trends.

Lastly, the Williams %R (WR) is an indicator that reflects the closing price level relative to the high-low range over a specified period.

$$WR = \left(\frac{H_n - C}{H_n - L_n} \right) \cdot -100 \tag{2.10}$$

In this formula, H_n represents the highest high over the past n periods, L_n is the lowest low over the same period, while C is the current closing price.

In addition to technical analysis, the advent of ML has marked a significant evolution in return prediction and portfolio construction. By utilizing sophisticated algorithms and extensive datasets and possibly integrating technical analysis as inputs and various other methods ML methods are adept at uncovering intricate, non-linear patterns and correlations that might elude traditional analytical approaches. Among the numerous techniques available, regression models and ANN stand out for their direct applicability in forecasting. These are complemented by auxiliary methods like clustering and dimensionality reduction, which aid in refining the predictive models. Additionally, more advanced strategies such as RL are gaining traction, further expanding the horizon of predictive analytics in finance. Adopting ML improves prediction accuracy and provides new opportunities for analyzing market dynamics using data-driven approaches.

2.2 Regression Algorithms

This section concisely describes several regression ML algorithms, each of which stands out for its particular methodologies and application contexts. The focus will be on CatBoost Regressor (CATR), Decision Tree Regressor (DTR), Epsilon-Support Vector Regressor (ESVR), K-Nearest Neighbor's Regressor (KNNR), LightGBM (LGBM), Random Forest Regressor (RFR), eXtreme Gradient Boosting Regressor (XGBR). These algorithms provide insights into the numerous types of methods used in modern ML, spanning from simple ensemble techniques to individual predictive models. A comprehensive understanding of these algorithms is essential for advancing and enhancing sophisticated predictive models, with implications for both theoretical progress and practical implementations.

2.2.1 CatBoost Regressor

The CATR algorithm, as proposed by Prokhorenkova et al. (2018), represents a significant advancement in the field of ML. CATR was developed with the intention of enhancing the existing state-of-the-art methods in ML, particularly focusing on reducing the time required for training and parameter tuning. This algorithm is a part of the gradient-boosting decision tree family. Still, it distinguishes itself by implementing an efficient mechanism designed to prevent overfitting, a common challenge in ML models.

One of the key strengths of CATR is its exceptional proficiency in handling categorical features. Unlike other algorithms, it does not require extensive pre-processing of categorical data, simplifying the data preparation phase and enhancing model performance. Furthermore, CATR demonstrates remarkable versatility and effectiveness when dealing with various types of datasets, including nonlinear, small, or unbalanced, making it a robust choice for a wide range of applications.

The utility of CATR are diverse, and its practical applications spanning different fields and big data scenarios, as noted by Hancock and Khoshgoftaar (2020). This adaptability has led to its widespread adoption in different fields. The algorithm's ability to deliver high-quality results efficiently makes it a valuable tool for data scientists and ML practitioners aiming to derive meaningful insights from complex datasets.

2.2.2 Decision Tree Regressor

Conversely, the DTR is recognized as a significant and widely-used predictive modeling technique within the field of ML. This method involves the creation of a decision tree, which serves as a predictive model, guiding data analysis. In this model, observations are mapped through a series of branches, each representing a decision point based on specific criteria or features of the data. These branches eventually lead to conclusions or predictions about specific target values, such as a class in classification tasks or a continuous value in regression.

The decision tree's structure is particularly noteworthy for its clarity and simplicity. It provides a transparent, interpretable visual representation of the decision-making process. In the structure of a decision tree, each branch and leaf represents a sequential series of query and response pairs, systematically running the analysis from initial data observations to a decision. This orderly and visual progression makes decision trees particularly accessible, especially to individuals with limited expertise in ML. Such a design aids in clarifying the model's decision-making process, thereby improving the understanding of how the model makes its decisions.

Moreover, this method performs well in handling outliers and can deal with missing values, thus showing robustness across various scenarios. Also, the versatility of the DTR algorithm is evident by its flexibility in its ability to handle both classification and regression tasks, making it a valuable tool for a wide array of applications (Lessmann et al. 2021; Shen et al. 2021). Its widespread use across such diverse fields underscores its effectiveness in extracting insights from various data types, contributing to decision-making processes in numerous industries.

2.2.3 Epsilon-Support Vector Regressor

The ESVR algorithm stands out from other regression techniques for its proficiency in capturing complex nonlinear functions, as highlighted by Drucker et al. (1996). This algorithm builds upon the foundational principles of the widely acknowledged Support Vector Machine (SVM) algorithm, initially designed for classification tasks, and adapts them for regression.

The ESVR algorithm's primary aim is to identify a nonlinear function that accurately represents the data. To achieve this, ESVR utilizes a fundamental theoretical technique known as a kernel. A kernel is a mathematical function that converts the input data into a higher-dimensional space. The significance of this transformation lies in its ability to render data linearly separable in a new dimensional space, even when it is not linearly separable in the original dimensions. Kernels act as bridges, allowing the algorithm to operate in a more expansive and complex feature space without the computational burden of directly computing the dimensions of this higher space.

Within the ESVR framework, after the transformation by the kernel, the algorithm seeks to find a function in this new space that approximates the data with a certain level of tolerance, defined by epsilon. This parameter introduces a tolerance band around the regression line, accepting predictions within this range as acceptable. This mechanism controls the model's sensitivity to errors, effectively preventing overfitting and ensuring robust performance. The practical applications of ESVR are vast, especially in sectors where understanding and predicting nonlinear dynamics is crucial (Y. Chen and Hao 2017; Dash et al. 2021).

2.2.4 K-Nearest Neighbor's Regressor

The KNNR stands as an efficient non-parametric model within the scope of regression analysis, utilizing the principles of the k-nearest neighbors algorithm to estimate continuous variables. Central to the KNNR methodology lies the principle of predicting target values by closely analyzing the characteristics of the nearest neighbors within the dataset. This method employs local interpolation to form sets of nearest neighbors, each contributing to the prediction of a target variable.

A distinctive aspect of the KNNR algorithm is its method of computation, which involves calculating the weighted average of the inverse Euclidean distances of the k-nearest multivariate neighbors. This calculation ensures that closer neighbors significantly influence the prediction, enhancing the accuracy and relevance of the model's output.

The rising prevalence of KNNR can be explained by multiple factors. Its straightforwardness generates a compelling option for numerous applications. Unlike some other regression models, KNNR does not require prior assumptions about the distribution of data, allowing for greater flexibility in its application across diverse datasets.

However, the KNNR model does have its limitations. One notable drawback is its performance speed, which tends to decrease as the size of the dataset increases. This is because the algorithm must compute distances between each query point and all points in the dataset, which becomes increasingly time-consuming with larger datasets. As a result, while KNNR is praised for its simplicity and effectiveness in smaller datasets, its scalability and efficiency in handling large data volumes can be a concern. This limitation is important when choosing KNNR for big data or real-time analysis applications.

2.2.5 LightGBM

An additional notable approach in the domain of ML algorithms is the LGBM, developed to handle large, high-dimensional datasets adeptly. This algorithm employs a gradient-boosting decision tree framework, specifically focusing on optimizing performance to improve efficiency and accelerate the training process. A key aspect of LGBM's design is its focus on maximizing parallel learning, allowing it to process vast datasets more quickly than traditional methods.

A significant and innovative modification in the LGBM algorithm is its unique strategy of constructing trees. Unlike conventional methods that grow trees horizontally in a level-wise manner, LGBM grows its trees vertically, also known as leaf-wise. This leaf-wise growth strategy is pivotal in how LGBM achieves high efficiency. By focusing on the best leaf to split during the growth of each tree, LGBM can reduce loss more rapidly compared to the level-wise approach. This method enables the algorithm to achieve higher accuracy with fewer resources, effectively reducing computational time and memory usage.

The practical implications of these advancements are substantial, particularly in the context of big data challenges. LGBM is widely acclaimed for its ability to manage an extensive number of data features and instances simultaneously, making it a highly sought-after tool among scholars and practitioners dealing with complex datasets, as noted by Hancock and Khoshgoftaar (2020). Its proficiency in handling large-scale data, factual accuracy, and reduced memory requirements render LGBM an invaluable tool in domains where fast processing of massive datasets is crucial.

2.2.6 Random Forest Regressor

The RFR algorithm, one pivotal model within the domain of ensemble learning methods, is widely recognized in ML for its efficacy in handling both regression and classification tasks. At its core, the RFR algorithm employs a multitude of decision trees, each developed independently based on a randomly selected subset of variables. The distinctive architecture of this algorithm serves as the foundation for its robustness and flexibility.

The RFR algorithm constructs the individual trees using a random subset of the data, ensuring that each tree develops distinct decision paths. Every tree undergoes an evaluation to determine its performance following the growing phase, and trees that do not show enough accuracy or robustness are removed from the ensemble. Subsequently, the remaining trees are combined, wherein the ultimate prediction is determined as an average result obtained from the collective predictions of all the trees within the forest. Utilizing an averaged process enhances the stability of predictions by mitigating variance, thereby improving the model's accuracy.

One of the main advantages of the RFR algorithm lies in its inherent ability to handle diverse datasets and tasks efficiently. By integrating diverse decision trees, each with its perspective on the data, the algorithm can capture a wide range of patterns and relationships within the dataset. Furthermore, the RFR is adept at preventing overfitting by using bootstrap aggregating methods during the training phase, where each tree is trained on a different subset of the training data. This methodology guarantees that every tree in the ensemble is trained on a distinct subset of the data, resulting in a more generalized model that exhibits strong performance on previously unseen data.

The versatility and robustness of the RFR algorithm have catalyzed its widespread application across diverse domains. Its adeptness in managing large and complex datasets, with a high degree of accuracy and an inherent capacity to prevent overfitting, creates a powerful ML tool.

2.2.7 eXtreme Gradient Boosting Regressor

The XGBR algorithm is distinguished as a highly optimized, distributed system for gradient boosting. The development of XGBR represents a significant leap in boosting algorithms, achieved through a series of enhancements in various aspects of computational efficiency. These enhancements include optimized cache access patterns, significantly reducing the time taken to retrieve data during computation, and innovative data compression techniques, which reduce memory usage while maintaining data integrity.

Another crucial component of XGBR is developing a scalable tree-boosting system that efficiently handles large datasets and complex modeling tasks, making it particularly effective for high-dimensional data. The scalability of XGBR ensures that it remains effective and efficient even as the size and complexity of the dataset increase.

The popularity of XGBR in the field of ML is also attributed to its exceptional performance across multiple applications. This is partly due to its robustness against multi-collinearity, a common issue in datasets with highly correlated predictor variables. This robustness improves the model's reliability and accuracy, making accurate predictions even in challenging datasets.

2.3 Artificial Neural Networks

ANNs are a key part of ML; they are bio-inspired computing systems comprised of interconnected processors known as neurons that are activated by diverse Activation Function (AF) based on specified weights and biases. By changing the weights of several coupled neurons via feed-forward optimization, this system aims to reduce the prediction error obtained by calculating the difference between predicted and expected outputs. The ANN can capture nonlinearities, a critical attribute for identifying intricate and diverse connections within data.

Additionally, the ability of the ANN to acquire knowledge and enhance performance through extensive data processing demonstrates their increasing power. With the rising availability of data, ANNs can be trained to refine and improve their predictions and decision-making abilities, continuously enhancing their performance.

Hence, utilizing these systems has emerged as a fundamental element in the progress of ML and AI, propelling innovation and effectiveness in various sectors.

2.3.1 Multilayer Perceptron

The neurons of the ANN are structured in different layers, creating a Multilayer Perceptron (MLP) network structure, which is one of the most widely implemented neural network topologies (Guresen et al. 2011). The basic MLP layers can be classified as input, output, and hidden. The other components in the models are weights, which are the connecting coefficients between layers, and AFs that trigger a signal given a weighted sum of its input. The general structure of the ANN is presented in Figure 2.3.

The inputs in an MLP are the initial data points that the network processes. These inputs are usually structured as a vector, each representing a unique feature of the data being analyzed. In the context of an ANN, these features are analogous to the sensory inputs in a biological nervous system, providing the raw information that the network will use to make decisions.

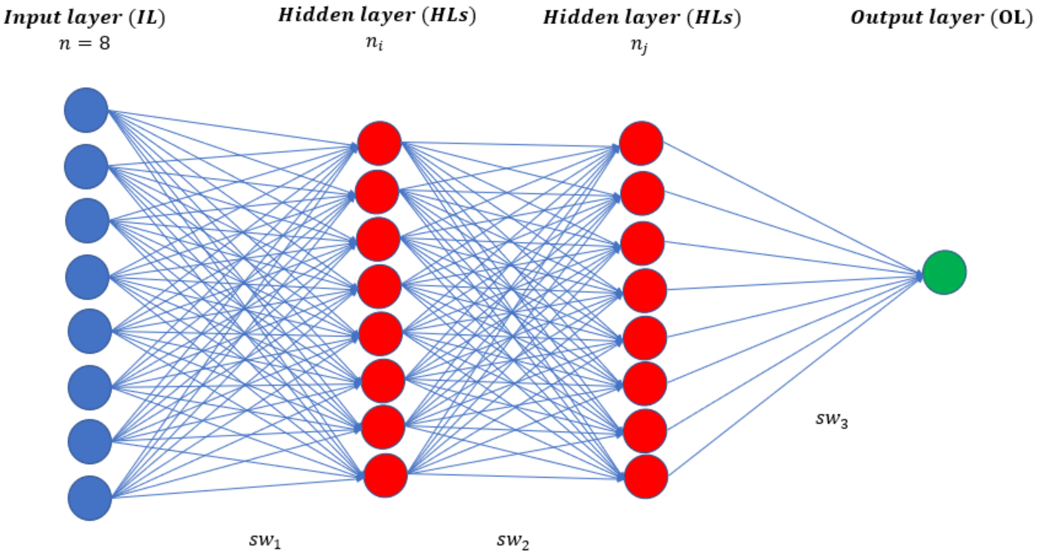


Figure 2.3: MLP neural network structure

In the training phase, an essential step involves the creation of an Random Matrix of Weights (RMW) to initialize the network's parameters. To prevent uniform responses across neurons, RMW is critical for introducing variability. A simple representation of the RMW is defined in Equation 2.11, providing an example of an illustration of a matrix representation consisting of three matrices, with n_i representing the number of nodes in the input layer, n_j for the first hidden layer, and n_k for the second hidden layer. By designating small random values to the weights, this randomized initiation assures that the network's learning process commences in a non-deterministic state, allowing the ANN to optimize and enhance these weights during training. The training process of an ANN model consists of four distinct stages, as outlined by Sagir and Sathasivan (2017). These stages include assigning small random values to the weights, a forward pass through the network, a backward pass for error computation, and updating the network's weights and biases.

$$\underbrace{\begin{pmatrix} a_{11} & a_{21} & \cdots & a_{n_j1} \\ a_{12} & a_{22} & \cdots & a_{n_j2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n_i} & a_{2n_i} & \cdots & a_{n_jn_i} \end{pmatrix}}_{rmw1} \underbrace{\begin{pmatrix} b_{11} & b_{21} & \cdots & b_{n_k1} \\ b_{12} & b_{22} & \cdots & b_{n_k2} \\ \vdots & \vdots & \ddots & \vdots \\ b_{1n_j} & b_{2n_j} & \cdots & b_{n_knj} \end{pmatrix}}_{rmw2} \underbrace{\begin{pmatrix} c_{11} \\ c_{12} \\ \vdots \\ c_{1n_k} \end{pmatrix}}_{rmw3} \tag{2.11}$$

Following the initialization and training processes, the role of AF assumes a critical role in an ANN. These functions allow the process of transforming the input signal of a neuron into an output signal, which is subsequently transmitted to the following network layer. Among the most commonly used AFs in ANNs are the Sigmoid Function (SF), Hyperbolic Tangent Function (HTF), and Rectified Linear Unit (RELU).

The SF maps any input value into a range between 0 and 1, as displayed in Equation (2.12), which makes it advantageous for models that require probability predictions. However, its limitations include the vanishing gradient problem, where gradients approach zero when input values are extremely low or high, impairing the process of learning.

$$SF(X) = \frac{1}{1 + e^{-x}} \tag{2.12}$$

On the other hand, the HTF maps inputs into a range between -1 and 1, as displayed in Equation (2.13), which provides a greater benefit than the SF in specific scenarios, such as when the data being processed inherently contains a great number of negative values or when the normalized data is centered around zero.

$$HTF(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.13}$$

Lastly, the RELU function has recently acquired considerable popularity. The RELU function is characterized by retaining its input's positive values while converting any negative input values to zero, as shown in Equation 2.14. The straightforwardness of RELU results in faster computing times. Additionally, this function addresses the vanishing gradient problem.

$$R(x) = \begin{Bmatrix} x & x > 0 \\ 0 & x <= 0 \end{Bmatrix} \tag{2.14}$$

The training of an MLP involves two main phases: the forward pass and the backward pass. In the forward pass, the input information is sent into the network and processed sequentially through each layer. In each layer, the ANN actively transforms the data by computing the weighted sum of its inputs and then applying an AF. This process is repeated layer by layer until the data reaches the final output layer when the network generates an output with its prediction.

In a distinct stage known as the backward pass or backpropagation, the ANN learns from its errors. The output coming out of the forward pass is compared against the objective outcome using a function loss function. The error can be defined as the difference between the predicted and actual outcomes. Different metrics, such as Mean Square Error (MSE) or Mean Absolute Error (MAE), actively measure this. Equations 2.15 and 2.16 indicate their slight variations, where $\hat{y_t}$ is the value of outcome predicted and y_t actual value at the moment t and n is the number of samples.

$$MSE = \frac{1}{n} \sum_{t=0}^{t=n-1} (y_t - \hat{y_t})^2 \tag{2.15}$$

$$MAE = \frac{1}{n} \sum_{t=0}^{t=n-1} |y_t - \hat{y_t}|^2 \tag{2.16}$$

The error is computed during backpropagation and is transmitted back through the network from the output layer to the input layer. At each layer, the gradients of the error with respect to the weights are calculated, providing information on how to modify the weights to reduce the error. Then, these gradients indicate the direction in which the weights should be adjusted to reduce the error. The weights are subsequently adjusted in accordance with a learning rate, determining the magnitude of these weight modifications.

The processes are iteratively repeated multiple times in accordance with the specified number of iterations. This involves exploring the entire dataset and adjusting the network's weights in order to minimize the prediction error. Ultimately, the performance of the model is evaluated using the Coefficient of Determination (R_2). The precise equations utilized for this evaluation are elaborated in 2.17.

$$R_2 = 1 - \frac{\sum_{t=0}^{n}(y_t - \hat{y}_t)^2}{\sum_{t=0}^{n}(y_t - \overline{y}_t)^2} \tag{2.17}$$

In the same manner that in Equation 2.15 and 2.16, where $\hat{y}_t$ represents the value of outcome predicted, and yt refers to the actual value at the same moment t.

Following the iterative processes outlined, DL takes this a step further by employing ANN with multiple layers. Deep architectures facilitate the extraction and manipulation of intricate features from raw data.

2.3.2 Recurrent neural networks

Recurrent Neural Network (RNN) is one type of ANN that deals with data that has sequential inputs. This architecture has been used to process speech and language data (Graves et al. 2013), especially predicting the next character and word in a given text (Y. Bengio 2000), large-scale acoustic modeling (Zia and Zahid 2018), but also can be used for more complex tasks (Lecun et al. 2015).

As mentioned previously, RNNs possess the unique capability to process sequential inputs, leveraging internal memory to handle these inputs effectively. Lecun et al. (2015) emphasize that RNNs maintain a sort of state vector within their hidden units, encapsulating information from earlier segments of the input sequence. Consequently, RNNs are adept at simultaneously managing past and present input data flows. These hidden units are crucial for the model because the RNN only takes one sequence at any given time.

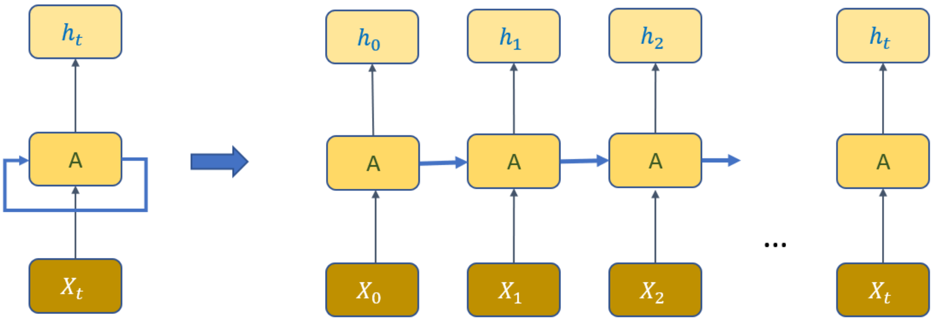


Figure 2.4: Recurrent neural network structure.

The general structure of the RNN is presented in Figure 2.4 and it is possible to see how an RNN can be mapped as a sequence of inputs. Where A represents a layer of the ANN containing vectors x with hidden state vectors h on time t.

2.3.2.1 Long-Short Term Memory

The advantage of an RNN is that it learns long-term dependencies over time. However, there are some error back-flow problems (Hochreiter and Schmidhuber 1997), showing difficulties in achieving a proper learning process; therefore, the information cannot be stored for a long time. Hochreiter and Schmidhuber (1997) introduced a novel solution proposing the Long-Short Term Memory (LSTM) model in order to correct this problem, augmenting the network with explicit memory and using hidden units to remember short-term and long-term values. The scheme of the main structure of an LSTM unit is shown in Figure 2.5.

The total number of units of the LSTM display to form a network with an input node and with input, output, and forget gates, where the gates will regulate the flow of the information. In the following equations (2.18- 2.23), it is possible to see the forward pass of the LSTM unit (Hochreiter and Schmidhuber 1997).

$$g_t = \widetilde{C} = \varphi(W_g x_t + U_g h_{t-1} + b_g) \tag{2.18}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{2.19}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{2.20}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{2.21}$$

$$C_t = g_t * i_t + f_t * C_{t-1} \tag{2.22}$$

$$h_t = o_t * \varphi(C_t) \tag{2.23}$$

Where:
i_t: represents the input gate activation vector,
f_t: is the forget gate activation vector,
o_t: the output gate activation vector,
C_t : the cell state vector,
h_t: the output vector of the LSTM unit,
σ : SF function,
b: biases for the respective gates,
W, U: weight matrices for the respective gates,
φ represents an HTF and $*$ is an element-wise product.

Though widely used, the LSTM model has led to some variations to mitigate its limitations. A notable example is the Gated Recurrent Unit (GRU) model, which specifically targets the vanishing and exploding gradient problems encountered in the traditional LSTM framework (Y. Bengio et al. 1994).

2.3.2.2 **Gated Recurrent Unit**

This model was introduced by Cho et al. (2014), proposing a novel ANN called RNN Encoder-Decoder consisting of two RNN. This model has fewer parameters and has proven to be competitive with other models like LSTM. Also, it is possible to observe that GRU as well as LSTM have gating units, which modulate a flow of

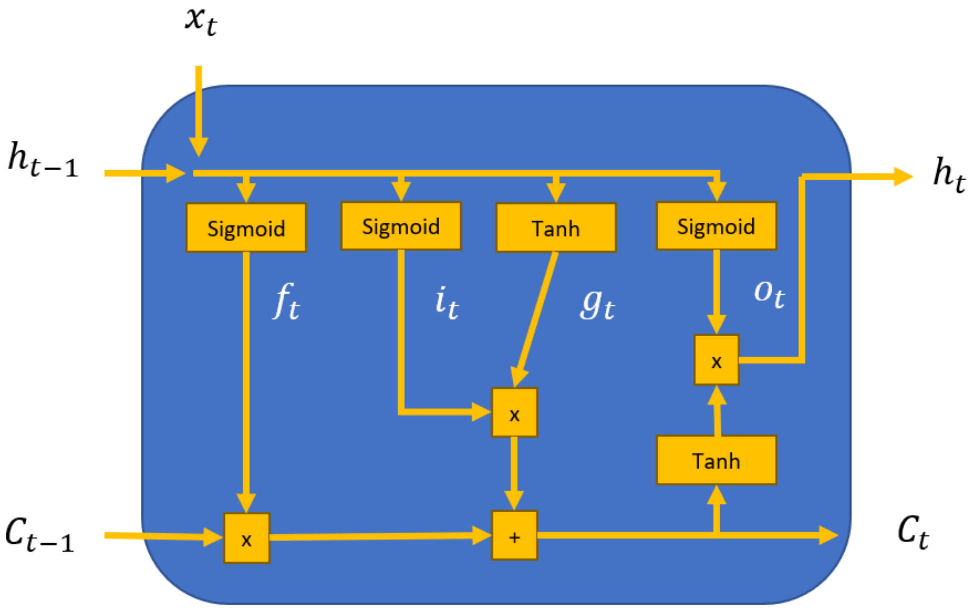


Figure 2.5: The general structure of the LSTM

information inside the unit cell; however, it does not have a separate memory cell (Chung et al. 2014).

According to Alom et al. (2019), this model is popular among people who work with RNN because the computational cost and the simplicity of GRU are better compared to others. The reduction in computational cost results from the lack of an output gate, enhancing the rate at which the model's computations are performed. (Pei et al. 2020).

This model has been successfully applied to many applications for pattern analysis where sequential data is used as input, including text recommendations with multi-task learning (Bansal et al. 2016), machine translation (X. Wang et al.

2016), speech recognition (Irie et al. 2016), among other applications for time series such as multivariate time series with missing values (Che et al. 2018).

Classical GRUs, as noted by Sak et al. (2014), often encounter the disadvantage of easily falling into local minima, particularly with small time series and complex time order. Additionally, Pei et al. (2020) highlight further limitations associated with GRUs concerning data handling, pointing out that GRUs struggle to discern the implicit information in time series data and that an imbalanced dataset can adversely impact the model's performance by affecting its convergence.

2.4 Clustering Techniques

Clustering is one of the so-called unsupervised learning methods in ML. The objective of clustering methods is descriptive rather than predictive, and it is used to group data instances into subsets to gather similar instances while different instances belong to different groups. To achieve efficient group separation, clustering methods rely on measuring the distance between data samples. These methods use mechanisms to evaluate the similarities within clusters and the differences between clusters (Mahdi et al. 2021).

This section describes the different cluster techniques. First, a brief introduction to Agglomerative Clustering (AGGLO) is given, followed by an overview of Balanced Iterative Reduction and Clustering by Hierarchy (BIRCH), and then K-Means Clustering Algorithms with Euclidean Distances (KNN-EUC) and K-Means Clustering Algorithms with Dynamic Temporal Warping (KNN-DTW). Finally, the MiniBatch Clustering (MNBT) and the Spectral Clustering (SPCT) techniques are examined.

2.4.1 Agglomerative clustering

AGGLO is one of the most common types of hierarchical clustering. This method initially assumes that each element represents an individual cluster. Then, in an iterative procedure, the two most similar groups are combined according to a criterion to measure distances between groups. The most common ways to measure the distance among groups are single, complete, average, and ward (Müllner 2013).

The agglomerative clustering results in a tree-based representation of all the unions called a dendrogram. After that, a cut of the dendrogram is performed to obtain the desired number of groups.

The formula of this technique as shown by Tokuda et al. (2022) to express the ward distance $D(w, v)$ between any two clusters u and v as,

$$D(u,v) = \sqrt{\frac{|v| + [s]}{T} D\left(v,s\right)^2 + \frac{|v| + [t]}{T} D\left(v,s\right)^2 - \frac{|v|}{T} D\left(v,s\right)^2} \tag{2.24}$$

Given a dataset that holds N elements, while s and t are the new pair of joined clusters, where $T = |v| + |s| + |t|$

Additionally, AGGLO has shown excellent performance in different tasks and has performed comparatively superior to other algorithms (Xiuge Wu et al. 2018).

2.4.2 Balanced Iterative Reduction and Clustering by Hierarchy

The BIRCH technique is another type of hierarchical clustering that can deal quickly and efficiently with large repositories. This algorithm creates a more diminutive representation of the large dataset that summarizes the large group, holding as much information as possible.

The BIRCH cluster is formed by creating a clustering feature tree composed of different nodes that are useful for calculating inter and intra distances of two clusters i and j. The centroid Euclidean distance can be calculated as

$$DO_{if} = \sqrt{\left(\left(\frac{LS_i}{N_i} - \frac{LS_j}{N_j}\right)^2\right)} \tag{2.25}$$

$$LS_i = \sum_{N}^{i=1} x_i \tag{2.26}$$

With N_i number of points, where x_i is a point in the cluster feature i. To calculate LS_j, the same procedure given for Equation 2.26 needs to be performed.

2.4.3 K-means clustering

K-means clustering is one of the most popular methods in ML to create clusters. It is a vector quantization method that seeks to split a dataset into k clusters. The goal is to set each observation in a specific group with the nearest mean to a cluster centroid. One crucial factor to consider when building a cluster is using a proper metric to measure the distance between all the data points to secure regularity and similitude among all the observations. There are two essential distance measures to consider when dealing with k-means: Euclidean Distance (EUC) and Dynamic Time Warping (DTW).

EUC is a distance function that is popularly used when using the k-means technique, which is based on measuring the distance of a line segment between two points in Euclidean space. This function is represented in Equation 2.27, as the square root of the sum of the distance between vectors x and y where the algorithm aims to split different sample sets into dissociated clusters.

$$\sqrt{\sum_{j=1}^{k} (x_j - y_j)^2} \tag{2.27}$$

However, one of the main problems with EUC is that it is not a normalized metric, and in high dimensional spaces, it tends to be augmented; therefore, a previous reduction technique is needed to avoid the curse of dimensionality. K-means clustering has been applied in finance to classify stock performance (Phongmekin and Jarumaneeroj 2018), hedging strategies (Sun et al. 2021), and financial time series forecasting (H. Jiang 2021; Lin et al. 2021).

Time series should receive special treatment for clustering because of the need to adapt the clustering techniques to time shifts. The DTW is a suitable measure to address this because it breaks the limitation of other one-to-one alignment metrics. This method attempts to discover all the possible paths using the dynamic programming technique, picking the one that renders the minimum distance between two series and building a matrix with the cumulative distances.

The DTW formula measures the distance between two time series x and y as,

$$DTW(x, y) = \min_{\pi} \sqrt{\sum_{(i,j)\in\pi} d\left(x_i, y_i\right)^2} \tag{2.28}$$

Where the path π satisfies additional boundary, continuity, and monotonicity constraints, since this algorithm has been developed primarily to deal with time series, applications are focused on classifying these datasets.

2.4.4 MiniBatch clustering

The MNBT clustering technique emerges as an innovative alternative to the traditional k-means algorithm, aimed at reducing its spatial and temporal cost. Despite the widespread adoption of k-means in various clustering tasks, its application often encounters challenges with memory efficiency, particularly when dealing with large datasets. To mitigate this, the MNBT technique introduces a novel approach that focuses on reducing memory consumption.

The core strategy of MNBT involves the creation of fixed-size, small random batches from the entire dataset. This process is conducted iteratively, where a new random sample is drawn to update the cluster centroids in each iteration. This approach significantly reduces memory usage compared to conventional k-means algorithms, which typically require access to the entire dataset for each update. By processing only a small portion of the data at a time, MNBT becomes particularly advantageous in scenarios with limited memory resources or where data cannot be entirely loaded into memory.

The computational efficiency of the MNBT clustering technique has attracted close attention, leading to its application in diverse big data projects, as indicated by Peng et al. (2018) and Tang and Fong (2018). MNBT has the ability to handle vast volumes of data swiftly and efficiently, making it a valuable tool in the realm of ML. Additionally, its utility extends to industrial applications, as highlighted by Messaoud et al. (2019), where the prioritization of real-time data processing and memory efficiency is of utmost importance. The integration of MNBT in these sectors demonstrates its versatility and effectiveness in addressing the practical challenges of clustering in complex and large-scale data environments.

2.4.5 Spectral clustering

The SPCT clustering technique is based on algebraic graph theory using information from the eigenvalues of matrices constructed from the graph. The resulting Laplacian matrix is defined as $L = D - A$, where A_{ij} measures the affinity between a data points x_i and x_j and D is

$$D = \sum_j A_{ij} \tag{2.29}$$

One of the most notable strengths of the SPCT is its handling of non-convex sample spaces. Unlike some traditional clustering methods that may prematurely converge to a local optimum, SPCT is more robust in exploring the data space. This characteristic is especially advantageous when the data displays intricate, non-linear patterns that traditional clustering methods struggle to capture accurately.

The effectiveness and robust theoretical foundations of SPCT have led to its widespread recognition across various sectors. Its ability to accurately analyze complex, non-linear data structures sets it apart from traditional clustering methods, making it particularly valuable. The adaptability of SPCT allows for customization to specific data characteristics, enhancing its utility in extracting meaningful insights from intricate datasets. This versatility and proven efficacy solidify SPCT's role as a crucial tool in the advancing field of data analysis and ML.

2.5 Dimensionality Reduction Techniques

In this section, different reduction techniques will be presented. Starting with an introduction to the AE and then going on to describe other commonly used reduction techniques.

2.5.1 Autoencoders

An AE is a form of an ANN composed of two elements: an encoder and a decoder. It can be defined as a learning circuit with the primary goal of converting inputs into outputs with the lowest error or the least possible amount of distortion

(Baldi 2011). The function of the encoder is to reduce the raw input multidimensional data into a lower dimension. The decoder takes this transformed data and reconstructs it as accurately as possible by minimizing the reconstruction error using the chain rule to backpropagate error derivatives. The general structure of an AE is shown in Figure 2.6.

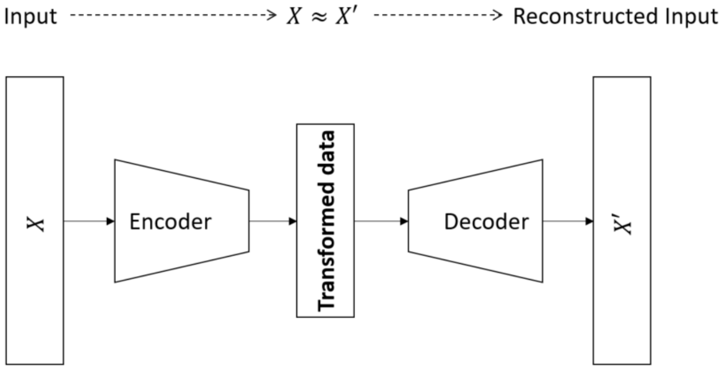


Figure 2.6: The structure of an AE

There are multiple forms of AE (Baldi 2011), such as those that are constructed using restricted Boltzmann machine (Le Roux and Yoshua Bengio 2008) and DL structures (Hinton and Salakhutdinov 2006) which change according to the complexity and arrangement of the data. Examples included fully connected deep AEs, convolutional neural networks AEs and RNN AEs, which can be applied to different tasks and fields such as feature extraction for econometrics (Gu et al. 2020a), fault diagnosis (R. Li et al. 2022), fraud detection (Fanai and Abbasimehr 2023), genetics (Eraslan et al. 2019), image processing (Masci et al. 2011), language translation (Sutskever et al. 2014), remote sensing (Dong et al. 2018), robotics (Park et al. 2018),and security (Al-Qatf et al. 2018) among others.

2.5.2 **Principal Component Analysis**

The Principal Component Analysis (PCA) is one of the most famous and widely used reduction techniques in multivariate statistical analysis. It can be applied to considerably diminish a large dataset's dimensions into a smaller one in an interpretable way while maintaining as much statistical information as possible and efficiently dealing with data that might have multicollinearity or missing values.

