



**UNIVERSITY OF
KWAZULU-NATAL**

**INYUVESI
YAKWAZULU-NATALI**

Machine learning vs traditional portfolio optimisation under different market conditions in Exchange Traded Funds portfolios

by

Linah Mutemeri

216076569

A dissertation submitted in fulfilment of the requirements for the degree

of

DOCTOR OF PHILOSOPHY (FINANCE)

School of Accounting, Economics and Finance

College of Law and Management Studies

Supervisor: Prof. Kerry-Ann McCullough

Co-Supervisor: Dr Helper Zhou

August 2025

DECLARATION

I, **Linah Mutemeri**, declare that:

- (i) The research reported in this dissertation/thesis, except where otherwise indicated, is my original research.
- (ii) This dissertation/thesis has not been submitted for any degree or examination at any other university.
- (iii) This dissertation/thesis does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.
- (iv) This dissertation/thesis does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:
 - (a) their words have been re-written, but the general information attributed to them has been referenced.
 - (b) where their exact words have been used, their writing has been placed inside quotation marks and referenced.
- (v) where I have reproduced a publication of which I am author, co-author or editor, I have indicated in detail which part of the publication was actually written by myself alone and have fully referenced such publications.
- (vi) this dissertation/thesis does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the dissertation/thesis and in the References sections.

Signed:



DEDICATION

To my wonderful Son, Willard, my mother, Joyce and my late father, Cleopas.

ACKNOWLEDGEMENTS

I want to take this opportunity to express my deepest gratitude and appreciation for all the people who helped me throughout the research process. First, I would like to thank my supervisors, Professor Kerry-Ann McCullough and Dr Helper Zhou, who provided me with continuous help, guidance, encouragement, and cooperation. I am genuinely thankful for your patience and understanding during challenging moments.

In addition, I would like to communicate my thankfulness for my family members: my brother Partson Tinarwo and his wife Felistas Nyakusvora who offered unconditional love and support and always believed in me. You are the reason that I am the person I am today, and I will do my best to continue deserving your blessings. I would like to thank my son Willard Junior Chirunga and my cousins Ashley and Allan who made me laugh all the time thereby easing my stressful times.

To all of the staff and colleagues in the School of Accounting, Economics and Finance, thank you for everything. This opportunity has been a life-changing experience, a source of incredible life-long memories, and a strong influence on my future. To my friends and workmates, it was a pleasure being able to meet you, and I send special thanks for your support and warm feelings.

Above all, “And whatever you do, whether in word or deed, do it all in the name of the Lord Jesus, giving thanks to God the Father through him” (Colossians 3:17, Holy Bible, New International Version).

ABSTRACT

The rapid advancement of machine learning (ML), a branch of artificial intelligence that enables systems to learn from data, has significantly reshaped financial decision-making. This study examines the comparative performance of traditional and ML-based portfolio optimization methods in Exchange-Traded Funds (ETFs), which provide diversified exposure to equities, bonds, commodities, and alternative assets. ETFs' liquidity, transparency, and cost-effectiveness make them an ideal context for testing optimization strategies.

The research employs a three-stage methodology. First, a Systematic Literature Review (SLR) assesses the global application of ML techniques in portfolio optimization, highlighting models such as Random Forest (RF), Long Short-Term Memory (LSTM), and Support Vector Machines (SVM). Second, an empirical analysis uses daily return data from 33 JSE-listed ETFs spanning 2013–2023 to compare three traditional models Markowitz Model (MM), Conditional Value-at-Risk (CVaR), and Mean Absolute Deviation (MAD) with three ML models: Enhanced Radial Basis Function Artificial Neural Network (K4-RANN), Genetic Algorithms (GA), and Extreme Gradient Boosting (XGBoost). Third, event study methodology evaluates portfolio resilience during South African market shocks, namely the COVID-19 pandemic (2020–2022), the July 2021 social unrest, and the April 2022 KwaZulu-Natal floods. Performance is measured through mean return, standard deviation, and Sharpe ratio.

Findings show that ML-based models generally outperform traditional approaches by generating higher risk-adjusted returns and improving diversification. Under normal market conditions, GA and K4-RANN produced portfolios with lower risk and stronger returns. During crises, XGBoost proved most effective, striking a balance between returns and volatility and demonstrating adaptability in turbulent markets. These results confirm the potential of ML models to enhance portfolio stability and resilience in both normal and distressed environments.

The study concludes that portfolio managers and institutional investors should integrate ML-based optimization, particularly ensemble models like XGBoost, into dynamic allocation strategies. For policymakers and financial institutions in emerging markets, investment in ML infrastructure and data capabilities is recommended to drive

innovation in asset management. Future research should investigate hybrid frameworks that merge traditional financial theory with ML to advance more comprehensive portfolio optimization approaches.

Keywords: Portfolio optimization, artificial intelligence, machine learning, Exchange Traded Funds, distressed market conditions, diversification.

TABLE OF CONTENTS

DECLARATION	ii
DEDICATION	iii
ACKNOWLEDGEMENTS	iv
ABSTRACT	v
TABLE OF CONTENTS	vii
LIST OF ACRONYMS	xi
LIST OF TABLES	xiii
LIST OF FIGURES	xiv
CHAPTER 1: INTRODUCTION AND BACKGROUND	1
1.1 Introduction	1
1.2 Background of the Study	4
1.3 Motivation of the Study	9
1.4 Research Problem	13
1.5 Research Objectives	15
1.6 Research Questions	15
1.7 Statement of Hypotheses	16
1.8 Significance of the Study and Contributions to Knowledge	16
1.9 Scope of the Study	17
1.10 Structure of Thesis	19
CHAPTER 2: THEORETICAL FOUNDATIONS	22
2.1 Introduction	22
2.2 Theories Underpinning Portfolio Optimization	22
2.2.1 Modern Portfolio Theory (MPT)	22
2.2.2 Portfolio Optimization for Fund of Funds	27
2.2.3 Efficient Market Hypothesis (EMH)	29
2.2.4 Adaptive Market Hypothesis (AMH)	32
2.2.5 MPT, EMH and AMH in Portfolio Optimization	34
2.3 Theoretical Foundations of ETFs	37
2.3.1 ETF Creation and Redemption Process	37

2.3.2 Benefits of ETFs.....	40
2.3.3 The ETF Portfolio (Fund of Funds).....	42
2.3.4 The Difference Between a Fund, and Fund of Funds.....	43
2.4 Summary	44
CHAPTER 3: MACHINE LEARNING MODELS FOR PORTFOLIO OPTIMIZATION	46
3.1 Introduction	46
3.1.1 The Role of Machine Learning in Finance	47
3.1.2 Machine Learning in Portfolio Management	48
3.1.3 Challenges and Solutions in Machine Learning for Portfolio Optimisation	49
3.2 A Critical Analysis of ML Models for Portfolio Optimization	52
3.2.1 Supervised Learning	52
3.2.2 Unsupervised Learning	72
3.3 Summary	76
CHAPTER 4: COMPARATIVE STUDY OF MACHINE LEARNING TECHNIQUES IN PORTFOLIO OPTIMISATION ACROSS DIFFERENT ASSET CLASSES	78
4.1 Introduction	78
4.2 Review of Empirical Literature	79
4.3 Data and Methodology.....	84
4.3.1 Planning	84
4.3.2 Conducting	86
4.4 Discussion of Findings and Analysis.....	94
4.4.1 Geographical Map.....	95
4.4.2 Review of ML Techniques and their implementation	96
4.4.3 Review of ML Techniques and their Effectiveness	98
4.4.4 Performance Metrics	100
4.4.5 Strengths and Weaknesses in Existing Research	101
4.4.6 Implications for Investors, Financial Institutions and Researchers	102
4.5 Summary	102
CHAPTER 5: THE PERFORMANCE OF TRADITIONAL VERSUS MACHINE LEARNING PORTFOLIO OPTIMISATION IN ETF PORTFOLIOS.....	104
5.1 Introduction	104
5.2 Traditional and Machine Learning Portfolio Optimization Techniques	106

5.2.1 The Markowitz Model (Traditional)	107
5.2.2 Conditional Value at Risk (CvaR) (Traditional)	108
5.2.3 Mean Absolute Deviation (MAD) (Traditional)	110
5.2.4 Enhanced Radial Basis Artificial Neural Network (K4-RANN) (ML).....	111
5.2.5 Evolutionary (Genetic) Algorithms (ML).....	112
5.2.6 Extreme Gradient Boosting (XG Boost) (ML)	113
5.3 Review of Empirical Literature	115
5.4 Gaps in Existing Literature.....	118
5.5 Data and Methodology.....	119
5.5.1 Data and Sampling.....	120
5.5.2 Preliminary Tests.....	121
5.5.3 Model Specification and Software	121
5.5.3 Performance Metrics	130
5.6 Results and Discussion	132
5.6.1 Preliminary Tests Results.....	133
5.6.2 Traditional Portfolio Optimization Results.....	136
5.6.3 Machine Learning Portfolio Optimization Results.....	143
5.6.4 Comparative Analysis.....	148
5.7 Summary	152
CHAPTER 6: ASSESSMENT ON THE PERFORMANCE OF ETF PORTFOLIOS CONSTRUCTED BY MACHINE LEARNING TECHNIQUES PERFORM UNDER DISTRESSED MARKET CONDITIONS	154
6.1 Introduction	154
6.2. Literature Review on Market Distress and Financial Crises.....	156
6.2.1 Market Distress and Financial Crises.....	156
6.2.2 Distressed Market Conditions and Portfolio Performance	162
6.2.3 ETF Performance Under Distressed Market Conditions	162
6.2.4 Market Distress and its Impact on the Portfolios' Performance	163
6.2.5 Literature Gap Analysis	166
6.3 Data and Methodology.....	167
6.3.1 Data Sources and Sample Period	167
6.3.2 Methodology.....	168

6.3.3 Empirical Results and Analysis	170
6.3.4 Performance Comparison During Normal and Distressed Periods	174
6.4 Summary and Conclusion.....	180
CHAPTER 7: SUMMARY, CONCLUSIONS AND RECOMMENDATIONS	182
7.1 Introduction	182
7.2 Summary of Key Findings.....	183
7.2.1 Machine Learning Portfolio Optimization Across Different Asset Classes	183
7.2.2 Traditional vs Machine Learning Portfolio Optimization Techniques in ETF Portfolios	184
7.2.3 ETF Portfolios Performance Under Distressed Market Conditions.....	185
7.3 Conclusions and Recommendations	187
7.4 Original Contributions of the Thesis.....	188
7.4.1 Original Contribution from Objective 1	188
7.4.2 Original Contribution from Objective 2	188
7.4.3 Original Contribution from Objective 3	189
7.4.4 Original Contribution from Objective 4	190
7.5 Implications of the Study.....	190
7.6 Limitations of the Study and Recommendations for Further Research.....	193
REFERENCES.....	195
APPENDIX 1: ETFs Information.....	245
APPENDIX 2: R CODES.....	246
APPENDIX 3A: ETHICAL CLEARANCE LETTER	295
APPENDIX 3B: AMENDED ETHICAL CLEARANCE LETTER.....	296

LIST OF ACRONYMS

ML	Machine Learning
ETF	Exchange Traded Funds
NAV	Net Asset Value
MPT	Modern Portfolio Theory
MVO	Mean-Variance Optimisation
CvaR	Conditional Value-at-Risk
MAD	Mean Absolute Deviation
MM	Markowitz Model
K4-RANN	Enhanced Radial Basis Artificial Neural Network
GA	Evolutionary Genetic Algorithms
XG Boost	Extreme Gradient Boosting
SLR	Systematic Literature Review
GARCH	Generalized Autoregressive Conditional Heteroskedasticity
MS-GARCH	Markov-Switching Generalized Autoregressive Conditional Heteroskedasticity
EMH	Efficient Market Hypothesis
AMH	Adaptive Market Hypothesis
CEF	Closed-End Funds
AI	Artificial Intelligence
ANN	Artificial Neural Network
PDS	Product Disclose Statements
LASSO	Least Absolute Shrinkage and Selection Operator
SVC	Support Vector Classification
SVR	Support Vector Regression

SVM	Support Vector Machines
RBF	Radial Basis Function
LightGBM	Light Gradient Boost Model
GBDT	Gradient Boosting Decision Tree
EFB	Exclusive Feature Bundling
TS	Target Statistics
GMM	Gaussian Mixture Model
RF	Random Forest
LSTM	Long Short-Term Memory
DNN	Deep Neural Network
SD	Standard Deviation
CNN	Convolutional Neural Networks
U. S	United States
RMSE	Root Mean Square Error
MDD	Maximum Drawdown
GANs	Generative Adversarial Networks
ASEAN	Association of Southeast Asian Nations

LIST OF TABLES

Table 4.1	Applied Filtering Options During the Search Process	87
Table 4.2	Summary of SLR Reviewed Articles	88-90
Table 5.1	Descriptive Statistics Results.....	124-125
Table 5.2	Stationarity and Cointegration Tests Results.....	133
Table 5.3	Markowitz Model Portfolio Optimization Results.....	135
Table 5.4	Conditional Value at Risk Portfolio Optimization Results.....	137
Table 5.5	Mean Absolute Deviation Portfolio Optimization Results.....	139
Table 5.6	K4-RANN Portfolio Optimization Results.....	141-142
Table 5.7	Evolutionary Genetic Algorithm Portfolio Optimization Results.....	143
Table 5.8	XGBoost Portfolio Optimization Results.....	145

LIST OF FIGURES

Figure 1.1	Relationship Between AI, ML, and DL.....	5
Figure 2.1	Overview of the Interrelation and Evolution of MPT, EMH, and AMH...34	
Figure 2.2	ETF Trading Process in Primary and Secondary Markets.....	37
Figure 3.1	The Hyperplane in Action in SVM.....	59
Figure 3.2	The Visualisation of Random Forest Trees.....	62
Figure 3.3	The Hidden Layers Transformation from an Input into an Output.....	69
Figure 3.4	Two Main Types of Hierarchical Clustering- Agglomerative and Divisive Clustering.....	74
Figure 4.1	Detailed SLR Framework	79
Figure 4.2	Inclusion of Machine Learning systems in Different Sectors.....	90
Figure 4.3	Overlay Visualization of Themes from SLR	92
Figure 4.4	SLR Articles Grouped by Location	94
Figure 4.5	Geographic Map Based on Bibliographic Data.....	95
Figure 4.6	Distribution of Machine Learning Models.....	97
Figure 5.1	The Efficient Frontier.....	105
Figure 5.2	Feedforward RANN Network.....	124
Figure 5.3	Sharpe Ratio for Portfolio Performance Assessment.....	130
Figure 5.4	Comparison Analysis of Models.....	147
Figure 6.1	Event Study Findings.....	169
Figure 6.2	Performance Comparison Under Normal Conditions.....	173
Figure 6.3	Performance Comparison Under Distressed Conditions.....	175

CHAPTER 1: INTRODUCTION AND BACKGROUND

1.1 Introduction

A collection of financial instruments, including equities, bonds, and other investment vehicles, held by an individual or organisation is known as a portfolio (Simonis, 2022). Optimisation of a portfolio refers to the process of carefully selecting and allocating assets in an investment portfolio with the goal of achieving the best possible combination of returns and risk based on an investor's specific objectives and constraints (Simonis, 2022). By investing in a portfolio of assets, the risk can be mitigated through the process of diversification (Zaimovic, Omani and Berilo, 2021). Diversification involves allocating investments across a wide range of assets with varying levels of risk and return, with the aim of reducing the overall risk associated with the portfolio in comparison to the risk inherent in each individual asset, within the given period (Kuang, 2022). However, the mere act of investing in a collection of assets is insufficient. It is of utmost importance to create a portfolio that is optimized in order to achieve the most favourable equilibrium between risk and return (Gambeta and Kwon, 2020).

A portfolio that lacks optimisation has the potential to be excessively concentrated in a specific asset or sector, thereby increasing risk and resulting in lower returns (De Spiegeleer *et al.*, 2023). Through the examination of projected returns, variances, and correlations of the assets within a collection of investments (portfolio), investors can formulate portfolios that generate the most favourable projected returns for a predetermined level of risk or the lowest conceivable risk for a predetermined level of projected returns (De Spiegeleer *et al.*, 2023). Overall, portfolio optimisation is a potent tool that empowers investors to construct diversified portfolios that are precisely adjusted to fulfil their investment aspirations while minimizing risk (Simonis, 2022).

The South African capital market presents a wide range of investment opportunities to investors, allowing them to establish investment positions in different industries and ensure a profitable return on investment (Andreoni, Robb, and Van Huellen, 2021). It facilitates the mobilization of capital and enables investors to establish positions across various asset classes to achieve their financial goals, including portfolio diversification and long-term wealth creation (Sugozu, Yasar and Verberi, 2023). The

key components of the South African capital market include equities, bonds, commodities and Exchange Traded funds (ETFs) (Shenjere, Ferreira-Schenk and Moodley, 2025).

The Johannesburg Stock Exchange (JSE) is the primary equity market in South Africa and one of the largest stock exchanges globally by market capitalization (Strydom, 2021). It lists companies from various sectors, including mining, financial services, retail, and technology. The JSE provides investors with access to a diverse range of shares, enabling equity exposure to both blue-chip companies and smaller, high-growth firms (Moyo, 2024). The South African bond market is robust, featuring government bonds and corporate bonds issued by leading companies. These instruments cater to investors seeking regular income and lower risk compared to equities. The bond market plays a pivotal role in funding infrastructure projects and corporate expansion in the country. South Africa is a significant player in the global commodities market, particularly in precious metals such as gold, platinum, and palladium (Zhang *et al.*, 2024). Commodity investments provide a hedge against inflation and currency fluctuations, offering diversification benefits within a portfolio. ETFs are a popular investment vehicle in the South African market, providing a cost-effective and transparent way to gain exposure to various asset classes. Most major asset categories, including equities, bonds, and commodities, are represented through ETFs (Kunjal, Peerbhai and Muzindutsi, 2021).

One particular financial asset that gives exposure is an ETF, and it also provides diversification benefits by investing in various securities that replicate the benchmark index (Dickason and Ferreira, 2018). For instance, Equity ETFs such as the Satrix Top 40 ETF offer exposure to the top 40 companies listed on the JSE. Bond ETFs like the NewFunds GOVI ETF track government bond indices. Commodity ETFs such as the Absa NewGold ETF tracks the price of gold, allowing investors to participate in the commodity market without physically holding the metal. However, the sheer similarity of ETFs within specific categories, such as equity indices or sector-specific funds, presents a unique challenge in portfolio optimization. When faced with a universe of highly comparable assets, identifying subtle variations that can enhance portfolio performance becomes a complex task (Marín-Rodríguez, González-Ruiz and Botero,

2025). As a consequence, it is rare to see portfolios made up of several already diversified funds, such as ETFs.

The optimization of investment portfolios has long been a cornerstone of financial management, aiming to achieve an ideal balance between risk and return. Traditional approaches, such as the Markowitz Model (MM), Conditional Value at Risk (CVaR), and Mean Absolute Deviation (MAD), have served as foundational techniques for constructing efficient portfolios. These methods rely on historical data and assume linear relationships between asset returns (Ogunsola *et al.*, 2024). While effective in many scenarios, they often fall short in capturing the complexities of modern financial markets, particularly during periods of high volatility or economic uncertainty. As a result, there is growing interest in leveraging machine learning (ML) techniques to enhance portfolio optimization.

Machine learning offers a dynamic and adaptive framework for tackling the limitations of traditional methods. By harnessing the power of algorithms such as K-Nearest Neighbor Randomized Averaging Neural Network (K4-RANN), Evolutionary (Genetic) Algorithms (GA), and Extreme Gradient Boosting (XGBoost), ML models can process vast amounts of data, identify intricate patterns, and adapt to non-linear relationships between assets (Gunjan and Bhattacharyya, 2023). These capabilities position ML as a promising alternative for optimizing portfolios, particularly in fluctuating market conditions. However, a comprehensive comparison of traditional and ML techniques in portfolio optimization remains underexplored, especially in the context of Exchange Traded Funds (ETFs) portfolios.

Therefore, the aim of this dissertation is not targeted at building a specific optimal portfolio, rather, the goal is to consider if these techniques offer portfolio diversification benefits even when the individual assets are highly similar, as is the case in the sample used here, where a variety of different ETFs are considered in creating an ETF portfolio through various optimisation methods. ETF portfolios have become an integral part of modern investment strategies due to their unique advantages, including diversification, cost-efficiency, liquidity, and transparency (Ogunsola *et al.*, 2024). They provide exposure to a wide range of asset classes, making them an ideal vehicle for testing portfolio optimization techniques. This study also focuses on the performance

of traditional and ML-based portfolio optimization methods in ETF portfolios under varying market conditions, such as bullish and bearish trends. By examining risk-adjusted returns, diversification effectiveness, and risk mitigation capabilities, this research aims to provide practical insights for investors seeking to navigate the complexities of financial markets.

1.2 Background of the Study

In recent years, the rapid advancement of machine learning (ML) has profoundly impacted daily lives. It has introduced unique methods and solutions in various industries and sectors (Sindi, 2023). Among the different areas that have sparked interest and attention, ML has gained significant traction in asset allocation. Saxena *et al.* (2023) mentioned that wealth management and financial advisory increasingly rely on professional automated ML tools to facilitate operations. Artificial Intelligence (AI) refers to the broad field of computer science dedicated to creating systems capable of performing tasks that would typically require human intelligence (Khaleel, Jebrel and Shwehdy, 2024). These tasks include reasoning, learning, problem-solving, perception, and decision-making. Within AI, there are various subfields and methodologies, with machine learning being one of the most prominent.

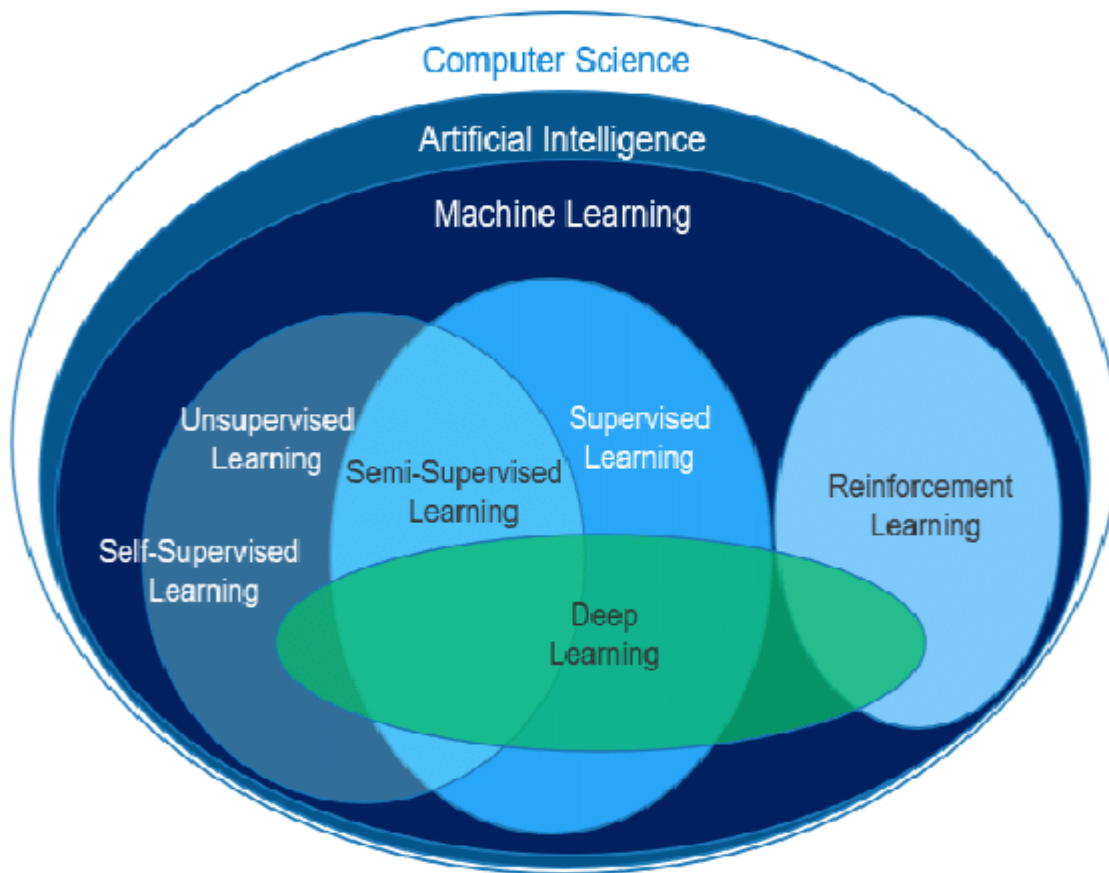


Figure 1.1 Relationship between AI, ML, and DL

Adopted from Gayatri, Srinivasu, Chaitanya and Sharma (2022)

AI is the umbrella concept encompassing systems that mimic human intelligence while machine learning is a subset of AI focusing on systems that learn from data to improve their performance over time without being explicitly programmed (Mehrotra, 2019). AI-driven systems can combine multiple ML models to create a dynamic, adaptive portfolio optimization solution (Ahamed and Raghavendra, 2023). For instance, data inputs (such as historical price data, and economic indicators), and model training (which involves ML models (e.g., GA, XGBoost) that are trained on this data to learn patterns). AI then constructs portfolios that maximize returns and minimize risks based on learned patterns, adapting to new market conditions in real-time. Thus, machine learning models for portfolio optimization are specialized applications of AI, enabling more precise, data-driven investment strategies within the broader AI framework.

Figure 1.1 illustrates the hierarchy and categories within the broader fields of Computer Science, Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) and how they relate to portfolio optimization. Computer science is the

foundational discipline that provides the computational tools and algorithms necessary for data processing, simulations, and optimizations in portfolio management. AI helps make investment decisions by mimicking human intelligence. This can include predictive models for market trends, risk assessment, and portfolio rebalancing. ML techniques analyse large datasets to identify patterns, optimize asset allocation, and predict returns or risks.

In portfolio management, ML models such as regression models predict future asset returns and classification models categorize assets into risk levels or market conditions (Bartram *et al.*, 2021). Supervised Learning involves labelled datasets where the model learns from historical data to predict asset prices, returns, or risk (Nagel, 2021). Unsupervised learning helps in clustering similar assets based on features such as volatility or correlation, which aids diversification and risk reduction. Semi-supervised learning combines both labelled and unlabelled data (Frigiola, 2024). This is useful when limited labelled financial data is available. Reinforcement learning models create an environment where decisions are made based on rewards and penalties. In portfolio optimization, this could mean adjusting weights dynamically to maximize returns while minimizing risk (Nuipian and Meesad, 2025). Deep Learning is useful for processing large, complex datasets like financial news, sentiment analysis, and non-linear patterns in asset returns (Pandya, 2024). Therefore, applying these techniques, portfolio optimization becomes more adaptive and precise, allowing for dynamic risk management, effective diversification, and robust return predictions under various market conditions.

Portfolio optimization issue emerges as one of the most challenging concerns in portfolio management (Gunjan and Bhattacharyya, 2023). The selection of asset weights within a portfolio to align with risk and return expectations amplifies the significance of this matter (Gunjan and Bhattacharyya, 2023). Despite numerous renowned models and techniques for portfolio optimization, the modern portfolio theory (MPT) continues to hold prominence due to its unique analysis of covariance (Surtee and Alagidede, 2022). MPT is a fundamental financial concept that provides a framework for optimizing the risk-return trade-off in a portfolio of assets. Harry Markowitz developed it in the 1950s and is the foundation for many portfolio management and asset allocation strategies. MPT suggests diversifying investments

across various assets with different risk and return profiles can allow investors to achieve a more efficient and stable portfolio (Geambasu *et al.*, 2013).

This dissertation assesses the effectiveness of ML in the context of portfolio optimization using a sample of ETFs (due to their structural similarities regardless of the particular market of asset the fund is linked to). During periods of significant uncertainty and fear, investors are exposed to ongoing and unforeseen difficulties. The uncertainty pertains to their portfolio investment strategies, which result in unanticipated economic circumstances. The distressed market conditions are influenced by extraordinary events beyond the scope of what can be classified as anticipated or foreseeable scenarios in everyday life (Heinonen, Karjalainen and Taylor, 2022). Consequently, the other objective of this study is to examine the response of the portfolio returns and risks constructed through the utilization of ML algorithms during market distress.

The concept of distressed market conditions encompasses extraordinary events that fall outside the realm of anticipated or foreseeable economic scenarios, often arising from crises that disrupt market stability (Taskinsoy, 2023). This study aims to investigate the response of portfolio returns and risks for ETF portfolios constructed using machine learning (ML) algorithms during such periods of market distress. By employing an event study methodology, this research evaluates the performance of ML-optimized ETF portfolios under the impact of three major economic shocks that have uniquely shaped the South African financial landscape.

The COVID-19 Pandemic (March 2020 – July 2022) which is a global health crisis that severely affected financial markets worldwide, triggering extreme volatility and economic uncertainty. While extensive literature exists on COVID-19's economic impact, there is relatively limited research on its intersection with ML-driven portfolio optimization in the South African market. The July 2021 Social Unrest and Riots, a period of severe civil disturbances following the arrest of former President Jacob Zuma, leading to widespread looting, infrastructure damage, and economic losses exceeding R50 billion (Phungula, 2024). This event presents a unique opportunity to analyse how ML-optimized portfolios responded to an episode of acute political and social instability—an often-overlooked but crucial risk factor in emerging markets. The April 2022 KwaZulu-Natal Floods which is one of the worst natural disasters in South

African history, causing severe destruction to infrastructure, supply chain disruptions, and economic losses of over R17 billion (Ngcamu, 2023). This event provides critical insight into how ML-based portfolios react to climate-induced financial shocks, an increasingly relevant area of study amid growing concerns over environmental risks.

While existing research has explored the effects of COVID-19 on financial markets, studies focusing on ML-based portfolio optimization during such crises remain scarce. Furthermore, most global studies on financial distress centre on traditional economic downturns rather than region-specific shocks. This research makes a novel contribution by incorporating distinctly South African market stressors beyond global crises like COVID-19 into the evaluation of ML-driven portfolio performance. By examining localized extraordinary events such as social unrest and climate disasters, this study provides a deeper understanding of how machine learning techniques navigate market shocks in an emerging market context. This localized approach enhances our understanding of portfolio resilience under unique stress conditions that are particularly relevant for South Africa and similar emerging economies. Ultimately, this research extends the existing body of knowledge by bridging the gap between machine learning applications in portfolio optimization and the financial impact of country-specific extraordinary events. The findings will offer valuable insights for investors, financial institutions, and policymakers seeking to enhance portfolio robustness in the face of localized market distress.

This study primarily focuses on utilising the ML approach in the specific context of the portfolio optimisation of ETF portfolios (fund of funds). A fund-of-fund (FOF) is a managed pooled vehicle that raises capital from investors to invest in multiple other private funds (Wang *et al.*, 2024). Unlike a fund that invests directly in companies, an FOF invests in other funds not limited private equity funds only. With one FOF, investors can achieve diversification across private equity markets, with a lower required minimum investment. A historical analysis shows that FOF, secondary, and co-investment funds exhibited higher upside and lower downside return potential relative to historical buyout, growth equity, and venture capital fund performance (Private Equity Perspectives, 2024). A discussion about the difference between a fund, and fund of funds (or multi-manager fund) is provided in Chapter 2.

Buetow and Hanke (2024) noted that ETFs cover various asset classes, from equities to bonds, commodities, real estate, and even alternative investments like cryptocurrencies. The versatility, rapid growth, and unique structural features of ETFs make them an interesting subject for the comparison of portfolio optimisation techniques. Using similar securities like ETFs in portfolio optimization offers numerous advantages that enhance both the efficiency and practicality of the process (Marín-Rodríguez *et al.*, 2025). ML optimization techniques have gained traction in the broader field of finance, but their adoption within ETFs in South Africa may not be as comprehensive as in other financial domains. There is significant potential for innovation and improvement in the asset allocation and portfolio optimization processes through machine learning (Manandhar, 2023). Therefore, this study assesses if ETF portfolios align well with machine learning-based optimization techniques in offering reliable historical data and aggregated exposures that improve model accuracy and are also adaptable to the South African context (natural disasters - April 2022 floods, Social Unrest and Riots - July 2021, COVID-19 Pandemic - March 2020 to July 2022).

1.3 Motivation of the Study

Currently, one of the primary concerns of investment and financial managers revolves around the pursuit of optimal decision-making at a rapid pace amidst a substantial influx of information and data within the realm of stocks and capital markets (Zanjirdar, 2020). Particularly when it comes to the augmentation of investment diversification within investment portfolios, the significance of making optimal decisions becomes highly noticeable, considering the limitations imposed by expected returns, risk levels, and the liquidity of assets, among other variables (Mandala *et al.*, 2023). Determining the desirable optimized portfolio for investment purposes, discerning the outperformance of certain stocks over others, selecting the deserving stocks for inclusion within the portfolio, and establishing the allocation of capital across these securities proves to be challenging due to market uncertainty and changing market conditions (Sindi, 2023). Traditional methods for portfolio optimization, such as Markowitz's Mean-Variance Optimization, have long been the industry standard (Kalayci, Ertenlice and Akbay, 2019). However, machine learning techniques have

gained traction in recent years due to their potential to provide more robust and adaptive solutions.

Traditional portfolio optimization methods, such as the Markowitz Model, Conditional Value at Risk (CVaR), and Mean Absolute Deviation (MAD), rely on assumptions about expected returns, risk, and correlations. While these models have been widely used, they often falter under real-world conditions, particularly in distressed markets. During periods of market turmoil, correlations between assets tend to increase, eroding diversification benefits and amplifying portfolio risk. This phenomenon is especially pronounced in ETFs, where underlying assets often exhibit high levels of similarity. Consequently, traditional models may struggle to construct robust portfolios that can withstand market volatility and deliver consistent performance (Shahid, 2024).

The primary motivation for this study was to assess and compare the performance of these approaches in the context of ETF portfolios where ETFs are assets in a portfolio created with different optimisation approaches. By doing so, the study provides investors and asset managers with a clearer understanding of the benefits and limitations of each approach, enabling them to make more informed investment decisions. Investment portfolio optimization is a crucial aspect of modern finance, aiming to construct portfolios that maximize returns while minimizing risks (Pfaff, 2016). Traditional portfolio optimization techniques, such as mean-variance optimization (MVO), have been used for decades. However, the financial landscape has evolved rapidly, characterized by increasing complexity, volatility, and interconnectedness. In recent years, ML algorithms have gained traction in various domains, including finance, due to their ability to handle large datasets, nonlinear relationships, and complex patterns (Nahar *et al.*, 2024).

On the same note, human beings possess a strong aversion towards committing errors (Sindi, 2023). This underlying characteristic of humanity stimulates a continuous pursuit of research and development to create innovative methodologies and tools to mitigate, if not entirely eradicate, perceived risks (Simonis, 2022). The advent of ML has significantly influenced and moulded our daily existence, offering fresh avenues and solutions that are applicable across a diverse range of industries and sectors (Zhou, Zhang and Wang, 2021). Portfolio managers and financial advisors

increasingly embrace artificially automated tools to accomplish their tasks, leading to a surge in the popularity of robo-advisors among investors (Bhatia *et al.*, 2021).

The motivation behind this research topic stems from the need to evaluate the performance of machine learning-based portfolio optimisation techniques against traditional methods, particularly under distressed market conditions. Kolawole (2024) mentioned that financial markets are prone to periods of instability, characterised by heightened volatility, rapid price fluctuations, and increased asset correlations. During such periods, conventional portfolio optimisation models may struggle to adapt, leading to suboptimal outcomes and potential losses for investors. Portfolios, including ETFs, are not immune to market turmoil, and investors seek strategies that can effectively manage risk and preserve capital during challenging market conditions.

There is a growing interest among academics, researchers, and practitioners in leveraging machine learning for financial applications. Understanding how machine learning techniques and investors can effectively apply them to portfolio optimisation is crucial for staying competitive in the evolving landscape of quantitative finance (Nazareth and Reddy, 2023). Therefore, the comparative study of machine learning techniques in portfolio optimisation across different asset classes seeks to evaluate the performance of various ML algorithms in constructing optimal investment portfolios. The comparison involves the different ML approaches, such as neural networks, support vector machines, decision trees, ensemble methods, and more, in optimising portfolios composed of assets from diverse classes, such as equities, fixed-income securities, commodities, and currencies.

From the realm of literature (Mukherji, Sandip and Jeong, 2021; Murthy, Shashidhar and Wald, 2023; Pitera, Marcin and Stettner, 2023), scholars have put forward alternative means and methods to construct a proficient portfolio that is capable of generating optimised returns while minimising the inclination towards risk in distressed market conditions. The significance of this subject is underscored by specific weaknesses observed in the traditional approaches (Markowitz model, mean absolute deviation, Conditional Value-at-Risk) (Mandala *et al.*, 2023). Some failings arise from determining optimal asset weights, which are highly contingent upon estimates of expected returns. Given that such estimates are often uncertain, the optimisation

process can result in unstable weights that exhibit poor performance when tested against out-of-sample data (Dai and Kang, 2022). DeMiguel, Garlappi and Uppal (2009) provided empirical evidence that a portfolio consisting of equal weights exhibits superior performance in terms of its Sharpe ratio in an out-of-sample scenario compared to both the optimal Markowitz portfolio and a range of other optimal portfolios. The question remains whether traditional portfolio optimisation techniques are powerful enough to outperform the ML approaches to portfolio optimisation in ETFs. Moreover, the matrix tends to be unstable when asset correlations increase, which typically coincides with the period when diversification becomes the most crucial and yet more challenging to achieve (Kelly and Xiu, 2023).

Financial markets are subject to volatility and distress (Dai and Kang, 2022). Market downturns, crises, and economic shocks can significantly impact the performance of investment portfolios. Given the potential benefits of machine learning techniques in adapting to changing market conditions, this study seeks to evaluate how unit trust portfolios constructed through these methods perform during periods of market distress. Understanding the resilience and effectiveness of machine learning-based portfolios in such situations is crucial for investors seeking to protect and grow their assets in turbulent times. The ongoing debate persists between the traditionalist and the ML faction that advocates for implementing ML methodologies in optimising investment portfolios. This dilemma is exacerbated by significant financial calamities, such as the depreciation of the Rand (2015-2016), credit downgrades (2017-2018), the Steinhoff accounting scandal (December 2017), and the persisting COVID-19 pandemic (2020-2022), which create a high degree of volatility within the financial markets (Sindi, 2023).

Research conducted on the application of ML in Africa and South Africa, includes that with a specific focus on language processing, medical image analysis, and astronomy (Heymans, Davel and Heerden, 2022; Andrew *et al.*, 2021; Venter *et al.*, 2020; Beukes *et al.*, 2020; Barnard and Heyns, 2020). Applying ML models, particularly in portfolio optimization, is currently absent in the South African literature. Therefore, the justification for study is not solely to assess the efficiency of ML techniques but also to provide information in the context of fund of funds portfolio optimisation, specifically, ETF portfolio optimization on how investors may maximize and reduce the risk in their

ETF portfolios. During these times of considerable uncertainty and concern, investors face constant and unforeseen challenges, which also extend to their portfolio investment strategies. This study aims to assess the performance of the ETF portfolios created using machine learning algorithms when faced with unfavourable market conditions.

1.4 Research Problem

The rapid advancement of machine learning has significantly impacted financial markets, particularly in asset allocation and portfolio optimization. Exchange-Traded Funds, as a diverse and liquid investment class, have become a critical component of modern portfolios, offering broad market exposure and diversification benefits. Despite the increasing integration of ML techniques in finance, there remains a limited understanding of their effectiveness in ETF portfolio optimization compared to traditional methods.

Existing research on portfolio optimization has largely focused on traditional models such as the Markowitz Model, Conditional Value at Risk (CVaR), and Mean Absolute Deviation (MAD). While these models provide a structured approach to asset allocation, they have notable limitations, including reliance on historical correlations, sensitivity to extreme market events, and constrained adaptability to dynamic market conditions. On the other hand, machine learning techniques, including shallow learning, deep learning, and gradient boosting, offer promising alternatives by capturing nonlinear relationships, adapting to changing market dynamics, and enhancing risk-adjusted returns. However, a comprehensive empirical comparison of traditional and ML-based portfolio optimization techniques in ETF portfolios remains an underexplored area.

ML techniques are solely noticeable in the research on developed countries such as the United States, China, England, and Japan (Nundlall and Van Zyl, 2023; Wekwete, Kufakunesu and van Zyl, 2022; Sindi, 2023), mainly focusing on stock returns predictions. Developing countries like South Africa exhibit fragmented research concerning utilising ML techniques and their comparability with conventional models in ETF portfolio optimisation. An observation by Habbab and Kampouridis (2022) and

Sindi (2023) asserts that integrating ML techniques could substantially augment the discernment of investors, analysts, and policymakers.

Furthermore, the performance of ETF portfolios optimized using ML techniques under distressed market conditions has not been extensively examined. While some studies have analysed the impact of the COVID-19 pandemic on financial markets, there is limited research on how ML-optimized ETF portfolios respond to South Africa-specific market shocks, such as the April 2022 floods and the July 2021 social unrest and riots. Understanding how ML-driven portfolio optimization performs under such conditions is essential for improving risk management and investment decision-making. While there is a fair amount of literature on COVID19 effects, there is not much on ML on COVID, and making an important contribution to our understanding of the SA market is considering SA specific shocks to the market - floods and riots.

Additionally, despite the growing body of literature on ML in portfolio optimization, a standardized evaluation framework remains lacking. Different studies employ varied datasets, performance metrics (e.g., Sharpe ratio, Sortino ratio, RMSE), and methodological approaches, making it difficult to compare the effectiveness of different ML models across asset classes and market environments. The absence of benchmark datasets and standardized evaluation metrics limits the generalizability and practical applicability of ML-driven portfolio strategies. Therefore, this study brings a unique contribution to the literature by providing foundational knowledge through systematically reviewing and comparatively analysing machine learning techniques in portfolio optimisation across all asset classes.

Thus, this study aims to address the research gaps on the inconsistency in performance measurement by employing a systematic literature review to identify trends, methodological advancements, and gaps in ML-based portfolio optimization research. To add on, the study also addresses the gap of comparative performance analysis by evaluating the effectiveness of traditional portfolio optimization methods (Markowitz, CVaR, MAD) versus ML techniques (K4-RANN, Genetic Algorithms, XGBoost) in optimizing ETF portfolios. The research gap on market condition resilience is also addressed by assessing the adaptability and robustness of ML-optimized ETF portfolios under both normal and distressed market conditions. Thus,

providing empirical evidence on the applicability of ML-based optimization techniques in the South African ETF market, an area with limited existing research. By addressing these research gaps, this study contributes to the growing field of financial technology and investment science, offering insights that can enhance decision-making for portfolio managers, institutional investors, and policymakers.

1.5 Research Objectives

The main objective of this study is to analyse the performance of traditional and machine learning portfolio optimization techniques in ETF portfolios. Consequently, this study considers the following objectives.

- i. To systematically review and analyse the implementation of machine learning techniques in portfolio optimization across different asset classes.
- ii. To systematically review and analyse the performance of machine learning techniques in portfolio optimization across different asset classes.
- iii. To assess the performance of the traditional versus machine learning portfolio optimisation techniques in exchange traded fund portfolios.
- iv. To assess how the exchange traded funds' portfolios optimized by machine learning techniques perform under distressed market conditions.

1.6 Research Questions

From the three study objectives outlined above, three research questions were asked in this study:

- i. To what extent have machine learning portfolio optimisation techniques been implemented across different asset classes?
- ii. To what extent have machine learning portfolio optimisation techniques performed across different asset classes?
- iii. How do traditional portfolio optimisation techniques perform against machine learning portfolio optimisation techniques in exchange-traded fund portfolios?
- iv. How do the exchange traded fund portfolios optimized by machine learning optimisation techniques perform under distressed market conditions?

1.7 Statement of Hypotheses

The statements of hypotheses were derived from each of your research questions, and each includes a null hypothesis (H_0). Therefore, H_1 represent null hypothesis for research question 1 and H_2 represent null hypothesis for research question up to H_4 as written below.

H_1 : Machine learning portfolio optimisation techniques have not been significantly implemented across different asset classes.

H_2 : Machine learning portfolio optimisation techniques do not significantly improve portfolio performance across different asset classes.

H_3 : There is no significant difference in the performance of traditional and machine learning portfolio optimisation techniques in exchange-traded fund portfolios.

H_4 : Machine learning-optimised exchange-traded fund portfolios do not perform significantly differently under distressed market conditions compared to normal conditions.

1.8 Significance of the Study and Contributions to Knowledge

This study makes a unique contribution to the field of portfolio management by addressing a critical gap: the resilience of Exchange-Traded Funds portfolios during economic downturns. While traditional portfolio optimization techniques have been extensively studied, the application of machine learning techniques to optimize ETF portfolios under distressed market conditions remains underexplored. This research introduces modern machine learning approaches specifically K4-RANN, Genetic Algorithms, and Extreme Gradient Boosting to assess their effectiveness in comparison to traditional models like the Markowitz Model, CVaR, and MAD.

By focusing on distressed market conditions, this study contributes to knowledge by offering a novel framework for evaluating how machine learning techniques perform when market stability is most at risk. This framework advances our understanding of portfolio optimization beyond stable market environments, providing new insights into risk mitigation strategies during periods of economic turbulence. Moreover, this research extends the academic toolkit by integrating machine learning techniques with traditional financial models, enabling a deeper exploration of adaptive and dynamic portfolio management strategies. These contributions not only expand the body of

knowledge in quantitative finance but also have significant practical implications: Fund managers - the findings can catalyse the adoption of machine learning algorithms, spurring innovation in investment strategies and leading to more sophisticated, data-driven decision-making processes. Investors - by identifying the most resilient optimization techniques, this research empowers investors with actionable insights to better protect their portfolios during market crises, reducing panic selling and extreme volatility. Financial stability - effective portfolio management strategies during distressed conditions can contribute to overall market stability, mitigating systemic risks and fostering investor confidence.

In sum, this study's comparative approach not only enriches the academic discourse on portfolio optimization under extreme market conditions but also informs real-world investment practices. Its unique contribution lies in demonstrating whether machine learning techniques offer superior risk-adjusted returns, risk mitigation, and diversification compared to traditional methods, thereby advancing both theoretical and practical dimensions of finance.

1.9 Scope of the Study

This study encompasses a detailed comparative analysis of traditional and machine learning-based portfolio optimization techniques, specifically focusing on ETF portfolios. The study is designed to provide a comprehensive evaluation of the performance of these techniques across various market conditions, including periods of distress. The analysis was conducted in the context of the asset classes, optimization techniques and market conditions. While the primary focus was on ETF portfolios, the study also included a systematic review of the application of machine learning techniques across different asset classes. This would provide a broader understanding of the effectiveness and adoption of these techniques. The study compared traditional optimization techniques (such as mean-variance optimization) with machine learning techniques (such as genetic algorithms, neural networks, and reinforcement learning) in terms of their effectiveness in portfolio optimization. Special emphasis was placed on evaluating the performance of ETF portfolios constructed using machine learning techniques during distressed market conditions. This would help in understanding the robustness and resilience of these portfolios.

The first objective aimed to systematically review and comparatively analyse the implementation and performance of machine learning techniques in portfolio optimization across different asset classes. A systematic literature review approach was utilised to provide a solid foundation for research and helps identify research gaps, enhances research rigor, informs research design and methodology decisions, generates novel research ideas (Sauer and Seuring, 2023). Systematic literature review involves a structured and rigorous approach to identifying, evaluating, and synthesizing relevant research literature on a specific topic.

The second objective focused on comparatively assessing the performance of the traditional versus machine learning portfolio optimisation techniques in exchange traded funds. Three traditional techniques namely, Markowitz model, CVaR and MAD and three categories of ML techniques, namely shallow learning (enhanced radial basis artificial neural network - K4-RANN), deep learning (Evolutionary Genetic Algorithms), and extreme gradient boosting (XG Boost) were utilized for ETF portfolio optimization. Their optimized results were compared using the three-performance metrics namely, mean portfolio returns, standard deviation and the Sharpe ratio. The higher the Sharpe ratio will imply higher performance of the ETF optimized portfolio. The lower the standard deviation of an ETF optimized portfolio implies that it carries low risk on the returns realised. Lastly, the higher the mean ETF portfolio returns implies better performance. These models and the performance metrics enabled this analysis to have a conclusion the portfolio optimization performance between traditional and machine learning models.

The final objective assessed how the exchange traded funds' portfolios constructed by machine learning techniques perform under distressed market conditions. The event study methodology was utilised to assess the impact of specific distressed events (natural disasters (April 2022 floods), Social Unrest and Riots (July 2021), COVID-19 Pandemic (March 2020 to July 2022) on the performance of the optimized ETF portfolios. A comparison of portfolio mean-returns, the level of risk (standard deviation) and the shape ratio during the distressed period to those in normal period. Furthermore, the results were also used to assess the robustness and effectiveness of the machine learning optimization techniques in handling market shocks. These event periods were analysed using event study methodology to assess how portfolios

optimized using ML techniques responded to major shocks, but the overall model training and performance evaluation drew from the entire 2013 - 2023 dataset.

1.10 Structure of Thesis

This doctoral study takes a form Doctor of Philosophy (PhD) by thesis as a monograph in the style of an introduction and theoretical review, followed by five chapters, each aligned to a single objective of the overall dissertation.

Chapter 1 - Background of the Study. This chapter provides the background context of the study. It discusses the research problem which informs the motivation for the study, outlines the objectives and research questions, and briefly introduces the methods of analysis.

Chapter 2 - Theoretical Framework. This chapter provides an overview of theories relevant to portfolio theory, diversification, and portfolio optimization. Theories discussed include modern portfolio theory, efficient market hypothesis and adaptive market theory. While empirical literature is discussed in Chapters 1, 2 and 3 as appropriate, the focused empirical literature reviews of each of the three objectives are discussed in the relevant section in Chapters 4, 5 and 6. This applies to the discussion of data and methodology too.

Chapter 3 presents a critical analysis of machine learning models used in portfolio optimization. There are three categories of machine learning techniques that are used in portfolio optimization. To add on, mathematical representations for each group and the basic parameterisation for each sub element of the three categories with their loss functions or objective functions were unpacked.

Chapter 4 presents the first objective and makes a significant contribution to the literature by systematically reviewing and comparatively analysing machine learning techniques in portfolio optimization across different asset classes. This chapter extends beyond the commonly studied stocks and bonds, offering a broader perspective on the application of ML techniques in various financial markets. By synthesizing existing research, it identifies trends, methodological advancements, and gaps in the literature, providing a foundation for future studies. This chapter contributes methodologically by critically evaluating how ML techniques have been applied,

highlighting their effectiveness, limitations, and adaptability across asset classes. It updates our understanding of financial technology by assessing the extent to which ML has transformed portfolio optimization, thereby informing both academic inquiry and industry practice. From a policy perspective, the findings offer insights into regulatory considerations and risk management strategies associated with ML-driven investment decisions. Overall, Chapter 4 enhances the discourse on ML in finance, guiding both theoretical advancements and practical applications in portfolio management.

Chapter 5 presents the second objective and makes a novel methodological contribution by empirically comparing traditional portfolio optimization techniques (Markowitz Model, CVaR, and MAD) with machine learning-based techniques (shallow learning, deep learning, and gradient boosting) specifically within ETF portfolios. This chapter advances the literature by addressing a critical gap, whether ML-based approaches offer superior performance, adaptability, and resilience compared to traditional models. By conducting a rigorous, data-driven assessment, it enhances our understanding of how ML techniques function in real-world ETF portfolio optimization. This chapter contributes methodologically by implementing and evaluating state of the art ML models within the structured framework of portfolio optimization, ensuring robust statistical validation. It updates the existing body of knowledge by determining whether ML techniques provide tangible benefits in terms of returns, risk reduction, and diversification within ETF portfolios. The findings have significant implications for policy and practice, informing investment strategies, regulatory considerations, and financial innovation. If ML techniques demonstrate superior performance, this could justify their broader adoption, reshaping portfolio management practices and influencing financial decision-making at institutional and retail levels.

Chapter 6 presents the third objective and makes a significant contribution by evaluating the performance of ETF portfolios optimized using machine learning techniques under distressed market conditions capturing three key events, pertinent to the South African context (natural disasters - April 2022 floods, Social Unrest and Riots - July 2021, COVID-19 Pandemic - March 2020 to July 2022). This unique contribution focuses on portfolio behaviour during periods of market turmoil offers critical insights into the robustness and adaptability of ML-optimized portfolios, an area

that remains underexplored in the existing literature. By assessing how these portfolios perform during times of economic uncertainty and market stress, the chapter enhances our understanding of the effectiveness and limitations of ML methods in real-world, high-risk environments. The chapter contributes methodologically by applying advanced ML techniques to analyse portfolio resilience under distressed conditions, a crucial aspect of modern portfolio management. It updates the existing body of knowledge by providing empirical evidence on how ML-optimized ETF portfolios behave in extreme market conditions, offering novel insights into their suitability for managing risk. The findings have direct implications for investors, asset managers, and policymakers, as they offer practical guidance on adapting portfolio strategies during times of market stress. This knowledge is crucial for improving decision-making, informing regulatory policies, and shaping future strategies in risk management and financial stability.

Chapter 4, 5 and 6 include the introduction, empirical review, empirical methods, results and discussions relevant to each method, and are thus fundamentally a set of working papers around the overarching theme of this doctoral dissertation. This approach of a series of working papers, rather than a traditional tome style thesis, is well established in the fields of Economics and Finance. An overarching conclusion and summary are presented in Chapter 7.

CHAPTER 2: THEORETICAL FOUNDATIONS

2.1 Introduction

This chapter provides an overview of the theories underpinning this study. The original concepts of this were equities bonds and cash, however, in the more modern space, with growing number of assets, there are types of portfolios, including those that take a fund of funds approach. Therefore, ETF portfolios are potentially useful for exploring portfolio optimisation techniques. The first section elucidates the underlying theoretical foundations of portfolio optimisation: modern portfolio theory, efficient market hypothesis and adaptive market hypothesis, and how they are linked to traditional understandings of portfolio optimisation. The second explains the historical context, development, and application of machine learning in portfolio optimisation, including the types of machine learning algorithms. In addition, challenges and considerations in applying machine learning to portfolio optimisation and recent advances and innovations in machine learning for portfolio optimisation are also discussed. The last section discusses the ETF creation and redemption process and highlights the benefits and concerns related to ETFs concerning portfolio optimisation.

2.2 Theories Underpinning Portfolio Optimization

This section explores the theoretical underpinnings of portfolio optimisation, offering a comprehensive view of its evolution and the principles driving its application in both traditional and machine-learning contexts. Understanding these theories is critical to assessing various optimisation methods' effectiveness and suitability for different investment scenarios.

2.2.1 Modern Portfolio Theory (MPT)

The theoretical foundations of Modern Portfolio Theory (MPT) stem from Markowitz's (1952) mean-variance framework. Markowitz (1952) was the first to introduce the concept of the risk-return trade-off in investment decision-making, along with a mathematical approach to portfolio selection. The work provided a systematic method for quantifying risk, minimizing it, and allocating capital across individual securities. Markowitz (1952) demonstrated that portfolio selection should be based on the expected return and variance of the entire portfolio rather than focusing solely on maximizing the returns of individual securities. Markowitz (1952) challenged the conventional belief that investors prioritize returns without considering risk,

highlighting the importance of diversification (Swisher and Kasten, 2005). He demonstrated that the expected return, variance, and correlation of assets influence the risk-return trade-off of investment opportunities.

As a result, constructing an optimal portfolio involves more than just considering the expected returns of individual asset classes or securities; it primarily depends on how the returns of one security correlate with the returns of others within the portfolio (Kim and Francis, 2013). Therefore, the theoretical foundation of diversification is upheld when the combination of asset classes or individual securities maximizes the portfolio's expected return for a given level of risk. Portfolio optimisation involves ascertaining the most suitable amalgamation of assets per predetermined goals (Gunjan and Bhattacharyya, 2023). These goals might encompass the maximisation of returns, the minimisation of risk, or a blend of both. Each optimisation method employs distinct portfolio metrics as fitness functions (Gunjan and Bhattacharyya, 2023). Due to the substantial number of available assets, manually selecting the appropriate set of assets is challenging and necessitates using advanced techniques.

Asset allocation, a fundamental area of research in asset management, is a crucial aspect of portfolio optimisation (Zhang *et al.*, 2020a). Moreover, it concerns identifying a set of investment assets that fulfil the criteria of an individual investor while simultaneously striving to optimise a specific objective (Ferreira *et al.*, 2021). Sophisticated models empower investors to incorporate their expectations and level of confidence. At the same time, the utilisation of extensive datasets enables the integration of market sentiments or other macroeconomic factors, such as the growth of the gross domestic product or inflation (Zhu *et al.*, 2020). Machine learning models play a crucial role in addressing this complex matter by autonomously deducing conclusions from vast data that encompass numerous input variables (Perrin and Roncalli, 2020).

To formulate the mean-variance portfolio, it is assumed a given portfolio consists of χ number of assets with w_i as the weight or proportion of the investor's wealth allocated to the i th asset in a portfolio, such that the sum of the weights of all assets in the portfolio is one, $\sum_{i=1}^n w_i = 1 \dots i = 1, 2, 3 \dots n$. The expected returns (r_p) and the variance (σ_p^2) of the portfolio can be expressed as follows:

Expected returns (r_p) of the portfolio:

$$E(r_p) = \sum_{i=1}^n w_i E(r_i) \quad (2.1)$$

Variance (σ_p^2) of the portfolio:

$$\sigma_p^2 = \sum_{i=1}^x \sum_{j=1}^x w_i w_j \sigma_{ij} \quad (2.2)$$

Where σ_{ij} is the covariance of the expected return of assets i and j .

The expected return of a portfolio is calculated as the weighted sum of the expected returns of its individual assets. However, portfolio risk is not merely the sum of individual asset variances (Benati and Conde, 2022). Instead, it is determined by both the variances of individual assets and the covariance between them (Irsan, Priscilla and Siswanto, 2022). By appropriately assigning asset weights, portfolio variance can be minimized. The portfolio construction process underscores the importance of holding assets with diverse characteristics to achieve effective diversification. Combining securities with low return covariances helps reduce the total variance of the portfolio (Mihalache, Mihalache, and Ende, 2021). The expected return and variance on all possible portfolios are determined by the weights w_1 and w_2 as follows:

$$E(r_p) = w_1 E(r_1) + w_2 E(r_2) \quad (2.3)$$

$$\sigma_p^2 = w_1^2 \sigma_1^2 + w_2^2 \sigma_2^2 + 2w_1 w_2 \rho_{12} \sigma_1 \sigma_2 \quad (2.4)$$

$$\sigma_p = (\sigma_p^2)^{1/2} \quad (2.5)$$

$$(Covariance) \sigma_{12} = \rho_{12} \sigma_1 \sigma_2 \quad (2.6)$$

When two securities are perfectly positively correlated, ρ_{12} , is 1, the portfolio's standard deviation becomes the weighted average of their individual standard deviations. In this case, both the expected return and standard deviation follow a linear pattern, increasing proportionally as the allocation to each security changes (Chatterjee, 2021). According to Castro *et al.* (2022), the optimal allocation between a

risky portfolio and a risk-free asset is determined by an investor's risk tolerance. Risk-averse investors prefer to minimize risk and, when faced with assets offering the same expected return, will select the one with lower risk. In contrast, Di Toro (2022) noted that risk-seeking investors are indifferent to risk levels and will prioritize investments with the highest expected return. As Chan *et al.* (2020) explained, risk-averse investors aiming to maximize utility based on risk and return tend to allocate more funds to the risk-free asset, whereas risk-seeking investors favour holding a larger portion of the risky asset portfolio.