This reduction technique can be defined in different ways. However, two main descriptions are frequently used (Bishop 2006). The first one labels the PCA as an orthogonal projection of the information onto a principal subspace with a linear space of different dimensions to maximize the projected data. This method can also be described as the minimized squared distance between the project and the data points.

The maximum variance formulation considers a set of data $X = \{x_1, ..., x_N\}$, with m dimensions and $x_i \in \mathbb{R}^d$. The objective is to obtain a projection with a lower dimension d in $\mathbb{R}^m$, where $m < d$.

By considering $m = 1$, which needs a projection vector $u_i \in \mathbb{R}^d$, each point x_i projects to $u^T x_i$ and the variance of the projected data is defined as,

$$\frac{1}{N}\sum_{N}^{n=1}(u_1^T x_n - u_1^T \bar{x})^2 = u_1^T S u_1 \tag{2.30}$$

Where,

$$S = \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})(x_n - \bar{x})^T \tag{2.31}$$

$$\bar{x} = \frac{1}{N}\sum_{N}^{n=1} X_n \tag{2.32}$$

Then $u_1^T S u_1$, which is the projected variance that needs to be maximized with respect to u_1. The $\|u_1\| \to \infty$ maximization of the constraint u_1 also needs to be prevented with a normalization constrain $u_1^T u_1 = 1$.

This technique has been applied to reduce large datasets in multiple disciplines such as medicine (Krishn et al. 2014), robotics (D. Zhang et al. 2014), and statistical process control (Fuentes-García et al. 2018). In financial markets, this technique is applied to predict the stock market in lower dimensions using multiple assets (Ghorbani and Chong 2020; Zhong and Enke 2017) and to create models to select stocks (Yu et al. 2014). In addition, Pasini (2017) applied PCA to different subgroups of stocks efficiently deal with portfolio management; likewise, Narayan et al. (2014), using macroeconomic and institutional data from emerging markets using PCA, presented an equity asset pricing model to generate dynamic trading strategies.

The incremental Principal Component Analysis (iPCA) is a modification and substitution of the PCA to deal with large datasets that struggle with memory management. The iPCA uses an amount of memory independent of the number of input samples and constructs a low-rank approximation for the input data. This adaptation has the advantage of allowing sparse inputs and being more memory efficient than the traditional PCA model and has been used successfully in different fields such as medicine (Gupta and Mittal 2019), chemistry (Bouhlel et al. 2018), and biometric systems (Zhu et al. 2018).

2.5.3 **Fast Fourier Transform**

The Fast Fourier Transform (FFT), takes a signal from its original data domain to a different representation, decomposing a series of values into components of different frequencies and expressing it as a function that sums all the periodic components.

The FFT H_k of N points h_k is given by the formula

$$H_k = \sum_{n=0}^{N-1} \mathrm{e}^{-2\pi j \frac{kn}{N}} h_k \tag{2.33}$$

While the inverse transform is,

$$h_k = \frac{1}{N} \sum_{k=0}^{N-1} \mathrm{e}^{-2\pi j \frac{kn}{N}} H_n \tag{2.34}$$

as provided by Press et al. (2007).

Even though this method is effective, it is computationally expensive; however, the FFT algorithm can compute these transformations rapidly, reducing the complexity. As a result, the FFT has multiple applications and has been successfully applied to image processing (Vyas et al. 2018).

2.6 Reinforcement Learning

This section delves into the fundamental aspects of RL, establishes the foundational framework for comprehending its significance within the wider scope of AI and ML, examines its fundamental components and theoretical foundations, and follows an introduction to the fundamental ideas of RL.

2.6.1 Introduction to RL

RL is considered a subfield of ML, and it can be defined as the automatic learning of optimal decisions over time (Lapan 2018). Traditional ML methods depend on labeled data sets for supervised learning or, in the case of unsupervised learning, to find hidden patterns. RL, on the other hand, has an agent that interacts with a changing and dynamic environment and learns from the decisions made inside it by receiving rewards or punishments. This process involves learning from the outcomes of the actions rather than from labeled data, allowing the agent to improve its performance adaptively in changing environments.

By exploring and interacting with the environment, the agent learns to make optimal decisions and develops a strategy known as policy, which maps the states of its environment to the actions taken at different states. One distinguishing characteristic of RL is its focus on sequential decision-making. In this framework, the outcomes of actions impact immediate rewards and subsequent situations, thereby influencing future rewards.

The versatility of RL allows it to be effectively utilized in various problem domains where decision-making is imperative. This method is applicable in various scenarios wherein an autonomous entity must make a sequence of decisions without explicit instructions. The agent learns to achieve its goal through iterative interactions with its environment.

2.6.2 Reinforcement Learning Elements

The main elements of the RL framework are action, agent, environment, policy, reward, and state. An agent is a program that operates autonomously, perceives its environment, endures over time, adapts to change, and pursues goals (Russell and Norvig 2010). The environment represents all external elements to the agent, typically a simplified model of reality or a given scenario. On the other hand, actions are the agents' different operations in this environment. Each action the agent undertakes directly impacts its current state, influencing its interaction with the environment.

The state is a representation of the agent's current condition within the environment in which the agent is deployed, which affects how it makes decisions. By considering the agent's decisions, it is possible to obtain insights into the optimal actions and strategies that maximize rewards over time, leading to an effective learning process (Lapan 2018). By considering the agent's decisions, it is possible to acquire valuable insights into the optimal actions and strategies that maximize rewards over time and penalize wrong behaviors.

The reward is a value that the environment provides to evaluate the agent's performance, ranging from negative to positive. It is delivered at various frequencies, ranging from constant intervals to once in an agent's lifetime. The reward's primary role is to offer feedback and, as the name of this technique suggests, reinforce the agent's behavior appropriately. Finally, the agent's main objective is to maximize its cumulative reward through a series of actions.

Finally, a policy determines the behavior of a learning agent at any specific moment, guiding its actions based on the current situation. A policy is a function that maps an agent's perceived environmental states to corresponding actions across various states, ranging from basic lookups to intricate computations, crucially directing the agent's actions and responses (Sutton and Barto 2018).

2.6.3 Markov Decision Processes

The Markov Decision Process (MDP) gives a mathematical framework essential for formulating problems in RL. The aim is to provide a straightforward structure for learning from interaction to achieve a specific goal. This framework encompasses the intricacies of decision-making scenarios in which both stochastic elements and the deliberate actions of the decision-maker influence the results.

In the context of RL, the MDP framework is implemented with the objective of modeling the agent-environment interaction as shown in Figure 2.7, where the agent's goal is to find a policy that maximizes cumulative rewards over time. Since the success of the agent depends on a series of previous actions, the MDP allows the agent to systematically evaluate the consequences of its actions and adapt its strategy to improve performance in complex environments. It also asserts that future states are independent of past states, given the present state. By employing this approach, the decision-making process can be simplified, breaking down the overall problem into smaller, discrete, and manageable sub-problems.

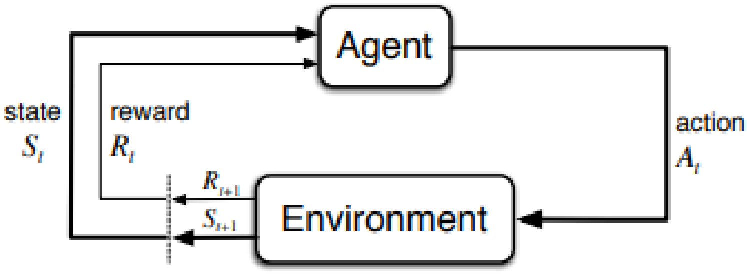


Figure 2.7: The general structure of the interaction of an agent-environment interaction in a MDP

Additionally, in an MDP-based RL setting, a notable challenge is the exploration-exploitation dilemma, which consists of a trade-off between exploring the environment in search of new knowledge and exploiting previously established knowledge.

Even though a given agent must explore and learn about unvisited states and different actions, engaging in excessive exploration can lead to inefficiency. On the other hand, exploitation involves utilizing the agent's existing knowledge to make decisions that yield the highest immediate reward. Therefore, the challenge is to find the balance between exploration and exploitation, ensuring that with proper exploration, there are no missing opportunities and the agent's accumulated knowledge is utilized efficiently through exploitation.

2.7 Explainability

This chapter focuses on Explainable Models in ML and AI, emphasizing the balance between algorithmic accuracy and the need for transparency.

2.7.1 Explainable Machine Learning models

Even though ML and RL techniques are expanding in different sectors and disciplines and are gaining popularity due to their strong performance, stakeholders are concerned about the opacity of these methods (Langer et al. 2021). Therefore, to satisfy the need for transparent models, XAI has emerged as an area of multidisciplinary research aimed at creating artificial systems that humans can read and understand (Dikmen and Burns 2022).

While certain models in data analysis and ML are inherently simple and easily interpretable, there are notable examples of such transparent models. These include linear regressions, decision trees, and rule-based algorithms. These models are widely accepted due to their inherent transparency and the ease with which their decision-making processes can be understood. This makes them particularly useful in scenarios where it is crucial to understand the rationale behind predictions. However, it is not always possible to rely on transparent models because they may not be suitable for all tasks, and some other models, like DL, might have better precision (A. Kim et al. 2020). Furthermore, models based on tree structures are not good at extrapolating (Staden et al. 2022), which limits their effectiveness in predicting outcomes beyond the range of the training data. This inability to generalize to unseen data can be a significant drawback in dynamic environments or situations where data constantly evolves.

Consequently, scholars must test and deal with complex challenges when selecting the appropriate model for a given task. These challenges include assessing the trade-off between model complexity and interpretability. When it is impossible to have a transparent algorithm, a different methodology needs to be conducted to elucidate the decisions. This added procedure is commonly known as a post-hoc model (Bastos and Matos 2022), which aims to deliver comprehensible information about the predictions given by the established model at different points.

The post-hoc explainable methods can be rather specific to a particular model, usually DL algorithms, or it can be an agnostic method built to deal with a wide range of algorithms. The non-specific models are intended to collect information about the prediction process and can be divided into three groups (Barredo Arrieta et al. 2020). The first group is the explanation by simplification and local explanations established on rule extraction techniques. The most famous model that falls into this category is the Local Interpretable Model-Agnostic Explanations, also known as LIME (Ribeiro et al. 2016), which builds the explanations by perturbing and sampling the original data set and then fits a subrogate model that mirrors the unknown global model on that sample. The resulting coefficients allow an explication of the prediction to be obtained.

The second collection of post-hoc agnostic models is based on feature relevance explanations by evaluating the importance and influence of each variable in the prediction process of a given model. One successful application is the SHAP, initially introduced by S. Lundberg and S.-I. Lee (2017) to interpret different complex ML models based on coalitional game theory (Shapley et al. 1953), and is considered the state-of-the-art technique for model-agnostic interpretation (Marcílio and Eler 2020).

The concept in terms of game theory is to create an approach where members of a given game should obtain a reward proportional to their marginal contribution to that game. Following this analogy, we can calculate an expected reward for a feature's marginal contribution to a specific model. SHAP values will represent this contribution by averaging the permutation of each variable on the conditional expectation in a hypothetical model prediction. Through this iterative process, it is possible to compute the impact that a single variable has on the general performance of the model when it is added and combined with the rest of the variables (S. M. Lundberg et al. 2020). The SHAP values are formulated in equation 2.35.

$$\varphi_i(f, x) = \sum_{h \in H} \frac{1}{|n|!} \left[f_x(y_i^h \cup i) - f_x(y_i^h) \right] \tag{2.35}$$

Where n represents the number of features, H is the set containing all the possible feature permutations, and y_i^h is the group of all features that precede feature i in permutation h.

2.7.2 Explainable Reinforcement Learning

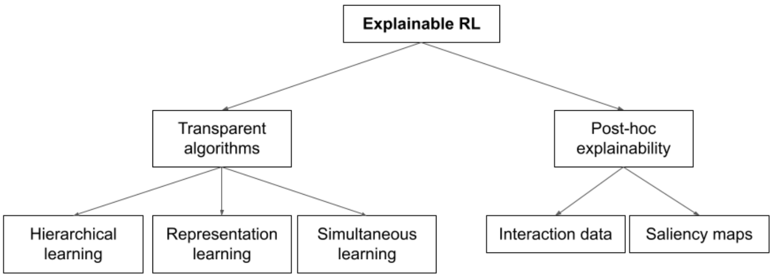
Despite recent advances in developing explanatory models for intelligent systems, the RL domain has not yet been fully explored in this context (Krajna et al. 2022). While RL has shown considerable promise as an efficient technique in various applications, its inherent complex nature presents unique explainability challenges. As a result, the demand for understandable and transparent systems has started to drive research in this area.

This growing interest is not only due to the potential of RL in solving complex problems but also originates from the demand for ethical and responsible AI, where understanding the decision-making process of models is crucial. As such, exploring explanatory models within RL is becoming a significant focus in the field, aiming to bridge the gap between advanced algorithmic performance and the need for clarity and accountability in AI-driven decisions.

Heuillet et al. (2021) categorize recent studies in Explainable Reinforcement Learning (XRL) based on the primary ideas outlined by Barredo Arrieta et al. (2020). These studies are divided into two main groups: transparent algorithms and post-hoc explainability. Transparent algorithms are divided into three subgroups: hierarchical, representation, and simultaneous learning. The post-hoc explainability group, on the other hand, is segmented into models based on interaction data and saliency maps. This categorization of various XRL approaches is illustrated in Figure 2.8.

Transparent algorithms can be explained by themselves by examining their architecture without applying an external model to understand their behavior. Examining the sub-groups shows that hierarchical learning involves a high-level agent learning an interpretable environment representation. In contrast, a low-level agent splits the main goal into a set of sub-goals and learns which is optimal. An efficient approach to filtering which sub-goals have been operated on is the Hindsight Experience Replay developed by Andrychowicz et al. (2017), which allows RL algorithms to learn policies from sparse rewards efficiently (Fang et al. 2019). In a different way, representation learning uses the explanation of a meaningful low-dimensional representation of the state (Lesort et al. 2018). In contrast, simultaneous learning is based on applying a method that allows the learning of the policy and the development of explanatory models via casual models (Madumal et al. 2020) and reward decomposition (Juozapaitis et al. 2019).

Another way to explain a model that is not transparent is by using post-hoc methods and creating a different algorithm for explainability, which is used after the training and testing of the main model. When investigating the sub-groups, we observe that interaction data allows the agent to extract characteristics of interest



.

Figure 2.8: The general scheme of XRL is based on the categorization made by (Heuillet et al. 2021)

by scrutinizing the history of interactions with the environment, capturing the key features in a given task (Sequeira and Gervasio 2020). In the same vein, saliency maps have been created to generate XRL, creating a visual interpretation using a topographically assembled map to display the importance of each pixel in a given image (Greydanus et al. 2018; Guo et al. 2021).

3

State of the art

The state-of-the-art chapter examines the latest advancements and applications of AI in the financial sector, mainly focusing on asset pricing and portfolio construction. This section of the thesis focuses on recent developments in the field, demonstrating how novel algorithms are transforming and improving traditional financial models and strategies.

Section 3.1 begins with an in-depth exploration of ML algorithms in Finance. This section examines how various ML models are being utilized to improve financial analysis. The paper also examines the development of these techniques and their increasing importance in financial decision-making.

In Section 3.2, the focus shifts to ANN and their effectiveness in predicting financial time series. This part explores the architecture of ANN, including DL models and how they are applied to financial analysis.

Section 3.3 addresses the role of clustering and dimensionality reduction techniques in Finance. Following this, Section 3.4 explores the application of RL in Finance. It discusses the latest advancements in RL algorithms and their application in Finance, underscoring the potential of RL to adapt to and learn from dynamic market conditions.

Lastly, Section 3.5 covers the topic of explainable models in Finance. It examines the growing need for transparency and interpretability in AI and ML models, particularly in a sector where decision-making carries significant economic implications. This section highlights the importance of XAI in ensuring compliance, building trust, and facilitating understanding among stakeholders in the financial industry.

Each section collectively presents a view of the latest advancements and applications of AI and ML in Finance, highlighting the dynamic interplay between technological innovation and financial expertise.

3.1 Machine Learning Algorithms in Finance: Trends and Applications

The Finance sector has significantly transformed in recent years, propelled by the development of ML algorithms. This shift has brought a period marked by data-driven decision-making and improved analytical skills. Integrating ML into financial practices has refined traditional analytical approaches and introduced innovative methods to tackle complex financial challenges. These algorithms have become essential across various domains, including predictive analytics for stock market trends, risk management, and the emerging field of algorithmic trading. The rise of ML in Finance is fueled by an exponential increase in data generation, availability, and storage capabilities, coupled with advanced computational power. This progress has enabled the analysis of vast and intricate datasets, showing insights that were difficult to obtain before, thereby transforming the financial domain.

Within the ML domain, a significant emphasis is placed on applying the CatBoost algorithm in the financial sector, where it is utilized for various purposes. These include predicting the repayment ability for home loans, anticipating loan defaults (Xia et al. 2020), detecting fraud in financial transactions, forecasting corporate failures (Jabeur et al. 2021a), assessing credit scores, and analyzing trends in stock indexes and oil price fluctuations (Jabeur et al. 2021b).

Additionally, within the finance sector, the **CatBoost!** (**CatBoost!**) algorithm has been instrumental in classifying various tasks, particularly those related to identifying and classifying diverse types of risks (X. Ma and Lv 2019), emphasizing its important role in improving the precision of risk management in financial operations.

The utilization of DTR is notably prevalent in the domain of Bitcoin (BTC) price prediction, examining various determinants (Rathan et al. 2019). This approach entails monitoring daily BTC price movements, covering data points such as open, high, low, and closing prices up to the current day. The primary aim is to assess and compare the predictive accuracy of diverse ML algorithms, particularly on decision trees and regression models.

Other robust implementations include variants of the SVM algorithm, such as the fine-tuned version and the ESVR algorithm, which is particularly notable for its effectiveness in regression tasks. It is designed to predict continuous values, making it highly relevant in financial contexts. It has been used as a novel ML approach for stock forecasting that leverages a grid search method for optimal kernel function selection and parameter tuning, significantly increasing accuracy while reducing time and memory usage and preventing data overfitting. It has been used as a novel ML approach for stock forecasting that leverages a grid search method for optimal kernel function selection and parameter tuning, significantly increasing accuracy while reducing time and memory usage and preventing data overfitting. The method's effectiveness is demonstrated through its application to diverse, large-scale stock datasets, showing superior accuracy and efficiency in stock prediction compared to existing methods (Dash et al. 2021). It has also been used in hybrid frameworks such as combining feature-weighted SVM and K-Nearest Neighbor (KNN) methodologies for predicting stock market indices, focusing on maximizing profits and minimizing investment risks.

Additionally, other authors demonstrate that the research elaborates on a sophisticated approach, assigning differential weights to features based on their classification relevance by computing the information gained. Y. Chen and Hao (2017) integrates a feature-weighted KNN technique for predicting future stock market indices, capitalizing on historical data to ascertain weighted nearest neighbors. The practical application of this model is exemplified through its successful performance in predicting short, medium, and long term indices in two major Chinese stock markets: the Shanghai and Shenzhen stock exchanges.

3.2 Predicting Financial Time Series with Artificial Neural Networks

The ANNs are a bio-inspired computing system based on many processors called neurons. They are connected to each other and are activated following different types of AF, which are triggered according to specific weights and biases that change after a sequence of feed-forward optimization steps to minimize an error function obtained by calculating the difference between predicted and expected outputs. The ANN has the advantage of nonlinearity and has been used in different fields, having a big impact in image processing, speech recognition, and language processing (Cao et al. 2018). They have also been successfully used in business applications, mainly in financial distress and bankruptcy analysis, stock price predictions, and credit scoring (Tkáč and Verner 2016).

Due to incorporating innovative and automated technologies into the workings of banks and other financial organizations, ANNs have become an increasingly important component of the decision-making process. Due to their consistency and objectivity, these models have been rapidly adopted, reducing human subjectivity and the likelihood of erroneous assumptions. In addition, ANNs enable investors and institutions to construct new powerful models by collecting information from historical data and enhancing already built models. Investors and institutions can gain a deeper comprehension of the data produced by the stock market and develop more robust models, hence improving the degree to which their forecasts are accurate.

The ANNs have also been used in the analysis of credit scoring, where a model is created to help lenders categorize the risk of borrowers and then decide whether to approve or reject certain loans for a specific client. Over the years, this kind of decisions in banks and financial institutions has been calculated by new and automated systems due to their consistency and objectivity (Marqués et al. 2012), eliminating human bias that a statistical model could have by making assumptions based on prior knowledge, letting investors create new powerful models by extracting information from past observations but also allowing institutions track their performance for further improvements of their models.

Using a MLP neural network, other scholars (Zhao et al. 2015) achieved better credit scoring than previous methods with a classification accuracy ranging from 83 to 87 %. Also, ANN and SVM techniques appear to be robust tools to predict cases of bankruptcy (Alaka et al. 2018).

3.2.1 Stock Market predictions

Different ANN applications to predict the stock market can be found in the literature. The use of ANN to predict the return of the Japanese Nikkei 225 index by using a hybrid approach based on genetic algorithm and simulated annealing (Qiu et al. 2016). Additionally, Pyo et al. (2017) predicted a stock exchange index, building three hypotheses to predict the daily closing prices of the Korea Stock Price Index 2000 by using an ANN and two SVMs models with ten different indicators and information obtained from Google trends. Those indicators used as input were the same used by other authors (Kara et al. 2011) in a three-layered feed-forward ANN and an SVM model to predict the direction of the Istanbul Stock Exchange National 100 index found that the 75.74 % performing prediction for ANN was significantly better than the SVM.

By using the four and nine prior days as inputs, Moghaddam et al. (2016) created an ANN using different training and transfer functions at different network structures, concluding that there is no distinct difference by using four or nine prior days as inputs. Even though the performance of the ANNs models built by these authors is better than previous works, the size of training and testing datasets (99 days in total) could be insufficient to obtain a good representation of different financial and economic cycles that affect the stock market.

A contrast between ANNs and classical statistical techniques are presented by Sagir and Sathasivan (2017) by comparing the use of the commonly used multiple linear regression with an ANN model predicting the Malaysian Stock Market Index by using three variables; volume at the day close, the financial times stock exchange bursa Malaysia index and market valuation of 63 days. The ANN was more accurate than the multiple linear regression model in this research with a coefficient of determination of 0.9256.

Another comparison with classical statistical techniques is performed by Adebiyi et al. (2014), where a commonly used method in time series forecasting, called autoregressive integrated moving average, which is a form of Box-Jenkins model and comparing it with a ANN model to predict the price of a single stock using 5680 observations. The authors concluded that the ANN model is better than the autoregressive model because it has a higher forecasting accuracy but is not statistically significant.

Guresen et al. (2011) used a DAN2 dynamic ANN to predict stock market data. They compared this model with a MLP and two hybrid models using GARCH. An example of integrating metaheuristics to predict financial data with ANN was performed by Gocken et al. (2016); in this article, an ANN is hybridized with a Genetic Algorithm and Harmony Search. In their model, the inputs were technical indicators and the previously mentioned algorithms were used to make an input selection to reduce the complexity of variable selection, concluding that hybrid ANN can be successfully used to forecast the stock market price movement.

Enke and Thawornwong (2005) introduced an information gain technique to evaluate the prediction of stock market return using data mining and ANN for level estimation and classification with macroeconomic variables as inputs. The ANN model was more accurate and showed more consistency than a linear regression forecast. Also, the authors created a trading simulation where the strategies guided by ANN models generated higher profits compared to other strategies that had the same risk exposure.

Another example of the hybrid model using ANN to predict financial time series is the work by H. Y. Kim and Won (2018), combining LSTM and GARCH models. These authors compared the performance of this model with other existing GARCH methodologies, moving averages, and other ANN models, concluding that LSTM single models are capable of learning temporal patterns of time-series data effectively with fewer prediction errors than deep feed-forward network-based integrated models.

Also, Shao et al. (2019) implemented an improved LSTM to predict the price of nickel, a commodity that become essential in world markets. These authors used the London Metal Exchange closing prices as a sample and proposed an improved model by integrating the swarm optimization algorithm to their LSTM. In a recent review of the applications of DL in the stock market, W. Jiang (2021) explains that scholars driven by the swift advancement and escalating application of DL models for forecasting the stock markets created over one hundred related papers since the year 2019 to provide an overview of the most recent progress. Similarly, Bustos and Pomares-Quimbaya (2020) performed a systematic review of predicting stock markets, mentioning that using ML algorithms such as ensemble models and DL has recently become more popular. Ensemble models have demonstrated strong predictive abilities and, in some cases, performed better than other popular classification ML algorithms and ANN. However, the systematic review also explained that DL models have not yet outperformed traditional ML models, and this is probably because the data used to train the algorithms is not adequate enough to form an accurate forecast.

3.3 Clustering and Dimensionality Reduction: Enhancing Financial Data Analysis

The importance of clustering methods applied to time series is gaining traction due to the increase in technological applications that require, generate, and store information in real-time. They play an essential role in different domains such as the internet of things (Yao et al. 2019; Yin et al. 2022), autonomous vehicles (W. Wang et al. 2020), medicine (Kaplan Berkaya et al. 2018), genetics (Oyelade et al. 2016) and finance (Cai et al. 2016; K. Kim and Song 2020), where data usually is complex, non-stationary and high dimensional with temporal dependencies.

Consequently, clustering large time series with high dimensional data sets is complex, and correct dimensionality reduction and efficient extraction of the important features are essential for clustering because they inhibit efficient examination of the data and interfere with the proper use of clustering techniques (Njah et al. 2021). For example, a review by Aghabozorgi et al. (2015) found that many authors focused on representing time series data in a lower dimension to be consistent with conventional clustering algorithms. On the other hand, Ismail Fawaz et al. (2019) in an exhaustive review of DL for time series classification, initially stated that the common attribute shared by those algorithms that outperform previous implementations is a transformation phase to convert the data series into a new feature space. This attribute raises the need for efficient transformation techniques since the most widely disseminated dimensionality-reduction method is the PCA; however, this technique is inherently linear and can only model linear interdependencies between the data set features (Qaraei et al. 2021).

3.4 Reinforcement Learning in Finance

As mentioned in Chapter 2, in recent years, the emergence of ML as a powerful and disruptive technology enabling computers to learn and predict financial events using multiple data sources (Kamruzzaman et al. 2022) has gained popularity. Since ML techniques bring significant advantages in analyzing complex phenomena, this has generated a growing interest in the financial literature (Goodell et al. 2021).

As a result, different approaches have also been used to address the construction of portfolios to maximize the distribution of resources using DL (Y. Ma et al. 2021), RL (Syu et al. 2020; Z. Zhang et al. 2020) and different optimization techniques such as particle swarm optimization (Thakkar and Chaudhari 2021), genetic algorithms (Cheong et al. 2017), hybrid techniques (Paiva et al. 2019) among others (Ge et al. 2014; Lai et al. 2020; J. Li et al. 2017; Perrin and Roncalli 2020).

Furthermore, ML, with broad applications in finance, has emerged as a prominent tool in portfolio management. These techniques can be used in different forms. The first approach deals with model-based methods which predict prices but do not deal directly with trading and thus require another method to execute the trade, which may be based on a system of rules derived from human experience or metaheuristics. Within this group, many authors (Heaton et al. 2017; Kamalov 2020; Ozbayoglu et al. 2020) rely heavily on the use of ANN to predict prices. On the other hand, a model-free approach creates a trading strategy without the need for an explicit price prediction process, where an ANN can create a portfolio vector by mapping variables from the assets (Betancourt and W.-H. Chen 2021) and typically uses the RL method to train the network.

Different authors have used RL in finance to create portfolios and trading systems with widespread acceptance in the financial community (Millea 2021; Ozbayoglu et al. 2020). Early work (John Moody et al. 1998) showed the feasibility of these techniques by presenting empirical results in controlled experiments to demonstrate the effectiveness of optimization methods in creating portfolios focusing on a custom Sharpe ratio; also, in the same manner, different attempts have been applied to automated trading systems using adaptive RL (M. Dempster and Leemans 2006; M. A. H. Dempster and Romahi 2002).

Various DL methods allowed researchers to train ANN techniques that enhance the use of RL and the use of popular algorithms such as Policy Gradients (PG), Advantage Actor-Critic (A2C), and Deep Q-Learning (DQL) (Mnih et al. 2015; Mnih et al. 2016; Sutton et al. 1999; Watkins and Dayan 1992) have emerged as important advances in the discovery of investment policies in the stock and futures markets (Deng et al. 2017; Dixon et al. 2020; J. Moody and Saffell 2001; Pendharkar and Cusatis 2018) as well as in the cryptocurrency market (Sattarov et al. 2020). Also, Jeong and H. Y. Kim (2019) implemented a DQL RL model to improve financial trading decisions by adjusting the number of shares in the portfolio and implementing transfer learning to deal with insufficient data and avoid overfitting. Likewise, Zarkias et al. (2019) executed a trading strategy applying DQL to predict the Euro-Dollar exchange rates using a trailing stop strategy. Similarly, Y. Li et al. (2019) used deep RL to design a trading strategy by using a LSTM network and extending the use of DQL to Asynchronous Advantage Actor-critic for better adaptation to trading market conditions.

Z. Zhang et al. (2020) designed an RL trading strategy testing A2C, DQL, and PG against traditional time-series momentum strategies using a dataset from 2011 to 2019 on 50 liquid futures contracts, showing that the RL models beat the classical models. Similarly, Yang et al. (2020) created an ensemble trading strategy using three based actor-critic algorithms. Brim (2020) utilized a model for trading pairs of cointegrated financial assets using DQL and double DQL, creating a strategy with positive returns. Additionally, Fengqian and Chao (2020) introduced k-line theory clustered learning features into the model to characterize candlesticks as a way to generalize price movements over time and then used DL to create an online control of the parameters in the environment. In a similar way, Taghian et al. (2022) trained an RL model with candlestick data to learn trading rules. Similarly, Hirchoua et al. (2021) developed an approach based on a rule-based policy for different agents trading against each other in a virtual environment.

Recently, some modifications to the RL techniques have been proposed to create financial portfolios; for example, Aboussalah and C.-G. Lee (2020) incorporated portfolio constraints and continuous actions on numerous assets into the proposed trading RL framework. Xing Wu et al. (2020) proposed a trading strategy using PG and DQL methods and recurrent ANN that outperformed other popular strategies. Further research conducted by Betancourt and W.-H. Chen (2021) created a novel portfolio management solution using RL, with a dynamic number of multiple assets, attempting to learn the optimal holding in order to minimize the transaction costs. Similarly, Lim et al. (2022) presented a dynamic rebalancing of portfolios using RL and LSTM networks.

A novel approach was proposed by Théate and Ernst (2021) by training the RL agents on generated artificial trajectories from a partial set of historical data from the stock market. Another original methodology in the construction of trading RL agents in the stock market is the utilization of multi-agents to deal with collective intelligence. Abdelkawy et al. (2021) created a multi-stock model based on synchronous multi-agents to deal with an extensive historical dataset. Meanwhile, Shavandi and Khedmati (2022) developed a multi-agent RL application that is tested on a historical dataset from an important currency pair, outperforming single agents based on different return and risk performance measures in different time frames.

3.5 Explainable Artificial Intelligence

The field of XAI has emerged as an active and fast-growing interdisciplinary research area, driven by a convergence of factors, including the rising pressure from stakeholders who demand transparency from AI systems and the growing use of AI in financial applications. The core objective of XAI research is to develop systems capable of making accurate predictions and decisions and explaining their reasoning to humans. (Dikmen and Burns 2022; Langer et al. 2021).

In previous publications, XAI has been used in finance with an increasing demand from banks and supervisory authorities to guarantee accountability, fairness, and transparency (Kuiper et al. 2022). In recent work, different ML applications have been developed with explainability; for example, Ohana et al. (2021) applied an XAI to analyze stock market crashes, while Sachan et al. (2020) implemented a decision support system that explains the procedure that determines the approval or rejection of a loan. Likewise, XAI has been applied to explain stock market trend prediction (Mandeep et al. 2022), auditing (C. Zhang et al. 2022), credit scoring (El Qadi et al. 2022), money laundry detection (Kute et al. 2021) and other financial applications (Hoepner et al. 2021).

Despite recent advances in the development of explanatory models for intelligent models, RL has not yet been fully explored in this field (Krajna et al. 2022), and research has begun to grow due to the current demand and rise of RL as an efficient technique. (Heuillet et al. 2021) classify the latest XRL studies according to the main ideas of (Barredo Arrieta et al. 2020), which are presented in two primary groups; transparent algorithms and post-hoc explainability. The initial group can subsequently be subdivided into three subgroups: hierarchical, representation, and simultaneous learning, while the second group is segmented into interaction data and saliency maps models, as mentioned in Figure 2.8, it is possible to categorize different XRLs levels of explainability.

As mentioned in Section 2.7, numerous studies use XRL from efficient approach to filtering sub-goals in different RL contexts, from Hindsight Experience Replay (Andrychowicz et al. 2017) to RL algorithms learning policies from sparse rewards efficiently (Fang et al. 2019). Representation learning uses meaningful low-dimensional state representations (Lesort et al. 2018). In contrast, simultaneous learning involves learning policies and developing explanatory models via causal (Madumal et al. 2020) models and reward decomposition (Juozapaitis et al. 2019).

Additionally, attention mechanisms are gaining significance due to their ability to provide detailed understanding. Studies such as Amirshahi and Lahmiri (2023) highlight the effectiveness of hybrid models incorporating attention layers for predicting cryptocurrency prices using an ensemble of language models to analyze social media sentiment. Similarly, X. Zhang et al. (2023) proposed an attention-based CNN-BiLSTM hybrid model for credit risk prediction in real estate enterprises, showing high accuracy. Meanwhile, Gradzki and Wójcik (2023) has demonstrated that Transformer ANN, featuring attention mechanisms, exhibits predictive prowess in intraday Forex trading, surpassing even the ResNet-LSTM benchmark models. These converging lines of evidence underscore the utility and versatility of attention mechanisms in financial modeling and prediction in the financial markets.

Additionally, works have shown the capabilities of XRL by using post-hoc methods and examining the chronology of interactions with the environment, capturing the key features in a given task (Sequeira and Gervasio 2020), and by generating saliency maps (Greydanus et al. 2018; Guo et al. 2021).

Despite the contributions from previous research, it is unclear, especially within the context of cryptocurrencies, how to ”unblack box” and explain results from ML algorithms. So far, we observe contradictory findings when various ML algorithms are applied for trading prediction. On the one hand, some studies (e.g., Z. Chen et al. 2020; Ozbayoglu et al. 2020) assert that ML is a powerful and popular forecasting tool for the financial markets. On the other hand, other studies (e.g., (Bussmann et al. 2020; Jabeur et al. 2021b)) posit that ML without explanations behaving like a 'black box' is not appropriate for regulated financial services, and investors avoid it due to its lack of trust. Because of these contradictions and limited explanations of how AI algorithms decide cryptocurrency trading, there is limited trust among investors and regulators. With the growing adoption of ML algorithms in technical trading prediction, the need for an ensemble method to achieve explainability of the results of cryptocurrency trading AI algorithms is urgent. To address existing research and practice limitations, we seek to offer a model to examine the application of XAI in predicting technical trading of cryptocurrency assets during financial uncertainty.

4

Neural Network-Based Forecasting of Financial Market Time Series

The financial market, a dynamic and complex domain, generates an immense volume of data daily, presenting significant challenges in decision-making. This chapter explores developing and testing multiples ANNs to predict the closing price of the Índice Bursátil Español 35 (IBEX); by using over a decade of historical data, the study explores five different neural network architectures, including a MLP, a simple RNN, a GRU network, and two LSTM networks.

The motivation subsection 4.1 focuses on the need for intelligent systems with ANN, underlining their role predicting prices in complex and non-stationary financial data.

In the methodology subsection 4.2, there is an extensive discussion on implementing five unique neural network architectures. This section explains the study's methodology, covering data collection, model development, and the choice of ANN.

The subsection 4.3 scrutinizes each model's results and efficacy in forecasting the IBEX closing price. It evaluates their accuracy and identifies potential limitations, critically analyzing their performance.

Finally, the discussion subsection 4.4 provides a detailed discussion of the implications of the study's findings by including a comparison of the model's predictive accuracy and computational efficiency, with a particular focus on the outstanding performance of the LSTM network.

This chapter aims to showcase the practical use and importance of ANN in predicting financial market trends, thus contributing valuable insights to the field.

4.1 Motivation

ANN stands out as a highly effective tool for stock market prediction, a complex intersection of Finance and Computer Science, where traditional methods like technical and fundamental analysis frequently stumble. Unlike linear models such as ARIMA and GARCH, or even conventional ML techniques ANNs possess a unique capability to learn and model non-linear relationships. This feature is significant in financial markets where data patterns are non-linear and influenced by various factors such as economic indicators, market sentiment, and global events. ANN, can assimilate and analyze vast amounts of data, extracting patterns often invisible to other analytical models. This capability allows them to provide more accurate predictions of stock market trends and movements (W. Jiang 2021).

Furthermore, the emergence of Big Data coupled with advancements in computational capabilities, notably through Graphical Processing Units, has markedly improved the performance of ANNs in stock market forecasting. DL models, as a branch of ANNs, have shown superior efficacy in various tasks, notably in time series forecasting, which is key for stock market analysis. In contrast to conventional ML, these models are adept at processing extensive unstructured data, such as market news and social media content, thereby providing deeper insights into market trends. Their increasing application in financial forecasting is attributed to enhanced computational power and greater access to data, signaling a transformative phase in financial time series prediction (Sezer et al. 2020). As such, these models can leverage the vast and varied data prevalent in financial markets, enclosing everything from price history to news articles and social media dynamics, thus facilitating a thorough analysis.

The ability of ANNs to learn and adapt to new data continuously makes them useful in the ever-evolving domain of financial markets. Asset management companies and investment banks are increasingly using AI and DL for stock trading, extending their application beyond academia. Considering these advancements and the factors outlined earlier in this thesis, the motivation for this section is to investigate different structures of ANN for predicting financial time series. This investigation explores the value and practical consequences of ANNs in real-world financial applications.

4.2 Methodology

This study utilized Python programming language to develop and test five distinct models using the IBEX as the data set, covering 3309 observations from January 4, 2010, to December 02, 2022. This dataset provided over a decade of daily closing prices and volume values, reflecting various market trends. The data was split into 80 % for model training and 20 % for testing, allowing validation of unseen data to measure forecasting accuracy.

The study focuses on recent periods for training and testing, adding novelty. Extended training benefits ANNs, as they require substantial historical data to learn from varied market scenarios, potentially improving predictions. The Anderson-Darling test, applied to closing prices, rejected the normal distribution hypothesis in the data sample.

For this analysis, the closing prices and volume values were normalized to a scale between 0 and 1 ($0 < x < 1$) for compatibility with the activation functions in the ANN. This normalization was crucial, given that the price values in the dataset did not adhere to a normal distribution. To assess the accuracy of the different models employed in this study, three key performance metrics were used: MSE, MAE, and the R_2. These metrics provided a comprehensive evaluation of the model's predictive performance.

The common input layer for the ANN architectures used in this study is presented in Eq. 4.1:

$$\begin{aligned} X_{input}(t) = f[&v(t-4), v(t-3), v(t-2), v(t-1), \\ &c(t-4), c(t-3), c(t-2), c(t-1)] \end{aligned} \tag{4.1}$$

Where $c(t)$ is the function for the closing price and $v(t)$ is the volume value at a given time t; also, in this research, we defined three variables to predict: the closing price is t and then the closing price of the next day, quoted as $t + 1$, and then two days after the closing price $t + 2$. All models were separately trained for different forecasting horizons $(t0, t1, t2)$. Despite the separate training for each forecast horizon, the same training and test sets were consistently used across all scenarios. This approach ensured a uniform benchmark for comparison while allowing the individual performance of each model under different forecasting horizons to be observed.

In this research, the function for the closing price at any given time t is denoted as $c(t)$, and the volume value at that time is represented by $v(t)$. The study focused on predicting three specific variables: the current day's closing price (t), the next day's closing price ($t + 1$), and the closing price two days later ($t + 2$). Each model underwent separate training for these distinct forecasting horizons $(t0, t1, t2)$. Despite this individual training for each time horizon, the same training and test datasets were uniformly applied across all models and scenarios. This methodical approach provided a consistent benchmark for comparison and allowed the individual performance of each model to be evaluated under different forecasting conditions.

4.3 Results

This research evaluates multiple ANN architectures for forecasting the IBEX stock index, focusing on how different node counts and structures affect accuracy. Five ANN architectures were compared using MSE, MAE, and R^2 performance metrics and their training times to assess computational efficiency.

The study implemented GRU, LSTM, simple RNN, and MLP models with two layers but varying node counts and an ARIMA model for benchmarking. A second

LSTM model with a final 128-node linear layer was included, known as LSTM II. Results showed that LSTM II had the lowest MSE (0.0006) and MAE (0.0174) and the highest R^2 (0.96) at t, but its training time was considerably longer (4.81 minutes) than other RNN methods, likely due to more parameters and the additional layer.

Model	t	Specifications	MSE	MAE	R2	Running Time
GRU	0	1 hidden nodes + 1 layer	0.0057	0.0639	0.68	2.67
LSTM I	0	1 hidden nodes + 1 layer	0.0019	0.0360	0.89	2.25
LSTM II	0	1 hidden nodes + 1 layer + 128	**0.0006**	**0.0174**	**0.96**	4.81
MLP	0	4 - 64 layers	0.0075	0.0765	0.58	5.84
RNN	0	1 hidden nodes + 1 layer	0.0141	0.0896	0.21	**1.75**
ARIMA	0	ARIMA (2,1,2)	0.1136	0.3089	<0	0.81
GRU	1	1 hidden nodes + 1 layer	0.0101	0.0686	0.44	1.44
LSTM I	1	1 hidden nodes + 1 layer	0.0331	0.1315	< 0	1.48
LSTM II	1	1 hidden nodes + 1 layer + 128	**0.0006**	**0.0172**	**0.96**	2.63
MLP	1	8 - 128 layers	0.0346	0.1776	< 0	13.15
RNN	1	1 hidden nodes + 1 layer	0.0437	0.1728	< 0	1.04
ARIMA	1	ARIMA (2,1,2)	0.1136	0.3089	<0	**0.79**
GRU	2	1 hidden nodes + 1 layer	0.0265	0.1274	< 0	1.42
LSTM I	2	1 hidden nodes + 1 layer	0.0166	0.1008	0.07	1.50
LSTM II	2	1 hidden nodes + 1 layer + 128	**0.0011**	**0.0223**	**0.94**	2.69
MLP	2	2 - 32 layers	0.0248	0.1323	< 0	2.07
RNN	2	1 hidden nodes + 1 layer	0.0298	0.1468	< 0	1.04
ARIMA	2	ARIMA (2,1,2)	0.1136	0.3089	<0	**0.76**

Table 4.1: The best results of different prediction techniques using multiple ANNs, including the running times of each method in seconds.

The LSTM model's capacity to identify non-linear financial time series data patterns was greatly improved by adding a final layer. In contrast, the original LSTM I model, lacking this extra layer, exhibited an MSE of 0.0019, MAE of 0.0360, and an R^2 of 0.89, still outperforming the GRU, simple RNN, and MLP models in predicting prices at t, $t+1$, and $t+2$ as detailed in Table 4.1. These results underscore the superiority of LSTM models in stock price prediction, highlighting

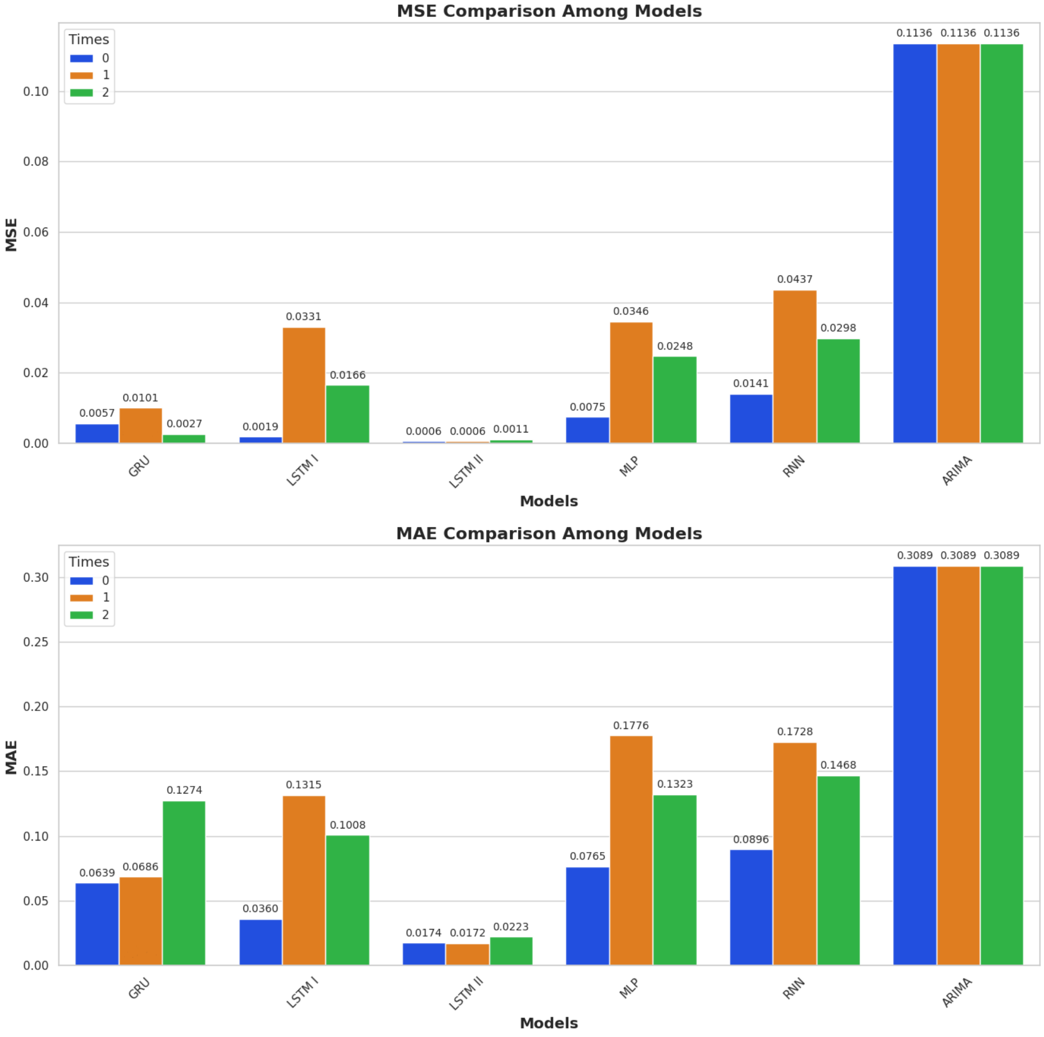


Figure 4.1: Evaluations measured by MSE and MAE.

the added value of incorporating a final linear layer to boost the LSTM II model's accuracy.

In contrast, models excluding LSTM demonstrated less proficiency in accurately forecasting stock prices, particularly for predictions one and two days ahead. For instance, the GRU and MLP models, at $t + 2$, reported higher error rates (MSE of 0.0265 and 0.0248, respectively) and negative R^2 values, indicating less predictive power compared to the LSTM variants. The simple RNN model showed an even higher MSE of 0.0298 and a negative R^2. These outcomes reinforce the conclusion that LSTM models, especially LSTM II, are more adept and reliable for short-term stock market predictions than their counterparts.

Furthermore, the ARIMA model's performance, as shown in Figure 4.1, with MSE and a MAE values of 0.1136 and 0.3089 respectively, along with consistently negative R^2 values across all time intervals, underscores its relative inadequacy in encapsulating the intricate dynamics of financial time series. This is particularly evident when compared to the capabilities of ANN approaches.

The present study emphasizes the significance of model architecture and layer complexity in predicting financial time series. The enhanced performance of the LSTM models, particularly LSTM II with an additional linear layer, demonstrates the significant potential of advanced DL techniques in accurately predicting stock market trends, providing crucial insights for future research and practical applications in financial analysis.

Additionally, in Figure 4.2, the comparative performance of various models against actual values is visually represented. To clarify the results, we designed a chart where the actual values serve as a baseline, depicted as a straight line. Against this baseline, we plotted the performance of each model, showcasing the deviation of their predictions from the actual values. Notably, the second LSTM model, labeled LSTM II, demonstrates the least variance, aligning more closely with the baseline. This indicates its superior accuracy in forecasting, in contrast to the other models, which exhibit greater errors and a more pronounced deviation from the baseline. This graphical representation underscores the enhanced predictive capability of the LSTM II model over its counterparts.

Figure 4.2: Evaluations of different ANN predicting time t, t+1, and t+2.

4.4 Conclusions

Using various ANNs for financial time series forecasting, particularly the IBEX, has shown high predictive accuracy with low errors. The LSTM model outperformed others, especially with an added final linear layer. Its effectiveness lies in its structure and training iterations. LSTM's ability to capture long-term dependencies, a critical factor for short-term forecasts, sets it apart from other models like GRU, MLP, basic RNN, and traditional methods like ARIMA.

This research emphasizes the remarkable capabilities of ANNs, with a special focus on LSTMs, in the realm of stock market forecasting. The findings of this study reveal that LSTMs exhibit superior proficiency in analyzing financial trends. This advancement is particularly significant given the stock market data's complexity and dynamic nature. The LSTM's ability to effectively capture and learn from the temporal dependencies in such data underscores its potential as a powerful tool for predictive analysis in the financial sector. This positions LSTMs as a pivotal innovation in ML, opening new pathways for more accurate and reliable financial predictions.

While LSTM demonstrate a high R_2 in forecasting stock market index closing prices, the task of determining the most effective ANN structure remains a complex challenge. This complexity arises because reductions in prediction error do not always follow a linear pattern when adjusting the ANN's architecture. Consequently, future research efforts should be directed towards creating heuristics to streamline the training process and optimize the optimization of ANN architectures.

However, the practical implementation of these models in financial trading presents intricate challenges for investors. They must navigate a multifaceted task that includes not just the accuracy of price predictions but also the assessment and management of risks associated with trading positions and the determination of appropriate trading sizes. Therefore, robust risk management protocols and comprehensive backtesting must underpin any trading strategy leveraging these advanced models.

Moreover, the interconnectedness of financial markets implies that various external factors could influence market prices and introduce biases. Hence, employing a more comprehensive feature selection could potentially enhance the model's predictive accuracy. Future research might focus on expanding the model's range of critical variables. Additionally, exploring different ANN architectures that can accurately predict closing values and developing a trading system capable of reliably forecasting financial markets is a promising avenue.

It's also important to reiterate the significance of a robust dataset and a thorough, reliable testing phase to effectively implement and utilize DL methods in financial time series analysis. This aspect becomes especially crucial in periods of high market uncertainty, where the complexity and scale of financial markets increase. Moreover, given the inherently volatile nature of market dynamics, it's essential to acknowledge that predicted outcomes may not always align precisely with future results. This uncertainty further underscores the need for careful model construction and evaluation in financial market forecasting.

Selecting the right data set is crucial in financial modeling, as using inappropriate data can lead to models that predict past patterns but fail to forecast future values. Each asset has a unique historical price record, making financial prediction models prone to overfitting. Future research should address this by testing models with data from multiple assets and synthetic datasets to identify biases from using a single data set. Utilizing DL models like generative adversarial networks can create numerous hypothetical data sets, enhancing model robustness.

With the economy's unpredictability and rapid technological changes, accurate financial forecasting has become essential for public and private investors and businesses. AI and ML can enhance asset allocation decisions by providing more accurate risk assessments, improving financial efficiency and growth. AI's potential to transform financial market forecasting and regulation lies in its ability to provide sophisticated, transparent prediction models, aiding market participants and regulators in making better decisions and increasing their chances of success.

5

Autoencoder-Enhanced Clustering: A Dimensionality Reduction Approach to Financial Time Series

The field of finance, characterized by its intricate and multidimensional data, presents unique challenges in data analysis and interpretation. This chapter explores a novel approach to clustering financial time series data, focusing on improving predictive models in the financial sector. It introduces an advanced clustering framework integrating AE to achieve a more compressed yet informative representation of complex financial time series data.

The motivation subsection 5.1 underlines the need for compression with clustering techniques in financial markets. This highlights the difficulties presented by the inherent complexity of the market and the need for improved techniques to classify financial time series effectively.

The methodology subsection 5.2 examines the implementation of the proposed clustering framework. It discusses the use of AE for dimensionality reduction and the application of various clustering algorithms. This section details the comprehensive methodology used in the study, including data sourcing and model development, focusing on major financial indices like the IBEX, Cotation Assistée en Continu 40 (CAC), Deutscher Aktienindex 30 (DAX), Standard and Poor's 500 (SPX), and Financial Times Stock Exchange 100 (UKX).

Subsection 5.3 examines the results, evaluating the effectiveness of the AE-based framework in clustering financial time series. The analysis includes comparing different dimensionality reduction and clustering techniques and evaluating their performance in different financial time series categories.

Finally, the subsection 5.4 offers an in-depth discussion of the implications of the study's findings. It explores how the advanced clustering methodology can lead to more accurate financial predictive models, providing valuable investment strategy optimization and risk management enhancement insights.

This chapter aims to contribute significantly to financial technology by demonstrating the potential of advanced clustering techniques, mainly using AE, in the context of financial time series data. It underscores the importance of these methodologies in enhancing the accuracy and efficiency of financial market analyses.

5.1 Motivation

Due to its dynamic and complex nature, the financial sector requires sophisticated analytical tools to decipher patterns from the market trends effectively. However, traditional methods cannot always handle the multidimensional and volatile nature of financial time series data, which poses a significant challenge for investment and risk managers who rely heavily on accurate and complex data assessments to make decisions.

The goal of this research lies in the requirement to improve the accuracy and reliability of clustering financial time series data. The existing gap in efficient clustering methods highlights the necessity for developing advanced techniques capable of handling the complexity and multi-dimensionality inherent in financial market data. The inherently multidimensional nature of data poses significant challenges for practical analysis and hinders the optimal application of clustering techniques (Njah et al. 2021).

Additionally, it has been observed that numerous researchers aim to represent time series data in reduced dimensionality, which aligns with the requirements of traditional clustering algorithms (Aghabozorgi et al. 2015). Conversely, a detailed review on DL for time series classification (Ismail Fawaz et al. 2019) has underscored that algorithms outperforming their predecessors frequently incorporate a transformation phase. This phase is crucial for transforming the data series into a new feature space, enhancing the algorithms' effectiveness and flexibility in processing time series data. This research seeks to address these challenges through the innovative use of AE to resolve the complexities of clustering intricate financial data.

AE, known for their efficiency in dimensionality reduction and feature extraction, offers a potential direction for transforming complex financial data into a more manageable and insightful form. By employing these DL tools, the methodology seeks to enhance the clustering of financial time series data, moving beyond traditional methods to adopt a more sophisticated, data-driven approach.

The decision to evaluate this approach across multiple dimensionality reduction and clustering algorithms and apply it to key financial indices like IBEX, CAC, DAX, SPX, and UKX, further emphasizes the work's practical and implications that might have in the financial world. It underlines a commitment to develop a theoretical framework and test its efficacy in real-world scenarios.

Ultimately, this research aims to bridge the gap in financial analytical methodologies by introducing a more refined and accurate clustering method. This advancement will provide insights for developing more precise investment strategies and robust risk management techniques, contributing significantly to financial analytics and decision-making.

5.2 Methodology

This paper employs AEs as a method for dimensionality reduction to assess their influence on six clustering techniques: AGGLO, BIRCH, KNN-EUC, KNN-DTW, MNBT, and SPCT[1]. The data used in this research includes intraday data from IBEX, CAC, DAX, SPX, and UKX. This research aims to compress the financial time series from four dimensions (open, high, low, and Closing Prices (CP)) into a single price value, thereby avoiding the curse of dimensionality in clustering. Another aim is to identify an optimal method that enables handling multidimensional financial data with traditional clustering techniques.

The method presented in this paper aims to transform the original 10-second price series into more extensive time frames, specifically 1, 5, 15, and 30-minute intervals. This approach involves comparing the effectiveness of the AE with other leading dimensionality reduction techniques, such as PCA, iPCA, and FFT. Additionally, the study aims to determine whether these reduction techniques can effectively minimize data dimensions without significant information loss. This assessment will be conducted by contrasting the reduced data with the CP, representing the input data without any reduction. The overall framework of this proposed methodology is depicted in Figure 5.1.

The time series data was sourced from the Bloomberg terminal, characterized by a granularity of 10 seconds, the finest resolution available among the indices mentioned earlier in Section 5.1. The selected dataset encompasses 01 September 2020 to 16 March 2021.

This study totaled 1,874,423 observations across five indices. Empty values were filled with the nearest previous value based on the assumption that the last known price represents the current value without new data. Each data set was then normalized. An LSTM structure with a Sigmoid AF was used for both the encoding and decoding networks in the AE. The input size for the encoder network was set as a three-dimensional matrix, reflecting the total number of observations, the total intraday prices per day, and the number of prices for each instrument.

The encoder's role in this framework is to create a compact data representation formatted as a three-dimensional matrix. The dimensions of this matrix correspond to the number of observations, the frequency of the data, and the value of a single price point. Conversely, the decoder processes the encoder's compressed data as

---

[1] In this study, ward AGGLO, BIRCH, MNBT, and SPCT techniques are applied within the proposed framework, implemented using the Scikit-learn Pedregosa et al. 2011, while Tslearn Tavenard et al. 2020 is used for the KNN-DTW and KNN-EUC algorithms.

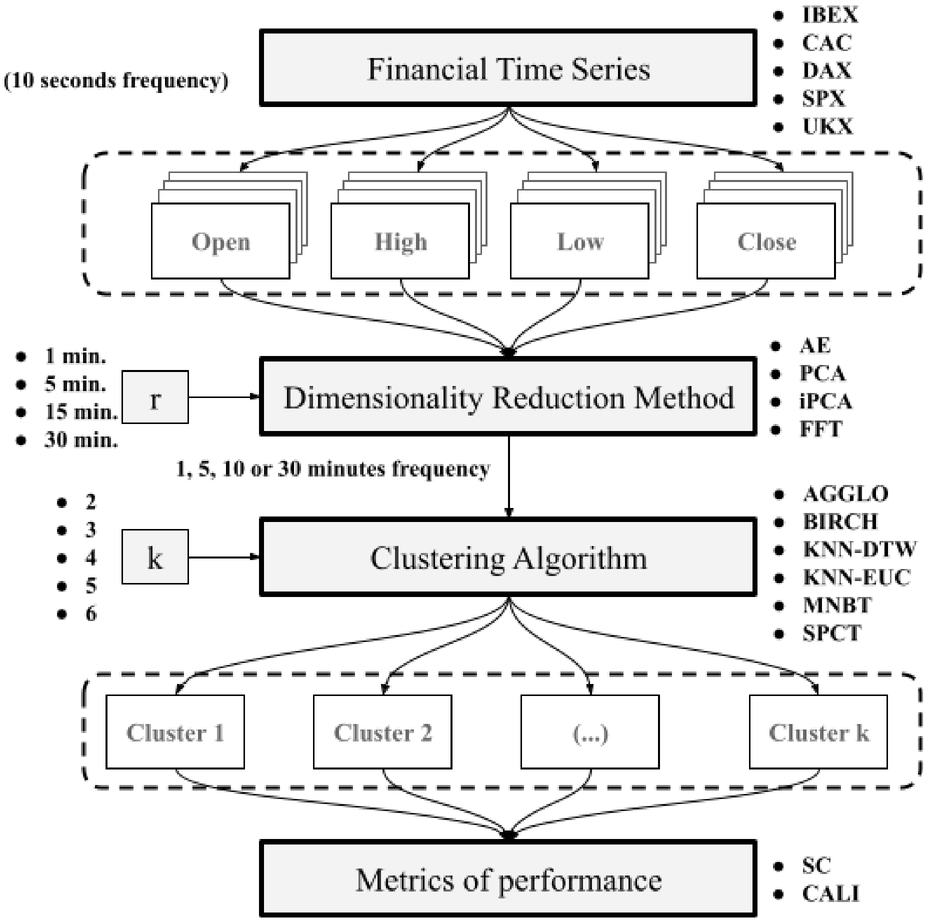


Figure 5.1: Evaluations measured by MSE and MAE.

its input. Its output is a matrix with dimensions identical to the encoder's input matrix. After decompressing the received information, the decoder compares it against the encoder's initial input. This comparison is crucial for assessing the model's accuracy, which is done using the MSE function.

After the data undergoes a compression process, it is then transferred to a suite of clustering algorithms, namely AGGLO, BIRCH, KNN-EUC, KNN-DTW, MNBT, and SPCT. Subsequently, the efficacy of these clustering techniques is evaluated by applying two distinct performance metrics. These metrics are important for estimating the effectiveness of each algorithm in accurately grouping the compressed data. By utilizing these performance measures, the study aims to determine each clustering method's relative strengths and weaknesses, providing insights into their suitability for specific data sets or analytical requirements.

The first metric used in this study is the SilHouette coefficient (SH), which measures how well-defined a cluster is. This metric is defined as follows:

$$SH = \frac{r - s}{max(s, r)} \tag{5.1}$$

In the SH metric, s represents the mean distances from a sample to all other points within the same class, and r pertains to the distances to points in the nearest cluster. It is important to note that the SH values range between -1 and 1. A negative SH value indicates an inaccurate clustering process, while a zero value suggests that clusters overlap. Conversely, a SH value of 1 signifies that the clusters are highly dense and well-separated. Thus, the greater the SH value, the more effective the model is in clustering.

The second metric, referred to as CALInski and Harabasz (CALI), quantifies the average similarity between clusters. It measures the dispersion ratio within clusters to that between clusters, aiming to maximize this ratio in scenarios where clusters are compact and well-separated. The CALI metric is defined in the following manner:

$$CALI = \frac{n_d - nk}{nk - 1} \cdot \frac{tr(B_g)}{tr(W_g)} \tag{5.2}$$

n_d reflects the total elements in a dataset, while d and nk are the number of clusters. While $tr(W_g)$ and $tr(B_g)$ represents the trace of the within-cluster and between-group dispersion matrices, which are denoted as:

$$W_g = \sum_{j=1}^{g} \sum_{x \epsilon M_j} (x - m_j)(x - m_j)^T \tag{5.3}$$

$$B_g = \sum_{j=1}^{g} n_j (m_j - m_D)(m_j - m_D)^T \tag{5.4}$$

The set of points in cluster j is defined as M_j and with cluster j and its center m_j. Where the center of D is defined as m_d and n_j is the number of observations in cluster j. A high CALI score implies a better performance of the cluster technique with well-separated and dense clusters.

The set of points belonging to cluster j is defined as M_j, with m_j representing the center of cluster j. The overall center of the dataset D is denoted as m_d, and n_j indicates the number of observations in cluster j. A high CALI score signifies the effectiveness of the clustering technique, characterized by well-separated and densely populated clusters.

5.3 Results

The experimentation was carried out to validate the performance of the presented framework. We then analyzed the impact of AE on six clustering techniques described below in Table 5.1, where each configuration has $k = \{2, 3, 4, 5, 6\}$ number of clusters. In addition, we compared this execution with CP, PCA, iPCA, and FFT, which are state-of-the-art reduction techniques previously explained in Section 3.3.

Each dimensionality reduction technique was applied to the following datasets: IBEX, CAC, DAX, SPX, and UKX. The original 10-second price series from these datasets was converted into a compressed time series of 1, 5, 15, and 30 minutes. Subsequently, these transformed datasets were processed through six distinct clustering algorithms. To ensure the robustness of the results, each configuration underwent four separate runs, each with a different random seed number. This process was replicated across five varying numbers of clusters and four epochs, culminating in a comprehensive total of 9600 experiments for thorough analysis.

Table 5.2 shows the best results in terms of CALI and SH for different indices and time frames, indicating variation in results based on the measurement metric.

Cluster Technique	**Parameters**
AGGLO	Linkage = Ward Affinity = Euclidean
BIRCH	Threshold = 0.01 Branching factor = 50
KNN-DTW	Distance = DTW
KNN-EUC	Distance = Euclidean
MNBT	Initialization = k-means++
SPCT	Eigen solver = arpack

Table 5.1: Table with the principal configurations of the different reduction techniques

Using CALI, the AE consistently yields the highest values across most scenarios, except for the UKX clustering in a 1-minute series, where FFT performs better.

Further analysis of CALI results reveals that the top-performing clustering techniques vary by index and time frame without a consistent pattern linked to specific algorithms. However, a uniform trend is observed in the ideal number of clusters (k), where six consistently yield the highest CALI values.

The analysis of results based on the SH metric reveals a lack of correlation with the best combinations identified by CALI. The top-performing reduction and cluster techniques differ for a specific index when comparing CALI and SH metrics. The only exception is observed in the 30-minute compressed representation of the UKX index, where the highest values for both CALI and SH were achieved using the AE and KNN-EUC clustering technique. This indicates that the effectiveness of reduction and clustering methods varies significantly depending on the evaluation metric used.

Analyzing performance based on the SH metric, Table 5.2 shows that in thirteen out of twenty combinations, PCA achieves the highest value, followed by AE. This variation seems to be influenced by the type of index: for clustering IBEX, CAC, and DAX data, PCA yields the highest SH values, whereas for SPX and UKX, AE performs best.

Moreover, the most effective clustering techniques vary depending on the data type they compress. For example, the highest SH values in IBEX clusters are

Exchange	Time Frame	Red_Cali	Cluster_Cali	k	CALI Score	Red_SH	Cluster_SH	k	SH Score
IBEX	1 minute	AE	MNBT	6	1673.31	PCA	AGGLO	2	0.8355
IBEX	5 minutes	AE	KNN-EUC	6	1871.78	PCA	AGGLO	2	0.8365
IBEX	15 minutes	AE	SPCT	6	1726.64	PCA	AGGLO	2	0.8388
IBEX	30 minutes	AE	KNN-EUC	6	1728.58	PCA	AGGLO	2	0.8418
CAC	1 minute	AE	SPCT	6	1163.60	PCA	KNN-EUC	2	0.8564
CAC	5 minutes	AE	MNBT	6	1182.68	PCA	KNN-EUC	2	0.8572
CAC	15 minutes	AE	SPCT	6	1185.16	PCA	KNN-EUC	2	0.8593
CAC	30 minutes	AE	KNN-EUC	6	1203.66	PCA	KNN-EUC	2	0.8620
DAX	1 minute	AE	MNBT	6	682.53	PCA	MNBT	2	0.7611
DAX	5 minutes	AE	MNBT	6	678.69	PCA	MNBT	2	0.7622
DAX	15 minutes	AE	SPCT	6	676.54	PCA	SPCT	2	0.7651
DAX	30 minutes	AE	MNBT	6	677.89	PCA	KNN-EUC	2	0.7736
SPX	1 minute	AE	SPCT	6	992.24	AE	MNBT	2	0.6591
SPX	5 minutes	AE	SPCT	6	1002.10	AE	KNN-EUC	2	0.6626
SPX	15 minutes	AE	SPCT	6	1008.60	AE	KNN-EUC	2	0.6628
SPX	30 minutes	AE	MNBT	6	1002.67	AE	KNN-EUC	2	0.6640
UKX	1 minute	FF	KNN-EUC	6	1167.09	AE	KNN-EUC	2	0.7549
UKX	5 minutes	AE	MNBT	6	1185.57	PCA	SPCT	2	0.7682
UKX	15 minutes	AE	SPCT	6	1213.87	AE	KNN-EUC	2	0.7605
UKX	30 minutes	AE	KNN-EUC	6	1232.10	AE	KNN-EUC	2	0.7680

Table 5.2: Comparative outcomes of various clustering methods on compressed Multi-Index Data: Optimal CALI and SH scores for each condition.

	AE	**CP**	**FFT**	**PCA**	**iPCA**
AE	-	0.89, 0.0, 0.11	0.95, 0.0, 0.05	0.96, 0.0, 0.04	0.87, 0.0, 0.13
CP	0.24, 0.0, 0.76	-	0.95, 0.0, 0.05	0.95, 0.0, 0.05	0.17, 0.0, 0.83
FFT	0.0, 0.0, 1.0	0.0, 0.0, 1.0	-	0.59, 0.0, 0.41	0.05, 0.0, 0.95
PCA	0.0, 0.0, 1.0	0.0, 0.0, 1.0	0.31, 0.0, 0.69	-	0.05, 0.0, 0.95
iPCA	0.25, 0.0, 0.75	0.7, 0.0, 0.3	1.0, 0.0, 0.0	1.0, 0.0, 0.0	-

Table 5.3: Comparative analysis of reduction techniques - Frequency of superior, equivalent, and inferior performance in percentage terms, with CALI scores below the diagonal and SH scores above

achieved using AGGLO, while KNN-EUC is predominant in clustering CAC data. Similar to CALI results, there is a consistent ideal number of clusters (k), which is two. This suggests that the choice of dimensionality reduction and clustering techniques and the number of clusters should be tailored to the specific characteristics of the financial index being analyzed.

	AGGLO	BIRCH	KNN-DTW	KNN-EUC	MNBT	SPCT
AGGLO	-	0.09, 0.83, 0.07	0.48, 0.1, 0.42	0.4, 0.09, 0.51	0.39, 0.09, 0.52	0.39, 0.1, 0.51
BIRCH	0.04, 0.83, 0.13	-	0.47, 0.1, 0.43	0.39, 0.1, 0.51	0.38, 0.1, 0.52	0.38, 0.1, 0.51
KNN-DTW	0.44, 0.1, 0.46	0.45, 0.1, 0.45	-	0.18, 0.3, 0.51	0.33, 0.14, 0.53	0.34, 0.14, 0.52
KNN-EUC	0.7, 0.09, 0.2	0.71, 0.1, 0.19	0.61, 0.3, 0.08	-	0.42, 0.18, 0.4	0.44, 0.18, 0.39
MNBT	0.67, 0.09, 0.23	0.68, 0.1, 0.22	0.6, 0.14, 0.26	0.31, 0.18, 0.51	-	0.43, 0.16, 0.41
SPCT	0.66, 0.1, 0.24	0.67, 0.1, 0.23	0.59, 0.14, 0.27	0.32, 0.18, 0.5	0.42, 0.16, 0.43	-

Table 5.4: Comparative analysis of clustering techniques - Frequency of outperforming, matching, and underperforming in percentage terms.

Table 5.3 is designed to compare different clusters, quantifying how often each reduction technique outperforms, matches, or underperforms others in percentage terms. Below the diagonal, which represents comparisons of the same techniques, the results are based on CALI. Above the diagonal, the results are measured using SH.

Similarly, Table 5.3 compares different clustering techniques in the same format. In both tables, the first number indicates the percentage by which a technique performed better, the same, or worse than others. This structure provides a comprehensive overview of the effectiveness of various reduction and clustering techniques when evaluated by different metrics.

Reviewing Table 5.3, it is evident that the AE significantly outperforms other reduction techniques, particularly in terms of CALI. The second-best technique is iPCA, which fares better than CP, FFT, and PCA. CP, in turn, generally outperforms FFT and PCA in overall percentage terms.