Investors will opt for higher-risk assets only if the expected return increase outweighs the corresponding rise in risk, with the risk-return trade-off guiding asset allocation decisions (Gambeta and Kwon, 2020). The set of optimal portfolios available to an investor is represented by the efficient frontier, which is derived using quadratic programming by either maximizing expected return for a given level of risk or minimizing risk for a specified expected return. Research has shown that asset allocation accounts for over 90% of a portfolio's total returns (Brinson, Hood and Beebower, 1986; Ibbotson and Kaplan, 2000). According to Ibbotson and Kaplan (2000), differences in asset allocation primarily determine the relative performance of multi-asset class funds, while other contributing factors include investment style, market timing, trading costs, and security selection (Andonov, Bauer and Cremers, 2012).

Li (2024) mentioned that MPT provides a foundation for assessing the performance of traditional portfolio optimisation techniques based on its principles. It emphasises the importance of diversification and the trade-off between risk and return (Geambasu *et al.*, 2013). When comparing traditional methods to machine learning portfolio optimisation techniques, MPT can serve as a benchmark for evaluating the effectiveness of the new methods. Machine learning models can optimise portfolios by considering a more comprehensive range of data and factors than traditional methods. MPT can help assess whether these new techniques result in portfolios that provide better risk-adjusted returns, a key objective in portfolio management.

Li (2024) stated that Modern Portfolio Theory serves as a fundamental framework for evaluating the performance of traditional portfolio optimization techniques. It highlights

the significance of diversification and the balance between risk and return (Geambasu *et al.*, 2013). When comparing traditional approaches to machine learning-based portfolio optimization, MPT can act as a benchmark for assessing the effectiveness of these newer methods. Machine learning models can enhance portfolio optimization by incorporating a broader set of data and factors than traditional techniques. MPT helps determine whether these advanced methods lead to portfolios with superior risk-adjusted returns, a key objective in portfolio management.

Overall, the risk element of MPT can be assessed using various mathematical formulations. Additionally, this risk can be mitigated by employing the principle of diversification, which involves judiciously selecting a well-balanced assortment of investment assets (Nam, 2023). By doing so, the collective risk factors of these assets are expected to be lower than those associated with investing in a single asset or asset class. Diversification lies at the heart of MPT and draws upon the well-established adage of not placing all of one's resources in a single investment (Fabozzi, Gupta and Markowitz, 2002; McClure, 2010; Veneeya, 2006).

MPT has shortcomings that can be addressed by integrating machine learning (ML) techniques. MPT assumes that asset returns follow a normal distribution, which is not always true. ML techniques can better handle non-linear relationships and non-normal distributions, allowing for more accurate modelling of asset returns (Simar, 2023). Osei-Brefo (2024) also postulated that MPT relies on static inputs such as expected returns, volatilities, and correlations, often estimated using historical data. ML can continuously update these parameters by incorporating new data and adjusting to changing market conditions, providing more dynamic and adaptive portfolio allocations.

Moreover, MPT faces challenges when dealing with a large number of assets due to the curse of dimensionality, which can result in computational difficulties and overfitting. Machine learning techniques, such as dimensionality reduction methods like principal component analysis (PCA), can help simplify the problem by reducing the number of features while retaining essential information, making portfolio optimization more manageable (Dominguez, 2023). Additionally, MPT's results are highly sensitive to input estimates, including expected returns and correlations, which

are prone to estimation errors. Machine learning models can address this issue by integrating alternative data sources, such as sentiment analysis from news articles and social media, to enhance the accuracy of return and correlation predictions (Jacquier *et al.*, 2022).

Bloch (2024) mentioned that MPT assumes stationarity in asset returns, which may not hold over longer time horizons or during significant market shifts. ML techniques, particularly those designed for time series forecasting, can better capture non-stationarity and structural breaks in asset returns, leading to more robust portfolio construction. Moreover, Zhang, Zohren and Roberts (2020) added that MPT primarily uses volatility to measure risk, overlooking other risk factors such as tail risk or downside volatility. ML algorithms can incorporate a broader range of risk measures and factor in asymmetric risk-return profiles to construct more resilient portfolios to extreme market events. MPT also assumes rational investor behaviour and efficient markets, neglecting the influence of behavioural biases and market inefficiencies. ML techniques can incorporate behavioural finance insights to account for investor sentiment, market anomalies, and other behavioural factors in portfolio construction.

Therefore, by leveraging machine learning techniques, investors can potentially overcome these limitations of MPT and develop more sophisticated and adaptive portfolio optimisation strategies that are better aligned with real-world market dynamics and investor preferences. MPT's concepts are particularly relevant for assessing portfolio performance under distressed market conditions (Said, 2022). It highlights the importance of correlation and covariance among assets and the impact of diversification on reducing portfolio risk (Sindi, 2023). Therefore, MPT can be used to evaluate how machine learning-constructed portfolios perform regarding risk management and returns during such challenging market scenarios. MPT has been widely used in institutional portfolio management but criticised for poor performance during market crises. However, based on optimisation, it requires enhancements for real-world applicability.

2.2.2 Portfolio Optimization for Fund of Funds

Fund of funds (FOF) is an investment vehicle that pools money to invest in other funds rather than directly in individual securities. The primary objective of portfolio

optimization for a FOF is to maximize returns while minimizing risks through diversification across various underlying funds (Larsen and Resnick, 2012). The key considerations for portfolio optimization of FOF includes diversification thus spreading investments across different fund types, such as equity, bond, or alternative asset funds. Optimizing this FOF portfolio involves selecting a combination of funds to balance risk and return based on the investor's goals. FOF managers aim to create a portfolio that delivers the best possible risk-adjusted returns. This involves considering metrics like the Sharpe ratio, Sortino ratio, or Conditional Value at Risk (CVaR) to assess the performance of the portfolio under different market conditions (Davies, Kat and Lu, 2009).

To add on, since funds often invest in overlapping assets, it is crucial to account for correlations among the underlying funds (Fricke and Wilke, 2023). Portfolio optimization methods aim to minimize exposure to highly correlated funds to enhance diversification (Al Janabi, 2021). Costs consideration is also important since managing a FOF involves costs such as management fees and expense ratios of the underlying funds. Optimization must take these into account to ensure net returns are sufficient to meet investment goals. Furthermore, constraints are another important consideration because minimum and maximum allocations, regulatory requirements, or sector/region exposure limits influence the optimization process (www.oldmutualinvest.com).

ETFs, which are publicly traded funds that track specific indices, commodities, or asset classes, are increasingly used in portfolio optimization due to their low cost, liquidity, and transparency (Buetow and Hanke, 2024). When focusing on ETF portfolios as a subset of a FOF portfolio optimization, ETF portfolios can be designed to represent a mix of asset classes (for example, equities, bonds, commodities) or sectors (for example, technology and healthcare), replicating the diversification benefits of traditional FOFs. ETFs trade on exchanges, enabling real-time pricing and liquidity. This is advantageous for optimization, as it allows frequent rebalancing based on changing market conditions. Since ETFs often track indices, they provide a straightforward way to gain exposure to broad market performance. Optimizing an ETF portfolio involves selecting ETFs that align with desired risk-return characteristics and strategic objectives (Karoui, Sayari, Dammak and Jeribi, 2024).

Therefore, in this study, focusing on ETF portfolios allows for a practical application of portfolio optimization concepts which includes performance analysis where comparing risk-adjusted returns of ETF portfolios optimized using traditional methods (for example, Markowitz Model, CVaR) and advanced techniques (for example, machine learning models like K4-RANN, Genetic Algorithms, and Extreme Gradient Boosting). Additionally, stress testing concept is examined by examining how the optimized ETF portfolios perform under distressed market conditions, aligning with assessing machine learning resilience during market downturns. As well as, investigating how economic factors and market conditions influence the risk and return dynamics of ETF portfolios. This approach not only contributes to the broader understanding of portfolio optimization but also provides actionable insights for practitioners and investors seeking to construct efficient ETF portfolios.

2.2.3 Efficient Market Hypothesis (EMH)

The Efficient Market Hypothesis (EMH), introduced by Fama (1970), suggests that asset prices are determined by the information related to a specific security. New information is quickly reflected in security prices, ensuring that securities are always fairly priced. This suggests that active management techniques that use historical price trends or company performance to find undervalued or overvalued assets are not lucrative, resulting in portfolios that are not adequately diversified in an efficient market (Bodie, Kane and Marcus, 2022). Markets are said to follow a random walk, where asset price movements are unpredictable and fluctuate in response to the random availability of new information. According to Mateus and Hoang (2021), competition among market participants to profit from new information helps maintain market efficiency. The EMH asserts that the market portfolio, which incorporates all available information, is the optimal portfolio that every investor should hold. Much of traditional finance theory is built on the EMH (Kamoun and Ibenrissoul, 2022). In an efficient market, investors cannot achieve abnormal profits from capital market transactions (Nyakurukwa and Seetharam, 2023). However, higher profits can be earned by investing in higher-risk assets.

The Efficient Market Hypothesis (EMH) is grounded in the idea that all available information in the market is fully, accurately, and instantly reflected in security prices (Fama, 1970). This means that in an efficient capital market, security prices represent

the best estimate of their intrinsic value. The EMH plays a crucial role in investment decision-making, as it suggests that if markets are inefficient in pricing securities, investors could potentially earn abnormal returns by exploiting these inefficiencies. Market participants are assumed to act rationally and make optimal decisions. According to He *et al.* (2021), security prices are influenced by various types of information. As a result, the EMH can be categorized into three levels of market efficiency based on the type of information that influences asset prices: weak-form efficiency, semi-strong-form efficiency, and strong-form efficiency. A weak-form efficient market means that information contained in historical prices cannot be used to predict future prices. This form assumes that asset prices follow a random walk, implying that investors cannot earn abnormal returns by using strategies based on analysing past prices or trading volumes to forecast future prices.

Investors are unable to secure higher profits from market inefficiencies over the long term. Studies by Appiah-Kusi and Menyah (2003), Simons and Laryea (2006), and Ntim *et al.* (2011) have investigated the weak-form efficiency of African stock markets, while research by Jefferis and Smith (2005), Simons *et al.* (2006), and Heymans and Santana (2018) have provided evidence supporting weak-form efficiency in South African stock markets. These findings indicate that asset returns behave as if they follow a random walk, with past price movements offering no predictive value for future security prices or returns. In contrast, Gräter and Struweeg (2015) reported that South African stock markets were inefficient.

In a semi-strong efficient market, security prices swiftly incorporate all publicly available information. As a result, investors cannot achieve excess returns by relying on fundamental or technical analysis to forecast future prices (Bodie *et al.*, 2022). However, early empirical studies challenged this notion, revealing that financial ratios and economic indicators possess significant predictive power for stock returns (Lewellen, 2004; Chen, 2009). Campbell and Shiller (1988) demonstrated that the dividend yield ratio was a strong predictor of stock market returns. This finding was later reinforced by Lewellen (2004), who exhibited that dividend yield ratios outperformed price-earnings and book-to-market value ratios in predicting US stock market returns.

Markets are considered strong-form efficient when the prices of securities reflect both private and public information (Mustafa and Ahmed, 2020). Hence, abnormal profits from insider information for investors, and this form represents complete market efficiency, means that a company's management will not be able to earn abnormal profits using insider information (Ongere, 2020). Information asymmetries that result in informed traders generating abnormal profits suggest that markets are not solid and efficient. Ogunsakin (2020) demonstrated the importance of the cost and quality of information and the degree of risk aversion as vital determinants of equilibrium informativeness.

Markets are deemed strong-form efficient when security prices fully incorporate both public and private information (Mustafa and Ahmed, 2020). Consequently, investors cannot achieve abnormal profits using insider information, as this level of efficiency implies that even a company's management cannot gain an advantage through privileged knowledge (Ongere, 2020). The presence of information asymmetries that enable informed traders to earn abnormal profits indicates that markets are not perfectly efficient. Ogunsakin (2020) highlighted the significance of information cost, quality, and the degree of risk aversion as key factors influencing equilibrium informativeness.

EMH suggests that consistently beating the market using any method, including machine learning techniques, is difficult. However, Chen *et al.* (2021), Asawa (2021), Snow (2020), and Choi and Kim (2024) have shown that specific machine-learning approaches can sometimes outperform traditional methods, at least in particular market conditions or for certain asset classes. By comparing the performance of traditional and machine learning portfolio optimisation techniques in ETFs, researchers can evaluate whether machine learning offers any advantages regarding risk-adjusted returns or other metrics.

EMH suggests market prices reflect all available information, including historical price data and other relevant factors. Therefore, any attempt to use machine learning techniques to predict future prices and optimise portfolios may face challenges in consistently outperforming the market (Padhi *et al.*, 2022). However, proponents of machine learning techniques argue that they can uncover patterns or relationships in

data that may not be immediately evident, potentially providing an edge in portfolio optimisation (Khoa and Huynh, 2021).

Machine learning models can adapt and evolve as new data becomes available (Janiesch, Zschech and Heinrich, 2021). This flexibility contrasts with the EMH assumption that market prices instantaneously reflect all available information. ML algorithms could continuously learn from market dynamics, adjusting real-time portfolio allocations to capitalise on emerging opportunities or mitigate risks. While EMH assumes market participants are rational and fully utilise available information, behavioural finance suggests otherwise. Machine learning techniques can analyse vast amounts of data on investor behaviour, sentiment, and psychological biases (Kumbure *et al.*, 2022). By incorporating insights from behavioural finance into portfolio optimisation strategies, ML algorithms may exploit market inefficiencies caused by irrational investor behaviour, challenging the EMH assumption of market efficiency (Li, 2024).

Moreover, EMH assumes that all relevant information is already reflected in stock prices. However, machine learning enables the analysis of non-traditional data sources, such as satellite imagery, social media sentiment, or even foot traffic data, which may provide unique insights into companies or industries. Integrating such alternative data into portfolio optimisation models could offer a competitive edge by identifying trends or market inefficiencies not captured by EMH. EMH showed mixed results, with some markets appearing efficient while others did not. In addition, EMH supports the development of passive investment strategies but is challenged by machine learning models that predict market anomalies (Azevedo *et al.*, 2023). Therefore, the research objectives outlined can provide valuable insights into the effectiveness of machine learning approaches in portfolio management, both in theory and in practice.

2.2.4 Adaptive Market Hypothesis (AMH)

The Adaptive Market Hypothesis is a hybrid theory that integrates elements of the Efficient Market Hypothesis (EMH) and behavioural finance. While theoretically significant, Modern Portfolio Theory (MPT) has been widely debated and criticized for its basic assumptions, which often fail to align with real-world market dynamics

(Najera, 2021). MPT is founded on the assumption of market efficiency (Lekhal and El Oubani, 2020). According to the EMH, asset prices in financial markets fully reflect all available information at any given time, making it impossible for investors to consistently achieve above-average risk-adjusted returns (Shleifer, 2000). However, real-world market inefficiencies have frequently been observed (Goldstein, 2023).

Furthermore, market anomalies indicate that markets do not consistently evolve toward efficiency over time. While the Efficient Market Hypothesis assumes a static market, this view has been contested (Lo, 2004). The Adaptive Market Hypothesis proposes that although a relationship between risk and return exists, it fluctuates over time in response to changing market conditions (Lo, 2005). According to the AMH, the presence of irrational investors and arbitrage opportunities is essential for markets to move toward efficiency, as suggested by the EMH. However, due to the market's continuous evolution, it never attains a permanent state of efficiency.

The AMH recognises that market conditions are not static and can go through various phases, including periods of distress or crisis (Lo and Zhang, 2024). In such situations, traditional portfolio optimisation techniques may not perform as expected, as they often rely on assumptions of market stability and linear relationships between assets (Bigdeli and Bengtsson, 2019). The AMH suggests that market participants adapt strategies based on available information and learning experiences (Munir, Sukor and Shaharuddin, 2022). In the context of portfolio optimisation, traditional optimisation techniques may not always effectively capture the complexity and changing dynamics of markets. By systematically reviewing and analysing machine learning techniques, which are designed to adapt to the evolving patterns and learn from data, researchers can assess whether these techniques better align with the principles of the AMH in optimising portfolios across various asset classes.

Traditional portfolio optimization techniques often rely on assumptions about market behaviour that may not hold in all market conditions (Zanjirdar, 2020). The AMH suggests that markets are adaptive and may exhibit changing dynamics over time (Kılıç, 2020). By comparing the performance of traditional techniques with machine learning-based approaches in ETFs, researchers can evaluate which method better accommodates the adaptive nature of markets as proposed by the AMH. This

assessment can provide insights into whether machine learning techniques offer advantages in navigating the complexities of ETFs under varying market conditions.

Al Janabi (2022) explained that distressed market conditions pose significant challenges for traditional portfolio optimization techniques, as they may not adequately account for sudden shifts in market dynamics and investor behaviour. According to the AMH, market participants adapt their strategies in response to changing conditions, including during periods of market distress (Enow, 2022). By assessing how ETF portfolios constructed using machine learning techniques perform under such conditions, researchers can explore whether these techniques demonstrate greater resilience or effectiveness in adapting to and navigating through turbulent market environments, consistent with the principles of the AMH. AMH has increasing empirical support from studies on market evolution and behavioural finance. Therefore, it is highly relevant, as machine learning models can adapt to changing market conditions and investor behaviour.

In essence, while traditional asset pricing models like APT assume market efficiency and rely on linear relationships between factors and asset returns, ML techniques have the potential to challenge these assumptions by uncovering non-linear patterns, handling big data, accounting for behavioural biases, and adapting to changing market conditions (Economics, 2021). Integrating ML into portfolio optimization processes could lead to more robust and effective investment strategies (Gunjan and Bhattacharyya, 2023).

2.2.5 MPT, EMH and AMH in Portfolio Optimization

Figure 2.1 below provides a simplified overview of the interrelation and evolution of MPT, EMH, and AMH. A discussion on key concepts and potential areas of conflict and evolution of these concepts follows after.

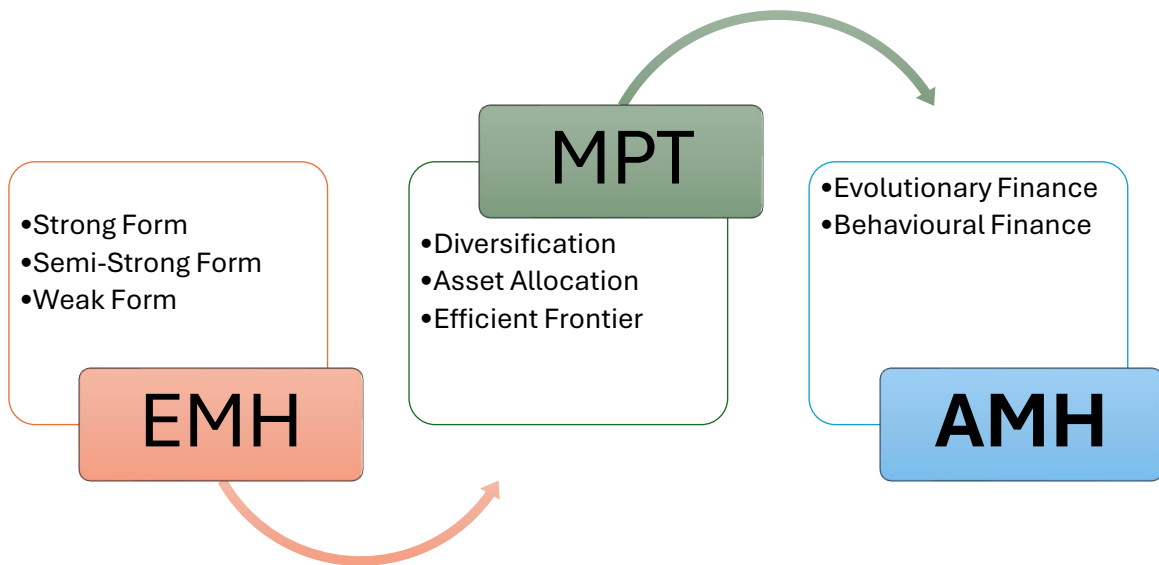


Figure 2.1 Interrelation and Evolution of MPT, EMH, and AMH

Source: Author's compilation (2024)

Khairajani (2023) stated that the EMH posits that markets efficiently incorporate all available information into asset prices and exists in three forms: weak, semi-strong, and strong. The weak form asserts that past price and volume data cannot predict future prices, the semi-strong form suggests that stock prices already reflect all publicly available information, and the strong form proposes that even insider information is rapidly integrated into prices. MPT emphasizes the efficient frontier, which represents the set of optimal portfolios that maximize expected returns for a given level of risk (Akkaya, 2021). It highlights the importance of diversification and efficient asset allocation to enhance portfolio performance. Nyakurukwa and Seetharam (2023) noted that the AMH incorporates elements of behavioural finance into the EMH framework, recognizing that market participants are not always rational and that market efficiency evolves over time. AMH integrates concepts from evolutionary finance, acknowledging that market dynamics and investor behaviour shift due to technological advancements, regulatory changes, and evolving investor preferences.

However, MPT and EMH have traditionally been closely aligned, with MPT relying on the assumption of market efficiency to construct optimal portfolios (Gunawan, 2024). Scherf (2024) stipulated that the behavioural finance challenges the notion of perfect

market efficiency by highlighting instances of irrational behaviour among investors, leading to potential conflicts between EMH and some aspects of AMH. Lo and Zhang (2024) added that AMH represents an evolution of EMH that acknowledges the role of investor psychology and market dynamics in shaping asset prices. It suggests that market efficiency is not a static concept but rather adapts to changing conditions and investor behaviour. Ziane *et al.* (2024) explained that technological advancements in portfolio optimization have led to the development of sophisticated quantitative models and algorithms that challenge some traditional assumptions of MPT and EMH. These advancements have influenced the evolution of both theories, with practitioners incorporating new tools and techniques to enhance portfolio management strategies.

MPT and EMH are often used together and MPT provides the framework for constructing portfolios, while EMH guides investors in making decisions based on available information (Akkaya, 2021). AMH complements both MPT and EMH by recognizing the limitations of human rationality and market efficiency. It offers a more realistic view of market behaviour by incorporating elements of investor psychology and evolutionary processes (Dowlin and Lucey, 2023). MPT assumes that investors are rational and that markets are efficient, which aligns with EMH (Gunawan, 2024). However, AMH challenges these assumptions by acknowledging the presence of irrational behaviour and market inefficiencies. Critics argue that MPT and EMH may oversimplify market dynamics and fail to account for real-world complexities, such as investor emotions, market bubbles, and systemic risks (Woode *et al.*, 2023). AMH provides a more nuanced perspective by incorporating these factors into its framework.

Over time, MPT and EMH have faced criticisms regarding their assumptions and predictive power. In response, researchers have refined these theories and developed alternative approaches, such as AMH (Simar, 2023). Technological advancements, such as high-frequency trading and big data analytics, have challenged the efficiency of markets and prompted revisions to traditional theories. Academic critique has spurred innovation in finance theory, leading to the development of more robust models that better capture the complexities of real-world markets (Haberly and Wójcik, 2022). Academic criticized MPT for oversimplification and unrealistic assumptions, challenged EMH by empirical evidence of anomalies and behavioural finance research

and AMH gaining traction for its realistic and adaptive approach to market dynamics. The evolution and response to new data gave birth to big data and artificial intelligence thus enhanced data analysis capabilities challenge traditional MPT and EMH, supporting AMH.

Therefore, there are still gaps in the literature given the MPT principles, there is need for dynamic and realistic models that incorporate market conditions, behavioural elements, and transaction costs. Focusing on EMH, further investigation into market anomalies and the role of information asymmetry and on AMH, more empirical studies to validate the hypothesis and explore its applications in algorithmic trading and AI-driven strategies. Therefore, these theories are relevant in contemporary portfolio management because each theory offers valuable insights but has limitations. Furthermore, a holistic approach that combines elements of MPT, EMH, and AMH may provide the most robust framework for portfolio management in the modern era.

2.3 Theoretical Foundations of ETFs

ETFs are closely linked to mutual funds; however, these funds differ in several aspects. Remarkably, ETFs are regarded as hybrid products that enjoy both open-end and closed-end funds (CEFs) benefits. Specifically, ETFs trade on exchanges like CEFs, following open-end funds' in-kind creation and redemption process (Madhavan, 2014). ETFs' unique creation and redemption process enables these funds to possess several advantages over traditional mutual funds. This section discusses the creation and redemption process of ETFs. The key benefits and concerns associated with ETFs are also highlighted.

2.3.1 ETF Creation and Redemption Process

An exchange-traded fund is a pooled investment vehicle comprising a mix of securities designed to mirror the performance of a specific benchmark index (Charteris, 2013). While ETFs share similarities with mutual funds, a key difference is that ETFs can be traded throughout the day, whereas mutual funds are only traded once daily (Farinella and Kubicki, 2018). Additionally, mutual funds are bought and sold at their closing net asset value (NAV), while ETFs are traded on exchanges at market-determined prices (Venkataraman and Venkatesan, 2016). ETFs can issue additional units as demand rises, helping them trade at prices close to their NAV. However, their price efficiency

may be influenced by arbitrage limitations, such as transaction costs (Petajisto, 2017). Given the growing popularity of ETFs, closely monitoring this investment vehicle has become increasingly important. At the same time, selecting a profitable ETF for investment remains a key challenge (Manandhar, 2023).

Malamud's (2016) Dynamic Equilibrium Model of ETFs describes the ETF market structure as comprising two distinct tiers: the primary market, managed by authorized participants, and the secondary market, usually a centralized securities exchange. In the primary market, authorized participants create and redeem ETF shares, while in the secondary market, retail and institutional investors buy and sell these shares. Figure 2.2 provides a visual representation of the ETF trading process across both markets.

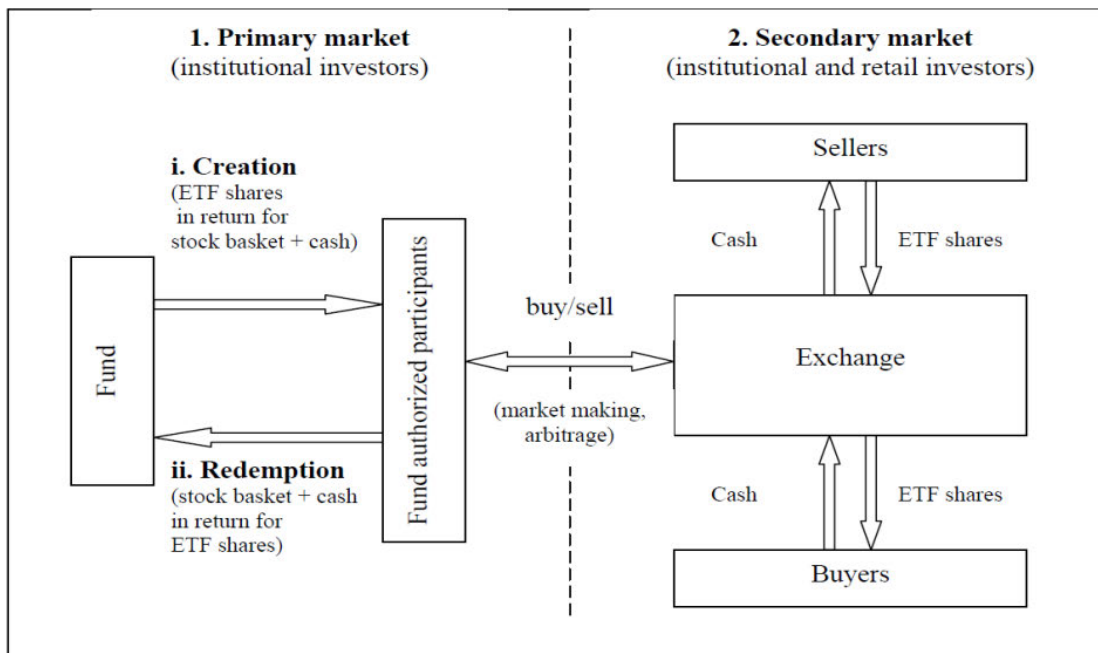


Figure 2.2 ETF Trading Process in Primary and Secondary Markets

Source: Adopted from Kunjal (2022)

In the primary market, ETF shares are created when authorized participants (usually large institutional investors) provide the fund with blocks of the underlying securities and cash (Lettau and Madhavan, 2018). The cash typically accounts for dividends, management fees, and rounding-off adjustments (Deville, 2008). The authorized participants receive equivalent ETF shares in exchange for the underlying securities and money. On the contrary, redemption occurs when authorized participants return

ETF shares to the fund and, in exchange, receive blocks of the underlying securities and cash (Staer, 2017). These creation and redemption transactions are called 'in-kind' transactions because they signify a barter of equivalent securities (Staer, 2017). Creation increases the number of ETF shares outstanding, while redemption decreases the number of ETF shares outstanding (Lettau and Madhavan, 2018). The outstanding ETF shares, which are tradable, are then bought or sold on securities exchanges by retail and institutional investors (Deville, 2008). It is noteworthy that authorized participants may also purchase or sell ETF shares in the secondary market.

The Dynamic Equilibrium Model of ETFs by Malamud (2016) suggests that the presence of two markets (primary and secondary markets) serves as an additional source of ETF liquidity. As a result, an improvement in the liquidity of the primary market contributes to additional liquidity in the secondary market (Malamud, 2016). Further, these creation and redemption processes allow authorized participants to absorb shocks in liquidity by creating or redeeming ETF shares as required (Fulkerson *et al.*, 2017). Similarly, any deviations between the ETF's NAV and its trade price may be arbitrated away by authorized participants. Specifically, if an ETF trades below its NAV, authorized participants purchase ETF shares from the secondary market (Ben-David *et al.*, 2018). These ETF shares are then redeemed for the underlying securities which may be sold in the market. The discount is then reduced as ETF prices face an upward pressure with a possible downward pressure on its NAV, and, subsequently, the price inefficiencies are reduced.

On the other hand, if an ETF is trading at a premium above its NAV, authorized participants purchase the underlying securities, which are then used to create new ETF shares (Fulkerson *et al.*, 2017). This increase in the supply of ETF shares puts a negative pressure on the ETF price and a possible positive pressure on its NAV, subsequently decreasing the premium and restoring price efficiency. As a result, price inefficiencies in the form of deviations of ETF prices from their NAVs are short-lived because of this creation and redemption process. However, the ability of arbitrageurs to reduce price deviations may be reduced by funding constraints, which increase when political, economic, and financial country risks increase, as well as informational asymmetries, which worsen when country risks rise (Gu *et al.*, 2018).

The unique creation and redemption process also allows ETFs to generate returns close to their benchmarks because changes in the underlying benchmarks can be easily replicated by authorized participants through the in-kind creation and redemption processes, subsequently improving the tracking ability of the ETF. On the contrary, in-kind baskets that do not perfectly replicate the benchmark or cause a delay in settlement of the in-kind transaction can hinder the tracking ability of the fund and lead to ETF tracking errors (Osterhoff and Kaserer, 2016). As a result, tracking errors may arise from increased country risk exposure, which makes it difficult for fund managers to replicate the underlying benchmarks perfectly, and it may cause delays in the settlement of in-kind transactions. Notably, ETFs can be physically or synthetically replicated. Physical ETFs physically hold the underlying securities, whereas synthetic ETFs replicate the benchmark using derivative contracts (Naumenko and Chystiakova, 2015). However, in South Africa, the Collective Investment Schemes Control Act 45 of 2002 prohibits synthetic ETFs because no leveraged or derivative products can be used in the collective investment scheme's portfolio (Peerbhai, 2020).

2.3.2 Benefits of ETFs

The arrival of ETFs has transformed the investment management landscape primarily because of their advantages over traditional funds. By purchasing a single ETF Portfolio, investors receive immediate exposure to the performance of a wide variety of the top performing securities within an index thereby avoiding the risk of putting all your eggs in one basket by having to buy each of the underlying constituents individually which requires complex and time-consuming activities like research and actively managing the underlying securities themselves (Wongsawatgul, 2019).

ETFs offer several benefits in portfolio optimization, making them a popular choice for investors seeking diversification, cost efficiency, and flexibility (Hill, Nadig and Hougan, 2015). One of the critical reasons for the soaring popularity of these funds is their ability to provide diversification benefits for investors. In particular, ETFs offer diversification benefits by investing in various asset classes, countries, industries, and styles (Arthur, 2013).

This is especially important for diversifying an investor's exposure to political, economic, and financial country risks. On the other hand, given that ETFs combine different asset classes, it is possible that country risk factors do not significantly influence pooled funds. Hence, these funds can reduce an investor's exposure to the effects of country risk factors. Country risks may, however, have a more significant effect on the pooled fund, and thus, ETFs may not assist in reducing country risk exposure. Ben-David *et al.* (2017) noted that ETFs' greater liquidity levels are another key advantage. ETFs can be bought and sold on an exchange throughout the trading day at market prices, providing flexibility and liquidity that mutual funds do not offer. Investors can use intraday price movements and execute strategies such as limit orders, stop-loss orders, and margin trading. Specifically, the intraday trading of ETFs provides investors with greater liquidity than mutual funds, which can only be traded at the end of the day at their closing NAVs (Ben-David *et al.*, 2017). However, this liquidity benefit of ETFs may be adversely impacted by political, economic, and financial country risks if risk-averse investors reduce their trading activities when political, economic, and financial country risks increase.

In addition to the advantages mentioned above, ETFs provide higher tax efficiency for investors. ETFs often use an in-kind creation and redemption process, which can help minimize capital gains distributions and improve tax efficiency for investors (Pakkala, 2022). The passive nature of many ETFs results in lower portfolio turnover, reducing the frequency of taxable events; unlike mutual funds, which have special tax rules, ETFs are regarded as tax efficient since their 'in-kind' redemption allows for a significant reduction or elimination of taxable capital gains when the ETF shares are exchanged for the underlying securities and not risk factors, that is, an increase in political, economic, and financial risks could increase the counterparty risk of ETFs. Furthermore, relative to ETFs tracking domestic benchmarks, the returns of ETFs tracking international benchmarks are restricted by exchange rate volatility, non-synchronous trading times, and the treatment of dividends, amongst other factors (Steyn, 2019).

Overall, the concerns associated with ETF markets suggest that ETFs can potentially disrupt broader financial markets. Therefore, although there are several benefits associated with investing in ETFs, some drawbacks to ETF investments need to be

considered before making investment decisions. These benefits or drawbacks of ETFs can be intensified by shocks in political, economic, and financial country risks (Bhattacharya and O'Hara, 2020). By incorporating ETFs into a portfolio optimisation strategy, investors can leverage these benefits to build a diversified, cost-effective, and flexible portfolio that aligns with their investment objectives and risk tolerance.

2.3.3 The ETF Portfolio (Fund of Funds)

A fund of funds (FOF) is one that, instead of investing in a pool of securities like stocks and bonds, buys shares in other funds (Larsen and Resnick, 2012). These multi-manager investments offer investors further diversification and access to the expertise of other skilled fund managers. The fund-of-funds approach has grown in recent years, particularly among institutional investors like pension funds and endowments seeking to spread risk and maximize returns. While open-end funds as a whole have grown 42% in assets worldwide from the first quarter of 2020 to the end of 2023, FOFs have grown 58% (Rochman and Eid, 2024).

FOFs come in various types, each designed for different investor needs, risk preferences, and investment objectives. These types can be broadly categorized based on the underlying assets they invest in, such as hedge funds, mutual funds, private equity, real estate, or commodities (Larsen and Resnick, 2012). In addition, some FOFs employ specific trading strategies or invest in particular sectors or geographic regions. An all-ETF portfolio is an investment portfolio made up entirely of exchange-traded funds, which are financial instruments that track indexes, sectors, commodities, bonds, or other asset classes (Mofokeng, 2018). Rochman and Eid (2024) stated that investing in a fund of funds is a strategy designed to enhance diversification and asset allocation, allowing investors to gain broader market exposure while mitigating risks compared to direct investments in individual securities.

This type of portfolio excludes direct investments in individual stocks, bonds, or other securities. In the real world, an all-ETF portfolio can be designed to achieve various investment goals, such as capital appreciation, income generation, or risk management. It typically includes ETFs representing different asset classes (e.g., equities, fixed income, commodities, and real estate) and may cover various geographical regions and sectors (Buetow and Hanke, 2024).

ETF Portfolios are well regulated by both the Johannesburg Stock Exchange (JSE) and Financial Services Board (FSB) ensuring that investors are protected against unjust treatment. Dividends from each ETF within the portfolio, are automatically re-invested by purchasing additional securities within each ETF boosting the overall performance of the ETF portfolio. Annual investment management and administration fees will also apply. Owning an ETF Portfolio does not give investors the right to vote at Annual General Meetings (AGMs) of the underlying ETFs, as you own a portion (unit) in the ETF and not the underlying securities themselves. Capital gains (profits) from the sale of an ETF Portfolios are subject to Capital Gains Tax (Mofokeng, 2018).

The buying price of each ETF in the portfolio will always differ slightly from its selling price, which reflects the net asset value (NAV) of the underlying fund. This variation arises due to ETF management fees and market dynamics, including supply and demand. Like their underlying securities, ETF prices fluctuate. However, the diversification advantage helps reduce the overall risk of financial loss (Rahym and Hawrami, 2020).

2.3.4 The Difference Between a Fund, and Fund of Funds

A fund is a type of investment vehicle that combines the capital of several investors to make direct investments in a variety of financial assets, including stocks, bonds, and real estate (Mohichexra and Leonidovna, 2023). Examples include mutual funds, ETFs, hedge funds, and private equity funds. In contrast, a FOF invests in other funds rather than individual securities, aiming to achieve broader diversification and leverage the expertise of multiple fund managers. While a fund typically focuses on a specific strategy, asset class, or market, a FOF employs a multi-fund approach, spreading risk across various strategies and asset classes (Larsen and Resnick, 2012). This structure provides an additional layer of diversification but also results in higher costs, as FOFs charge two layers of fees: one for managing the FOF itself and another for the underlying funds. Funds are generally managed by a single team specializing in the fund's focus area, while FOFs rely on managers who evaluate and select top-performing funds (Larsen and Resnick, 2012).

The risk and return profiles also differ, with funds offering returns based on their specific strategies and assets, while FOFs typically have lower risk due to their

diversification but may yield slightly reduced net returns after fees. Accessibility and liquidity vary; most funds like mutual funds and ETFs are liquid and accessible, whereas FOFs, especially those invested in private equity or hedge funds, may have liquidity constraints. Funds are ideal for investors seeking targeted exposure to specific markets, while FOFs suit those who desire a professionally managed, diversified portfolio without needing to research individual funds themselves. For example, an S&P 500 index fund focuses on U.S. equities, whereas a target-date retirement fund, as a FOF, invests in various funds to build a diversified retirement portfolio (Larsen and Resnick, 2012).

2.4 Summary

In conclusion, modern portfolio theory provides a robust framework for understanding the dynamics of portfolio construction and optimisation. By emphasising the importance of diversification and the trade-off between risk and return, MPT has become a cornerstone in portfolio management. As established by Markowitz in 1952, the core principle of MPT revolves around the notion that investors should not solely focus on maximising returns but rather consider the relationship between expected return, variance, and correlation of assets within a portfolio. This concept underscores the significance of constructing well-diversified portfolios that balance risk and return to achieve optimal outcomes. Moreover, the formulation of the mean-variance portfolio, as outlined in the theoretical framework, provides a mathematical basis for portfolio optimisation, considering the expected returns and variances of individual assets and their correlations. Thus, MPT is a valuable guide for investors in navigating financial markets and making informed investment decisions.

Moreover, the EMH provides key insights into how financial markets operate and their impact on portfolio management. EMH suggests that asset prices incorporate all available information, making it challenging for investors to consistently achieve above-market returns. While EMH presents obstacles for active management strategies, it also recognizes the potential of machine learning techniques to identify hidden patterns or relationships in data, which could create opportunities for portfolio optimization. While EMH suggests that markets are generally efficient, it does not preclude the possibility of market inefficiencies or anomalies, especially during distressed market conditions. Therefore, exploring the effectiveness of machine

learning approaches in portfolio management within the framework of EMH can offer valuable insights into the potential benefits of leveraging technology to enhance investment strategies.

Lastly, the Adaptive Market Hypothesis introduces a dynamic perspective on market behaviour, acknowledging the evolving nature of financial markets and the adaptive strategies of market participants. AMH suggests that traditional portfolio optimization techniques may not adequately capture the changing dynamics of markets, particularly during periods of distress or crisis. In contrast, machine learning techniques offer the potential to adapt to changing market conditions and capture nonlinear patterns and relationships in data. By systematically evaluating machine learning approaches in portfolio optimization under varying market conditions, researchers can assess whether these techniques better align with the principles of AMH and provide advantages in navigating the complexities of financial markets. Overall, the theoretical frameworks of MPT, EMH, and AMH offer valuable insights into portfolio management strategies and provide a foundation for further research. In summary, while MPT, EMH, and AMH provide valuable insights into financial markets, they each have strengths, limitations, and areas of overlap. Their interrelation and occasional conflict reflect the ongoing evolution of finance theory in response to new data, technological advancements, and academic inquiry.

CHAPTER 3: MACHINE LEARNING MODELS FOR PORTFOLIO OPTIMIZATION

3.1 Introduction

In the dynamic realm of investment management, optimal portfolio allocation is paramount (Samuels, 2024). As traditional methods encounter limitations in adapting to the complexities of modern markets, the integration of machine learning techniques emerges as a compelling avenue for enhancing portfolio optimization strategies (Sahu and Kumar, 2024). This chapter embarks on a comprehensive exploration of various machine-learning methodologies across diverse asset classes. This chapter attempts to illuminate their respective strengths, weaknesses, and potential for revolutionizing portfolio management practices by unpacking these techniques. Through rigorous analysis and empirical validation, this chapter aims to offer valuable insights into the evolving landscape of investment decision-making, paving the way for more robust and adaptive portfolio optimization frameworks.

In recent years, the rapid advancement of machine learning techniques has offered new opportunities for portfolio optimisation across different asset classes (Habbab and Kampouridis, 2024). Machine learning algorithms can handle large datasets, capture complex patterns, and adapt to changing market conditions, making them attractive tools for investment professionals seeking to enhance portfolio performance and risk management (Mandala *et al.*, 2023). Machine learning is situated within the broader framework of Artificial Intelligence (AI), as noted by Lund *et al.* (2023). As previously acknowledged, AI is a comprehensive discipline that aims to equip computers with cognitive abilities, utilising any necessary means, as Shtini (2023) discussed. It involves training a model using a dataset and utilising the trained model to make predictions or decisions without explicit instructions on executing the task (Simonis, 2022). Ayodele (2010) posited that the algorithm is presented with the desired input and output behaviour, thereby enabling it to gradually acquire knowledge of the mapping and correlation between the two entities, employing statistical models. The implementation of solutions does not necessitate direct programming, contingent on the type of machine learning algorithm employed; instead, the algorithm can be perpetually trained using fresh data.

3.1.1 The Role of Machine Learning in Finance

Machine learning has become increasingly important in finance due to its ability to process vast amounts of data, detect patterns, and provide actionable insights (Dixon, Halperin and Bilokon, 2020). These algorithms can analyse extensive financial data, including market prices, economic indicators, and corporate financials (Aziz *et al.*, 2022). By examining this data, machine learning models can uncover patterns, correlations, and trends that may not be immediately obvious to human analysts. These models improve our understanding of market dynamics, asset performance, and risk factors (Aziz *et al.*, 2022). Machine learning techniques are also valuable for predictive modelling in finance (Broby, 2022). By training on historical data, machine learning models can make accurate predictions about future market trends, asset prices, and financial indicators, assisting investors and financial institutions in making informed decisions, managing risks, and optimizing portfolio allocations (Singla and Jangir, 2020). Machine learning can significantly improve risk management in finance (Aziz and Dowling, 2019). By examining historical data and current market conditions, these models can estimate the likelihood of various risk events, including market crashes, credit defaults, or instances of fraud. They facilitate proactive risk mitigation strategies, early warning systems, and better decision-making for risk management and hedging (El Hajj and Hammoud, 2023). Additionally, machine learning algorithms can identify fraudulent activities and strengthen security in financial transactions. By analysing patterns and detecting anomalies in transactional data, these models can pinpoint suspicious behaviour, flag potential cases of fraud, and enhance security measures to protect against financial crimes (Priya and Saradha, 2021).

Hansen (2020) mentioned that machine learning is crucial in algorithmic trading and quantitative investment strategies. Machine learning models can analyse real-time market data, identify trading signals, and execute trades automatically. These models can adapt to changing market conditions and adjust trading strategies, accordingly, improving trading performance and reducing human biases. Additionally, Kumar *et al.* (2022) stated that machine learning techniques can be applied to customer data analysis in the finance industry. By analysing customer behaviour, preferences, and demographics, machine learning models can personalise financial services, offer tailored product recommendations, and improve customer satisfaction. Machine learning models lead to better customer engagement, retention, and loyalty. Frank and

Ray (2024) explained that machine learning algorithms provide advanced optimisation techniques for portfolio management and asset allocation. By considering multiple factors, such as risk tolerance, return objectives, and market conditions, machine learning models can optimise portfolio allocations, construct diversified portfolios, and minimise risk while maximising returns.

Machine learning plays a crucial role in the financial sector by enabling decisions based on data, enhancing risk management, refining trading strategies, and delivering key insights to investors, financial institutions, and clients. By leveraging machine learning, the finance industry can improve efficiency, precision, and profitability while adapting to the increasingly complex financial markets. One notable application of machine learning in finance is portfolio optimization. Through the use of advanced algorithms, investors are able to sift through vast amounts of data including historical records, fundamental metrics, and market news to detect patterns and make predictions (Kumbure *et al.*, 2022). This approach empowers investors to make more informed decisions and fine-tune their portfolios for optimal returns. Additionally, machine learning techniques assess various elements, such as economic indicators and market sentiment (Khan *et al.*, 2020), enabling a deeper understanding of market dynamics and facilitating informed investment choices.

3.1.2 Machine Learning in Portfolio Management

Machine learning algorithms enable asset managers to process large volumes of data quickly and effectively, helping them identify patterns, trends, and anomalies that could influence investment decisions (de Prado, 2020). These algorithms can autonomously analyse data, uncover valuable insights, and provide actionable recommendations for asset managers. Wamba-Taguimdje (2020) noted that the integration of machine learning into asset management will significantly transform the industry, improving operational efficiency and effectiveness. By harnessing artificial intelligence, asset managers can automate labour-intensive tasks, enhance decision-making, and achieve better investment outcomes for their clients. Additionally, Zhou *et al.* (2021) highlighted that machine learning opens up new opportunities for innovative business models and investment strategies. Asset managers can craft tailored investment solutions that meet their clients' specific financial goals and risk profiles, ultimately leading to improved performance and higher client satisfaction.

Lee and Shin (2020) emphasized that as machine learning continues to evolve, its potential to revolutionize asset management becomes more evident. With its capability to process large datasets, generate valuable insights, and provide accurate predictions, machine learning is transforming the way asset managers operate within the industry. According to Goodell (2021), machine learning has several uses in asset management, with forecasting asset values and market trends being a crucial one. Machine learning algorithms can provide prediction models that assist asset managers in making better investment decisions by analysing past market data and taking into account variables including macroeconomic indicators, social sentiment, and news sentiment (Zhou *et al.*, 2021).

In summary, the asset management sector has changed as a result of the use of machine learning techniques. Asset managers can improve their analytical skills, obtain insightful knowledge, and make more accurate investment decisions by utilizing artificial intelligence. Machine learning is used in many areas of asset management, including trade execution, risk management, portfolio rebalancing, investment research, and asset price prediction. It is anticipated that machine learning's involvement in portfolio management will improve as technology advances, opening up new avenues for the sector. By giving investors powerful tools to analyse financial data, make informed decisions, and optimize their portfolios for better results, machine learning has completely changed the field of portfolio optimization. Investors can stay ahead of the competition and more accurately negotiate the struggles of financial markets with the help of machine learning algorithms.

3.1.3 Challenges and Solutions in Machine Learning for Portfolio Optimisation

Although the benefits of machine learning in asset allocation and portfolio management are clear, as discussed, several key challenges must be addressed. First, advanced machine learning models like ANNs can outperform traditional human-based methods, but they can also make incorrect decisions by placing undue emphasis on misleading relationships between variables. For example, Avramov *et al.* (2019) found that ANNs, when trained to select stocks with high returns, sometimes end up choosing illiquid stocks. Second, these models contribute to the complexity of interpretation.

In simple terms, portfolio managers may struggle to explain the optimized algorithm equation, as it may not be grounded in traditional financial or economic theories (Avramov *et al.*, 2019). This difficulty in explanation can hinder the effective assessment of how machine learning models perform during highly unpredictable events, potentially leading to systemic risk, where multiple machine learning models make the same errors simultaneously (Avramov *et al.*, 2019). For instance, if several firms or investors are using contrarian strategies and incorrect model assumptions are made, the complexity of machine learning models could prevent users from identifying and mitigating potential systemic risks. Moreover, because developing new algorithms is expensive, many companies opt to purchase pre-built models, which many investment firms adopt at a lower cost (Sturgeon, 2021).

Bacon (2023) highlighted that attributing portfolio performance becomes increasingly difficult, especially during periods of poor performance. For instance, Barra trading models, which break down risk based on multiple factors to explain short-term stock return fluctuations, typically rely on linear relationships between returns and factors. In contrast, machine learning models do not impose such linear constraints, making it harder to explain to portfolio clients why the portfolio underperformed. Limited progress has been made in making machine learning models in finance easier to understand, interpret, and communicate to clients (Molnar, 2020). Furthermore, despite the greater volume and variety of available data compared to a decade ago, data quality remains essential. To avoid suboptimal returns caused by poor-quality data in machine learning models, it's crucial to use a sufficiently large dataset and avoid replacing missing values with statistical substitutes like averages. While portfolios' product disclosure statements (PDS) state that past performance does not guarantee future results, machine learning models can struggle to predict future asset prices or returns if they are not properly trained on data from past extreme events (Patel and Lincoln, 2019). This increases the risk of machine learning models failing during a crisis.

While machine learning holds significant promise for portfolio optimization, there are several challenges to be addressed. A key challenge is data quality, as financial data tends to be noisy and may include outliers, which can negatively impact the performance of machine learning algorithms (Gan, Wang and Yang, 2020). To mitigate

this issue, fund managers or data modelers must ensure that the data used for training machine learning models is both clean and trustworthy (Karimi-Mamaghan, 2022). Liu (2020) mentioned that obtaining high-quality financial data can be complex due to noise, missing values, and inconsistencies. Additionally, having enough historical data for training robust models can be challenging (Han *et al.*, 2021). Data preprocessing techniques such as cleaning, imputation, and normalisation can help address data quality issues. Moreover, alternative data sources and techniques like augmentation can help supplement limited historical data (Mumuni and Mumuni, 2022).

Santos and Papa (2022) explained that overfitting happens when a model learns to memorize the training data instead of generalizing patterns, which results in poor performance on new, unseen data. To prevent overfitting, regularization techniques like L1/L2 regularization, dropout, or early stopping can be applied (Rezaeezade and Batina, 2022). Cross-validation and out-of-sample testing can also assess a model's generalization performance. Financial markets are inherently dynamic, with market conditions evolving, which can render static models ineffective (Akhavanpour, 2023). Implement adaptive models that continuously learn and adapt to changing market conditions (Sharma and Mehta, 2024). Techniques like online learning, rolling window approaches, or incorporating time-series features can help capture market dynamics (Wang *et al.*, 2021).

Janabi (2021) mentioned that portfolio optimisation involves considering transaction costs, liquidity constraints, and other investment constraints, which can complicate the optimisation process. Incorporating transaction cost models into the optimisation process and considering constraints explicitly during portfolio construction may help (Beraldi *et al.*, 2021). Techniques such as mean-variance optimisation with transaction cost estimates or incorporating constraints through quadratic programming can address these issues (Sindi, 2023).

Abdollahi and Pradhan (2023) added that models may be sensitive to small changes in input data or parameters, leading to instability and unreliable predictions. Ensemble methods, model averaging, or robust optimisation techniques can enhance model stability and robustness (Arpit *et al.*, 2022). Sensitivity analysis and stress testing can also identify potential vulnerabilities and mitigate risks (Pang *et al.*, 2020). Therefore,

implementing machine learning for portfolio optimisation requires careful consideration of these challenges and appropriate solutions to address them. By leveraging advanced techniques and methodologies, financial institutions can harness the power of machine learning to build more effective and robust portfolio management strategies. Furthermore, it is vital to consider the source of the financial data, as different sources may have varying levels of accuracy and reliability.

3.2 A Critical Analysis of ML Models for Portfolio Optimization

This section presents a background and explanation of machine learning techniques used in portfolio optimisation. MLs are categorised into three groups, namely, supervised learning, unsupervised learning, and reinforcement learning, and there are sub-elements under each category. In addition, mathematical representations for each group and the basic parameterisation for each sub-element of the three categories, along with their loss or objective functions, are unpacked and contextualised. The sub-categories of the three machine learning classes are critically analysed in the sections that follow.

3.2.1 Supervised Learning

Supervised learning is a machine learning approach where the model learns from labelled data to make predictions. In supervised learning, the dataset contains input-output pairs where the model is trained to find a mapping from inputs to the outputs. This approach encompasses two main types: regression and classification (Teles *et al.*, 2021). In portfolio optimization, supervised learning techniques in regression and classification are valuable for portfolio construction, risk assessment, and return prediction. Sub-elements of these models are also unpacked in the subsections that follow.

3.2.1.1 Regression Models

Regression aims to predict a continuous output variable based on input features. In other words, regression models are used when the target variable is a real-valued number, such as forecasting stock prices, temperatures, or sales volumes (Chatzidis, 2019). The model learns the relationship between the input variables and a numerical outcome, identifying patterns that allow it to predict values in a continuous range. In regression, the objective is to minimize the difference between expected and actual

values. Regression is typically used in supervised learning when the goal is to predict a continuous outcome, such as future returns, risk levels, or other financial metrics that aid in portfolio optimization (Terrell, 2019).

Linear Regression

Linear regression is a statistical technique used to model the relationship between a dependent variable and one or more independent variables by fitting a linear equation to the observed data. In simple terms, it helps us understand how the value of one variable changes in response to changes in another variable. (Behera *et al.*, 2023). The basic idea behind linear regression is to find the best-fitting line (or plane in higher dimensions) through the data points. After obtaining the coefficients from the regression analysis, one can use this information to construct an optimal portfolio (Ma, Han and Wang, 2020). This might involve selecting assets with coefficients that suggest they positively impact on the overall portfolio return and weighing them accordingly. Conversely, one might avoid assets with negative coefficients or low correlation with the rest of the portfolio.

Efron *et al.* (2004) mentioned that at first glance, the application of linear regression concepts may seem unsuitable for addressing the challenges of modern portfolio theory. However, the risk minimization problem can be effectively framed as a regression problem, where the estimated coefficients represent the optimal portfolio weights. Suppose one has an opportunity to invest in n assets with returns $R_1, \dots, R_n$. Letting R be the row vector of asset returns, Σ be their covariance matrix of asset returns and w the portfolio allocation vector satisfying the budget constraint $W_1 = 1$. As shown by Fan, Zhang and Yu (2012), the risk minimization objective function is given by.

$$\min Var(R_p) = \min Var(wR) = \min_w w \Sigma w \quad (3.1)$$

$Var(R_p)$ represents the variance of the portfolio return.

s.t. $w_1 = 1$, can be assimilated to regression problem that implicitly includes the budget constraint.

Variance measures the dispersion of returns around the mean return, indicating the level of risk or volatility associated with the portfolio (Jiang, 2010). As a result, the portfolio optimization problem can be redefined using a regression model. Mean-variance portfolios, which are constructed based on the sample mean and covariance matrix of asset returns, can be viewed as a function of the ordinary least squares estimator for the linear regression coefficients (Dadashi *et al.*, 2022).

Ridge Regression

Tikhonov introduced Ridge regression in the 1940s/50s, also known as Tikhonov regularization, as a more stable alternative to the standard least squares estimator. It is widely used in literature as a remedy for the multicollinearity problem among regression predictor variables. Hoerl and Kennard (1988) helped popularize the Tikhonov regularization model. Ridge regression, or L_2 regularization, adds the sum of the squared coefficients (excluding the intercept) to the loss function. In the presence of multicollinearity, even though least squares estimates remain unbiased, their variances become large, causing the estimated parameters to deviate significantly from the true values.

When the regression variables in a model are highly correlated, the regression coefficient of any given variable becomes highly dependent on which other predictor variables are included or excluded from the model. This means that the prediction variables often do not capture the inherent effect of a specific predictor on the response variable, but rather only its partial or marginal impact. For instance, a highly significant positive coefficient for one variable could be offset by a similarly large negative coefficient for its correlated counterpart (Trevor and Robert, 2009). As the correlation between predictors increases, creating a model with high variance becomes increasingly unrealistic. Ridge regression addresses this by producing models with lower variance and shrinking the regression coefficients by adding a positive penalty proportional to the squared values of the standard linear regression problem. The Ridge coefficients aim to minimize the penalized residual sum of squares (Prasad, 2024).

$$\hat{w}_{Ridge} = \underset{w}{\operatorname{argmin}} (\sum_{i=1}^N (y_i - w_0 - \sum_{j=1}^p w_j x_{ij})^2 + \lambda \sum_{j=1}^p w_j^2) \quad (3.2)$$

$\lambda > 0$ represents the complexity parameter that controls the amount of shrinkage: the larger the value of λ , the greater is the amount of shrinkage imposed on the regression parameters. Tian and Zhang (2022) explained that the regularization parameter λ controls the trade-off between fitting the data and penalizing large coefficients. Cross-validation techniques can be used to select an appropriate value for λ . While in the latter the Ridge regression is expressed in its Lagrangian form, with λ as the tuning parameter, another equivalent and interesting way of formulating the problem, is by explicating the size constraint:

$$\begin{aligned} \hat{w}_{Ridge} = \underset{w}{\operatorname{argmin}} & \left(\sum_{i=1}^N (y_i - w_0 - \sum_{j=1}^p w_j x_{ij})^2 \right) \\ \text{s. t. } & \|w\|_2^2 \leq t^2 \end{aligned} \quad (3.3)$$

where $\|w\|_2 = (\sum_{j=1}^p w_j^2)^{\frac{1}{2}}$ is the Euclidean norm of the vector of weights.

Therefore, ridge regression is extensively used in financial modelling to predict stock prices, portfolio returns, and credit risk assessment. It helps reduce the risk of overfitting, which is crucial when dealing with economic data. Ridge regression helps address multicollinearity issues and overfitting in portfolio optimization, leading to more stable and robust portfolios (Islam and Pandow, 2024). Incorporating a regularization term ensures that the estimated asset weights are more reliable and less sensitive to the noise in the data. In the context of portfolio optimization, ridge regression can be particularly useful for improving the stability and performance of the optimization process, especially when dealing with large numbers of assets and complex models (Islam and Pandow, 2024).

Lasso Regression

While Ridge regression offers notable improvements over traditional least squares regression, it does not resolve the issue of variable selection, which becomes particularly challenging in the Markowitz optimization problem when dealing with a large number of assets. As a result, various studies (Brodie *et al.*, 2009; DeMiguel *et al.*, 2009b; Yen, 2010; Yen and Yen, 2011; Carrasco and Noumon, 2011; Fan *et al.*, 2012; Fernandes *et al.*, 2012) have focused on another penalization approach in portfolio optimization: the Least Absolute Shrinkage and Selection Operator (LASSO). LASSO regression, or L_1 regularization, adds the sum of the absolute values of the coefficients (excluding the intercept) to the loss function. This method enhances numerical stability and predictive performance. It addresses two key issues with least squares estimates: model parsimony and the interpretability of coefficient values. By selecting a more parsimonious model, LASSO helps manage risk by eliminating assets with redundant or noisy information. This selection process results in a more stable and reliable portfolio, reducing the risk of overfitting past data, which may not accurately predict future performance (Yen and Yen, 2011).

Traditional portfolio optimization models, such as the Markowitz mean-variance optimization, often assume that the returns and risks of assets are known and stable over time. By integrating Lasso Regression into the optimization process, one can improve the estimation of expected returns and covariances. This integration results in more robust portfolio optimization models that better capture the underlying dynamics of asset returns. The latter preserves many of the beneficial properties of Ridge regularization, but it generates sparse models that are easier to interpret and often outperform those produced with an L_2 penalty (Trevor *et al.*, 2009). The LASSO's objective function can be simply defined by

$$\hat{w}^{LASSO} = \min_w \sum_{i=1}^N (y_i - w_0 - \sum_{j=1}^p w_j x_{ij})^2 \quad (3.4)$$

$$s. t. = \sum_{j=1}^p |w_j| \leq t.$$

Therefore, Lasso regression minimizes the mean squared error (or another loss function) while imposing a constraint on the sum of the absolute values of the portfolio weights (the L_1 norm) (Sadik, Et-Tolba and Nsiri, 2024). This constraint helps in

managing overfitting and enhancing model interpretability and finally evaluate the performance of the portfolio on the test set and compare with other portfolio optimization methods (e.g., mean-variance optimization).

Ridge uses the L_2 norm (squared coefficients): $\lambda \sum_{j=1}^p w_j^2$ while Lasso uses the L_1 norm (absolute coefficients): $\lambda \sum_{j=1}^p |w_j|$. Ridge regression shrinks the coefficients towards zero but does not set any of them exactly to zero. This means all features are kept in the model, though they may have very small coefficients. Lasso regression can shrink some coefficients to exactly zero, effectively performing feature selection by excluding some features from the model (Sadik, Et-Tolba and Nsiri, 2024).

As done for the Ridge regression, it is possible to re-parameterise the constant w_0 by standardising the predictors. In this way, again, the solution for $\hat{w}_0$ is $\bar{y}$, and thereafter it is possible to fit a model without an intercept and rewrite the LASSO problem in the equivalent Lagrangian form.

$$\hat{w}^{LASSO} = \min_w \sum_{i=1}^N (y_i - \sum_{j=1}^p w_j x_{ij})^2 + \lambda \sum_{j=1}^p |w_j|. \quad (3.5)$$

This time the L_2 penalty $\lambda \sum_{j=1}^p w_j^2$, is replaced by the L_1 penalty $\lambda \sum_{j=1}^p |w_j|$.

One can gather historical data on asset prices, returns, and potential predictors in portfolio optimisation. Then, Lasso Regression will be applied to predict asset returns using the gathered predictors, identifying the most relevant factors (Li, Liang and Ma, 2022) and selecting the assets with non-zero coefficients from the Lasso model. One has to use the selected assets and their predicted returns in a portfolio optimisation model, applying mean-variance optimisation or other techniques. Lastly, the portfolio must be evaluated and adjusted to manage risks, using the insights from the Lasso model to avoid overfitting (Li, Liang and Ma, 2022). Therefore, using Lasso Regression in portfolio optimisation can enhance the model's predictive power, improve risk management, and create more efficient and manageable portfolios.

Elastic Net Regression

Based on the previous discussion, LASSO regression appears to be a superior choice compared to Ridge for variable selection. However, it has notable limitations in certain situations. First, when the number of variables exceeds the number of observations ($p > n$), LASSO can select at most n variables before reaching its limit. Second, if there is a group of variables with high pairwise correlations, LASSO tends to randomly choose only one variable from the group, disregarding the others (Zou and Hastie, 2005). Third, in typical cases where $n > p$, if there are strong correlations between predictors, the prediction performance of LASSO is often outperformed by Ridge regression (Tibshirani, 1996). To address these limitations, Zou and Hastie (2005) proposed a solution that has gained considerable attention: a new regularization method that simultaneously performs automatic variable selection and continuous shrinkage while accounting for groups of correlated variables. This method, called the "Elastic Net," includes grouped selection in its objective function.

$$\begin{aligned} \hat{w} = \underset{w}{\operatorname{argmin}} \quad & \sum_{i=1}^N (y_i - \sum_{j=1}^p x_{ij} w_j)^2 \\ \text{s. t.} \quad & \sum_{j=1}^p (\alpha |w_j| + (1-\alpha) w_j^2) \leq t. \end{aligned} \tag{3.6}$$

As mentioned earlier, the penalization term is a convex combination of the LASSO and Ridge penalties. The L_1 component of the penalty encourages sparsity in the coefficients, while the quadratic part encourages the averaging of highly correlated features, thereby stabilizing the L_1 regularization path (Zou and Hastie, 2005). In portfolio optimization, the objective is to build a portfolio that either maximizes return for a specified level of risk or minimizes risk for a given level of return. Traditional approaches like Mean-Variance Optimization can be sensitive to estimation errors in expected returns and the covariance matrix. Elastic Net Regression can be employed in this context to address some of these challenges. Therefore, Elastic Net Regression provides a robust and flexible approach to address some of the key challenges in portfolio optimization, such as multicollinearity and the high-dimensional nature of financial data (Sun, 2021). By leveraging its ability to produce sparse and stable

solutions, it enhances the traditional methods of portfolio construction, leading to potentially better performance and risk management.

3.2.1.2 Classification Models

Suthaharan (2016) stated that classification models deal with predicting a discrete label or category. The model learns to classify input data into distinct classes or groups based on the patterns in labelled data. For example, classification models can distinguish between spam and non-spam emails or identify fruit types based on attributes like colour and shape. In classification, the objective is to correctly categorise each instance into one of the predefined classes. Classification is employed when the aim is to categorise assets or portfolio states rather than predict a continuous variable. In portfolio optimisation, classification often aids in risk management, asset selection, and dynamic allocation.

Support Vector Machines (SVM)

Sabzekar and Hasheminejad (2021) noted that support vector machines (SVM) can be applied to both classification and regression tasks, through support vector classification (SVC) and support vector regression (SVR), respectively. However, SVM is best suited for smaller datasets due to its long processing times. This section focuses on SVC, particularly its application to portfolio optimization. SVM utilizes the kernel trick to map the data into a higher-dimensional space, where the machine learning task is then framed as a convex optimization problem. Unlike other machine learning methods, SVM finds optima analytically rather than heuristically (Shafaghat and Dezvareh, 2021). In real-world scenarios, data is often not linearly separable in its original space. This means instances with different labels overlap in the input space, preventing a linear hyperplane from separating the classes accurately. Attempting to find a nonlinear boundary in the input space increases computational demands during optimization, as the separating surface must be at least second order (López and Crossa, 2022).

Instead, SVM uses predefined kernel functions, such as linear, polynomial, or radial basis functions (RBF), to map the data, where a linear separator can distinguish between different classes (Chhajer, Shah and Kshirsagar, 2022). The SVM optimization process focuses on learning a linear discriminant surface in the mapped

space. The choice and configuration of the kernel function are essential for achieving optimal SVM performance. SVM is a supervised algorithm designed to divide a high-dimensional space into regions, each corresponding to a class, by fitting the best hyperplane separator $x^T w + b = 0$ (Cortes and Vapnik, 1995). The quality of the separator relies on maximising the margin width between the closest instances of both classes. SVM can be seen as an optimization problem combining margin maximization and penalty minimization, both objectives being regularized by hyparameter C. Its Lagrangian dual is of the following form.

$$\max_{\alpha} J(\alpha) = \sum_{k=1}^N \alpha_k - \frac{1}{2} \sum_{i=1}^N \sum_{k=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (3.7)$$

Subject to $\sum_{k=1}^N \alpha_k y_k = 0, 0 \leq \alpha_k \leq C, k \in (1; N)$

Where K is a Kernel whose function satisfies Mercer's theorem, including among others linear, polynomial or Radial Basis Function (RBF). This is shown in the figure 3.1 below. A hyperplane is a decision boundary that separates different classes of data points. In the context of SVM, the hyperplane is chosen to maximize the margin between the classes, which is the distance between the hyperplane and the nearest data points from each class and these nearest points are called support vectors (Basak *et al.*, 2021).

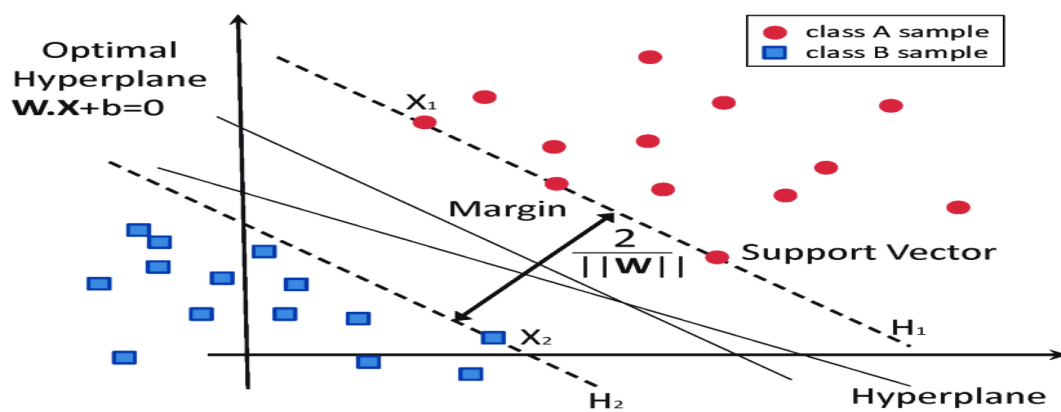


Figure 3.1 The Hyperplane in Action in SVM

Source: Basak *et al.* (2021)

In portfolio optimization, assets can be classified based on their characteristics, such as return and risk profiles. Pan *et al.* (2021) added that SVM can classify assets into different categories (e.g., high-risk, low risk) by finding the optimal hyperplane that separates these classes. The hyperplane can represent the optimal trade-off between risk and return. By classifying assets into those that meet the desired risk-return criteria and those that do not, SVM can help construct a portfolio that aligns with an investor's risk tolerance and return expectations (Sami, 2023). By classifying historical market data, SVM can be used in different market regimes (e.g., bull or bear markets). The hyperplane would separate data points corresponding to other regimes, aiding in adjusting the portfolio strategy according to the prevailing market conditions.

Additionally, SVMs offer different kernel functions to transform input data into higher-dimensional spaces, allowing for a more effective separation of classes (Sami, 2023). Three standard kernels used in SVMs are linear, polynomial and radial basis functions, and they are explained below, along with their mathematical formulations and their relevance to portfolio optimization:

$$\text{Linear Kernel} \quad K(x_i, x_j) = x_i^T x_j \quad (3.8)$$

The linear kernel computes the dot product between input feature vectors in the original space. It is suitable when the data is linearly separable or when computational efficiency is crucial (Pan *et al.*, 2021). The linear kernel can be adequate in portfolio optimisation when the relationships between assets are approximately linear. It assumes that the portfolio weights are directly optimised based on linear combinations of asset returns.

$$\text{Polynomial Kernel} \quad K(x_i, x_j) = (x_i^T x_j + r)^d \quad (3.9)$$

The polynomial kernel computes the dot product raised to a power, optionally adding a constant term r . It captures non-linear relationships between data points (Pan *et al.*, 2021). Portfolio optimization often encounters non-linear relationships among asset returns. The polynomial kernel can help capture these non-linearities, potentially leading to better separation of different market regimes or asset classes.

$$\text{Radial Basis Function (RBF) Kernel} \quad K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (3.10)$$

The RBF kernel maps data into an infinite-dimensional space by measuring the similarity (exponentially decaying with distance) between input vectors. RBF kernels are versatile for handling complex, non-linear relationships in data. In portfolio optimization, they can capture intricate dependencies between asset returns, potentially improving the robustness of the optimized portfolio against market fluctuations.

Therefore, the choice of a kernel in SVM impacts how well the model can separate different classes or patterns in the data. In portfolio optimization, the kernel choice should reflect the complexity and nature of relationships between asset returns (Mandal and Thakur, 2024). More complex kernels (like polynomial or RBF) can lead to overfitting if not correctly tuned. In contrast, simpler kernels (like linear) may be unfit if they have non-linear relationships (Belanche-Muñoz and Wiejacha, 2023). SVMs with appropriate kernels can be used in portfolio optimization to improve the accuracy of risk estimation, enhance diversification strategies, and effectively manage non-linear dependencies between asset returns. Therefore, selecting SVM kernels in portfolio optimization depends on the underlying structure and relationships within the asset returns data. Linear, polynomial, and RBF kernels offer varying degrees of complexity to handle different types of data patterns and relationships, thereby impacting the performance and robustness of the portfolio optimization process.

Random Forest Regressions

Random forest is an ensemble of decision trees and one of the many machine learning algorithms (Genuer *et al.*, 2020) and was proposed by Breiman (2001). Due to its simple and easy-to-understand design logic and non-overfitting computational design (Breiman, 2001), it has been popular among various emerging algorithms with its excellent optimization capabilities, and it has been widely used and explored in recent studies (Kaczmarek and Perez, 2022; Pinelis and Ruppert, 2022; Ma, Han and Wang, 2021). The random forest is the average,

$$R(x) = \frac{1}{T} \sum D_t(x) \quad (3.11)$$

$D_t(x)$ is the predicted outcome from that single tree t and T is the number of trees in a random forest. A decision tree iteratively partitions the space of x , branching on conditions relating to those features. If $a_{it}x \geq b_{it}^t$, then we move to the right child node of i in the tree, otherwise we move the left child node i in the tree. In most commonly used implementations of random forest, splits are axis aligned such that only one dimension of x is involved in the split, $x_j \geq b_i^t$. After a number of splits, we arrive at a leaf, which is a set of x which share the same score. Thus, the problem is to maximise.

$$\max_{x,z,q^t} R(x) = \frac{1}{T} \sum_{t=1}^T D^t(x) \quad (3.12)$$

subject to $B(w)z \leq c(w)$

$(z, w) = x$

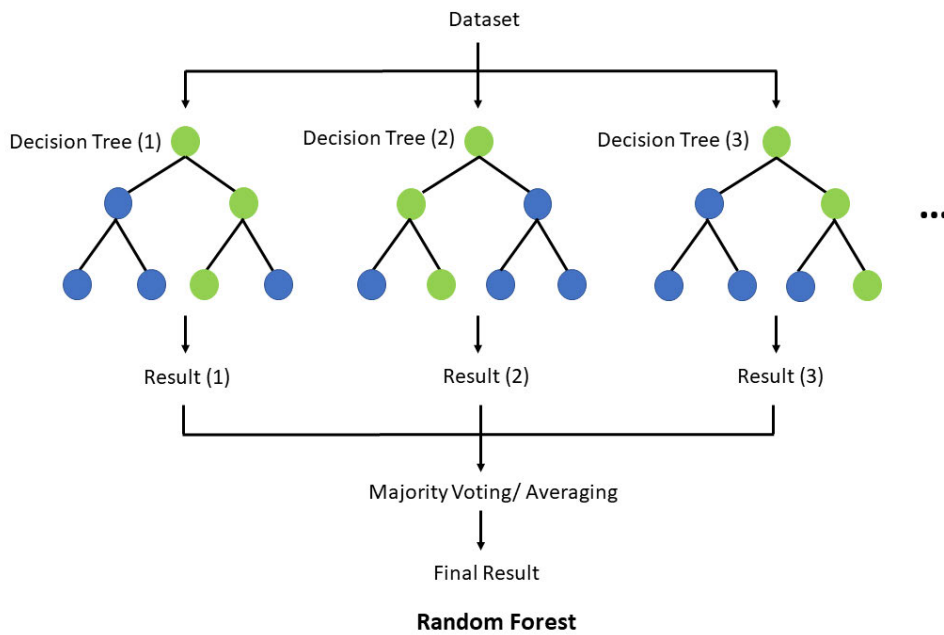


Figure 3.2 Visualisation of Random Forest Trees

Source: Breitung (2023)

They can enhance decision-making processes by leveraging their predictive power and robustness in portfolio optimisation. Random forests combine the predictions of multiple decision trees to improve accuracy and reduce overfitting (Pinelis and Ruppert, 2022). Each tree in the forest is trained on a random subset of the data, and predictions are aggregated (usually by averaging in regression or majority voting in classification). Each split in a tree considers a random subset of features, adding a layer of randomness to create diverse trees (Pinelis and Ruppert, 2022). By ranking assets based on predicted returns and risks, random forests can assist in selecting a subset of assets for the portfolio, potentially improving the overall performance compared to traditional methods. Random Forests can simulate various market conditions and assess how the portfolio would perform under different scenarios, helping in stress testing and robust optimisation.

Mazraeh *et al.* (2022) explained the advantages of using random forests in portfolio optimisation: accuracy, robustness and flexibility. Random forests can model complex relationships and interactions between predictors and target variables. The ensemble nature makes them less prone to overfitting than individual decision trees. They can handle a mix of continuous and categorical predictors and are relatively insensitive to the scaling of input features. However, it has challenges of Interpretability and computational cost, where random forest models are less interpretable than simpler models like linear regression. Training and predicting with random forests can be computationally intensive, especially with large datasets. Therefore, by integrating random forests into the portfolio optimisation process, investors can potentially improve their decision-making through more accurate predictions and robust analysis of asset behaviours.

eXtreme Gradient Boosting (XGBoost)

XGBoost is a set of weak classifier decision trees that focuses on training new trees to correct the errors made by previous ones (Dey *et al.*, 2016). This process follows a gradient descent-like approach. The trees are trained in a sequential manner (Chen and Guestrin, 2016). Initially, a regression function is created to fit the data, which generates errors known as residuals. These residuals are used to create another regression function, which is combined with the previous one to minimize errors. This iterative process leads to increasingly complex regression functions, ultimately

resulting in a significant reduction in root mean squared error (Dezhkam and Manzuri, 2023).

In portfolio optimization, the objective function for eXtreme Gradient Boosting (XGBoost) typically involves minimizing a loss function that balances the trade-off between risk and return. Below is a detailed explanation by Dezhkam and Manzuri (2023) of how one may construct such an objective function with three components: expected return, risk (variance or standard deviation) and regularization terms (terms that penalize complexity to avoid overfitting, standard in machine learning models like XGBoost). The objective function can be written as:

$$Objective = Loss + \Omega(Model) \quad (3.13)$$

Where:

Objective is the overall objective function.

Loss is the loss function representing the difference between predicted and actual returns.

Ω is the regularization term to control the model complexity.

The loss function can be constructed using the mean-variance optimization framework:

$$Loss = -\lambda(w^T r) + (1 - \lambda)w^T \Sigma w \quad (3.14)$$

Where:

w is the weight vector of the assets in the portfolio.

r is the vector of expected returns for each asset.

Σ is the covariance matrix of asset returns.

λ is the risk aversion coefficient (a parameter to balance risk and return).

The regularization term in XGBoost typically includes:

$$\Omega = \gamma T + \frac{1}{2} \alpha \sum_{j=1}^T w_j^2 \quad (3.15)$$

T is the number of terminal nodes (leaves) in the tree.

γ is the parameter to control the number of leaves.

α is the parameter to control the L2 regularization term on leaf weights.

Combining these components, the XGBoost objective function for portfolio optimization can be written as:

$$L = -\lambda(w^T r) + (1 - \lambda)w^T \sum w + \gamma T + \frac{1}{2} \alpha \sum_{j=1}^T w_j^2 \quad (3.16)$$

Mercangöz (2021) noted that portfolio optimisation aims to allocate assets to maximise return and minimise risk. Applying XGBoost in this context would involve predicting future asset returns and using these predictions to inform optimisation. The components of the XGBoost objective function can be contextualised as follows: if the goal is to predict future returns, the loss function can measure the difference between expected and actual returns (Morstad and Bock, 2021). Custom loss functions are tailored to financial metrics (e.g., Sharpe ratio, Sortino ratio) and can be implemented depending on the specific goals. Penalises overly complex models that might overfit the training data. In portfolio optimisation, this helps avoid excessively optimistic predictions that don't generalise well to new data. Therefore, XGBoost can be a valuable tool in portfolio optimisation by providing accurate forecasts of asset returns, which can then be used to optimise portfolio allocation. The objective function in XGBoost, comprising a loss function and a regularisation term, ensures that the model is accurate and generalisable, which is crucial for making reliable financial decisions (Morstad and Bock, 2021).

Light Gradient Boost Model (LightGBM)

LightGBM is a boosting ensemble learning model introduced by Microsoft in 2017. It is essentially a gradient boosting decision tree (GBDT) algorithm, but with two innovative techniques: gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB), which enhance the standard GBDT approach (Ke *et al.*, 2017).

Additionally, unlike traditional GBDT-based algorithms like XGBoost (Daoud, 2019), LightGBM grows its trees vertically, while other algorithms grow them horizontally (Ke *et al.*, 2017). This distinction makes LightGBM particularly effective and efficient for handling large-scale datasets. Given the supervised training sample set $X = \{x_i y_i\}_i^n = 1$, the goal of LightGBM is to find approximation of $f(x)$ to a certain optimal function that minimizes the expected value of the loss function $L(y, f(x))$ as follows:

$\hat{f} = \underset{f}{\operatorname{argmin}} E[L(y, f(x))]$ where the boosting strategy is employed in LightGBM, that is a number of T regression trees $f_i(X)$ are built to approximate the final model in an additive form as follows.

$$f_T(X) = \sum_{i=1}^T f_i(X) \quad (3.17)$$

Accordingly, at step t, the objective function of LightGBM can be expressed as follows.

$$T_t = \sum_{i=1}^T L(y_i, f_{t-1}(x_i) + f_t(x_i)) \quad (3.18)$$

Masuda (2024) mentioned that LightGBM can optimise various objective functions, such as maximising returns, minimising risk, or achieving a balance between risk and return (e.g., Sharpe ratio optimisation). The algorithm can incorporate constraints, such as budget, sector, and regulatory requirements, making it suitable for real-world portfolio optimisation problems. LightGBM is designed to handle large datasets efficiently, making it ideal for portfolio optimisation problems involving many assets and historical data points. Once trained, LightGBM models can quickly generate predictions, enabling rapid rebalancing and adjustment of portfolios in response to market changes (Uykun, 2024). Chen *et al.* (2020) added that LightGBM can be integrated with other optimisation techniques, such as evolutionary algorithms or traditional mean-variance optimisation, to enhance performance. For example, LightGBM can predict returns, which are then fed into an evolutionary algorithm for optimisation. LightGBM can be used as part of an ensemble of models, combining predictions from multiple algorithms to improve robustness and accuracy.

Chou and Chen (2024) pointed out the benefits of using LightGBM in portfolio optimisation, which include accuracy, efficiency and flexibility. LightGBM's ability to capture complex relationships in the data can lead to more accurate asset returns and risk predictions. Its efficiency in handling large datasets and fast inference speeds make it suitable for real-time portfolio management and rebalancing. LightGBM's capability to handle various features and integrate with other optimisation techniques provides a flexible framework for portfolio optimisation. Therefore, LightGBM can significantly enhance the process of portfolio optimisation by giving accurate predictions, efficient training and inference, and the ability to handle complex, real-world constraints and objectives.

CatBoost

CatBoost is a novel gradient-boosting decision tree (GBDT) algorithm that excels at handling categorical features (Alshari, Saleh and Odabaş, 2021). Unlike traditional GBDT algorithms, it differs in the following ways: 1) It processes categorical features during the training phase rather than during preprocessing. This allows CatBoost to utilize the entire dataset for training. Prokhorenkova *et al.* (2018) highlight that target statistics (TS) are an effective method for managing categorical features with minimal information loss. Specifically, CatBoost performs a random permutation of the dataset for each example and calculates an average label value for instances with the same category value.

In the training of decision trees, each tree is optimised using gradient boosting technique, *i.e.*, a gradient descent operation is performed on the objective function to minimise the loss function between the predicted and true values. The objective function used by Catboost is a weighted log-likelihood loss function, which is defined as.

$$L(y, f(x, \theta)) = -\frac{1}{N} \left[\sum_{i=1}^N (w_i y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \right] \quad (3.19)$$

Where w_i is the weight of sample, y_i is the true label, $\hat{y}_i$ is the predicated value of the model. During the training process, Catboost also uses an adaptive learning rate technique that automatically adjusts the size of the learning rate based on the

information of the gradient, thus accelerating the training of the model (Ibrahim *et al.*, 2020)

Toochaei and Moeini (2023) explained that the predictions from CatBoost models are incorporated into an objective function that maximizes return and minimizes risk. The optimization can include constraints like budget, maximum or minimum investment in certain assets, sector diversification, etc. Predictions from CatBoost can be fed into a GA to optimize the portfolio. The GA can explore different combinations of asset weights to find an optimal portfolio. Therefore, by leveraging CatBoost in portfolio optimization, investors can harness the power of machine learning to make more informed and data-driven investment decisions.

Neural Networks (NNs)

A neural network is a type of machine learning that utilizes a layered structure to represent data (Dastres and Soori, 2021). Its design is inspired by the human brain. The model receives data, referred to as the input, which forms the first layer of the network. This input is then processed as it moves through the subsequent layers, called hidden layers. The network may have one or multiple hidden layers. These hidden layers transform the input into an output, which is the final layer and represents the model's target (Li *et al.*, 2021). Neural networks can be used in portfolio optimization to improve the process of selecting a mix of assets that maximizes return while minimizing risk. The core concept of applying neural networks for this task involves solving the optimization problem using the backpropagation algorithm. To achieve this, it is necessary to carefully design the architecture of the neural network.

The backpropagation algorithm primarily relies on stochastic gradient descent, which computes the gradients of the loss function with respect to the weights and biases. These weights and biases are subsequently updated using the gradient information (Zhang *et al.*, 2020). The neural network is trained using a single data point, which often serves as the starting point for the stochastic gradient descent algorithm that optimizes the network. This data point also functions as the input for the first layer of the neural network. Notably, there is no specific training data for the output layer, as it is not trained in a supervised manner. The convergence of stochastic gradient descent has been explored through the lens of convex minimization and stochastic

approximation theories. In summary, stochastic gradient descent is highly likely to converge to a global minimum for convex or quasi-convex objective functions when the learning rate is reduced appropriately and under moderate assumptions. However, when the objective function is non-convex, it will likely converge to a local minimum (Bottou, 1998; Kiwiel, 2001), in accordance with the Robbins–Zigmond theorem (Robbins and Siegmund, 1971).

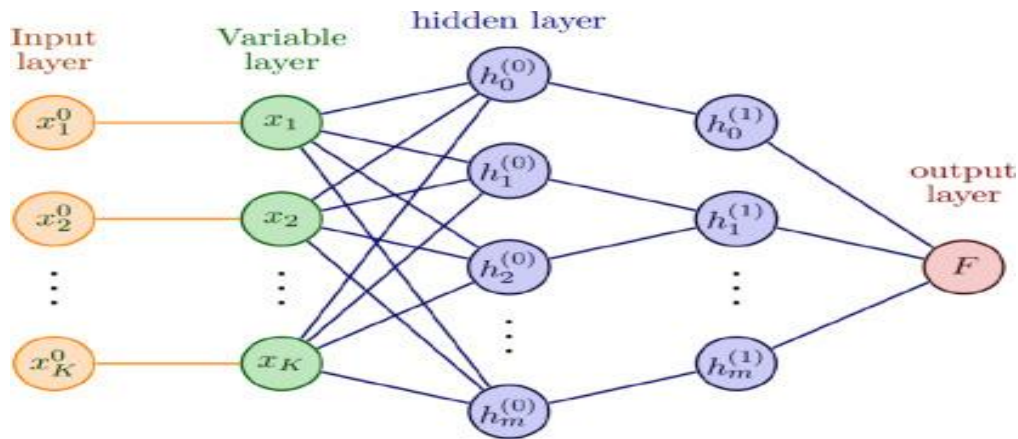


Figure 3.3 The Hidden Layers Transformation from Input to Output

Source: Saikia et al. (2020)

In the context of neural network-based portfolio optimization, the objective function is designed to maximize returns and minimize risk, typically measured by metrics such as Sharpe ratio, variance, or other risk-adjusted return measures. The following is a general form of the objective function for portfolio optimization using a neural network.

$$\text{Objective} = \max \left(\frac{\sum_{i=1}^N w_i r_i - R_f}{\sqrt{\sum_{i=1}^N \sum_{j=1}^N w_i w_j \sigma_{ij}}} \right) \quad (3.20)$$

In the neural network context, the loss function is typically the negative of the objective function (since optimization algorithms usually minimize the loss function);

$$\text{Loss} = - \text{Objective} = - \left(\frac{\sum_{i=1}^N w_i r_i - R_f}{\sqrt{\sum_{i=1}^N \sum_{j=1}^N w_i w_j \sigma_{ij}}} \right) \quad (3.21)$$

To avoid overfitting and ensure a more robust solution, regularization terms (L_1 and L_2) can be added which encourages sparsity in portfolio weights and smaller weights where L_1 is $\lambda \sum_{j=1}^N |w_j|$ and L_2 is $\lambda \sum_{j=1}^N w_j^2$. Therefore, including regularization, the combined loss function becomes:

$$\text{Loss} = - \left(\frac{\sum_{i=1}^N w_i r_i - R_f}{\sqrt{\sum_{i=1}^N \sum_{j=1}^N w_i w_j \sigma_{ij}}} \right) + \lambda_1 \sum_{j=1}^N |w_j| + \lambda_2 \sum_{j=1}^N w_j^2 \quad (3.22)$$

Where:

w_i is the weight of the $i - th$ asset in the portfolio.

r_i is the expected return of the $i - th$ asset.

σ_{ij} is the covariance between the returns of the $i - th$ and $j - th$ assets.

R_f is the risk-free rate.

λ_1 and λ_2 are regularization parameters.

In the portfolio optimisation context, neural networks can be integrated with traditional optimisation techniques like Mean-Variance Optimization or more advanced techniques like Genetic Algorithms and Particle Swarm Optimisation (Gunjan and Bhattacharyya, 2023). Ekmekcioglu (2022) wrote that a neural network has the benefits of non-linearity, adaptability and complex feature handling where NNs can capture non-linear relationships between assets and their returns, providing more accurate predictions and better optimisation results. NNs can continuously learn and adapt to new data, making them suitable for dynamic market environments. NNs can process and learn from many features and data points, which is beneficial in the multifaceted nature of financial markets.

Ekmekcioglu (2022) added that NNs have challenges of overfitting, data requirements and computational intensity where NNs can be overfit to historical data, making them less effective in predicting future returns. They require large amounts of high-quality data for training, and training neural networks can be computationally expensive and time-consuming. Therefore, neural networks offer a powerful tool for enhancing portfolio optimisation by providing sophisticated methods for return prediction and risk estimation. However, their application requires careful consideration of data quality, model complexity, and potential overfitting issues.

3.2.2 Unsupervised Learning

Unsupervised learning techniques in portfolio optimisation focus on identifying patterns, clustering assets, or reducing dimensionality within financial data without predefined labels or target variables. These techniques aim to find structure in the data that can be leveraged for better asset allocation, diversification, and risk management. The following are some standard unsupervised learning techniques used in portfolio optimisation.

K-Means Clustering

MacQueen first introduced the k-means clustering algorithm in 1967 (Wu *et al.*, 2022). The k-means algorithm partitions a given sample set into k clusters based on the distance between the samples, aiming to make the points within each cluster as close as possible and maximize the distance between different clusters. The algorithm consists of two distinct stages. In the first stage, k centers are randomly selected, with the value of k predefined. The second stage involves classifying each data point into the nearest center (Nanda *et al.*, 2010). This iterative process continues until the criterion function reaches its minimum. The clustering process is repeated for values of k from 1 to C , with the objective being to minimize the squared error E . K-means clustering seeks to divide n observations into k clusters, with each observation assigned to the cluster whose mean is closest. The algorithm starts with a single cluster, using its center as the mean of the data. For data clustering purposes, metrics such as return, risk, skewness, current ratio, and liquidity ratios are commonly used.

K clusters each summarized by a prototype μ_k and then assignment of data x_i to a cluster represented by responsibilities $r_{ik} \in \{0, 1\}$ with $\sum_{k=1}^K r_{ik} = 1$. K-means loss function is represented by.

$$J = \sum_{i=1}^n \sum_{k=1}^K r_{ik} \|x_i - \mu_k\|^2 \quad (3.23)$$

Therefore, the loss function is minimized by fixing μ_k and minimize J w.r.t r_{ik} and also fixing r_k by minimizing J w.r.t μ_k .

Sieranoja (2020) pointed out that the limitation of K-means is that challenging assignments of data points to clusters can cause a small perturbation to a data point to flip it to another cluster. Sieranoja (2020) also explained that the solution is to use the Gaussian Mixture Model (GMM), which assumes spherical clusters and equal probabilities for each cluster. However, Kenig, Todros and Adali (2024) explained that GMM is sensitive to outliers, and the solution is to use a robust loss function. In the context of portfolio optimisation, K-Means clustering can be used to group similar assets, which can help diversify the portfolio and manage risk more effectively.

Gaussian Mixture Model (GMM)

A GMM is a probabilistic model that assumes all the data points are generated from a mixture of several Gaussian distributions with unknown parameters. Each component in the mixture corresponds to a Gaussian distribution, characterised by its mean and covariance (Gao *et al*, 2020). GMMs are commonly used in clustering, density estimation, and as a generative model for various applications. It is a probabilistic story where each cluster is associated with a Gaussian distribution. To generate data, randomly choose a cluster k with probability π_k and sample from its distribution.

$$\text{Likelihood } \Pr(x) = \sum_{k=1}^K \pi_k \mathcal{N}(X|\mu_k, \Sigma_k) \quad (3.20)$$

but notice that the sum over the components appears inside the log, thus coupling all the parameters. Given the complete data $(x, z) = (x_i, z_i)$, $i = 1, \dots, n$. The parameters can be estimated by maximizing the complete data log likelihood.

$$\log \Pr(x, z|\pi, \mu, \Sigma) = \sum_{i=1}^N \sum_{k=1}^K Z_{ik} \{\log \pi_k + \log \mathcal{N}(x_i|\mu_k, \Sigma_k)\} \quad (3.21)$$

There is a need to use a procedure that would optimize the log likelihood by working with the (easier) complete log likelihood. First, fill-in the latent variables using current estimate of the parameters and then adjust parameters based on the filled-in variables and finally, maximise the expected complete log likelihood using the equation below.

$$E[\log \Pr(x, z|\pi, \mu, \Sigma)] = \sum_{i=1}^N \sum_{k=1}^K Z_{ik} \{\log \pi_k + \log \mathcal{N}(x_i|\mu_k, \Sigma_k)\} \quad (3.22)$$

Bezzina (2023) explained that the benefits of using GMM in portfolio optimisation include capturing complex return distributions, risk management, and scenario

analysis, where GMM can model non-normal distributions, providing a more accurate representation of asset returns. By better understanding the distribution of returns, GMM can enhance risk management practices, leading to more resilient portfolios and the ability to generate realistic return scenarios, which allows for more comprehensive stress testing and scenario analysis. Therefore, in portfolio optimisation, the goal is to allocate assets to maximise returns while minimising risk. Traditional methods, like the Markowitz Mean-Variance Optimisation, assume returns are normally distributed. However, financial returns often exhibit skewness and kurtosis, which deviate from normality.

Hierarchical Clustering

Hierarchical clustering organizes data into nested binary clusters, forming a hierarchical structure or tree. One key advantage of hierarchical clustering is its ability to recognize that complex systems, like financial markets, often display clear hierarchical structures that significantly influence the dynamics of such systems (Simon, 1962). This method offers a quantitative framework for understanding these hierarchies, which can aid in more accurate system modelling (Raffinot, 2018a). However, hierarchical clustering requires that the objects being classified have numerical values for a set of features. As a result, each object is represented by a feature vector, which together form the feature matrix, where each row represents an object, and each column represents a feature. The analysis is carried out by examining the rows of this matrix. These features are used to calculate the distances between objects, which then form the basis for clustering objects with similar features are grouped together, while those with differing features are placed in separate clusters (Murtagh and Contreras, 2017). In portfolio construction, for example, the correlation matrix can serve as the foundation for analysis, where each asset is treated as an object, and the features are defined as the pairwise correlations between the assets.

Dzuba and Krylov (2021) further explained that hierarchical clustering consists of two primary stages: (1) organizing the data to reflect the distances between objects, and (2) performing the clustering process while introducing hierarchy. The first stage involves selecting an appropriate distance metric and assessing the data accordingly, which typically includes calculating the distances between feature vectors. The second

stage focuses on grouping the data based on the distances calculated in the first stage and applying specific linkage criteria to determine the distance between clusters (Murtagh and Contreras, 2017). Additionally, it's important to note that there are two main types of hierarchical clustering: agglomerative and divisive. In general, the two clusters with the smallest distance are merged into a new cluster. However, different linkage criteria can define the distances between clusters in various ways, leading to the generation of distinct hierarchical trees.

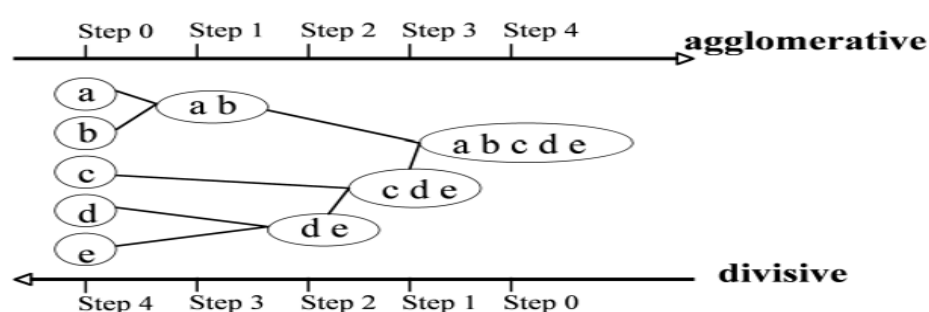


Figure 3.4 Hierarchical clustering: Agglomerative and Divisive

Source: Burghardt, Sewell and Cavanaugh (2022)

Hierarchical clustering can be utilized in portfolio optimization to group assets based on their similarity in returns or other relevant metrics (Duarte and De Castro, 2020). Marti *et al.* (2021) explained that hierarchical clustering helps categorize assets into clusters based on their statistical similarities or distances. This model can help identify groups of assets that move together or have similar risk-return profiles. By clustering assets, portfolio managers can ensure diversification by selecting assets from different clusters. Thus, diversification reduces the risk of over-concentration in a particular sector or asset type. Hierarchical clustering can assist in constructing portfolios by selecting representative assets from each cluster to achieve specific objectives like maximizing returns or minimizing risk.

Vichi, Cavicchia and Groenen (2022) put forward that the dissimilarity for two disjoint groups G and H , $d(G, H)$ is computed from pairwise dissimilarities $D(i, j)$, where $i \in G$, $j \in H$. Then single linkage tends to yield extended clusters.

$$D_{SL}(G, H) = \min_{i \in G, j \in H} D(i, j) \quad (3.23)$$

Then a complete linkage tends to yield round clusters represented as follows.

$$D_{CL}(G, H) = \min_{i \in G, j \in H} D(i, j) \quad (3.24)$$

Lastly a group average which is a trade-off between the two. Not invariant under monotone increasing transform.

$$D_{GA}(G, H) = \frac{1}{n_G n_H} \sum_{i \in G, j \in H} D(i, j) \quad (3.25)$$

Therefore, understanding which assets cluster together can provide insights into potential risk exposures. For instance, assets in the same cluster may exhibit correlated behaviour, affecting risk management strategies. Clustering can also aid in dynamically adjusting portfolio allocations as correlations and relationships between assets evolve. In practice, hierarchical clustering models provide a structured approach to analysing and managing portfolios based on underlying asset similarities or correlations.

3.3 Summary

ML can bring about a significant transformation in numerous domains by empowering computers to acquire knowledge and render decisions in manners that can prove arduous or unattainable for humans to accomplish manually. Supervised learning entails the provision of examples to the machine, along with the training and instruction of the model to establish a mapping between input data and finite target labels (Simonis, 2022). If the objective variable is a continuous numerical value, the supervised learning algorithm resolves a regression quandary instead. The supervised learning models that are widely recognised encompass linear regression, decision trees, random forests, and artificial neural networks (Khosravikia and Clayton, 2021).

Jani *et al.* (2020) explained that using unsupervised learning algorithms arises when the algorithm does not receive any labelled output in the training dataset. Ciaburro and Iannace (2021) supplemented this notion by asserting that the ultimate objective of this undertaking is to uncover and extrapolate clusters of analogous instances within the data, or more succinctly, discerning patterns and relationships within the data encompassed by the sample set being scrutinised. Hierarchical clustering, principal component analysis, and autoencoders are well-known algorithms associated with unsupervised learning. The following chapter covered the systematic comparative analysis showing the algorithms, various applications they have been used for and their application in finance, if any, with their best and worst accuracy scores.

CHAPTER 4: COMPARATIVE STUDY OF MACHINE LEARNING TECHNIQUES IN PORTFOLIO OPTIMISATION ACROSS DIFFERENT ASSET CLASSES¹

4.1 Introduction

With the advent of ML techniques, there has been a paradigm shift in how portfolio optimisation is approached (Sindi, 2023). This systematic literature review aims to critically analyse and synthesise existing research concerning applying various ML algorithms in optimising investment portfolios spanning diverse asset classes. Portfolio optimisation, a cornerstone of modern finance, aims to construct investment portfolios that maximise returns while minimising risk (Jayeola *et al.*, 2020). Traditionally, this has been accomplished by applying various mathematical and statistical techniques. The comparative study of machine learning techniques in portfolio optimisation across different asset classes has emerged as a vital area of research, offering promising avenues for improving investment strategies and enhancing portfolio performance.

Portfolio managers typically invest in various asset classes, including equities, fixed income, commodities, and alternative investments. Each asset class has distinct characteristics, requiring customized optimization strategies. A thorough review encompassing different asset classes offers valuable insights into the applicability and effectiveness of various machine learning techniques across diverse investment landscapes (Mirete-Ferrer *et al.*, 2022). According to Habbab and Kampouridis (2024), machine learning techniques have the potential to enhance portfolio performance by uncovering hidden patterns, leveraging non-linear relationships, and adapting more effectively to changing market conditions compared to traditional optimization methods. Understanding which machine learning techniques deliver superior results across different asset classes can assist investors in making more informed decisions and improving risk-adjusted returns (Habbab and Kampouridis, 2024).

¹ Conference Paper: **Comparative Study of Machine Learning Techniques in Portfolio Optimization Across Different Asset Classes**, this was presented at International Conference on Governance and Accountability (ICGA) which happened from the 5th to the 8th of November 2024 at The Zimbali Capital, Durban, South Africa.

Shahid (2024) explained that traditional portfolio optimisation techniques like Modern Portfolio Theory have long been the cornerstone of investment decision-making. However, these methods often rely on assumptions that may not hold in real-world market conditions, leading to suboptimal results. In contrast, ML algorithms promise to extract meaningful insights from vast datasets, adapt to dynamic market conditions, and potentially uncover non-linear relationships that traditional methods may overlook (Patnaik and Patnaik, 2024). Therefore, this chapter aims to review the effectiveness of ML techniques, which involves assessing the performance of various ML algorithms, such as neural networks, support vector machines, decision trees, and ensemble methods, in optimising investment portfolios across different asset classes and Assessing the limitations and challenges by highlighting the limitations and challenges associated with the use of ML in portfolio optimisation, such as data availability, overfitting, model interpretability, and computational complexity.

This chapter is organized as follows. Section 4.2 provides a review of the empirical literature (systematic literature review), detailing the search strategy, selection criteria, and data extraction process. Sections 4.3 and 4.4 cover the data and methodology, and the findings and analysis of the selected studies, respectively, followed by a discussion, critical evaluation of the research findings, and recommendations for future research. Lastly, Section 4.5 concludes the review with a summary of the main findings and their implications for both practice and research. In this context, Chapter 4 aims to address research objective 1, which involves evaluating the performance of machine learning algorithms (such as neural networks, support vector machines, decision trees, and ensemble methods) in optimizing investment portfolios across various asset classes. The chapter also explores the limitations and challenges associated with applying machine learning to portfolio optimization.

4.2 Review of Empirical Literature

A Systematic Literature Review (SLR) is a rigorous and structured approach to synthesizing existing research on a specific topic. Unlike traditional literature reviews, SLRs follow a predefined protocol to systematically identify, evaluate, and synthesize all relevant studies (Varsha, Chakraborty and Kar, 2024). The process typically involves defining clear research questions, establishing inclusion/exclusion criteria, conducting a comprehensive, unbiased search, critically appraising and synthesizing

findings and identifying research gaps and trends as presented in figure 4.1 (Carrera-Rivera *et al.*, 2022).

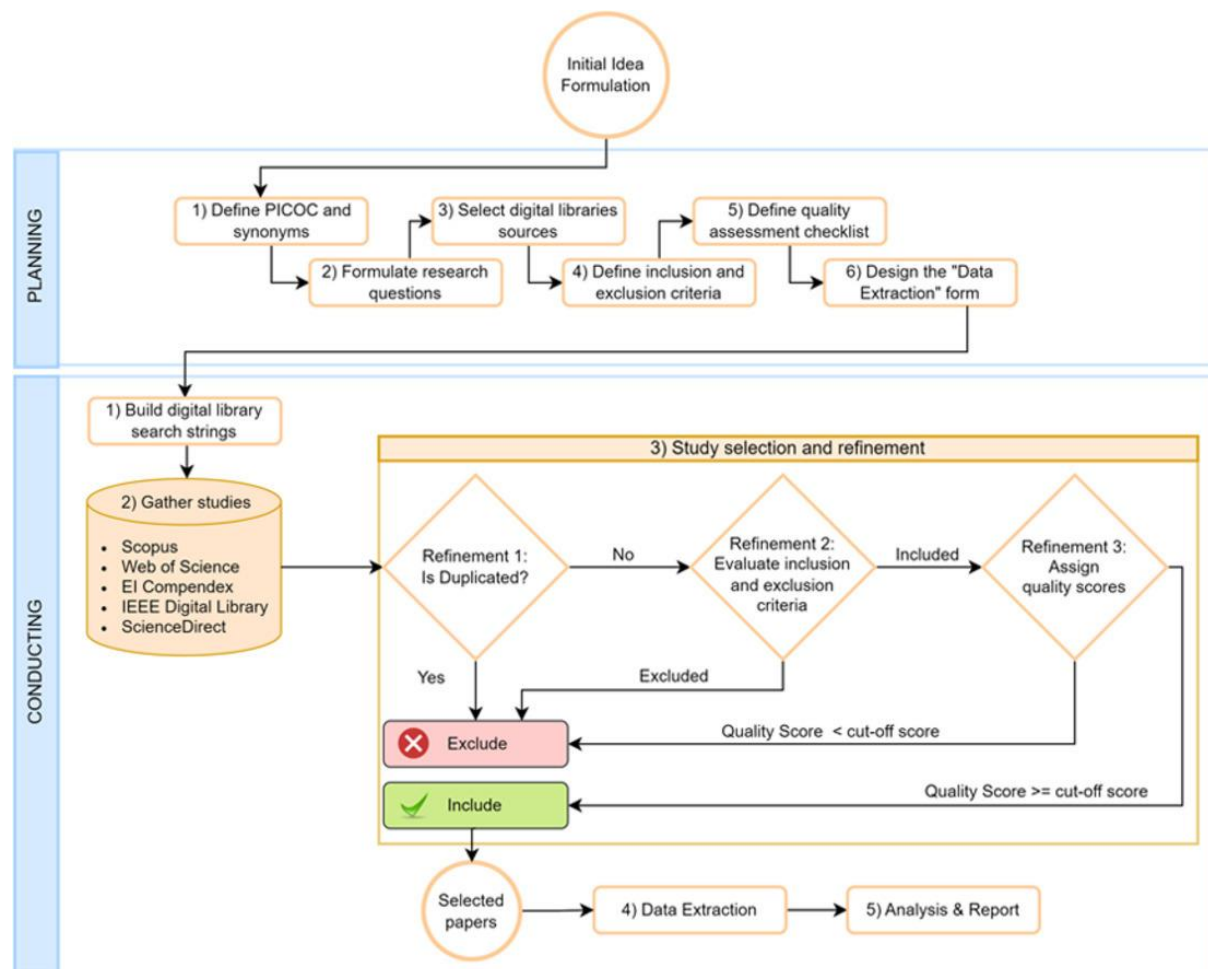


Figure 4.1 Detailed SLR framework

Source: Carrera-Rivera, Ochoa, Larrinaga and Lasa (2022)

At the PhD level, a systematic literature review offers significant value by providing a comprehensive overview of existing research, allowing for a deeper understanding of the topic. Through the synthesis of high-quality evidence, it establishes a solid theoretical foundation for new research, ensuring that future studies are grounded in robust, evidence-based insights. The structured methodology enhances transparency and reproducibility, making it easier for other researchers to replicate or extend the review, thereby increasing its credibility. Additionally, SLRs are instrumental in identifying research gaps and inconsistencies, guiding scholars toward underexplored areas and helping to avoid redundant studies (Novykov *et al.*, 2023).

In terms of contributing to knowledge, SLRs pave the way for identifying novel research directions by highlighting areas that have received limited attention, fostering innovation and the expansion of the field. They often lead to the development of new theoretical frameworks by synthesizing diverse findings, offering fresh perspectives and advancing academic discourse. Furthermore, by comparing and contrasting existing approaches, SLRs enhance methodological rigor, refining research techniques and improving future studies (Fransisca *et al.*, 2024). Finally, SLRs encourage cross-disciplinary insights, promoting interdisciplinary advancements that broaden the scope of knowledge and create new opportunities for research integration across fields.

Conducting a review of the relevant literature is a crucial aspect of any academic discipline (Mengist, Soromessa and Legese, 2020). It aids in mapping and assessing existing knowledge and identifying gaps on specific issues, ultimately contributing to the development of the knowledge base. Systematic literature review (SLR) differs from traditional narrative reviews by employing a replicable, scientific, and transparent methodology. It facilitates the collection of all relevant publications and documents that meet predefined inclusion criteria to address a specific research question (Maier, 2013). SLR follows clear and systematic procedures to minimize bias during the searching, identification, evaluation, synthesis, analysis, and summarization of studies (Mengist, Soromessa and Legese, 2020). When executed correctly with minimal error, it can yield reliable findings and conclusions that help decision-makers and researchers take informed actions. A well-structured SLR process is essential, ensuring that the review is carefully planned before the actual work begins (Pullin and Stewart, 2006). It contributes to generating more accurate estimates on the topic under investigation.

The use of a systematic literature review is supported by previous research that has employed this methodology to examine similar topics. For instance, Milhomem and Dantas (2021) reviewed the software/programming languages, constraints, heuristic and exact approaches, and technical and fundamental analytical types utilized to solve the portfolio optimization problem.

Their systematic review provided valuable insights for researchers, investors, and finance professionals on various aspects of portfolio optimization. Their findings emphasized the importance of carefully considering the input data used in optimization models. The study also suggested that the optimization results could be suboptimal compared to the 1/N trading strategy, depending on the extent of estimation errors in the input parameters.

Novykov *et al.* (2023) conducted a systematic literature review on the application of deep learning in portfolio management. Following the methodology outlined by Linnenluecke *et al.* (2020), Massaro *et al.* (2016), and Fisch and Block (2018), they began by identifying relevant literature using a carefully constructed search phrase, then filtered the resulting publications and provided both descriptive and analytical insights. The review revealed a strong prevalence of reinforcement learning algorithms in this field. Other prominent deep learning models, such as convolutional neural networks (CNN) and recurrent neural networks (RNN), along with their derivatives, are particularly effective for time-series forecasting. The authors observed a growing number of publications in this area, likely driven by advancements in computing power, hardware availability, and data access. The review also highlighted several promising applications and pointed out future research directions, including a better balance of the risk-reward spectrum, innovative approaches to reduce data dimensionality and preprocess inputs, a greater focus on direct weight generation, the exploration of new deep learning architectures, and more consistent choices of data.

Fransisca *et al.* (2024) conducted a systematic literature review on robust portfolio mean-variance in stock investment, examining studies published between 1995 and 2024. This allowed for an in-depth analysis of the evolution and effectiveness of robust portfolio optimization methods over time. The SLR followed the Preferred Reporting Items for Systematic Reviews and Meta-Analysis (PRISMA) guidelines. The review's findings introduced a novel approach combining robust optimization methods with a Genetic Algorithm (GA) to improve robust portfolio mean-variance optimization. The study identified that factors such as uncertainty parameters, cardinality constraints, optimization constraints, risk-aversion parameters, robust covariance estimators, and robustness measures were insufficient on their own to create portfolios that could maintain performance amid market uncertainties. As a result, GAs was incorporated

to address the complex optimization challenges associated with RPMV, and parameter adjustments were made to enhance solution accuracy. The study further found that the RPMV model optimized with a GA demonstrated greater stability and provided higher returns compared to traditional mean-variance models. The implications of the study highlight the importance of managing uncertainty in asset returns, adapting dynamic portfolio parameters, and leveraging GAs for solving portfolio optimization problems under uncertainty, as they offer near-optimal solutions with relatively lower computational costs.

Gallucci, Santulli and Lagasio (2022) noted that environmental, social, and governance (ESG) risks have garnered increasing academic attention since rating agencies began issuing ESG risk scores. This shift has raised questions about whether ESG risks are conceptualized by scholars in alignment with, or independent of, practitioners' views. Although often overlooked in prior literature, this issue is important as it can help determine whether there is an overlap between academic research on ESG and the practices of industry professionals. Additionally, there had been no comprehensive analysis of how ESG risks have been addressed in previous ESG portfolio studies. Their research conducted a systematic literature review using bibliometrix software, analysing 170 papers published across 88 journals between 1996 and 2021. A key finding of the study was that there is a significant overlap between ESG research and market practices.

Fernandes, Kulkarni and Pande (2023) highlighted that portfolio optimization presents a significant challenge in both research and the financial services sector. In theory, Quantum Computing can rapidly determine the optimal portfolio composition for high-return, low-risk investments, outperforming current supercomputers. To explore this, the authors conducted a review of 44 papers published between 2003 and 2023, focusing on classical and quantum machine learning (QML) approaches for Mutual Fund portfolio optimization. Given the broad scope of portfolio optimization, the study specifically concentrated on mutual funds and their connection to equity markets. The findings emphasized the growing importance of quantum machine learning in solving optimization problems as technology evolves. It was suggested that tangible benefits in terms of solution quality, computational speed, or both are likely to emerge in certain use cases. While Quantum Computing promises advancements in fields such as

machine learning and optimization, the study also pointed out that the application of QML to Mutual Fund portfolio optimization has yet to receive sufficient attention despite its considerable potential.

These SLR studies demonstrate that this approach is a suitable one for addressing research objective one. What can be noted from these existing SLRs is that there is a notable gap across the literature regarding machine learning models in optimising portfolios under dynamic market conditions. Specifically, there is limited research on integrating novel machine learning techniques, such as advanced deep learning architectures, with traditional optimisation methods to improve risk management and enhance portfolio stability in uncertain environments. Additionally, the use of machine learning models on different financial investments, specifically ETFs, remains underexplored. These gaps will be addressed in the chapters which follow, offering several contributions to existing knowledge in this field.

4.3 Data and Methodology

4.3.1 Planning

Planning begins with formulating the idea and continues through to designing the data extraction form. This process follows the steps of defining the protocol, formulating research questions, selecting digital library sources, and establishing inclusion and exclusion criteria (Carrera-Rivera *et al.*, 2022).

4.3.1.1 Defining the Protocol

The first step of a systematic literature review involves outlining the procedures to be followed, serving as a log of the activities to be carried out. It is encouraged to seek feedback from peers while developing the protocol to ensure the review's consistency and validity, and to help identify when modifications are needed (Wohlin *et al.*, 2012). A key goal of the protocol is to guarantee the review's replicability. The PICOC (Population, Intervention, Comparison, Outcome, and Context) criteria break down the review's objectives into searchable keywords, which aid in formulating research questions (Petersen, Vakkalanka and Kuzniarz, 2015).

4.3.1.2 Formulate Research Questions

Clearly defined research question(s) are the key elements which set the focus for study identification and data extraction (Wohlin *et al.*, 2012). The questions are formulated based on the PICOC criteria. The research question for this chapter: to what extent have machine learning portfolio optimisation techniques been implemented across different asset classes, and how did they perform? This research synthesizes insights from diverse financial markets, offering a comprehensive, multi-dimensional perspective on the applicability and efficacy of machine learning techniques in portfolio optimization. By transcending traditional linear optimization frameworks, this study advances modern portfolio theory, introducing a paradigm shift toward more adaptive, non-linear methodologies.

The research not only contributes novel empirical evidence but also establishes a robust evaluative framework for the future role of machine learning in portfolio management, bridging theoretical advancements with practical application. This framework equips fund managers, institutional investors, and policymakers with tailored guidance for selecting optimal techniques across varying asset classes and market conditions, thereby enhancing strategic investment decision-making.

In addition to offering a substantive contribution to the academic discourse on portfolio theory, this study redefines best practices in portfolio construction, positioning itself as a pivotal reference for both scholarly inquiry and industry innovation. Its findings will influence the trajectory of portfolio optimization research, underscoring the transformative potential of machine learning in navigating complex, volatile financial environments.

4.3.1.3 Select digital library sources

The validity of a study depends on the proper selection of a database since it must adequately cover the area under investigation (Vieira and Gomes, 2009). For this chapter the articles published from January 2015 to December 2023 were used to conduct an empirical analysis. Articles were downloaded from Scopus, EBSCOhost, Google Scholar, Science Direct, SpringerLink and Wiley Online Library using combinations of appropriate keywords ("Machine Learning*" OR "Deep Learning*" OR "Adaptive Algorithms*" OR Artificial Intelligence*" OR "Data-Driven Algorithms*" AND

"Portfolio Optimization" OR "Asset Allocation" OR "Portfolio Management" OR "Portfolio Construction" OR "Risk-Return Optimization" OR "Portfolio Balancing" OR "Optimal Asset Distribution" AND "Distressed Market" OR "Uncertainty"). The arXiv and SSRN databases were also searched to ensure that the most up-to-date research papers were included in the sample.