Regarding SH results, the trend is similar: AE again leads with the best outcomes, albeit by a smaller margin compared to CALI results. iPCA remains the second most effective technique, followed by CP. These findings highlight the relative strengths of different reduction techniques in processing and analyzing financial data.

Table 5.4 compares different clustering techniques to the comparison of reduction techniques in Table 5.3. AGGLO exhibits inconsistent performance: it is equivalent to BIRCH 83% of the time in terms of CALI and SH, but does not show notable improvement over other methods.

BIRCH performs on par with KNN-DTW but is less effective than KNN-EUC, MNBT, and SPCT when evaluated using CALI, and similarly to a lesser extent with SH. Notably, the KNN-EUC technique surpasses all others in both CALI and SH metrics. MNBT ranks as the second-best clustering method, followed by SPCT, which outperforms AGGLO, BIRCH, and KNN-DTW in both metrics. This analysis underscores the varied effectiveness of different clustering techniques in handling financial data.

5.4 Conclusions

This research addresses the important need for efficient reduction and classification of information in financial markets. It seeks to identify an appropriate method for effectively categorizing multidimensional financial data. By introducing a novel hybrid approach that combines AE-based compression with various clustering algorithms, this paper makes a significant contribution to unsupervised learning methodologies. These methodologies are particularly relevant for handling the complexities associated with high-dimensional data (Wen et al. 2019; L. Zhang et al. 2017) and time series classification (He et al. 2022; Liu et al. 2019). This study, therefore, extends the current understanding and capabilities in these areas, offering valuable insights and practical solutions for financial data analysis.

Furthermore, exploring how algorithms can aid stakeholders in automating financial research or operations is important and relevant, particularly in scenarios characterized by high volatility and substantial data volumes. This need becomes increasingly significant during short and rapid intervals, such as the intraday sessions of the stock market. During these periods, the stock market experiences swift fluctuations and an influx of information that can be overwhelming for traditional analysis methods. Intelligent systems can provide timely insights and support informed decision-making, which is crucial in domains where significant financial implications can arise within short periods of time. Therefore, understanding and enhancing the capability of algorithms to handle the intricacies of intraday market movements not only supports stakeholders dealing with complexities of the financial markets but also contributes to advancing financial research methodologies.

In this research, various reduction techniques were employed to evaluate the efficacy of ML and DL algorithms in processing intraday financial time series. The study particularly highlights the efficiency of AE in this context. It was observed that AE surpassed other reduction methods in creating more compact and distinct clusters. This was evidenced by applying the techniques to five different indexes across four distinct time frames, thereby contributing to the literature on clustering intraday financial time series (Shi et al. 2021).

Consequently, the findings suggest that those developing strategies based on intraday time series data should consider integrating AE with other clustering techniques. This approach is recommended over solely relying on traditional methods like PCA or just using the closing price. Integrating AE in such strategies provides a more nuanced and effective analysis of intraday financial data.

While AE and KNN-EUC have demonstrated efficiency and superiority in this study, there are instances of variability in the results when comparing the outcomes of the clustering techniques across different exchanges and time frames. There are specific scenarios where the highest individual values do not align with the overall averages. For instance, in terms of the SH metric, certain exchanges and time frames yielded the highest values with PCA, even though AE was, on average, the most successful technique by a significant margin.

This discrepancy indicates that while AE generally excels in handling multidimensional intraday data, its effectiveness can vary depending on specific contexts and conditions. Consequently, this research solidifies the appropriateness of AE for processing multidimensional intraday data. Future studies, therefore, might pivot towards identifying the most effective clustering technique to pair with AE-compressed time series data. Such focused research could further refine and optimize the application of these techniques in financial data analysis.

The evaluation of clustering techniques using reduced data provides a clear insight into the practical utility of reduction methods like AE, underscoring its advantages over other techniques. Recognizing the effectiveness of these processes enables the exploration of larger datasets, paving the way for research into models that can incorporate multiple and varied variables. This expansion significantly enhances the analytical capabilities of both investors and scholars.

AE's utility extends beyond merely simplifying the dimensionality of intraday prices. It can also be effectively used to condense extensive datasets encompassing various prices from different financial instruments and markets. The findings from this study suggest that reduction techniques like AE can be instrumental in developing advanced investment strategies. These strategies could create diverse classifications to simulate more realistic market scenarios, a key area of interest for future research. This approach can potentially bring about more sophisticated and practical investment models, further enriching the field of financial data analysis.

6

Application of Explainable Artificial Intelligence in Predicting Digital Asset Prices

Cryptocurrency markets are notably volatile, making the prediction of prices particularly challenging. This chapter explores the development and application of an XAI model to predict cryptocurrency prices, focusing on BTC and ETHereum (ETH).

The subsection 6.1 discusses the motivation for transparent and understandable AI systems in cryptocurrency prediction. It addresses the challenges posed by the high volatility of cryptocurrencies and the 'black box' nature of many ML models.

In the methodology subsection 6.2, the paper explores the development of the XAI model. It details selecting suitable analytical techniques and optimizing parameters for technical trading prediction. This section also describes the data collection process and the rationale behind focusing on BTC and ETH.

The results subsection 6.3 examines the performance of the developed model in forecasting the prices of cryptocurrencies. The section evaluates the model's results, particularly during periods of economic uncertainty, and discusses its strengths and limitations.

Finally, the subsection 6.4 provides an in-depth analysis of the study's implications. It highlights the significance of using XAI in Finance, especially for investors and regulators. This part also discusses how the model contributes to the understanding and overcoming of challenges associated with the opaque nature of ML models in financial predictions.

This chapter also aims to enhance the field of financial technology by demonstrating the practical application and importance of XAI in predicting cryptocurrency market trends, offering valuable insights for both research and practice in this dynamic sector.

6.1 Motivation

Cryptocurrency trading marks a new era in finance, distinguished by its exceptional volatility and complexity. The valuation of cryptocurrencies like BTC and ETH is shaped not only by standard economic factors but also by technological progress, regulatory shifts, and significant global incidents, including the COVID-19 pandemic and the 2008 financial crisis. This intricate mix of factors makes predicting cryptocurrency prices is difficult.

The motivation for this study derives from two key aspects. Firstly, the extreme volatility and economic uncertainty characteristic of cryptocurrencies. These digital assets exhibit notable price swings, creating a high-risk yet potentially rewarding investment environment. Factors like the COVID-19 pandemic and the 2008 financial crisis exemplify how economic turmoil can amplify the already erratic nature of cryptocurrency markets.

In such unpredictable settings, conventional predictive models often fall short. These models generally assume market consistency and linear data patterns, assumptions that are frequently overturned by significant economic changes. This scenario needs a more resilient and adaptable analytical method capable of handling the unpredictable nature of cryptocurrencies and adapting to the complexities during economic distress. Developing and enhancing such models is essential to deal with the uncertain domain of cryptocurrency investment and a deeper understanding of financial markets in turbulent times.

Additionally, cryptocurrencies combine characteristics of traditional financial and speculative assets, leading to independence from regional monetary policies (Atsalakis et al. 2019; Klein et al. 2018). They often show a low correlation with other financial instruments, distinguishing them in the financial market. Furthermore, the cryptocurrency market exhibits irrational and inefficient characteristics, including herding behavior (Rubbaniy et al. 2021). These unique aspects challenge the application of forecasting models based on fundamental analysis. Additionally, cryptocurrency market inefficiency restricts predictive models rooted in the EMH (Ozbayoglu et al. 2020; W. Zhang et al. 2018). Therefore, this scenario underscores the needs for developing unique methodologies to analyze and predict cryptocurrency market trends.

The second key motivation for this research is to address the opaque nature of many sophisticated ML models. These models often operate as 'black boxes,' lacking transparency in their decision-making processes, posing challenges in trust and interpretability. In the financial sector, where accuracy is important, and errors can have significant repercussions, the need for XAI models becomes essential. While some studies endorse the efficacy of ML in financial market forecasting (Z. Chen et al. 2020; Ozbayoglu et al. 2020), others have some objections (Bussmann et al. 2020; Jabeur et al. 2021b), due to lack of transparency.

This research aims to develop an explainable AI model for predicting cryptocurrency prices, filling a significant void in financial technology. By demystifying the model's decision-making, the study seeks to offer a precise and comprehensible tool to users like investors and regulators.

Therefore, This section emphasizes the need for advanced, transparent, and flexible AI models to handle the intricacies of the cryptocurrency market effectively.

6.2 Methodology

The methodological framework covers the data collection of cryptocurrencies and macro-economic variables, data pre-processing including technical indicators and filling missing values, followed by model development and hyperparameter selection. After identifying the most optimal model configuration with the least errors, prediction is conducted. The framework concludes with an explanation of the algorithm for model interpretation.

6.2.1 Data collection

The dataset for this research comprises the volume, along with the open, high, low, closing, and adjusted closing prices of ETH and BTC, spanning from September 1, 2015, to June 30, 2021. This data, which encompasses a total of 2130 days, was sourced from the Binance exchange. A key focus of the study is to evaluate the predictability of ML algorithms during periods of high uncertainty, particularly using the COVID-19 pandemic as a critical timeframe for this examination.

The models were trained using data from the first available day in the dataset until the beginning of the COVID-19 pandemic. The onset of the pandemic is identified based on the timeline the World Health Organization provided. Specifically, the study recognizes December 31, 2019, as the starting point of the COVID-19 pandemic, aligning with the date when the first obtained reports from the Wuhan Municipal Health Commission regarding cases of viral pneumonia in Wuhan, People's Republic of China. This demarcation is a pivotal reference in analyzing the performance and predictability of ML models during this globally impactful event.

The dataset was divided into training and testing phases for the ML algorithms. The training phase involves fitting the model parameters using a set of examples, while the testing phase validates the results by predicting new data. Given the dynamic nature of the cryptocurrency market, three testing intervals were established: the first and second quarters of 2020, the third and fourth quarters of 2020, and the first and second quarters of 2021. The training phase for each interval consists of data available before the start of that period. Figure 6.1 illustrates the train-test split.

6.2.2 Data pre-processing

The dataset used in this research initially includes 2130 data points each for BTC and ETH), covering aspects like volume, open, high, low, closing, and adjusted closing prices CP. Additionally, the dataset incorporates similar data for thirteen

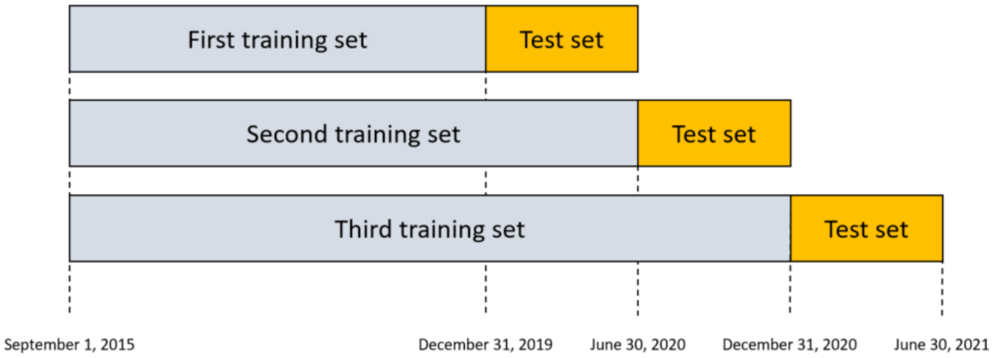


Figure 6.1: The timeline for three training and test periods, with training sets starting from September 1, 2015, and test sets at the end of 2019, 2020, and mid-2021.

other cryptocurrencies: XRP, Litecoin (LTC), Dogecoin (DOGE), Peercoin (PPC), BitShares (BTS), Stellar (XLM), Nxt (NXT), MaidSafeCoin (MAID), Namecoin (NMC), Tether (USDT), Cardano (ADA), Binance Coin (BNB), and USD Coin (USDC). These cryptocurrencies were selected based on their significant market activity during the study's training and testing phases (Grobys et al. 2020).

The dataset is further enriched with eight major indices from the Bloomberg terminal: Bloomberg Commodity Index (BCOM), CAC, DAX, SPX, UKX, Volatility Index (VIX), NASDAQ-100 Index, and Gold Spot Price PHLX Gold/Silver Sector Index (XAU). These indices are integrated to explore the potential impact of worsening economic conditions on the volatility of cryptocurrency prices during the COVID-19 pandemic, as investigated in the referenced study (Nikolopoulos et al. 2021).

Since the cryptocurrency market operates continuously throughout the year, adjusting the prices of traditional indices to ensure dataset consistency was necessary. On days when these markets were closed, the gaps were filled with the last available price. This approach was also applied to address any missing values in the cryptocurrency dataset. Such adjustments are crucial to maintain a balanced dataset and prevent issues during data processing with certain ML algorithms.

Considering the prevalent use of technical indicators in analyzing cryptocurrencies by investors and scholars (Vo and Yost-Bremm 2020), this study incorporates seven widely used technical indicators. These include Moving Average (MA) for various periods (5, 10, 20, 50, 100, 200 days), MACD, RSI, SO, OBV, WR, and data on previous closing prices for 1 to 8 days. [2]. After segmenting the data and ensuring all inputs were correctly incorporated, each training and testing set was normalized independently to prevent data leakage from the training to the testing set.

6.2.3 Model adoption process

In this research, various ML techniques were applied to address the unique characteristics of the cryptocurrency market, such as high volatility, inefficiencies, and the prevalence of high-frequency trading (D. G. Baur and Dimpfl 2018). The primary objective was to forecast the next day's closing price of ETH and BTC using data from these cryptocurrencies, along with technical indicators and macroeconomic variables (X. Li and C. A. Wang 2017).

To this end, nine different ML algorithms were evaluated to determine which offered the most accurate and reliable predictions for short-term cryptocurrency prices. The algorithms utilized in this study were implemented using Python, leveraging widely recognized and open-source computational libraries [3]. The specific algorithms we tested include DTR, KNNR, RFR, XGBR, LGBM, CATR. Additionally, we explored four different ANN architectures: a standard MLP, a conventional LSTM configuration, a Stacked LSTM (SLSTM), and a Bidirectional LSTM (BiLSTM). These algorithms and network structures were selected after reviewing relevant literature, ensuring their appropriateness for the research objectives.

To determine the most suitable hyperparameters for each model (Bergstra and Yoshua Bengio 2012), a randomized parameter optimization search was undertaken [4]. The main objective of this search was to identify the set of hyperparameters that minimizes the prediction error. For this purpose, the MSE metric was utilized as

---

[2] A detailed list of inputs can be found in Appendix B.1

[3] Numpy, Scikit-learn, XGBoost, CatBoost, Keras

[4] Details of the hyperparameters for each model can be found in Appendix C

the primary criterion to evaluate the overall error of each model. This approach was instrumental in ensuring the selection of the most efficient model configurations to enhance the accuracy of the predictive analysis.

The randomized parameter optimization search was independently carried out across three different testing periods to ensure a robust estimation of each model's performance. We selected the model demonstrating the lowest average MSE after this comprehensive evaluation. This approach was critical to ensure that the research was conducted using the most effective model, enhancing the predictions' accuracy and minimizing potential errors.

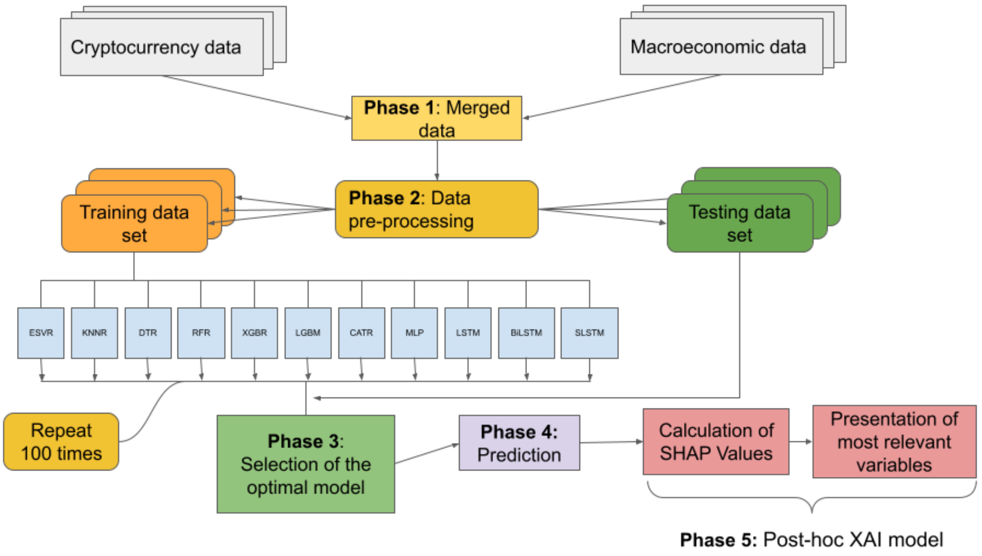
6.2.4 Model interpretation

To discern which variables the ML algorithm prioritizes during each period, an XAI framework was employed. This approach calculates the importance of each variable in the prediction process. After selecting the most suitable model for addressing the research question, predictions are made. We utilized SHAP values, employing a SHAP explainer [5] (S. Lundberg and S.-I. Lee 2017), to decipher how other cryptocurrencies, technical analysis indicators, and macroeconomic variables contribute to predicting ETH and BTC prices across three different testing periods during the COVID-19 pandemic.

The methodology is outlined in Figure 6.2, which illustrates the model development and interpretation process. Phase 1 involves constructing a mixed dataset that integrates cryptocurrency prices from various currencies with macroeconomic data. Phase 2 entails normalizing and dividing this dataset into several training and testing sets, which are then fed into various ML models for comprehensive exploration. In Phase 3, the model with the lowest prediction error is selected. Following this, in Phase 4, predictions are made using the optimal model. Phase 5 brings clarity to the AI 's decision-making process by calculating SHAP values using Equation 2.35. This study introduces the innovative use of SHAP values in three distinct periods to highlight the most relevant variables and transform them to compute each variable's relative significance. This approach offers a practical method for assessing the contribution of different variables in predicting cryptocurrency prices.

---

[5] Using the Python library SHAP



.

Figure 6.2: Explainable AI model for predicting cryptocurrency prices

6.3 Results

6.3.1 Model performance

Following a comprehensive randomized parameter optimization search to identify the optimal model and its corresponding ideal hyperparameters, we determined that the RFR algorithm consistently exhibits the lowest MSE across the three testing periods. Consequently, we contend that the RF algorithm is the most effective ML model for predicting the prices of ETH and BTC during periods of economic uncertainty.

Indeed, the effectiveness of the RFR algorithm is highlighted by its capability to reduce the mean of the squared discrepancies between the actual prices and the predicted values in terms of MSE. This reduction reduces prediction error, thereby underscoring the model's reliability and precision in estimating cryptocurrency prices, particularly in fluctuating economic circumstances. The RFR algorithm's proficiency in handling such volatility effectively demonstrates its suitability and robustness as a tool for forecasting in the dynamic and often unpredictable cryptocurrency markets.

Algorithms	**ETH**			**BTC**			
	MSE 1	MSE 2	MSE 3	MSE 1	MSE 2	MSE 3	Average
Decision tree	0.039916	0.01968	0.014941	**0.022477**	0.044211	0.117307	0.043089
K-NN	0.06037	0.020451	0.029823	0.109731	0.022159	0.145628	0.064694
Random Forest	0.025617	0.013093	0.009101	0.043714	0.011374	0.058414	**0.026886**
XGBoost	0.028661	0.014827	0.008415	0.04962	0.015811	0.051339	0.028112
Multilayer	**0.022562**	**0.010876**	0.039155	0.033399	0.086046	**0.039572**	0.038602
Light BM	0.028075	0.019216	**0.008272**	0.052902	0.010795	0.056746	0.029334
CAT	0.032937	0.020056	0.012241	0.037527	0.027449	0.081612	0.035304
LSTM Simple	0.04303	0.034054	0.048363	0.045041	**0.010523**	0.129956	0.051828
LSTM Stack	0.07169	0.043402	0.099508	0.061454	0.014348	0.086535	0.062823
Bidirectional LSTM	0.023358	0.011756	0.011873	0.080382	0.019465	0.244418	0.065209

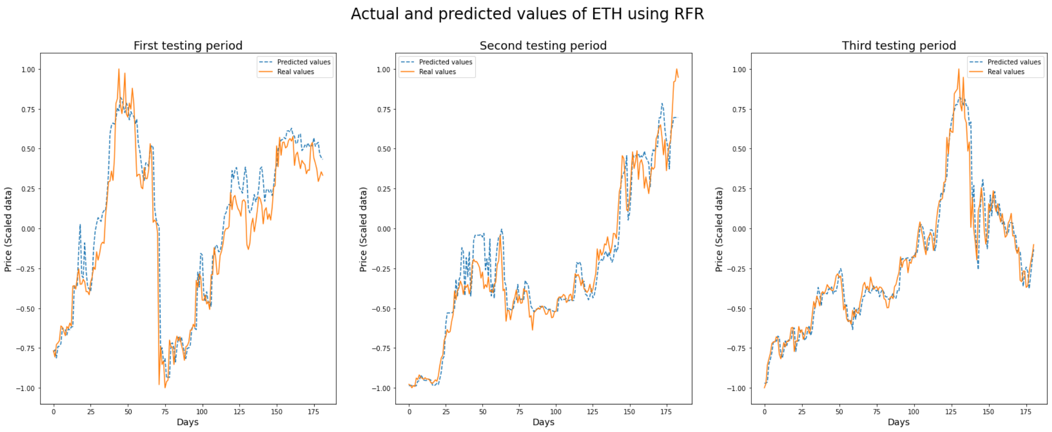
Table 6.1: Prediction errors from different ML algorithms.

In predicting ETH prices, Table 6.1 reveals varying levels of MSE across different periods. Specifically, the MSE for the RFR is recorded at 0.025617 for the first period. In the second period, there is a noticeable improvement with the MSE decreasing to 0.013093. The third period's most significant enhancement is observed, where the prediction error is the lowest, indicated by an MSE of 0.009101.

To further analyze and visually represent the forecasting process on test sets, Figures 6.3 and 6.4 depict the actual versus predicted values of ETH and BTC, respectively, as forecasted by the RFR algorithm. These figures provide a clear graphical representation of the algorithm's performance, illustrating its accuracy in predicting the prices of these cryptocurrencies over the specified periods.

For BTC, Table 6.1 indicates that the prediction errors are generally higher than those for ETH. Specifically, in the first test dataset, an MSE of 0.043714 is noted, signifying a higher degree of error in the predictions. The second test dataset shows some improvement with an MSE of 0.011374. However, Table Table 6.1 also highlights that the last testing period posed significant challenges for forecasting, with the RFR algorithm recording an MSE of 0.058414 during this phase.

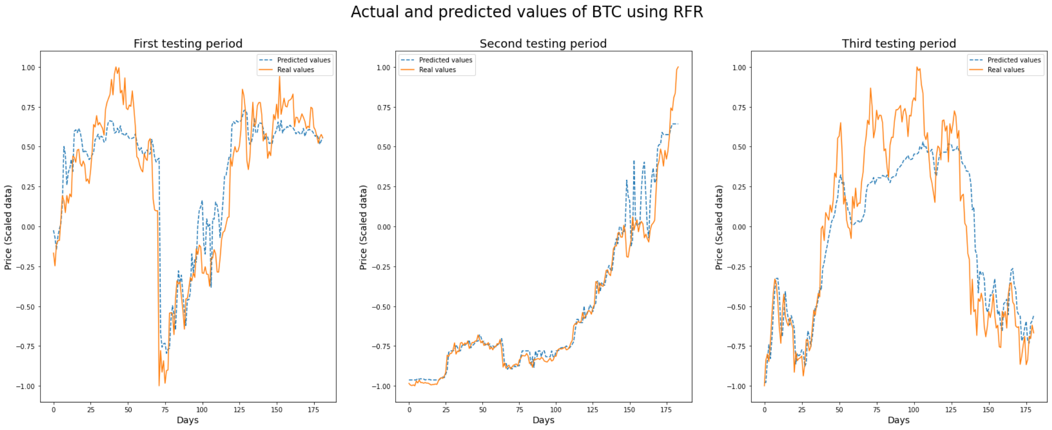
It is important to mention that this final testing period proved difficult for all the algorithms in accurately predicting BTC values, not just for the RFR algorithm. This increase in prediction error during the last period reflects the inherent challenges



.

Figure 6.3: Prediction of ETH using RFR

and unpredictability associated with forecasting BTC prices, especially during heightened market volatility or unusual economic conditions.



.

Figure 6.4: Prediction of BTC using RFR

The observation that the RFR algorithm has the lowest average MSE in predicting cryptocurrency prices is noteworthy. However, it's important to recognize that other models, such as the MLP and BiLSTM, exhibit lower error values in certain testing periods for forecasting the prices of ETH and BTC as shows in Table 6.1. This variance in performance across different periods underscores that no single, universally superior model consistently predicts prices more accurately in all situations with a static set of parameters.

Most research in cryptocurrency price prediction focuses on a single period, often not addressing periods of economic uncertainty. However, recent studies have started comparing the performance of predictive models across different periods (Abedin et al. 2021) and specifically during times of uncertainty (Nikolopoulos et al. 2021; Stevenson et al. 2021). This research contributes to this emerging perspective by emphasizing the importance of conducting multiple tests over time to provide a more comprehensive understanding.

This study also highlights the evolution of ML models, evident in the changing configurations of hyperparameters for each algorithm over time to yield better predictions for specific timeframes. This ongoing reconfiguration suggests that adapting models to market dynamics is essential for accurate forecasting. Consequently, a thorough review of variables should be integrated into predictive models. Utilizing explanatory tools, such as SHAP values, can be instrumental in measuring the significance of each feature during specific periods. This approach ensures that the predictive model remains relevant and effective as market conditions evolve.

6.3.2 Variable importance

This section outlines a method for understanding the variables an algorithm prioritizes when making a prediction. Clarifying the contributing factors in forecasting is crucial to mitigate the 'black box' nature often associated with ML. This clarity is especially important in uncertain times to facilitate informed decision-making and ensure transparency and auditability in investment decisions. To determine the significance of each variable following predictions made with the RFR algorithm, we employed SHAP values. This methodology is recognized for its effectiveness in providing explainability and deriving insights from the interaction of predominant features in complex, multi-task datasets (Barredo Arrieta et al. 2020).

The results from this technique are presented in an $n \cdot m$ matrix, where n represents different periods and m denotes the variables. By transposing this matrix, we calculate each variable's average of the absolute SHAP values and then determine their relative percentage. This process allows for a more intuitive presentation of

SHAP values, enabling the contribution of each variable to be viewed in percentage terms.

Analyzing the variable participation within the ETH price predictions reveals that the previous day's closing price and the adjusted closing price are the most influential, accounting for nearly 50 % of the prediction influence between them. This pattern indicates that the primary predictive variable is the instrument's price rather than an external factor. However, in the third period, where the predictive error is lower, the influence of variables related to the closing price diminishes compared to the other periods. Notably, the high and low prices from the previous day emerge as significant variables, varying in importance across different periods, as illustrated in Figure 6.5. This figure initially displays a straightforward representation of the SHAP values before calculating each variable's relative percentage.

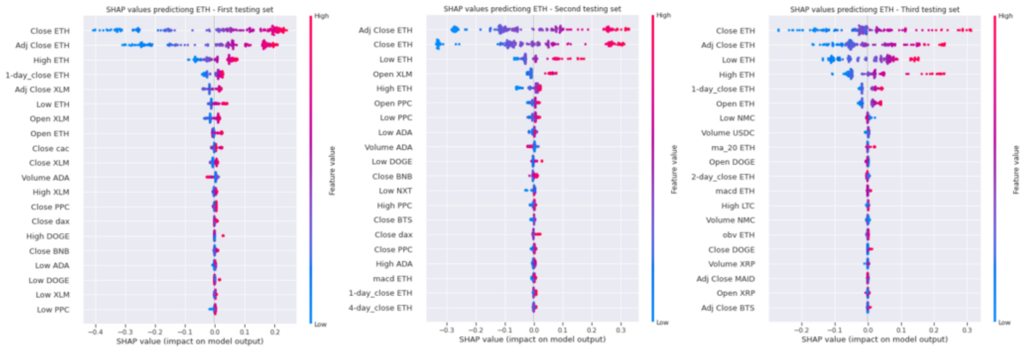


Figure 6.5: SHAP values representation for the prediction of ETH.

In the analysis of additional variables beyond the primary cryptocurrency prices, it is observed that technical indicators generally do not significantly impact the predictions across the various periods. Notable exceptions are the two-day price lag and the 20-day MA, particularly in the third period, as indicated in Table 6.2.

Interestingly, the algorithm identifies the closing prices of other cryptocurrencies as key variables. For instance, in the first testing period, the closing price of XLM is more influential than the closing price of ETH. It continues to contribute significantly in the second testing period. However, its relevance diminishes in subsequent periods, possibly due to market corrections affecting XLM and the growing investor interest in other cryptocurrencies like DOGE during the last testing period.

Notably, the trading volume of ETH is not deemed as one of the most impactful variables. This observation is particularly striking, considering that the algorithm attributes importance to the trading volumes of other cryptocurrencies during the second and third periods. This disparity in the algorithm's valuation of different variables underscores the complexity of the factors influencing cryptocurrency price predictions and highlights the dynamic nature of market influences across different periods.

In analyzing the ten most influential variables for predicting BTC prices, patterns similar to those observed with ETH emerge. Again, the adjusted closing and closing prices stand out as the most critical variables. However, predicting BTC prices appears more challenging than ETH, particularly in the third period, where the influence of closing values is comparatively lower than in other testing periods. For instance, Figure 6.6, which presents a straightforward depiction of SHAP values for BTC predictions, shows no distinct dominance of the closing price over other variables in the third period.

In contrast to earlier periods, the algorithm in the last period gives considerable weight to lagged closing prices. Furthermore, analogous to the findings with ETH, BTC predictions also do not heavily rely on technical indicators or macroeconomic variables as primary factors, except in the first period, as indicated in Table 6.3. The participation of the XLM cryptocurrency is notably significant in the initial periods, but its relevance diminishes in subsequent periods.

These observations highlight the complexity and variability inherent in cryptocurrency price prediction. The algorithms' shifting focus from one variable to another across different periods reflects the dynamic and evolving nature of the cryptocurrency markets, underscoring the necessity for adaptive and responsive predictive models.

ML is a practical methodology for predicting cryptocurrency prices with low error margins. This effectiveness is further improved when the optimal model is analyzed for feature importance using SHAP values. Through this examination, it becomes evident that cryptocurrencies' closing, high, and low prices are the most critical variables for successfully predicting the prices of ETH and BTC, particularly

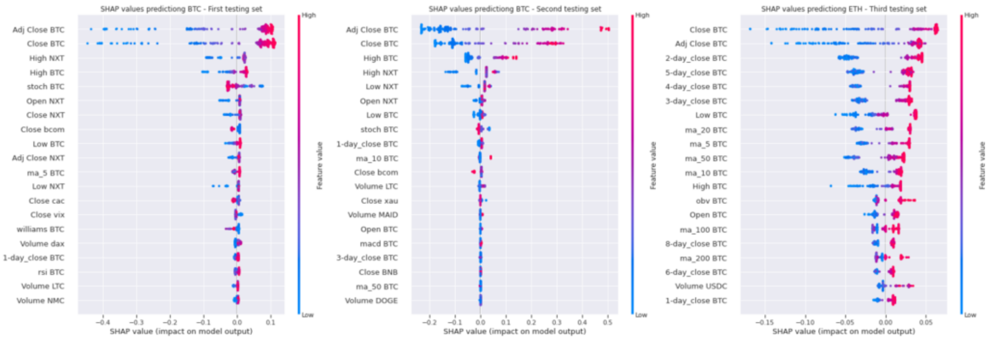


Figure 6.6: SHAP values representation for the prediction of BTC.

during periods of uncertainty. These findings highlight the significant influence of the inherent price movements of these cryptocurrencies in the prediction models.

Additionally, the prices of other cryptocurrencies also play a notable role in the prediction process, underscoring the interconnected nature of the cryptocurrency market. However, it's interesting to observe that technical analysis indicators and macroeconomic variables, often considered important in traditional financial markets, have little to no significant impact on cryptocurrency price prediction models. This distinction underscores the unique dynamics of the cryptocurrency market, where internal market factors and inter-cryptocurrency relationships are more pivotal in shaping price movements than external economic indicators or technical analysis tools.

6.4 Conclusions

The COVID-19 pandemic has been critical for testing new technologies, particularly in financial markets. In this research, we have demonstrated the value of an indispensable technological tool designed to navigate and adapt to the challenges presented by high-frequency trading in inefficient market environments. We focused on developing an XAI model capable of predicting cryptocurrency prices during economic uncertainty.

This initiative was driven by the limitations observed in existing research, particularly regarding the opacity of ML models' decision-making processes in cryptocurrency price prediction. The model displayed in this work insight into how these predictions are made. The findings challenge the notion that a broad array of variables, including technical rules (Corbet et al. 2019; Hudson and Urquhart 2021), macroeconomic indicators (Jabeur et al. 2021c), or extensive technical analysis (Grobys et al. 2020; Vo and Yost-Bremm 2020), is necessary for effective prediction. Instead, we establish that ML can yield accurate predictions using limited variables (Ibrahim et al. 2021) without relying on additional tools.

Interestingly, the results suggest that DL algorithms do not consistently outperform traditional ML algorithms (W. Jiang 2021). This observation indicates that the most complex models do not automatically have an edge over more traditional ML approaches in predicting cryptocurrency prices.

Given these insights, the XAI model shown in this research has significant practical implications. For practitioners such as investors, it clears prevalent market misconceptions and aids in developing optimal investment portfolios during uncertain times. Additionally, the model offers regulators a reliable tool to manage better and respond to market volatility. By providing a trustworthy and transparent approach to understanding market dynamics, this research contributes to more informed decision-making in the rapidly evolving and often unpredictable realm of financial markets.

This study contributes to cryptocurrency research, particularly in the context of economic uncertainty and the decision-making processes of ML models. Unlike previous research that predominantly focuses on stable economic conditions, this work demonstrates the effective implementation of ML models during periods of economic uncertainty. Specifically, we have shown that integrating multiple inputs and incorporating explainability into the prediction process makes it possible to generate accurate predictions and discern which features are critical in forecasting cryptocurrency price movements. Understanding the variables involved in predictions enhances decision-making and strengthens trust in ML models, providing stability in economic uncertainty.

A novel aspect of this research is the use of explainable ML models for predicting cryptocurrency prices during uncertain economic times, a first in this domain to our knowledge. The model is valuable to academic research and practical applications, filling a gap left by existing models that have not incorporated explainable frameworks in their design. By introducing explainability, the model advances the current understanding and contributes significantly to developing cryptocurrency and AI research. This model allows researchers and practitioners to comprehend future predictions better, fostering transparency and accountability in trading.

The timely findings extend the literature on forecasting during periods of uncertainty. The research outcomes have immense practical implications, equipping investors with crucial insights for informed decision-making and aiding in reducing market inefficiencies. Using ML models in this context can also assist regulators in reviewing existing governance mechanisms within the cryptocurrency trading ecosystem. Moreover, the interaction between management science and financial investment provides valuable guidance for policymakers in enhancing transparency and trust in cryptocurrency trading.

The research underscores the need for frequent evolution in the prediction process of cryptocurrency prices, as the complexity of models and the nature of data are subject to change. The study highlights the importance of adapting hyperparameters and incorporating new variables for accurate forecasting, cautioning against a one-size-fits-all approach in predicting cryptocurrency prices. The dynamic nature of digital assets and emerging new cryptocurrencies necessitate periodic adjustments to ensure compelling predictions.