4.3.1.4 Define inclusion and exclusion criteria

Carrera-Rivera *et al.* (2022) emphasized the importance of defining inclusion and exclusion criteria prior to conducting the review in order to avoid bias, though these criteria can be adjusted later if needed. The selection of primary studies relies on these predefined criteria. In the initial selection, articles are included or excluded based on their abstract and key bibliographic details, which are further examined in the conducting stage. When there is uncertainty, the article is skimmed to help determine its relevance for the review. Therefore, the inclusion criteria for this study involved utilizing machine learning combined with other statistical tools or ensembles, focusing exclusively on studies comparing machine learning models in portfolio optimization and published in English. The exclusion criteria omit other subsets of AI and Computational Intelligence, studies lacking adequate details on the machine learning models used, and review papers or book chapters.

4.3.1.5 Define the Quality Assessment (QA) checklist

As outlined by Carrera-Rivera *et al.* (2022), evaluating the quality of an article requires a tool or framework that guides the assessment process. A common approach is to use a checklist that includes various factors for evaluation. A numerical scale is then applied to assess these criteria and quantify the quality assessment (Yang *et al.*, 2021). Zhou *et al.* (2015) provided a comprehensive description of assessment criteria in software engineering, categorizing them into four key aspects of study quality: Reporting, Rigor, Credibility, and Relevance. Therefore, it is crucial to apply the same scale consistently across all criteria established in the quality assessment.

4.3.2 Conducting

Once the protocol is defined, conducting the review requires the author to adhere to each of the previously outlined steps. Using tools can streamline the process, and

standard tools like Excel or Google Sheets enable collaborative work among multiple researchers. Additionally, an online tool specifically designed for conducting systematic literature reviews (SLRs) is Parsif.al. This tool allows researchers to set goals and objectives, import articles via BibTeX files, remove duplicates, define selection criteria, and generate reports.

4.3.2.1 Gather Studies

Carrera-Rivera *et al.* (2022) noted that databases with advanced search features allow researchers to perform queries based on titles, abstracts, keywords, and criteria such as publication years or research areas. Most databases support the use of logical operators (e.g., AND, OR). In general, bibliometric data from articles can be exported as CSV or BibTeX files, which are useful for both data extraction and quantitative and qualitative analysis. Additionally, researchers are encouraged to utilize reference management software like Zotero, Mendeley, Endnote or Jabref, which facilitates the easy import of bibliographic information into the software.

4.3.2.2 Study Selection and Refinement

This stage involves identifying any duplicate articles that appear across different searches within the selected databases. Articles are then included or excluded based on the established selection criteria, which are assessed by reading the titles and abstracts. The quality of the articles is then evaluated using a predefined scale. For instance, an article's quality in Parsif.al is evaluated using a simple scale: YES = 1, PARTIALLY = 0.5, and NO or UNKNOWN = 0. A cut-off score is defined to filter out articles that do not meet the quality standards. The QA process may require a brief review of the full text of each article. The goal of this stage is to eliminate duplicate and irrelevant articles. The limitation on the number of Boolean connectors resulted in 123 duplicate articles entering the screening phase.

After reviewing the titles and abstracts of the remaining articles, it became clear that several were not relevant to portfolio optimization. This was due to the search queries having multiple interpretations. For instance, while some studies referred to “portfolio” in financial terms, others used it synonymously to describe commitment or allocation of time, efforts, or resources. As a result, many unrelated articles were identified and removed. Additionally, articles that did not employ a model related to the optimization

domains mentioned, particularly those using machine learning methods, were also excluded. Out of the initial 123 articles, the irrelevant ones were discarded, leaving 61 for eligibility verification. This process follows a similar approach to the one used by Nazareth and Reddy (2023), who applied filtering options in their search, as shown in the table below.

Table 4.1: Applied Filtering during the SLR Search Process

Filtering options	Specification
Published year	2015 Onwards
Document type	Journal only
Language	English
Published stage	Final
Text availability	Full texts available
Area of study	Portfolio Optimization
Method used	Machine Learning technique

Source: Author's own compilation (2024)

To decide whether articles are suitable for review at this point, the author read each one in its entirety. In order to identify the most related articles, the eligibility stage entailed further filtering of the articles based on the inclusion and exclusion criteria.

4.3.2.3 Data Extraction

The articles that met the study selection criteria were then carefully and critically reviewed. Following this, the researcher filled out the required information using the "data extraction" form. This form captured the study characteristics and findings from

each included article through a thorough reading process. Data extraction is particularly valuable when the data requires manipulation, assumptions, or inferences. Therefore, the extracted information can be synthesized for qualitative analysis (Carrera-Rivera *et al.*, 2022). In the end, 33 relevant articles were chosen based on these criteria and used to form a database for qualitative analysis. The selected articles, as outlined in Table 4.1, include details such as the authors, publication year, study period, keywords, machine learning models used, performance metrics, and other pertinent financial application information.

Table 4.2 Summary of SLR Reviewed Articles

No:	Names of author(s) & Year	Study period	Country	Models employed	Asset Class	Performance metrics	Results
1.	Ma, Han, Wang, (2021)	2012, to 2015	China	DMLP, LSTM neural network, CNN, SVM and RF	Top 20 individual stocks	Annual returns, MDD, Sharpe ratio and Calmar ratio	RF was the best model
2.	Kumar et al. (2018)	2010 to 2017	China	SVM, RF, KNN, NB, Softmax	Amazon, Bata, Bosch, Cipla, and Eicher motor	Accuracy, F-measure	SVM performed the best
3.	Krauss et al. (2017)	1989 to 2015	United States	DNN, gradient-boosted trees, RF and several ensembles of these methods	Top 500 companies in the U. S	SD, Sharpe and Sortino ratios	Ensemble method was the best model
4.	Lee and Yoo (2020)	1997 to 2016	United States	recurrent neural network and (LSTM) neural network.	Top 50 companies	Sharpe ratio and IR	LSTM neural network performed the best
5.	Fischer and Krauss (2018)	1992 to 2015	United States	LSTM, RF and DNN	All the constituent stocks of the S&P 500	RMSE	LSTM neural networks- was the best model
6.	Paiva et al. (2019)	2001 to 2016	Brazil	SVM and Random+	All the Ibovespa index constituents	SD, Sharpe and Sortino ratios	SVM+ 1/N performed the best
7.	Huotari et al. (2020)	2014 to 2018	United States	ANN, RNN	Russell 2000 Index constituents	Sharpe ratio, p-value	ANN was the best model
8.	Ta et al. (2020)	2008 to 2018	United States	LSTM neural network, and SVM	Sectorial indices	Sharpe Ratio Herfindahl Index	LSTM neural network performed the best
9.	Chen, Zhang, and Lou (2020)	2008 to 2019	United States	LSTM and RNN	5 stocks (Pegasystems Inc.; Wolfspeed, Inc.; Shell plc; Exxon Mobil Corporation; and DigitalOcean Holdings)	APY, Sharpe ratio Calmar ratios	LSTM neural network performed the best
10.	Kaczmarek and Perez (2021)	1999 to 2019.	United States	Random Forest, DNN	9 sectors of the S&P 500	Annual mean return standard deviation Sharpe ratio Sortino ratio	Random Forest was the best model
11.	Wang, Li, Zhang, and Liu (2020)	1994 to 2019	United Kingdom	LSTM and SVM	Top 20 stocks	SD, Sharpe, and Sortino ratios	LSTM was the best performing model

12.	Tan, Yan and Zhu (2019)	2013 to 2017.	China	Random Forest and DNN	China A-shares, B-shares, H-shares, Red-chips, P- chips	Annual returns, MDD, Sharpe ratio and Calmar ratio	Random Forest was the best model
13.	Ayan (2023)	1994 to 2019	United Kingdom	LSTM, SVM and Random Forest	FTSE 100 Index constituents	Mean Return, Standard Deviation, Sortino ratio and Sharpe Ratio	Random Forest was the best performing model
14.	Ma et al. (2021)	2007 to 2015	China	Random Forest, SVR, LSTM, CNN and DMLP	China Securities 100 Index component stocks	Excess return, standard deviation, information ratio, total maximum drawdown, and turnover rate.	CNN was the best performing model
15.	Ngo and Nguyen (2023)	2016.to 2022	Viet Nam	CNN, LSTM, RNN	VN30 stocks	Sharpe ratio, Sortino ratio	LSTMs performed best
16.	Paiva, Cardoso, Hanaoka, and Duarte (2019)	2000 to 2015	Brazil	SVM and Random Forest	135 stocks listed on the Ibovespa	Cardinality, Mann-whiteness, Dunn's and Kruskal-Wallis tests	SVM+MV was the best performing model
17.	Chen, Zhong, and Chen (2020)	2010 to 2017	China	RNN and Generic Algorithm,	The SSE 50 index constituents	Sharpe ratio and IR	Generic Algorithm was the best performing model
18.	Liu (2019)	2000 to 2019	USA	SVM and LSTM RNN	S&P 500 financial stock index	RMSE	SVM was the best performing model
19.	Chen, Zhang, Mehlawat, and Jia (2022)	2009 to 2019	China	XG- Boost, LSTM, SVR and Random Forest	randomly selected 24 stocks in the SSE 50 index	SD, Sharpe and Sortino ratios	XG-Boost was the best performing model
20.	Dahal, Gupta and Pokhrel (2024)	2015 to 2019	China	MFNN, RNN, CNN, LSTM, SVM and RF	MIDCAP 50 constituents	Sharpe ratio Standard deviation Annual returns	MFNN was the best performing model
21.	Ma, Han, and Wang (2021)	2007 to 2020)	China	SVR, DMLP, LSTM, CNN and Random Forest	49 stocks after neglecting some stocks	SD, IR, TR, MD	Random Forest was the best performing model
22.	Kumar, Jain. and Singh (2021)	2010 to 2016	India	RNN and LSTM	top 10 stocks financial sector, top 10 stocks by market capitalization, top 10 individual stocks energy sector.	Variance CVaR Sharpe ratio	-RNN was better in stability
23.	Wu et al. (2021)	2017 to 2019	Taiwan	Portfolio Management System (PMS) using reinforcement learning with two neural networks (CNN and RNN).	50 premium stocks approved by the government of Taiwan, Taiwan 50 (TW50)	Sharpe ratio	PMS was the best performing model
24.	Min et al. (2021)	2010 to 2019	China	LSTM and XGBoost	daily returns of 12 industries	Sharpe ratios and Calmar ratios	XGBoost was the best performing model
25.	Chen, Zhang, Mehlawat, and Jia, (2021).	2009 to 2019	China	RF, SVR, XGBoost AND LSTM	100 large cap companies	mean return standard deviation Sharpe ratio Sortino ratio	XGBoost was superior

26.	Mousavi Anzaehai, and Nikoomaram (2022)	2014 to 2019	Iran	SVM and CatBoost	TEDPIX constituents	LMSE and Sharpe ratio	SVM+ 1/N performed the best
27.	Mohiuddin, (2021)	2018 to 2021	India	LSTM and MLP	Top 10 stocks	RMSE and expected returns	LSTM was the best performing model
28.	Alexandre (2022)	2003 to 2018	Brazil	Random Forests, gradient boosting, and neural networks and Ensemble model	572 Brazilian stocks	monthly return monthly standard deviation	Random Forests model was the best performing model
29.	Nazarethand Reddy (2023)	2002 to 2022	United Kingdom	CNN, LSTM, BiLSTM, and CNN-LSTM	FTSE AIM UK 50 Index constituents	Sharpe Ratio Herfindahl Index	CNN-LSTM was the best performing model
30.	Park et al. (2022)	2009 to 2019	USA	DNN, Q-Learning and LSTM	FTSE 250 Index constituents	Cum. return, Sharpe ratio, Turnover	LSTM was the best performing model
31.	Rasekhschaffe and Jones (2019)	2008 to 2018	USA	SVM, Ada Boost and DNN	small, medium and large cap stocks	Excess return, Alpha	SVM was the best performing model
32.	Xiaolei, et.al (2020)	2010 to 2020	India	RF, SVM, LGBM	NIFTY 50 constituents	P value and Sharpe ratio	LGBM was the best performing model
33.	Habbab and Kampouridis (2024)	2019 to 2021	US, UK, and Australia	SVR, XGBoost and LSTM	three asset classes (stocks, bonds, and real estate)	Sharpe ratio	SVR was the best-performing algorithm

Source: Author's Compilation (2024)

4.3.2.4 Preliminary Analysis and Report

Figure 4.2 gives an overview of the inclusion of machine learning systems in different sectors before the explanation and discussion of the results from the SLR.

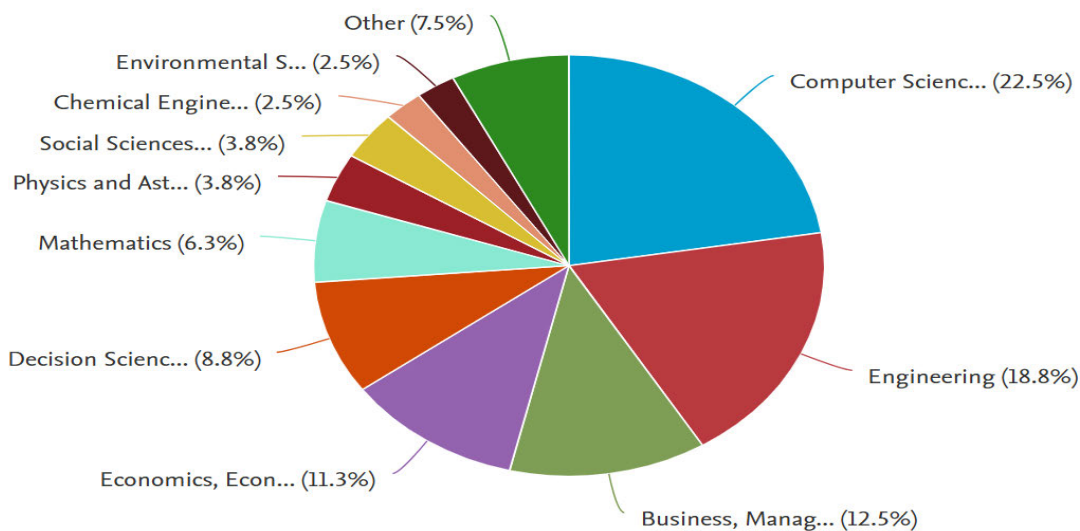


Figure 4.2 Inclusion of Machine Learning systems in Different Sectors

Source: Author's compilation from literature review (2024)

Which sector has the greatest number of significant machine learning systems inclusion? Machine Learning (ML) is widely adopted across diverse sectors, with Computer Science (22.5%) leading due to its role in developing ML algorithms for applications like natural language processing, computer vision, and robotics. Engineering (18.8%) follows, leveraging ML for predictive maintenance, automation, and system optimization in areas such as autonomous vehicles and smart grids. Business and Management (12.5%) and Economics (11.3%) use ML for decision-making, forecasting, and operational efficiency, while Decision Sciences (8.8%) integrate ML to handle uncertainty and optimize complex systems. Mathematics (6.3%) provides the theoretical foundation for ML algorithms, and Physics and Astronomy (3.8%) apply ML to analyse massive experimental datasets. Sectors like Social Sciences (3.8%), Chemical Engineering (2.5%), and Environmental Sciences (2.5%) are increasingly adopting ML for diverse applications, from policy analysis and material discovery to climate modelling. ML's versatility and potential for innovation are further demonstrated by its growing role in fields such as healthcare, agriculture, and law, emphasizing its transformative impact across domains.

After searching through the databases, a list of 123 identified papers were reached, which in turn lead to the overlay visualization shown in figure 4.3 before cutting down to the 33 articles. The VOSviewer overlay visualization highlights the evolution and focus areas in research on machine learning techniques in portfolio optimization. Central themes include portfolio optimization, machine learning, and financial data processing, indicating their frequent interconnection. The clusters reveal foundational work on stochastic models, genetic algorithms, and artificial intelligence, as well as recent advancements like deep learning and reinforcement learning, which emphasize risk management and predictive modelling under uncertainty. Emerging trends, represented by newer keywords like reinforcement learning and uncertainty, show a shift toward integrating advanced machine learning approaches in portfolio management and financial markets.

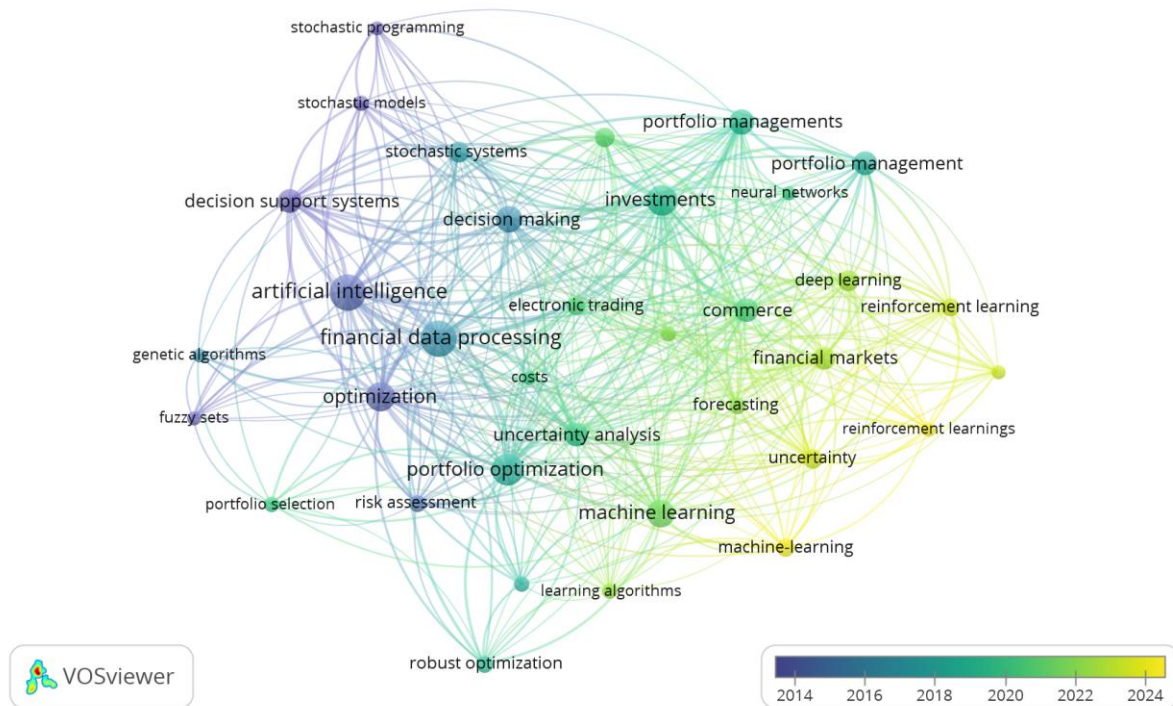


Figure 4.3 Overlay Visualization of Themes from SLR

Source: Author's compilation (2024)

Exploring the detail in Figure 4.3 offers the first set of insights to this SLR. A systematic review should focus on comparing the effectiveness of these evolving techniques, considering their ability to handle risks, uncertainty, and diversification across various asset classes.

The timeline indicates earlier emphasis (pre-2016) on traditional optimization approaches, such as stochastic programming, robust optimization, and genetic algorithms. These methods were instrumental in laying the groundwork for quantitative finance, emphasizing risk management through more deterministic and rule-based models. By 2018, the focus shifts toward machine learning and artificial intelligence techniques, with terms like neural networks, learning algorithms, and decision support systems gaining prominence. This marks the integration of data-driven approaches, indicating that the research community started exploring how machine learning can enhance traditional optimization. Recent years (2020–2024) see a surge in advanced machine learning methodologies, particularly deep learning and reinforcement learning, with an emphasis on handling volatility, predictive modelling, and decision-

making under uncertainty. This transition suggests that machine learning is becoming more sophisticated in its application to complex financial problems.

This progression is somewhat expected given the broader trend in data science toward leveraging AI and machine learning to solve increasingly complex problems. In finance, this shift is driven by the need for real-time, data-rich solutions to manage risk and optimize returns in volatile markets. The evolution from robust, rule-based optimization models to adaptive, machine-learning-driven approaches shows how financial markets have grown more dynamic, necessitating models capable of real-time learning and adjustment to new market conditions. The keywords associated with diversification and uncertainty hint that machine learning is not confined to equity markets but is also being adopted across various asset classes, from bonds to alternative investments, enhancing the potential for portfolio diversification and risk mitigation.

4.4 Discussion of Findings and Analysis

This section of the chapter discusses the method used for reviewing the performance and comparison of machine learning techniques in portfolio optimization across different asset classes. The analysis phase examines the synthesized data and extracts meaningful information from the selected articles. This section also details the discussion of the key findings in terms of the geographical map, the application of machine learning techniques across different assets classes and their effectiveness and lastly the performance metrics used.

4.4.1 Geographical Map

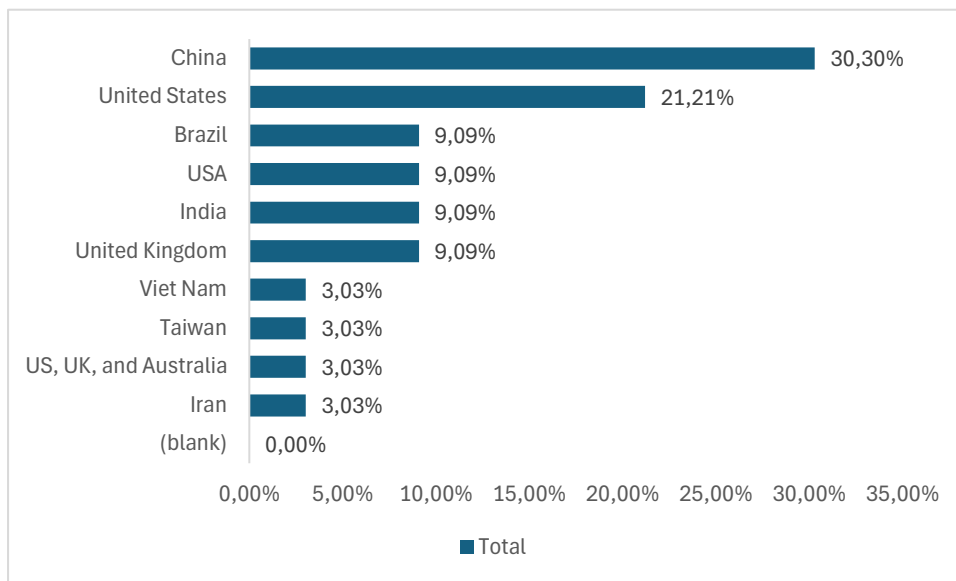


Figure 4.4 SLR Articles Grouped by Location

Source: Author's compilation (2024)

The bibliographic map illustrates the geographical distribution of research applying machine learning techniques across various asset classes. China leads with a significant contribution of 30.30%, followed by the United States at 21.21%, highlighting their dominant role in advancing machine learning research in finance. Other countries, including Brazil, India, the USA, and the United Kingdom, each contribute 9.09%, indicating a broader global interest in leveraging these technologies. Vietnam, Taiwan, and a combined category (US, UK, and Australia) represent smaller contributions at 3.03%, suggesting emerging interest or limited resources in these regions. Iran also contributes 3.03%, while a blank category indicates some unspecified or unclassified entries. The distribution emphasizes a concentration of research activity in developed and emerging economies, with China and the US taking the lead in exploring innovative approaches to asset class analysis using machine learning portfolio optimization techniques.

Additionally, figure 4.5 adds to the geographical map analysis based on bibliographic data. Figure 4.5 shows that China and India are the most recent economies applying machine learning portfolio optimization from 2022 to 2023.

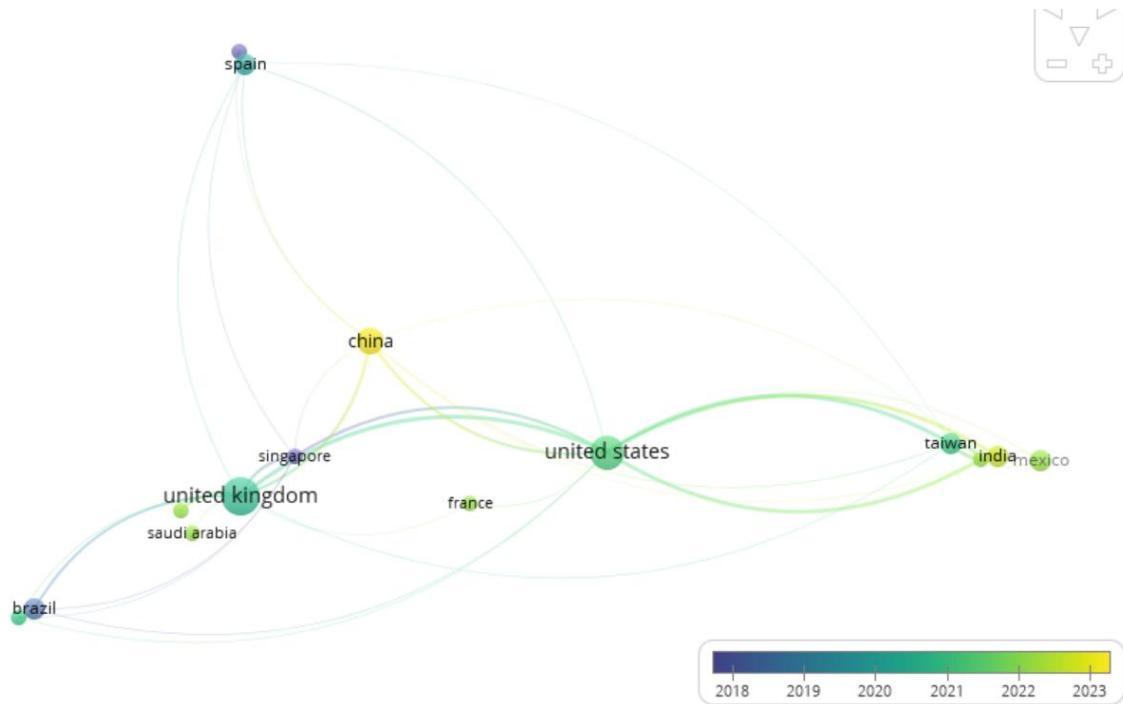


Figure 4.5 Geographic Map based on Bibliographic Data

Source: Author's estimation (2024)

4.4.2 Review of ML Techniques and their implementation

The chapter was aimed at answering the research question: To what extent have machine learning portfolio optimisation techniques been implemented across different asset classes? In attempting to achieve the aforementioned objective and provide the answer to the research question the study undertook a strategic literature review of 33 works and found out that Models employed include SVM, RF, LSTM, DNN, ensemble methods and hybrid approaches including LSTM, XGBoost).

Distribution of Machine Learning Models in Reviewed Studies (n = 33)

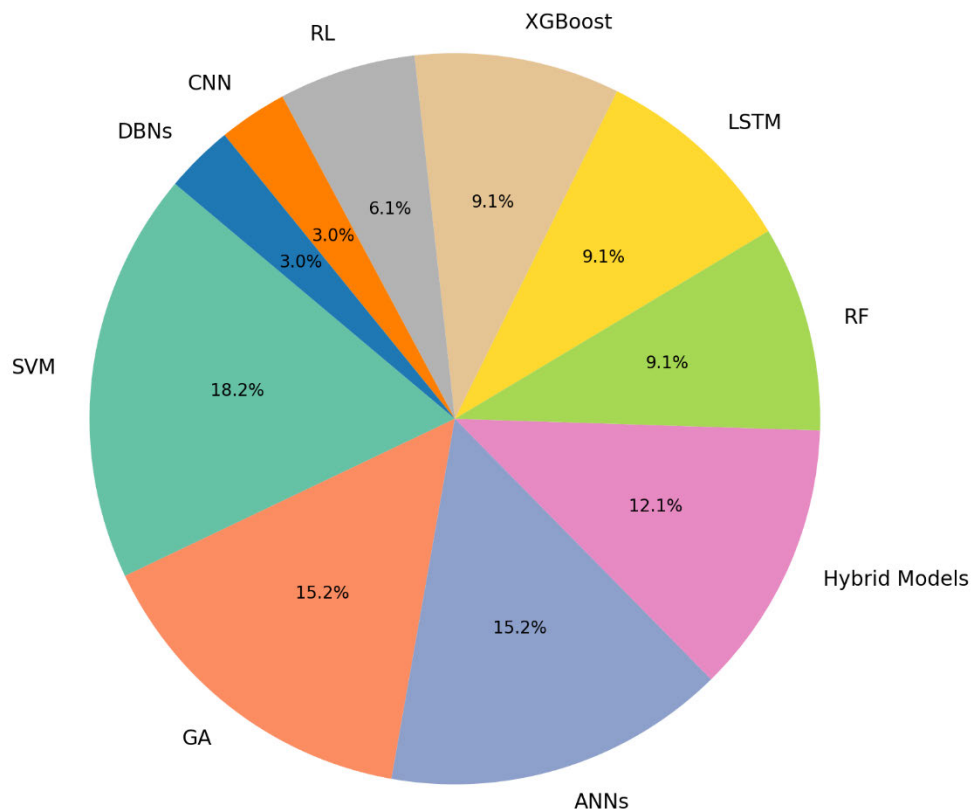


Figure 4.6: Distribution of Machine Learning Models

The analysis of the 33 studies reviewed reveals distinct trends in the application of machine learning techniques for portfolio optimization. SVM emerged as the most widely used model, appearing in 6 studies, which accounts for approximately 18.2% of the sample. GA and ANNs followed closely, each used in 5 studies (15.2%). Hybrid models combinations of two or more ML techniques were also notable, featuring in 4 studies (12.1%), indicating a growing interest in integrated approaches for improved performance. Ensemble and deep learning methods such as RF, LSTM networks, and XGBoost each appeared in 3 studies (9.1%), showcasing their popularity for handling complex, high-dimensional data. Less frequently used were Reinforcement Learning (6.1%), CNNs, and DBNs, each appearing in 1 or 2 studies. This distribution underscores the dominance of supervised learning and evolutionary techniques in financial ML research, while also highlighting the emerging role of hybrid and deep learning models in addressing portfolio optimization challenges.

4.4.3 Review of ML Techniques and their Effectiveness

In terms of model performance variability, different machine learning models demonstrate varied performance across different asset classes and evaluation metrics. For instance, Random Forest models often outperform other models in multiple studies (Ma *et al.*, 2021; Tan, Yan, and Zhu, 2019; Kaczmarek and Perez, 2021), but other models like LSTM neural networks (Fischer and Krauss, 2018; Ta *et al.*, 2020; Ngo and Nguyen, 2023) and XGBoost (Min *et al.*, 2021; Chen *et al.*, 2021) also show superior performance in certain contexts. Studies that use ensemble methods and advanced techniques (e.g., Krauss *et al.*, 2017; Alexandre, 2022) or hybrid models combining multiple algorithms (e.g., Pinelis and Ruppert, 2022) often achieve superior results, indicating the benefit of integrating different machine learning techniques. To add on, some studies focus on specific sectors or indices (e.g., financial sector stocks in Liu, 2019; FTSE 100 Index in (Wang, Shen and Li, 2022), suggesting that model performance can be context dependent.

None of the previous studies studied portfolio optimization in ETFs. Many of the studies focused on stock indices only however given that ETFs track various asset classes and represent a low-cost transformation of an index fund which is easy to access (Steyn, 2019). According to Dannhauser (2017), ETFs have greater transparency, lower management fees, and tax efficiencies and, thus, are able to attract both retail and institutional investments. To add on machine learning techniques are classified into 3 families namely shallow learning, deep learning, and gradient boosting, none of all these families have been introduced to ETF portfolio optimization. Moreover, none of the previous studies carried out their research in the African stock market specifically South Africa. Therefore, these gaps have been attended to in the next chapter.

Random Forest (RF): RF often shows strong performance, frequently outperforming other models such as LSTM, SVM, and DNN. RF is used across various asset classes, including individual stocks, sector indices, and large indices (e.g., S&P 500, FTSE 100). RF is evaluated using metrics such as Sharpe ratio, Sortino ratio, standard deviation (SD), and annual returns. For instance, in studies by Ma *et al.* (2021), Kaczmarek and Perez (2021), and Alexandre (2022), RF was the best-performing model, highlighting its robustness in different market conditions and data sets. Sharpe

ratio has been widely used across studies to measure risk-adjusted return. Models like LSTM, RF, and ensemble methods frequently maximize this metric, indicating their effectiveness in optimizing return for a given risk level.

Long Short-Term Memory (LSTM): LSTM performs exceptionally well in many studies, particularly in handling time-series data due to its ability to capture long-term dependencies. LSTM is applied to top company stocks, indices like S&P 500 and FTSE 100, and various sector-specific stocks. Metrics such as RMSE, Sharpe ratio, and Calmar ratio are used to gauge LSTM's performance. Studies by Fischer and Krauss (2018), Ta *et al.* (2020), and Ngo and Nguyen (2023) show LSTM outperforming other models, demonstrating its effectiveness in financial forecasting and portfolio management.

Support Vector Machine (SVM): SVM shows strong performance, especially when combined with other methods (e.g., SVM+ 1/N) or in specialized contexts. SVM is used for individual stocks, sectoral indices, and broader indices. It often competes with RF and LSTM. SVM's performance is evaluated using accuracy, F-measure, Sharpe ratio, and RMSE. For example, in studies by Kumar *et al.* (2018) and Liu (2019), SVM performed the best, indicating its reliability in classification and regression tasks.

Convolutional Neural Networks (CNN): CNNs are less commonly used alone but often perform well in combination with other models (e.g., CNN-LSTM). Applied to individual stocks and indices, CNNs are valued for their pattern recognition capabilities. Evaluated using metrics like Sharpe ratio and Herfindahl Index. Wu *et al.* (2021) and Pinelis and Ruppert (2022) found CNN-based models to be highly effective in specific scenarios.

Ensemble Methods: Ensemble methods often outperform individual models due to their ability to integrate multiple algorithms' strengths. Used across various asset classes, including top U.S. companies and Brazilian stocks. Metrics such as Sharpe ratio, Sortino ratio, and annual returns are used. Krauss *et al.* (2017) and Alexandre (2022) demonstrated that ensemble methods can generate superior results. Models were frequently applied to individual stocks and broad market indices like S&P 500, FTSE 100, and sector-specific indices. Techniques like LSTM and RF were particularly effective here. To add on, sectoral applications are common, with studies focusing on

financial, energy, and technology sectors. Models like SVM and RF often show strong performance in these specific contexts. The effectiveness of models can vary by region due to different market conditions. Studies encompass a range of countries, including the U.S., China, Brazil, and India, demonstrating the versatility and adaptability of machine learning techniques across diverse markets.

The analysis of various machine learning techniques in portfolio optimization indicates that no single model consistently outperforms others across all scenarios. However, models like Random Forest and Long Short-Term Memory consistently demonstrate strong performance across different asset classes and metrics. The selection of a model typically depends on the specific context, asset class, and performance metric being prioritized. Using multiple evaluation metrics, such as the Sharpe ratio, Sortino ratio, and Root Mean Square Error (RMSE), allows for a more thorough assessment of a model's effectiveness, underscoring the importance of risk-adjusted returns and prediction accuracy in portfolio optimization strategies. This variety highlights the broad applicability of machine learning models in financial markets. Regarding evaluation metrics, a range of performance measures is used in different studies, including Sharpe ratio, Sortino ratio, annual returns, maximum drawdown (MDD), RMSE, and accuracy. The choice of metrics is often aligned with the specific goals of the portfolio optimization, whether focusing on risk-adjusted returns, prediction accuracy, or overall stability.

4.4.4 Performance Metrics

The Sharpe Ratio has been widely used across studies to measure risk-adjusted return, with models like LSTM, RF, and ensemble methods frequently maximizing this metric, indicating their effectiveness in optimizing return for a given risk level. Similarly, the Sortino Ratio focuses on downside risk, making it a crucial metric for assessing models' performance in avoiding significant losses, as used in studies by Krauss *et al.* (2017) and Paiva *et al.* (2019). Annual Returns and Maximum Drawdown (MDD) are essential for understanding absolute performance and risk of significant losses, highlighted in studies like Ma *et al.* (2021) and Tan *et al.* (2019) for evaluating models like RF and DNN. RMSE is used to measure prediction accuracy, particularly in time-series forecasting, with LSTM and SVM often evaluated using RMSE, as seen in Fischer and Krauss (2018) and Mohiuddin (2021). The Information Ratio (IR)

measures risk-adjusted return relative to a benchmark, employed in studies like Lee and Yoo (2018) and Chen, Zhong and Chen (2020).

4.4.5 Strengths and Weaknesses in Existing Research

The studies employ a wide variety of machine learning models, including traditional models like Support Vector Machine and Random Forest, as well as advanced neural networks such as Long Short-Term Memory and Convolutional Neural Networks. This diversity allows for comprehensive comparative analyses. The use of multiple performance metrics such as Sharpe ratio, Sortino ratio, maximum drawdown, annual returns, and Root Mean Squared Error provides a thorough evaluation of model performance. These metrics capture different aspects of risk-adjusted returns and prediction accuracy. The studies cover a broad range of countries (e.g., USA, China, Brazil, India) and various time periods, enhancing the generalizability of the findings. This geographic and temporal diversity helps in understanding the models' performance in different market conditions and regulatory environments. The inclusion of different asset classes such as individual stocks, sectoral indices, and market indices (e.g., S&P 500, FTSE 100) adds robustness to the analysis. This variety helps in assessing the models' adaptability and performance across different types of financial assets. Some studies focus on emerging markets (e.g., Vietnam, Brazil, Iran), which are often underrepresented in financial research. This inclusion helps in understanding the applicability of ML models in less mature and potentially more volatile markets.

On the other side, there is a lack of standardization in methodologies, including differences in data pre-processing, feature selection, and model evaluation techniques. This inconsistency makes it challenging to directly compare the results across different studies. Many studies might suffer from data bias and overfitting due to limited datasets or the tendency to use in-sample data for model training and testing. This can lead to overly optimistic performance metrics that may not generalize well to unseen data. While some studies explore ensemble methods, this area remains underrepresented. Ensemble methods often enhance predictive performance and robustness, suggesting a need for more extensive research in combining multiple ML models.

Certain asset classes, such as bonds, commodities, and real estate, are rarely analysed. Additionally, many studies focus on well-known indices or large-cap stocks, neglecting small-cap stocks and other niche markets that could benefit from ML-driven insights. The rapid evolution of ML techniques means that newer algorithms like Transformer models, attention mechanisms, and advanced reinforcement learning approaches are not adequately explored. These methods could potentially offer superior performance in portfolio optimization.

4.4.6 Implications for Investors, Financial Institutions and Researchers

Individual and institutional investors can leverage the insights from these studies to enhance their portfolio performance. Understanding which models perform best under specific conditions can help in selecting the most appropriate strategy for different market environments. Banks, hedge funds, and other financial institutions can implement advanced machine learning models to optimize their asset management strategies. The demonstrated success of models like Random Forest and LSTM in various studies provides a roadmap for institutions aiming to adopt data-driven approaches. The diversity of methods and metrics used in these studies offers a rich foundation for future research. Researchers can build on these findings by exploring new machine learning techniques, hybrid models, or by applying these models to different markets and asset classes.

4.5 Summary

Models employed include SVM, RF, LSTM, DNN, ensemble methods and hybrid approaches including LSTM, XGBoost). Different models perform best in different studies, with RF, LSTM, SVM, and ensemble methods frequently cited as top performers. Performance metrics vary across studies, including annual returns, standard deviation (SD), Sharpe ratio, Sortino ratio, mean return, RMSE, excess return, etc. Sharpe ratio is commonly used as a key metric to evaluate model performance, along with other risk-adjusted metrics like Sortino ratio and information ratio. Results indicate varying model performances across studies and datasets. RF, LSTM, SVM, and ensemble methods appear as top performers in different studies. SVM often stands out in terms of classification accuracy, while LSTM tends to excel in time series forecasting tasks where forecasting was also used in portfolio optimization.

Ensemble methods show promising results in portfolio optimization from multiple models. Overall, while certain models like LSTM, RF, and SVM consistently perform well across different studies, the choice of the best-performing model can vary depending on factors such as the dataset characteristics, study period, and specific performance metrics of interest highlighting the importance of experimenting with different methodologies for portfolio optimization.

The literature reviewed reveals a growing body of research on the financial applications of machine learning since 2015. Recently, there has been a continuous surge in interest around stock market portfolio optimization, with a notable spike in cryptocurrency research in 2021, reflecting heightened attention from both researchers and academics. Significant contributions in this field have come from China and the United States, followed by India. However, a comprehensive global comparison of machine learning models in portfolio optimization remains limited. This study seeks to address this gap by systematically reviewing the most widely applied machine learning techniques in portfolio optimization. The review uncovers several important findings about the use of machine learning in portfolio optimization across different countries and timeframes. Future research should focus on achieving greater standardization in data processing and evaluation metrics to enable more direct comparisons across studies. Establishing benchmarks and using common datasets could aid in this effort.

CHAPTER 5: THE PERFORMANCE OF TRADITIONAL VERSUS MACHINE LEARNING PORTFOLIO OPTIMISATION IN ETF PORTFOLIOS

5.1 Introduction

Traditional portfolio optimization techniques, rooted in theories such as modern portfolio theory and Markowitz mean-variance optimization, have long been the cornerstone of asset allocation strategies (Bechis, 2020). These methods typically rely on statistical analysis and historical data to construct portfolios that balance risk and return based on predetermined criteria (Hult, 2012). On the other hand, machine learning algorithms offer the promise of leveraging vast amounts of data and complex relationships to optimize portfolios in a dynamic and adaptive manner (Devan, Thirunavukkarasu and Shanmugam, 2023). Therefore, by incorporating machine learning models, investors aim to uncover patterns, exploit market inefficiencies, and potentially achieve superior risk-adjusted returns. Ta, Liu and Addis (2018) mentioned that with the advent of machine learning techniques and the increasing availability of data, there has been growing interest in applying ML algorithms to portfolio optimization. Machine learning techniques offer the potential to capture more complex patterns in data, incorporate non-linear relationships, and adapt to changing market conditions more effectively compared to traditional methods (Sarker, 2021).

Machine learning techniques can handle large and complex datasets, including various asset classes and historical market data, which may provide more insights into portfolio construction compared to traditional techniques (Jaeger *et al.*, 2021). Conversely, traditional portfolio optimization techniques often rely on simplifying assumptions about asset returns distributions, correlations, and risk measures, which may not fully capture the complexities of financial markets (Kolm, Tütüncü and Fabozzi, 2014). Machine learning models can accommodate more flexible assumptions and adapt to changing market conditions, potentially leading to more robust portfolio strategies (Dixon, Halperin and Bilokon, 2020). However, the question remains if ML techniques can offer better diversified portfolios compared to traditional optimization techniques in exchange traded funds?

Effective risk management is crucial in portfolio optimization to protect against downside risk and unexpected market movements (Bechis, 2020). Machine learning algorithms can incorporate a broader range of risk factors and nonlinear relationships,

allowing for more sophisticated risk management strategies compared to traditional methods (Aziz and Dowling, 2019). Comparing the performance of traditional versus machine learning portfolio optimization techniques requires comprehensive back testing and performance evaluation. Metrics such as risk-adjusted returns, volatility, maximum drawdown, and Sharpe ratio can be used to assess the effectiveness of different portfolio strategies over historical data (Molyboga and L'Ahelec, 2017). Therefore, the comparison between traditional and machine learning portfolio optimization techniques in the context of ETFs is an area of active research and debate within the investment community. While machine learning offers the potential for more sophisticated and adaptive portfolio strategies, it also poses challenges related to data availability, model interpretability, and computational complexity. Therefore, a nuanced understanding of both traditional and machine learning techniques, as well as their respective strengths and limitations, is essential for effective portfolio management in the ETF space.

Understanding the comparative performance of traditional versus machine learning portfolio optimization techniques is crucial for investors, portfolio managers, and researchers alike. This research endeavours to shed light on this important question by conducting a comprehensive comparative assessment of these methodologies, evaluating their efficacy under varying market conditions. Despite the growing interest in the application of machine learning techniques to portfolio optimization, there remains a notable gap in the literature regarding their comparative performance with traditional methods specifically in the context of ETF portfolios.

Furthermore, existing research often lacks comprehensive evaluations across various performance metrics, such as risk-adjusted returns, portfolio diversification, and robustness between ML and traditional portfolio optimisation techniques. Understanding the strengths and weaknesses of each approach in managing ETF portfolios is essential for informing investment decisions and improving portfolio management practices. Therefore, this study aims to address this gap by conducting a rigorous comparative assessment of traditional and machine learning portfolio optimization techniques applied to ETFs in South Africa. By examining their performance various performance metrics, such as risk-adjusted returns, portfolio

diversification, this research seeks to provide valuable insights into the relative efficacy of these methodologies and their implications for investors and portfolio managers.

In comparing the performance of traditional versus machine learning portfolio optimisation techniques in exchange traded funds, the study seeks to provide answers to the following research questions; Do machine learning portfolio optimisation techniques yield higher risk adjusted returns than traditional portfolio optimisation techniques in ETF portfolios? Do machine learning portfolio optimisation techniques reduce risk more than traditional portfolio optimisation techniques in ETFs? Do machine learning portfolio optimisation techniques offer more effective diversification than traditional portfolio optimisation techniques in ETFs?

The chapter is structured as follows: Section 5.2 offers an overview of the traditional and machine learning portfolio optimization techniques selected for the study. Section 5.3 presents a review of the relevant empirical literature. Section 5.4 outlines the data and methodology employed in the study. Section 5.5 discusses the results and provides the empirical analysis. Finally, Section 5.6 concludes the chapter.

5.2 Traditional and Machine Learning Portfolio Optimization Techniques

Three traditional models, namely the Markowitz model, CVaR, and MAD, were employed. Additionally, three models, each from three different families of machine learning techniques were utilized. These families include shallow learning, an enhanced radial basis artificial neural network known as K4-RANN, and a deep learning technique with Evolutionary (Genetic) Algorithms. Gradient Boosting with XG Boost was also considered.

Traditional portfolio optimization techniques are rooted in mathematical and statistical models and have been widely used due to their simplicity, theoretical foundations, and intuitive appeal. The following techniques primarily focus on achieving efficient allocation of assets by minimizing risk or maximizing returns, often subject to constraints such as budget limits, sector diversification, or regulatory requirements. Traditional methods leverage historical data to estimate key parameters, including expected returns, variances, and covariances, which form the basis of portfolio construction.

5.2.1 The Markowitz Model (Traditional)

The Markowitz model, also known as modern portfolio theory, was developed by Harry Markowitz in 1952 and laid the foundation for portfolio management and investment theory. It provides a framework for investors to optimize their portfolios by balancing risk and return. The central idea of the Markowitz model is that investors seek to maximize returns while minimizing risk (Mangram, 2013). Return refers to the profit or loss generated by an investment, while risk refers to the uncertainty or variability of those returns (Markowitz and Van Dijk, 2008). The efficient frontier, as defined by Markowitz, is a set of optimal portfolios that provide the best risk-return trade-off (Gambeta and Kwon, 2020), the highest expected return for a given level of risk, or the lowest risk for a given level of return (Kumar, 2018). The figure below illustrates the efficient frontier concept.

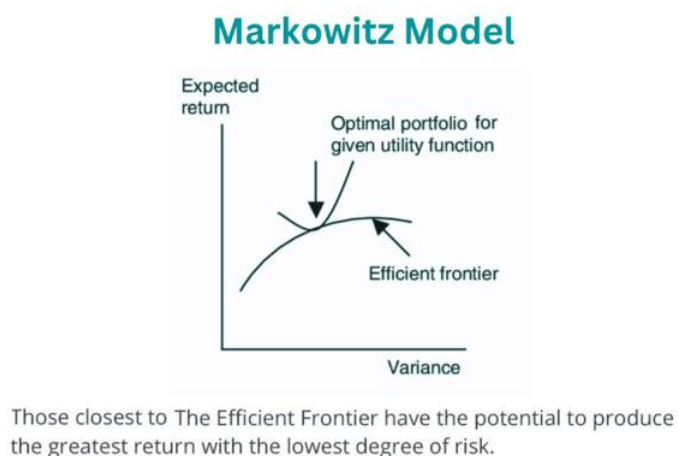


Figure 5.3 The Efficient Frontier

Source: Gajendrakar (2016)

The efficient frontier represents a set of investment portfolios that are expected to deliver the highest returns for a given level of risk (Agarwal and Muppalaneni, 2022). A portfolio is considered efficient if no other portfolio offers higher returns for the same or less risk (Zanjirdar, 2020). The position of portfolios on the efficient frontier depends on the investor's risk tolerance. The efficient frontier forms a curved line because each increase in risk corresponds to a relatively smaller increase in returns (Agarwal and Muppalaneni, 2022). This reflects the concept of diminishing marginal returns to risk, which creates the curvature (Kimball *et al.*, 2024). By diversifying assets within a

portfolio, both returns can be enhanced and risks reduced, positioning the portfolio on the efficient frontier (Maiti, 2021). Thus, diversification plays a key role in forming an efficient portfolio along the curved line.

The Markowitz model defines risk as the variance or standard deviation of returns. The expected return refers to the average return anticipated from an investment (Chen, 2023). By calculating the expected return and variance of individual assets, investors can create portfolios that align with their desired risk and return levels (Rustagi, 2021). An important feature of the Markowitz model is the consideration of the correlation between assets. Correlation gauges how closely the returns of two assets move in relation to each other (Mercangöz, 2021). Diversifying across assets with low or negative correlations can effectively reduce the overall risk of the portfolio.

The Markowitz model offers a mathematical framework for optimizing portfolios based on an investor's risk preferences (Mittal, Bhattacharya and Mandal, 2022). Using mathematical methods like mean-variance optimization, investors can determine the ideal asset allocation to meet their investment objectives (Agarwal and Muppalaneni, 2022). The model is based on certain assumptions, such as investors being rational and risk-averse, making decisions based on expected returns and the standard deviation of returns. Additionally, it assumes that asset returns follow a normal distribution (Cavacini, 2022).

Although the Markowitz model offers a valuable theoretical foundation, its practical implementation faces challenges, including the estimation of expected returns and variances, the assumption of constant correlation, and its sensitivity to input parameters. The model revolutionized portfolio management by introducing a structured method for balancing risk and return (Milhomem and Dantas, 2020). Despite its limitations and criticisms, it continues to be a cornerstone of finance and forms the basis for many contemporary portfolio management strategies (Sanjay, 2022).

5.2.2 Conditional Value at Risk (CvaR) (Traditional)

CVaR represents the expected loss given that the loss exceeds a certain threshold. It measures the average loss beyond the CVaR threshold, considering all scenarios where losses exceed this threshold (Benati and Conde, 2022). The formula for

minimizing CVaR was initially developed by Rockafellar and Uryasev (2000). The employment of the optimization procedure to decrease CVaR, however, requires a numerical routine that lacks resilience when confronted with extreme returns, as observed by Huang *et al.* (2008). CVaR can be interpreted as the expected magnitude of loss beyond the CVaR level. For example, if the CVaR at a 95% confidence level is R1 million and the CVaR is R1.5 million, it means that if losses exceed R1 million, on average they are expected to be R1.5 million.

Incorporating CVaR into portfolio optimization models the goal is to construct a portfolio that maximizes returns while managing risks effectively (Dixit and Tiwari, 2020). Traditional mean-variance optimization considers only the first two moments of the return distribution (mean and variance), often assuming a normal distribution (Rigamonti, 2020). Thus, incorporating CVaR into portfolio optimization allows investors to explicitly consider downside risk and tailor portfolios to their risk preferences. CvaR provides a more comprehensive measure of risk compared to CVaR since it considers the tail of the distribution (Agrawal, Koolen and Juneja, 2021). It reflects the severity of extreme losses and also useful for risk-averse investors who are concerned not only about the probability of losses but also about the magnitude of potential losses (Filippi, Guastaroba and Speranza, 2020). However, CvaR has some limitations where it can be computationally intensive, especially for large portfolios or complex risk models and relies on the accuracy of the underlying probability distribution assumption. CVaR may not be coherent with respect to some risk management principles (Lempinen, 2024).

When constructing a portfolio, investors aim to achieve an optimal balance between risk and return (Koumou, 2020). Traditional mean-variance optimization may not adequately capture extreme downside risks (Rigamonti, 2020). CVaR, on the other hand, explicitly considers these extreme scenarios, making it suitable for risk-averse investors or those concerned with tail risk (Wang *et al.*, 2022). In portfolio optimization using CVaR, the objective is to minimize the CVaR of the portfolio subject to certain constraints. The constraints could include exposure limits, regulatory requirements, or specific investment guidelines (Benati and Conde, 2022).

5.2.3 Mean Absolute Deviation (MAD) (Traditional)

Konno and Yamazaki (1991) introduced the mean absolute deviation (MAD) as a risk measure for portfolio assessment. This MAD approach utilizes a linear programming model that has been demonstrated to be equivalent to Markowitz's model, while offering enhanced computational efficiency (Silva, Alem and Carvalho, 2017). The MAD linear programming model can be formulated as the "MAD" model. By proposing MAD as an alternative to the variance-based risk measure in the mean-variance (MV) model, which assumes asset returns follow a normal distribution, Konno and Yamazaki aimed to improve portfolio risk measurement. The MAD model has been explored in portfolio optimization, with Zenios and Kang (1993) applying it to mortgage-backed securities. In portfolio optimization, the MAD model is a method for assessing risk and constructing efficient portfolios (Ararat *et al.*, 2024). Unlike other techniques that focus on variance or standard deviation, the MAD model considers the absolute deviations of returns from the portfolio's mean return (Lam, Lam and Jaaman, 2021).

MAD measures risk by calculating the average absolute deviation of individual asset returns from the mean return of the portfolio (Ararat *et al.*, 2024). This provides a measure of the dispersion or volatility of returns. In portfolio optimization, the objective is typically to maximize returns while minimizing risk. The MAD model aims to minimize the sum of absolute deviations from the mean return, subject to certain constraints such as minimum and maximum weight limits for individual assets, budget constraints, and possibly other constraints related to portfolio characteristics or investment objectives (Konno and Yamazaki, 1991).

The MAD model has several advantages, including simplicity of calculation and interpretation. It also provides a more robust measure of risk compared to variance or standard deviation, as it is less sensitive to extreme values or outliers in the data. While the MAD model offers simplicity and robustness, it may not capture all aspects of risk, particularly for investors with non-linear utility functions or preferences. Additionally, the MAD model assumes that returns are normally distributed, which may not always hold true in practice. Therefore, MAD model is a useful tool in portfolio optimization, providing a straightforward approach to measuring and managing risk in investment portfolios. However, like any model, it should be used in conjunction with

other techniques and considerations to make well-informed investment decisions (Chen *et al.*, 2021).

5.2.4 Enhanced Radial Basis Artificial Neural Network (K4-RANN) (ML)

Machine learning techniques extend beyond static optimization and incorporate predictive modelling, adaptive learning, and scenario analysis. By doing so, they can improve risk-adjusted returns, enhance diversification, and mitigate risks under both normal and stressed market conditions. Their flexibility also allows them to account for behavioural factors, transaction costs, and evolving market dynamics.

K4-RANN is a variant of the traditional Radial Basis Function Neural Network (RBFNN) specifically designed for portfolio optimization (Leon-Delgado *et al.*, 2018). RBFNN is a type of artificial neural network that uses radial basis functions as activation functions. It consists of three layers: input layer, hidden layer, and output layer. The input layer receives the features of the problem, such as historical returns, risk measures, and constraints. The hidden layer computes the radial basis functions, which transform the inputs into a higher-dimensional space. The output layer computes the linear combination of the transformed inputs to produce the output, which in portfolio optimization, could be the optimal portfolio weights (Kajiji and Dash, 2012; 2013).

K4-RANN introduces enhancements over traditional RBFNNs to better suit the complexities of portfolio optimization problems. One common enhancement is the inclusion of additional features or input variables that capture more aspects of the portfolio optimization problem, such as transaction costs, liquidity constraints, and market conditions. Another enhancement could involve optimizing the parameters of the radial basis functions and the network architecture using advanced optimization techniques like genetic algorithms or particle swarm optimization. K4-RANN may also incorporate regularization techniques to prevent overfitting, such as $L1$ or $L2$ regularization, dropout, or early stopping (Tao *et al.*, 2019).

In portfolio optimization, the K4-RANN model takes historical data of asset returns, risk measures, and other relevant features as input. The model aims to learn the relationship between these input features and the optimal portfolio weights that

maximize return or minimize risk, subject to constraints. During training, the K4-RANN adjusts its parameters using historical data to minimize a predefined loss function, such as mean squared error or mean absolute deviation between predicted and actual portfolio returns. Once trained, the K4-RANN can be used to predict optimal portfolio weights for new market conditions or investment scenarios (Dash and Kajiji, 2015).

K4-RANN offers several benefits for portfolio optimization which includes that it can capture complex nonlinear relationships between input features and optimal portfolio weights. It can handle various types of constraints and objectives, making it adaptable to different portfolio management strategies (Dash and Kajiji, 2020). By incorporating enhancements, such as additional input features and regularization techniques, K4-RANN can improve the accuracy and robustness of portfolio optimization models. Despite its benefits, K4-RANN may suffer from issues like overfitting, especially when dealing with limited or noisy data. Selecting appropriate input features and tuning model hyperparameters are crucial steps in building an effective K4-RANN model. Like other machine learning models, K4-RANN requires careful validation and testing to ensure its performance and generalization ability in real-world portfolio management scenarios. Overall, the K4-RANN model represents a sophisticated approach to portfolio optimization that leverages the power of artificial neural networks to address the complexities of modern investment decision-making (Dash and Kajiji, 2015).

5.2.5 Evolutionary (Genetic) Algorithms (ML)

Evolutionary algorithms, particularly genetic algorithms, have been widely used in portfolio optimization due to their ability to efficiently search large solution spaces for optimal or near-optimal solutions (Rifki and Ono, 2012). An algorithm of evolution, often known as a genetic algorithm, is an algorithm for optimization that is founded on Darwin's concept of evolution by means of natural selection (Eiben and Smith, 2015; Simon, 2013). The algorithm of evolution begins with an initial population of potential solutions, which are typically produced in a random fashion. During each iteration, novel potential solutions are produced by opting for a duo of solutions that exhibit superior performance within the population, amalgamating the knowledge derived from these two solutions (recombination), and introducing minor random disruptions (mutation) (Liagkouras and Metaxiotis, 2015). The freshly generated solutions take the place of the older, substandard-performing solutions within the population. Through an

iterative process that entails modifying previously identified well-performing solutions and selectively retaining superior ones, progressively enhanced solutions are unearthed over time.

Anagnostopoulos and Mamanis (2010) explained that in portfolio optimization, the objective function represents the measure used to evaluate the performance of a portfolio. Common objectives include maximizing return while minimizing risk (as measured by standard deviation or variance), maximizing the Sharpe ratio, or achieving a specific target return with minimum risk. Each potential portfolio is encoded as a chromosome in the genetic algorithm. The chromosome represents a potential solution and typically consists of a set of genes, each corresponding to a particular asset in the portfolio. The value of each gene may represent the weight of the corresponding asset in the portfolio.

Genetic algorithms offer several advantages for portfolio optimization, including their ability to handle nonlinear and non-convex objective functions, their flexibility in accommodating various constraints (for example, budget constraints, asset class constraints), and their capability to search a large solution space efficiently (Anadani *et al.*, 2023). However, they may suffer from issues such as premature convergence, which can be addressed through appropriate parameter tuning and population diversity maintenance strategies. Additionally, the performance of genetic algorithms in portfolio optimization depends on factors such as the choice of encoding scheme, selection mechanisms, and genetic operators (Anadani *et al.*, 2023).