This research explored how XAI algorithms can forecast cryptocurrency prices during economic uncertainty, with the COVID-19 pandemic serving as a pertinent test case. While the model provides valuable insights, we acknowledge certain limitations that open future research avenues. Expanding the dataset to encompass different economic periods, applying the model to a broader range of cryptocurrencies, and validating the model with traditional financial assets during uncertain times are potential areas for further exploration.

	% Shapley value		
	First testing period	Second testing period	Third testing period
Adj Close BTC	25.75	37.94	8.68
Close BTC	25.05	28.70	11.69
High NXT	6.84	7.51	*
High BTC	6.39	10.82	3.35
SO BTC	5.58	1.42	*
Open NXT	2.81	1.61	*
Close NXT	2.70	*	*
Close bcom	2.10	0.97	*
Low BTC	1.79	1.52	5.29
Adj Close NXT	1.76	*	*
Low NXT	1.68	4.56	*
1-day_close BTC	0.89	1.03	1.72
MA 10 BTC	*	1.01	3.73
2-day_close BTC	*	*	8.04
5-day_close BTC	*	*	6.08
4-day_close BTC	*	*	6.06
3-day_close BTC	*	0.07	5.49
MA 20 BTC	*	*	4.98
MA 5 BTC	1.74	*	4.94
MA 50 BTC	*	0.07	4.22

Table 6.2: Shapley prediction ETH with RFR.
(* Features not relevant in this period)

	% Shapley value for BTC prediction		
	First testing period	Second testing period	Third testing period
Adj Close BTC	25.75	37.94	8.68
Close BTC	25.05	28.70	11.69
High NXT	6.84	7.51	*
High BTC	6.39	10.82	3.35
SO BTC	5.58	1.42	*
Open NXT	2.81	1.61	*
Close NXT	2.70	*	*
Close bcom	2.10	0.97	*
Low BTC	1.79	1.52	5.29
Adj Close NXT	1.76	*	*
Low NXT	1.68	4.56	*
1-day_close BTC	0.89	1.03	1.72
MA 10 BTC	*	1.01	3.73
2-day_close BTC	*	*	8.04
5-day_close BTC	*	*	6.08
4-day_close BTC	*	*	6.06
3-day_close BTC	*	0.07	5.49
MA 20 BTC	*	*	4.98
MA 5 BTC	1.74	*	4.94
MA 50 BTC	*	0.07	4.22

Table 6.3: Shapley prediction BTC with RFR.
(* Features not relevant in this period)

7

Portfolio Construction Using Explainable Reinforcement Learning

This chapter addresses the critical intersection of ML and financial trading, particularly emphasizing algorithmic transparency and explainability challenges. It presents a novel study on applying an XRL tailored for portfolio management in the financial markets. This research moves beyond basic asset prediction, focusing instead on developing concrete, actionable trading strategies.

The motivation subsection 7.1 discusses the necessity of advancing beyond the traditional 'black-box' approaches in financial ML. It emphasizes the importance of algorithmic transparency and explainability, especially in trading and portfolio management.

In the methodology subsection 7.2, the study explores the design and implementation of the XRL model. This section provides a comprehensive discussion of the custom trading environment crafted to simulate the financial conditions of the CAC index. The approach allows the model to dynamically adapt to market changes by iteratively learning from historical data.

The subsection 7.3 and 7.4 meticulously analyze the experimentation and the empirical results obtained from the model. Out-of-sample tests highlight how the XRL model successfully outperforms equally weighted and classical portfolios, showing its practical efficacy in real-world trading scenarios.

Finally, the subsection 7.5 examines the broader implications of this research. It underscores the dual contribution of the study: improving algorithmic planning in financial trading and enhancing transparency and interpretability in financial AI. This chapter presents a novel approach to tackling the problem of the 'black box' in RL, offering a transparent framework for investment portfolio management.

This chapter aims to enrich the financial technology field by demonstrating the practical application and strategic advantages of XRL in portfolio management. It contributes significantly to the evolving field of algorithmic trading, providing valuable insights for academic research and practical financial applications.

7.1 Motivation

The evolution of ML in financial trading has been fast, transforming how markets are analyzed and traded. Despite these advancements, the field faces a persistent challenge: the opacity of algorithmic decision-making processes. This issue is particularly pronounced in high-frequency trading and portfolio management, where decisions must be accurate and understandable to stakeholders.

This research aims to contribute to the need for greater transparency and explainability in financial ML models. In the high-stakes world of financial trading, where decisions can have significant economic impacts, the underlying logic of algorithmic decisions must be transparent and interpretable. The lack of transparency inhibits trust in automated systems and complicates regulatory compliance and risk management.

This study aims to bridge this gap by introducing an XRL model tailored explicitly for portfolio management. The model goes beyond predicting asset prices by formulating a concrete, actionable trading strategies. This approach departs from traditional models that often prioritize predictive accuracy over interpretability.

The focus on the CAC index provides a relevant and challenging testing ground for this research. The index's complex and dynamic nature requires a model capable of adapting quickly to market changes. The model's ability to learn iteratively from historical data and adjust its strategies accordingly is central to its goal.

In summary, this research is motivated by the dual challenge of enhancing algorithmic efficiency in financial trading while concurrently addressing the need for transparency and interpretability in AI models. The study also aims to advance in creating more reliable and transparent approaches to managing investment portfolios.

7.2 Methodology

This work brings together the RL and XAI and applies it to the financial field to create an automated trading system. This section will describe our steps to create our RL model and scrutinize its explanatory properties. First, we will start by presenting the setup, explaining the decision process by introducing the concept of state, action space, and reward function, followed by the presentation of the agent and its structure and then continue with the exposition of the attention layers that will be added to the agent to obtain explainability.

7.2.1 The decision process

Given that an investor seeks to maximize their profits, under the concept of modern portfolio theory we can see that the expected utility function (Arrow et al. 1974; Pratt 1964) is defined as,

$$\mathbb{E}\left[U(W_T)\right] = \mathbb{E}\left[U\left(W_0 + \sum_{t=1}^{n} \delta W_t\right)\right] \tag{7.1}$$

where over time T the utility function U has a final wealth function W_T. This framework proposes that an investor's satisfaction or utility from final wealth is determined not solely by initial wealth W_0 but also by the cumulative changes in wealth δW_t across periods. The expectation $\mathbb{E}$ encapsulates the probabilistic outcomes of different investment decisions, factoring in the likelihood and impact of potential gains or losses over time. This approach, emphasizing the trade-off between risk and return, underlines the investor's objective to craft a portfolio that not only aims for the highest possible returns but also aligns with their risk tolerance.

In this research we assume that the investor is risk neutral and their utility function is linear since the objective is to maximize the expected cumulative returns, as shown in Equation 7.2.

$$\mathbb{E} = \sum_{t=1}^{T} \delta W_t \tag{7.2}$$

Assuming a linear utility function for an investor, particularly under modern portfolio theory and in modern investment approaches (Z. Zhang et al. 2020), is a theoretical simplification that facilitates the mathematical modeling of investment strategies. Using a linear function means that the investor is considered risk-neutral. This means that the investor values incremental gains in wealth equally. Risk-neutral investors are aware of the risks involved, yet they choose not to factor them into their decision-making process at the portfolio construction stage, focusing instead on maximizing expected returns irrespective of the variance in those returns.

Thus, the objective of the RL in this study is to maximize an investor's wealth by following a sequential or MDP in which the agents learn by interacting with an environment at time steps to achieve a goal. At any point in time t, the agent obtains a representation of the environment denoted as a state s_t in a state space $\mathcal{S}$ and then takes action a_t from the action space $\mathcal{A}$, following a policy $\pi(a_t|s_t)$.

Because the agent takes action a_t in a certain state s_t, it receives a scalar reward r_t, and according to the dynamics of the environment, it also receives the transition to the next state s_{t+1} for a state transition probability $\mathcal{P}(s_{t+1}|s_t, a_t)$ and a reward function $\mathcal{R}(s, a)$. The constant interplay of the agent with the environment yields a trajectory $\tau = [s_0, a_0, r_0, s_1, a_1, r_1, \ldots, s_n, a_n, r_T]$. The goal of the agent is to maximize the expected return at time t as denoted in the following equation,

$$G_t = \sum_{j=t+1}^{n} \gamma^{j-t-1} r_j \tag{7.3}$$

where $\gamma \in (0, 1]$ is the discount factor, in this way MDP can be defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$. Therefore, the RL agent focuses on developing a maximization r_t, thus optimizing the function $\mathbb{E}(G)$ is equivalent to optimizing Equation 7.2, where a given investor maximizes the expected cumulative returns.

To create a reward function $\mathcal{R}$ that satisfies and relies on Equation 7.1, in this research, we calculate the returns by creating a vector q that represents the shares of assets that an investor has, where its components are symbolized as q_i where $i \in 0, 1, .., n$ in a specific market with n financial assets. Each component q_i must receive a certain weight w_i, representing the proportion of this element in a portfolio. In this way, the expected returns and reward function can be represented as,

$$\mathcal{R} = \mathbb{E}[q_n] = \sum_{i=1}^{n} w_i \mathbb{E}[q_i] \tag{7.4}$$

The action space $\mathcal{A}$ is the set of all possible actions and determines how the agent reacts in a given environment, and in this research, we define an action a_t at a given time t as,

$$a_t = w : w \in \mathbb{R}, 0 \leq w \leq 1 \tag{7.5}$$

where,

$$\sum_{i=1}^{n} w_i \leq 1 \tag{7.6}$$

In this way, the agent is trained to find an optimal w_i for each q_i component. Since, in the initial stage, the agent has no prior knowledge of the optimal values of w, these are started randomly. Subsequently, these change according to the agent's training through RL. The modification of the weights generates an adjustment in the number of shares held in our model, which must be translated into a purchase or sale of assets to adjust the portfolio within our simulation environment.

Regarding the state space, $\mathcal{S}$ can be considered the set that holds all the possible variables in the agent's environment. It is impossible to know which variables will influence the movement of a certain financial asset. However, in this work, we have created a state space with features other authors have used in the literature.

7.2.2 The agent

The goal of the RL agent is to find an optimal policy π to maximize the reward function $\mathcal{R}$, to achieve this, an ANN can be a function approximator that outputs a policy $\pi_\theta(a_t \mid s_t)$ where the parameters of the function are represented by θ.

The policy gradients method to update θ based on rewards uses gradient ascent on the expected cumulative returns of the policy represented as $J(\theta)$:

$$\nabla_\theta J(\theta) = \mathbb{E}_\tau \left[\sum_{t=0}^{T-1} \nabla_\theta log \pi_\theta(a_t \mid s_t) G_t \right] \tag{7.7}$$

However, this process is not always efficient because the policy update is developed once the episode ends, so a proposed modification to solve this is the A2C method, which is the synchronous version of the Asynchronous Advantage Actor-Critic algorithm proposed by Mnih et al. (2016) to update the policy in real-time. This solution relies on two models, one is a network that plays the role of an actor creating a policy that updates through the output of a critic that estimates the value function in a given state. The A2C method maximizes the objective function by updating the policy π_θ as defined in the following equation:

$$\nabla_\theta J(\theta) = \mathbb{E}_\tau \left[\sum_{t=0}^{T-1} \nabla_\theta log \pi_\theta(a_t \mid s_t) A_{advantage}(S, A) \right] \tag{7.8}$$

The improvement of A2C relies on the addition of an advantage function that is expressed as:

$$A_{advantage}(S_t, A_t) = R_t + \gamma V(S_{t+1} \mid w) - V(S_t \mid w) \tag{7.9}$$

Where $V(s \mid w)$ represents a state function with parameters w from the critic network that can be updated by minimizing the temporal difference error using gradient descent as:

$$J(\theta) = (R_t + \gamma V(S_{t+1} \mid w) - V(S_t \mid w))^2 \tag{7.10}$$

Since the agent must process the information contained in the state space, in this research, we have designed a multi-head LSTM neural network structure to process the information of each asset in a parallel way; by doing this, the data of an asset is processed by a network LSTM independent of that of another asset as shown in Figure 7.1.

The results of each network are concatenated to be then processed by another ANN that will finally give two outputs: the critic's values and the actor's actions.

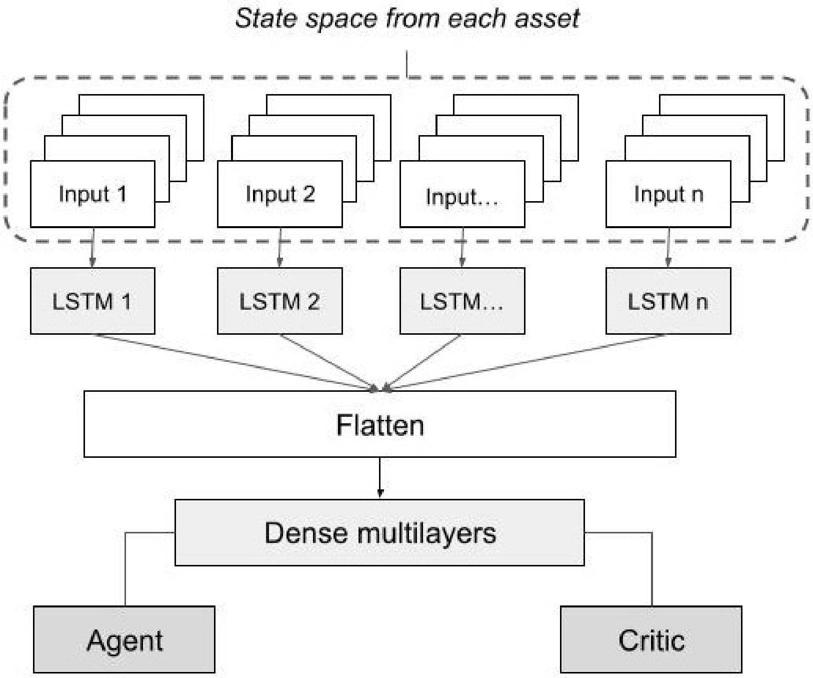


Figure 7.1: General scheme of the agent's architecture.

The actor's output determines the agent's actions and asset weights, which is why a vector with several elements equal to the number of assets in the portfolio represents it. These elements comprise the action space expressed in Equation 7.5 and 7.6, which go directly to the environment to perform a buying or selling action.

7.2.3 The environment

After the agent has delivered a series of actions to be taken, these must be tested in an environment that allows financial conditions to be simulated. To achieve a realistic scenario, We have created an environment that enable trading shares in the market by emulating the set of shares available for transactions.

The environment starts once it receives the sets containing the current state s_t, composed of ten elements with four lagging variables. In this way, we develop $\mathbb{S}$ incorporating the different opening, high, low, and closing prices and volume at a given time t. Additionally, we include different technical indicators commonly used

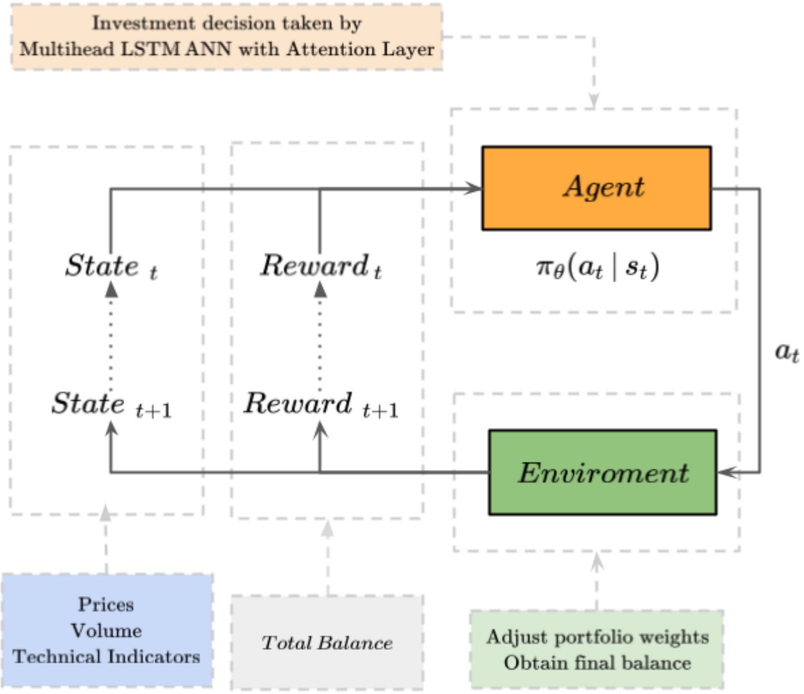


Figure 7.2: General scheme of the RL training process shown in this paper.

to predict financial markets. In this paper, we include MACD, RSI, and 14, 21, and 100-day moving averages MA.

The MA technical indicator can be described as:

$$MA(k) = \frac{1}{N}\sum_{t=k}^{N+k-1} closing\,price_t \tag{7.11}$$

where N stands for the number of data points and k is the lagging period.

$$RSI = 100 - 100/(1 - RS_p) \tag{7.12}$$

where RS_p is the relationship between the averages of up price movements with the average of down price change in period p.

For the MACD formula, it is first necessary to calculate the EMA since it is the difference between the fast and a slow formula, and it is defined as:

$$MACD(fast, slow) = EMA_{fast} - EMA_{slow} \tag{7.13}$$

$$EMA_t = \alpha * closing\,price_t + (1 - \alpha) * EMA_{t-1} \tag{7.14}$$

and α stands for the degree of decrease, calculated as $\frac{2}{t+1}$, where t is the day of EMA.

Additionally, the environment needs a set of action vectors from the agent a_t; this information calculates how much capital should be positioned in each asset. The weights from this vector are divided over the initial cash amount [6] and then divided by a purchase price, with a rounded down to the next lower integer. The simulation environment employs an adaptive pricing model to refine this process further and mitigate the challenges of executing buy or sell orders within a given trading day. This model strategically selects execution prices within the day's observed price range, embracing the market's inherent volatility and liquidity constraints. By incorporating a random selection mechanism within a trading day, the simulation reflects a more realistic scenario where the exact price at which an order is filled cannot be predetermined, thus closely mimicking the unpredictability and fluctuations of market conditions. This refined approach tackles the uncertainty associated with trade execution prices but also aids in preventing overfitted strategies.

Once there is a change in the weightings, the portfolio is readjusted and starts by selling shares if the agent's weight is less than that of the portfolio, and buying is done if the weighting is increased and there is enough cash because, in this RL application, we do not consider leverage. Once the simulation is finished, the environment produces the final balance with which the performance will be evaluated.

Once the simulation has started, the agent sends an updated set of portfolio weights, which will rebalance the portfolio if different from the previous actions. This action implies the simulation of the sale or purchase of assets. If the new weight of a specific asset is greater than the old one, a purchase must be made. In each step within the simulator, we first sell all the shares that must be readjusted and then buy those necessary to balance the portfolio according to the agent's actions.

---

[6] A complete presentation of the variables involved in the agent and the environment phase are presented in Table 7.2.

The environment is implemented utilizing the OpenAI Gym interface [7], featuring a custom-designed simulation environment tailored explicitly for the demands of algorithmic trading research. Detailed pseudo-code outlining the operational framework of the trading environment is delineated within the structure of Algorithm 1, illustrating its comprehensive structure and functionality.

7.2.4 The attention layer

Due to the great need to understand the variables involved in the decision-making process of different ML algorithms, especially neural networks, different models have emerged that help show the importance of the different features that have become popular over the years. Within the taxonomy of the literature, two large groups of XAI can be identified: transparent algorithms and post-hoc explainability (Barredo Arrieta et al. 2020). Since our paper intends to provide explainability, and as we opted for an RL which is not transparent by nature (Heuillet et al. 2021) and the implantation of a post-hoc algorithm is impractical due to the intricate configuration of the network, we decided to add an attention layer to the agent architecture, between the inputs and the LSTM network, as shown in Figure 7.3.

An attention layer is a vector added to the policy network and helps elucidate each variable's weight and memorize long information concatenations. To compute attention att_k for each variable k in different time steps, we use a softmax function:

$$att_k = softmax(FW_k x_k) \tag{7.15}$$

where the input feature has a learned weight FW_k and x_k represents a single variable over time. Afterward, the inputs from the state vector for each individual share are weighted by the calculated attention vector and go to the LSTM as input y_k, where:

$$y_k = a_k \odot x_k \tag{7.16}$$

Multiple studies have used attention layers, especially in computer-assisted decision support (Barredo Arrieta et al. 2020; Kaji et al. 2019). Figure 7.3 shows

---

[7] Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, Tang J, et al. Openai gym. 2016.

Algorithm 1 Algorithm of the trading agent in the simulated environment

```
Require: Actions from the agent
Require: State space
Ensure: Final Balance
$N \leftarrow$ number of assets
$A \leftarrow$ set of actions from the agent
$Prices \leftarrow$ set of actual prices from the assets
$PW \leftarrow$ set of weights of the assets in the portfolio
Commission Rate $\leftarrow 0.005$                                  ▷ Define the commission rate
Balance $= \sum_{i=1}^{N} PW_i \cdot Prices_i$
Cash $=$ Cash $-$ Balance
Total Commissions $= 0$                                             ▷ Initialize total commissions
while $Balance > 0$ do
    for $i = 1$ to $Episodes$ do
        if $Episode = 0$ then                                       ▷ Initial Buy of stocks
            for $j = 1$ to $N$ do                                   ▷ Iteration through the different assets
                Buy $n_j$ shares
                Commission $= n_j \cdot Prices_j \cdot$ Commission Rate
                Cash $= Cash - (n_j \cdot Prices_j + Commission)$
                Total Commissions $+= Commission$
                Initial portfolio $= PW_{ij} * Prices_{ij}$
                Balance $= Portfolio + Cash$
            end for
        else                                                        ▷ Rebalance the portfolio
            for $j = 1$ to $N$ do                                   ▷ Iteration through the different assets
                $n_j \leftarrow$ number of shares different between A and PW
                if $A_{ij} < PW_{ij}$ then                          ▷ Selling phase
                    SELL $n_{ij}$ shares
                    Commission $= n_{ij} \cdot Prices_{ij} \cdot$ Commission Rate
                    Cash $= Cash + (n_{ij} \cdot Prices_{ij} - Commission)$
                    Total Commissions $+= Commission$
                    $PW_{ij} = A_{ij}$
                else if $A_{ij} > PW_{ij}$ and Cash $> 0$ then      ▷ Buying phase
                    BUY $n_j$ shares
                    Commission $= n_j \cdot Prices_j \cdot$ Commission Rate
                    Cash $= Cash - (n_j \cdot Prices_j + Commission)$
                    Total Commissions $+= Commission$
                    $PW_{ij} = A_{ij}$                              ▷ Update of the weights
                else
                    Continue
                end if
            end for
        end if
    end for
end while
Final Balance $= \sum_{k=1}^{N} PW_k * Prices_k + Cash - Commissions$   ▷ This value used as Reward
```

the representation of the addition of an attention layer to the general scheme of the agent previously shown in Figure 7.1. By adding this layer, it is possible to obtain the attention vector att_k to determine the relevance of each feature before passing the inputs to the LSTM. The details of the architecture of the ANN, marked as Model architecture I and II in Figure 7.1, are shown in full detail in Appendix C.1 and C.2, respectively.

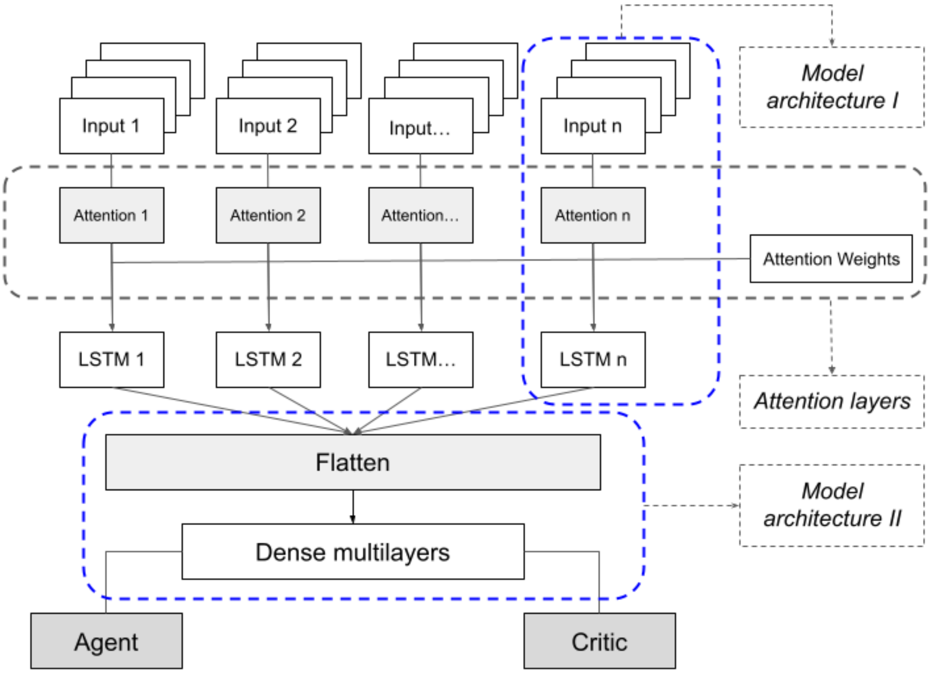


Figure 7.3: General scheme of the agent's architecture with an attention layer.

7.2.5 Model overview

Integrating all components, the research we are addressing involves an RL agent whose objective is to maximize expected cumulative returns by optimizing the action-selection policy within the constraints of the financial market, represented by an MDP. The optimization thus, centers around finding a policy $\pi^*(a_t|s_t)$ that maximizes the expected cumulative return, given by:

$$\pi_\theta^* =_{\pi_\theta} \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t \mathcal{R}(s_t, a_t) \mid \pi_\theta\right] \tag{7.17}$$

Where $\gamma \in (0, 1]$ is the discount factor, emphasizing the preference for immediate rewards over future rewards, and T represents the investment horizon. This formulation starts with a state space $\mathcal{S}$ as a comprehensive set of market indicators hypothesized to impact the price and, hence, the trading decisions, making $\mathcal{S}$ a multi-dimensional space.

Secondly, the action space $\mathcal{A}$ is formulated as a continuum of portfolio allocations across n assets, where each action $a_t \in \mathcal{A}$ at time t is a vector of weights $(w_1, w_2, ..., w_n)$ subject to the constraint $\sum_{i=1}^{n} w_i \leq 1$ and $0 \leq w_i \leq 1$ for each w_i. This represents the proportion of the total portfolio value allocated to each asset.

The reward function $\mathcal{R}(s_t, a_t)$ captures the immediate return from taking action a_t in state s_t and is directly related to the change in portfolio value δW_t as a result of this action. With the probabilistic nature of moving from one state to another after taking an action of $\mathcal{P}(s_{t+1}|s_t, a_t)$. Enhancing the model's explainability, attention mechanisms are integrated to identify which features within the state space $\mathcal{S}$ predominantly influence the selection of actions. This aligns with the goal of developing an automated trading system that not only maximizes returns but also ensures transparency and interpretability, meeting the growing need for XAI in finance.

To summarize, our approach utilizes RL techniques to develop a strategy focused on maximizing expected returns, as detailed in Section 7.2.1. This strategy is refined through the interactions between an agent, as outlined in Section 7.2.2, and a simulated environment, as presented in Section 7.2.3. Furthermore, incorporating attention mechanisms, as described in Section 7.2.4, enhances the model's ability to discern significant market features influencing trading decisions.

7.3 Experimentation

The research utilizes a comprehensive dataset spanning from January 3, 2005, to December 31, 2021, encompassing 4,435 observations. The dataset exhibits a 100 % completeness rate, with no missing values across the observation period. The data source is from a global provider of financial data called Bloomberg. The details of the dataset are shown in Table 7.1

Stock Symbol	Mean Close Price	SD (Price)	Min/Max Close Price	Lowest Price	Highest Price
AIR	50.11	34.19	8.47/139.0	8.12	139.40
BNP	52.84	13.13	20.78/91.6	20.08	92.40
OR	143.66	84.34	46.96/429.8	46.00	433.65
TTE	44.42	6.41	21.8/63.05	21.12	63.40
EL	79.53	38.30	26.28/193.36	26.08	195.00
MC	179.36	152.75	35.32/734.7	34.34	741.60
KER	218.29	184.33	28.86/792.1	28.42	798.00
RMS	330.58	294.16	47.73/1675.5	47.03	1678.00
SAN	69.04	13.92	37.71/100.1	35.86	100.55
SU	60.25	26.96	20.15/173.18	19.42	173.78

Table 7.1: Summary of the data set

Once all the components were assembled and codified, a training process was carried out using 1000 episodes, each comprising a maximum of 200 steps, and each step was equivalent to one day of negotiation. The starting day of each of these episodes was chosen randomly within the training period. Since it is impossible to know the actual price at which the assets would be bought or sold in the market on a given day due to uncertain market conditions, a random value between the lowest and highest price of the respective day was taken to calculate the opening and closing price of the first and last day of the episode, for detailed configurations in this process, refer to Table 7.2.

Once a sample episode has been taken, it iterates through the trading days, where different states are given to the agent. Once received, the agent processes them through a neural network where two values are obtained: the score of the actor and the critic. Although these values let calculate the agent's performance in creating a profitable portfolio, they also allow calculating the return of a simulated investment in the sampled period, which will be used to calculate the agent's reward after the episode is finished. The episode's reward is divided by the initial investment, and then it is added to a running reward, as seen in Equation 7.18, which will be used for further network backpropagation.

Once the training has finished, the explanatory part of the model is examined to obtain information relevant to the importance of the variable. This begins by

Hyperparameter	Value	Hyperparameter	Value
Number of LSTMs	10	Number of common layers	3
Node of LSTMs	32	Number of drop layers	3
Activation of LSTMs	Sigmoid	Nodes of common layers	250, 125, 250
Activation of common layers	LeakyReLU, Sigmoid	Values of drop layers	0.99, 0.8, 0.5
Final layer	Softmax	Optimizer	RMSprop
Learning rate	0.01	Loss function	Huber
Initial budget	1000000	Commission rate	0.5%
Number of Inputs	10	Number of lagged variables	5
Input matrix size	10 x 5	Technical analysis variables	MACD, RSI
Price variables	Open, High, Low Close, Volume	Moving averages periods	14, 21, 100

Table 7.2: Parameter specifications for the RL Model.

collecting the states of each asset, which are subsequently entered into the explanatory layer to perform a partial extraction of the model, obtaining the att_k values previously described above in Equation 7.15. Then, using these values from the explanatory layers, we obtain the softmax activation of a single state, disregarding the lagged variables for this analysis.

$$running\ reward = 0.05 \cdot episode\ reward + (1 - 0.05) \cdot running\ reward \quad (7.18)$$

The environment, agent structure, simulation, and graphs were coded using the Python programming language, and each neural network was coded using Keras [8] and Tensorflow [9]. The whole implementation was carried out in a Google cloud virtual machine instance and a general scheme of the training process can be seen in Figure 7.2.

7.4 Results

After training with 1000 episodes, it is possible to see the running reward progressively increasing. In order to test and prove the trading agent's efficiency, we

---

[8] F. Chollet et al., Keras, https://keras.io, 2015.

[9] M. Abadi, A. Agarwal, P. Barham, et al., TensorFlow: Large-scale machine learning on heterogeneous systems, 2015.

took an out-of-sample period, where the agent is compared against a benchmark, which we defined as an equally weighted portfolio. Additionally, to compare the performance of our proposed portfolio, we added two additional architectures: one without an attention layer and another substituting an LSTM layer with a dense layer to assess their impact on the trading strategy's overall performance. Furthermore, to further compare the results for our proposed portfolio, we incorporated two variants of the Markowitz model: one static version and another that adjusts positions annually. The first model calculates the portfolio weights at the end of the training period, considering the past earnings and volatility to create a portfolio that will not change until the end of the testing set. The second Markowitz model, however, recalculates the portfolio weights at the end of each year during the out-of-sample period, using the most recent data to adjust its allocations. This dynamic approach allows the portfolio to adapt to changes in the market environment and is more realistic in reflecting on how portfolio managers often operate, continuously reassessing and adjusting their investment strategies to align with current market conditions and outlooks.

The RL agent, specifically the one we propose that utilizes LSTM and Attention mechanisms, displays the best performance as indicated by its total cumulative return of 66.70 %, the lowest standard deviation of 0.0114, the shallowest maximum drawdown of -31.15 %, and the highest Sharpe ratio of 0.98. which achieved a total return of 62.60 %, a standard deviation of 0.0117, and a maximum drawdown of -34.72 %, as shown in Table 7.3. Our proposed RL model not only demonstrates superior risk-adjusted returns but also shows enhanced resilience in market downturns. As shows Figure 7.4, the variations of our RL model also outperform traditional and static portfolio management strategies, such as the Annual Rebalanced Markowitz and Static Markowitz portfolios, which posted negative returns and higher volatility, highlighting the limitations of these methods in adapting to market fluctuations.

The LSTM Only, without the attention layer, is better than our benchmark, yet it was not as effective as the model incorporating both LSTM and Attention mechanisms. It secured a total return of 59.08 %, slightly higher than the standard deviation of 0.0147, and encountered a deeper maximum drawdown of -39.18 %. Meanwhile, the RL with a dense layer instead of an LSTM performed with a total return of 63.48 %, with a standard deviation of 0.0144, a maximum drawdown of -38.48 %, and a Sharpe ratio of 0.77.

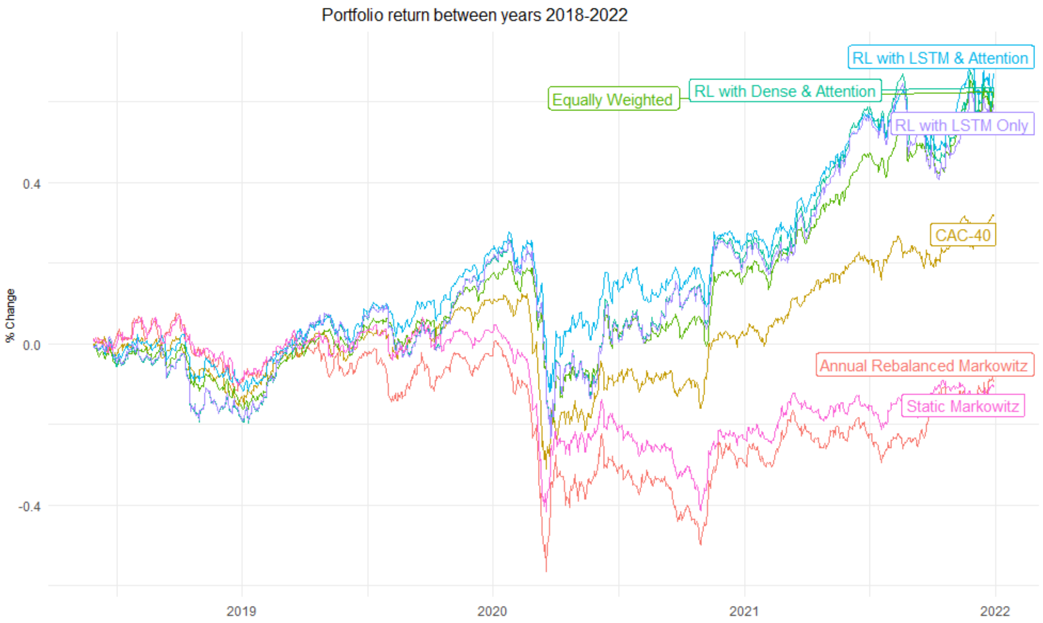


Figure 7.4: Performance of the agent in an out-of-sample period compared to the benchmark and the CAC-40 index.

The RL models have demonstrated superior performance compared to techniques associated with the Markowitz portfolios. The performance of the Annual Rebalanced Markowitz and Static Markowitz portfolios during the out-of-sample period reveals significant differences compared to the other strategies outlined in Table 7.3. The Annual Rebalanced Markowitz portfolio experienced a negative return of 9.60 %, with a standard deviation of returns at 0.0197, the highest among all portfolios examined, indicating higher volatility and risk. Furthermore, its maximum drawdown reached -59.50 % and a Sharpe ratio of -0.09.

Similarly, the Static Markowitz portfolio also encountered negative performance, with a total return of -10.86 %, the lowest among all portfolios examined. Its standard deviation was 0.0147, suggesting a somewhat lower risk profile than the Annual Rebalanced Markowitz portfolio but still higher than most strategies analyzed. The maximum drawdown for the Static Markowitz was -45.48 %, and a Sharpe ratio of -0.12.

Portfolio	Total Return	Standard Deviation	Maximum Drawdown	Sharpe Ratio
RL with LSTM & Attention	**0.6670**	**0.0114**	**-0.3115**	**0.98**
RL with LSTM Only	0.5908	0.0147	-0.3918	0.76
RL with Dense & Attention	0.6348	0.0144	-0.3848	0.77
CAC	0.3154	0.0129	-0.3855	0.60
Equally Weighted	0.6260	0.0117	-0.3472	0.95
Annual Rebalanced Markowitz	-0.0960	0.0197	-0.5950	-0.09
Static Markowitz	-0.1086	0.0147	-0.4548	-0.12

Table 7.3: Table with the main results from the experimentation in the out-of-sample test.

To explain the agent's most important variables, we obtain the data of our explanatory layer of attention located between each of the inputs and each of the networks of LSTM, as shown in Figure 7.3. In order to obtain this descriptive data, a call was made to the agent's neural network, where a partial extraction of each of the attention layers was performed using the data chosen as an out-of-sample. The main idea involves knowing which state values are the most important in the agent's decision-making process, and we can see that these vary according to the different assets and time.

The main advantage of mixing a multi-head structure with independent explanatory layers, like the one we presented in this paper, is that it allows us to extract relevant information from each LSTM independently and discover the critical variables for each asset. Additionally, it should be noted that the structure of the states given to the agent is three-dimensional because, at each step, each LSTM network is fed with ten variables together with its lagged values for four episodes, as explained in Section 7.3. However, in this research, for the purpose of analyzing the explainable values coming from the attention vector att_k, we will only consider those values at time t, which we will call Q values. In this way, we will merely show the importance of the inputs without the lag variables; also, this facilitates the construction of flat two-dimensional schemes that are easier to analyze visually.

If we analyze the Q values, we can see that not all variables have the same importance in every asset, nor is their importance constant over time. For example, in the graphs for AIR and BNP in Figure 7.5, 14-day MA emerges as a significant indicator, reflecting short-term trends and momentum in pricing; therefore, it is possible to assume that the model captures that this stock is more sensitive to short-

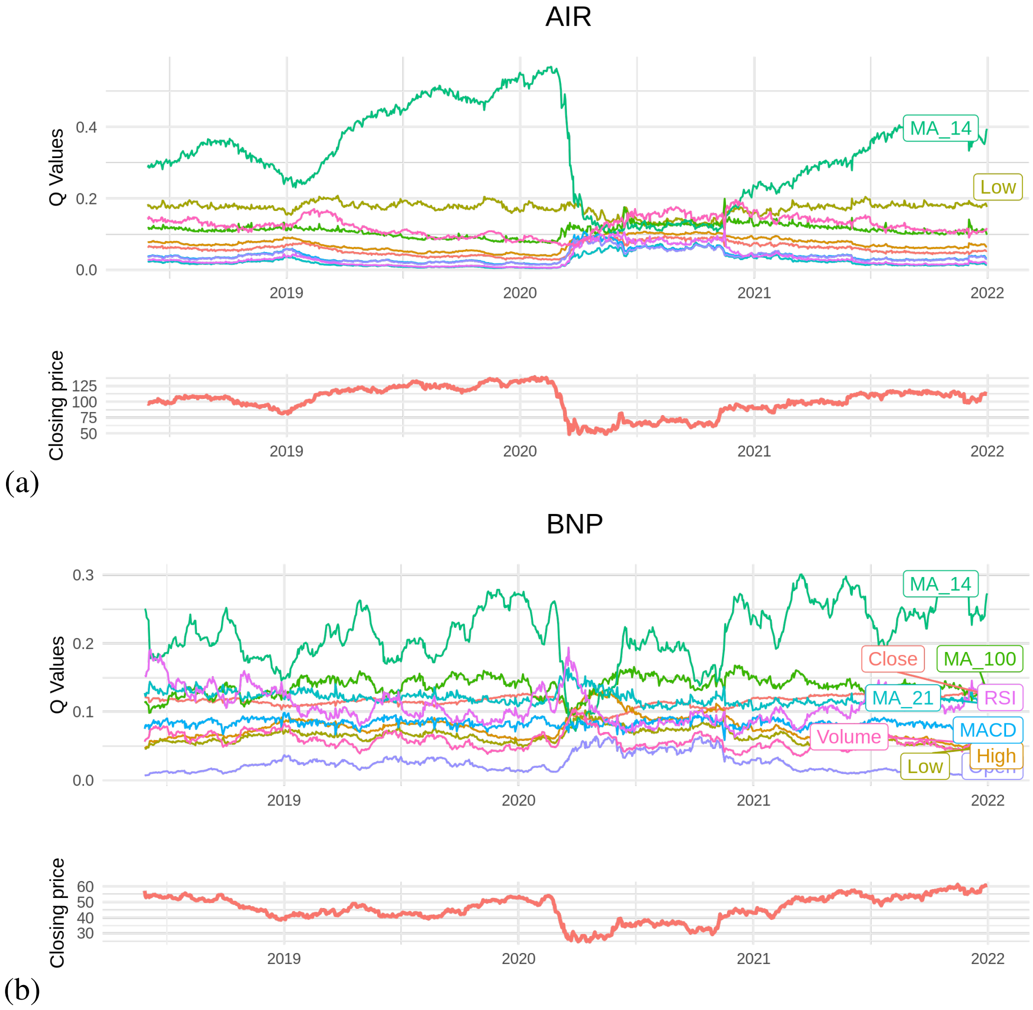


Figure 7.5: Graphical representation of Q values and closing prices in the out-of-sample period for (a) AIR, (b) BNP.

term fluctuations and uses this information to make more informed trading decisions. However, the BNP asset also appears to be affected by a broader spectrum of moving averages and other technical indicators, suggesting a more complex interplay of factors influencing its price. This complexity indicates that the RL model identifies simple trend-following strategies and integrates multiple indicators to assess the overall market context and asset-specific behaviors.

Similarly, the SAN asset considers the 21-day MA more important, followed by the high and closing values, as shown in Figure 7.6. This suggests a sensitivity to trends extending over three weeks. High and closing prices are also notable factors for SAN, indicating that the asset's price movement within a given day can substantially impact investment decisions.

Also, we see that certain LSTMs in some assets consider the price and volume of the assets to be more important than the technical analysis data. For example, in Figures C.3, we see SU, MC, and EL, respectively, illustrating that the model gives considerable weight to raw price and volume data. For EL, by observing the Q values from the given figure, the attention given to the closing price indicates a high value placed on the final price at which the asset settles at the end of the trading day. Following the closing price, the 21-day MA is also given significant importance, suggesting that the model values the trends and momentum established over a longer period. This may imply that the asset is influenced by medium-term trends, which could be representative of market cycles or recurring trading patterns.

In the case of MC, the results indicate an increasing significance of momentum, as measured by the RSI indicator, in the model's decision-making process. This emphasis on the RSI indicator suggests that the model is progressively utilizing it to identify overbought or oversold conditions as critical decision-making points. For SU, the prominence of volume in the Q values suggests a strong correlation between trading volume and price movement, which might indicate that volume spikes precede significant price changes. This can be a sign that the model considers market momentum and liquidity before making decisions, inferring that, in SU's case, volume is a leading indicator of market activity.

Analysis of Figure C.4 further demonstrates the diversity in how the RL model assesses the value of different data types across various assets. The observed variability in data significance indicates the model's ability to recognize and assign

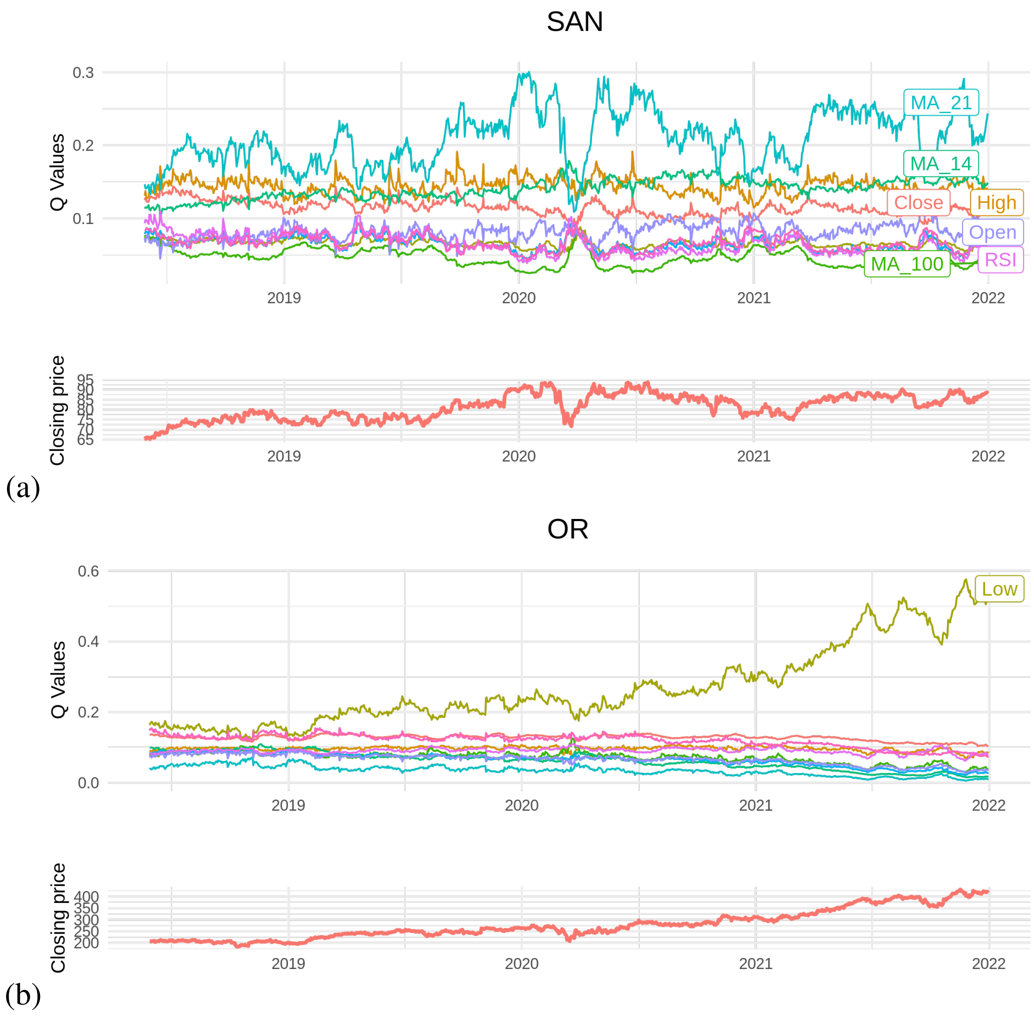


Figure 7.6: Graphical representation of Q values and closing prices in the out-of-sample period for (a) SAN, and (b) OR stock.

importance to different data points uniquely for each asset. In the case of the remaining assets, indicators such as KER, RMS, and TTE exhibit distinct patterns of importance, further highlighting the model's refined grasp of the unique market dynamics associated with each asset. The KER considers the closing price the most critical variable, while RMS is the 21-period MA. This variation in indicator preference across KER and RMS underscores the RL model's capability to discern and adapt to the specific market behaviors and trends relevant to each asset. For KER, prioritizing the closing price may indicate a focus on the final market sentiment at the end of the trading day, which could be a key indicator of the asset's stability or volatility, as is for EL and SU.

On the other hand, RMS's emphasis on the 21-period MA as the most critical variable points to a strategic focus on medium-term trends. This pattern mirrors observations in SAN, alongside AIR and BNP, which also underscore the importance of medium-term trend indicators, notably identifying the 14-period MA as crucial. Finally, the TTE suggests a sophisticated combination of volume analysis, momentum tracking, and trend following, with adaptability to prioritize different indicators as their relevance shifts over time.

7.5 Conclusions

The main goal of this research paper was to explore how an XRL application could create a profitable portfolio in the stock market. It is appropriate and challenging to determine the most predominant variables considered by an RL agent to assist investors, traders, or stakeholders in creating investment portfolios. Therefore, we have developed an explainable RL model in line with this goal that has proven to not only build a stock portfolio effectively with better performance than an equally weighted portfolio but also to show the most relevant features for each asset.

The use of RL can significantly improve the creation of investment portfolios due to its ability to continuously learn and add or remove different variables throughout its implementation, and it is not only influenced by a fixed set of variables. In addition, adding components that help explain the RL agent allows for greater confidence and auditing of certain decisions it may have to make over time. With this research, we have filled a gap in the financial literature by adding an explainable RL agent that can use DL to create portfolios and explain key features simultaneously.

Although our implementation and results propose critical and innovative insights into research and the financial markets, some limitations remain, as with other impactful research. From a practitioner's perspective, implementing an RL model to predict the stock market is challenging due to the market's complex nature. Key hurdles to implementing a trading strategy based on our model include handling vast amounts of data, the computational resources needed for real-time processing, dealing with regulatory obligations, and due diligence processes.

Additionally, the market's unpredictability, driven by factors beyond historical data, can make outcomes uncertain. Despite these difficulties, the potential benefits of optimized trading strategies make it a compelling venture for those in the field. However, we believe that the limitations that emerge from this work may benefit future research lines. First, our research only focuses on ten stocks in a specific market, such as the CAC; however, this exploration can be extended to different markets and regions, further expanding the capacity to create and diversify investment portfolios.

Second, our research has a limited exploration of the input variables taken by the agent and does not conduct an exhaustive review with multiple inputs that could affect the creation of an investment portfolio. The main advantage of ML and RL models is that numerous variables and a significant amount of data can be used. This flexibility also allows it to be extended to be used in Big Data applications, which utilize asset price data and other variables, such as financial market, macroeconomic, or alternative data. Finally, future research can validate our work by applying it to different markets in different periods; however, the application of attention-layered RL to explain the importance of variables in a financial RL application also offers a promising direction in future research.

Third, the implementation shown is primarily tailored for the risk-neutral investor and does not consider other types of investors, such as risk-averse investors. Therefore, future research can consider integrating various risk measures and preferences into the RL model to cater to different investor profiles by adjusting the reward structure to account for the degree of risk aversion and incorporating risk metrics.

8

Conclusions and Future Research

This final chapter summarizes the main contributions and potential future research directions from this thesis while also validating the hypotheses defined at the beginning of the thesis in Section 1.4. Initially, this chapter reviews the results obtained and the significant contributions made in Section 8.1. Following this, in Section 8.2, the chapter outlines the future research lines that have emerged from the experimentations of this thesis.

8.1 Contributions

This doctoral dissertation comprehensively examines various advanced techniques in asset pricing, portfolio construction, and financial market prediction that strongly emphasize ML and AI. Concretely, the research explores the complexities of traditional asset pricing models and portfolio construction techniques, offering an evaluation of these methods with ML and AI algorithms. This exploration includes an extensive exploration of various regression models and their application in predicting financial returns with greater accuracy. Furthermore, the dissertation presents an in-depth analysis of ANNs, highlighting their efficacy in financial time series forecasting.

Additionally, an extensive part of the study is dedicated to clustering techniques and dimensionality reduction, employing methods to cluster complex financial data more effectively. Additionally, RL is another cornerstone of this research, where its elements and applications in finance are examined, particularly in the context of portfolio construction. Also, a pivotal aspect of this dissertation is its focus on the explainability of AI models in finance, addressing the growing need for transparency and interpretability in ML applications.

This research effectively bridges the gap between traditional financial theories and modern computational techniques, setting a foundation to explore the integration of ML and AI in financial decision-making. This integration ensures that these advanced tools are effective and comprehensible. The systems created as part of this study are structured into four distinct modules, aligning with the contributions outlined in this dissertation. The attributes and functionalities of these modules are summarized in the following manner:

- **Neural Network-Based Forecasting of Financial Market Time Series.**

    - By applying an LSTM model stands out due to its unique structure and training methodology, which enable it to capture long-term dependencies effectively. This feature is particularly important for accurate short-term forecasting. This ability distinguishes LSTM from other models like GRU, MLP, basic RNN, and traditional methods like ARIMA, primarily in terms of its high predictive accuracy.
    - Additionally, the application of LSTMs exhibits exceptional ability in analyzing and understanding financial market trends, effectively handling the dynamic nature of stock market data, making them a powerful tool for predictive analysis.
    - Finally LSTMs demonstrate remarkable proficiency in analyzing financial market trends, adept at handling the dynamic nature of stock market data, positioning them as a potent tool for predictive analysis that helps decision-makers.

- **Autoencoder-Enhanced Clustering: A Dimensionality Reduction Approach to Financial Time Series.**
    - The research presents an innovative clustering framework tailored for financial time series data, employing AE to achieve a condensed yet insightful data representation. This innovative approach addresses financial markets' inherent complexities and multidimensional nature, offering a novel solution to clustering financial time series challenges. By leveraging autoencoders for dimensionality reduction, the framework ensures that the vast and intricate data of financial indices can be effectively analyzed and categorized, enhancing the precision and utility of financial models.
    - The detailed evaluation of the proposed framework, achieved by applying diverse dimensionality reduction and clustering algorithms, marks a contribution. This extensive analysis validates the framework's efficacy across various financial indices, such as IBEX, CAC, DAX, SPX, and UKX, showcasing its robustness and flexibility. The extensive testing confirms the framework's reliable applicability under different market scenarios, establishing it as a useful resource for financial analysts and practitioners.
    - The study significantly enhances financial time series data clustering, creating more precise financial predictive models. This progress provides insights for optimizing investment strategies and improving risk management, representing an advancement in financial analysis tools. The potential benefits for investors, portfolio managers, and the broader financial sector are considerable, underscoring the need for ongoing innovation in applying ML to financial data analysis.
- **Application of Explainable Artificial Intelligence in Predicting Digital Asset Prices.**
    - The study develops an XAI model for cryptocurrency price prediction, focusing on BTC and ETH, addressing a marked research gap. Introducing explainability into AI models for financial predictions is crucial

    for demystifying their decision-making process, offering insights into prediction methodologies. This is especially important for cryptocurrencies, which are known for their unpredictability and complex market dynamics.

  - The research enhances the field by identifying optimal analytical techniques and parameters for technical trading predictions in the volatile cryptocurrency market. It offers a method for selecting and fine-tuning predictive models, providing a valuable framework for applying ML in financial markets specifically tailored to the challenges of cryptocurrencies.
  - The study's implications reach beyond academia, providing practical advantages for investors and regulators. The XAI model developed can aid stakeholders in understanding cryptocurrency trends, supporting informed decisions in uncertain economic times. This is important as cryptocurrencies grow in the global financial markets. The findings improve cryptocurrency investment decisions, promoting a more stable and informed financial environment.

- **Portfolio Construction Using Explainable Reinforcement Learning.**
  - The research presents an XRL model tailored for portfolio management, moving from only price prediction to developing actionable trading strategies. This development offers investors and portfolio managers a practical tool, enabling market movement predictions and specific action recommendations. Such a model enhances real-world trading applicability, marking a contribution to financial trading.
  - The study employs the model in a simulated trading environment mirroring the CAC index's conditions, enabling dynamic adaptation to market changes through iterative learning from historical data. This adaptability is key in the volatile and unpredictable financial market, ensuring the model's trading strategies stay relevant and effective over time.
  - The study's empirical results show that the model offers transparency and superior performance over traditional portfolios in out-of-sample

tests. This dual achievement tackles the ’black-box’ issue in financial ML by improving algorithmic strategy planning and enhancing interpretability. It provides a transparent framework for investment portfolio management, advancing the field by linking advanced ML methods with practical financial trading applications.

8.1.1 Hypotheses Validation

This section analyzes the contributions of this dissertation, evaluating whether the four hypotheses stated in Section 1.4 have been validated or not based on the design, implementation, and evaluations conducted throughout the study.

Hypothesis 1:

> “*By using ANN and DL in forecasting financial market trends could result in markedly improved accuracy in predicting market movements. With the utilization of advanced data processing capabilities, this approach is expected to outperform conventional methods in detecting and understanding complex market patterns*.”

To validate this hypothesis, The study deeply analyzed LSTM networks, highlighting their ability to utilize long-term data dependencies, which is key for forecasting financial market trends. LSTM outshone models like GRU, MLP, basic RNN, and ARIMA due to their superior pattern recognition over extended periods.

The empirical evaluation showed LSTMs’ remarkable capability in financial data analysis, establishing them as a potent tool for predictive analysis. This research reinforces ANNs and DL’s role in financial forecasting, underlining LSTMs’ contribution to enhancing prediction accuracy and reliability.

Hypothesis 2:

> “*By developing efficient dimensionality reduction methods using AE can substantially simplify the analysis of high-dimensional financial data. These methods are anticipated to significantly enhance clustering algorithms’ performance by effectively reducing the complexity of financial*

> *time series data. This improvement in data processing is expected to lead to more accurate identification of key market patterns and trends.*"

To validate this hypothesis, the research applied an AE-based dimensionality reduction on multiple financial indices, including IBEX, CAC, DAX, SPX, and UKX, to see if it would improve clustering algorithm performance and enable more accurate market pattern identification. Comprehensive experiments and comparisons with traditional clustering methods showed significant enhancements in clustering granularity and quality with AE. These results confirm that efficient dimensionality reduction can greatly facilitate the analysis of complex financial data, leading to a better understanding of market dynamics. The effectiveness of this approach in distinguishing financial time series categories highlights AE' potential to transform financial data analysis and contribute to more precise and insightful financial models, with significant benefits for investment strategy and risk management.

Hypothesis 3:

> "*This hypothesis asserts that using XAI techniques, particularly SHAP, in cryptocurrency market forecasting will enhance transparency of predictions. Integrating an explainable method is expected to unravel the AI's decision-making process, balancing advanced predictive power with user understandability and trust. This enhancement facilitates a clearer understanding of the key variables influencing decision-making..*"

To validate this hypothesis, developing and applying an XAI model tailored to the cryptocurrency market, focusing on the dynamics during volatile periods, like the COVID-19 pandemic and the 2008 financial crisis. The model aimed to elucidate the drivers of cryptocurrency price changes using XAI techniques, notably SHAP, to enhance predictive accuracy and user trust by making AI decisions transparent.

The model successfully predicted cryptocurrency prices, focusing on BTC and ETH, and identified key influencing variables. The models showed their ability to handle the market's volatility, particularly during economic downturns. The results confirm that using XAI techniques enhances transparency in the cryptocurrency market, providing insights into analytical methods and parameter optimization for trading. This advancement tackles the traditional ML models' lack of clarity, facilitating better decision-making, which is important in periods of economic uncertainty.

Hypothesis 4:

> "*This hypothesis suggests that utilizing RL models for portfolio management with explainable components can enhance transparency and greatly improve portfolio optimization compared to traditional methods. The RL model can adjust to intricate and evolving environments and is expected to outperform conventional portfolio management strategies by effectively identifying investment opportunities. By combining effectiveness and explainability, this RL models are expected to build trust with stakeholders, gaining a deeper insight into the relevant variables that impact decision-making.*"

To validate this hypothesis, a solution was created that integrates an XRL model into financial portfolio management, employing stacked LSTM networks with attention mechanisms for predicting asset allocation strategies. This method leverages the LSTM's capacity for understanding temporal financial data and attention layers to identify critical predictive elements. Implemented in a simulated setting that mirrors the CAC index, the model adjusts to diverse financial scenarios by learning from past market trends.

Out-of-sample tests reveal the model's trading strategy substantially outperforms a standard equally weighted portfolio and traditional portfolios, affirming the hypothesis through evidence of the RL model's skill in managing financial market intricacies and identifying investment opportunities. Including explainable components within the model addresses a significant void in financial ML by enhancing transparency and building trust by explaining its decision-making framework.

This research represents an advancement in financial portfolio management, demonstrating that an XRL model can compete with and outperform conventional portfolio creation approaches. The fusion of high performance, explainability, and adaptability highlights the XRL in financial decision-making, providing investors and portfolio managers with advanced, transparent, and efficient tools for optimizing investment portfolios in complex market environments.

8.1.2 Limitations

This section outlines the limitations of the proposed methods:

- **Neural Network-Based Forecasting of Financial Market Time Series.**

  The study's use of multiple neural network architectures (MLP, RNN, LSTM, GRU) for modeling displays a comprehensive approach. However, training solely on historical data may overlook the financial markets' unpredictable nature, influenced by other factors such as political events and market sentiment. While having the lowest error rate, the LSTM network presents significant computational demands, highlighting potential scalability issues for real-time trading. Focusing on a single financial index could also restrict the models' applicability to different markets or instruments. The lack of discussion on overfitting, validation beyond the training dataset, transaction costs, and market impact points to possible practical challenges. Additionally, the necessity for continuous model updates to keep pace with market changes indicates a need for continued adaptation.

- **Autoencoder-Enhanced Clustering: A Dimensionality Reduction Approach to Financial Time Series.**

  Introducing a novel clustering framework that uses AE to enhance the granularity and quality of financial time series clustering marks a significant advancement in financial data analysis. However, this approach also encounters several limitations that deserve consideration. Firstly, the effectiveness of AE in compressing and representing financial time series data centers on the selection of hyperparameters and the architecture of the AE itself. This complexity can introduce a steep learning curve and necessitates a fine-tuning process that might not be straightforward in all financial contexts.

  Additionally, the inherent non-stationarity and high volatility of financial markets can make clustering financial indices challenging, as the underlying dynamics of these markets are constantly evolving. This evolution can induce a once-optimal clustering model that is less effective over time, suggesting the need for continuous reevaluation and adjustment of the model.

The multidimensional nature of financial data also implies that while the model may perform well on the indices studied (IBEX, CAC, DAX, SPX, and UKX), its applicability and performance on other financial instruments or in different market conditions remain to be validated. Moreover, the study's focus on clustering might overlook the causal relationships between different financial indices, which could offer deeper insights into market dynamics.

Lastly, while enhancing clustering methodologies aids in refining financial predictive models, the practical implementation of these insights into investment strategies and risk management requires careful consideration of transaction costs, regulatory constraints, and the adaptability of these models in real-time trading environments. The balance between model complexity, interpretability, and operational efficiency remains a critical challenge for leveraging advanced ML techniques in the financial sector.

- **Application of Explainable Artificial Intelligence in Predicting Digital Asset Prices.**

  The development of an XAI model for predicting cryptocurrency prices, particularly during periods of economic uncertainty like the COVID-19 pandemic or the 2008 financial crisis, represents an advancement in both financial technology and the broader application of AI in uncertain markets. However, this approach is not without its limitations and challenges.

  One primary limitation is cryptocurrencies' inherent unpredictability and volatility, which can be caused by many factors beyond historical price movements, including regulatory changes, market sentiment, technological advancements, and macroeconomic indicators. While the model aims to provide a more transparent and understandable framework for prediction, its accuracy and reliability are ultimately contingent on the quality and relevance of the input data, which can change rapidly and be unpredictably in the case of cryptocurrencies.

Additionally, the model's effectiveness during economic uncertainties might not necessarily translate to stable economic periods, suggesting a potential limitation in its adaptability or the need for adjustment under different market conditions. The XAI, while addressing the 'black box' issue, also requires a balance between the simplicity and the complexity of the models necessary to capture the complexities of cryptocurrency markets. Over-simplification could lead to the omission of critical factors, while too much complexity may reduce the model's accessibility and interpretability for practitioners.

Another challenge is the model's generalization capabilities. While it is optimized for BTC and ETH, the two most valuable cryptocurrencies, its performance on other cryptocurrencies with different market behaviors and liquidity levels may vary. This limitation raises questions about the scalability of the approach and its applicability across the diverse and growing array of digital currencies.

Lastly, the dependence on technical trading indicators and historical data may not fully account for external shocks or unforeseen events, which is particularly relevant in economic crises. The model's predictions are as good as the data and the assumptions underlying its design. Therefore, while it offers valuable insights and a step towards more transparent predictive models in cryptocurrency, users must remain conscious of these limitations and be cautious in their decision-making processes.

- **Portfolio Construction Using Explainable Reinforcement Learning.**

  The introduction of an XRL portfolio management model represents a notable advancement in integrating ML within the financial trading domain, especially addressing the critical challenges of algorithmic transparency and explainability. Despite its innovative approach and the promising results in outperforming traditional methods to create portfolios in out-of-sample tests, there are inherent limitations and challenges associated with this methodology.

  Firstly, while the model's ability to adapt dynamically to market changes in a custom trading environment is challenging and extensive, the specificity of this environment may limit the generalization of the findings. Financial markets vary considerably in volatility, liquidity, and the types of assets traded. The model's performance and applicability might differ under different market conditions or with other financial indices, requiring further validation across diverse trading environments to ensure its robustness and versatility.

Another challenge lies in the balance between explainability and performance. While the model aims to provide actionable trading strategies with increased transparency, the simplification required to achieve explainability might compromise the model's ability to capture the complex, nonlinear dynamics that characterize financial markets. This balance is crucial, as overly simplistic models may not fully leverage the predictive power of ML. In contrast, overly complex models may confuse users and stakeholders.

Moreover, the dependence on historical data for iterative learning, while fundamental to RL approaches, may not always accurately predict future market movements, particularly in the face of unprecedented events or market shocks. The model's capacity to adapt to new, unforeseen market conditions remains an open question, highlighting the need for ongoing adaptation and refinement of the model.

Finally, while the study makes significant progress toward algorithmic transparency, the practical implementation of such models in real-world trading strategies requires careful consideration of regulatory compliance, ethical considerations, and the potential risk impact. The deployment of automated trading strategies, especially those capable of executing high volumes of trades based on ML predictions, must be managed cautiously to avoid unintended consequences such as market manipulation or systemic risk.

8.2 Future Research Lines

This section will cover and describe the future research lines opened by this doctoral thesis, with complex challenges, promising opportunities, and intriguing questions that merit additional exploration.

8.2.1 Realistic Simulations

The development of realistic simulations for training and testing different intelligent models stands out as a compelling area of exploration. Proper simulations can provide a controlled yet realistic and versatile environment for ML models and AI implementations to learn from various scenarios, including those that are rare or difficult to replicate in the real world. The importance of this advancement is particularly significant in finance, where the capability to accurately model and predict market behaviors under diverse conditions is key.

In finance, realistic simulations can improve the predictive power and reliability of AI models, allowing for testing financial models not only against historical but also with hypothetical future market scenarios, including extreme market volatility or rare economic events, without the risk of actual financial loss. This capability can ensure the development of robust financial strategies, risk management protocols, and investment models resilient to market uncertainties.

Therefore, there is an opportunity and need to simulate more realistic scenarios that consider multiple hypothetical events and variables and not only rely on past data. This can be achieved by creating scenarios with multi-modal synthetic data from different sources. In this context, generative AI systems can generate diverse and realistic data, enabling more comprehensive scenario analysis. In this context, Generative Adversarial Network (GAN)s emerge as a powerful tool for recreating time series data (Liao et al. 2023), significantly enhancing the training of ML models and facilitating the simulation of diverse scenarios. This ANN can generate highly realistic time series data, mimicking the complexities and patterns in actual financial markets or other time-dependent data domains. By utilizing GAN, researchers can vastly improve the robustness of ML models. This is particularly useful in finance, where GANs can simulate market conditions under various scenarios, including extreme volatility, economic shifts, and other critical factors that impact financial decision-making.

8.2.2 Multiple Inputs

The importance of incorporating multiple sources of information for ML models in finance cannot be overstated. The dependence solely on historical data in finance carries the risk of overlooking the unforeseen events of financial markets, which are often influenced by multiple factors, such as financial factors and economic and political events.

Integrating news and using Natural Language Processing (NLP) can allow models to process and analyze the subtleties of market sentiment, investor perceptions, and the potential implications of current events on future market movements. This enriched data input can enable models to develop a more holistic and dynamic understanding of market conditions, enhancing their predictive capabilities. For instance, sentiment analysis, derived from social media and financial news platforms, can offer early signals of market movements, enabling proactive rather than reactive responses.

Furthermore, integrating such various data sources can help mitigate the risk of models becoming excessively reliant on historical patterns, which may not always indicate future performance in the face of unprecedented events or shifts in market sentiment. Incorporating the immediate impacts of economic and political events makes ML models more adaptable and resilient, enabling them to anticipate and respond to the dynamic financial market landscape. This approach to data integration can enhance financial forecast accuracy, asset valuation, and risk management strategies.

Additionally, the use of NLP can improve the explainability of models by enabling them to process and interpret textual information. This not only enhances the transparency of the decision-making process but also allows financial professionals to gain a deeper understanding of the variables influencing model predictions.

8.2.3 Risk Management

Finance is witnessing a transformative shift towards decentralized finance, and the development of ML models and AI systems can not be underestimated in this evolution. These technologies are crucial in mitigating systemic risk by enhancing risk management, transparency, and efficiency in the decentralized financial ecosystem. However, it also increases the risk of potential vulnerabilities and the need for cybersecurity measures to protect the system from potential dangers. Therefore, ML and AI in finance must be accompanied by continuous monitoring, ethical considerations, and ongoing research to ensure trustworthy and secure integration into the evolving financial markets.

Future research must address risks based on historical returns, systemic risk, and simulated scenarios, considering the evolving financial field that intertwines multiple markets, including those that are neither centralized nor regulated. This approach to risk assessment can improve AI systems by enhancing their accuracy and ability to mitigate emerging risks.

8.2.4 Language Models

The surprising progress in NLP has facilitated the development of Large Language Model (LLM)s with considerable success. The use of these techniques for developing AI models in Finance has the potential for more sophisticated and context-aware algorithms that can extract valuable insights from unstructured text data, allowing for the improvement of data-driven strategies and the expansion of datasets that can be used by integrating data that was previously unexplored.

Additionally, LLMs can improve how AI models interact with users of predictive models by enhancing NLP understanding and generation of human-like conversations. This results in more effective and user-friendly interactions, making it easier for users to input queries, interact, receive explanations, and potentially gain insights from the models, thereby improving the overall user experience in various applications.

Therefore, integrating LLMs with financial data marks a key step towards achieving the full potential of AI in Finance, offering extensive data analysis capabilities and improved user interaction. This synergy between advanced NLP techniques and financial modeling is set to improve how financial institutions, investors, and analysts approach data-driven decision-making, making it more efficient.

8.2.5 Multi-agents

Multiple agents can significantly enhance AI system capabilities, enabling collaborative work to facilitate tasks and improving efficiency across multiple domains. In the context of RL, the decision-making process controlled by a single agent can be improved by incorporating multiple agents. This addition can allow the system to consider multiple inputs, explore diverse strategies, and collaborate in decision-making, eventually guiding to better learning in dynamic environments.

Implementing this multi-agent method in Finance can optimize trading strategies and risk management. Moreover, integrating data from multiple sources in different ways is becoming increasingly important in evolving financial markets. However, this task is not straightforward, as it requires training and coordinating a network of agents, each designed to analyze specific aspects or information from the market or execute distinct strategies.