5.2.6 Extreme Gradient Boosting (XG Boost) (ML)

XGBoost is a statistical framework that treats boosting as a numerical optimization problem, with the goal of minimizing the model's loss by adding weak learners through a gradient descent-like procedure (May, 2022). A hybrid approach combining XGBoost with an enhanced firefly algorithm has been proposed, where the firefly algorithm is used to optimize XGBoost's hyperparameters. XGBoost is a robust machine learning algorithm widely used across various fields, including finance and portfolio optimization. As part of the ensemble learning family, it is known for its efficiency, accuracy, and scalability (Hongjoong, 2021).

XGBoost can be employed to predict asset returns based on historical data. By training the model on features such as past prices, trading volumes, technical indicators, economic indicators, and sentiment analysis data, it can learn patterns and relationships within the data to forecast future returns (Elena, 2021). These predictions can then be used as inputs for optimizing the portfolio. Besides return prediction, XGBoost can also be used to forecast various risk metrics such as volatility, Value at Risk (VaR), or Conditional Value at Risk (CVaR). Accurate risk estimation is crucial in portfolio optimization to ensure that the selected assets align with the investor's risk tolerance and objectives (Gumelar *et al.*, 2020).

XGBoost can help in identifying the most influential features or factors affecting asset returns or risks. Feature selection is essential for building parsimonious and interpretable models, especially in the context of portfolio optimization, where identifying relevant factors can lead to more efficient allocation decisions (Wang, Cheng and Dong, 2023). XGBoost can directly contribute to the construction of the optimal portfolio by integrating return and risk predictions into the optimization process (May, 2022). Modern portfolio theory aims to maximize returns for a given level of risk or minimize risk for a given level of return. XGBoost predictions can guide the allocation of assets to achieve these objectives effectively (May, 2022).

XGBoost can aid in developing dynamic rebalancing strategies by continuously updating return and risk predictions as new data becomes available. This adaptive approach ensures that the portfolio remains aligned with changing market conditions and investor preferences over time (Jiang, Ji and Chang, 2020). Ensemble techniques, such as blending multiple XGBoost models or combining XGBoost with other machine learning algorithms, can further enhance portfolio optimization performance. By leveraging the strengths of different models, ensemble strategies can improve prediction accuracy and robustness. XGBoost can also be used to optimize portfolio transactions by considering transaction costs, market impact, and liquidity constraints. Incorporating these factors into the optimization process helps in achieving more realistic and feasible portfolio allocations (Naik and Albuquerque, 2022).

Overall, XGBoost offers a versatile framework for portfolio optimization, enabling investors to make informed decisions by leveraging advanced machine learning

techniques for return and risk prediction, feature selection, portfolio construction, rebalancing, and transaction cost optimization (Simar, 2023).

5.3 Review of Empirical Literature

There is a limited number of previous studies that have compared traditional and machine learning optimization techniques. The study area that has been extensively researched on is the comparison in the predictive power between traditional and machine learning techniques. According to the author's findings, the following are the few past studies related to this study.

Benhamou *et al.* (2023) conducted an extensive comparative analysis of traditional portfolio allocation methods and deep reinforcement learning (DRL), examining their principles, methodologies, and performance in optimizing returns relative to risk. The study found that a basic form of DRL converges with traditional methods, while a myopic agent driven by immediate rewards represents the dynamic counterpart to traditional approaches. To assess the performance of both methods, the study tested a portfolio problem consisting of 11 major assets, including four key equity indices: S&P 500, Eurostoxx 50, Nikkei 225, FTSE 100; four primary bond indices: US 10-year TNote, European Bund, UK 10-year Gilt, Japanese Government Bond 10-year; and three major commodities indices: Brent Oil, Gold, and Copper. The study period spanned from 2013 to 2023. The findings revealed that DRL showed notable improvements over traditional methods, consistently outperforming them with higher annual returns, superior risk-adjusted returns (Sharpe ratio), and more effective management of maximum drawdowns. These results underscored the potential of DRL to outperform conventional methods in portfolio allocation.

Wu *et al.* (2022) conducted a thorough systematic review on the application of machine learning methods and elements to optimize portfolios and allocate risk across various factors in the United States. The study compared the performance of different machine learning algorithms (deep learning neural networks and LSTM) with traditional portfolio optimization techniques (CVaR and MAD). The findings revealed that traditional portfolios with the least variance outperformed machine learning models in minimizing risk. The study utilized portfolio data from 49 industry-based U.S. firms, spanning 50 years of monthly individual stock returns (1970–2020). The results showed that the

portfolios optimized using traditional methods outperformed equally weighted portfolios in terms of standard deviation and Sharpe ratio.

Pretorius and van Zyl (2022) highlighted that while traditional portfolio management methods can integrate specific investor preferences, they depend on accurate forecasts of asset returns and covariances. In contrast, reinforcement learning (RL) methods do not rely on such explicit forecasts and are more suitable for multi-stage decision processes. To address the limitations in previous research, Pretorius and van Zyl (2022) conducted a study across three markets with varying economic trends from 2010 to 2020, using data from Yahoo Finance. By incorporating investor preferences into the RL models' reward functions, they offered a more comprehensive comparison against traditional methods in the risk-return space. They also modelled transaction costs more realistically by considering nonlinear changes driven by market volatility and trading volume. The findings suggested that RL methods may offer advantages over traditional convex mean-variance optimization techniques under certain market conditions. Specifically, RL models outperformed traditional single-period optimization (SPO) and multi-period optimization (MPO) models in upward trending markets, although only within certain risk limits. In sideways trending markets, the performance of SPO and MPO models was similar to that of the RL models for most of the tested excess risk range. The primary objective was to select three markets—one with an upward trend, one with a downward trend, and one with a stable or sideways trend. The secondary objective was to choose markets where these trend conditions persisted for long enough periods to encompass both the training and testing phases.

Zhang *et al.* (2018) examined the portfolio optimization problem within the Markowitz mean-variance framework in China. Their paper reviewed several extensions of Markowitz's MV model, aiming to guide researchers in identifying key trends and focus areas on portfolio selection studies, while also providing practitioners with valuable theoretical tools such as dynamic optimization, robust optimization, and fuzzy optimization, among others. Despite the comprehensive nature of the review, it had some limitations. However, the researchers believed the study covered the majority of the relevant literature in the finance field. The review highlighted that real-world financial markets are uncertain and complex, while theoretical models tend to be much simpler. As a result, they concluded that further work is needed to bridge this gap and

improve the practical applicability of these models, potentially through approaches like machine learning.

It is widely recognized by practitioners that individuals benefit from market participation; however, investors often demonstrate low levels of diversification, leading to portfolio performance that carries idiosyncratic risk. D'Acunto and Prabhala (2018) suggested that machine learning tools provide investors with the opportunity to achieve better outcomes by reducing unsystematic risk exposure through increased diversification. Additional research underscores how the cost-reduction features of machine learning, enhanced transparency, and its ability to bypass behavioural biases (Gargano and Rossi, 2018) and cognitive limitations associated with human financial advisors (Rossi and Utkus, 2021) contribute to improved returns for investors. Specifically, there is broad agreement among researchers regarding the characteristics of the typical investor: low income, low diversification, and high trading activity prior to adopting machine learning tools. Despite the proven effectiveness of these tools, many end-users remain hesitant to embrace automated machine learning strategies, as noted by Filiz *et al.* (2022).

Additionally, Asawa (2022) offers a review of various machine learning portfolio optimization techniques, such as clustering-based, support vector machine-based, and genetic algorithm-based methods. Al-Muharraqi (2023) presents a literature review on the use of machine learning in optimizing stock portfolios, focusing on the most widely used machine learning algorithms for predicting stock returns and the most common approaches for portfolio optimization. Meanwhile, Untwal (2020) utilizes machine learning for financial forecasting, incorporating both supervised and unsupervised learning through a data frame with multiple regression models and factor analysis. Together, these studies provide a thorough overview of how machine learning algorithms are applied in investment strategies, portfolio allocation, and risk management, emphasizing the most effective techniques for portfolio optimization and financial forecasting.

Cavalcante *et al.* (2016) offer a comprehensive perspective, providing an overview of the applications of computational intelligence to financial data from studies conducted between 2009 and 2015. In addition to the survey, the paper established a

standardized framework for developing these algorithms. Similarly, Bahrammirzaee (2010) reached comparable conclusions, suggesting that artificial intelligence algorithms generally demonstrate higher accuracy than traditional statistical methods. However, the study did not find evidence of outperformance on an absolute scale. To address this gap, the current study will conduct a comparative analysis, providing strong evidence of machine learning's outperformance over traditional stochastic models.

Several authors have reviewed portfolio optimization methods. In this area, some research focuses on defining different policy network configurations, with results that outperform various traditional baseline methods (Wang *et al.*, 2019; Liang *et al.*, 2018; Lu, 2017; Jiang and Liang, 2017; Wang *et al.*, 2021; Deng *et al.*, 2016; Cong *et al.*, 2021). Hu *et al.* (2015) explored the application of evolutionary computation in discovering rules for algorithmic trading in shares, based on a literature review in China. Ertenlice and Kalayci (2018) studied swarm intelligence techniques for portfolio optimization, discussing algorithms and their applications in Turkey. Additionally, Zhang, Gao, and Sun (2021) reviewed academic literature on the use of AI in finance, concluding that machine learning has significant potential to enhance investment performance, risk management, and operational efficiency within the financial industry.

5.4 Gaps in Existing Literature

From the empirical literature unpacked above, none of the studies were done in Africa or developing countries, previous studies focused on stock returns only and none of the studies explored portfolio optimization in ETFs. ETFs help both individual and companies to easily diversify their investments and get the benefits of greater economies of scale, such as reduced transaction costs.

There is still a gap in the literature on whether machine learning optimization techniques do really outperform the traditional optimization techniques in ETFs. ML techniques are solely noticeable in developed countries such as the United States, China, England, and Japan (Nundlall and Van Zyl, 2023; Wekwete, Kufakunesu and van Zyl, 2022; Sindi, 2023) mainly focusing stock returns predictions. Contrariwise, developing countries like South Africa exhibit fragmented research concerning the utilization of ML techniques and its comparability with conventional models in ETFs

portfolio optimization. This observation was made by Habbab and Kampouridis (2022) and Sindi (2023), who assert that the integration of ML techniques could substantially augment the discernment of investors, analysts, and policy makers.

Shivaprakash *et al.* (2022) contend that if developing countries procrastinate in embracing ML techniques, they may encounter competitive disadvantage, as early adopters of such techniques gain a competitive edge by enhancing their processes, products, and services. Delaying the implementation of ML may result in falling behind competitors who utilize this technology to make data-driven decisions and optimize their operations. Therefore, considering the utilization of ML techniques in portfolio optimization in unit trusts, can potentially alleviate the aforementioned challenges of huge losses under uncertainty conditions. From the given contextual background, this study aims to address the gap by employing three ML methodologies and contrasting their efficacy in optimizing a portfolio in comparison to traditional optimization techniques. The traditional techniques will be the Markowitz model, CVaR, and MAD for comparison with three categories of ML techniques, namely shallow learning (enhanced radial basis artificial neural network - K4-RANN), deep learning (Evolutionary Genetic Algorithms), and gradient boosting (XGBoost). Therefore, this study is the first of its kind to incorporate ETFs and also include developing countries like SA.

5.5 Data and Methodology

A detailed account is provided of the quantitative analysis that was carried out for the methodology employed as well as a description of the data set considered. The research is established on the optimization of ETF portfolios, which were optimized using daily returns observed in the market over a sample period of ten years. The optimization was done by utilizing both traditional optimization models and machine learning optimization algorithms, in accordance with the approach employed by professionals in the finance industry following D'Hondt *et al.* (2020). The creation of the optimized portfolio served as a basis for examining the developments that occur during periods of market distress.

5.5.1 Data and Sampling

This research follows the methodology and sample period employed by Sindi (2023). The dataset covers a time period of 10 years, from 2014 to 2024, with a focus on daily price of historical data from a sample of 33 listed ETFs. The ETF daily returns data were derived from historical daily closing prices sourced from the IRESS database, in accordance with Kunjal and Peerbhai (2021). To prevent survivorship bias, ETFs that were delisted during the sample period were also included in the analysis, as recommended by Kunjal (2022), Tsalikis (2020), and Peyper (2014), addressing concerns related to small sample sizes. As a result, to enhance the reliability of the findings, each ETF in the sample was required to have been registered and listed on the JSE for a minimum of one year, resulting in a final sample of 33 JSE-listed ETFs (see Appendix 3). To add on, delisted ETFs were also included to retain performance data during the sample period to avoid survivorship bias. Therefore, the forward-filling and interpolation in R-Studio was used to resolve any gaps in the data that may occur due to incomplete records (Gomes, Guimarães and Silva, 2021; Tunkiel, Sui and Wiktorski, 2021).

ETF returns are calculated using Net Asset Values or daily prices which gives NAV returns or price returns respectively (Hill, Nadig and Hougan, 2015). ETF daily prices were used to calculate daily returns, not Net Asset Values (NAVs) because ETF prices represent the actual prices at which an ETF is traded on the market and reflect real-time investor sentiment, liquidity, and market demand (Buetow and Hanke, 2024). These are the prices investors use to buy or sell ETFs, making them more relevant for portfolio optimization. The NAV return of an ETF can be calculated based on the value of its underlying assets at the end of the trading day. NAV returns are useful for assessing the intrinsic value of an ETF, but they do not capture intraday price movements, which are essential for return calculations and risk assessments in portfolio optimization (Hu, Liang and Yuan, 2024). Therefore, portfolio optimization requires accurate return data since investors transact based on market prices (not NAVs). Hence, using ETF prices ensures that the calculated returns reflect the actual gains or losses experienced by investors (Bossaerts *et al.*, 2024)

Daily returns were annualized to annual simple mean average returns as opposed to annual compounded returns. For this study, the primary objective follows ETF portfolio

optimization rather than focusing on growth of portfolios, therefore mean average annualization method was applied following Bodie, Kane and Marcus (2014) and Kanuri (2016). The emphasis is on risk-return trade-offs, diversification, and performance metrics rather than the exact growth trajectory of the ETF portfolios. This approach is supported by Dimson, Marsh and Staunton (2009) who discussed the use of arithmetic averages in return estimation and portfolio analysis, particularly in the context of historical data and acknowledges that simple mean averages are more suitable for portfolio optimization.

5.5.2 Preliminary Tests

To summarize the statistical properties of the data, providing insight into the underlying characteristics of ETF returns. Descriptive statistics with mean, median, standard deviation, skewness, kurtosis and minimum and maximum were done. Stationarity tests were also performed to determine whether the ETF return series are stationary (mean, variance, and covariance are constant over time), a crucial assumption for most econometric models. Cointegration analysis was also done to identify whether a long-term equilibrium relationship exists between ETF returns, which can impact the selection of portfolio optimization techniques.

5.5.3 Model Specification and Software

Three traditional models were employed, namely the Markowitz model, CVaR, and MAD. Additionally, three models from three different families of machine-learning techniques were utilised. These families include shallow learning, an enhanced radial basis artificial neural network called K4-RANN. Furthermore, deep learning techniques with the Evolutionary (Genetic) Algorithm were employed. Lastly, gradient boosting with extreme gradient boosting was utilised. The previous section provided a brief discussion of each of these six models.

Furthermore, this section delved deeper into the econometric aspects to address the second research question. All six models were run in R studio software, although another competitive software was Python. However, *R studio* was chosen due to its better performance and other advantages over Python, which are discussed below. *R studio* is much better for most finance applications that do actual analyses, as it requires large data sets and multiple levels of analysis (Okoye and Hosseini, 2024).

Base *R studio* is powerful, but some libraries substantially add to its power. Not to mention, the visualisation capabilities of *R studio* are beneficial to finance academics as lots of pictures are needed. Therefore, Python is a better choice for building an analysis application or website as it is an end-to-end language (Okoye and Hosseini, 2024). Both perform data analysis, but Python is more powerful for building application-like solutions and R for the actual analysis and visualisation.

Furthermore, in the context of data processing, although Python's capabilities for data collection are only marginally less efficient than those of R, the distinction becomes more pronounced in data manipulation and analysis, but R demonstrates significant advantages. As a language specifically designed for statistical computing, R excels in performing financial analyses, which rely on statistical methodologies. Tasks such as linear regression, Monte Carlo simulations, and advanced data manipulation are often more streamlined and efficient in R compared to Python.

5.5.2.1 The Markowitz Model

The Mean Variance model makes use of parameters denoting the return, variance, and covariance of the chosen stocks (Lai, Xing and Chen, 2011). The Markowitz model approach places emphasis on the aspect of diversification in investment to optimize investment returns through the formation of portfolios that are considered efficient (Balqis, Subiyanto and Supian, 2021). There exist various criteria for the formation of an efficient portfolio, namely: a portfolio that presents a higher rate of return at given risk and portfolios that offer reduced risk while maintaining the same rate of return.

Based on a study conducted by Balqis, Subiyanto and Supian (2021), the recommended approach for constructing an optimal portfolio involves employing the Markowitz method, which is articulated in the objective function as follows:

$$\text{Min } (\sigma^2 = \min \sum_{i=1}^n w_i^2 \sigma_i^2 + \sum_{i=1}^n \sum_{j=1}^n w_i w_j \sigma_{ij}) \quad , i \neq j \quad (5.1)$$

With constraints:

1. $\sum_{i=1}^n w_i = 1$
2. $w_i \geq 0$ for $i = 1, \dots, n$
3. $\sum_{i=1}^n w_i R_i = R_p$

Where $\sigma_i^2 = E[(R_i - E(R_i))^2]$ is the variance of risk on asset i and $\sigma_{ij} = E[(R_i - E(R_i))(R_j - E(R_j))]$ is the covariance between assets i and j .

The optimization process involves both the objective function and the constraints. The objective is to maximize the portfolio's expected return for a given level of risk or, alternatively, to minimize risk for a specified level of expected return. This can be expressed as a quadratic optimization problem, as shown in equation 5.2. The optimization may include constraints such as budget constraints (the sum of weights must equal 1), non-negativity constraints (weights must be non-negative), and potentially additional constraints based on investor preferences or regulatory requirements. By applying the Markowitz Model to ETFs, investors can construct diversified portfolios that seek to achieve an optimal balance between risk and return, utilizing the advantages of ETFs to strengthen their investment strategy.

5.5.2.2 Conditional Value at Risk (CvaR)

Conditional Value at Risk (CVaR), also known as Expected Shortfall (ES), is a risk measure that assesses the tail risk of a portfolio. It quantifies the expected loss in the worst-case scenarios beyond a specified confidence level (Radak *et al.*, 2024). In portfolio optimization, CVaR is commonly used to minimize potential extreme losses when constructing an investment portfolio, including Exchange Traded Funds. Key concepts within CVaR include Value at Risk (VaR) and Conditional Value at Risk. VaR measures the maximum potential loss over a given time period for a specified confidence interval (e.g., 95% or 99%) (Molina, 2021). However, Haslbeck, Bringmann and Waldorp (2021) noted that VaR does not indicate the magnitude of losses beyond this threshold. CVaR addresses this limitation by evaluating the average loss that occurs beyond the VaR threshold, offering a more complete perspective on tail risk.

The goal of the objective function (5.3) is to minimize the Conditional Value at Risk (CVaR). Constraints stipulate that the sum of the investment types must equal one. The boundaries (5.8) indicate the decision variables of the problem.

$$\text{Min CVaR} = \sum_{i=1}^n x_i [\alpha r_{i1} - (1 + \alpha) r_{i2}] \quad (5.2)$$

With constraints:

$$\sum_{i=1}^n x_i = 1$$

$$x_i = \{0, 1\}, i = 1, 2, \dots, n$$

Where:

x_i are the weights of assets in the portfolio

r_i are the returns of an asset i .

Gambeta and Kwon (2020) noted that CVaR in portfolio optimization uses objective function, constraints and risk-return trade-off. In CVaR-based portfolio optimization, the objective is often to minimize CVaR subject to certain constraints, such as a desired level of return or budget constraints. Constraints can include budget constraints (the sum of weights equals 1), non-negativity constraints (weights are non-negative), and bounds on individual asset weights. CVaR optimization helps balance the trade-off between risk and return by focusing on reducing extreme losses rather than just overall portfolio variance. Therefore, by incorporating CVaR into portfolio optimization, investors can create ETF portfolios that are better equipped to handle extreme market events and reduce potential downside risks.

5.5.2.3 Mean Absolute Deviation (MAD)

Gunjan and Bhattacharyya (2023) noted that the utilization of the model offers the benefits of decreased computational intricacy and delivers optimal outcomes. Furthermore, it has been observed that the mean absolute deviation surpasses the mean-variance Markowitz model (Gunjan and Bhattacharyya, 2023). Consequently, this investigation will employ the aforementioned model in the context of asset allocation and portfolio optimization within ETFs. Instead of using variance as the risk measure, it uses the mean absolute deviation, which is the average of the absolute differences between each return and the mean return. This approach has some advantages which includes robustness to outliers (MAD is less sensitive to extreme values compared to variance) and simplicity (the calculation is straightforward and interpretable).

In terms of mathematical representation, the objective function was denoted by Konno and Yamazaki (1991) as follows:

$$\text{Minimize } w(x) = E \left| \sum_{j=1}^n R_j x_j - E \sum_{j=1}^n R_j x_j \right| \quad (5.3)$$

Subject to $\sum_{j=1}^n E I R_j I x_j \geq \rho M_0$, and M_0

$0 \leq x_j \leq u_j$, $j = 1, 2, \dots, n$ where R_j is the return of asset j , x_j is the amount invested in asset j , ρ is a parameter representing the minimal rate of return required by an investor, M_0 is the total amount of fund and u_j is the maximum amount of money which can be invested in an asset j .

Therefore, by using the MAD model, investors can construct a portfolio of ETFs that balances return expectations with a more intuitive and robust measure of risk. This approach can be particularly useful in markets with volatile assets or when a simple, yet effective risk measure is preferred.

5.5.2.4 Enhanced Radial Basis Artificial Neural Network (K4-RANN)

The radial basis function artificial neural network (RANN) is a shallow-learning nonlinear ANN, characterized by having a single hidden layer between the input and output layers. This structure classifies it as a shallow learning network. In such networks, the output layer generates its output as a linear combination of the radial basis functions and neuron parameters. The K4-RANN model is a hybrid machine learning technique suitable for portfolio optimization in ETFs. Within the framework of the generalized RANN method, the optimal weighting values, often represented as w_j , are determined by applying a supervised least-squares approach to a specific subset of the data, referred to as the training set (Dash and Kajiji, 2020).

Additionally, Dash and Kajiji (2020) highlighted that certain processes involve complex nonlinear data, which poses challenges to the assumptions underlying traditional linear regression techniques. The literature suggests that a multivariate feedforward radial basis function (RBF) artificial neural network (RANN) is an effective method for analysing nonlinear relationships involving correlated variables (Leon-Delgado *et al.*, 2018). In general, RANN mapping relies on partial information or learning through examples. As shown in Fig. 5.1, the generalized RANN is a three-layer ($J1 \rightarrow J2 \rightarrow J3$) feedforward neural network.

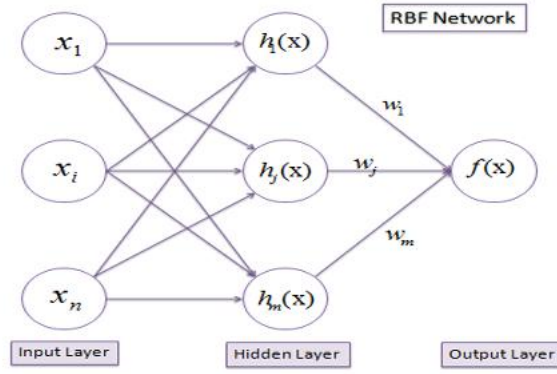


Figure 5.2 Feedforward RANN Network

Source: Dash and Kajiji (2020)

The overview of K4-RANN includes three parts; firstly, K-Nearest Neighbor (KNN) which is a non-parametric method used for classification and regression. It works by finding a predefined number of training samples closest in distance to the point of interest and making predictions based on these neighbours. Secondly, Randomized Averaging which involves combining multiple models to reduce variance and improve prediction accuracy. Randomized averaging techniques, such as bootstrap aggregating (bagging), are often used to enhance the robustness of predictions. Thirdly, Neural Network (NNs) are a class of machine learning models inspired by the human brain. They are particularly powerful for capturing complex patterns in data through layers of interconnected neurons. Therefore, the combination of these three methods gives the K4-RANN model. The optimization process involves selecting a mix of ETFs that maximizes expected return for a given level of risk. The K4-RANN model can be used to forecast returns and risks, which are then fed into a portfolio optimization algorithm (for example, mean-variance optimization, genetic algorithms) to construct an optimal portfolio.

The supervised learning function can be described as $y = f(x)$, where the output vector y is a function of the input vector x , which consists of p inputs (Dash and Kajiji, 2020). This function can be expressed as follows:

$$f(x_i) = \sum_{j=1}^m w_j h_j(x) \quad (5.4)$$

where, m is the number of basic functions (centres), h is the number of hidden units, w is the weight vector, and $i = 1 \dots p$ where p is the number of input vectors. As shown in equation (10) the K4-RANN minimizes a modified SSE or cost function C .

$$C = \frac{\text{argmin}_k}{k} (\sum_{i=1}^p (y - f(x_i I k))^2 + \sum_{j=1}^m k_j w_j^2) \quad (5.5)$$

Dash and Kajiji (2020) explained that the outcome of implementing the K4-RANN yields the retrieval of a collection of weights (w_j) (in such a way that the sum of squared errors is minimized) while simultaneously optimizing the accuracy of the anticipated suitability (i.e., smoothness). The estimated weights bear resemblance to the parameters of a nonlinear least-squares regression. When employed in the context of the networked production function, for every element j in the set N (representing securities in the investor's portfolio), the procedure establishes a relationship between the production of daily returns r_j , and the j -ETFs.

Implementing K4-RANN in ETF portfolio optimization includes the use of the trained K4-RANN model to predict future returns and risks of the selected ETFs. Followed by feeding the predictions into a portfolio optimization algorithm to determine the optimal weights for each ETF in the portfolio. Lastly, back testing the optimized portfolio on historical data to evaluate its performance and adjust the model parameters as needed. There are advantages of K4-RANN in ETF portfolio optimization includes leveraging the strengths of KNN for identifying similar assets, neural networks for capturing complex patterns, and randomized averaging for reducing prediction variance. It produces more robust and stable predictions due to the ensemble nature of the model and also can adapt to changing market conditions by retraining on new data. However, it poses some challenges of complexity and computationally intensive. The model can be complex to implement and requires careful tuning of hyperparameters and training neural networks and performing randomized averaging can be computationally intensive. Therefore, by integrating the K4-RANN model into the portfolio optimization process, investors can potentially enhance their ability to select and allocate ETFs in a manner that balances risk and return effectively.

5.5.2.5 Evolutionary Genetic Algorithm (GA)

Lazo *et al.* (2000) postulated that Evolutionary Genetic Algorithms are inspired by the process of natural selection and are used to find approximate solutions to optimization and search problems. They are particularly useful in portfolio optimization due to their ability to handle complex, non-linear, and multi-modal landscapes, making them suitable for finding optimal allocations among different financial assets, such as ETFs. This study utilized a GA to choose a predetermined number of assets. The objective is to optimize the evaluation function in order to determine the assets that exhibit a more promising return, a reduced risk, and a lower correlation with the remaining assets (Lazo *et al.*, 2000). Consequently, the evaluation function is defined by the subsequent expression.

$$Z = \max \prod_{i=1}^n \prod_{j=1}^n x_i r_i (1 - \sigma_i)(1 - \rho_{ij}) \quad (5.6)$$

$$\text{Constraint; } \sum_{i=1}^n x_i = m ,$$

$$x_i = 0, 1$$

$$\text{Where: } x_i: \begin{cases} 1 & \text{if asset } i \text{ present} \\ 0 & \text{if asset } i \text{ absent} \end{cases}$$

r_i : Return on the ETF.

σ_i : inherent risk of the ETF.

ρ_{ij} : Correlation coefficient of the ETF.

m : Number of assets which should hold in the ETF portfolio.

n : Number of all available ETFs

Implementing GA for ETF portfolio optimization involves defining the objective function thus constructing a fitness function that reflects the optimization goals, such as maximizing the Sharpe ratio or minimizing the risk for a given return level. Followed by initializing parameters such as population size, crossover rate, mutation rate, and the number of generations as we as defining the chromosome structure to represent ETF weights. Generate an initial population of random portfolios and ensure the sum of ETF weights in each portfolio equals 1 (or 100%). Lastly, calculate the fitness of each portfolio using historical returns and covariance data of the ETFs.

Li and Shi (2022) explained the key concepts in portfolio optimization using Genetic Algorithms involves chromosome representation where in the context of portfolio optimization, a chromosome can represent a potential solution, i.e., a portfolio of ETFs. Each gene in the chromosome corresponds to the weight of an ETF in the portfolio. Another concept of population initialization which a population of potential solutions (portfolios) is initialized. Each individual in the population represents a different allocation of ETFs. Lastly, the fitness function which evaluates how well each portfolio meets the optimization criteria where the common objectives include: maximizing return, minimizing risk (e.g., variance or standard deviation) and maximizing the Sharpe ratio (risk-adjusted return).

Therefore, using Genetic Algorithms for ETF portfolio optimization provides a robust method to navigate the complex landscape of financial markets. By mimicking the process of natural selection, GAs can efficiently search for optimal portfolio allocations that balance risk and return, offering a valuable tool for investors aiming to maximize their investment performance.

5.5.2.6 XG Boost (Extreme Gradient Boosting)

XGBoost is an advanced version of gradient boosting optimized for speed and performance. It has gained popularity due to its efficiency, accuracy, and ability to manage large datasets. XGBoost is built upon a collection of weak classifier decision trees and focuses on training new decision trees to correct the errors made by previous ones (Dey *et al.*, 2016). This process is carried out using a gradient descent-like approach, with each tree being trained sequentially (Chen and Guestrin, 2016). Initially, a regression function is created and fitted to the dataset, resulting in errors known as residual errors. These residual errors are then used to create another regression function to improve the model. By combining the previous and current regression functions, errors are minimized, and the regression function becomes progressively more complex, leading to a significant reduction in the root mean squared error. The following regularized objective function is employed to train the model.

$$\min(\sum_{i=1}^n l(y_i + \hat{y}_i)) + \sum_k \Omega(f_k) \quad (5.7)$$

where $l(y_i + \hat{y}_i)$ refers to the loss function and $\Omega(f_k)$ refers to the regularization term that can be calculated as follows:

$$\Omega(f_k) = \gamma T + \frac{1}{2} \lambda \|\omega\|^2 \quad (5.8)$$

Where γ is the L1 regularization coefficient, T is number of leaves, λ is the L2 regularization coefficient, ω is the leaf weight.

Once the predictions are made, traditional optimization algorithms (e.g., Mean-Variance Optimization, Genetic Algorithms) can be combined with XGBoost to select the optimal portfolio. The predicted returns and risks serve as inputs to these algorithms. An objective function is defined (e.g., maximize return for a given risk level) and optimization algorithms are used to select the optimal ETF portfolio based on the predicted returns and risks. Lastly, back testing is applied to test the optimized portfolio on historical data to evaluate its performance.

5.5.3 Performance Metrics

The best performing model was selected based on performance metrics, mean portfolio returns, portfolio standard deviation and the Sharpe ratio. This study follows the previous literature (Al-Muharraqi and Messaadia, 2023; Bacon, 2023; Lobato, Rodríguez and Romero, 2021), they utilised the above-mentioned metrics. The higher the Sharpe ratio will imply higher performance of the ETF optimized portfolio. The lower the standard deviation of an ETF optimized portfolio implies that it carries low risk on the returns realised. Lastly, the higher the mean ETF portfolio returns implies better performance. Therefore, the comparison between the selected traditional and machine learning optimization models' performance was analysed based on these three metrics.

5.5.3.1 Mean Portfolio Returns

Bacon (2023) explained that portfolio return is defined as the sum of the products of each asset's return and its respective weight in the overall portfolio. It reflects the return of the entire portfolio, rather than just the return of an individual asset.

$$R_p = \sum_{i=1}^n w_i r_i \quad (5.9)$$

Where;

w_i is the weight of each ETF in the portfolio

r_i is the annualized return of each ETF in the portfolio

5.5.3.2 Portfolio standard deviation

Clarke, De Silva, and Thorley (2013) stated that portfolio standard deviation represents the volatility of the portfolio, calculated using three key factors: the standard deviation of each asset in the portfolio, the weight of each asset within the portfolio, and the correlation between each pair of assets.

$$\text{Variance of Portfolio} = w_1^2 * Var_1 + w_2^2 * Var_2 + \dots + w_n^2 * Var_n + 2 * w_1 * w_2 * Cov_{12} + \dots + 2 * w_1 * w_n * Cov_{1n} + 2 * w_2 * w_3 * Cov_{23} + \dots + 2 * w_{n-1} * w_n * Cov_{n-1n} \quad (5.10)$$

Portfolio standard deviation measures the volatility of a portfolio, calculated by considering the correlation between asset pairs, individual asset standard deviations, and their respective weights. To assess portfolio risk, it's important to analyse the standard deviations of the assets, their weightings, and correlations, as these factors collectively offer insights into potential fluctuations. A portfolio with a high standard deviation indicates increased risk and more unpredictable returns. When comparing portfolios, standard deviation helps highlight differences in stability and volatility. While a higher standard deviation suggests greater risk, it may also be associated with higher potential returns. Investors assess standard deviation in conjunction with potential rewards to make well-informed decisions (Clarke, De Silva and Thorley, 2013).

5.5.3.3 Sharpe Ratio

The Sharpe ratio (Sharpe, 1966) evaluates the risk premium of an investment portfolio in relation to its overall risk. The risk premium, or excess return, is the difference between the portfolio's return and the risk-free rate, often represented by the yield on a Treasury security. The total risk is measured as the standard deviation of the portfolio's returns. The numerator indicates the reward for taking on risk above the risk-free rate, while the denominator captures the portfolio's return volatility.

$$\text{Sharpe ratio} = \frac{R_p - R_f}{\sigma_p} \quad (5.11)$$

Where;

R_p = return of portfolio

R_f = risk-free rate

σ_p = standard deviation of the portfolio's excess return

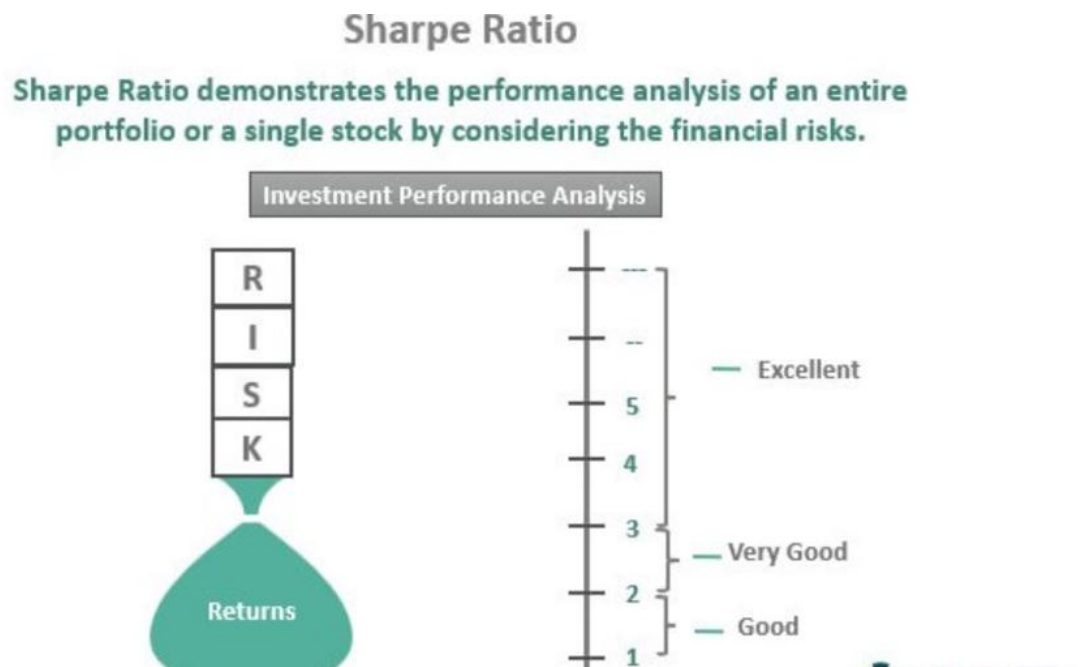


Figure 5.3 Sharpe Ratio for Portfolio Performance Assessment

Source: Hatami, Ali and Volkman (2022)

5.6 Results and Discussion

This section shows the explanation of results of each of the optimization models separately and then followed by the comparison explanation using the performance metrics utilised. Lastly is the discussion of the results relating to the empirical literature as well as the implications

5.6.1 Preliminary Tests Results

Before conducting portfolio optimization, it is crucial to ensure that the underlying data meets the necessary statistical assumptions (Pfaff, 2016). This study carried out a series of preliminary tests, including descriptive statistics, stationarity tests, and cointegration analysis, to assess the properties of the dataset and the relationships between variables. These tests play a vital role in providing insights into data characteristics, ensuring methodological rigor, and establishing a foundation for robust optimization results. Table 5.1 provides the descriptive statistics of the 33 ETFs used in the analysis. The dataset summarizes the statistical properties of daily returns for various Exchange-Traded Funds. Most ETFs exhibit small mean and median returns, often close to zero, indicating low average daily performance, though some, such as NGPLD (0.082) and ETF500 (0.035), show higher average returns. Volatility, measured by standard deviation, varies significantly across ETFs, with higher-risk funds like CTOP50 (0.093) and NFSWIX (0.097) contrasting with the lower-risk SYGUK (0.016). Kurtosis values suggest that most ETFs have light tails and fewer extreme returns, but outliers such as STXCAP (5.640) and ETFGRE (7.394) exhibit significant deviations, indicating extreme values. Skewness is generally near zero, implying mild asymmetry; however, some ETFs like ETFRHO (1.372) and ETF500 (-0.813) display notable skewness. Extreme return values highlight substantial risks, with notable lows (e.g., STXFIN at -0.838) and highs (e.g., STXCAP at 0.0872). Overall, ETFs like NGPLD and ETF500 stand out for better returns, while high kurtosis and skewness in certain funds suggest potential for unusual performance or risk, providing valuable insights for portfolio optimization and risk assessment.

Table 5.1 Descriptive Statistics Results

ETF	Mean	Median	S.D	Kurtosis	Skewness	Minimum	Maximum
NGPLD	0,082	0,072	0,088	1,653	0,036	-0,345	0,0827
STXSWX	-0,001	0,000	0,083	3,295	-0,104	-0,242	0,0979
STXFIN	-0,006	0,026	0,068	1,909	-0,152	-0,838	0,0511
NGPLT	0,004	0,002	0,019	1,229	0,036	-0,110	0,0137
SYGEU	0,010	0,004	0,027	1,436	0,002	-0,979	0,0913
SYGUK	0,006	0,003	0,016	2,356	-0,283	-0,543	0,0574
STXRES	0,015	0,033	0,055	1,327	-0,240	-0,827	0,038

ETFSAP	-0,030	-0,018	0,064	3,974	-0,166	-0,801	0,0660
ETFGLD	0,025	0,005	0,032	1,965	0,551	-0,404	0,0370
ETFPLT	0,004	0,000	0,024	3,786	-0,081	-0,285	0,0193
SYGWD	0,017	0,028	0,040	1,348	0,046	-0,489	0,092
CSPROP	-0,033	-0,025	0,082	2,346	-0,168	-0,676	0,0906
GLD	0,024	0,011	0,029	1,848	0,495	-0,381	0,0436
STXSHA	-0,003	0,000	0,089	1,483	-0,514	-0,058	0,0926
FNBT40	0,003	0,013	0,059	3,151	-0,088	-0,078	0,0131
STXDIV	-0,002	0,000	0,003	1,185	-0,075	-0,957	0,0245
CTOP50	0,001	0,000	0,093	1,996	-0,376	-0,887	0,0702
ETFRHO	0,056	0,087	0,021	3,640	1,372	-0,478	0,0872
GLPROP	0,009	0,000	0,029	1,518	-0,036	-0,049	0,0727
AMIB50	0,002	0,000	0,089	3,526	-0,178	-0,384	0,0873
STXQUA	-0,004	0,000	0,082	3,464	-0,051	-0,013	0,0964
SYGP	0,009	0,023	0,076	3,962	-0,238	-0,700	0,0641
SYGT40	-0,010	0,008	0,076	2,151	-0,404	-0,112	0,0755
ETF500	0,035	0,020	0,075	4,141	-0,813	-0,592	0,0903
ETFGGB	0,028	0,012	0,020	1,403	0,421	-0,230	0,0403
ETFGRE	0,028	0,060	0,057	7,394	-0,080	-0,421	0,0803
STXVEQ	-0,019	0,000	0,083	4,747	-0,241	-0,061	0,0797
SYGSW4	-0,012	0,000	0,068	3,213	0,463	-0,188	0,0305
NFSWIX	0,005	0,000	0,097	1,462	0,041	-0,558	0,028
ETFBND	0,014	0,034	0,084	3,059	-0,673	-0,794	0,0679
STXDEQ	0,007	0,000	0,090	1,507	-0,616	-0,709	0,0445
STXCAP	0,015	0,087	0,002	5,640	1,372	-0,478	0,0872
ETFEMA	0,009	0,000	0,029	1,518	-0,036	-0,049	0,0727

Table 5.2 shows the stationarity analysis results conducted on various ETFs using the Augmented Dickey-Fuller (ADF), Phillips-Perron (PP), and KPSS tests reveals that all ETFs are stationary at their levels, classified as integrated of order zero ($I(0)$). The ADF and PP tests generally confirm stationarity, with most ETFs showing significance

at the 1%, 5%, or 10% levels. While the KPSS test occasionally produces higher values, it aligns with the overall classification of $I(0)$. Notable ETFs, such as STXSHA, STXDEQ, and STXCAP, exhibit robust stationarity with strong significance in both the ADF and PP tests. Conversely, ETFs like SYGSW4, SYGUK, and SYGT40 display weaker stationarity, significant only at the 10% level, but still meet the criteria for $I(0)$. These results indicate that all ETFs are suitable for further econometric modelling without requiring differencing.

Table 5.2 Stationarity and Cointegration Tests Results

ETF	ADF	PP	KPSS	Order of Integration
NGPLD	-3.36**	-3.15**	1.60	$I(0)$
STXSWX	-1.85*	-1.79*	4.03	$I(0)$
STXFIN	-3.00**	-2.91**	0.58	$I(0)$
NGPLT	-2.50*	-2.33*	2.94	$I(0)$
SYGEU	-1.45*	-1.29**	5.21	$I(0)$
SYGUK	-1.05*	-0.80*	4.65	$I(0)$
STXRES	-2.11*	-1.78**	3.29	$I(0)$
ETFSAP	-3.05**	-2.72*	0.90	$I(0)$
ETFGLD	-3.21**	-3.12**	1.14	$I(0)$
ETFPLT	-3.59***	-2.98**	2.22	$I(0)$
SYGWD	-1.53*	-1.49*	4.66	$I(0)$
CSPROP	-1.14*	-1.05*	3.89	$I(0)$
GLD	-2.21*	-2.08**	3.08	$I(0)$
STXSHA	-3.47***	3.28***	1.92	$I(0)$
FNBT40	-2.01*	-2.08**	2.89	$I(0)$
STXDIV	-1.84*	-1.63*	4.21	$I(0)$
CTOP50	-2.02*	-1.98*	2.47	$I(0)$
ETFRHO	-3.66**	-3.75**	3.60	$I(0)$
GLPROP	-3.85***	-2.79*	4.03	$I(0)$
AMIB50	-2.00**	-1.91**	0.98	$I(0)$
STXQUA	-1.50*	-3.33**	2.74	$I(0)$
SYGP	-2.45*	-1.29*	3.21	$I(0)$
SYGT40	-2.05*	-1.80*	4.69	$I(0)$
ETF500	-3.11**	-1.38*	2.29	$I(0)$
ETFGGB	-3.75**	-3.72*	1.90	$I(0)$
ETFGRE	-1.21**	-3.12**	2.14	$I(0)$
STXVEQ	-2.59***	-3.78**	2.72	$I(0)$
SYGSW4	-1.36**	-2.49**	3.66	$I(0)$
NFSWIX	-2.85*	-1.75*	2.89	$I(0)$
ETFBND	-2.00**	-2.18*	2.08	$I(0)$
STXDEQ	-2.50**	3.48***	1.72	$I(0)$
STXCAP	-3.45***	-2.68**	2.19	$I(0)$
ETFEMA	-2.05**	-1.83*	4.71	$I(0)$

***, **, * Indicates significance at 1%, 5% and 10% levels, respectively.

Gujarati and Porter (2009), Enders (2014), Engle and Granger (1987) Harris and Sollis (2003) and Enders (2014) support that If the data is stationary at level (i.e., it is $I(0)$), then there is no need to test for cointegration, as cointegration is a concept that applies to non-stationary series that become stationary upon differencing (i.e., $I(1)$). Cointegration tests are used to determine whether a long-term equilibrium relationship exists among $I(1)$ variables.

5.6.2 Traditional Portfolio Optimization Results

The results presented in table 5.3 reflect the application of these methods to a dataset of ETFs, analysing their effectiveness in achieving risk-adjusted returns, risk reduction, and diversification. These findings provide a baseline for comparing the performance of machine learning-based portfolio optimization techniques. Table 5.3 shows the optimization results of utilising Markowitz Model (MM) to assign weights to the 33 selected ETFs and these weights (Weights_MM) represent the percentage of the overall portfolio invested in each ETF to obtain the best risk-reward ratio. In this table (Weight_MM_select) determines whether an ETF was included in the final portfolio denoted by the “yes” or “no” and the optimized portfolio had a final of 15 ETFs from the initial 33 ETFs. This is the common goal in portfolio optimization: to spread risk across which is also called diversification.

The weights vary significantly with some ETFs receiving high allocations like Satrix Defensive EQ ETF (STXDEQ) has 0.466 and the lowest allocated to Satrix Quality SA ETF (STXQUA) has 0.002. The variation reflects the Markowitz Model’s optimization process which aims at balancing the expected returns with risk (standard deviation) of each ETF. The weights allocated are meant to maximise the ETF portfolio return for a given level of risk or to minimize risk for the same expected portfolio return.

Given these results, it implies that the larger weight allocated to Satrix Defensive EQ ETF (STXDEQ) indicates a desire for stability and lower risk while still aiming for the highest return. The same applies to ETFs like Satrix SWIX Top 40 Portf (STXSWX) and 1InvestGold ETF (ETFGLD) were allocated zero weight meaning that the model did not consider them ideal for inclusion in this specific portfolio. This may imply that risk -return profiles do not align well with the overall optimization goal and restrictions.

Table 5.3 Markowitz Model (MM) Portfolio Optimisation Results

ETF	Weights_ MM	Weight_ MM _select (15 Selected)	Average returns
New Gold Palladium ETF (NGPLD)	0,022	Yes	0,0009
Satrix SWIX Top 40 Portf (STXSWX)	0	No	0
Satrix FINI Portfolio (STXFIN)	0,09	Yes	0,0372
New Gold Platinum ETF (NGPLT)	0,086	Yes	0,0308
Sygnia Itrix Eurostoxx50 (SYGEU)	0	No	0
Sygnia Itrix FTSE100 (SYGUK)	0,014	Yes	0,0066
Satrix RESI Portfolio (STXRES)	0,004	Yes	0,0498
1Invest SA Property ETF (ETFSAP)	0	No	0
1InvestGold ETF (ETFGLD)	0	No	0
1InvestPlatinum ETF (ETFPLT)	0	No	0
Sygnia Itrix MSCI World (SYGWD)	0,006	Yes	-0,0332
10X SAProp Income (CSPROP)	0	No	0
New Gold Issuer Ltd (GLD)	0,108	Yes	0,0093
Satrix Shariah T40 ETF (STXSHA)	0	No	0
FNB TOP 40 ETF (FNBT40)	0,008	Yes	0,0174
Satrix DIVI Plus Portfol (STXDIV)	0	No	0
10X Top 50 (CTOP50)	0,116	Yes	0,0138
1InvestRhodium ETF (ETFRHO)	0	No	0
10X Global Prop (GLPROP)	0	No	0
EasyETF AMI Big50 ETF (AMIB50)	0	No	0
Satrix Quality SA ETF (STXQUA)	0,002	Yes	0,0001
Sygnia Itrix Gbl Pro ETF (SYGP)	0	No	0
Sygnia Itrix Top 40 ETF (SYGT40)	0,006	Yes	0,0045
1Invest S&P500 Index ETF (ETF500)	0	No	0
1Invest G7 Govt Bond ETF (ETFGGB)	0	No	0
1Invest Global REIT ETF (ETFGRE)	0	No	0
Satrix Value Equity ETF (STXVEQ)	0,008	Yes	0,0242
Sygnia Itrix SWIX40 ETF (SYGSW4)	0,062	Yes	0,0011
NEWFUNDS SWIX 40 ETF (NFSWIX)	0	No	0
1Invest SA Bond ETF (ETFBND)	0,002	Yes	0,0092
Satrix Defensive Eq ETF (STXDEQ)	0,466	Yes	0,0275
Satrix Capped ALL Share (STXCAP)	0	No	0
1Invest MSCI EM Asia ETF (ETFEMA)	0	No	0

Source: Author's estimation (2024)

The ETF portfolio is created using the Markowitz model to attain an optimal risk-reward ratio, utilizing diversification across several asset classes and sectors. The different weights allocated to the selected ETFs indicate a strategic allocation based on the historical returns and risk-return characteristics and also ensuring that the ETF optimized portfolio is in line with the investors goal. To maintain the ideal risk-return balance, the weights and selections may need to be re-evaluated on a periodic basis as market conditions change. To summarize, the Markowitz portfolio optimization algorithm has successfully chosen a subset of 15 ETFs from the full sample of 33 ETFs, allocating weights to build a diversified, risk-adjusted ETF portfolio aiming at maximizing returns.

Table 5.4 Conditional Value at Risk (CvaR) Portfolio Optimisation Results

ETF	Weights_ CvaR	Weight_ CvaR _select (26 Selected)	Average returns
New Gold Palladium ETF (NGPLD)	0,014	Yes	0,0007
Satrix SWIX Top 40 Portf (STXSWX)	0,022	Yes	0,009
Satrix FINI Portfolio (STXFIN)	0,046	Yes	0,0207
New Gold Platinum ETF (NGPLT)	0,03	Yes	0,0106
Sygnia Itrix Eurostoxx50 (SYGEU)	0,018	Yes	0,0023
Sygnia Itrix FTSE100 (SYGUK)	0,018	Yes	0,0008
Satrix RESI Portfolio (STXRES)	0,022	Yes	0,0038
1Invest SA Property ETF (ETFSAP)	0,004	Yes	0,0023
1InvestGold ETF (ETFGLD)	0,036	Yes	0,0126
1InvestPlatinum ETF (ETFPLT)	0,004	Yes	0,0113
Sygnia Itrix MSCI World (SYGWD)	0,084	Yes	0,0045
10X SAProp Income (CSPROP)	0	No	0
New Gold Issuer Ltd (GLD)	0,19	Yes	0,0479
Satrix Shariah T40 ETF (STXSHA)	0	No	0
FNB TOP 40 ETF (FNBT40 0	0	No	0
Satrix DIVI Plus Portfol (STXDIV)	0,09	Yes	0,0038
10X Top 50 (CTOP50)	0,038	Yes	0,0164
1InvestRhodium ETF (ETFRHO)	0	No	0
10X Global Prop (GLPROP)	0,002	Yes	0,0063
EasyETFs AMI Big50 ETF (AMIB50)	0	No	0
Satrix Quality SA ETF (STXQUA)	0,018	Yes	-0,0433
Sygnia Itrix Gbl Pro ETF (SYGP)	0,048	Yes	-0,0037
Sygnia Itrix Top 40 ETF (SYGT40)	0	No	0
1Invest S&P500 Index ETF (ETF500)	0,008	Yes	0,0391
1Invest G7 Govt Bond ETF (ETFGGB)	0,032	Yes	0,0054
1Invest Global REIT ETF (ETFGRE)	0,032	Yes	0,0031
Satrix Value Equity ETF (STXVEQ)	0,074	Yes	0,0115
Sygnia Itrix SWIX40 ETF (SYGSW4)	0,052	Yes	0,0062
NEWFUNDS SWIX 40 ETF (NFSWIX)	0	No	0
1Invest SA Bond ETF (ETFBND)	0,022	Yes	0,0143
Satrix Defensive Eq ETF (STXDEQ)	0,034	Yes	0,0023
Satrix Capped ALL Share (STXCAP)	0,052	Yes	0,0052
1Invest MSCI EM Asia ETF (ETFEMA)	0,01	Yes	0,009

Source: Author's estimation (2024)

From Table 5.4 out of 33 ETFs, 26 were selected to be part of the optimized portfolio using the CvaR optimization model implying that these 26 ETFs jointly minimize risk while providing a considerable return profile. The following ETFs were not chosen 10X SProp Income (CSPROP), Satrix Shariah T40 ETF (STXSHA), FNB TOP 40 ETF (FNBT40), 1InvestRhodium ETF (ETFRHO), EasyETFs AMI Big50 ETF (AMIB50), Sygnia Itrix Top 40 ETF (SYGT40) and NEWFUNDS SWIX 40 ETF (NFSWIX). The

weights allocated to each selected ETF differ, New Gold Issuer Ltd (GLD) obtained the highest weight (0.19), implying that it carries a significant position in the portfolio based on risk and return characteristics. The selection of the final 26 ETF shows diversification was achieved. Diversification is a basic notion in portfolio optimization that distributes risk across multiple assets. The selection of ETFs includes those that track several asset classes, sectors, and locations, which increases diversification.

The CvaR model optimizes the portfolio the portfolio in order to manage potential excessive losses which particularly very important during periods of turbulence. This guarantees that the portfolio is more protected from adverse market fluctuations that exceed the standard VaR threshold. The allocation to various ETFs is designed to provide the best possible stability of risk and return. ETFs with higher weights are projected to generate higher risk-adjusted returns. These results can help investors and portfolio managers understand which ETFs contribute the most to the portfolio's risk and return profile, allowing them to make more informed decisions regarding prospective alterations. Overall, the CVaR portfolio optimization model selected 26 ETFs from a pool of 33 to limit potential extreme losses while aiming for optimal returns. The allocation of weights to these ETFs reflects their contribution to the overall ETF portfolio's risk and return features, showcasing a well-diversified investment strategy aimed at managing risk effectively.

Table 5.5 Mean Absolute Deviation (MAD) Portfolio Optimisation Results

ETF	Weights_ MAD	Weight_MAD_select (11 selected)	Average returns
New Gold Palladium ETF (NGPLD)	0	No	0
Satrix SWIX Top 40 Portf (STXSWX)	0,004	Yes	0,0071
Satrix FINI Portfolio (STXFIN)	0,016	Yes	0,0362
New Gold Platinum ETF (NGPLT)	0,004	Yes	0,0328
Sygnia Itrix Eurostoxx50 (SYGEU)	0,046	Yes	0,0301
Sygnia Itrix FTSE100 (SYGUK)	0	No	0
Satrix RESI Portfolio (STXRES)	0	No	0
1Invest SA Property ETF (ETFSAP)	0	No	0
1InvestGold ETF (ETFGLD)	0	No	0
1InvestPlatinum ETF (ETFPLT)	0,002	Yes	0,012
Sygnia Itrix MSCI World (SYGWD)	0,034	Yes	0,0193
10X SAProp Income (CSPROP)	0	No	0
New Gold Issuer Ltd (GLD)	0	No	0
Satrix Shariah T40 ETF (STXSHA)	0,004	Yes	0,0114
FNB TOP 40 ETF (FNBT40)	0	No	0
Satrix DIVI Plus Portfol (STXDIV)	0	No	0
10X Top 50 (CTOP50)	0	No	0
1InvestRhodium ETF (ETFRHO)	0,006	Yes	0,0072
10X Global Prop (GLPROP)	0	No	0
EasyETF AMI Big50 ETF (AMIB50)	0	No	0
Satrix Quality SA ETF (STXQUA)	0	No	0
Sygnia Itrix Gbl Pro ETF (SYGP)	0	No	0
Sygnia Itrix Top 40 ETF (SYGT40)	0,002	Yes	0,0231
1Invest S&P500 Index ETF (ETF500)	0	No	0
1Invest G7 Govt Bond ETF (ETFGGB)	0	No	0
1Invest Global REIT ETF (ETFGRE)	0	No	0
Satrix Value Equity ETF (STXVEQ)	0,11	Yes	0,0581
Sygnia Itrix SWIX40 ETF (SYGSW4)	0	No	0
NEWFUNDS SWIX 40 ETF (NFSWIX)	0,772	Yes	0,0741
1Invest SA Bond ETF (ETFBND)	0	No	0
Satrix Defensive Eq ETF (STXDEQ)	0	No	0
Satrix Capped ALL Share (STXCAP)	0	No	0
1Invest MSCI EM Asia ETF (ETFEMA)	0	No	0

Source: Author's estimation (2024)

From table 5.5, the `Weights_MAD` column was generated using the MAD optimization model, which aims to reduce the average absolute deviation of portfolio returns from the mean portfolio return, indicating the total number of 11 ETFs chosen for the final ETF portfolio. The highest weight was allocated to NEWFUNDS SWIX 40 ETF (NFSWIX) with 0.772 while the lowest was allocated to 1InvestPlatinum ETF (ETFPLT) and Sygnia Itrix Top 40 ETF (SYGT40) with weights of 0.002. To add on,

The MAD optimization model aims to create a portfolio with the lowest possible average absolute deviation from the mean return. This model is less sensitive to extreme values compared to variance-based models, making it robust to outliers. The portfolio optimization process resulted in the selection of 11 ETFs out of the 33 considered. These selected ETFs are those with non-zero weights, meaning they contribute to the final portfolio. ETFs with a weight of 0 were not selected for the optimized portfolio. These were deemed less optimal by the MAD model based on the objective to minimize absolute deviation. 11 selected ETFs selected to form the optimal ETF portfolio represent wide range of industries and asset classes which aids to portfolio diversification i.e, equities (Satrix SWIX Top 40 Portf (STXSWX)), commodities (1InvestPlatinum ETF (ETFPLT)), international indices (Sygnia Itrix MSCI World (SYGWD)).

By selecting ETFs with lower absolute deviations, the MAD model aims to reduce the risk associated with return volatility. This method can result in more consistent portfolio performance. The chosen ETFs are intended to work effectively together, balancing each other's risks and returns in order to attain the required ETF portfolio characteristics. Investors can utilize the MAD-optimized weights to build their ETF portfolios, ensuring that they invest in the appropriate ETFs in the stated proportions. The MAD model aids in the creation of a less volatile portfolio, which is especially useful for risk-averse investors. By carefully selecting ETFs that contribute positively to the portfolio's goal, investors might potentially obtain higher returns while properly controlling risk. Therefore, MAD optimization model was used to choose and allocate weights to ETFs, with the goal of creating an optimal, diversified, and stable portfolio which was exhibited in Table 5.5.