Therefore, one promising line of research in the field of Finance and AI is the one that aims to tackle the technical complexity of coordinating multiple agents, ensuring data quality from various sources, and handling the computational demands of processing large volumes of information in real-time. Also, using learning agents with LLMs in simulated environments using multiple inputs appears to hold potential for advancing our understanding of complex systems in Finance.

A set of learning agents, each equipped with LLM capabilities, could be designed to specialize in different aspects of the financial markets, such as analyzing market sentiment from news or speech, monitoring economic indicators, or executing trading strategies based on different learned patterns. When operating in a simulated environment, these agents can interact with each other in a simulated market, allowing them to learn from the environment and from each other, thereby improving their decision-making processes over time. In this simulation, the agents can also create code and test against different benchmarks to test it in different tasks.

By adding realistic simulations of complex market scenarios, the agents can learn how to manage risk by applying traditional risk tools such as stress testing under extreme market conditions and assessing the systemic risk of rare events. Additionally, integrating LLMs into this multi-agent setup enhances the agents' ability to understand language inputs or computer code, making it possible to incorporate unstructured data from multiple sources into their analysis.

By allowing a group of agents to learn autonomously within a controlled and evolving environment, equipped with tools for data retrieval, and allowing them to explore ideas without constraints while performing tasks under surveillance and auditing, we are not only enhancing processes within the financial industry but also progression towards its complete automation. This advancement could imply a paradigm shift, where the financial markets are no longer just a tool for exchanging goods between humans but rather a dynamic market where machines trade resources between them without the intervention of humans.

APPENDIX A

Acronyms

A2C Advantage Actor-Critic

ADA Cardano

AE Autoencoders

AF Activation Function

AGGLO Agglomerative Clustering

AI Artificial Intelligence

ANN Artificial Neural Networks

APT Arbitrage Pricing Theory

ARIMA Auto-Regressive Integrated Moving Average

BiLSTM Bidirectional LSTM

BIRCH Balanced Iterative Reduction and Clustering by Hierarchy

BCOM Bloomberg Commodity Index

A. Acronyms

BNB	Binance Coin
BTC	Bitcoin
BTS	BitShares
CAC	Cotation Assistée en Continu 40
CALI	CALInski and Harabasz
CAPM	Capital Asset Pricing Model
CATR	CatBoost Regressor
CP	Closing Prices
DAX	Deutscher Aktienindex 30
DL	Deep Learning
DOGE	Dogecoin
DQL	Deep Q-Learning
DTW	Dynamic Time Warping
DTR	Decision Tree Regressor
EMA	Exponential Moving Average
EMH	Efficient Market Hypothesis
ESVR	Epsilon-Support Vector Regressor
ETH	ETHereum
EUC	Euclidean Distance
FFT	Fast Fourier Transform
GAN	Generative Adversarial Network
GARCH	Generalized Autoregressive Conditional Heteroskedasticity

GRU Gated Recurrent Unit

HTF Hyperbolic Tangent Function

IBEX Índice Bursátil Español 35

iPCA incremental Principal Component Analysis

KNN K-Nearest Neighbor

KNN-EUC K-Means Clustering Algorithms with Euclidean Distances

KNN-DTW K-Means Clustering Algorithms with Dynamic Temporal Warping

KNNR K-Nearest Neighbor's Regressor

LLM Large Language Model

LGBM LightGBM

LSTM Long-Short Term Memory

LTC Litecoin

MA Moving Average

MACD Moving Average Convergence Divergence

MAE Mean Absolute Error

MAID MaidSafeCoin

MDP Markov Decision Process

ML Machine Learning

MLP Multilayer Perceptron

MNBT MiniBatch Clustering

MSE Mean Square Error

NLP Natural Language Processing

NMC Namecoin

NXT Nxt

OBV On-Balance Volume

PCA Principal Component Analysis

PG Policy Gradients

PPC Peercoin

R_2 Coefficient of Determination

RELU Rectified Linear Unit

RFR Random Forest Regressor

RL Reinforcement Learning

RMW Random Matrix of Weights

RNN Recurrent Neural Network

RSI Relative Strength Index

SF Sigmoid Function

SH SilHouette coefficient

SMA Simple Moving Average

SPX Standard and Poor's 500

SHAP SHapley Additive exPlanations

SO Stochastic Oscillator

SPCT Spectral Clustering

SVM Support Vector Machine

SLSTM Stacked LSTM

UKX Financial Times Stock Exchange 100

USDC USD Coin

USDT Tether

VIX Volatility Index

WR Williams %R

WMA Weighted Moving Average

XAI Explainable Artificial Intelligence

XAU PHLX Gold/Silver Sector Index

XGBR eXtreme Gradient Boosting Regressor

XLM Stellar

XRL Explainable Reinforcement Learning

APPENDIX

B

Supplementary Analysis and Extended Insights from Chapter 6

The appendix chapter contains supplementary information and detailed insights derived from Chapter 6. This inclusion is designed to provide an extended analysis and a deeper understanding of the concepts, methodologies, and findings discussed in that chapter, offering additional data that support and elaborate on the main contents.

B.1 Input Variables for the ML Models

In the following appendix, we present the input variables used for our models to predict the prices of ETH and BTC. These variables have been carefully selected based on their relevance to the cryptocurrency market and their potential impact on price movements.

$$\{Open|High|Low|Close|AdjClose|Volume\}x\{Crypto\,Currencies\}$$

$$\{Close|Volume\}x\{CAC|DAX|SPX|UKX|NDX|\}$$

$$\{Close\}x\{BCOM|VIX|XAU\}$$

$$\{Moving\,averages\,5, 10, 20, 50, 100, 200\}x\{BTC|ETH\}$$

$$\{RSI|STOCH|WILLIAMS|MACD|OBV\}x\{BTC|ETH\}$$

$$\{Lagged\,close\,price\,1, 2, ..., 8\,days\}x\{BTC|ETH\}$$

$$\begin{aligned}\{\text{Crypto Currencies}\} = \{&\text{XRP|LTC|ETH|DOGE|PPC|}\\ &\text{BTS|XLM|NXT|MAID|NMC|BTC|}\\ &\text{USDT|ADA|BNB|USDC}\}\end{aligned}$$

B.2 Search Spaces for the Optimal Set of Hyperparameters

This section presents an extensive set of tables detailing the search spaces for the optimal hyperparameter combinations for each ML algorithm used in our study. These tables serve as a comprehensive guide to understanding the varied hyperparameter configurations, which differ according to the specific algorithms. These configurations were thoroughly explored during the model selection phase of our research.

The primary aim of rigorously evaluating these potential hyperparameter combinations was to finely tune each algorithm. This optimization process was crucial to achieve the highest possible accuracy and performance in predicting the closing prices of ETH and BTC one day ahead. By methodically identifying the most effective hyperparameters, we aimed to enhance the predictive capabilities of each algorithm, thereby contributing valuable insights to the field of financial market forecasting.

Algorithm	Hyperparameter	Search Space
ESVR	C	1.0-1.9
	Kernel	Linear, poly, rbf, sigmoid
	tol	1e-3, 1e-4, 1e-5
	Gamma	Scale, auto
	Epsilon	0.01, 0.1, 0.2
KNNR	n-neighbors	2-20
	Weights	Uniform, distance
	Algorithms	Auto
	Leaf size	2-50
	p	1-100
	Metric	Minkowski
	Metric param.	None
DTR	Splitter	Best, random
	Maximum depth	None
	Maximum features	None
	Maximum samples leaf	1-100
	Maximum weight fraction leaf	0.0, 0.1, 0.2, 0.3, 0.4, 0.5
	Maximum leaf nodes	None
RFR	Number of estimators	10-200
	Max. depth	None
	Minimum samples leaf	2-20
	Minimum weight fraction leaf	0
	Max. leaf nodes	None
	Bootstrap	True
	Random state	0
	Ccp. alpha	0
	Minimum samples split	1-10
	Maximum features	Auto
	Minimum impurity decrease	0
	Oob. score	False
	Number of jobs	-1
	Max. samples	None

APPENDIX

C

Supplementary Analysis and Extended Insights from Chapter 7

This appendix section encompasses additional information and in-depth insights pertaining to Chapter 7. Its purpose is to show the detailed structures for model architecture I and II, shown in Figure 7.3. These images are part of the graphical representation of the ANN model created in Python by the Keras library.

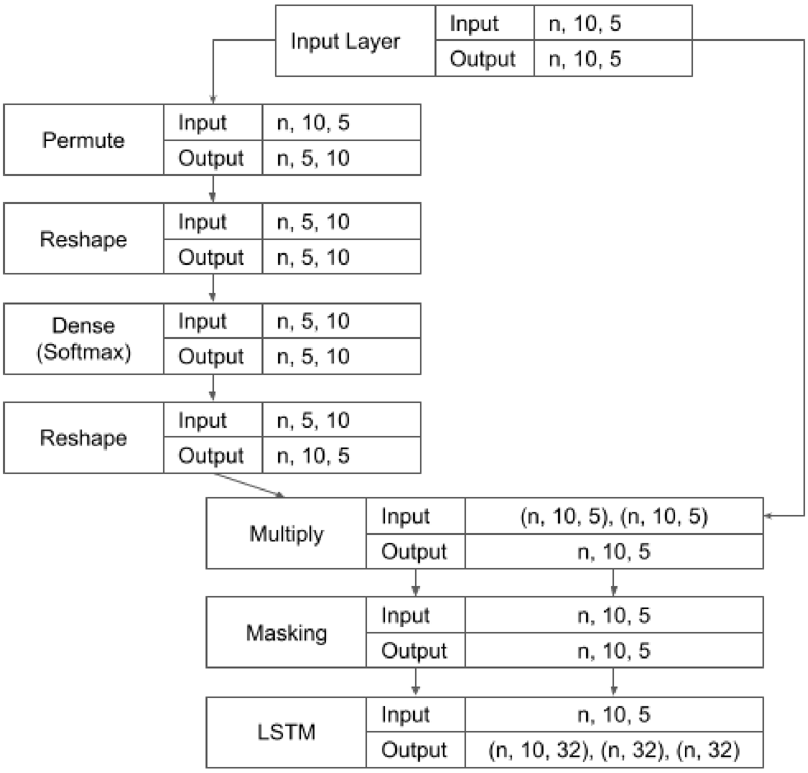


Figure C.1: General scheme of the agent's architecture with an attention layer.

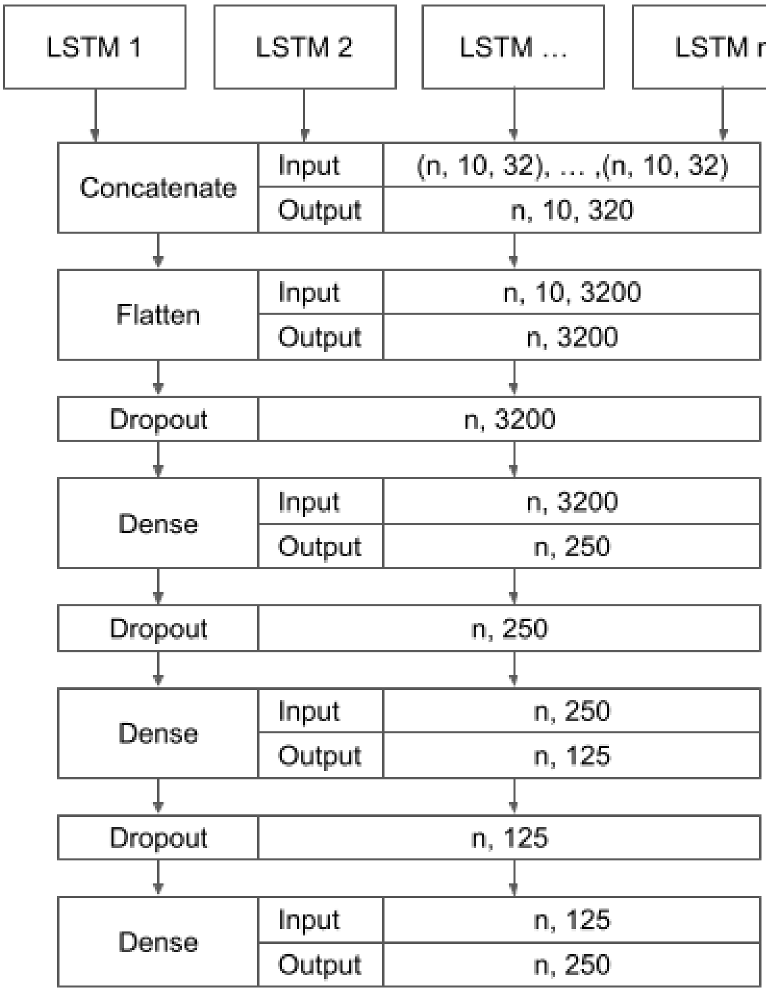


Figure C.2: General scheme of the agent's architecture with an attention layer.

This appendix also shows the graphs that display Q values and closing prices in the out-of-sample period that are not shown in the previous Section 7.4. The graphs are for the assets EL, MC, SU, KER, RMS, and TTE.

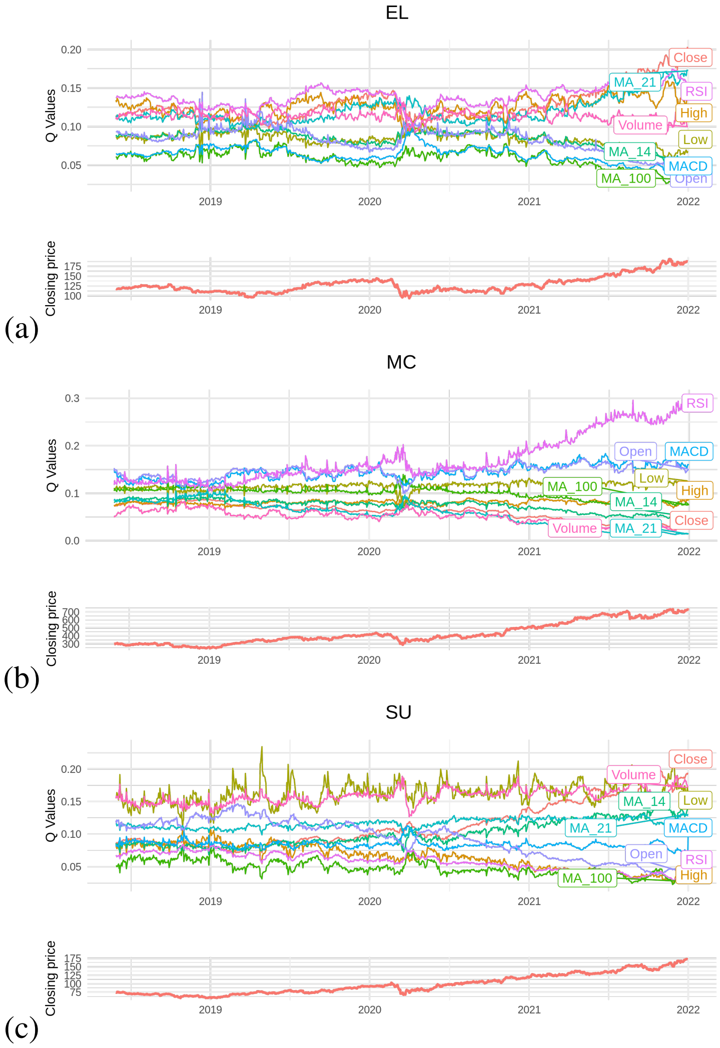


Figure C.3: Graphical representation of Q values and closing prices in the out-of-sample period for (a) EL (b) MC, and (c) SU stock.

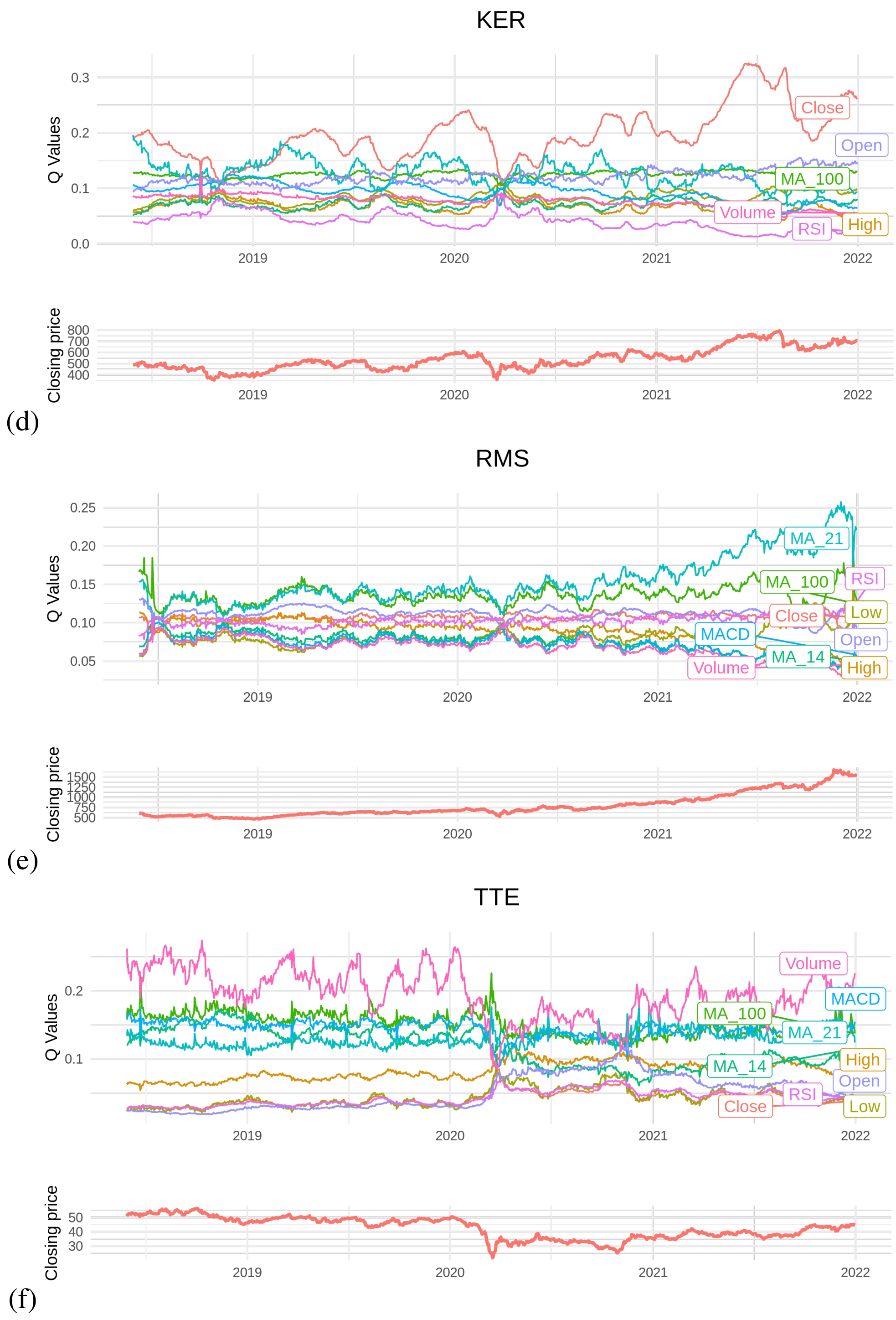


Figure C.4: Graphical representation of Q values and closing prices in the out-of-sample period for (a) KER (b) RMS, and (c) TTE stock.

APPENDIX

D

Bibliography

Bibliography

Abdelkawy, Rasha, Walid M. Abdelmoez, and Amin Shoukry (2021). «A synchronous deep reinforcement learning model for automated multi-stock trading». In: *Progress in Artificial Intelligence* 10.1, pp. 83–97. ISSN: 2192-6360. DOI: `10.1007/s13748-020-00225-z`.

Abedin, Mohammad Zoynul, Mahmudul Hasan Moon, M Kabir Hassan, and Petr Hajek (2021). «Deep learning-based exchange rate prediction during the COVID-19 pandemic». In: *Annals of Operations Research*, pp. 1–52. DOI: `https://doi.org/10.1007/s10479-021-04420-6`.

Aboussalah, Amine Mohamed and Chi-Guhn Lee (2020). «Continuous control with Stacked Deep Dynamic Recurrent Reinforcement Learning for portfolio optimization». In: *Expert Systems with Applications* 140, p. 112891. ISSN: 0957-4174. DOI: `10.1016/j.eswa.2019.112891`.

Adebiyi, Ayodele Ariyo, Aderemi Oluyinka Adewumi, and Charles Korede Ayo (2014). «Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction». In: *Journal of Applied Mathematics* 2014, pp. 1–7.

Aghabozorgi, Saeed, Ali Seyed Shirkhorshidi, and Teh Ying Wah (2015). «Time-series clustering – A decade review». In: *Information Systems* 53, pp. 16–38. ISSN: 0306-4379.

Alaka, Hafiz A. et al. (2018). «Systematic review of bankruptcy prediction models: Towards a framework for tool selection». In: *Expert Systems with Applications* 94, pp. 164–184. ISSN: 0957-4174.

Alom, Md Zahangir et al. (May 2019). «A State-of-the-Art Survey on Deep Learning Theory and Architectures». In: *Electronics* 8.3, p. 292.

Amirshahi, Bahareh and Salim Lahmiri (2023). «Investigating the effectiveness of Twitter sentiment in cryptocurrency close price prediction by using deep learning». In: *Expert Systems*, e13428. DOI: `https://doi.org/10.1111/exsy.13428`.

Andrychowicz, Marcin et al. (2017). «Hindsight Experience Replay». In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc.

Arrow, Kenneth Joseph et al. (1974). *Essays in the theory of risk-bearing*. Vol. 121. North-Holland Amsterdam.

BIBLIOGRAPHY

Atsalakis, George S., Ioanna G. Atsalaki, Fotios Pasiouras, and Constantin Zopounidis (2019). «Bitcoin price forecasting with neuro-fuzzy techniques». In: *European Journal of Operational Research* 276.2, pp. 770–780. DOI: `https://doi.org/10.1016/j.ejor.2019.01.040`.

Bagnara, Matteo (2022). «Asset Pricing and Machine Learning: A critical review». In: *Journal of Economic Surveys*.

Baldi, Pierre (2011). «Autoencoders, Unsupervised Learning and Deep Architectures». In: *Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning Workshop - Volume 27*. UTLW'11. Washington, USA: JMLR.org, pp. 37–50.

Bansal, Trapit, David Belanger, and Andrew Mccallum (2016). «Ask the GRU». In: *Proceedings of the 10th ACM Conference on Recommender Systems - RecSys 16*.

Barredo Arrieta, Alejandro et al. (2020). «Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI». In: *Information Fusion* 58, pp. 82–115. ISSN: 1566-2535. DOI: `10.1016/j.inffus.2019.12.012`.

Bastos, João A. and Sara M. Matos (2022). «Explainable models of credit losses». In: *European Journal of Operational Research* 301.1, pp. 386–394. DOI: `https://doi.org/10.1016/j.ejor.2021.11.009`.

Baur, Dirk (2012). «Financial contagion and the real economy». In: *Journal of Banking & Finance* 36.10, pp. 2680–2692.

Baur, Dirk G. and Thomas Dimpfl (2018). «Asymmetric volatility in cryptocurrencies». In: *Economics Letters* 173, pp. 148–151. DOI: `https://doi.org/10.1016/j.econlet.2018.10.008`.

Bengio, Y. (2000). «Probabilistic neural network models for sequential data». In: *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*. DOI: `10.1109/ijcnn.2000.861438`.

Bengio, Y., P. Simard, and P. Frasconi (1994). «Learning long-term dependencies with gradient descent is difficult». In: *IEEE Transactions on Neural Networks* 5.2, pp. 157–166.

Bergstra, James and Yoshua Bengio (2012). «Random Search for Hyper-Parameter Optimization». In: *J. Mach. Learn. Res.* 13.null, pp. 281–305.

Betancourt, Carlos and Wen-Hui Chen (2021). «Deep reinforcement learning for portfolio management of markets with a dynamic number of assets». In: *Expert Systems with Applications* 164, p. 114002. ISSN: 0957-4174. DOI: `10.1016/j.eswa.2020.114002`.

Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag. ISBN: 0387310738.

Bouhlel, Jihéne et al. (2018). «Comparison of common components analysis with principal components analysis and independent components analysis: Application to SPME-GC-MS volatolomic signatures». In: *Talanta* 178, pp. 854–863. ISSN: 0039-9140.

Brim, Andrew (2020). «Deep Reinforcement Learning Pairs Trading with a Double Deep Q-Network». In: *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0222–0227. DOI: `10.1109/CCWC47524.2020.9031159`.

Bussmann, Niklas, Paolo Giudici, Dimitri Marinelli, and Jochen Papenbrock (2020). «Explainable AI in Fintech Risk Management». In: *Frontiers in Artificial Intelligence* 3. DOI: `https://doi.org/10.3389/frai.2020.00026`. URL: `https://www.frontiersin.org/article/10.3389/frai.2020.00026`.

Bustos, O and A. Pomares-Quimbaya (2020). «Stock market movement forecast: A Systematic review». In: *Expert Systems with Applications* 156, p. 113464. ISSN: 0957-4174.

Cai, Fan, Nhien-An Le-Khac, and Tahar Kechadi (2016). *Clustering Approaches for Financial Data Analysis: a Survey*. arXiv: `1609.08520 [q-fin.GN]`.

Cao, Weipeng, Xizhao Wang, Zhong Ming, and Jinzhu Gao (2018). «A review on neural networks with random weights». In: *Neurocomputing* 275, pp. 278–287. ISSN: 0925-2312.

Che, Zhengping, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu (2018). «Recurrent Neural Networks for Multivariate Time Series with Missing Values». In: *Scientific Reports* 8.1.

Chen, Yingjun and Yongtao Hao (2017). «A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction». In: *Expert Systems with Applications* 80, pp. 340–355. DOI: `https://doi.org/10.1016/j.eswa.2017.02.044`.

Chen, Zheshi, Chunhong Li, and Wenjun Sun (2020). «Bitcoin price prediction using machine learning: An approach to sample dimension engineering». In: *Journal of Computational and Applied Mathematics* 365, p. 112395. DOI: `https://doi.org/10.1016/j.cam.2019.112395`.

Cheong, Donghyun, Young Min Kim, Hyun Woo Byun, Kyong Joo Oh, and Tae Yoon Kim (2017). «Using genetic algorithm to support clustering-based portfolio optimization by investor information». In: *Applied Soft Computing* 61, pp. 593–602. ISSN: 1568-4946. DOI: `10.1016/j.asoc.2017.08.042`.

Cho, Kyunghyun et al. (2014). «Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation». In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Chung, Junyoung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio (2014). «Empirical evaluation of gated recurrent neural networks on sequence modeling». English (US). In.

Corbet, Shaen, Veysel Eraslan, Brian Lucey, and Ahmet Sensoy (2019). «The effectiveness of technical trading rules in cryptocurrency markets». In: *Finance Research Letters* 31, pp. 32–37. DOI: `https://doi.org/10.1016/j.frl.2019.04.027`.

Dash, Ranjan Kumar, Tu N Nguyen, Korhan Cengiz, and Aditi Sharma (2021). «Fine-tuned support vector regression model for stock predictions». In: *Neural Computing and Applications*, pp. 1–15. DOI: `https://doi.org/10.1007/s00521-021-05842-w`.

Dempster, M.A.H. and V. Leemans (2006). «An automated FX trading system using adaptive reinforcement learning». In: *Expert Systems with Applications* 30.3. Intelligent Information Systems for Financial Engineering, pp. 543–552. ISSN: 0957-4174. DOI: `10.1016/j.eswa.2005.10.012`.

Dempster, Michael Alan Howarth and Yazann S Romahi (2002). «Intraday FX trading: An evolutionary reinforcement learning approach». In: *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, pp. 347–358.

Deng, Yue, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai (2017). «Deep Direct Reinforcement Learning for Financial Signal Representation and Trading». In: *IEEE Transactions on Neural Networks and Learning Systems* 28.3, pp. 653–664. DOI: `10.1109/TNNLS.2016.2522401`.

Dikmen, Murat and Catherine Burns (2022). «The effects of domain knowledge on trust in explainable AI and task performance: A case of peer-to-peer lending». In: *International Journal of Human-Computer Studies* 162, p. 102792.

Dixon, Matthew F., Igor Halperin, and Paul Bilokon (2020). «Applications of Reinforcement Learning». In: *Machine Learning in Finance: From Theory to Practice*. Cham: Springer International Publishing, pp. 347–418. ISBN: 978-3-030-41068-1. DOI: `10.1007/978-3-030-41068-1_10`.

Dong, G., G. Liao, H. Liu, and G. Kuang (2018). «A Review of the Autoencoder and Its Variants: A Comparative Perspective from Target Recognition in Synthetic-Aperture Radar Images». In: *IEEE Geoscience and Remote Sensing Magazine* 6.3, pp. 44–68.

Drucker, Harris, Christopher J. C. Burges, Linda Kaufman, Alexander J. Smola, and Vladimir Vapnik (1996). «Support Vector Regression Machines». In: *NIPS*, pp. 155–161.

El Qadi, Ayoub, Maria Trocan, Natalia Díaz-Rodríguez, and Thomas Frossard (2022). «Feature contribution alignment with expert knowledge for artificial intelligence credit scoring». In: *Signal, Image and Video Processing*. ISSN: 1863-1711. DOI: `10.1007/s11760-022-02239-7`.

Enke, David and Suraphan Thawornwong (2005). «The use of data mining and neural networks for forecasting stock market returns». In: *Expert Systems with Applications* 29.4, pp. 927–940. ISSN: 0957-4174.

Eraslan, Gökcen, Lukas M Simon, Maria Mircea, Nikola S Mueller, and Fabian J Theis (2019). «Single-cell RNA-seq denoising using a deep count autoencoder». In: *Nature communications* 10.1, pp. 1–14.

Fama, Eugene F (1970). «Efficient capital markets: A review of theory and empirical work». In: *The journal of Finance* 25.2, pp. 383–417.

Fama, Eugene F. and Kenneth R. French (1993). «Common risk factors in the returns on stocks and bonds». In: *Journal of Financial Economics* 33.1, pp. 3–56. ISSN: 0304-405X. DOI: `https://doi.org/10.1016/0304-405X(93)90023-5`. URL: `https://www.sciencedirect.com/science/article/pii/0304405X93900235`.

— (2015a). «A five-factor asset pricing model». In: *Journal of Financial Economics* 116.1, pp. 1–22. ISSN: 0304-405X. DOI: `https://doi.org/10.1016/j.jfineco.2014.10.010`. URL: `https://www.sciencedirect.com/science/article/pii/S0304405X14002323`.

— (Aug. 2015b). «Dissecting Anomalies with a Five-Factor Model». In: *The Review of Financial Studies* 29.1, pp. 69–103. ISSN: 0893-9454. DOI: `10.1093/rfs/hhv043`. eprint: `https:`

`//academic.oup.com/rfs/article-pdf/29/1/69/24450717/hhv043.pdf`. URL: `https://doi.org/10.1093/rfs/hhv043`.

Fanai, Hosein and Hossein Abbasimehr (2023). «A novel combined approach based on deep Autoencoder and deep classifiers for credit card fraud detection». In: *Expert Systems with Applications* 217, p. 119562. ISSN: 0957-4174.

Fang, Meng et al. (2019). «DHER: Hindsight Experience Replay for Dynamic Goals». In: *International Conference on Learning Representations*.

Fengqian, Ding and Luo Chao (2020). «An Adaptive Financial Trading System Using Deep Reinforcement Learning With Candlestick Decomposing Features». In: *IEEE Access* 8, pp. 63666–63678. DOI: `10.1109/ACCESS.2020.2982662`.

Feuerriegel, Stefan, Mateusz Dolata, and Gerhard Schwabe (2020). «Fair AI: Challenges and opportunities». In: *Business & information systems engineering* 62, pp. 379–384.

French, Craig W (2003). «The Treynor capital asset pricing model». In: *Journal of Investment Management* 1.2, pp. 60–72.

Fuentes-García, Marta, Gabriel Maciá-Fernández, and José Camacho (2018). «Evaluation of diagnosis methods in PCA-based Multivariate Statistical Process Control». In: *Chemometrics and Intelligent Laboratory Systems* 172, pp. 194–210. ISSN: 0169-7439.

Ge, Bingfeng, Keith W. Hipel, Liping Fang, Kewei Yang, and Yingwu Chen (2014). «An Interactive Portfolio Decision Analysis Approach for System-of-Systems Architecting Using the Graph Model for Conflict Resolution». In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 44.10, pp. 1328–1346. DOI: `10.1109/TSMC.2014.2309321`.

Ghorbani, Mahsa and Edwin KP Chong (2020). «Stock price prediction using principal components». In: *Plos one* 15.3, e0230124.

Gocken, Mustafa, Mehmet Ozcalici, Asli Boru, and Ayse Tugba Dosdogru (2016). «Integrating metaheuristics and Artificial Neural Networks for improved stock price prediction». In: *Expert Systems with Applications* 44, pp. 320–331. ISSN: 0957-4174.

Goodell, John W., Satish Kumar, Weng Marc Lim, and Debidutta Pattnaik (2021). «Artificial intelligence and machine learning in finance: Identifying foundations, themes, and research clusters from bibliometric analysis». In: *Journal of Behavioral and Experimental Finance* 32, p. 100577. ISSN: 2214-6350. DOI: `10.1016/j.jbef.2021.100577`.

Gradzki, Przemysław and Piotr Wójcik (2023). «Is attention all you need for intraday Forex trading?» In: *Expert Systems*, e13317. DOI: `https://doi.org/10.1111/exsy.13317`.

Graves, Alex, Abdel-Rahman Mohamed, and Geoffrey Hinton (2013). «Speech recognition with deep recurrent neural networks». In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*.

Greydanus, Samuel, Anurag Koul, Jonathan Dodge, and Alan Fern (2018). «Visualizing and understanding atari agents». In: *International conference on machine learning*. PMLR, pp. 1792–1801.

Grobys, Klaus, Shaker Ahmed, and Niranjan Sapkota (2020). «Technical trading rules in the cryptocurrency market». In: *Finance Research Letters* 32, p. 101396. DOI: `https://doi.org/10.1016/j.frl.2019.101396`.

Gu, Shihao, Bryan Kelly, and Dacheng Xiu (2020a). «Autoencoder asset pricing models». In: *Journal of Econometrics*. ISSN: 0304-4076.

— (2020b). «Empirical asset pricing via machine learning». In: *The Review of Financial Studies* 33.5, pp. 2223–2273.

Guo, Suna (Sihang) et al. (2021). «Machine versus Human Attention in Deep Reinforcement Learning Tasks». In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., pp. 25370–25385.

Gupta, V. and M. Mittal (2019). «A Comparison of ECG Signal Pre-processing Using FrFT, FrWT and IPCA for Improved Analysis». In: *IRBM* 40.3, pp. 145–156. ISSN: 1959-0318.

Guresen, Erkam, Gulgun Kayakutlu, and Tugrul U. Daim (2011). «Using artificial neural network models in stock market index prediction». In: *Expert Systems with Applications* 38.8, pp. 10389–10397. ISSN: 0957-4174.

Hancock, John T and Taghi M Khoshgoftaar (2020). «CatBoost for big data: an interdisciplinary review». In: *Journal of big data* 7.1, pp. 1–45. DOI: `https://doi.org/10.1186/s40537-020-00369-8`.

He, Guoliang et al. (2022). «Online Rule-Based Classifier Learning on Dynamic Unlabeled Multivariate Time Series Data». In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 52.2, pp. 1121–1134.

Heaton, James B, Nick G Polson, and Jan Hendrik Witte (2017). «Deep learning for finance: deep portfolios». In: *Applied Stochastic Models in Business and Industry* 33.1, pp. 3–12.