5.6.3 Machine Learning Portfolio Optimization Results

In this study, advanced machine learning algorithms were utilised, including K4-RANN, Evolutionary (Genetic) Algorithms, and Extreme Gradient Boosting, to construct optimized portfolios across different asset classes. Our objective was to evaluate these techniques' performance in terms of risk-adjusted returns, risk mitigation, and diversification benefits compared to traditional models such as the Markowitz Model, CVaR, and MAD.

Table 5.6 K4-RANN Portfolio Optimisation Results

ETF	Weights_ K4-RANN	Weight_ K4-RANN _select (21 selected)	Average returns
New Gold Palladium ETF (NGPLD)	0	No	0
Satrix SWIX Top 40 Portf (STXSWX)	0	No	0
Satrix FINI Portfolio (STXFIN)	0	No	0
New Gold Platinum ETF (NGPLT)	0,002	Yes	0,0302
Sygnia Itrix Eurostoxx50 (SYGEU)	0,006	Yes	0,0225
Sygnia Itrix FTSE100 (SYGUK)	0,004	Yes	0,0116
Satrix RESI Portfolio (STXRES)	0	No	0
1Invest SA Property ETF (ETFSAP)	0	No	0
1InvestGold ETF (ETFGLD)	0	No	0
1InvestPlatinum ETF (ETFPLT)	0,002	Yes	0,0044
Sygnia Itrix MSCI World (SYGWD)	0,002	Yes	-0,0096
10X SProp Income (CSPROP)	0,002	Yes	0,0129
New Gold Issuer Ltd (GLD)	0,144	Yes	0,0785
Satrix Shariah T40 ETF (STXSHA)	0,002	Yes	0,0069
FNB TOP 40 ETF (FNBT40)	0	No	0
Satrix DIVI Plus Portfol (STXDIV)	0,048	Yes	-0,002
10X Top 50 (CTOP50)	0,006	Yes	0,0638
1InvestRhodium ETF (ETFRHO)	0	No	0
10X Global Prop (GLPROP)	0,054	Yes	0,0016
EasyETF AMI Big50 ETF (AMIB50)	0	No	0
Satrix Quality SA ETF (STXQUA)	0,006	Yes	0,0162
Sygnia Itrix Gbl Pro ETF (SYGP)	0	No	0
Sygnia Itrix Top 40 ETF (SYGT40)	0,002	Yes	0,0018
1Invest S&P500 Index ETF (ETF500)	0,008	Yes	0,0026
1Invest G7 Govt Bond ETF (ETFGGB)	0,08	Yes	0,0302
1Invest Global REIT ETF (ETFGRE)	0,082	Yes	0,0025
Satrix Value Equity ETF (STXVEQ)	0	No	0
Sygnia Itrix SWIX40 ETF (SYGSW4)	0,022	Yes	0,0362
NEWFUNDS SWIX 40 ETF (NFSWIX)	0	No	0

1Invest SA Bond ETF (ETFBND)	0,002	Yes	0,0246
Satrix Defensive Eq ETF (STXDEQ)	0,018	Yes	0,0044
Satrix Capped ALL Share (STXCAP)	0,002	Yes	0,0096
1Invest MSCI EM Asia ETF (ETFEMA)	0,506	Yes	0,0129

Source: Author's estimation (2024)

Table 5.6 presents that K4-RANN optimization model selected 21 ETFs out of a total of 33 to include in the optimized portfolio. The selection is based on the model's allocating analysis and the select ETFs which contribute to achieving the optimal balance of risk and return. The ETF with the maximum weight allocation 1Invest MSCI EM Asia ETF (ETFEMA) with 0.506 while the minimum weight was allocated to (New Gold Platinum ETF (NGPLT), 1InvestPlatinum ETF (ETFPLT), Sygnia Itrix MSCI World (SYGWD), 10X SProp Income (CSPROP), Satrix Shariah T40 ETF (STXSHA), Sygnia Itrix Top 40 ETF (SYGT40), 1Invest SA Bond ETF (ETFBND) and Satrix Capped ALL Share (STXCAP) with 0.002. This shows the model's ability to select an important ETF that contribute to the maximum ETF portfolio returns. The weight allocations are the diversification strategy to spread the risk across the selected 21 ETFs.

The K4-RANN optimization model stresses The K4-RANN approach stresses diversity by selecting a mixed range of ETFs from different sectors and assets classes which contributes to the reduction of portfolio risk. By utilising machine learning optimization technique, the ETF portfolio optimization process gains improved pattern identification and data analysis, potentially leading to greater performance than traditional optimization models. The weight allocations vary to reflect the model's approach to risk and return trade-off. The higher the weights for some ETFs indicate better predicted performance or reduced risk, whereas low allocations to others highlight their function in portfolio diversification. Therefore, table 5.6 summarizes how the K4-RANN model optimized from 33 to 21 ETFs by carefully extracting and weighting the ETFs creating 21 ETF portfolio that optimizes returns while minimizing risk. Thus, demonstrating the potential benefits of utilizing machine learning optimizing techniques in financial markets and financial decision making.

Table 5.7 Evolutionary Genetic Algorithm portfolio optimization results

ETF	Weights_ GA	Weight_ GA _select (29 selected)	Average returns
New Gold Palladium ETF (NGPLD)	0,0357	Yes	0,0245
Satrix SWIX Top 40 Portf (STXSWX)	0,0349	Yes	0,0212
Satrix FINI Portfolio (STXFIN)	0,0134	Yes	0,0247
New Gold Platinum ETF (NGPLT)	0,0364	Yes	0,0163
Sygnia Itrix Eurostoxx50 (SYGEU)	0,0288	Yes	0,0406
Sygnia Itrix FTSE100 (SYGUK)	0,0251	Yes	0,0028
Satrix RESI Portfolio (STXRES)	0,0651	Yes	0,0902
1Invest SA Property ETF (ETFSAP)	0,0191	Yes	0,0106
1InvestGold ETF (ETFGLD)	0,0367	Yes	0,0169
1InvestPlatinum ETF (ETFPLT)	0,0045	Yes	0,0128
Sygnia Itrix MSCI World (SYGWD)	0,0469	Yes	0,0206
10X SAProp Income (CSPROP)	0	No	0
New Gold Issuer Ltd (GLD)	0,0044	Yes	0,0081
Satrix Shariah T40 ETF (STXSHA)	0,0632	Yes	0,018
FNB TOP 40 ETF (FNBT40)	0	No	0
Satrix DIVI Plus Portfol (STXDIV)	0,0408	Yes	-0,0182
10X Top 50 (CTOP50)	0,0470	Yes	0,0117
1InvestRhodium ETF (ETFRHO)	0,0148	Yes	0,0027
10X Global Prop (GLPROP)	0,0251	Yes	-0,0216
EasyETF AMI Big50 ETF (AMIB50)	0,0233	Yes	0,0313
Satrix Quality SA ETF (STXQUA)	0,0132	Yes	-0,0033
Sygnia Itrix Gbl Pro ETF (SYGP)	0,0274	Yes	0,01
Sygnia Itrix Top 40 ETF (SYGT40)	0,0159	Yes	0,0035
1Invest S&P500 Index ETF (ETF500)	0	No	0
1Invest G7 Govt Bond ETF (ETFGGB)	0,0551	Yes	0,005
1Invest Global REIT ETF (ETFGRE)	0,0581	Yes	0,0236
Satrix Value Equity ETF (STXVEQ)	0,0232	Yes	0,0094
Sygnia Itrix SWIX40 ETF (SYGSW4)	0,0442	Yes	0,0323
NEWFUNDS SWIX 40 ETF (NFSWIX)	0	No	0
1Invest SA Bond ETF (ETFBND)	0,0410	Yes	0,0045
Satrix Defensive Eq ETF (STXDEQ)	0,0571	Yes	0,0212
Satrix Capped ALL Share (STXCAP)	0,0430	Yes	0,0147
1Invest MSCI EM Asia ETF (ETFEMA)	0,0554	Yes	0,0163

Source: Author's estimation (2024)

Table 5.7 summaries the ETF portfolio optimization results from the Evolutionary Genetic Algorithm model. To form an optimal portfolio 29 ETFs out of 33 were selected by the model for the optimized portfolio. The selection implies that 29 ETFs selected

were the best based on the GA optimization criteria that is risk minimization, return maximization and diversification. The weights the selected the ETFs in the optimal portfolio vary with highest allocated to Satrix RESI Portfolio STXRES with 0.0651 and Satrix Shariah T40 ETF (STXSHA) with 0,0632 while the lowest weights allocated to 1nvestPlatinum ETF (ETFPLT) at 0.0045 and New Gold Issuer Ltd (GLD) with 0,0044. ETFs not selected implies that the GA model found them not effectively contributing to the optimization objectives (risk reduction and return maximization).

The GA is a machine learning optimization technique and therefore the results show a well-organized output to optimizing the ETF portfolio by picking the most relevant assets and allocating weights based on their relative value. This should aid in developing ETF portfolios that matches with the targeted risk-return trade-offs as well as helping investors to achieve their investment goals.

Lastly, Table 5.8 summaries the optimization results from the XGBoost. This model's weights allocation was different from other models where all the ETFs were selected to achieve the optimal portfolio. Each ETF was allocated the weight that reflects how much of the entire portfolio value should be allocated to it. From the results, the highest weight was allocated to 10X SAProp Income (CSPROP) with 0,0673, Sygnia Itrix Top 40 ETF (SYGT40) with 0,0671 and 10X Top 50 (CTOP50) with 0,0648 while the lowest weight was allocated to Sygnia Itrix SWIX40 ETF (SYGSW4) with 0,0029. This implies that the model's weight allocation was stable and consistent across different iterations. The ETFs with higher weights are deemed more favourable by the XGBoost model implying that they provide a higher return relative to their risk. Conversely, ETFs with lower weights are considered to be providing less risk and return trade-off. These results also show that some ETFs offer high returns or better risk-adjusted returns compared to others. To add on, the model being able to include all 33 ETFs implies that the model's strategy is robust and not subject to significant variability.

Table 5.8 XGBoost Portfolio Optimization Results

ETF	Weights_ XG Boost	Weight_XgBoost_count (33 selected)	Average returns
New Gold Palladium ETF (NGPLD)	0,0277	33	0,0107
Satrix SWIX Top 40 Portf (STXSWX)	0,0361	33	0,0238
Satrix FINI Portfolio (STXFIN)	0,0064	33	-0,0115
New Gold Platinum ETF (NGPLT)	0,0492	33	-0,0003
Sygnia Itrix Eurostoxx50 (SYGEU)	0,0327	33	0,0083
Sygnia Itrix FTSE100 (SYGUK)	0,0096	33	0,0023
Satrix RESI Portfolio (STXRES)	0,0057	33	0,0036
1Invest SA Property ETF (ETFSAP)	0,0477	33	0,0145
1InvestGold ETF (ETFGLD)	0,0081	33	0,0109
1InvestPlatinum ETF (ETFPLT)	0,0574	33	0,0113
Sygnia Itrix MSCI World (SYGWD)	0,0199	33	0,0107
10X SAProp Income (CSPROP)	0,0673	33	0,0119
New Gold Issuer Ltd (GLD)	0,0157	33	0,0239
Satrix Shariah T40 ETF (STXSHA)	0,0198	33	0,0213
FNB TOP 40 ETF (FNBT40)	0,0170	33	0,0193
Satrix DIVI Plus Portfol (STXDIV)	0,0247	33	0,0019
10X Top 50 (CTOP50)	0,0648	33	0,0185
1InvestRhodium ETF (ETFRHO)	0,0362	33	0,0017
10X Global Prop (GLPROP)	0,0024	33	0,0382
EasyETF AMI Big50 ETF (AMIB50)	0,0131	33	0,008
Satrix Quality SA ETF (STXQUA)	0,0213	33	0,0273
Sygnia Itrix Gbl Pro ETF (SYGP)	0,0237	33	0,0445
Sygnia Itrix Top 40 ETF (SYGT40)	0,0671	33	0,0109
1Invest S&P500 Index ETF (ETF500)	0,0486	33	-0,011
1Invest G7 Govt Bond ETF (ETFGGB)	0,0359	33	0,0034
1Invest Global REIT ETF (ETFGRE)	0,0386	33	0,0403
Satrix Value Equity ETF (STXVEQ)	0,0148	33	0,0022
Sygnia Itrix SWIX40 ETF (SYGSW4)	0,0029	33	0,0225
NEWFUNDS SWIX 40 ETF (NFSWIX)	0,0429	33	0,0023
1Invest SA Bond ETF (ETFBND)	0,0437	33	0,0204
Satrix Defensive Eq ETF (STXDEQ)	0,0267	33	0,0351
Satrix Capped ALL Share (STXCAP)	0,0456	33	0,0128
1Invest MSCI EM Asia ETF (ETFEMA)	0,0267	33	0,0337

Source: Author's estimation (2024)

Therefore, for investors need to consider market circumstances as ETF performance change and periodically evaluate and rebalance the portfolio to ensure that the ideal weights and high risk adjusted returns are maintained. Tracking the ETF portfolios performance to confirm that the risk-adjusted returns match the model's expectations.

Adjustments may be required based on actual performance vs the model's projections. Overall, these weights are critical for creating an optimized portfolio using the XGBoost model, which balances risk and return while aiming for optimal diversification.

5.6.4 Comparative Analysis

This study seeks to delve into a comparative analysis of traditional and ML-based optimization approaches. By evaluating their performance in key metrics such as risk-adjusted returns, risk reduction, and diversification, the research aims to uncover the potential advantages and limitations of these paradigms. This exploration is particularly relevant in the context of exchange-traded funds, a popular investment vehicle known for their transparency, liquidity, and diversification benefits. The results in figure 5.3 below shows the comparison results.

Figure 5.3 summaries the comparative performance between traditional versus machine learning portfolio optimization techniques in ETFs. These results compare these models with respect to risk-adjusted returns, risk reduction and effective diversification. In terms of risk-adjusted returns, Bacon (2021) explained that the Sharpe ratio is used to measure the risk-adjusted returns, and a high Sharpe ratio indicates better risk-adjusted performance. From the results, XGBoost model has the highest Sharpe ratio of 1,034 implying that it provides the best risk-adjusted returns as compared to other models. Cvar and K4-RANN have also shown relative higher Shape ratio compared to other models. MAD is the optimizing model that provided the lowest Sharpe ratio of 0,4754 indicating less favourable risk-adjusted returns.

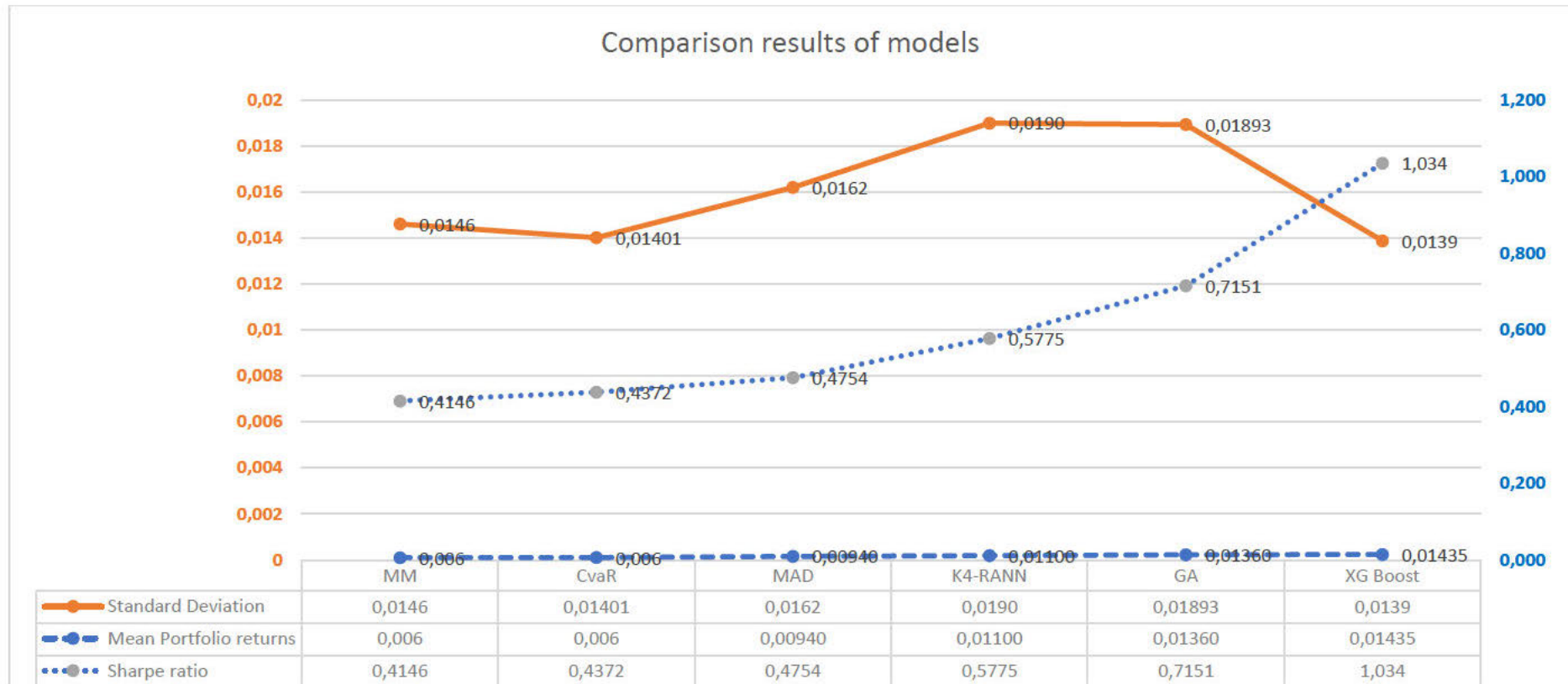


Figure 5.3: Comparison Analysis of Models

Source: Author's own estimations (2024)

The comparison results of the models (MM, CVaR, MAD, K4-RANN, GA, and XG Boost) reveal that XG Boost is the most effective model, achieving the highest mean portfolio return of 0.01435, the lowest risk with a standard deviation of 0.0139, and the best diversification, as indicated by its Sharpe ratio of 1.034. This highlights XG Boost's ability to deliver high returns per unit of risk. While GA (Genetic Algorithm) and K4-RANN also demonstrate competitive returns of 0.01360 and 0.01100, respectively, they come with higher risk levels, with GA showing a standard deviation of 0.01893 and K4-RANN at 0.0190. On the other hand, traditional models such as MM and CVaR exhibit the lowest returns (both at 0.006) but maintain relatively lower risk, with standard deviations of 0.0146 and 0.01401, respectively. However, their Sharpe ratios, 0.4146 for MM and 0.4372 for CVaR, reflect poor risk-adjusted returns and limited diversification benefits. Overall, machine learning models outperform traditional techniques, with XG Boost emerging as the clear leader in delivering high returns, low risk, and effective diversification.

Contextualising these results to ETF portfolio optimization, the traditional optimization models specifically the Markowitz model has moderate mean portfolio returns and standard deviation, but its Sharpe ratio is lower than machine learning optimization models. This implies that the Markowitz model has less effective risk-adjusted returns. MAD has the lowest risk-adjusted returns and the most risk. CVaR outperformed the MAD and MM in terms of mean portfolio returns and risk-adjusted performance. In contrast, among the machine learning optimization models, the evolutionary Genetic algorithm shows the best performance. The GA model has the highest Sharpe ratio and lowest standard deviation making it the most effective in terms risk-adjusted returns and risk reduction. To add on, XG Boost provides the right balance of risk and return outperforming the traditional optimization models. The XGBoost optimization model has the highest mean portfolio returns which works well for an investor who is return oriented.

Machine learning optimization techniques specifically the GA and K4-RANN exhibit higher risk-adjusted returns and minimum risk more effectively than traditional optimization techniques. The XGBoost model also promises to be the next best with the highest returns implying that it can be used as a useful tool for diversification. Traditional portfolio optimization models can be successful in producing high returns

like the CvaR but often underperform the machine learning optimization techniques in terms of risk-adjusted returns and risk reduction.

H1: Do machine learning portfolio optimisation techniques yield higher risk adjusted returns than traditional portfolio optimisation techniques in ETF portfolios?

The Sharpe ratio is a typical measure of risk-adjusted return and the higher the ratio the better risk adjusted performance (Pav, 2021). Machine learning portfolio optimization techniques, particularly the Genetic Algorithm, produced higher risk adjusted returns than traditional portfolio optimization techniques. The results shows that machine learning techniques, particularly GA and K4-RANN, produce larger risk-adjusted returns than traditional optimization techniques. This suggests that utilisation of machine learning methods into ETF portfolio optimization can lead to improved performance in terms of balancing returns and related risks.

H2: Do machine learning portfolio optimisation techniques reduce risk more than traditional portfolio optimisation techniques in ETF portfolios?

A lower standard deviation indicates less risk. Machine learning portfolio optimization techniques in ETFs are more effective at-risk reduction than traditional optimization techniques. Traditional optimization models specifically MM, CVaR, and MAD, have high values of standard deviations, indicating high risk. MAD, in instance, has the highest risk and is hence the least effective risk-reduction model. This comparison analysis establishes that machine learning techniques, particularly GA and K4-RANN, are more effective in reducing risk than traditional portfolio optimization models. Incorporating machine learning models into ETF portfolio optimization can result in more stable and less risky investment portfolios.

H3: Do machine learning portfolio optimisation techniques offer more effective diversification than traditional portfolio optimisation techniques in ETF portfolios?

As mentioned before that diversification is not directly measured by the models used, but it considers both the risk (standard deviation) and the returns characteristics of the ETF portfolios. Effective ETF portfolio diversification typically leads to a combination of lower risk and high returns (Herzog *et al*, 2023). Machine learning portfolio

optimization techniques offer more effective diversification than traditional portfolio optimization techniques in ETFs. This has been proved by the XGBoost optimization model having the lowest standard deviation along with the highest Sharpe ratio of 1,034 and therefore it produces highly diversified ETF portfolios. The other machine learning optimization models (K4-RANN and GA also have higher returns than the traditional optimization models while maintaining moderate low risk thus achieving effective diversification.

5.7 Summary

Traditional methods, including the Markowitz model, CVaR, and MAD, emphasize risk-return balance based on historical data and statistical metrics. Machine learning models such as K4-RANN, Evolutionary Genetic Algorithms, and XGBoost leverage advanced computational capabilities to uncover complex patterns and adapt to dynamic market conditions. Using daily data from 33 ETFs listed on the JSE over ten years, the chapter systematically implements these models to construct optimized portfolios. Performance evaluation employs metrics such as mean portfolio returns, standard deviation, and the Sharpe ratio to assess the effectiveness of each approach. Results show that machine learning models consistently achieve higher diversification and better risk-adjusted returns, underscoring their potential to address limitations in traditional methods like rigid assumptions and data sensitivity.

The findings highlight significant differences in portfolio performance and robustness under varied market conditions. Traditional models are constrained by their reliance on historical correlations and risk measures, while machine learning techniques adapt to evolving market dynamics, enhancing predictive accuracy and optimization. However, challenges in computational complexity and interpretability remain for machine learning approaches. The study provides crucial insights for investors and portfolio managers in South Africa and similar emerging markets, demonstrating the practicality of integrating advanced machine learning techniques for superior ETF portfolio management. By bridging gaps in literature and practice, this research contributes to the understanding of innovative financial technologies in optimizing investment strategies. Therefore, in comparison, traditional optimization models like MM and CVaR show moderate diversification, while MAD exhibits poor diversification due to its high risk and low returns. The machine learning optimization techniques,

particularly GA and K4-RANN, demonstrated more effective diversification by providing lower risk and higher returns compared to traditional optimization techniques. This suggests that utilising machine learning methods into ETF portfolio optimization can lead to more stable and well-diversified investment portfolios.

CHAPTER 6: ASSESSMENT ON THE PERFORMANCE OF ETF PORTFOLIOS CONSTRUCTED BY MACHINE LEARNING TECHNIQUES PERFORM UNDER DISTRESSED MARKET CONDITIONS

6.1 Introduction

The financial landscape has witnessed substantial transformations with the advent of sophisticated technologies, particularly in the realm of investment management (Allioui and Mourdi, 2023). Among these innovations, the application of ML techniques in constructing Exchange Traded Fund portfolios has garnered significant attention. ETFs have become an increasingly popular investment vehicle due to their flexibility, liquidity, and cost-effectiveness (Jankov, 2021). They offer investors a diversified portfolio that can track a wide array of indices, sectors, commodities, or bonds (McMillan *et al.*, 2011). With the advancement of technology, the use of machine learning techniques to construct and optimize ETF portfolios has gained significant traction (Simar, 2023). These techniques promise enhanced predictive power and adaptive strategies to navigate market complexities (Strielkowski *et al.*, 2023). However, the efficacy of these machine learning-constructed ETF portfolios under distressed market conditions remains a critical area of analysis, particularly in emerging markets such as South Africa.

Studying portfolio performance under distressed market conditions is crucial for several reasons; it helps in identifying the vulnerabilities of a portfolio (Shtini, 2023). Antunes *et al.* (2023) also added that distressed conditions can reveal which assets or asset classes are more prone to significant losses. Conducting stress tests under these conditions can help in assessing how the portfolio might behave during severe market downturns. Understanding how a portfolio performs during market stress can aid in developing strategies to mitigate losses (Bacon, 2023). This is vital for preserving capital during periods of high volatility and uncertainty. Investors are more likely to trust a portfolio manager who has demonstrated an understanding of how their portfolio will perform under adverse conditions and has strategies in place to manage such risks (Bacon, 2023). To add on, regulatory bodies often require financial institutions to conduct stress tests to ensure they have adequate capital to withstand distressed conditions. Studying portfolio performance in such scenarios helps in meeting these regulatory requirements (Jing *et al.*, 2023). The insights gained can be used to optimize the portfolio design, making it more robust and diversified to handle

future distressed conditions effectively. Therefore, studying portfolio performance under distressed market conditions is not just a theoretical exercise but a practical necessity for effective risk management, strategic planning, and ensuring long-term financial stability and growth.

An overview of machine learning techniques in portfolio construction have been thoroughly discussed in previous chapters. The primary purpose of this chapter is to evaluate the performance of ETF portfolios constructed using machine learning techniques during periods of market distress in South Africa. Distressed market conditions, characterized by high volatility, sharp declines in asset prices, and reduced liquidity, present unique challenges for portfolio management (Eisfeldt, Lustig and Zhang, 2023). Understanding how these machine learning-based ETF portfolios perform under such conditions is crucial for several reasons; Investors need to know how resilient these portfolios are in adverse market conditions to manage risks effectively. Insights from this research can guide the development of robust investment strategies that can withstand market downturns. Assessing the performance of these portfolios contributes to the broader understanding of market dynamics and stability in emerging markets. This study highlights the practical implications of machine learning in financial markets, potentially driving further innovation in the field. Therefore, by focusing on the South African market, this research also sheds light on the unique characteristics and challenges of emerging markets, which often exhibit different dynamics compared to developed markets. By doing so, it seeks to contribute valuable insights to investors, portfolio managers, and financial technology developers, enhancing their ability to navigate and thrive in challenging market environments.

This chapter's aim is to provide a comprehensive assessment of the viability and robustness of machine learning-constructed ETF portfolios in an emerging market particularly in South Africa under distressed market conditions. The aim leads this chapter to answer the following sub-research questions; how do machine learning-constructed ETF portfolios manage downside risk during distressed market conditions? Do machine learning-constructed ETF portfolios maintain consistent performance in terms of returns and risk-adjusted returns during periods of market stress? How do machine learning techniques handle volatility in ETF portfolios during distressed market conditions? The distressed periods include (natural disasters,

floods (April 2022), market volatility, COVID-19 Pandemic (March 2020 to July 2022.), historical events, Social Unrest and Riots (July 2021) because these were common international and domestic events, well established in other papers.

The chapter is organized as follows; section 6.2 provides conceptual literature and review of existing literature on ETF portfolio construction, studies on market distress and its impact on financial instruments, gap analysis and contribution of the current study. Section 6.3 provides a brief description of the machine learning techniques used for ETF portfolio construction, definition, criteria used for distressed market conditions, data sources and sample period, performance metrics for evaluating ETF portfolios such as risk-adjusted returns and Sharpe ratio. Section 6.4 contains previous studies on market distress and its impact on the performance of portfolios. Section 6.5 contains data analysis which includes summary statistics of the dataset, analysis of market conditions during the sample period, implementation of machine learning models and portfolio construction and performance comparison of machine learning-constructed portfolios under normal and distressed conditions. Section 6.6 reports the results, empirical analysis, and the discussion of the results. Finally, section 6.7 concludes the chapter.

6.2. Literature Review on Market Distress and Financial Crises

6.2.1 Market Distress and Financial Crises

Market distress and financial crises are periods of intense economic instability marked by sharp declines in asset prices, increased volatility, and financial institution failures. While market distress may involve temporary volatility, financial crises are deeper, affecting credit availability, investor confidence, and often leading to economic recessions. Historical crises, like the 2008 Global Financial Crisis, reveal the interconnectedness of global markets and the need for robust regulatory measures to prevent such collapses. Understanding these events helps policymakers and investors manage risks, improve crisis response strategies, and build more resilient financial systems.

6.2.1.1 Market Distress Conditions

Bhattacharya *et al.* (2023) defined the distressed market conditions as a state in the financial markets where assets are being sold off rapidly, prices are declining

significantly, and overall market confidence is low. These conditions are often characterized by heightened volatility, reduced liquidity, and a general sense of panic or fear among investors. Distressed market conditions are characterized by high volatility which are frequent and large fluctuations in asset prices; declining asset prices which is the persistent downward trends in the prices of securities, commodities, or other financial instruments; increased selling pressure where investors will be selling off assets rapidly, often at a loss, to avoid further losses (Mo *et al.*, 2024). Reduced liquidity which entails difficulty in buying or selling assets without causing a significant change in the price; wider bid-ask spreads which is the difference between the buying and selling prices of assets widens, indicating decreased market efficiency. Credit tightening where lenders become more risk-averse, leading to higher borrowing costs and stricter lending standards; market sentiment when negative sentiment prevails, driven by fear, uncertainty, and doubt. Economic indicators are often accompanied by deteriorating economic indicators such as rising unemployment, declining consumer confidence, and reduced industrial production (Mo *et al.*, 2024).

6.2.1.2 Financial Crises

Calomiris and Jaremski (2024) defined a financial crisis as a broader and more severe form of market distress that typically involves widespread disruption to financial institutions, markets, and the economy as a whole. Financial crises can take various forms, including banking crises, currency crises, sovereign debt crises, and systemic financial crises. The following are the characteristics of financial crises: *Banking Crises* which leads to bank runs where large numbers of depositors withdraw their funds simultaneously due to concerns about the bank's solvency. *Insolvency* where banks' liabilities exceed their assets, leading to failure or the need for government intervention. *Credit Crunch* which can lead to severe reduction in the availability of credit, causing a slowdown in economic activity (Calomiris and Jaremski, 2024).

Currency crises which lead to rapid depreciation which implies a sudden and sharp decline in the value of a country's currency (Udo and Agbai, 2023). The currency crises can also lead to foreign exchange reserves depletion where central banks run out of foreign reserves trying to defend the currency value (Akdogan, 2020). Capital flight can also be caused where large outflow of capital as investors lose confidence in the country's economy. *Sovereign Debt Crises* is another type of financial crises which

lead to default risk where the possibility that a country will be unable to meet its debt obligations (Koh *et al.*, 2020). It also leads to rising bond yields where investors demand higher yields for holding the country's debt, reflecting increased risk. *Systemic Financial Crises* which lead to interconnected failures where the collapse of one financial institution leads to a domino effect, causing failures across the financial system (Pandey, 2024). To add on, market freezes where key financial markets seize up, halting the flow of funds. Therefore, the government may intervene where large-scale government actions, such as bailouts, nationalizations, and monetary policy interventions, to stabilize the system.

6.2.1.3 The Connection Between Market Distress and Financial Crises

Financial crises and market distress are closely interconnected phenomena in the realm of economics and finance, often influencing and exacerbating each other (Zemla, 2021). Financial crisis refers to a situation where the value of financial institutions or assets drops rapidly. It typically involves the collapse of financial institutions, stock markets, and the financial system as a whole (Fligstein, 2021). Market distress, on the other hand, involves significant declines in market prices, high volatility, and disruptions in normal market operations, leading to a loss of investor confidence (Nalban and Smădu, 2021).

Sufi and Taylor (2022) explained financial crises can be triggered by a variety of factors including excessive debt levels, asset bubbles, bank runs, currency crises, and systemic failures within financial institutions. Market Distress can be triggered by financial crises, as well as by economic recessions, geopolitical events, natural disasters, and sudden shifts in investor sentiment. Therefore, financial crisis often leads to market distress (Bolton *et al.*, 2021). For instance, the 2008 financial crisis, which started with the collapse of major financial institutions, led to widespread market distress characterized by plummeting stock prices, increased volatility, and a liquidity crunch. Market distress can exacerbate financial crises. Falling asset prices can lead to losses for financial institutions, triggering further selling and a downward spiral. This was evident during the 1929 stock market crash, which deepened the Great Depression (Sufi and Taylor, 2022).

Financial Crises lead to severe economic downturns, as credit markets freeze, businesses struggle to obtain financing, and consumer confidence drops (Zemla, 2021). Market Distress affects the broader economy through wealth effects (reduced household wealth leading to lower consumption), higher cost of capital (making it more expensive for businesses to finance operations and investments), and increased uncertainty (which can dampen economic activity) (Sufi and Taylor, 2022). During Financial Crises, governments and central banks often intervene with bailouts, monetary easing, and fiscal stimulus to stabilize the financial system (Mody and Nedeljkovic, 2024). In periods of market distress, authorities might implement measures such as market interventions, regulatory changes, and temporary suspensions of trading to restore confidence and stability.

The following are the examples of financial crisis that led to market distress; 2008 Global Financial Crisis triggered by the collapse of Lehman Brothers and the subprime mortgage crisis, this financial crisis led to severe market distress globally, with stock markets crashing and a prolonged period of economic recession. COVID-19 Pandemic, the initial outbreak led to market distress with rapid declines in stock prices and high volatility. The financial strain on businesses and individuals raised concerns about a potential financial crisis, prompting swift intervention by central banks and governments. Therefore, connection between financial crises and market distress is cyclical and interdependent. Financial crises can cause market distress, which in turn can deepen and prolong the crisis. Understanding this connection is crucial for developing effective policies and interventions to mitigate the impact of such events on the economy (Sufi and Taylor, 2022).

Both distressed market conditions and financial crises can have severe economic consequences, including recessions, high unemployment rates, loss of wealth, and long-term damage to financial systems and institutions (Guerra and Eboreime, 2021). Effective policy responses and regulatory frameworks are critical in managing and mitigating the impacts of such events. Market distress and financial crises have occurred throughout history, often triggered by a combination of economic, political, and social factors (Ceylan, 2021). Here are some notable examples and each of these crises had unique causes and effects, but they all underscored the interconnectedness

of global markets and the importance of sound financial regulation and risk management.

Tulip Mania (1634-1637) which happened in Netherlands. Often considered the first recorded speculative bubble, Tulip Mania saw the prices of tulip bulbs in the Dutch Republic soar to extraordinary levels before collapsing. At the peak, some tulip bulbs sold for more than ten times the annual income of a skilled artisan. When the bubble burst, prices plummeted, leaving many investors in financial ruin (Enow, 2023).

South Sea Bubble (1720) which happened in Great Britain. The South Sea Company was granted a monopoly to trade with South America by the British government. Speculation in the company's stock led to a massive bubble, with share prices skyrocketing. When it became evident that the company's profits were grossly exaggerated, the bubble burst, causing significant financial losses and economic downturn in Britain (Enow, 2023).

Panic of (1837) which happened in United States. Triggered by speculative lending practices, a sharp decline in cotton prices, and restrictive lending policies from the Bank of England, the Panic of 1837 led to a major financial crisis. Banks collapsed, businesses failed, and unemployment soared, resulting in a severe economic depression that lasted for several years (Enow, 2023).

Long Depression (1873-1896) which was global, with significant impact in Europe and the United States. Initiated by the collapse of the Vienna Stock Exchange and subsequent bank failures, the Long Depression was characterized by prolonged economic stagnation, deflation, and high unemployment. It highlighted the vulnerabilities of the global economic system and led to significant changes in financial regulation and banking practices (Enow, 2023).

Wall Street Crash of 1929 which happened in United States. The stock market crash of October 1929 marked the beginning of the Great Depression. Over a few days, the market lost a significant portion of its value, leading to widespread panic and massive financial losses. The ensuing economic downturn affected the global economy, causing widespread unemployment, poverty, and social upheaval (Enow, 2023).

Asian Financial Crisis (1997-1998) which happened in East and Southeast Asia. Beginning in Thailand with the collapse of the Thai baht, the crisis spread to other Asian economies, causing severe currency devaluations, stock market declines, and economic turmoil. The crisis highlighted weaknesses in the financial structures of these countries, leading to significant reforms and increased international financial cooperation (Enow, 2023).

Dot-com Bubble (2000) which happened in United States, with global impact. The late 1990s saw a speculative boom in internet-based companies, with many businesses going public at inflated valuations despite lacking substantial revenue or profit. The bubble burst in 2000, causing many dot-com companies to fail and leading to a sharp decline in the NASDAQ index. The aftermath saw significant financial losses and a cooling of the tech sector (Enow, 2023).

Global Financial Crisis (2007-2008) which was global, originating in the United States. Triggered by the collapse of the housing market and the resulting mortgage-backed securities crisis, the financial system faced severe stress. Major financial institutions failed or required bailouts, leading to a global recession. The crisis exposed significant weaknesses in financial regulation and risk management practices, prompting widespread reforms (Enow, 2023).

European Debt Crisis (2010-2012) which happened in Europe. Following the global financial crisis, several European countries, including Greece, Portugal, Ireland, and Spain, faced severe sovereign debt problems. The crisis led to bailouts from the European Union and the International Monetary Fund and prompted austerity measures and economic reforms. It exposed flaws in the Eurozone's economic integration and governance structures (Enow, 2023).

COVID-19 Pandemic (2020) which was global. The pandemic led to unprecedented global economic disruption, with widespread lockdowns, travel restrictions, and a sharp decline in economic activity. Stock markets experienced extreme volatility, and governments implemented massive fiscal and monetary interventions to stabilize economies. The pandemic highlighted vulnerabilities in global supply chains and the need for robust crisis response mechanisms (Enow, 2023).

6.2.2 Distressed Market Conditions and Portfolio Performance

Andreou *et al.* (2021) explained that portfolios that are not well-diversified or are overly exposed to high-risk assets may suffer disproportionately during these periods. Even traditionally stable assets like bonds may experience price fluctuations, albeit usually less severe than equities. Koumou (2020) also added that portfolios relying heavily on diversification to reduce risk may find their risk management strategies less effective. The benefits of diversification are diminished, leading to potential losses across multiple asset classes. Investors may face challenges in rebalancing portfolios or liquidating positions, potentially leading to forced sales at unfavourable prices. This can exacerbate losses, particularly in illiquid assets or markets. Boonman (2023) Portfolios heavily weighted in riskier assets may underperform as capital flows out of these assets. Conversely, portfolios with significant exposure to safe-haven assets might see relative outperformance or reduced losses.

Dahlquist and Ibert (2024) noted that expected returns may fall short, particularly for portfolios that do not adapt to the changing conditions. Long-term investment strategies may need re-evaluation as risk-adjusted returns diminish. Higher risk can lead to increased drawdowns and reduced risk-adjusted returns (Mirza *et al.*, 2022). Ge and Weisbach (2021) also added that portfolios that do not adjust their risk exposure may suffer more severe losses. Portfolios constructed using machine learning techniques may require recalibration or re-training of models to account for the new market dynamics (Trichilo, 2020). Failure to do so can lead to suboptimal performance or unexpected losses. Portfolios may need to be rebalanced more frequently, and asset allocations might require adjustments to align with the new risk-return environment. Strategic shifts, such as increasing cash holdings or shifting to defensive sectors, could become necessary (Harvey *et al.*, 2021).

6.2.3 ETF Performance Under Distressed Market Conditions

Although ETFs are generally known for their high liquidity, they have faced significant illiquidity during periods of financial distress. The 2010 Flash Crash, which caused a 9% drop in the DJIA, led to price dislocations and liquidity disruptions in ETFs. Equity-based ETFs with domestic securities were particularly impacted (Clements, 2020). Grund (2020) further noted that in response to the Flash Crash, the SEC introduced a rule mandating the suspension of trading for specific securities and ETFs experiencing

high volatility under certain conditions. This rule was first enforced on August 24, 2015, when trading was halted for over 300 ETFs.

The SEC's position is supported by Borkovec *et al.* (2010), who point out that during the 2010 Flash Crash, there was a notable lack of liquidity in ETFs and rising spreads. By examining how authorized participants' (APs') arbitrage activity decreases as the underlying stocks' illiquidity rises, Pan and Zeng (2017) provide a potential explanation for the order book's depletion. AP trading volume sharply declines during market turbulence, increasing the risk of liquidity in times of financial hardship. Similarly, arbitrageurs decrease their trading activity during market turbulence, which may exacerbate liquidity problems, according to Ben-David *et al.* (2012). Further evidence of a destabilizing effect of ETFs on securities was provided by Ben-David *et al.* (2018), who showed that stocks with higher ETF ownership show increased negative skewness during financial crises. Furthermore, Madhavan (2012) demonstrated that ETFs linked to very illiquid stocks suffer a drop in trading volume in fragmented markets, which causes price mispricing. Dannhauser (2017) looked at the corporate bond market and discovered that while educated investor ownership increased, liquidity trader participation decreased following the launch of corporate bond ETFs.

Malamud (2016) created a dynamic equilibrium model of the ETF market in which authorized participants (APs) are used by the ETF sponsor to create and redeem shares. Demand shocks are transmitted to the prices of underlying securities through the interaction between the ETF sponsor and APs, which serves as a conduit for shock propagation. The study showed that ETFs could use the arbitrage mechanism to affect component stock volatility. Furthermore, Malamud (2016) discovered a positive correlation between the liquidity of the underlying securities and the magnitude of shock propagation. The study concluded that the arbitrage channel is how non-fundamental shocks at the ETF level are conveyed, which causes constituent equities to become more volatile.

6.2.4 Market Distress and its Impact on the Portfolios' Performance

Jiang *et al.* (2024) introduced a dynamic robust portfolio selection model designed to minimize worst-case scenarios using Conditional Value at Risk as the risk measure. The proposed empirical model for portfolio dynamics incorporates three key features:

(i) it captures tail dependence between assets through a mixture of copula functions, (ii) it accounts for conditional heteroscedasticity and leverage effects using a GJR-GARCH model, and (iii) it incorporates extreme events by employing parametric and semiparametric hybrid models for the marginal distribution of asset returns. The study evaluated the portfolio's performance before and during the COVID-19 pandemic using statistical metrics such as the Sharpe ratio, cumulative returns, and volatility. The findings revealed that the CVaR portfolio outperformed benchmark portfolios during periods of market turmoil while also demonstrating strong performance in stable market conditions.

Zhu and Fukushima (2009) and Han *et al.* (2017) also explored optimal portfolios based on maximizing the CVaR risk measure. However, unlike these studies, Jiang, Olmo, and Atwi (2024) incorporated a mixture copula model to characterize the dynamics of multivariate returns and employed the GJR-GARCH model to account for extreme events and conditional heteroscedasticity. These enhancements played a crucial role in demonstrating the model's robust performance during periods of market stress.

D'Hondt *et al.* (2020) explored the potential influence of machine learning on financial markets, particularly during periods of market stress, and highlighted this as a key area for future research, including events like the 2008 financial crisis. The study suggested that machine learning could be especially valuable in such conditions by offering low-cost, automated investment management that is less susceptible to human emotions and biases. Additionally, D'Hondt *et al.* (2020) argued that machine learning could help mitigate "herding behaviour" among investors, where individuals follow the actions of others rather than making independent, data-driven decisions.

Sindi (2023) analysed the effectiveness of machine learning techniques in forecasting stock returns during periods of market distress in Belgium. The study considered events related to terrorist attacks, natural disasters, and political changes. The terrorist attack category included the 9/11 attacks, and the Paris terrorist attacks, while natural disasters encompassed the Thailand tsunami and the Fukushima nuclear disaster. Political changes were represented by the Brexit referendum outcome and Donald Trump's presidential election. The analysis covered the period from January 2010 to

September 2022, using a proxy market of 50 stocks selected from the MSCI index. The findings indicated that machine learning-based portfolios consistently delivered more meaningful results for investors, particularly in terms of risk-adjusted returns.

From the analysis of natural disaster events, the following insights were drawn: For both the Thailand tsunami and the Fukushima nuclear disaster, investors initially favoured the Markowitz portfolio during both the pre-event and post-event periods. However, during the event itself, the machine learning portfolio demonstrated a stronger return potential relative to its level of exposure. This effect was particularly pronounced in the case of the Thailand tsunami, where the ML portfolio remained the preferred choice for investors into the first quarter of the post-event period. For the two political events examined, the ML strategy consistently delivered higher returns relative to risk across all periods. However, the study did not delve into portfolio optimization within specific sectors or the impact of domestic financial events, leaving an unexplored research gap. To address this, the present study will focus on unit trust returns in developing countries.

Corbett *et al.* (2020) examined the impact of the COVID-19 pandemic on gold and cryptocurrencies. Their findings revealed a significant relationship between Bitcoin's volatility and that of major Chinese stock markets, specifically the Shanghai-Shenzhen 300 (SS 300) index and the S&P 500 index, during periods of severe financial stress. Similarly, Shaikh and Huynh (2021) utilized regression models to assess the effects of the COVID-19 pandemic on global equity markets, commodities, and the foreign exchange (FX) market, using the investors' fear index as a measure. Their results indicated that options serve as the most effective hedge against uncertainty, such as COVID-19, and that option-based implied volatility is the most reliable indicator of investor sentiment and market fluctuations.

Ngo *et al.* (2021) employed the fixed-effects model and the generalized method of moments (GMM) to analyse the impact of green finance development on the economic position of ASEAN countries amid a severe crisis, such as the COVID-19 pandemic. Their findings highlighted a positive relationship between green finance, capital formation, and government spending on education, all of which contributed to the economic growth of ASEAN nations.

6.2.5 Literature Gap Analysis

This study aims to address a gap and introduce new knowledge to the existing literature on ETF portfolio performance under market distress, particularly focusing on the application of machine learning optimization techniques. Jiang, Olmo and Atwi (2024) emphasized the importance of robust portfolio selection using risk measures like Conditional Value at Risk (CVaR) but focused primarily on traditional econometric models like GJR-GARCH without integrating ML techniques. D'Hondt *et al.* (2020) and Sindi (2023) explored the potential and effectiveness of ML techniques in financial markets, yet Sindi (2023) did not delve into portfolio optimization extensively, leaving a gap in the specific application of ML techniques in portfolio optimization. There is a lack of comprehensive research on the assessment of ML techniques in constructing and optimizing portfolios in ETFs, under distressed market conditions.

Jiang, Olmo and Atwi (2024) showed the superiority of their GJR-GARCH-EVT model in portfolio selection under market distress but did not compare their results to ML-based techniques. There is a need for detailed comparative analysis between ML-portfolio optimization models in constructed ETF portfolios during the periods of market distress. Jiang, Olmo and Atwi (2024) and Corbett *et al.* (2020) examined the impact of specific distress event like the COVID-19 pandemic on various asset classes and portfolio performances. Sindi (2023) assessed ML techniques during distressed conditions but limited the scope to stock returns forecasting in Belgium without focusing on ETFs. The specific performance of ETF portfolios constructed using ML techniques under various types of market distress has not been thoroughly investigated.

Therefore, the contribution of this study uncovers the utilisation advanced ML techniques to construct and optimize ETF portfolios, addressing the gap in the literature where ML techniques have not been applied to ETF portfolios under distressed market conditions. To add on, this study also conducts a comparative analysis between ML optimization techniques providing insights into the relative effectiveness and robustness of ML techniques. By analysing a wide array of market distress events, the study offers a comprehensive view of how different types of market shocks impact the performance of ML constructed ETF portfolios.

Unlike previous studies that predominantly focused on developed markets, this study extended the analysis to include emerging markets, providing valuable insights into how ML constructed ETF portfolios perform in different economic environments and the study empirically validated the performance of ML constructed ETF portfolios using statistical measures namely Sharpe ratio, standard deviation, cumulative returns and volatility, demonstrating their effectiveness during both calm and turbulent periods. Therefore, by addressing these gaps, this study significantly adds a new knowledge on understanding the potential and limitations of ML optimization techniques in constructing resilient and high-performing ETF portfolios under distressed market conditions.

6.3 Data and Methodology

This section describes the method of analysis used to achieve the third objective where the definition, identification of distressed market conditions with data sources and sample period were explained. Lastly, performance metrics for evaluating ETF portfolios using risk-adjusted returns, Sharpe ratio, mean returns and standard deviation are also explained.

6.3.1 Data Sources and Sample Period

Following previous studies (Shtini, 2023; D'Hondt *et al.*, 2020), returns are calculated from historical daily price data of 33 ETFs listed on the JSE. To be specific, ETFs returns calculated from the last chapter were used in the analysis for this chapter. The distressed market period comprised of three types of disasters which happened from 2020 to 2022 namely, (natural disasters - April 2022 floods), Social Unrest and Riots - July 2021), COVID-19 Pandemic - March 2020 to July 2022). The normal period is from January 2017 to February 2020 which did not influence the distressed period. The distressed period from March 2020 to July 2022, captured the full effect of the three types of events.

Saharan *et al.* (2010) explained that when conducting research that involves analysing periods "before" and "during" a particular event, the choice of these periods is crucial for the validity and robustness of the analysis. The event of interest must be clearly defined and significant enough to impact the variables being studied. For example, in financial studies, a period of economic recession or a market crash might be chosen.

Pandey and Kumari (2021) also added that the period before the event should represent a "normal" state, free from the influence of the upcoming event and this helps in establishing a baseline for comparison. For this study, the normal period is from January 2017 to February 2020 which did not influence the distressed period. COVID-19 was a sudden crisis in South Africa, as it was in most parts of the world. The virus spread rapidly across countries, including South Africa, leading to an abrupt and widespread impact on public health, the economy, and social systems. The suddenness of the crisis was marked by the rapid increase in infections and the immediate need for a response, which included a strict nationwide lockdown announced by the South African government in March 2020. The "during" period which is the distressed period in this study captured the full effect of the three types of events from March 2020 to July 2022.

Following the theoretical support by Pandey and Kumari (2021) who postulated that both periods must be long enough to provide sufficient data for statistical analysis. If the period is too short, the analysis might suffer from low statistical power. For this reason, this study used daily data which has more than 500 observations to support the statistical power. Attention was also taken to avoid overlap between the before and during periods and the periods are consistent with the nature of the events. The choice of periods aligned with the research question. In this case, the chapter aims to assess the performance of ETF portfolios formulated by machine learning optimization techniques, so the periods chosen are comparable across normal and distressed periods to avoid bias. To add on, comparative analysis as to which of the ML optimization techniques performs better during the periods of distress by giving high returns at low risk.

6.3.2 Methodology

The philosophy of the methodology employed in this chapter is an event study which was similar to Alam *et al.* (2020), Pandey and Kumari (2021), Sayed and Eledum (2023), however this study utilised three machine learning models applied in the previous chapter (an enhanced radial basis function neural network mode an enhanced radial basis artificial neural network (K4-RANN), Evolutionary Genetic Algorithms (GA) and Extreme gradient boosting (XGBoost)).

The event study methodology is a widely used empirical technique in finance and economics to assess the impact of a specific event on the value of a firm, a financial asset or a portfolio. The idea is to examine whether the event has led to an abnormal return or a change in the behaviour of the portfolio compared to what would have been expected under normal circumstances. This chapter aims to assess how ETF portfolios optimized by machine learning techniques perform under distressed market conditions and therefore the event study methodology was appropriate to assess the impact of specific distressed events (natural disasters (April 2022 floods), Social Unrest and Riots (July 2021), COVID-19 Pandemic (March 2020 to July 2022) on the performance of the optimized portfolios.

Abnormal returns represent the difference between the actual returns observed during the event period and the expected returns under normal circumstances (Farooq, Nasir, Bilal and Quddoos, 2021). Abnormal returns are important because they isolate the impact of the event (such as a natural disaster or social unrest) by controlling for normal market movements (Indupurnahayu *et al.*, 2022). Abnormal returns were measured by subtracting the expected return (mean returns under normal period) from the actual return (mean returns under distressed period.) By comparing the abnormal returns, the level of risk (standard deviation) and the shape ratio during the distressed period to those in normal period, the robustness and effectiveness of the machine learning optimization techniques in handling market shocks.

By using machine learning models to optimize portfolios before and during the distressed period, and then comparing their performance, thus effectively conducting an event study. The "event" is the transition from normal to distressed market conditions. In this chapter, the performance was measured by comparing how the machine learning-optimized portfolios performed relative to expectations based on their performance during normal conditions and also compare the performance across different models. The event study methodology was applicable and effectively used in this chapter because the distressed period was used as an "event" and used the framework to analyse changes in portfolio performance around this event as well as aligning with the chapter's objective.

In this study, different market conditions were defined and identified through an event study methodology, which was used to distinguish between normal and distressed market periods. Rather than applying econometric regime-switching models such as Markov-Switching or threshold models, the study adopted a qualitative and event-based approach grounded in real-world economic shocks relevant to the South African context. Specifically, three significant events were selected as proxies for distressed market conditions: the COVID-19 pandemic (March 2020 to July 2022), the July 2021 social unrest and riots, and the April 2022 KwaZulu-Natal floods. These events were chosen based on their severe economic and financial impacts, as well as the disruption they caused in market liquidity, investor confidence, and ETF price behaviour. Market conditions outside these event windows were classified as normal periods. The classification thus enabled a comparative analysis of portfolio performance under systematically different stress environments. This event-driven regime definition allowed for evaluating the adaptability and robustness of machine learning optimization models across both stable and crisis periods, providing insights into their practical effectiveness in real-world, volatile markets.

6.3.3 Empirical Results and Analysis

The growing interest in machine learning techniques for portfolio optimization stems from their ability to uncover complex patterns, adapt to changing market dynamics, and address limitations of traditional models. The results presented below aim to provide insights into how machine learning-driven portfolios perform under various market conditions, particularly during periods of distress. These findings contribute to the growing body of research on the practical utility and robustness of machine learning methods in the field of portfolio management specifically in ETFs.

Figure 6.1 presents the performance of ETF portfolios optimized using three different machine learning techniques K4-RANN, Genetic Algorithm, and XGBoost under normal and distressed market conditions. The metrics considered are mean returns, standard deviation (SD) and Sharpe Ratio. The findings assess the performance of ETF portfolios constructed using machine learning techniques (K4-RANN, GA, and XGBoost) under normal and distressed market conditions based on standard deviation (risk), mean returns, and Sharpe ratio (risk-adjusted returns). K4-RANN exhibited higher risk during distressed conditions (standard deviation increased from 0.0568 to

0.0886), with a slight improvement in mean returns (0.0057 to 0.0084) and a relatively stable Sharpe ratio (0.1004 to 0.0948). In contrast, GA demonstrated exceptional performance in distressed markets, with a dramatic reduction in risk (standard deviation decreased from 0.0819 to 0.0209), a rise in mean returns (0.0523 to 0.0648), and a sharp increase in the Sharpe ratio (0.6386 to 3.1005), indicating superior risk-adjusted returns and strong adaptability. Similarly, XGBoost showed resilience under distress, with a decrease in risk (0.0687 to 0.0369), an improvement in mean returns (0.0817 to 0.0924), and a significant increase in the Sharpe ratio (1.1892 to 2.5041), reflecting robust performance. Overall, GA and XGBoost outperformed K4-RANN during distressed conditions, with GA emerging as the most effective technique for managing risk and delivering exceptional risk-adjusted returns, followed by XGBoost. While K4-RANN maintained consistency, its higher volatility during market stress indicates potential areas for improvement.

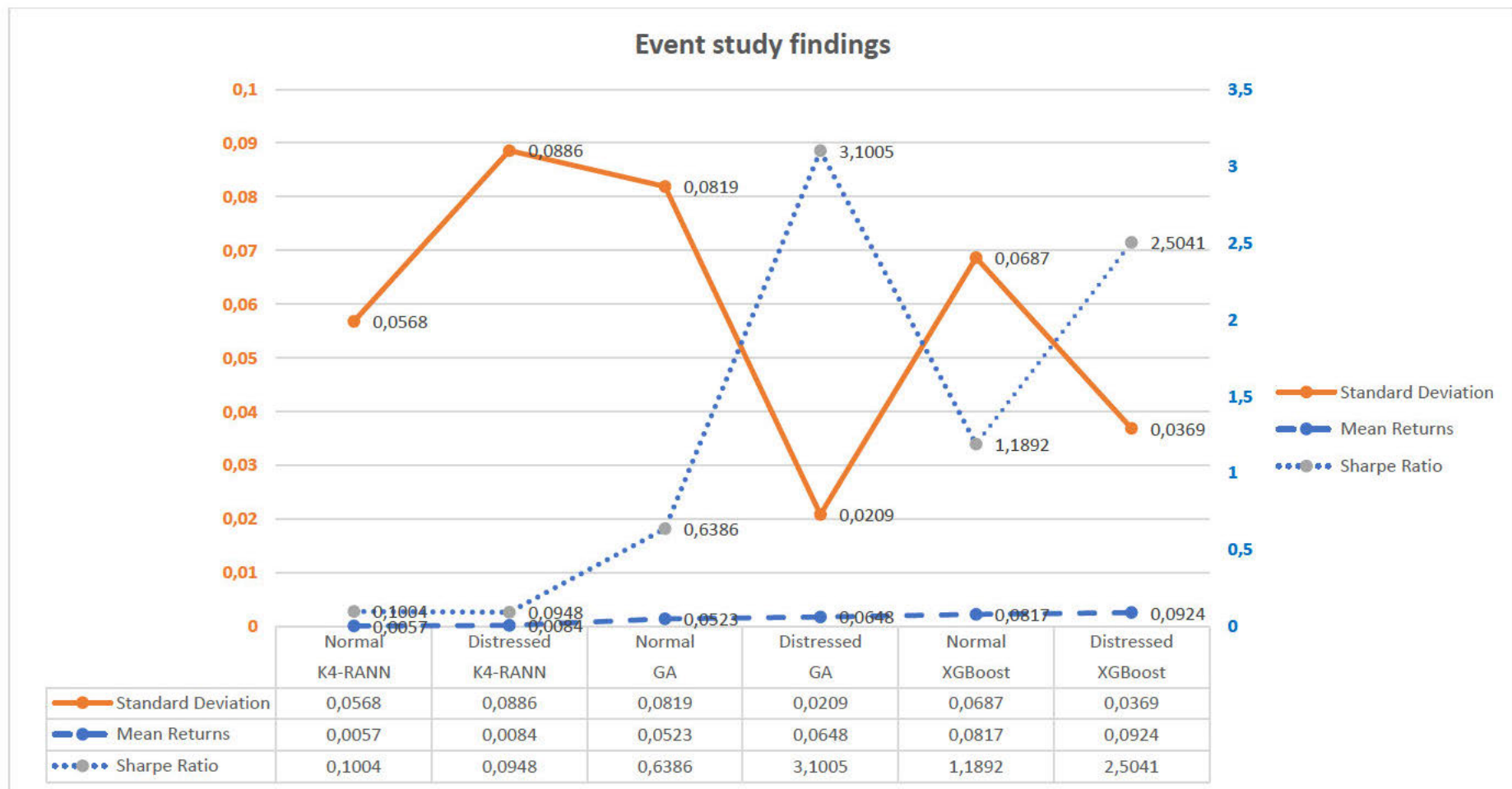


Figure 6.1 Event Study Findings

Source: Author's own estimations (2024)

The event study findings present an insightful comparison of the performance of ETF portfolios optimized using three machine learning techniques K4-RANN, Genetic Algorithm and XGBoost under normal and distressed market conditions. The key metrics considered in this analysis to measure the performance are mean portfolio returns, standard deviation and Sharpe ratio. The results highlight the strengths and weakness of each model, providing valuable guidance for ETF portfolio optimization depending on market conditions and investor preferences.