Heuillet, Alexandre, Fabien Couthouis, and Natalia Díaz-Rodríguez (2021). «Explainability in deep reinforcement learning». In: *Knowledge-Based Systems* 214, p. 106685. ISSN: 0950-7051. DOI: `10.1016/j.knosys.2020.106685`.

Hinton, G. E. and R. R. Salakhutdinov (2006). «Reducing the Dimensionality of Data with Neural Networks». In: *Science* 313, pp. 504–507.

Hirchoua, Badr, Brahim Ouhbi, and Bouchra Frikh (2021). «Deep reinforcement learning based trading agents: Risk curiosity driven learning for financial rules-based policy». In: *Expert Systems with Applications* 170, p. 114553. ISSN: 0957-4174.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). «Long Short-Term Memory». In: *Neural Computation* 9.8, pp. 1735–1780.

Hoepner, Andreas GF, David McMillan, Andrew Vivian, and Chardin Wese Simen (2021). «Significance, relevance and explainability in the machine learning age: an econometrics and financial data science perspective». In: *The European Journal of Finance* 27.1-2, pp. 1–7.

Hudson, Robert and Andrew Urquhart (2021). «Technical trading and cryptocurrencies». In: *Annals of Operations Research* 297.1, pp. 191–220. DOI: `https://doi.org/10.1007/s10479-019-03357-1`.

Ibrahim, Ahmed, Rasha Kashef, and Liam Corrigan (2021). «Predicting market movement direction for bitcoin: A comparison of time series modeling methods». In: *Computers & Electrical Engineering* 89, p. 106905. DOI: `https://doi.org/10.1016/j.compeleceng.2020.106905`.

Irie, Kazuki, Zoltán Tüske, Tamer Alkhouli, Ralf Schlüter, and Hermann Ney (Aug. 2016). «LSTM, GRU, Highway and a Bit of Attention: An Empirical Overview for Language Modeling in Speech Recognition». In: *Interspeech 2016*.

Ismail Fawaz, Hassan, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller (July 2019). «Deep learning for time series classification: a review». In: *Data Mining and Knowledge Discovery* 33.4, pp. 917–963.

Jabeur, Sami Ben, Cheima Gharib, Salma Mefteh-Wali, and Wissal Ben Arfi (2021a). «CatBoost model and artificial intelligence techniques for corporate failure prediction». In: *Technological Forecasting and Social Change* 166, p. 120658. DOI: `https://doi.org/10.1016/j.techfore.2021.120658`.

Jabeur, Sami Ben, Rabeh Khalfaoui, and Wissal Ben Arfi (2021b). «The effect of green energy, global environmental indexes, and stock markets in predicting oil price crashes: Evidence from explainable machine learning». In: *Journal of Environmental Management* 298, p. 113511. DOI: `https://doi.org/10.1016/j.jenvman.2021.113511`.

Jabeur, Sami Ben, Salma Mefteh-Wali, and Jean-Laurent Viviani (2021c). «Forecasting gold price with the XGBoost algorithm and SHAP interaction values». In: *Annals of Operations Research*, pp. 1–21. DOI: `https://doi.org/10.1007/s10479-021-04187-w`.

Jeong, Gyeeun and Ha Young Kim (2019). «Improving financial trading decisions using deep Q-learning: Predicting the number of shares, action strategies, and transfer learning». In: *Expert Systems with Applications* 117, pp. 125–138. ISSN: 0957-4174. DOI: `10.1016/j.eswa.2018.09.036`.

Jiang, Huihai (2021). «Cryptocurrency price forecasting based on shortterm trend KNN model». In: *2021 IEEE 3rd International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*. IEEE, pp. 1165–1169.

Jiang, Weiwei (2021). «Applications of deep learning in stock market prediction: Recent progress». In: *Expert Systems with Applications* 184, p. 115537. ISSN: 0957-4174.

Juozapaitis, Zoe, Anurag Koul, Alan Fern, Martin Erwig, and Finale Doshi-Velez (2019). «Explainable reinforcement learning via reward decomposition». In: *IJCAI/ECAI Workshop on explainable artificial intelligence*.

Kaji, Deepak A et al. (2019). «An attention based deep learning model of clinical events in the intensive care unit». In: *PloS one* 14.2, e0211057.

Kamalov, Firuz (2020). «Forecasting significant stock price changes using neural networks». In: *Neural Computing and Applications* 32.23, pp. 17655–17667.

Kamruzzaman, M. M., Omar Alruwaili, and Dhiyaa Aldaghmani (2022). «Measuring systemic and systematic risk in the financial markets using artificial intelligence». In: *Expert Systems* n/a.n/a, e12971. DOI: `https://doi.org/10.1111/exsy.12971`.

Kaplan Berkaya, Selcan et al. (2018). «A survey on ECG analysis». In: *Biomedical Signal Processing and Control* 43, pp. 216–235. ISSN: 1746-8094.

Kara, Yakup, Melek Acar Boyacioglu, and Ömer Kaan Baykan (2011). «Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange». In: *Expert Systems with Applications* 38.5, pp. 5311–5319. ISSN: 0957-4174.

Kim, A. et al. (2020). «Can deep learning predict risky retail investors? A case study in financial risk behavior forecasting». In: *European Journal of Operational Research* 283.1, pp. 217–234. DOI: `https://doi.org/10.1016/j.ejor.2019.11.007`.

Kim, Ha Young and Chang Hyun Won (2018). «Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models». In: *Expert Systems with Applications* 103, pp. 25–37.

Kim, Kyungwon and Jae Wook Song (2020). «Analyses on Volatility Clustering in Financial Time-Series Using Clustering Indices, Asymmetry, and Visibility Graph». In: *IEEE Access* 8, pp. 208779–208795.

Klein, Tony, Hien Pham Thu, and Thomas Walther (2018). «Bitcoin is not the New Gold – A comparison of volatility, correlation, and portfolio performance». In: *International Review of Financial Analysis* 59, pp. 105–116. DOI: `https://doi.org/10.1016/j.irfa.2018.07.010`.

Krajna, Agneza, Mihael Kovac, Mario Brcic, and Ana Šarčević (2022). «Explainable Artificial Intelligence: An Updated Perspective». In: *2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)*, pp. 859–864. DOI: `10.23919/MIPRO55190.2022.9803681`.

Krishn, Abhinav, Vikrant Bhateja, Akanksha Sahu, et al. (2014). «Medical image fusion using combination of PCA and wavelet analysis». In: *2014 international conference on advances in computing, communications and informatics (ICACCI)*. IEEE, pp. 986–991.

Kuiper, Ouren, Martin van den Berg, Joost van der Burgt, and Stefan Leijnen (2022). «Exploring Explainable AI in the Financial Sector: Perspectives of Banks and Supervisory Authorities». In: *Artificial Intelligence and Machine Learning*. Ed. by Luis A. Leiva, Cédric Pruski, Réka Markovich, Amro Najjar, and Christoph Schommer. Cham: Springer International Publishing, pp. 105–119. ISBN: 978-3-030-93842-0.

Kute, Dattatray Vishnu, Biswajeet Pradhan, Nagesh Shukla, and Abdullah Alamri (2021). «Deep Learning and Explainable Artificial Intelligence Techniques Applied for Detecting Money Laundering–A Critical Review». In: *IEEE Access* 9, pp. 82300–82317. DOI: `10.1109/ACCESS.2021.3086230`.

Lai, Zhao-Rong, Pei-Yi Yang, Liangda Fang, and Xiaotian Wu (2020). «Reweighted Price Relative Tracking System for Automatic Portfolio Optimization». In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 50.11, pp. 4349–4361. DOI: `10.1109/TSMC.2018.2852651`.

Langer, Markus et al. (2021). «What do we want from Explainable Artificial Intelligence (XAI)? – A stakeholder perspective on XAI and a conceptual model guiding interdisciplinary XAI research». In: *Artificial Intelligence* 296, p. 103473. ISSN: 0004-3702.

Lapan, Maxim (2018). *Deep Reinforcement Learning Hands-On: Apply modern RL methods, with deep Q-networks, value iteration, policy gradients, TRPO, AlphaGo Zero and more*. Packt Publishing Ltd.

Le Roux, Nicolas and Yoshua Bengio (2008). «Representational power of restricted Boltzmann machines and deep belief networks». In: *Neural computation* 20.6, pp. 1631–1649.

Lecun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). «Deep learning». In: *Nature* 521.7553, pp. 436–444.

Lesort, Timothée, Natalia Díaz-Rodríguez, Jean-Franois Goudou, and David Filliat (2018). «State representation learning for control: An overview». In: *Neural Networks* 108, pp. 379–392. ISSN: 0893-6080. DOI: `10.1016/j.neunet.2018.07.006`.

Lessmann, Stefan, Johannes Haupt, Kristof Coussement, and Koen W. De Bock (2021). «Targeting customers for profit: An ensemble learning framework to support marketing decision-making». In: *Information Sciences* 557, pp. 286–301. DOI: `https://doi.org/10.1016/j.ins.2019.05.027`.

Li, Jinzhong, Guanjun Liu, Chungang Yan, and Changjun Jiang (2017). «Robust Learning to Rank Based on Portfolio Theory and AMOSA Algorithm». In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47.6, pp. 1007–1018. DOI: `10.1109/TSMC.2016.2584786`.

Li, Ranran et al. (2022). «A Novel Symmetric Stacked Autoencoder for Adversarial Domain Adaptation Under Variable Speed». In: *IEEE Access* 10, pp. 24678–24689.

Li, Xin and Chong Alex Wang (2017). «The technology and economic determinants of cryptocurrency exchange rates: The case of Bitcoin». In: *Decision Support Systems* 95, pp. 49–60. ISSN: 0167-9236. DOI: `https://doi.org/10.1016/j.dss.2016.12.001`.

Li, Yang, Wanshan Zheng, and Zibin Zheng (2019). «Deep Robust Reinforcement Learning for Practical Algorithmic Trading». In: *IEEE Access* 7, pp. 108014–108022. DOI: `10.1109/ACCESS.2019.2932789`.

Liao, Shujian et al. (2023). «Sig-Wasserstein GANs for conditional time series generation». In: *Mathematical Finance* n/a.n/a. DOI: `https://doi.org/10.1111/mafi.12423`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1111/mafi.12423`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1111/mafi.12423`.

Lim, Qing Yang Eddy, Qi Cao, and Chai Quek (2022). «Dynamic portfolio rebalancing through reinforcement learning». In: *Neural Computing and Applications* 34.9, pp. 7125–7139. ISSN: 1433-3058. DOI: `10.1007/s00521-021-06853-3`.

Lin, Guancen, Aijing Lin, and Jianing Cao (2021). «Multidimensional KNN algorithm based on EEMD and complexity measures in financial time series forecasting». In: *Expert Systems with Applications* 168, p. 114443.

Lintner, John (1965a). «Security Prices, Risk, and Maximal Gains From Diversification». In: *The Journal of Finance* 20.4, pp. 587–615. ISSN: 00221082, 15406261. URL: `http://www.jstor.org/stable/2977249` (visited on 12/05/2023).

— (1965b). «The Valuation of Risk Assets and the Selection of Risky Investments in Stock Portfolios and Capital Budgets». In: *The Review of Economics and Statistics* 47.1, pp. 13–37. ISSN: 00346535, 15309142. URL: `http://www.jstor.org/stable/1924119` (visited on 12/05/2023).

Liu, Chien-Liang, Wen-Hoar Hsaio, and Yao-Chung Tu (2019). «Time Series Classification With Multivariate Convolutional Neural Network». In: *IEEE Transactions on Industrial Electronics* 66.6, pp. 4788–4797.

Lo, Andrew W. and A. Craig MacKinlay (1988). «Stock Market Prices Do Not Follow Random Walks: Evidence from a Simple Specification Test». In: *The Review of Financial Studies* 1.1, pp. 41–66.

López de Prado, Marcos M. (2020). *Machine Learning for Asset Managers*. Elements in Quantitative Finance. Cambridge University Press.

Lundberg, Scott and Su-In Lee (2017). «A Unified Approach to Interpreting Model Predictions». In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc.

Lundberg, Scott M et al. (2020). «From local explanations to global understanding with explainable AI for trees». In: *Nature machine intelligence* 2.1, pp. 56–67. DOI: `https://doi.org/10.1038/s42256-019-0138-9`.

Ma, Xiaomeng and Shuliang Lv (2019). «Financial credit risk prediction in internet finance driven by machine learning». In: *Neural Computing and Applications* 31.12, pp. 8359–8367. DOI: `https://doi.org/10.1007/s00521-018-3963-6`.

Ma, Yilin, Ruizhu Han, and Weizhong Wang (2021). «Portfolio optimization with return prediction using deep learning and machine learning». In: *Expert Systems with Applications* 165, p. 113973. DOI: `https://doi.org/10.1016/j.eswa.2020.113973`.

Madumal, Prashan, Tim Miller, Liz Sonenberg, and Frank Vetere (2020). «Explainable reinforcement learning through a causal lens». In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 03, pp. 2493–2500.

Mahdi, Mahmoud A., Khalid M. Hosny, and Ibrahim Elhenawy (2021). «Scalable Clustering Algorithms for Big Data: A Review». In: *IEEE Access* 9, pp. 80015–80027.

Mandeep et al. (2022). «Machine Learning Based Explainable Financial Forecasting». In: *2022 4th International Conference on Computer Communication and the Internet (ICCCI)*, pp. 34–38. DOI: `10.1109/ICCCI55554.2022.9850272`.

Marcílio, Wilson E. and Danilo M. Eler (2020). «From explanations to feature selection: assessing SHAP values as feature selection mechanism». In: *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pp. 340–347. DOI: `https://doi.org/10.1109/SIBGRAPI51738.2020.00053`.

Markowitz, Harry (1952). «Portfolio Selection». In: *The Journal of Finance* 7.1, pp. 77–91.

Marqués, A.I., V. García, and J.S. Sánchez (2012). «Exploring the behaviour of base classifiers in credit scoring ensembles». In: *Expert Systems with Applications* 39.11, pp. 10244–10250. ISSN: 0957-4174.

Masci, Jonathan, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber (2011). «Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction». In: *Artificial Neural Networks and Machine Learning – ICANN 2011*. Ed. by Timo Honkela, Włodzisław Duch, Mark Girolami, and Samuel Kaski. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 52–59. ISBN: 978-3-642-21735-7.

Menkhoff, Lukas (2010). «The use of technical analysis by fund managers International evidence». In: *Journal of Banking & Finance* 34.11, pp. 2573–2586. ISSN: 0378-4266.

Messaoud, Seifeddine, Abbas Bradai, and Emmanuel Moulay (2019). «Online GMM clustering and mini-batch gradient descent based optimization for industrial IoT 4.0». In: *IEEE Transactions on Industrial Informatics* 16.2, pp. 1427–1435.

Millea, Adrian (2021). «Deep Reinforcement Learning for Trading - A Critical Survey». In: *Data* 6.11. ISSN: 2306-5729. DOI: `10.3390/data6110119`.

Mnih, Volodymyr et al. (2015). «Human-level control through deep reinforcement learning». In: *Nature* 518.7540, pp. 529–533. ISSN: 1476-4687. DOI: `10.1038/nature14236`.

Mnih, Volodymyr et al. (20–22 Jun 2016). «Asynchronous Methods for Deep Reinforcement Learning». In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, pp. 1928–1937.

Moghaddam, Amin Hedayati, Moein Hedayati Moghaddam, and Morteza Esfandyari (2016). «Stock market index prediction using artificial neural network». In: *Journal of Economics, Finance and Administrative Science* 21.41, pp. 89–93. ISSN: 2077-1886.

Moody, J. and M. Saffell (2001). «Learning to trade via direct reinforcement». In: *IEEE Transactions on Neural Networks* 12.4, pp. 875–889. DOI: `10.1109/72.935097`.

Moody, John, Lizhong Wu, Yuansong Liao, and Matthew Saffell (1998). «Performance functions and reinforcement learning for trading systems and portfolios». In: *Journal of Forecasting* 17.5-6, pp. 441–470.

Mossin, Jan (1966). «Equilibrium in a Capital Asset Market». In: *Econometrica* 34.4, pp. 768–783. ISSN: 00129682, 14680262. URL: `http://www.jstor.org/stable/1910098` (visited on 12/05/2023).

Müllner, Daniel (2013). «fastcluster: Fast hierarchical, agglomerative clustering routines for R and Python». In: *Journal of Statistical Software* 53, pp. 1–18.

Nagel, Stefan (2021). *Machine learning in asset pricing*. Vol. 8. Princeton University Press.

Nannini, Luca, Agathe Balayn, and Adam Leon Smith (2023). «Explainability in AI Policies: A Critical Review of Communications, Reports, Regulations, and Standards in the EU, US, and UK». In: *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '23. Chicago, IL, USA: Association for Computing Machinery, pp. 1198–1212. ISBN: 9798400701924. DOI: `10.1145/3593013.3594074`. URL: `https://doi.org/10.1145/3593013.3594074`.

Narayan, Paresh Kumar, Seema Narayan, and Kannan Sivananthan Thuraisamy (2014). «Can institutions and macroeconomic factors predict stock returns in emerging markets?» In: *Emerging Markets Review* 19, pp. 77–95. ISSN: 1566-0141.

Nguyen, Duc Khuong, Georgios Sermpinis, and Charalampos Stasinakis (2023). «Big data, artificial intelligence and machine learning: A transformative symbiosis in favour of financial technology». In: *European Financial Management* 29.2, pp. 517–548. DOI: `https://doi.org/10.1111/eufm.12365`.

Nikolopoulos, Konstantinos, Sushil Punia, Andreas Schäfers, Christos Tsinopoulos, and Chrysovalantis Vasilakis (2021). «Forecasting and planning during a pandemic: COVID-19 growth rates, supply chain disruptions, and governmental decisions». In: *European journal of operational research* 290.1, pp. 99–115. DOI: `https://doi.org/10.1016/j.ejor.2020.08.001`.

Njah, Hasna, Salma Jamoussi, and Walid Mahdi (2021). «Breaking the curse of dimensionality: hierarchical Bayesian network model for multi-view clustering». In: *Annals of Mathematics and Artificial Intelligence* 89.10, pp. 1013–1033.

Ohana, Jean Jacques, Steve Ohana, Eric Benhamou, David Saltiel, and Beatrice Guez (2021). «Explainable AI (XAI) Models Applied to the Multi-Agent Environment of Financial Markets». In: *Explainable and Transparent AI and Multi-Agent Systems: Third International Workshop, EXTRAAMAS 2021, Virtual Event, May 3–7, 2021, Revised Selected Papers*. Berlin, Heidelberg: Springer-Verlag, pp. 189–207. ISBN: 978-3-030-82016-9. DOI: `10.1007/978-3-030-82017-6_12`.

Oyelade, Jelili et al. (2016). «Clustering Algorithms: Their Application to Gene Expression Data». In: *Bioinformatics and Biology Insights* 10, BBI.S38316.

Ozbayoglu, Ahmet Murat, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). «Deep learning for financial applications : A survey». In: *Applied Soft Computing* 93, p. 106384. ISSN: 1568-4946. DOI: `10.1016/j.asoc.2020.106384`.

Paiva, Felipe Dias, Rodrigo Tomás Nogueira Cardoso, Gustavo Peixoto Hanaoka, and Wendel Moreira Duarte (2019). «Decision-making for financial trading: A fusion approach of machine learning and portfolio selection». In: *Expert Systems with Applications* 115, pp. 635–655. ISSN: 0957-4174. DOI: `10.1016/j.eswa.2018.08.003`.

Park, Daehyung, Yuuna Hoshi, and Charles C Kemp (2018). «A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder». In: *IEEE Robotics and Automation Letters* 3.3, pp. 1544–1551.

Pasini, Giorgia (2017). «Principal component analysis for stock portfolio management». In: *International Journal of Pure and Applied Mathematics* 115.1, pp. 153–167.

Pedregosa, F. et al. (2011). «Scikit-learn: Machine Learning in Python». In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Pei, Songwen et al. (2020). «3DACN: 3D Augmented convolutional network for time series data». In: *Information Sciences* 513, pp. 17–29.

Pendharkar, Parag C. and Patrick Cusatis (2018). «Trading financial indices with reinforcement learning agents». In: *Expert Systems with Applications* 103, pp. 1–13. ISSN: 0957-4174. DOI: `10.1016/j.eswa.2018.02.032`.

Peng, Kai, Victor CM Leung, and Qingjia Huang (2018). «Clustering approach based on mini batch kmeans for intrusion detection system over big data». In: *IEEE Access* 6, pp. 11897–11906.

Perrin, Sarah and Thierry Roncalli (2020). «Machine learning optimization algorithms & portfolio allocation». In: *Machine Learning for Asset Management: New Developments and Financial Applications*, pp. 261–328.

Phongmekin, Athit and Pisit Jarumaneeroj (2018). «Classification Models for Stock's Performance Prediction: A Case Study of Finance Sector in the Stock Exchange of Thailand». In: *2018 International Conference on Engineering, Applied Sciences, and Technology (ICEAST)*. IEEE, pp. 1–4.

Pratt, John W. (1964). «Risk Aversion in the Small and in the Large». In: *Econometrica* 32.1/2, pp. 122–136. ISSN: 00129682, 14680262. (Visited on 03/21/2024).

Press, William H, Saul A Teukolsky, William T Vetterling, and Brian P Flannery (2007). *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press.

Prokhorenkova, Liudmila, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin (2018). «CatBoost: unbiased boosting with categorical features». In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc.

Pyo, Sujin, Jaewook Lee, Mincheol Cha, and Huisu Jang (2017). «Predictability of machine learning techniques to forecast the trends of market index prices: Hypothesis testing for the Korean stock markets». In: *PLoS ONE* 12.

Qaraei, Mohammadreza, Saeid Abbaasi, and Kamaledin Ghiasi-Shirazi (2021). «Randomized non-linear PCA networks». In: *Information Sciences* 545, pp. 241–253.

Al-Qatf, M., Y. Lasheng, M. Al-Habib, and K. Al-Sabahi (2018). «Deep Learning Approach Combining Sparse Autoencoder With SVM for Network Intrusion Detection». In: *IEEE Access* 6, pp. 52843–52856.

Qiu, Mingyue, Yu Song, and Fumio Akagi (2016). «Application of artificial neural network for the prediction of stock market returns: The case of the Japanese stock market». In: *Chaos, Solitons & Fractals* 85, pp. 1–7. ISSN: 0960-0779.

Rathan, Karunya, Somarouthu Venkat Sai, and Tubati Sai Manikanta (2019). «Crypto-Currency price prediction using Decision Tree and Regression techniques». In: *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 190–194. DOI: `https://doi.org/10.1109/ICOEI.2019.8862585`.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016). «"Why Should I Trust You?": Explaining the Predictions of Any Classifier». In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, pp. 1135–1144. DOI: `https://doi.org/10.1145/2939672.2939778`.

Roelofs, Rebecca et al. (2019). «A meta-analysis of overfitting in machine learning». In: *Advances in Neural Information Processing Systems* 32.

Ross, Stephen A (1976). «The arbitrage theory of capital asset pricing». In: *Journal of Economic Theory* 13.3, pp. 341–360. ISSN: 0022-0531.

Rubbaniy, Ghulame, Shoaib Ali, Costas Siriopoulos, and Aristeidis Samitas (2021). «Global Financial Crisis, COVID-19, Lockdown, and Herd Behavior in the US ESG Leader Stocks». In: *COVID-19, Lockdown, and Herd Behavior in the US ESG Leader Stocks (June 16, 2021)*. DOI: `https://doi.org/10.2139/ssrn.3868114`.

Russell, Stuart J and Peter Norvig (2010). *Artificial intelligence a modern approach*. London.

Sachan, Swati, Jian-Bo Yang, Dong-Ling Xu, David Eraso Benavides, and Yang Li (2020). «An explainable AI decision-support-system to automate loan underwriting». In: *Expert Systems with Applications* 144, p. 113100.

Sagir, Abdu and Saratha Sathasivan (Sept. 2017). «The use of artificial neural network and multiple linear regressions for stock market forecasting». In: *MATEMATIKA* 33, pp. 1–10.

Sak, Hasim, Andrew W. Senior, and Françoise Beaufays (2014). «Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition». In: *ArXiv* abs/1402.1128.

Sattarov, Otabek et al. (2020). «Recommending Cryptocurrency Trading Points with Deep Reinforcement Learning Approach». In: *Applied Sciences* 10.4. ISSN: 2076-3417. DOI: `10.3390/app10041506`.

Sequeira, Pedro and Melinda Gervasio (2020). «Interestingness elements for explainable reinforcement learning: Understanding agents' capabilities and limitations». In: *Artificial Intelligence* 288, p. 103367. ISSN: 0004-3702. DOI: `10.1016/j.artint.2020.103367`.

Sezer, Omer Berat, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu (2020). «Financial time series forecasting with deep learning : A systematic literature review: 2005–2019». In: *Applied Soft Computing* 90, p. 106181. ISSN: 1568-4946.

Shao, Bilin, Maolin Li, Yu Zhao, and Genqing Bian (2019). «Nickel Price Forecast Based on the LSTM Neural Network Optimized by the Improved PSO Algorithm». In: *Mathematical Problems in Engineering* 2019, pp. 1–15.

Shapley, Lloyd S, HW Kuhn, and AW Tucker (1953). «Contributions to the Theory of Games». In: *Annals of Mathematics studies* 28.2, pp. 307–317.

Sharpe, William F. (1964). «Capital Asset Prices: A Theory of Market Equilibrium Under Conditions of Risk». In: *The Journal of Finance* 19.3, pp. 425–442.

Shavandi, Ali and Majid Khedmati (2022). «A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets». In: *Expert Systems with Applications* 208, p. 118124. ISSN: 0957-4174. DOI: `10.1016/j.eswa.2022.118124`.

Shen, Yuming et al. (2021). «Financial Feature Embedding with Knowledge Representation Learning for Financial Statement Fraud Detection». In: *Procedia Computer Science* 187. 2020 International Conference on Identification, Information and Knowledge in the Internet of Things, IIKI2020, pp. 420–425. DOI: `https://doi.org/10.1016/j.procs.2021.04.110`.

Shi, Yong, Bo Li, Guangle Du, and Wei Dai (2021). «Clustering framework based on multi-scale analysis of intraday financial time series». In: *Physica A: Statistical Mechanics and its Applications* 567, p. 125728.

Staden, Heletjé E. van, Laurens Deprez, and Robert N. Boute (2022). «A dynamic "predict, then optimize" preventive maintenance approach using operational intervention data». In: *European Journal of Operational Research*. DOI: `https://doi.org/10.1016/j.ejor.2022.01.037`.

Stevenson, Matthew, Christophe Mues, and Cristián Bravo (2021). «The value of text for small business default prediction: A Deep Learning approach». In: *European Journal of Operational Research* 295.2, pp. 758–771. DOI: `https://doi.org/10.1016/j.ejor.2021.03.008`.

Sun, Mei, Yan Zhang, Rongpu Chen, Yulian Wen, and Peiyao Nie (2021). «Hedging strategy for commodity futures based on SVM-KNN». In: *International Journal of Reasoning-based Intelligent Systems* 13.3, pp. 139–146.

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le (2014). «Sequence to Sequence Learning with Neural Networks». In: *CoRR* abs/1409.3215. arXiv: `1409.3215`.

Sutton, Richard S and Andrew G Barto (2018). *Reinforcement learning: An introduction*. MIT press.

Sutton, Richard S, David McAllester, Satinder Singh, and Yishay Mansour (1999). «Policy Gradient Methods for Reinforcement Learning with Function Approximation». In: *Advances in Neural Information Processing Systems*. Ed. by S. Solla, T. Leen, and K. Müller. Vol. 12. MIT Press.

Syu, Jia-Hao, Mu-En Wu, and Jan-Ming Ho (2020). «Portfolio Management System with Reinforcement Learning». In: *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 4146–4151. DOI: `10.1109/SMC42975.2020.9283359`.

Taghian, Mehran, Ahmad Asadi, and Reza Safabakhsh (2022). «Learning financial asset-specific trading rules via deep reinforcement learning». In: *Expert Systems with Applications* 195, p. 116523. ISSN: 0957-4174. DOI: `10.1016/j.eswa.2022.116523`.

Tang, Rui and Simon Fong (2018). «Clustering big IoT data by metaheuristic optimized mini-batch and parallel partition-based DGC in Hadoop». In: *Future Generation Computer Systems* 86, pp. 1395–1412.

Tavenard, Romain et al. (2020). «Tslearn, A Machine Learning Toolkit for Time Series Data». In: *Journal of Machine Learning Research* 21.118, pp. 1–6.

Thakkar, Ankit and Kinjal Chaudhari (2021). «A comprehensive survey on portfolio optimization, stock price and trend prediction using particle swarm optimization». In: *Archives of Computational Methods in Engineering* 28.4, pp. 2133–2164.

Théate, Thibaut and Damien Ernst (2021). «An application of deep reinforcement learning to algorithmic trading». In: *Expert Systems with Applications* 173, p. 114632. ISSN: 0957-4174.

Tkáč, Michal and Robert Verner (2016). «Artificial neural networks in business: Two decades of research». In: *Applied Soft Computing* 38, pp. 788–804. ISSN: 1568-4946.

Tokuda, Eric K., Cesar H. Comin, and Luciano da F. Costa (2022). «Revisiting agglomerative clustering». In: *Physica A: Statistical Mechanics and its Applications* 585, p. 126433. ISSN: 0378-4371.

Vo, Au and Christopher Yost-Bremm (2020). «A High-Frequency Algorithmic Trading Strategy for Cryptocurrency». In: *Journal of Computer Information Systems* 60.6, pp. 555–568. DOI: `https://doi.org/10.1080/08874417.2018.1552090`.

Vyas, Aparna, Soohwan Yu, and Joonki Paik (2018). «Fourier Analysis and Fourier Transform». In: *Multiscale Transforms with Application to Image Processing*. Springer, pp. 15–43.

Wafi, Ahmed. S., Hassan Hassan, and Adel Mabrouk (2015). «Fundamental Analysis Models in Financial Markets - Review Study». In: *Procedia Economics and Finance* 30. IISES 3rd and 4th Economics and Finance Conference, pp. 939–947. ISSN: 2212-5671.

Wang, W., A. Ramesh, J. Zhu, J. Li, and D. Zhao (2020). «Clustering of Driving Encounter Scenarios Using Connected Vehicle Trajectories». In: *IEEE Transactions on Intelligent Vehicles* 5.3, pp. 485–496.

Wang, Xing et al. (2016). «Neural Machine Translation Advised by Statistical Machine Translation». In.

Watkins, Christopher J. C. H. and Peter Dayan (1992). «Q-learning». In: *Machine Learning* 8.3, pp. 279–292. ISSN: 1573-0565. DOI: `10.1007/BF00992698`.

Wen, Long, Liang Gao, and Xinyu Li (2019). «A New Deep Transfer Learning Based on Sparse Auto-Encoder for Fault Diagnosis». In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 49.1, pp. 136–144.

Wu, Xing et al. (2020). «Adaptive stock trading strategies with deep reinforcement learning methods». In: *Information Sciences* 538, pp. 142–158. ISSN: 0020-0255. DOI: `10.1016/j.ins.2020.05.066`.

Wu, Xiuge, Tinghuai Ma, Jie Cao, Yuan Tian, and Alia Alabdulkarim (2018). «A comparative study of clustering ensemble algorithms». In: *Computers & Electrical Engineering* 68, pp. 603–615. ISSN: 0045-7906.

Xia, Yufei, Lingyun He, Yinguo Li, Nana Liu, and Yanlin Ding (2020). «Predicting loan default in peer-to-peer lending using narrative data». In: *Journal of Forecasting* 39.2, pp. 260–280. DOI: `https://doi.org/10.1002/for.2625`.

Xu, Yongan, Jianqiong Wang, Zhonglu Chen, and Chao Liang (2023). «Sentiment indices and stock returns: Evidence from China». In: *International Journal of Finance & Economics* 28.1, pp. 1063–1080.

Yang, Hongyang, Xiao-Yang Liu, Shan Zhong, and Anwar Walid (2020). «Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy». In: *Proceedings of the First ACM International Conference on AI in Finance*. ICAIF '20. New York, New York: Association for Computing Machinery. ISBN: 9781450375849. DOI: `10.1145/3383455.3422540`.

Yao, Xuanxia, Jiafei Wang, Mengyu Shen, Huafeng Kong, and Huansheng Ning (2019). «An improved clustering algorithm and its application in IoT data analysis». In: *Computer Networks* 159, pp. 63–72. ISSN: 1389-1286.

Yin, Chunyong, Sun Zhang, Jin Wang, and Neal N. Xiong (2022). «Anomaly Detection Based on Convolutional Recurrent Autoencoder for IoT Time Series». In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 52.1, pp. 112–122.

Yu, Huanhuan, Rongda Chen, and Guoping Zhang (2014). «A SVM Stock Selection Model within PCA». In: *Procedia Computer Science* 31. 2nd International Conference on Information Technology and Quantitative Management, ITQM 2014, pp. 406–412. ISSN: 1877-0509.

Zarkias, Konstantinos Saitas, Nikolaos Passalis, Avraam Tsantekidis, and Anastasios Tefas (2019). «Deep Reinforcement Learning for Financial Trading Using Price Trailing». In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3067–3071. DOI: `10.1109/ICASSP.2019.8683161`.

Zhang, Chanyuan, Soohyun Cho, and Miklos Vasarhelyi (2022). «Explainable Artificial Intelligence (XAI) in auditing». In: *International Journal of Accounting Information Systems* 46. 2021 Research Symposium on Information Integrity & Information Systems Assurance, p. 100572. ISSN: 1467-0895. DOI: `10.1016/j.accinf.2022.100572`.

Zhang, Daohui, Xingang Zhao, Jianda Han, and Yiwen Zhao (2014). «A comparative study on PCA and LDA based EMG pattern recognition for anthropomorphic robotic hand». In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4850–4855.

Zhang, Liangwei, Jing Lin, and Ramin Karim (2017). «Sliding Window-Based Fault Detection From High-Dimensional Data Streams». In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47.2, pp. 289–303.

Zhang, Wei, Pengfei Wang, Xiao Li, and Dehua Shen (2018). «The inefficiency of cryptocurrency and its cross-correlation with Dow Jones Industrial Average». In: *Physica A: Statistical Mechanics and its Applications* 510, pp. 658–670. DOI: `https://doi.org/10.1016/j.physa.2018.07.032`.

Zhang, Xinsheng, Yulong Ma, and Minghu Wang (2023). «An attention-based Logistic-CNN-BiLSTM hybrid neural network for credit risk prediction of listed real estate enterprises». In: *Expert Systems*, e13299. DOI: `https://doi.org/10.1111/exsy.13299`.

Zhang, Zihao, Stefan Zohren, and Stephen Roberts (2020). «Deep reinforcement learning for trading». In: *The Journal of Financial Data Science* 2.2, pp. 25–40.

Zhao, Zongyuan et al. (2015). «Investigation and improvement of multi-layer perceptron neural networks for credit scoring». In: *Expert Systems with Applications* 42.7, pp. 3508–3516. ISSN: 0957-4174.

Zhong, Xiao and David Enke (2017). «Forecasting daily stock market return using dimensionality reduction». In: *Expert Systems with Applications* 67, pp. 126–139. ISSN: 0957-4174.

Zhu, Yani, Chaoyang Zhu, and Xiaoxin Li (2018). «Improved principal component analysis and linear regression classification for face recognition». In: *Signal Processing* 145, pp. 175–182. ISSN: 0165-1684.

Zia, Tehseen and Usman Zahid (Aug. 2018). «Long short-term memory recurrent neural network architectures for Urdu acoustic modeling». In: *International Journal of Speech Technology* 22.1, pp. 21–30.