The results also highlight the importance of optimizing ETF portfolios using machine learning models, particularly under distressed market conditions where traditional methods may struggle to adapt. The event study findings indicate that all three machine learning portfolio optimization models—K4-RANN, Genetic Algorithm (GA), and XGBoost—exhibited positive abnormal returns, meaning they generated higher mean returns during distressed periods compared to normal conditions. GA demonstrated the highest increase in mean returns (0.0125), followed closely by XGBoost (0.0107), while K4-RANN had the smallest abnormal return (0.0027), suggesting it was the least robust under distressed conditions. In terms of risk, both K4-RANN and XGBoost experienced an increase in standard deviation, whereas GA had the lowest standard deviation in the distressed period (0.0209), indicating better risk control. The Sharpe ratio analysis further supports GA's superiority, as it saw the most significant improvement (0.6386 to 3.1005), followed by XGBoost (1.1892 to 2.5041), while K4-RANN's Sharpe ratio slightly declined (0.1004 to 0.0948), meaning its return increase came with disproportionately higher risk. Overall, GA emerged as the most effective model under market stress, balancing return gains with controlled risk and enhanced risk-adjusted performance, while XGBoost also performed well. In contrast, K4-RANN proved to be the least resilient to distressed conditions, with a minimal increase in returns and a decline in risk-adjusted performance.

These results emphasize the critical role of machine learning models in constructing resilient ETF portfolios, particularly during periods of heightened market stress. Machine learning techniques like GA and XGBoost demonstrated superior adaptability, offering both risk reduction and improved risk-adjusted returns compared to K4-RANN. By leveraging machine learning models, investors can enhance portfolio performance, mitigate risks, and achieve better diversification, even during turbulent market

conditions. This underscores the growing importance of integrating advanced optimization methods to navigate the complexities of financial markets and enhance decision-making processes for ETF investments.

6.3.4 Performance Comparison During Normal and Distressed Periods

The performance comparison graphs for the three machine learning optimization models (K4-RANN, Genetic algorithm and XGBoost) during normal and distressed periods reveals important information about their behaviour over time. Figures 6.2 and 6.3 support the numerical estimations in figure 6.1 by showing the visual performance of these selected machine learning optimization models.

Figure 6.2 depicts the optimized portfolio returns from the three selected machine learning optimization models (K4-RANN which is labelled KNN_RBF, GA and XGBoost also labelled XGB) over the normal period from 2017-2020. The models compared the actual (which is for all the 33 unoptimized ETF portfolio) represented by the red line. The portfolio returns by the Genetic Algorithm optimization model are presented by the green line. The blue line represents the optimized portfolio returns by the K4-RANN model and lastly, the purple line shows the optimized portfolio returns by the XGBoost model.

All models experienced times of fluctuations in returns with actual portfolio returns exhibiting extreme movements indicated by the long spikes and declines. This implies that unoptimized portfolio returns were more volatile. The Genetic algorithm model appeared to be more stable with few spikes and declines compared to the actual or unoptimized portfolio returns. There are times, particularly around 2019, when the GA showed a more noticeable decline indicating more losses relative to other models. On the other hand, for most of the period, K4-RANN and XGBoost were tracking each other as well as the actual portfolio returns. When comparing with actual portfolio returns, XGBoost and K4-RANN underperformed somewhat at times because both models showed less portfolio returns than the actual portfolio returns during some periods. In terms of stability of the models and their ability to withstand volatile periods, portfolio returns from the three machine learning optimization models (GA, K4-RANN and XGBoost) exhibit more stability (thus less volatile) as compared to the actual

portfolio returns. This demonstrates that machine learning optimization techniques are helpful at lowering return volatility though GA model showed more variation.

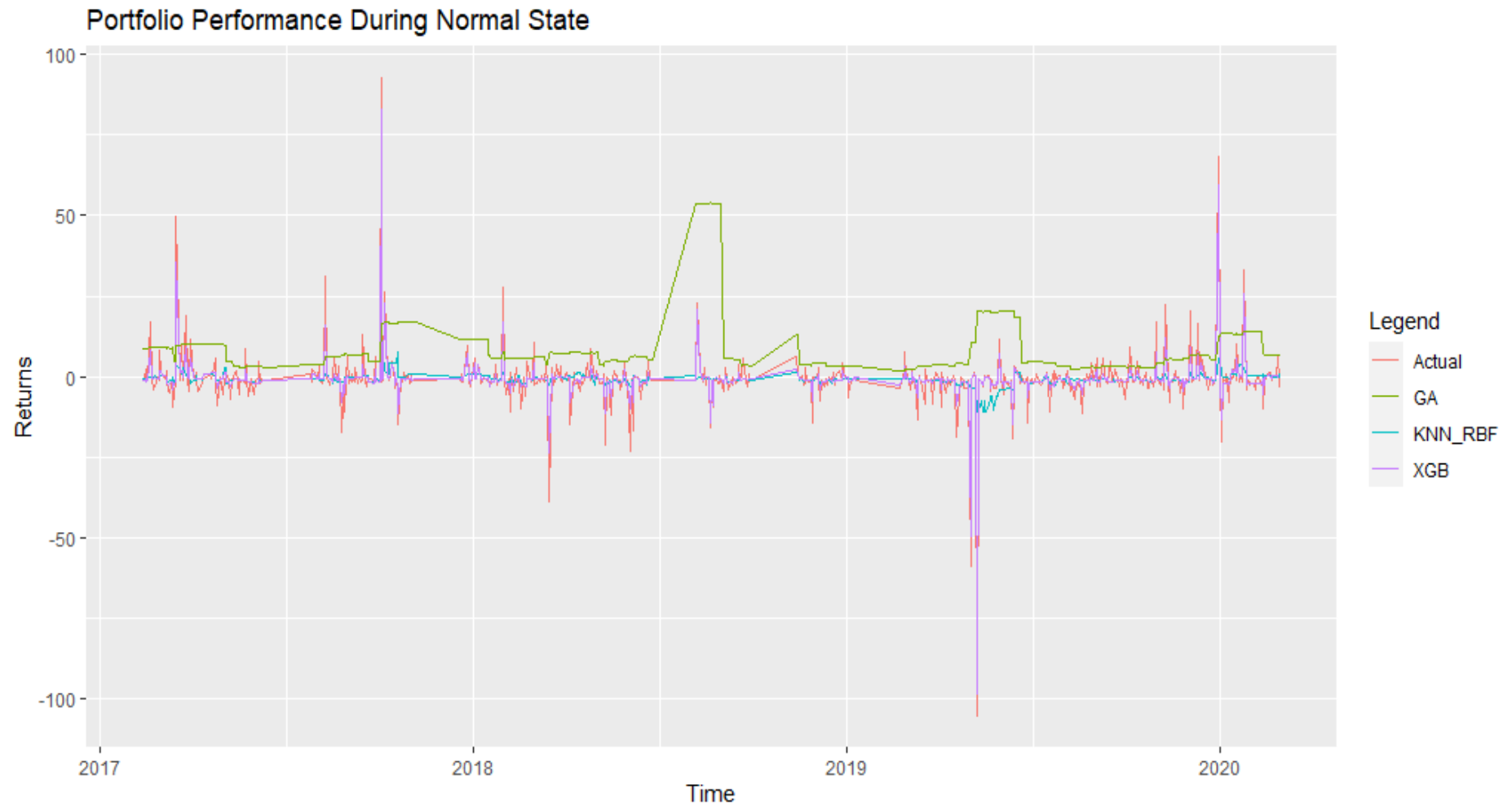


Figure 6.2 Performance under Normal Conditions

Source: Author's estimation (2024)

The second part of the objective aims to uncover the model's performance, from the results shown in figure 6.2, GA model performs more consistently, with less portfolio return spikes. However, during some periods, GA deviated more significantly as compared to the actual portfolio returns. This implies that while it lowers volatility, it may also lose out on capturing some market movements. K4-RANN and XGBoost had similar performance, with both displaying a smoother portfolio returns pattern than the unoptimized portfolio returns (actual). XGBoost may offer an added advantage in terms of higher returns in certain periods but both models performed well in terms lowering the volatility.

Overall, the selected machine learning optimization models performed well in terms of minimizing volatility as compared to the actual or unoptimized portfolio returns. However, there is a trade-off; they may reduce the opportunity for higher returns during specific periods. The Genetic Algorithm exhibited some significant volatility while K4-RANN and XGBoost showed more consistent performance with lower volatility, thus showing their potential effectiveness in optimizing ETF portfolios during normal market conditions.

Figure 6.3 shows the portfolio returns from three machine learning optimization techniques that were employed during the distressed market conditions. The GA model is represented in green line, K4-RANN is in blue line, XGBoost is in purple and the actual (unoptimized) is in red. During the distressed period, the models' performance was quite consistent with mean portfolio returns being around zero. This shows that during the difficult period, the machine learning optimization models primarily maintain the ETF portfolio values without making significant gains or losses. The ETF portfolio returns showed several strong spikes and dips, which were most visible in the XGBoost (purple line) while the GA model showed more stability without long spikes. These spikes indicate that the models occasionally respond to market fluctuations resulting in high short-term gains or losses.

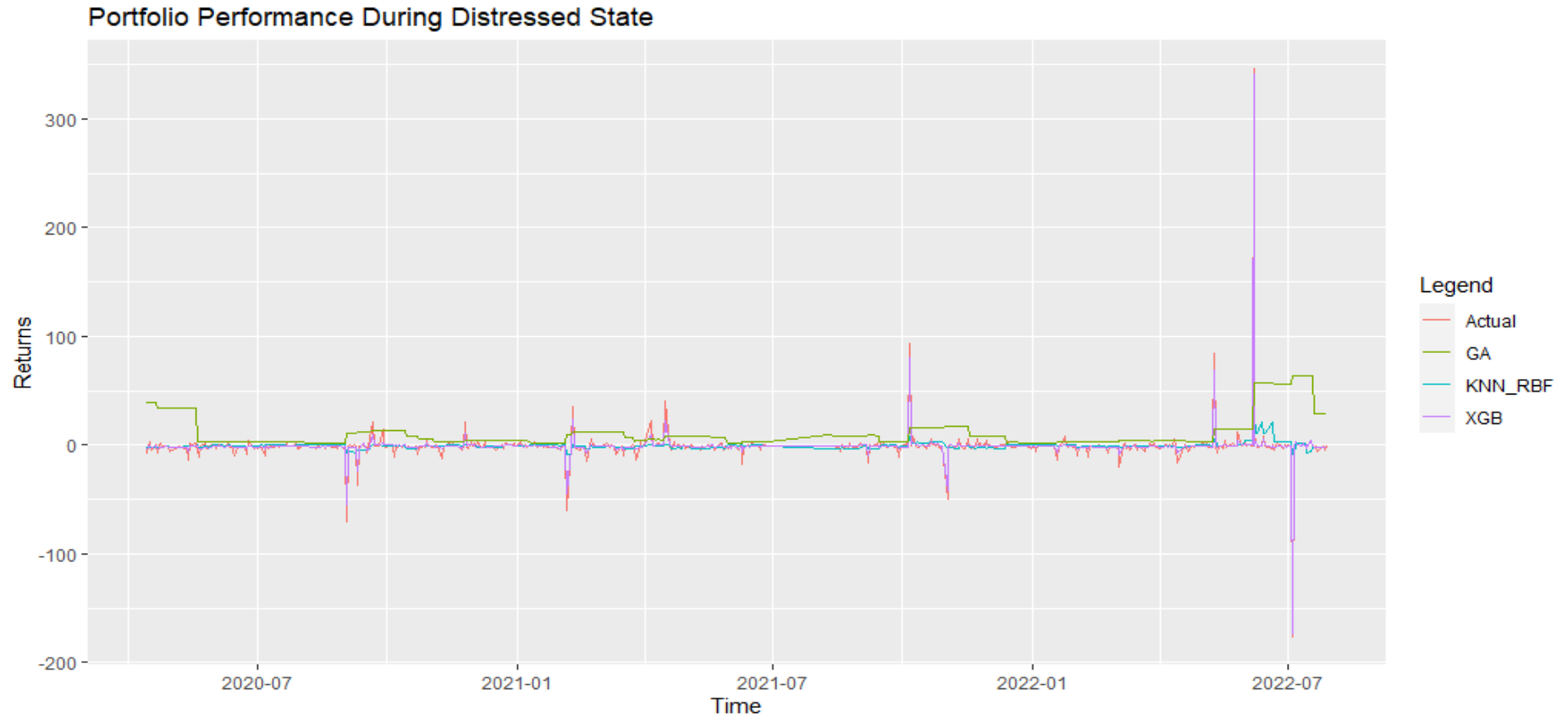


Figure 6.2 Performance under Distressed Conditions

Source: Author's own estimation (2024)

Unoptimized ETF portfolio represented by the red line showed similar pattern, but the spikes and dips were lower as compared to XGBoost and this implies that XGBoost may have the potential to produce high returns during the optimizing ETF portfolios at certain events, but it also carries a higher risk. K4-RANN model appears to follow the same path, and the actual portfolio returns implying that it may be more closely related to the actual market performance during the distressed period. Figure 6.3 also shows that GA model offers a balance between stability and responsiveness to negative market fluctuations. GA model is more stable as compared to the XGBoost though it does capture upwards tendencies during few occurrences. From the three models XGBoost showed high volatility, and this implies that it has a high risk of leading the ETF portfolios to incur major losses therefore a more aggressive technique would recommend where increased risk is acceptable.

Overall, GA and K4-RANN models give more stability when optimizing ETF portfolios under extreme market conditions in terms of volatility of portfolio returns. Moreover, XGBoost is potentially more profitable as well as exposing portfolio returns to a greater risk resulting from distressed market conditions. Therefore, under distressed market conditions, the recommended mode would be determined by the ETF's risk tolerance.

Therefore, the performance comparison plots for the different models during normal and distressed periods provide additional insights into their behaviour over time. During the normal period from January 2017 to February 2020, the three machine learning optimization models shows spikes and dips with the actual (unoptimized) ETF portfolio returns exhibiting high volatility. The GA model shows more stability than the actual ETF portfolio returns while K4-RANN and XGBoost shows similar and producing stable returns than the actual ETF portfolio returns implying that they effectively optimize the ETF portfolios as well as reducing volatility. However, there is a trade-off because they may limit the chance of realising higher ETF portfolio returns during specific periods. Overall, the GA model appears to be more controlled, although K4-RANN and XGBoost deliver consistent high ETF portfolio returns with lower volatility.

During the distressed time from March 2020 to July 2022, the models managed to maintain the ETF portfolio value, demonstrating stability in distressed market

conditions. However, the spikes and dips were displayed by the XGBoost model indicating that it responds more to market conditions. The Genetic algorithm model showed more stability than XGBoost. The XGBoost is shown being more volatile but realising highest mean average returns supporting the estimations in Table 6.1. This implies that it is a good model for capturing severe market swings but also has the chance to produce higher ETF portfolio returns.

6.4 Summary and Conclusion

This chapter compares the performance of ETF portfolios optimized using three machine learning optimization models (K4-RANN, Genetic Algorithm and XGBoost) under normal and distressed market conditions. The portfolio performance metrics utilised were mean returns, standard deviation and the Sharpe ratio as well as comparing each model's ability to realise high risk adjusted returns. During distressed market conditions, the K4-RANN model outperforms normal market conditions in terms of risk-adjusted performance, with slightly greater returns and better risk management. However, at times of turbulence, volatility remains significantly higher. This makes it a good option for investors who prioritize stability and risk-adjusted performance, especially during market distress. The GA model produces high returns in both normal and distressed periods, but at the expense of much increased volatility. The risk-adjusted performance falls during the distressed time due to excessive risk. The substantial increase in standard deviation and the corresponding decrease in Sharpe ratio indicate that GA model might be better suited for investors with a higher risk tolerance who are willing to accept increased volatility in pursuit of high returns. In terms of risk-adjusted returns, XGBoost outperforms the other models both during normal and distressed periods. It maintains a decent mix between increased returns and average risk, even during market downturns, though volatility rises dramatically. The model's ability to maintain a high Sharpe Ratio even in difficult conditions demonstrates its efficiency in optimizing ETF portfolios.

Overall, the findings emphasize the importance of tailoring machine learning portfolio optimization to both market conditions and the risk preferences of investors. While XGBoost appears to offer the most balanced approach, K4-RANN provides a safer alternative for those concerned with risk, and GA offers opportunities for higher returns at the cost of greater risk. In terms of the best risk-adjusted performance, XGBoost

model performed the best during both normal and distressed periods when optimizing ETF portfolios. To add on, Genetic Algorithm exhibits that it is the riskiest model due to the fact that during the distressed periods it is able to optimize ETF portfolios by realising high returns, but the risk level also increases making it less favourable for distressed periods. XGBoost is recommended for those seeking higher returns with manageable risk, while K4-RANN is more suited for investors preferring a diversified approach. GA should be approached cautiously due to its high volatility. Therefore, the analysis underlines the importance of considering both return and risk in ETF portfolio optimization, particularly during periods of market distress.

CHAPTER 7: SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

7.1 Introduction

ETFs have indeed been a hot topic in finance for quite some time. While they offer many benefits, including diversification, liquidity, and relatively low costs, they also present challenges and potential risks. One ongoing concern is the sheer variety of ETFs available and the complexity it introduces for investors. With ETFs covering everything from broad market indices to niche sectors and commodities, investors may find it difficult to navigate and choose the right funds for their portfolios. This complexity can lead to confusion and even poor investment decisions especially distressed market conditions. Despite the increasing popularity and utilization of machine learning techniques in portfolio optimization, there is a notable absence of empirical studies comparing the performance of ML-based approaches with traditional methods, specifically focusing on ETF portfolios, especially in times of market distress.

Existing literature has highlighted the limitations of traditional approaches, particularly in handling uncertainty in expected returns and instability in asset correlations during turbulent market periods. However, there is limited research investigating whether ML-based approaches offer superior performance, resilience, and adaptability in managing ETF portfolios under such adverse market conditions. Additionally, while ML has been applied in various domains within Africa and South Africa, its application in portfolio optimization, particularly in the context of ETF portfolios, remains unexplored within the South African literature. Therefore, there is a pressing need to bridge this research gap by conducting empirical studies that rigorously compare the performance of traditional and ML-based portfolio optimization techniques, particularly focusing on ETF portfolios, under varying market conditions, including periods of distress and volatility. Such research efforts would provide valuable insights for investors and asset managers in effectively managing and allocating capital in ETF portfolios, especially during challenging market environments.

To add on, ML techniques are solely noticeable in developed countries such as the United States, China, England, and Japan mainly focusing stock returns predictions. Contrariwise, developing countries like South Africa exhibit fragmented research concerning the utilization of ML techniques and its comparability with conventional models in ETFs portfolio optimization. This observation was made by Habbab and

Kampouridis (2022) and Sindi (2023), who assert that the integration of ML techniques could substantially augment the discernment of investors, analysts, and policy makers. Shivaprakash et al. (2022) contend that if developing countries procrastinate in embracing ML techniques, they may encounter competitive disadvantage, as early adopters of such techniques gain a competitive edge by enhancing their processes, products, and services. Delaying the implementation of ML may result in falling behind competitors who utilize this technology to make data-driven decisions and optimize their operations.

To this end, this study aimed to systematically review and comparatively analyse the implementation and performance of machine learning techniques in portfolio optimization across different asset classes. The analysis comparatively assessed the performance of the traditional versus machine learning portfolio optimisation techniques in exchange traded funds. This analysis further assessed how the exchange traded funds' portfolios constructed by machine learning techniques perform under distressed market conditions. Therefore, this doctoral dissertation also investigated the effectiveness of machine learning in optimizing ETF portfolios under distressed market conditions addresses both practical concerns of investors and academic interests in advancing financial modelling and portfolio management techniques.

7.2 Summary of Key Findings

This section summarises the findings from the three analytical chapters which analysed machine learning vs traditional portfolio optimisation techniques under different market conditions in Exchange Traded Fund portfolios.

7.2.1 Machine Learning Portfolio Optimization Across Different Asset Classes

A comparative analysis of machine learning techniques that have been used in portfolio optimization across different classes was investigated in chapter 4. The results provided in section 4.4 indicated that machine learning models such as Random Forest, Long Short-Term Memory, Support Vector Machines and ensemble methods exhibit varied performance across asset classes and evaluation metrics. RF and LSTM consistently outperform others in handling large datasets and time series data, respectively. Ensemble models combining multiple algorithms showed superior

results in portfolio optimization across asset classes like equities and sector indices, indicating their potential for enhanced prediction accuracy. The studies reviewed span multiple regions, with a focus on countries like the U.S., China, the U.K., Brazil, India, and Vietnam. This highlights the global relevance of ML in financial markets and its adaptability to diverse market conditions.

Key metrics used include the Sharpe ratio, Sortino ratio, annual returns, standard deviation, and root mean square error. These metrics help measure the models' ability to balance risk and return effectively. Models like LSTM, RF, and ensemble methods frequently maximize these risk-adjusted metrics. ML models have been applied to individual stocks, sector-specific stocks, and market indices. LSTM and RF are particularly effective in predicting trends in broad market indices like the S&P 500 and FTSE 100. However, there is a gap in research on the application of ML techniques to exchange-traded funds.

7.2.2 Traditional vs Machine Learning Portfolio Optimization Techniques in ETF Portfolios

In Chapter 5, the study compared the performance of three traditional portfolio optimization models (Markowitz, CVaR, and MAD) with three machine learning portfolio optimization models (K4-RANN, Evolutionary Genetic Algorithms, and XGBoost) in optimizing ETF portfolios. Traditional models like Markowitz, CVaR, and MAD allocated different weights to the ETFs, focusing on balancing risk and return. For example, the Markowitz model allocated more weight to ETFs with stable returns like Satrix Defensive EQ ETF, while the CVaR model focused on managing extreme losses, allocating a larger proportion to safer ETFs. Machine learning models, such as K4-RANN and XGBoost, demonstrated higher flexibility in handling complex datasets and relationships. K4-RANN focused on identifying patterns and relationships among ETFs, selecting 21 ETFs, with a particular focus on diversification across asset classes. Genetic Algorithms performed well in balancing risk and return, selecting 29 ETFs, with higher weights allocated to Satrix RESI Portfolio (STXRES) and Satrix Shariah T40 ETF (STXSHA).

Machine learning models generally provided superior risk-adjusted returns compared to traditional techniques. The Sharpe ratio was consistently higher in machine learning

models like K4-RANN and Genetic Algorithms, indicating better risk-adjusted returns. The XGBoost model yielded the highest Sharpe ratio, demonstrating its capacity to optimize returns relative to risk, surpassing traditional models like CVaR and MAD. Machine learning techniques offered more effective diversification strategies than traditional methods. For example, K4-RANN and Genetic Algorithms incorporated diverse asset classes, which reduced portfolio risk while maintaining expected returns. Traditional methods like CVaR focused heavily on risk minimization, selecting ETFs that provided more protection against extreme market conditions. However, they did not offer as much diversification as machine learning models.

Therefore, machine learning models, such as Evolutionary Genetic Algorithms and K4-RANN, adapted better to changing market conditions, outperforming traditional methods in periods of market distress. This adaptability is a crucial advantage in dynamic financial environments like ETFs. Traditional models were less adaptive due to their reliance on historical data assumptions, while machine learning models captured non-linear relationships and changing patterns more effectively.

7.2.3 ETF Portfolios Performance Under Distressed Market Conditions

The results on assessment on how the ETF portfolios optimized by machine learning optimization techniques performed under distressed market conditions are outlined in section 6.5.3 in chapter 6. Three machine learning portfolio optimization models (K4-RANN, Genetic Algorithm and XGBoost) were evaluated in terms of their performance in optimizing ETF portfolios during normal and distressed market conditions.

XGBoost consistently produced the highest returns in both normal and distressed conditions, making it the most robust model. Its mean returns increased with a relatively controlled rise in standard deviation, demonstrating a good balance between risk and return even in challenging market environments. K4-RANN maintained stable risk-adjusted returns, showing improved Sharpe ratios during distressed periods. Despite a moderate increase in standard deviation, it performed well in preserving risk-adjusted returns, making it suitable for risk-averse investors. Genetic Algorithm produced high returns but with significantly increased volatility during distressed conditions. Although the mean returns increased and the standard deviation also increased sharply, making it less favourable for risk-averse investors.

In terms of the risk-return trade-offs, the Genetic Algorithm showed a classic risk-return trade-off, with higher returns accompanied by higher risk during distressed periods. The decrease in its Sharpe ratio during distress indicates that, while returns increased, they did not adequately compensate for the added risk. XGBoost offered the best balance between risk and return, even during periods of heightened market uncertainty. Its ability to maintain a high Sharpe ratio during distressed periods demonstrates its effectiveness in handling market volatility while delivering strong returns. Comparing the models according to the volatility and market conditions, all models exhibited increased volatility during distressed periods, but XGBoost managed to maintain a relatively moderate increase in risk compared to GA. This implies that XGBoost is better at controlling volatility, making it suitable for investors seeking a balance between risk and return. The K4-RANN model performed best in terms of risk-adjusted returns, suggesting that it is more capable of preserving portfolio stability during volatile markets, even if it does not always maximize returns.

Focusing on the model suitability based on investor risk tolerance, XGBoost is the most versatile model, suitable for investors seeking both higher returns and manageable risk, making it ideal for those who can tolerate some volatility in exchange for stronger returns. K4-RANN is recommended for risk-averse investors, particularly those focused on maintaining stable risk-adjusted returns during periods of market distress. Genetic Algorithm should be approached with caution due to its higher volatility and risk during distressed periods, making it more appropriate for aggressive investors willing to take on significant risk for higher potential returns.

These results have implications for ETF portfolio management, XGBoost offers a strong case for its use in ETF portfolio optimization due to its superior ability to balance risk and return, especially during volatile markets. This makes it a compelling option for portfolio managers looking for robust performance across different market conditions. K4-RANN is valuable for investors prioritizing risk management and consistent performance, as it demonstrated stable returns with improved risk-adjusted metrics during periods of market distress. Genetic Algorithm provides opportunities for high returns, but its increased volatility during market turmoil suggests it is more suitable for high-risk, high-reward strategies. Therefore, XGBoost emerged as the top-performing model under both normal and distressed conditions, offering the best

combination of risk control and return optimization. K4-RANN is better suited for investors prioritizing stability, while Genetic Algorithm can be considered by those seeking aggressive returns despite higher risks. These findings underscore the importance of tailoring portfolio optimization strategies to investor risk tolerance and market conditions. Overall, these findings highlight the importance of aligning portfolio optimization strategies with investor risk profiles and market conditions to maximize performance and stability across different scenarios.

7.3 Conclusions and Recommendations

The study concludes that machine learning techniques present a compelling alternative to traditional portfolio optimization methods, particularly when applied to ETF portfolios. Across the three objectives, the findings consistently demonstrate that ML models such as K4-RANN, Genetic Algorithms, and Extreme Gradient Boosting outperform traditional models namely the Markowitz Model, Conditional Value-at-Risk, and Mean Absolute Deviation in terms of delivering higher risk-adjusted returns, improved diversification, and better adaptability under dynamic market conditions. Notably, XGBoost emerged as the most balanced and robust performer during periods of market distress, including the COVID-19 pandemic, the July 2021 South African social unrest, and the April 2022 KwaZulu-Natal floods. K4-RANN offered portfolio stability for risk-averse investors, while GA generated high returns but with greater volatility, making it more suitable for aggressive strategies. These results underscore the potential of ML to improve the precision and responsiveness of investment strategies, especially in emerging markets where market shocks are more frequent and pronounced.

In light of these findings, the following recommendations are proposed. First, institutional investors, fund managers, and asset management firms should consider integrating ML models particularly ensemble and neural network-based algorithms into their portfolio construction and optimization frameworks. The adoption of ML tools can enhance decision-making processes and enable more responsive and data-driven investment strategies. Second, given the volatility of emerging markets such as South Africa, investment strategies should be stress-tested using localized economic and socio-political events to ensure portfolio resilience. Third, financial institutions and policymakers should invest in the development of data infrastructure, machine learning

capabilities, and technical expertise to support the implementation of these advanced techniques. Lastly, future research is encouraged to explore hybrid optimization frameworks that combine traditional financial theories with machine learning models to achieve even greater robustness and flexibility in portfolio management. By embracing ML techniques, stakeholders can better navigate uncertainty, improve portfolio outcomes, and drive innovation in financial markets.

7.4 Original Contributions of the Thesis

7.4.1 Original Contribution from Objective 1

The original contribution from Objective 1 lies in the comprehensive and structured Systematic Literature Review that critically evaluates how machine learning techniques have been implemented in portfolio optimization across diverse asset classes globally. Unlike prior fragmented or asset-specific reviews, this study presents a consolidated and comparative synthesis that spans equity, bond, commodity, multi-asset, and alternative investment portfolios. It introduces a unique SLR framework that classifies ML models such as supervised, unsupervised, reinforcement learning, ensemble, and deep learning approaches while evaluating their methodological rigor, implementation trends, performance metrics, and data requirements. The study identifies under-researched areas such as the application of ML in fund-of-funds and Exchange Traded Fund portfolios and offers an insightful thematic map showing the geographical concentration and sectoral implementation of ML techniques. This extensive mapping not only highlights the dominance of ML research in developed markets like the U.S., China, and the U.K., but also draws attention to the research gap in emerging economies such as South Africa. Furthermore, the review critiques inconsistencies in performance evaluation and data quality across existing studies, thereby laying a solid foundation for establishing standardized benchmarking practices in ML-based portfolio optimization. By aggregating fragmented knowledge and identifying empirical gaps, Objective 1 contributes methodologically, theoretically, and contextually to the growing field of financial data science and investment analytics.

7.4.2 Original Contribution from Objective 2

The original contribution from Objective 2 is the first empirical, model-to-model comparative assessment of traditional versus ML-based portfolio optimization techniques within ETF portfolios, using South African market data. The study goes

beyond theoretical evaluation by applying a consistent performance measurement framework across six models: three traditional (Markowitz Model, CVaR, MAD) and three ML models (K4-RANN, Genetic Algorithms, and XGBoost). It uses historical daily returns of 33 JSE-listed ETFs over a ten-year period (2013–2023), offering a uniquely long-term and data-rich perspective tailored to the structure and behaviour of ETFs in an emerging market. By using standardized performance indicators mean return, standard deviation, and Sharpe ratio the research isolates the effectiveness of each model in enhancing portfolio returns, minimizing risk, and promoting diversification. The results reveal that ML models, especially GA and XGBoost, consistently deliver superior risk-adjusted returns and are more responsive to market dynamics compared to traditional models, which often underperform under non-normal or volatile conditions. This empirical evidence challenges the long-standing reliance on traditional optimization techniques and provides actionable insights for investors and fund managers seeking to adopt algorithmic strategies in ETF portfolio construction. The study's originality is also evident in its context-specific contribution it fills a significant gap in South African finance literature by empirically validating the practical benefits of ML in ETF-based investing, where such comparative studies are notably scarce.

7.4.3 Original Contribution from Objective 3

Objective 3 makes a unique methodological contribution and adding knowledge to the literature by directly comparing the performance of traditional portfolio optimization techniques (Markowitz model, CVaR, and MAD) with ML-based techniques (shallow learning, deep learning, and gradient boosting), specifically within the realm of ETF portfolios. By conducting this comparative practical assessment, the study addresses a knowledge gap in the literature regarding the effectiveness of ML techniques in optimizing ETF portfolios. The study offers insights into whether ML portfolio optimization methods provide superior performance, adaptability, and resilience compared to traditional methods. The comparative analysis allows for a comprehensive evaluation of the strengths and weaknesses of each approach within the specific context of ETF portfolios. The assessment is crucial for understanding the benefits of incorporating ML techniques into optimizing ETF portfolios. The study findings can have practical implications for portfolio managers, investors, and

researchers in the financial industry. If ML methods outperform traditional techniques, it could justify the adoption of these advanced techniques in real-world portfolio management practices, potentially leading to improved investment strategies and better risk management.

7.4.4 Original Contribution from Objective 4

Objective 4 contributes new knowledge to the literature by examining the performance of ETF portfolios optimized using ML techniques under distressed market conditions. Focusing on how ML-optimized portfolios behave during periods of market turbulence and uncertainty addresses a critical aspect of portfolio management that is particularly relevant for investors and asset managers. Understanding how ML-based approaches perform under stress provides valuable insights into their robustness and suitability for managing ETF portfolios in challenging market environments. The objective also offers valuable insights into portfolio behaviour by examining the performance of ML-optimized ETF portfolios during distressed market conditions. It allows investors and asset managers to assess the resilience of these portfolios and identify potential areas for improvement. Such insights are crucial for optimising portfolio strategies and managing risk effectively. The findings from this objective are directly applicable to investors and asset managers. They provide practical guidance on the suitability of ML-based portfolio optimisation techniques for managing ETF portfolios, especially in turbulent market environments. This relevance aids knowledge in informing decision-making processes within the investment community.

7.5 Implications of the Study

The findings of this study offer rich interpretative insights into the comparative effectiveness of machine learning (ML) and traditional portfolio optimization methods, particularly in the context of ETF portfolios. From a theoretical standpoint, the study demonstrates that ML models such as K4-RANN, Genetic Algorithms (GA), and Extreme Gradient Boosting (XGBoost) are able to capture complex, non-linear patterns in financial data that traditional models like the Markowitz Model (MM), Conditional Value at Risk (CVaR), and Mean Absolute Deviation (MAD) often overlook. These ML models outperform traditional techniques in terms of delivering higher risk-adjusted returns, improved diversification, and greater adaptability across varying

market regimes. For instance, while traditional models rely on historical correlations and static assumptions, ML techniques learn and adapt from dynamic data, making them better suited for modern, volatile markets. K4-RANN emerged as the most stable model, making it a valuable tool for risk-averse investors, while GA demonstrated high return potential, albeit with increased volatility. XGBoost provided the most balanced performance, adapting effectively in both normal and distressed market conditions.

The interpretability of the study is further enhanced through its examination of ETF portfolios under South Africa-specific market shocks such as the COVID-19 pandemic, the July 2021 social unrest, and the April 2022 floods. The study finds that ML-optimized portfolios showed greater resilience and adaptability during these periods of heightened market stress, suggesting that ML techniques provide investors with more robust tools for managing unforeseen events. This has important implications for how we understand portfolio behaviour in emerging markets, where political, social, and environmental uncertainties are more frequent and often underrepresented in global financial models.

The practical implications of this research are significant for various financial stakeholders. For portfolio managers and individual investors, the findings offer concrete evidence that machine learning models can construct more efficient and diversified portfolios, even when dealing with highly similar assets such as ETFs. These models not only enhance performance in stable markets but also provide stronger protection during periods of distress. This means that investment strategies can now be more closely aligned with investor risk preferences—risk-tolerant investors may opt for high-return strategies like GA, while conservative investors may prefer the stability offered by K4-RANN. For institutional investors, the research supports the integration of ML into existing asset allocation frameworks, paving the way for more sophisticated and adaptive portfolio construction tools.

At a policy and institutional level, the study encourages financial institutions to invest in data infrastructure, algorithmic training, and the development of hybrid investment models that combine human expertise with AI-driven insights. The research also has broader implications for emerging markets, particularly South Africa, by demonstrating the feasibility and relevance of localized ML applications in financial portfolio

management. It introduces a new dimension to risk management by incorporating real-world shocks into the portfolio optimization process something often missing from traditional models.

Investors and financial institutions can use these findings to optimize portfolio management by selecting appropriate ML models based on asset class and market conditions. The effectiveness of RF and LSTM in different contexts provides a roadmap for applying ML in asset management strategies. In conclusion, ML techniques like RF, LSTM, and ensemble methods show great promise for portfolio optimization across different asset classes, but challenges like data limitations and model interpretability need to be addressed to enhance their practical applicability.

In Investors and portfolio managers perspective, investors seeking higher risk-adjusted returns and better diversification may benefit from incorporating machine learning techniques like K4-RANN and XGBoost into their ETF portfolio strategies. Machine learning techniques offer potential advantages in managing portfolios during turbulent market conditions, making them valuable for portfolio managers aiming to minimize risk while maximizing returns. Traditional models like Markowitz and CVaR are still useful for risk-averse investors who prioritize minimizing losses, but they may underperform in volatile markets compared to machine learning approaches. In summary, machine learning portfolio optimization techniques generally outperform traditional methods in managing ETF portfolios, especially in terms of risk-adjusted returns, diversification, and adaptability to market conditions. These findings provide significant insights for portfolio managers and investors looking to optimize ETF portfolios using advanced data-driven techniques.

On the other hand, the integration of machine learning models into financial decision-making, particularly in portfolio optimization, raises important ethical and regulatory considerations. ML models often function as “black boxes,” meaning their internal workings and decision logic may be opaque to end-users, raising transparency and accountability concerns in investment management. This lack of interpretability could potentially mislead investors or result in unintended biases, particularly if the input data reflects historical inequalities or market inefficiencies. Additionally, the use of personal or proprietary financial data by these algorithms must comply with data protection

regulations such as the Protection of Personal Information Act (POPIA) in South Africa and the General Data Protection Regulation (GDPR) internationally. Financial institutions deploying ML techniques are also subject to regulatory scrutiny to ensure that algorithmic trading or automated asset allocation does not create systemic risks or promote unfair practices. As such, it is critical for practitioners to implement explainable AI (XAI) frameworks, ensure robust model validation, and maintain compliance with financial regulations to foster trust, fairness, and transparency in the adoption of ML-driven portfolio management strategies.

Overall, this study makes a novel contribution by bridging the gap between academic theory and real-world application. It not only expands the academic discourse on portfolio optimization by introducing machine learning models into ETF portfolio analysis, but it also equips investors, fund managers, and policymakers with practical tools and insights for navigating increasingly complex and unpredictable financial environments. The results affirm the importance of aligning optimization techniques with market conditions and investor objectives, with ML emerging as a highly promising frontier in both theoretical finance and applied investment strategy.

7.6 Limitations of the Study and Recommendations for Further Research

The study also did not consider the impact of real-world constraints such as taxes, trading costs, or regulatory limitations, which could influence the practicality of implementing machine learning-based portfolio optimization in a live setting. Future research could include real-world trading constraints, such as liquidity issues, transaction costs, and taxes, to provide a more practical evaluation of machine learning models in portfolio optimization. Incorporating regulatory constraints that affect portfolio management, such as ESG (Environmental, Social, Governance) criteria or sectoral caps, could make the models more applicable to real-world scenarios.

Another limitation is that many machine learning models, particularly neural networks, are often considered "black boxes" due to their complexity. This lack of transparency and interpretability can hinder their acceptance by financial institutions where explainability of decisions is crucial. Future studies should focus on developing models that not only perform well but also provide insights into their decision-making

processes. Therefore, by addressing these areas, future research can further enhance the effectiveness and adoption of machine learning techniques in portfolio optimization, providing more robust tools for investors and financial institutions.

While machine learning techniques demonstrate considerable promise in portfolio optimization, their application is not without limitations. A key concern is overfitting, which occurs when a model captures noise or random fluctuations in the training data rather than the underlying structure. This results in models that perform exceptionally well in-sample but fail to generalize to new or unseen data, leading to poor out-of-sample performance. Overfitting is particularly problematic in financial datasets, which are often noisy, high-dimensional, and subject to regime shifts. Techniques such as cross-validation, regularization, and feature selection can help mitigate this risk but do not eliminate it entirely. Another notable limitation is the lack of interpretability in many ML models, especially in complex architectures such as neural networks and ensemble methods like XGBoost. These models function as "black boxes," making it difficult for portfolio managers, regulators, and investors to understand how specific decisions are made or what factors drive allocation outcomes. This opacity can hinder trust and accountability, particularly in high-stakes financial environments. As such, the use of explainable AI (XAI) techniques, combined with rigorous back testing and sensitivity analysis, is essential to ensure that ML-driven portfolio optimization models are both robust and transparent. Addressing these limitations is crucial for the practical adoption and regulatory acceptance of ML techniques in finance.

Portfolio managers, financial institutions and investors are recommended to embrace machine learning techniques for their adaptability, particularly XGBoost and K4-RANN, which handle dynamic and uncertain financial environments effectively. For investors seeking a balanced risk-return approach, XGBoost is highly recommended. Risk-averse investors should consider K4-RANN for its stability and consistent performance, while those willing to accept higher risks in exchange for potential high returns can explore Genetic Algorithms. Combining traditional and ML approaches may also offer a balanced strategy, as traditional models like CVaR provide more conservative options in managing extreme risks.

REFERENCES

- Abner, D.J., 2016. The ETF handbook: how to value and trade exchange traded funds. *John Wiley and Sons*.
- Aboussalah, A.M. and Lee, C.G., 2020. Continuous control with stacked deep dynamic recurrent reinforcement learning for portfolio optimization. *Expert Systems with Applications*, 140, pp.112891.
- Ade-Ibijola, A. and Okonkwo, C., 2023. Artificial Intelligence in Africa: Emerging Challenges. In: Eke, D.O., Wakunuma, K., Akintoye, S. (eds) *Responsible AI in Africa. Social and Cultural Studies of Robots and AI*. https://doi.org/10.1007/978-3-031-08215-3_5
- Agarwal, S. and Muppalaneni, N.B., 2022. Portfolio optimization in stocks using mean–variance optimization and the efficient frontier. *International Journal of Information Technology*, 14(6), pp.2917-2926.
- Agrawal, S., Koolen, W.M. and Juneja, S., 2021. Optimal best-arm identification methods for tail-risk measures. *Advances in Neural Information Processing Systems*, 34, pp.25578-25590.
- Ahamed, S. and Raghavendra, S.P., 2023. Fundamentals of Artificial Intelligence & Machine Learning. *Academic Guru Publishing House*.
- Akdogan, I.U., 2020. Understanding the dynamics of foreign reserve management: The central bank intervention policy and the exchange rate fundamentals. *International Economics*, 161, pp.41-55.
- Akkaya, M., 2021. Behavioral portfolio theory. In *Applying Particle Swarm Optimization: New Solutions and Cases for Optimized Portfolios*, Springer International Publishing, 9(27), pp. 29-48.
- Al Janabi, M.A., 2021. Multivariate portfolio optimization under illiquid market prospects: a review of theoretical algorithms and practical techniques for liquidity risk management. *Journal of Modelling in Management*, 16(1), pp.288-309.

- Al Janabi, M.A., 2022. Optimization algorithms and investment portfolio analytics with machine learning techniques under time-varying liquidity constraints. *Journal of Modelling in Management*, 17(3), pp.864-895.
- Alam, M.N., Alam, M.S. and Chavali, K., 2020. Stock market response during COVID-19 lockdown period in India: An event study. *The Journal of Asian Finance, Economics and Business*, 7(7), pp.131-137.
- Alexander, S., Coleman, T.F. and Li, Y., 2006. Minimizing CVaR and VaR for a portfolio of derivatives. *Journal of Banking and Finance*, 30(2), pp.583-605.
- Allioui, H. and Mourdi, Y., 2023. Exploring the full potentials of IoT for better financial growth and stability: A comprehensive survey. *Sensors*, 23(19), pp.8015.
- Alshari, H., Saleh, A.Y. and Odabaş, A., 2021. Comparison of gradient boosting decision tree algorithms for CPU performance. *Journal of Institute of Science and Technology*, 37(1), pp.157-168.
- Anadani, I., Sharma, A., Dave, D. and Sharma, A., 2023, July. A Genetic Algorithm Approach for Portfolio Optimization. *International Conference on Data Science and Applications* 6(34), pp. 113-124.
- Anagnostopoulos, K.P. and Mamanis, G., 2010. A portfolio optimization model with three objectives and discrete variables. *Computers & Operations Research*, 37(7), pp.1285-1297.
- Andreoni, A., Robb, N. and Van Huellen, S., 2021. Profitability without investment: How financialization undermines structural transformation in South Africa. *Journal of Banking and Finance* 19(12), pp. 213-236.
- Andreou, C.K., Andreou, P.C. and Lambertides, N., 2021. Financial distress risk and stock price crashes. *Journal of Corporate Finance*, 67, pp.101870.
- Antunes, J., Wanke, P., Fonseca, T. and Tan, Y., 2023. Do ESG Risk Scores Influence Financial Distress? Evidence from a Dynamic NDEA Approach. *Sustainability*, 15(9), pp.7560.

Ararat, Ç., Cesarone, F., Pinar, M.Ç. and Ricci, J.M., 2024. MAD risk parity portfolios. *Annals of Operations Research*, 336(1), pp.899-924.

Arcuri, M.C., Gandolfi, G. and Laurini, F., 2023. Robust portfolio optimization for banking foundations: a CVaR approach for asset allocation with mandatory constraints. *Cent Eur J Oper Res* 31, pp. 557–581. <https://doi.org/10.1007/s10100-022-00821-5>.

Arreola, J.L., Yagi, D. and Johnson, A.L., 2020. Insights from machine learning for evaluating production function estimators on manufacturing survey data. *Journal of Productivity Analysis*, 53, pp.181-225.

Asawa, Y.S., 2021. Modern machine learning solutions for portfolio selection. *IEEE Engineering Management Review*, 50(1), pp.94-112.

Atique, M.M.A.B., Hoque, M.N. and Uddin, M.J., 2022, September. Employee Attrition Analysis Using CatBoost. In *International Conference on Machine Intelligence and Emerging Technologies*, 57(9) pp. 644-658.

Ayodele, T.O., 2010. Types of machine learning algorithms. *New advances in machine learning*, 3(19-48), pp.5-1.

Azevedo, V., Kaiser, G.S. and Mueller, S., 2023. Stock market anomalies and machine learning across the globe. *Journal of Asset Management*, 24(5), pp.419-441.

Aziz, S. and Dowling, M., 2019. Machine learning and AI for risk management. *Springer International Publishing*, 11(43), pp. 33-50.

Aziz, S., Dowling, M., Hammami, H. and Piepenbrink, A., 2022. Machine learning in finance: A topic modelling approach. *European Financial Management*, 28(3), pp.744-770.

Babaei, G., Giudici, P. and Raffinetti, E., 2022. Explainable artificial intelligence for crypto asset allocation. *Finance Research Letters*, 47, pp.102941.

Bacon, C.R., 2021. Practical risk-adjusted performance measurement. *John Wiley and Sons*, pp. 876 – 891.

Bacon, C.R., 2023. Practical portfolio performance measurement and attribution. *John Wiley & Sons*.

Bahrammirzaee, A., 2010. A comparative survey of artificial intelligence applications in finance: artificial neural networks, expert system and hybrid intelligent systems. *Neural Computing and Applications*, 19(8), pp. 1165-1195.

Balqis, V.P., Subiyanto, S. and Supian, S., 2021. Optimizing stock portfolio with markowitz method as a reference for investment community decisions. *International Journal of Research in Community Services*, 2(2), pp.71-76.

Banerjee, S., Gupta, S., Ghosh, A. and Bandyopadhyay, G., 2022. Optimal Portfolio Selection of Equity Mutual Funds using Genetic Algorithm: An Indian Perspective. *Mathematical Statistician and Engineering Applications*, 71(4), pp.4620-4631.

Banks, E., 2004. Alternative risk transfer: integrated risk management through insurance, reinsurance, and the capital markets. *John Wiley & Sons*.

Bányai, A., Tatay, T., Thalmeiner, G. and Pataki, L., 2024. The Impact of Rebalancing Strategies on ETF Portfolio Performance. *Journal of Risk and Financial Management*, 17(12), pp.533.

Bartram, S.M., Branke, J. and Motahari, M., 2020. Artificial intelligence in asset management. *CFA Institute Research Foundation*.

Bartram, S.M., Branke, J., De Rossi, G. and Motahari, M., 2021. Machine learning for active portfolio management. *Journal of Financial Data Science*, 3(3), pp.9-30.

Bäuerle, N. and Glauner, A., 2021. Minimizing spectral risk measures applied to Markov decision processes. *Mathematical Methods of Operations Research*, 94(1), pp.35-69.

Behera, J., Pasayat, A.K., Behera, H. and Kumar, P., 2023. Prediction based mean-value-at-risk portfolio optimization using machine learning regression algorithms for multi-national stock markets. *Engineering Applications of Artificial Intelligence*, 120, pp.105843.

Belanche-Muñoz, L.A. and Wiejacha, M., 2023. Analysis of Kernel Matrices via the von Neumann Entropy and Its Relation to RVM Performances. *Entropy*, 25(1), pp.154.

Benati, S. and Conde, E., 2022. A relative robust approach on expected returns with bounded CVaR for portfolio selection. *European Journal of Operational Research*, 296(1), pp.332-352.

Bentéjac, C., Csörgő, A. and Martínez-Muñoz, G. 2021. A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, 54, pp.1937-1967.

Bera, A.K.; Park, S.Y., 2008. Optimal portfolio diversification using the maximum entropy principle. *Economics Review.*, 27, pp.484–512.

Bezzina, P., 2023. Improving portfolio construction using deep generative machine learning models applying generative models on financial market data (Master's thesis, University of Malta).

Bhatia, A., Chandani, A., Atiq, R., Mehta, M. and Divekar, R., 2021. Artificial intelligence in financial services: Qualitative research to discover robo-advisory services. *Qualitative Research in Financial Markets*, 13(5), pp.632-654.

Bhattacharya, A. and O'Hara, M., 2020. ETFs and systemic risks. *CFA Institute Research Foundation*.

Bhattacharya, A., Johnson, J., Faramarzi, A., Jindal, N. and Johnson, R.W., 2023. Marketing capability and the turnaround of financially distressed firms. *Journal of the Academy of Marketing Science*, pp.1-21.

Bianchi, M. and Briere, M., 2021. Robo-Advising: Less AI and More XAI? *Available at SSRN 3825110*.

Bigdeli, S. and Bengtsson, F., 2019. Portfolio Optimization: A DCC-GARCH forecast with implied volatility. *Available at diva-portal.org*.

Blitz, D., Hoogteijling, T., Lohre, H. and Messow, P., 2023. How can machine learning advance quantitative asset management? *The Journal of Portfolio Management*. *Available at SSRN 4321398*.

Bloch, D.A., 2024. Stocks and Options Portfolio Optimisation with Reinforcement Learning. *Available at SSRN 4682061*.

Bolton, P., Kacperczyk, M., Hong, H.G. and Vives, X., 2021. Resilience of the financial system to natural disasters. *CEPR Press*.

Boonman, T.M., 2023. Portfolio capital flows before and after the Global Financial Crisis. *Economic Modelling*, 127, pp.106440.

Bossaerts, P., Duffy, J., Rabanal, J.P., Rud, O. and Yadav, N., 2024. ETF indexing strategies and asset prices: Experimental evidence. *Available at SSRN 4788035*.

Brabazon, A., Kampouridis, M., and O'Neill, M., 2020. Applications of genetic programming to finance and economics: past, present, future. *Genetic Programming and Evolvable Machines*, 21(1), pp, 33–53.

Breitung, C., 2023. Automated stock picking using random forests. *Journal of Empirical Finance*, 72, pp.532-556.

Broby, D., 2022. The use of predictive analytics in finance. *The Journal of Finance and Data Science*, 8, pp.145-161.

Buetow Jr, G.W. and Hanke, B., 2024. The Exchange Traded Fund Landscape: Past, Present, and Future. *Journal of Portfolio Management*, 50(9), pp. 187-213.

Buetow Jr, G.W. and Hanke, B., 2024. The Exchange Traded Fund Landscape: Past, Present, and Future. *Journal of Portfolio Management*, 50(9), pp. 71-93.

Buetow Jr, G.W. and Hanke, B., 2024. The Exchange Traded Fund Landscape: Past, Present, and Future. *Journal of Portfolio Management*, 50(9), pp. 38-45.

Burghardt, E., Sewell, D. and Cavanaugh, J., 2022. Agglomerative and divisive hierarchical Bayesian clustering. *Computational Statistics and Data Analysis*, 176, pp.107566.

Calomiris, C.W. and Jaremski, M., 2024. The Puzzling Persistence of Financial Crises: A Selective Review of 2,000 Years of Evidence. *Journal of Financial Intermediation*, 6(3), pp.101090.

Calomiris, C.W. and Jaremski, M.S., 2024. *The Puzzling Persistence of Financial Crises* (No. w32213). National Bureau of Economic Research.

Can C, Li Jianyong X, Wensheng NM., 2018 Tool wear state recognition based on support vector machine and particle filter. *J Vib Shock*, 37(17): pp. 48–55+71.

Cao, Hieu K., Han K. Cao, and Binh T. Nguyen., 2020. Delafo: An efficient portfolio optimization using deep neural networks. In *Advances in Knowledge Discovery and Data Mining: 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11–14, 2020, Proceedings, Part I* 24, pp. 623-635.

Carrera-Rivera, A., Ochoa, W., Larrinaga, F. and Lasa, G., 2022. How-to conduct a systematic literature review: A quick guide for computer science research. *X*, 9, pp.101895.

Cartanyà Caro, P., 2022. *A deep learning approach to portfolio optimization* (Bachelor's thesis, Universitat Politècnica de Catalunya).

Carter, P. and Ayres, S., Investing for impact in African private equity funds.

Castro, L.D., Galvao, A.F., Montes-Rojas, G. and Olmo, J., 2022. Portfolio selection in quantile decision models. *Annals of finance*, 18(2), pp.133-181.

Cavacini, C., 2022. Asset allocation: comparison between the Markowitz approach and the Black-Litterman method (Doctoral dissertation, Politecnico di Torino).

Cavalcante, R.C., Brasileiro, R.C., Souza, V.L., Nobrega, J.P., and Oliveira, A.L., 2016. Computational intelligence and financial markets: A survey and future directions. *Expert Systems with Applications*, 55, pp.194-211.

Çetin B., 1998. Application of Markowitz Mean Variance Model in the Istanbul Stock Exchange (Doctoral dissertation, Bilkent Universitesi, Turkey).

Ceylan, I.E., 2021. The impact of firm-specific and macroeconomic factors on financial distress risk: A case study from Turkey. *Universal Journal of Accounting and Finance*, 9(3), pp.506-517.

Chan, R.H., Clark, E., Guo, X. and Wong, W.K., 2020. New development on the third-order stochastic dominance for risk-averse and risk-seeking investors with application in risk management. *Risk Management*, 22(2), pp.108-132.

Chatterjee, S., 2021. A new coefficient of correlation. *Journal of the American Statistical Association*, 116(536), pp.2009-2022.

Chaweewanchon, A. and Chaysiri, R., 2022. Markowitz Mean-Variance Portfolio Optimization with Predictive Stock Selection Using Machine Learning. *International Journal of Financial Studies*, 10(3), pp.64.

Chen, J., Dai, G. and Zhang, N., 2020. An application of sparse-group lasso regularization to equity portfolio optimization and sector selection. *Annals of Operations Research*, 284, pp.243-262.

Chen, M.Y., Chen, C.T. and Huang, S.H., 2023. Knowledge distillation for portfolio management using multi-agent reinforcement learning. *Advanced Engineering Informatics*, 57, pp.102096.

Chen, Q., Zhang, W. and Lou, Y., 2020. Forecasting stock prices using a hybrid deep learning model integrating attention mechanism, multi-layer perceptron, and bidirectional long-short term memory neural network. *IEEE Access*, 8, pp.117365-117376.

Chen, W., Zhang, H. and Jia, L., 2022. A novel two-stage method for well-diversified portfolio construction based on stock return prediction using machine learning. *The North American Journal of Economics and Finance*, 63, pp.101818.

Chen, Y., Liu, K., Xie, Y. and Hu, M., 2020. Financial trading strategy system based on machine learning. *Mathematical problems in engineering*, 2020(1), pp.3589198.

Chen, Z., 2023, February. A Study of Asset Allocation Based on the Markowitz Model. *International Conference on Business and Policy Studies* 7(31), pp. 222-229.

Chen, Z., Song, S., Wei, Z., Fang, J. and Long, J., 2021. Approximating median absolute deviation with bounded error. *Proceedings of the VLDB Endowment*, 14(11), pp.2114-2126.

Chen, T. and Guestrin, C., 2016. XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 7, pp. 785-794.

Chhajjer, P., Shah, M. and Kshirsagar, A., 2022. The applications of artificial neural networks, support vector machines, and long–short term memory for stock market prediction. *Decision Analytics Journal*, 2, pp.100015.

Choi, C. and Kim, J., 2024. Outperforming the tutor: Expert-infused deep reinforcement learning for dynamic portfolio selection of diverse assets. *Knowledge-Based Systems*, 27(4), pp.111739.

Chou, J.S. and Chen, K.E., 2024. Optimizing investment portfolios with a sequential ensemble of decision tree-based models and the FBI algorithm for efficient financial analysis. *Applied Soft Computing*, 158, pp.111550.

Chuai, G., Ma, H., Yan, J., Chen, M., Hong, N., Xue, D., Zhou, C., Zhu, C., Chen, K., Duan, B. and Gu, F., 2018. Deep CRISPR: optimized CRISPR guide RNA design by deep learning. *Genome biology*, 19, pp.1-18.

Ciaburro, G. and Iannace, G., 2021. Machine learning-based algorithms to knowledge extraction from time series data: A Review. *Data*, 6(6), pp.55.

Clements, R., 2020. New Funds, Familiar Fears: Do Exchange Traded Funds Make Markets Less Stable? Part I, Liquidity Illusions. *House of Business and Tax LJ*, 20, pp.14.

Dahal, K.R., Gupta, A. and Pokhrel, N.R., 2024. Predicting the Direction of NEPSE Index Movement with News Headlines Using Machine Learning. *Econometrics*, 12(2), pp.16.

Dahlquist, M. and Ibert, M., 2024. Equity returns expectations and portfolios: Evidence from large asset managers. *The Review of Financial Studies*, 37(6), pp.1887-1928.

Dai, Z. and Kang, J., 2022. Some new efficient mean–variance portfolio selection models. *International Journal of Finance and Economics*, 27(4), pp.4784-4796.

Dalne, G., 2023. ESG-Integrated machine learning portfolio optimization strategies: performance analysis and applications. *The Review of Financial Studies*, 7(16), pp.2315-2327.

Dash Jr, G.H. and Kajiji, N., 2021. Efficient Calibration of a Multi-Objective Artificial Network to Amplify Directional Volatility Spillovers In European Government Bond Markets. *An International Journal*, 3(1), pp, 323.

Dash, G. H., Kajiji, N., and Vonella, D., 2018. The role of supervised learning in the decision process to fair trade U.S. Municipal debt. *Euro Journal on Decision Processes*. doi: <https://doi.org/10.1007/s40070-018-0079-2>.

Dash, G.H. and Kajiji, N., 2015. Near High Frequency Production Economics for Global Wealth Creation: Automated Trading, Neuroeconomics and Trading Fundamentals. *Journal of Banking and Finance*, 21, pp. 811-823.

Dash, G.H. and Kajiji, N., 2020. January. Multiobjective Behavioural Portfolio Selection with Efficient ESG Factors and Learning Network Estimation of Asset Returns. In 2nd Frontiers of Factor Investing Conference, Lancaster University, UK. [Online]. Retrieved: <http://wp.lancs.ac.uk/fofi2020/files/2020/04/FoFI-2020-123-Gordon-Dash-Jr.pdf> [Accessed 25 Mar 2023].

Dash, G.H. and Kajiji, N., 2021. Behavioural Portfolio Management with Layered ESG Goals and Ai Estimation of Asset Returns. Available at SSRN 3953440.

Dastres, R. and Soori, M., 2021. Artificial neural network systems. *International Journal of Imaging and Robotics (IJIR)*, 21(2), pp.13-25.

Davies, R., Kat, H. & Lu, S. 2009. Fund of hedge funds portfolio selection: A multiple-objective approach. *Derivative Hedge Funds*, 15, pp. 91–115. <https://doi.org/10.1057/jdhf.2009.1>

de Prado, M.M.L., 2020. Machine learning for asset managers. *Cambridge University Press*.

De Spiegeleer, J., Höcht, S., Jakubowski, D., Reyners, S. and Schoutens, W., 2023. ESG: A new dimension in portfolio allocation. *Journal of Sustainable Finance and Investment*, 13(2), pp.827-867.

Deep, A.T., 2024. Advanced financial market forecasting: integrating Monte Carlo simulations with ensemble Machine Learning models. *Journal of Economic literature*, 49(2), 326365.

Dellermann, D., Calma, A., Lipusch, N., Weber, T., Weigel, S. and Ebel, P., 2021. The future of human-AI collaboration: a taxonomy of design knowledge for hybrid intelligence systems. arXiv preprint arXiv:2105.03354.

Devan, M., Thirunavukkarasu, K. and Shanmugam, L., 2023. Algorithmic Trading Strategies: Real-Time Data Analytics with Machine Learning. *Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online)*, 2(3), pp.522-546.

Dezhkam, A. and Manzuri, M.T., 2023. Forecasting stock market for an efficient portfolio by combining XGBoost and Hilbert–Huang transform. *Engineering Applications of Artificial Intelligence*, 118, p.105626.

Di Toro, M., 2022. Asymmetric Volatility Bias and Loss Aversion in Investment Decision-Making. *International Journal of Research and Review*, 39(3), 312–320.

Diagne, M., 2002. Financial Risk Management and Portfolio Optimization Using Artificial Neural Networks and Extreme Value Theory (Doctoral dissertation, Technische Universität Kaiserslautern).

Diao, L., Niu, D., Zang, Z. et al., 2019. Short-term Weather Forecast Based on Wavelet Denoising and Catboost, In: 2019 *Chinese Control Conference (CCC)*. *IEEE*, pp. 3760- 3764.

Dimson, E., Marsh, P. and Staunton, M., 2009. Triumph of the optimists: 101 years of global investment returns. In *Triumph of the Optimists*. Princeton University Press.

Dixit, V. and Tiwari, M.K., 2020. Project portfolio selection and scheduling optimization based on risk measure: a conditional value at risk approach. *Annals of Operations Research*, 285(1), pp.9-33.

Dixon, M.F., Halperin, I. and Bilokon, P., 2020. Machine learning in finance (Vol. 1170). New York, NY, USA: *Springer International Publishing*.

Dominguez, A.R., 2023. Portfolio optimization based on neural networks sensitivities from assets dynamics respect common drivers. *Machine Learning with Applications*, 11, pp.100447.

Dorogush, A.V., Ershov, V. and Gulin, A., 2018. CatBoost: gradient boosting with categorical features support. arXiv preprint arXiv:1810.11363.

Dowling, M. and Lucey, B., 2023. Behavioral finance moving forward. *Advanced Introduction to Behavioral Finance*, pp.144.

Duarte, F.G. and De Castro, L.N., 2020. A framework to perform asset allocation based on partitional clustering. *IEEE access*, 8, pp.110775-110788.

Dzuba, S. and Krylov, D., 2021. Cluster analysis of financial strategies of companies. *Mathematics*, 9(24), pp.3192.

Economics, F., 2021. *TOWARDS A FRAMEWORK FOR ASSET PRICING IN DEVELOPING EQUITY MARKETS* (Doctoral dissertation, University of the Witwatersrand).

Ehiedu, V.C. and Obi, C.K., 2022. Efficient Market Hypothesis and the Nigerian stock exchange in the midst of global financial crises. *International Journal of Academic Management Science Research (IJAMSR)*, 6(8), pp.263-273.

Eiben, A.E., Smith., 2015. What is an evolutionary algorithm? Introduction to evolutionary computing, pp.25-48.

Eisfeldt, A.L., Lustig, H. and Zhang, L., 2023. Complex asset markets. *The Journal of Finance*, 78(5), pp.2519-2562.

Ekmekcioglu, O., 2022. *Topics in optimization via Deep Neural Networks* (Doctoral dissertation, bilkent university).

El Hajj, M. and Hammoud, J., 2023. Unveiling the influence of artificial intelligence and machine learning on financial markets: A comprehensive analysis of AI applications in trading, risk management, and financial operations. *Journal of Risk and Financial Management*, 16(10), p.434.

Elena, P., 2021. Predicting the movement direction of omxs30 stock index using xgboost and sentiment analysis.

Emekter, R., Jirasakuldech, B. and Beaves, R., 2023. Efficient Asset Allocation for Individual Investors in the ETF World. *Financial Planning Research Journal*, 8(1), pp.79-98.

Enow, S.T., 2022. Evidence of Adaptive Market Hypothesis in International Financial Markets. *Journal of Academic Finance*, 13(2), pp.48-55.

Enow, S.T., 2023. Investigating causality and market contagion during periods of financial distress and its implications. *The Econometrics Journal*, 11(2), 308–325.

Erdaş, M.L., 2020. Developing a portfolio optimization model based on linear programming under certain constraints: An application on Borsa Istanbul 30 Index. *TESAM Akademi Dergisi*, 7(1), pp.115-141.

Eslami Ghayour, Arash and Yousefi, Ali., 2023. Performance Analysis of Gradient Boosting Models for Forex Market Prediction. Available at SSRN: <https://ssrn.com/abstract=4450688> or <http://dx.doi.org/10.2139/ssrn.4450688>.

Etchegaray, A.A., 2021. Portfolio Optimization Using Evolutionary Algorithms (Doctoral dissertation, Universidade NOVA de Lisboa. Portugal).

Fabozzi, F., Gupta, F., and Markowitz, H., 2002. The legacy of modern portfolio theory. *Journal of Investing*, 11(3), pp. 7-22.

Fan, J., Zhang, J. and Yu, K., 2012. Vast portfolio selection with gross-exposure constraints. *Journal of the American Statistical Association*, 107(498), pp.592-606.

Fang F., 2011. Research on power load forecasting based on Improved BP neural network. Harbin Institute of Technology. *The American economic review* 98(1), 394–425.

Fang, Z., Tan, T., Yan, J., Lu, Q., Pitt, M. and Hanna, S., 2022. Automated portfolio-based strategic asset management based on deep neural image classification. *Automation in Construction*, 142, p.104481.

Farooq, U., Nasir, A., Bilal and Quddoos, M.U., 2021. The impact of COVID-19 pandemic on abnormal returns of insurance firms: cross-country evidence. *Applied Economics*, 53(31), pp.3658-3678.

Fernandes, L., Kulkarni, M. and Pande, M.B., 2023, December. A Systematic Literature Review of Classical and Quantum Machine Learning Approaches for Mutual Fund Portfolio Optimization. *IEEE Pune Section International Conference (PuneCon)*, pp. 1-6.

Filippi, C., Guastaroba, G. and Speranza, M.G., 2020. Conditional value-at-risk beyond finance: a survey. *International Transactions in Operational Research*, 27(3), pp.1277-1319.

Fischer Black and Robert Litterman., 1991. Global asset allocation with equities, bonds, and currencies. Fixed Income Research. *IMF Economic Review* 61(3), 520–559.

Fischer, T.G. and Krauss, C. 2018. Statistical arbitrage in cryptocurrency markets. *Journal of Risk and Financial Management*, 12(1), pp.31.

Fligstein, N., 2021. The banks did it: An anatomy of the financial crisis. *Harvard University Press*.

Forrester, J.B. and Kalivas, J.H., 2004. Ridge regression optimization using a harmonious approach. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 18(7-8), pp.372-384.

Frank, E. and Ray, R., 2024. Machine learning in portfolio management and asset allocation. *Econometrica* 84(2), pp. 445–510.

Fransisca, D.C., Sukono, Chaerani, D. and Halim, N.A., 2024. Robust Portfolio Mean-Variance Optimization for Capital Allocation in Stock Investment Using the Genetic Algorithm: A Systematic Literature Review. *Computation*, 12(8), pp.166.

Fricke, D. and Wilke, H., 2023. Connected funds. *The Review of Financial Studies*, 36(11), pp.4546-4587.

Frigiola, A., 2024. *Supervised Contrastive Learning for Classification of Market Stock Series* (Doctoral dissertation, Politecnico di Torino).

Gallucci, C., Santulli, R. and Lagasio, V., 2022. The conceptualization of environmental, social and governance risks in portfolio studies A systematic literature review. *Socio-Economic Planning Sciences*, 84, pp.101382.

Gambeta, V. and Kwon, R., 2020. Risk return trade-off in relaxed risk parity portfolio optimization. *Journal of risk and financial management*, 13(10), pp.237.

Gan, L., Wang, H. and Yang, Z., 2020. Machine learning solutions to challenges in finance: An application to the pricing of financial products. *Technological Forecasting and Social Change*, 153, pp.119928.

Gao, G., Jiang, H., Vink, J.C., Chen, C., El Khamra, Y. and Ita, J.J., 2020. Gaussian mixture model fitting method for uncertainty quantification by conditioning to production data. *Computational Geosciences*, 24(2), pp.663-681.

Gayatri, T., Srinivasu, G., Chaitanya, D.M.K. and Sharma, V.K., 2022. A Review on Optimization Techniques of Antennas Using AI and ML/DL Algorithms. *International Journal of Advances in Microwave Technology*, 7(2), pp.288-295.

Ge, S. and Weisbach, M.S., 2021. The role of financial conditions in portfolio choices: The case of insurers. *Journal of Financial Economics*, 142(2), pp.803-830.

Geambasu, C., Sova, R., Jianu, I. and Geambasu, L., 2013. Risk measurement in post-modern portfolio theory: differences from modern portfolio theory. *Economic Computation & Economic Cybernetics Studies and Research*, 47(1), pp.113-132.

Ghahtarani, A.; Najafi, A.A., 2018. Robust optimization in portfolio selection by m-MAD model approach. *Econ. Comput. Econ. Cybern. Stud. Res.* 52, pp. 279–291.

Goldstein, I., 2023. Information in financial markets and its real effects. *Review of Finance*, 27(1), pp.1-32.

Gomes, D., Guimarães, V. and Silva, J., 2021. A Fully Automatic Gap Filling Approach for Motion Capture Trajectories. *Applied Sciences*, 11(21), pp.9847.

Goodell, J.W., Kumar, S., Lim, W.M. and Pattnaik, D., 2021. Artificial intelligence and machine learning in finance: Identifying foundations, themes, and research clusters from bibliometric analysis. *Journal of Behavioral and Experimental Finance*, 32, p.100577.

Graewe, U. Horst, and E. Séré., 2018. Smooth solutions to portfolio liquidation problems under price-sensitive market impact. *Stochastic Processes and Their Applications*, 128(3), pp. 979–1006.

Gramespacher, T. and Posth, J.A., 2021. Employing explainable AI to optimize the return target function of a loan portfolio. *Frontiers in Artificial Intelligence*, 4, pp.693022.

Grund, S., 2020. Exchange-Traded Funds (ETFs) and Systemic Risk. *Available at SSRN 3583028*.

Gu, S., Kelly, B. T., and Xiu, D., 2019. Empirical Asset Pricing via Machine Learning <http://dx.doi.org/10.2139/ssrn.3159577>.

Gu, Shihao and Kelly, Bryan T. and Xiu, Dacheng., 2019. Empirical Asset Pricing via Machine Learning. Chicago Booth Research Paper No. 18-04, 31st *Australasian Finance and Banking Conference* 2018, Available at <http://dx.doi.org/10.2139/ssrn.3159577>.

Guerra, O. and Eboreime, E., 2021. The impact of economic recessions on depression, anxiety, and trauma-related disorders and illness outcomes—a scoping review. *Behavioral Sciences*, 11(9), p.119.

Gumelar, A.B., Setyorini, H., Adi, D.P., Nilowardono, S., Widodo, A., Wibowo, A.T., Sulistyono, M.T. and Christine, E., 2020, September. Boosting the accuracy of stock market prediction using xgboost and long short-term memory. *International Seminar on Application for Technology of Information and Communication (iSemantic)* pp. 609-613.

Gunawan, T.I., 2024. Understanding Investment Decision-making: A Qualitative Inquiry into High-Frequency Trading, Investment Strategies, and Portfolio Performance in the Financial Market. *Golden Ratio of Finance Management*, 4(2), pp.131-141.

Gunjan, A. and Bhattacharyya, S., 2023. A brief review of portfolio optimization techniques. *Artificial Intelligence Review*, 56(5), pp.3847-3886.

Gupta P., Mehlawat M. K. and Mittal G., 2012., Asset portfolio optimization using support vector machines and real-coded genetic algorithm, *Journal Global Optimization*, 53(2), pp. 297-315.

Gupta, U. and Gupta, D., 2023. Least squares structural twin bounded support vector machine on class scatter. *Applied Intelligence*, 53(12), pp.15321-15351.

Habbab, F.Z. and Kampouridis, M., 2022. Optimizing Mixed-Asset Portfolios with Real Estate: Why Price Predictions? In 2022 *IEEE Congress on Evolutionary Computation*, pp. 1-8.

Habbab, F.Z. and Kampouridis, M., 2024. An in-depth investigation of five machine learning algorithms for optimizing mixed-asset portfolios including REITs. *Expert Systems with Applications*, 235, p.121102.

Haberly, D. and Wójcik, D., 2022. Sticky power: Global financial networks in the world economy, *Oxford University Press*, pp. 400.

Hansen, K.B., 2020. The virtue of simplicity: On machine learning models in algorithmic trading. *Big Data and Society*, 7(1), pp.205-209.

Harry M Markowitz.,1968. Portfolio selection. *Yale University Press*.

Harvey, C.R., Rattray, S. and Van Hemert, O., 2021. Strategic risk management: Designing portfolios and managing risk. *John Wiley and Sons*.

Haslbeck, J.M., Bringmann, L.F. and Waldorp, L.J., 2021. A tutorial on estimating time-varying vector autoregressive models. *Multivariate behavioral research*, 56(1), pp.120-149.

Hatami, Z., Ali, H. and Volkman, D., 2022, December. Evaluating portfolio performance by highlighting network property and the Sharpe ratio in the stock market. In *International Conference on Business, Economics and Finance*. 27(1), pp.49-65.

Haugh MB, Lo AW., 2001. Computational challenges in portfolio management. *Comput Sci Eng* 3(3), pp. 54–59.

He, P., Sun, Y., Zhang, Y. and Li, T., 2021. COVID–19's impact on stock prices across different sectors—An event study based on the Chinese stock market. In *Research on Pandemics* pp. 66-80.

Heinonen, S., Karjalainen, J. and Taylor, A., 2022. Landscapes of our uncertain Futures. Towards mapping and understanding crisis-related concepts and definitions. Landscapes of our uncertain Futures. Towards mapping and understanding crisis-related concepts and definitions. *FFRC eBOOKS*, 7, pp. 1272-1297

Herzog, B., Jones, J. and Safaee, S., 2023. Remember to Diversify Your Active Risk: Evidence from US Equity ETFs. *Journal of Beta Investment Strategies*, 14(2). pp. 182-191.

Hieu, L.T., 2020. Deep reinforcement learning for stock portfolio optimization. *arXiv preprint arXiv:2012.06325*.

Hill, J.M., Nadig, D. and Hougan, M., 2015. *A comprehensive guide to exchange-traded funds (ETFs)*. CFA Institute Research Foundation.

Hoe, L.W. and Siew, L.W., 2016. June. Portfolio optimization with mean-variance model. In *AIP Conference Proceedings* 1739(1). pp. 23-34.

Hongjoong, K.I.M., 2021. Mean-Variance Portfolio Optimization with Stock Return Prediction Using Xgboost. *Economic Computation and Economic Cybernetics Studies and Research*, 55(4). pp. 56-66.

https://corporate.vanguard.com/content/dam/corp/research/pdf/benefits_of_a_fund_of_funds_strategy_in_private_equity.pdf

https://corporate.vanguard.com/content/dam/corp/research/pdf/benefits_of_a_fund_of_funds_strategy_in_private_equity.pdf

<https://corporatefinanceinstitute.com/resources/career-map/sell-side/capital-markets/fund-of-funds-fof/>

<https://link.springer.com/article/10.1057/jdhf.2009.1>

<https://www.caseware.com/resources/blog/applications-for-machine-learning-in-different-sectors/>

<https://www.oldmutualinvest.com/individual/funds/multi-managed-funds/>

<https://www.proquest.com/docview/923120562?fromopenview=true&pq-origsite=gscholar&sourcetype=Scholarly%20Journals>

<https://www.uob.com.sg/personal/wealth/content-hub/invest/understanding-unit-trusts-what-they-are-and-how-they-work.html>

Hu, D., Liang, H. and Yuan, Z., 2024. Excess return and tracking errors of Chinese ETFs. *Finance Research Letters*, 67, pp.105858.

Huang, X., Guidolin, M., Newton, D., Platanakis, E. and Ye, X., 2023. Generalized Black-Litterman with Decision Fusion. *Available at SSRN*.

Huotari, T., Savolainen, J. and Collan, M., 2020. Deep reinforcement learning agent for S&P 500 stock selection. *Axioms*, 9(4), pp.130.

Ibrahim, A.A., Ridwan, R.L., Muhammed, M.M., Abdulaziz, R.O. and Saheed, G.A., 2020. Comparison of the CatBoost classifier with other machine learning methods. *International Journal of Advanced Computer Science and Applications*, 11(11). pp. 13 -19.

Indupurnahayu, I., Nurhayati, I., Endri, E., Marlina, A., Yudhawati, D. and Muniroh, L., 2022. Islamic Bank Merger and Economic Crisis: Event Study Analysis. *Quality-Access to Success*, 23(187), pp. 113-121.

Irsan, M.Y.T., Priscilla, E. and Siswanto, S., 2022. Comparison of Variance Covariance and Historical Simulation Methods to Calculate Value at Risk on Banking Stock Portfolio. *Jurnal Matematika, Statistika dan Komputasi*, 19(1), pp.241-250.

Islam, K.U. and Pandow, B.A., 2024, February. Machine Learning Approaches for Enhanced Portfolio Optimization: A Comparative Study of Regularization and Cross-Validation Techniques. In *2024 11th International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 1440-1443). IEEE.

Jacquier, A., Kondratyev, O., Lipton, A. and de Prado, M.L., 2022. Quantum Machine Learning and Optimisation in Finance: On the Road to Quantum Advantage. *Journal of Financial Economics*, 87(2), pp.388-417.

Jaeger, M., Krügel, S., Marinelli, D., Papenbrock, J. and Schwendner, P., 2021. Interpretable machine learning for diversified portfolio construction. *The Journal of Financial Data Science*, 3(3), pp.31-51.

Jana, P.; Roy, T.K.; Mazumder, S.K., 2009. Multi-objective possibilistic model for portfolio selection with transaction cost. *Journal of Computing and Applied Mathematics*, 228, pp.188–196.

Jang, J. and Seong, N., 2023. Deep reinforcement learning for stock portfolio optimization by connecting with modern portfolio theory. *Expert Systems with Applications*, 218, pp.119556.

Jani, K.H., Jones, K.A., Jones, G.W., Amiel, J., Barron, B. and Elhadad, N., 2020. Machine learning to extract communication and history-taking skills in OSCE transcripts. *Medical Education*, 54(12), pp.1159-1170.

Janiesch, C., Zschech, P. and Heinrich, K., 2021. Machine learning and deep learning. *Electronic Markets*, 31(3), pp.685-695.

Jankov, Z., 2021. Evaluation the performance of exchange traded funds (ETFs) listed on the international stock markets. *Ecoforum Journal*, 10(1). pp. 253-262.

Jayeola, D., Sufahani, S., Akinmoji, O., Abd Wahab, M.H., Idrus, S.Z.S. and Mamat, M., 2020, September. Optimal method of optimizing risk of portfolio. In *IOP Conference Series: Materials Science and Engineering* 917(1), pp. 012058).

Jhaveri, S., Khedkar, I. and Kantharia, Y., 2019. Success Prediction using Random Forest, CatBoost, XGBoost and AdaBoost for Kickstarter Campaigns, *3rd International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, pp. 1170-1173.

Jiang, X., 2010. Return dispersion and expected returns. *Financial Markets and Portfolio Management*, 24, pp.107-135.

Jiang, Z., Ji, R. and Chang, K.C., 2020. A machine learning integrated portfolio rebalance framework with risk-aversion adjustment. *Journal of Risk and Financial Management*, 13(7), p.155.

Jin. W., 2018 Human motion recognition method based on support vector machine optimization. *Electronic Designing and Engineering* 26(17). pp. 6–9+16

Jing, D., Imeni, M., Edalatpanah, S.A., Alburaikan, A. and Khalifa, H.A.E.W., 2023. Optimal selection of stock portfolios using multi-criteria decision-making methods. *Mathematics*, 11(2), p.415.

Junkyu Jang, NohYoon Seong., 2023, Deep reinforcement learning model. *Journal of Financial Services Research*, 25(1), pp.57-69.

Kaczmarek, T. and Perez, K., 2022. Building portfolios based on machine learning predictions. *Economic research-Ekonomska istraživanja*, 35(1), pp.19-37.

Kajiji, N. ,2001. Adaptation of Alternative Closed Form Regularization Parameters with Prior Information to the Radial Basis Function Neural Network for High Frequency Financial Time Series. *Journal of International Money and Finance*, 62, pp.25-51.

Kalayci, C.B., Ertenlice, O. and Akbay, M.A., 2019. A comprehensive review of deterministic models and applications for mean-variance portfolio optimization. *Expert Systems with Applications*, 125, pp.345-368.

Kaminskyi, A. and Nehrey, M., 2021. Passing through COVID-19 financial shock by Artificial Intelligence ETFs: changes in risk-return correspondence. *International Review of Economics and Finance*, 71, pp.830-852.

Kaminskyi, A., Nehrey, M. and Butylo, D., 2022. Traditional and Alternative Assets in Portfolio Management: ETF Using Approach. In *Distributed Sensing and Intelligent Systems: Proceedings of ICDSIS 2020*. pp. 41-57.

Kamoune, A. and Ibenrissoul, N., 2022. Traditional versus Behavioral Finance Theory. *International Journal of Accounting, Finance, Auditing, Management and Economics*, 3(2-1), pp.282-294.

Kanuri, S., 2016. Hedged ETFs: Do they add value?. *Financial Services Review: The Journal of Individual Financial Management*, 25(2), pp.181-198.

Karbasi, D., Nazemi, A. and Rabiei, M.R., 2021. An optimization technique for solving a class of ridge fuzzy regression problems. *Neural Processing Letters*, 53(5), pp.3307-3338.

Karimi-Mamaghan, M., Mohammadi, M., Meyer, P., Karimi-Mamaghan, A.M. and Talbi, E.G., 2022. Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art. *European Journal of Operational Research*, 296(2), pp.393-422.

Karoui, A.T., Sayari, S., Dammak, W. and Jeribi, A., 2024. Unveiling outperformance: a portfolio analysis of top AI-related stocks against it indices and robotics ETFs. *Risks*, 12(3), p.52.

Kasenbacher, G., Lee, J. and Euchukanonchai, K., 2017. Mean-variance vs. mean-absolute deviation: A performance comparison of portfolio optimization models. University of British Columbia. *Journal of Economic Surveys*, 34(5), pp.1170-1186.

Kelly, B.T. and Xiu, D., 2023. Financial machine learning. *National Bureau of Economic Research*. 31502. pp.7-19.

Kenig, O., Todros, K. and Adali, T., 2024. Robust Gaussian Mixture Modeling: A K-Divergence Based Approach. *IEEE Transactions on Signal Processing*. 17, pp.31-62.

Khadjeh, N. A., Aghabozorgi, S., Wah, T. Y. and Ngo, D. C. L., 2015. Text mining of news headlines for FOREX market prediction: A Multi-layer Dimension Reduction Algorithm with semantics and sentiment. *Expert Systems with Applications*, 42, pp.306-324.

Khairajani, D., 2023. Efficient market hypothesis: A conceptual framework. In *Emerging Trends and Innovations in Industries of the developing world*, CRC Press pp. 140-145).

Khaleel, M., Jebrel, A. and Shwehdy, D.M., 2024. Artificial Intelligence in Computer Science: <https://doi.org/10.5281/zenodo.10937515>. *International Journal of Electrical Engineering and Sustainability*, pp.01-21.

Khan, W., Malik, U., Ghazanfar, M.A., Azam, M.A., Alyoubi, K.H. and Alfakeeh, A.S., 2020. Predicting stock market trends using machine learning algorithms via public sentiment and political situation analysis. *Soft Computing*, 24(15), pp.11019-11043.

Khoa, B.T. and Huynh, T.T., 2021. Is it possible to earn abnormal return in an inefficient market? An approach based on machine learning in stock trading. *Computational Intelligence and Neuroscience*, 22(5), pp.350-359.

Khosravikia, F. and Clayton, P., 2021. Machine learning in ground motion prediction. *Computers and Geosciences*, 148, pp.104700.

Kılıç, Y., 2020. Adaptive market hypothesis: Evidence from the Turkey stock market. *Journal of Applied Economics and business research*, 10(1), pp.28-39.

Kimball, M.S., Reck, D., Zhang, F., Ohtake, F. and Tsutsui, Y., 2024. Diminishing marginal utility revisited. *National Bureau of Economic Research*, 51(4), pp.1523-1535.

Kitchenham, B. and Charters, S., 2007. Guidelines for performing systematic literature reviews in software engineering. *Journal of Applied Statistics*, 28(5), pp.573-583.

Koh, W.C., Kose, M.A., Nagle, P.S.O., Ohnsorge, F. and Sugawara, N., 2020. Debt and financial crises. *Applied Financial Economics*, 24(24), pp.1617-1630.

Kolawole, W., 2024. Path Dependence and Market Dynamics in Financial Systems. *Investment Management and Financial Innovations*, 17(4), pp.356-366.

Kolm, P.N., Tütüncü, R. and Fabozzi, F.J., 2014. 60 years of portfolio optimization: Practical challenges and current trends. *European Journal of Operational Research*, 234(2), pp.356-371.

Konno, H., 2003. Portfolio optimization of small-scale fund using mean-absolute deviation model. *International Journal of Theory and Applied Finance*, 6, pp. 403–418.

Konno, H., and Yamazaki, H., 1991. Mean-absolute deviation portfolio optimization model and its applications to Tokyo stock market. *Management Science*, 37(5), pp.519-531. <http://dx.doi.org/10.1287/mnsc.37.5.519>.

Konno, H.; Kobayashi, K., 1997. An integrated stock-bond portfolio optimization model. *Journal of Economics and Dynamic Control*. 21, pp. 1427–1444.

Koumou, G.B., 2020. Diversification and portfolio theory: a review. *Financial Markets and Portfolio Management*, 34(3), pp.267-312.

Krauss, C., Do, X.A. and Huck, N., 2017. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research*, 259(2), pp.689-702.

Kshatriya and K. Prasanna., 2018. Genetic algorithm-based portfolio optimization with higher moments in global stock markets, *Journal of Risk*, 20(4), pp. 1–26.

Kshetri, N., 2021. The role of artificial intelligence in promoting financial inclusion in developing countries. *Journal of Global Information Technology Management*, 24(1), pp.1-6.

Kuang, W., 2021. Which clean energy sectors are attractive? A portfolio diversification perspective. *Energy Economics*, 104, pp.105644.

Kumar, C.N., Keerthana, D., Kavitha, M. and Kalyani, M., 2022, June. Customer loan eligibility prediction using machine learning algorithms in banking sector. *7th*

international conference on communication and electronics systems (ICCES), pp. 1007-1012.

Kumar, G., Jain, S. and Singh, U.P., 2021. Stock market forecasting using computational intelligence: A survey. *Archives of computational methods in engineering*, 28(3), pp.1069-1101.

Kumar, V., 2018. A simplified perspective of the markowitz portfolio theory. *International Journal of Research and Analytical Reviews*, 5(3), pp.193-6.

Kumbure, M.M., Lohrmann, C., Luukka, P. and Porras, J., 2022. Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications*, 197, pp.116659.

Kumbure, M.M., Lohrmann, C., Luukka, P. and Porras, J., 2022. Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications*, 197, pp.116659.

Kunjai, D., Peerbhay, F. and Muzindutsi, P.F., 2021. The performance of South African exchange traded funds under changing market conditions. *Journal of Asset Management*, 22(5), pp.350-359.

Kurani, A., Doshi, P., Vakharia, A. and Shah, M., 2023. A comprehensive comparative study of artificial neural network (ANN) and support vector machines (SVM) on stock forecasting. *Annals of Data Science*, 10(1), pp.183-208.

Lai, T.L., Xing, H. and Chen, Z., 2011. Mean–variance portfolio optimization when means and covariances are unknown. *Journal of International Economics*, 111, pp.61-80.

Lam, W.S., Lam, W.H. and Jaaman, S.H., 2021. Portfolio optimization with a mean–absolute deviation–entropy multi-objective model. *Entropy*, 23(10), pp.1266.

Larsen, G.A., J.R. & Resnick, B.C. 2012, "An Optimization Strategy for Enhancing the Performance of Fund-of-Funds Portfolios", *Journal of Portfolio Management*, vol. 38, no. 2, pp. 147-154,12.

Lassance, N., DeMiguel, V. and Vries, F., 2022. Optimal portfolio diversification via independent component analysis. *Operations Research*, 70(1), pp.55-72.

Lee, I. and Shin, Y.J., 2020. Machine learning for enterprises: Applications, algorithm selection, and challenges. *Business Horizons*, 63(2), pp.157-170.

Lee, S.I. and Yoo, S.J., 2020. Threshold-based portfolio: the role of the threshold and its applications. *The Journal of Supercomputing*, 76(10), pp.8040-8057.

Lei Z, Mengxi Y, Chaoen X, Youheng D., 2018. Hardware Trojan detection based on optimized support vector machine algorithm. *Application of Electrical Technology*, 44(11), pp.17–20.

Lekhal, M. and El Oubani, A., 2020. Does the Adaptive Market Hypothesis explain the evolution of emerging markets efficiency? Evidence from the Moroccan financial market. *Heliyon*, 6(7), pp.57-69.

Lempinen, J., 2024. Evaluating market risk models in financial risk management: a comparative analysis of value-at-risk and conditional value-at-risk using parallel computation. *The Journal of Portfolio Management*, 39(2), pp.19-27.

Leon-Delgado, H. d., Praga-Alejo, R. J., Gonzalez-Gonzalez, D. S., and Cantú-Sifuentes, M., 2018. Multivariate statistical inference in a radial basis function neural network. *Expert Systems with Applications*, 93, pp.313-321.
<https://doi.org/https://doi.org/10.1016/j.eswa.2017.10.024>.

Li, H. and Shi, N., 2022. Application of genetic optimization algorithm in financial portfolio problem. *Computational Intelligence and Neuroscience*, 2022(1), pp.5246309.

Li, W., 2024. Analysis of Financial Market Efficiency. *International Journal of Global Economics and Management*, 2(2), pp.244-253.

Li, X., 2024. Research on the Development of Investment Portfolio Theory. *Highlights in Business, Economics and Management*, 40, pp.1044-1053.

Li, X., Liang, C. and Ma, F., 2022. Forecasting stock market volatility with a large number of predictors: New evidence from the MS-MIDAS-LASSO model. *Annals of Operations Research*, pp.1-40.

Li, Y., Yi, J., Chen, H. and Peng, D., 2021. Theory and application of artificial intelligence in financial industry. *Data Science in Finance and Economics*, 1(2), pp.96-116.

Liagkouras, K. and Metaxiotis, K., 2015. Efficient portfolio construction with the use of multi-objective evolutionary algorithms: Best practices and performance metrics. *International Journal of Information Technology and Decision Making*, 14(03), pp.535-564.

Liang Z, Chen H, Zhu J, Jiang K, Li Y (2018) Adversarial deep reinforcement learning in portfolio management. *Emerging Markets Review*, 9(1), pp.17-39. [arXiv:1808.09940:1-11](https://arxiv.org/abs/1808.09940)

Lin, W., Olmo, J. and Taamouti, A., 2023. Portfolio selection under systemic risk. *Journal of Money, Credit and Banking*. 25, pp.112-133.

Liu, G., 2020. Data quality problems troubling business and financial researchers: A literature review and synthetic analysis. *Journal of Business and Finance Librarianship*, 25(3-4), pp.315-371.

Liu, Q., Guo, Z. and Wang, J., 2012. A one-layer recurrent neural network for constrained pseudoconvex optimization and its application for dynamic portfolio optimization. *Neural Networks*, 26, pp.99-109.

Liu, R., Nageotte, F., Zanne, P., de Mathelin, M. and Dresch-Langley, B., 2021. Deep reinforcement learning for the control of robotic manipulation: a focussed mini review. *Robotics*, 10(1), pp.22.

Liu, S. and Xiao, C., 2021. Application and comparative study of optimization algorithms in financial investment portfolio problems. *Mobile Information Systems*, 2021, pp.1-10.

Lo, A.W. and Zhang, R., 2024. The Adaptive Markets Hypothesis: An Evolutionary Approach to Understanding Financial System Dynamics. *Oxford University Press*.

Lukomnik, J. and Hawley, J.P., 2021. Moving beyond modern portfolio theory: Investing those matters. *Open Economies Review*, 22(2), pp.293-315.

Lund, B.D., Wang, T., Mannuru, N.R., Nie, B., Shimray, S. and Wang, Z., 2023. ChatGPT and a new academic reality: Artificial Intelligence-written research papers and the ethics of the large language models in scholarly publishing. *Journal of the Association for Information Science and Technology*, 74(5), pp.570-581.

Ma, Y., Han, R. and Wang, W., 2020. Prediction-based portfolio optimization models using deep neural networks. *Ieee Access*, 8, pp.115393-115405.

Ma, Y., Han, R. and Wang, W., 2021. Portfolio optimization with return prediction using deep learning and machine learning. *Expert Systems with Applications*, 165, p.113973.

Mahapatra, B., 2023. Investigation of portfolio optimization with realistic constraints in Indian equity market using random immigration genetic algorithm. *Journal of Asia-Pacific Business*, 20(1), pp.225-239.

Maier, H.R., 2013. What constitutes a good literature review and why does its quality matter? *Environmental Modelling Software*., 43, pp.3-4.

Maiti, M., 2021. Efficient Frontier and Portfolio Optimization. *Applied Financial Econometrics: Theory, Method and Applications* pp. 89-111.

Manandhar, R., 2023. Factors influencing investment decisions in mutual funds in Nepal (Doctoral dissertation, Department of Management).

Mandal, P.K. and Thakur, M., 2024. Higher-order moments in portfolio selection problems: A comprehensive literature review. *Expert Systems with Applications*, 238, pp.121625.

Mandala, J., Soehaditama, J.P., Hernawan, M.A., Yulihapsari, I.U. and Sova, M., 2023. Forecasting Stock Returns: A Literature Review. *Indonesian Journal of Business Analytics*, 3(2), pp.171-182.

Mangram, M.E., 2013. A simplified perspective of the Markowitz portfolio theory. *Global journal of business research*, 7(1), pp.59-70.

Manzini, M.R., 2019. Impact of Tighter Regulations on Performance of Hedge Funds in South Africa: A case study of Selected Industry Stakeholders (Doctoral dissertation, University of the Witwatersrand, Faculty of Commerce, Law and Management).

Marín-Rodríguez, N.J., González-Ruíz, J.D. and Botero, S., 2025. Towards New Strategies for Investing: Insights on Sustainable Exchange-Traded Funds (ETFs). *World*, 6(1), pp.8.

Marín-Rodríguez, N.J., González-Ruíz, J.D. and Botero, S., 2025. Towards New Strategies for Investing: Insights on Sustainable Exchange-Traded Funds (ETFs). *World*, 6(1), pp.8.

Markowitz, H. M., 1952. Portfolio Selection, *The Journal of Finance*. 7 (1).

Markowitz, H.M. and Van Dijk, E., 2008. Risk-return analysis. *Handbook of asset and liability management*, pp. 139-197.

Marti, G., Nielsen, F., Bińkowski, M. and Donnat, P., 2021. A review of two decades of correlations, hierarchies, networks and clustering in financial markets. *Progress in information geometry: Theory and applications*, pp.245-274.

Masuda, J.S., 2024. Portfolio Optimization Using a Hybrid Machine Learning Stock Selection Model (Doctoral dissertation, Massachusetts Institute of Technology).

Mateus, C. and Hoang, B.T., 2021. Frontier markets, liberalization and informational efficiency: Evidence from Vietnam. *Asia-Pacific Financial Markets*, 28(4), pp.499-526.

Maurer, R. and Reiner, F., 2002. International asset allocation with real estate securities in a shortfall risk framework: the viewpoint of German and US investors. *Journal of Real Estate Portfolio Management*, 8(1), pp.27-43.

May, K., 2022. Forecast Based Portfolio Optimisation Using XGBoost (Doctoral dissertation, University of the Witwatersrand, Johannesburg).

Mazraeh, N.B., Daneshvar, A., Madanchi zaj, M. and Roodposhti, F.R., 2022. Stock portfolio optimization using a combined approach of multi objective grey wolf optimizer and machine learning preselection methods. *Computational Intelligence and Neuroscience*, 2022(1), p.5974842.

McClure, B., 2010. Modern portfolio theory: Why it's still hip. Investopedia. Retrieved on 12/10/11 from <http://www.investopedia.com/articles/06/MPT.asp#axzz1g3JQY7nY>.

McMillan, M., Pinto, J.E., Pirie, W.L. and Van de Venter, G., 2011. Investments: Principles of portfolio and equity analysis. *John Wiley and Sons*.

Medeiros M. C., Passos A. M., Vasconcelos, G. F., 2014. Parametric portfolio selection: Evaluating and comparing to Markowitz portfolios. *Revista Brasileira de Finanças*, 12(2), pp. 257-284.

Mehrotra, D., 2019. Basics of artificial intelligence & machine learning. *Notion Press*.

Mei, Y., Wu, K., 2022. Application of multi-objective optimization in the study of anti-breast cancer candidate drugs. *Sci Rep* 12, pp.19347 <https://doi.org/10.1038/s41598-022-23851-0>

Mengist, W., Soromessa, T. and Legese, G., 2020. Method for conducting systematic literature review and meta-analysis for environmental science research. *Methods X*, 7, pp.100777.

Mercangöz, B.A., 2021. Portfolio optimization. In Applying Particle Swarm Optimization: New Solutions and Cases for Optimized Portfolios. *Springer International Publishing*, pp. 15-27.

Metawa, M. Elhoseny, K. Hassan, and A. E. Hassanien., 2016. Loan portfolio optimization using Genetic Algorithm: a case of credit constraints, in *Proceedings of the 12th International Computer Engineering Conference*. Boundless Smart Societies, pp. 59–64.

Mihalache, M., R. Mihalache, O. and van den Ende, J., 2021. International diversification and MNE innovativeness: A contingency perspective of foreign subsidiary portfolio characteristics. *Management International Review*, 61(6), pp.769-798.

Milhomem, D.A. and Dantas, M.J.P., 2020. Analysis of new approaches used in portfolio optimization: a systematic literature review. *Production*, 30, pp.20190144.

Milhomem, D.A. and Dantas, M.J.P., 2021. Analysis of New Approaches Used in Portfolio Optimization: A Systematic Review. *Evolutionary and Memetic Computing for Project Portfolio Selection and Scheduling*, 26, pp.125.

Min, L., Dong, J., Liu, J. and Gong, X., 2021. Robust mean-risk portfolio optimization using machine learning-based trade-off parameter. *Applied Soft Computing*, 113, pp.107948.

Mirete-Ferrer, P.M., Garcia-Garcia, A., Baixauli-Soler, J.S. and Prats, M.A., 2022. A review on machine learning for asset management. *Risks*, 10(4), p.84.

Mirza, N., Rizvi, S.K.A., Saba, I., Naqvi, B. and Yarovaya, L., 2022. The resilience of Islamic equity funds during COVID-19: Evidence from risk adjusted performance, investment styles and volatility timing. *International Review of Economics & Finance*, 77, pp.276-295.

Mittal, S., Bhattacharya, S. and Mandal, S., 2022. Characteristics analysis of behavioural portfolio theory in the Markowitz portfolio theory framework. *Managerial finance*, 48(2), pp.277-288.

Mo, C.Y.J., Yang, S.Y. and Zhang, X., 2024. Does Adaptive Learning Neutralize Interbank Market Liquidity Hoarding under a Distressed Market Condition? *Theoretical Economics Letters*, 14(3), pp.1267-1293.

Mody, A. and Nedeljkovic, M., 2024. Central bank policies and financial markets: Lessons from the euro crisis. *Journal of Banking and Finance*, 158, pp.107033.

Mofokeng, T.S.V., 2018. An Alternative to Alternatives: Comparing the Risk-Adjusted Performance of ETF-Core Portfolios vs the ALSI. *The Journal of Portfolio Management*, 28(1), pp.44-55.

Mohichexra, B. and Leonidovna, M.N., 2023. An In-Depth Exploration of Investments and Their Types. *International Journal of Advance Scientific Research*, 3(12), pp.102-109.

Mohiuddin, R. A, 2021. Lstm-based deep learning model for stock prediction and predictive optimization model. *EURO Journal on Decision Processes*, 9, pp.100001.

Molina, A.M., 2021. Estimation of Value at Risk and Conditional Value at Risk (Doctoral dissertation, CHUO UNIVERSITY).

Moni. M.S., Kaiser M.A., Arefin M.S., 2022. Machine Intelligence and Emerging Technologies. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 491. Springer, Cham. https://doi.org/10.1007/978-3-031-34622-4_51

Montesinos López, O.A., Montesinos López, A. and Crossa, J., 2022. Support vector machines and support vector regression. In Multivariate Statistical Machine Learning Methods for Genomic Prediction, *Springer International Publishing*. pp. 337-378.

Morstad, H.O. and Bock, L.T., 2021. The Three Musketeers of Portfolio Allocation: Risk, Return, and Machine Learning: A data-driven approach to portfolio allocation using machine learning and Markowitz in the Norwegian equity market (Master's thesis).

Mousavi Anzaehaei, S.M. and Nikoomaram, H., 2022. A comparative study of the performance of Stock trading strategies based on LGBM and CatBoost algorithms. *International Journal of Finance and Managerial Accounting*, 7(26), pp.63-75.

Moyo, V., 2024. Sources and Cost of Long-Term Capital. In Firm, Equity Asset, and Fixed Income Securities Valuation: Practical Examples and Case Studies for Accounting and Financial Analyst Students pp. 129-218.

Munir, A.F., Sukor, M.E.A. and Shaharuddin, S.S., 2022. Adaptive market hypothesis and time-varying contrarian effect: Evidence from emerging stock markets of South Asia. *SAGE Open*, 12(1), p.21582440211068490.

Murialdo, P.; Ponta, L.; Carbone, A., 2021. Inferring multi-period optimal portfolios via detrending moving average cluster entropy. *EPL*, 133, pp.1–8.

Mustafa, K.H. and Ahmed, A.M., 2020. A Critical Review of the Market Efficiency Concept. *Shirkah: Journal of Economics and Business*, 5(2), pp.271-285.

Nagel, S., 2021. *Machine learning in asset pricing* (Vol. 1). Princeton University Press.

Nahar, J., Jahan, N., Shorna, S.A. and Joy, Z.H., 2024. Foundations, Themes, And Research Clusters in Artificial Intelligence and Machine Learning in Finance: A Bibliometric Analysis. *Academic Journal on Science, Technology, Engineering and Mathematics Education*, 4(03), pp.63-74.

Naik, M.J. and Albuquerque, A.L., 2022. Hybrid optimization search-based ensemble model for portfolio optimization and return prediction in business investment. *Progress in Artificial Intelligence*, 11(4), pp.315-331.

Nalban, V. and Smădu, A., 2021. Asymmetric effects of uncertainty shocks: Normal times and financial disruptions are different. *Journal of Macroeconomics*, 69, pp.103331.

Nam, G., 2023. MSE History, Transition and Development. *Journal of Namibian Studies: History Politics Culture*, 35, pp.554-577.

Nazareth, N. and Reddy, Y.V.R., 2023. Financial applications of machine learning: A literature review. *Expert Systems with Applications*, 219, pp.119640.

Nel, E.C., 2013. The Business Trust and Its Role as an Entity in the Financial Environment (Doctoral dissertation, Nelson Mandela Metropolitan University).

Neto, E.L.P., Freire, V. and Delgado, K.V., 2020, October. Risk Sensitive Markov Decision Process for Portfolio Management. *Mexican International Conference on Artificial Intelligence*, pp. 370-382.

Ngcamu, B.S., 2023. Application of the disaster management cycle and climate change: Studying flood disasters in South Africa. *Social Sciences & Humanities Open*, 8(1), pp.100657.

Ngo, V.M., Nguyen, H.H. and Van Nguyen, P., 2023. Does reinforcement learning outperform deep learning and traditional portfolio optimization models in frontier and developed financial markets? *Research in International Business and Finance*, 65, pp.101936.

Nguyen, N.H., Tong, K.T., Lee, S., Karamanli, A. and Vo, T.P., 2022. Prediction compressive strength of cement-based mortar containing metakaolin using

explainable Categorical Gradient Boosting model. *Engineering Structures*, 269, pp.114768.

Ning, B., Lin, F.H.T. and Jaimungal, S., 2021. Double deep q-learning for optimal execution. *Applied Mathematical Finance*, 28(4), pp.361-380.

Nobrega, W. and Sinha, A., 2008. Riding the Indian Tiger. Ashish Sinha: *Wiley*.

Novykov, V., Bilson, C., Gepp, A., Harris, G. and Vanstone, B.J., 2023, "Deep learning applications in investment portfolio management: a systematic literature review", *Journal of Accounting Literature*, Vol. ahead-of-print No. ahead-of-print. <https://doi.org/10.1108/JAL-07-2023-0119>

Nti, I. K., Adekoya, A. F., and Weyori, B. A. (2019). A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review*, 53(4), pp. 3007–3057. <https://doi.org/10.1007/s10462-019-09754-z>

Nuipian, W. and Meesad, P., 2025. *Dynamic Portfolio Management with Deep Reinforcement Learning* (Doctoral dissertation, King Mongkut's University of Technology North Bangkok).

Nundlall, T. and Van Zyl, T.L., 2023. Machine Learning for Socially Responsible Portfolio Optimisation. *Financial Services Review*, 24(3), pp.249-270. arXiv preprint arXiv:2305.12364.

Nyakurukwa, K. and Seetharam, Y., 2023. Alternatives to the efficient market hypothesis: An overview. *Journal of Capital Markets Studies*, 7(2), pp.111-124.

OECD., 2021. Artificial Intelligence, Machine Learning and Big Data in Finance: Opportunities, Challenges, and Implications for Policy Makers, <https://www.oecd.org/finance/artificial-intelligence-machine-learning-big-data-in-finance.htm>.

Ogunsakin, Y.O., 2020. The Impact of Market Efficiency in The Case Of Semi Strong Efficiency. *Pacific-Basin Finance Journal*, 27, pp.49-71.

Ogunsola, O.Y., Adebayo, Y.A., Dienagha, I.N., Ninduwezuor-Ehiobu, N. and Nwokediegwu, Z.S., 2024. The role of exchange-traded funds (ETFs) in financing

sustainable infrastructure projects: a conceptual framework for emerging markets. *Gulf Journal of Advance Business Research*, 2(6), pp.473-482.

Okoye, K. and Hosseini, S., 2024. R Programming: Statistical Data Analysis in Research. *Springer Nature*. 70(1), pp.101498.

Ombongi, P.N., 2014. Determinants of financial performance of unit trusts in Kenya. *Econometrica*, 47(2), pp.263-292.

Ongere, C., 2020. Testing weak form of market efficiency of Exchange Traded funds at NSE Market (Doctoral dissertation, University of Nairobi).

Osei-Brefo, E., 2024. Advances in machine learning algorithms for financial risk management (Doctoral dissertation, University of Reading).

Padhi, D.K., Padhy, N., Bhoi, A.K., Shafi, J. and Yesuf, S.H., 2022. An intelligent fusion model with portfolio selection and machine learning for stock market prediction. *Computational Intelligence and Neuroscience*, 7(4), pp.631-651.

Paiva, F.D., Cardoso, R.T.N., Hanaoka, G.P. and Duarte, W.M., 2019. Decision-making for financial trading: A fusion approach of machine learning and portfolio selection. *Expert Systems with Applications*, 115, pp.635-655.

Pakkala, J.E.T., 2022. Do active ETFs gain value over their passive counterparts. *Journal of Asset Management*, 10(4), pp.253-262.

Pan X, Wu F., 2018. Study on ACC optimization algorithm based on SVM in intrusion detection. *J Longyan Univ*, 36(05):18–22

Pan, S., Zheng, Z., Guo, Z. and Luo, H., 2022. An optimized XGBoost method for predicting reservoir porosity using petrophysical logs. *Journal of Petroleum Science and Engineering*, 208, pp.109520.

Pan, Y., Zhang, L., Wu, X. and Skibniewski, M.J., 2020. Multi-classifier information fusion in risk analysis. *Information Fusion*, 60, pp.121-136.

Pandey, D.K. and Kumari, V., 2021. Event study on the reaction of the developed and emerging stock markets to the 2019-nCoV outbreak. *International Review of Economics & Finance*, 71, pp.467-483.

Pandey, S., 2024, June. Financial Crises and Their Impacts on Global Economies: Lessons from the 2008 Financial Crisis. In *International Journal for Research Publication and Seminar*, 15(2), pp. 150-156.

Pandya, J.B., 2024. Deep Learning Approach for Stock Market Trend Prediction and Pattern Finding (Doctoral dissertation, Gujarat Technological University Ahmedabad).

Parameswaran, S.K., 2022. Fundamentals of financial instruments: An introduction to stocks, bonds, foreign exchange, and derivatives. *John Wiley and Sons*.

Parisi, L. and Manaog, M.L., 2023. Optimal Machine Learning-and Deep Learning-driven algorithms for predicting the future value of investments: A systematic review and meta-analysis. *Journal of Financial Economics*, 87(2), pp.388-417.

Park, J., Hwang, S., Lee, H. and Lee, I., 2022. Deep recurrent Q-network methods for Mmwave beam tracking systems. *IEEE Transactions on Vehicular Technology*, 71(12), pp.13429-13434.

Pav, S.E., 2021. The Sharpe Ratio: Statistics and Applications. *Chapman and Hall/CRC*.

Pedersen, L.H., Babu, A. and Levine, A., 2021. Enhanced portfolio optimization. *Financial Analysts Journal*, 77(2), pp.124-151.

Peng, Y. and Linetsky, V., 2022, November. Portfolio Selection: A Statistical Learning Approach. In *Proceedings of the Third ACM International Conference on AI in Finance*, pp. 257-263.

Perrin, S. and Roncalli, T., 2020. Machine learning optimization algorithms & portfolio allocation. Machine Learning for Asset Management: *New Developments and Financial Applications*, pp.261-328.

Petajisto, A., 2017. Inefficiencies in the pricing of exchange-traded funds. *Financial Analysts Journal*, 73(1), pp.24-54.

Petersen, K., Vakkalanka, S. and Kuzniarz, L., 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and software technology*, 64, pp.1-18.

Petticrew, M. and Roberts, H., 2008. Systematic reviews in the social sciences: A practical guide. *John Wiley and Sons*.

Peyper, W.H., 2014. Comparing different exchange traded funds in South Africa based on volatility and returns (Doctoral dissertation, Northwest University).

Pfaff, B., 2016. Financial risk modelling and portfolio optimization with R. *John Wiley and Sons*.

Phungula, N., 2024. Understanding the Dynamics of South Africa's July 2021 Social Unrest. *Journal of Nation-building & Policy Studies*, 8(1), pp.71-87.

Pinelis, M. and Ruppert, D., 2022. Machine learning portfolio allocation. *The Journal of Finance and Data Science*, 8, pp.35-54.

Pooe, K., 2021. A comparison of evolutionary computation and deep reinforcement learning for portfolio optimization (Doctoral dissertation, North-West University, South Africa).

Priya, G.J. and Saradha, S., 2021, February. Fraud detection and prevention using machine learning algorithms: A Review. *7th International Conference on Electrical Energy Systems (ICEES)* pp. 564-568.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V. and Gulin, A., 2018. CatBoost: unbiased boosting with categorical features. Advances in neural information processing systems, *Journal of Financial Services Research*, 25(1), pp.57-69.

Pullin, A.S. and Stewart, G.B., 2006. Guidelines for systematic review in conservation and environmental management. *Conservation biology*, 20(6), pp.1647-1656.

Qin, C., Zhang, Y., Bao, F., Zhang, C., Liu, P. and Liu, P., 2021. XGBoost optimized by adaptive particle swarm optimization for credit scoring. *Mathematical Problems in Engineering*, 2021, pp.1-18.

Radak, V., Damjanović, A., Rankovic, V. and Drenovak, M., 2024. Mean-expected shortfall portfolio optimization using a genetic algorithm. *Ekonomski Horizonti*, 26(2), pp.143-156.

Rahym, B. and Hawrami, D., 2020. Factor ETFs-Risk Exposure and Diversification Benefits. *International Advances in Economic Research*, 13(3), pp.313-333.

Rama Cont., 2022. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance*, 1(2), pp.223–236.

Ramesh, V., Glass, R.L. and Vessey, I., 2004. Research in computer science: an empirical study. *Journal of systems and software*, 70(1-2), pp.165-176.

Rämö H., 2002. Doing things right and doing the right things time and timing in projects. *International Journal of Project Management*, 20(7), pp.569–574.

Rasekhschaffe, K.C. and Jones, R.C., 2019. Machine learning for stock selection. *Financial Analysts Journal*, 75(3), pp.70-88.

Rifki, O. and Ono, H., 2012. A survey of computational approaches to portfolio optimization by genetic algorithms. *18th International Conference Computing in Economics and Finance. Society for Computational Economics*. 32(3), pp.355-368.

Rigamonti, A., 2020. Mean-variance optimization is a good choice, but for other reasons than you might think. *Risks*, 8(1), p.29.

Rochman, R. and Eid Jr, W., 2024. Performance of Funds of Funds (FOFs) in Brazil. *Available at SSRN 4740923*.

Roebbers L. M., Selvi A., Vera J. C., 2019. Using column generation to solve extensions to the Markowitz model. *The Engineering Economist*, 64(3), pp. 275-288.

Rubesam Alexandre, 2022. Machine learning portfolios with equal risk contributions: Evidence from the Brazilian market. *Emerging Markets Review*, 51, pp.100891.

Rustagi, R.P., 2021. Investment Analysis and Portfolio Management. *Sultan Chand & Sons*.

Sabzekar, M. and Hasheminejad, S.M.H., 2021. Robust regression using support vector regressions. *Chaos, Solitons and Fractals*, 144, pp.110738.

Sadath, S., VM, S. and Mary, A., 2023. Investors' awareness towards mutual fund schemes with special reference to Ernakulam Residence. *Journal of Accounting and Taxation*, 4(2), pp.19-28.

Sadik, S., Et-Tolba, M. and Nsiri, B., 2024. A Modified Adaptive Sparse-Group LASSO Regularization for Optimal Portfolio Selection. *IEEE Access*. 22(3), pp.1048-1083.

Saharan, V.A., Kulhari, H., Jadhav, H., Pooja, D., Banerjee, S. and Singh, A., 2010. Introduction to research methodology. In *Principles of Research Methodology and Ethics in Pharmaceutical Sciences*, pp. 1-46.

Sahu, B.R.B. and Kumar, P., 2024. Portfolio Rebalancing Model Utilizing Support Vector Machine for Optimal Asset Allocation. *Arabian Journal for Science and Engineering*, pp.1-27.

Saikia, P., Baruah, R.D., Singh, S.K. and Chaudhuri, P.K., 2020. Artificial Neural Networks in the domain of reservoir characterization: A review from shallow to deep models. *Computers and Geosciences*, 135, pp.104357.

Salas Najera, C., 2021. Embracing Machine Learning to Tackle Portfolio Optimisers Limitations. *CFA Institute Research Foundation Forthcoming*.

Sami, H.M., 2023. Intelligent Financial Portfolio Construction: Machine Learning Based Optimization (Doctoral dissertation, United International University).

Samuels, J.I., 2024. Enhancing Investment Management Strategies: A Comprehensive Analysis of Financial Instruments and Risk Mitigation Techniques. *Journal of Econometrics*, 115(1), pp.53-74.

Sanjay, K., 2022. Security analysis and portfolio management. PHI Learning Pvt. Ltd.

Santín, D., Delgado, F. J., and Valiño, A., 2007. The measurement of technical efficiency: a neural network approach. *Applied Economics*, 36(6), pp.627-635. doi: <https://doi.org/10.1080/0003684042000217661>.

Sarkar, P., Dwibedi, P., Deore, S.S., Gawali, T., Kulkarni, M.D. and Saha, A., 2024. Portfolio Optimization in Dynamic Markets: Reinforcement Learning for Investment. *International Journal of Intelligent Systems and Applications in Engineering*, 12(13s), pp.386-395.

Sarker, I.H., 2021. Machine learning: Algorithms, real-world applications and research directions. *SN computer science*, 2(3), pp.160.

Sauer, P.C. and Seuring, S., 2023. How to conduct systematic literature reviews in management research: a guide in 6 steps and 14 decisions. *Review of Managerial Science*, 17(5), pp.1899-1933.

Saxena, P., Saxena, V., Pandey, A., Flato, U. and Shukla, K., 2023. Multiple Aspects of Artificial Intelligence. *Book Saga Publications*.

Sayed, O.A. and Eledum, H., 2023. The short-run response of Saudi Arabia stock market to the outbreak of COVID-19 pandemic: An event-study methodology. *International Journal of Finance and Economics*, 28(3), pp.2367-2381.

Scherf, M., 2024. Exploring Financial Behavior: A multi-study thesis on adaptive markets, decision-making, comparison-based emotions, and Green Finance. *Journal of International Money and Finance*, 20(3), pp.297-325.

Seyedeh Asra Ahmadi, Ali Peivandizadeh., 2022. Sustainable Portfolio Optimization Model Using PROMETHEE Ranking: A Case Study of Palm Oil Buyer Companies, *Discrete Dynamics in Nature and Society*, 2022(11), pp. 74-81. <https://doi.org/10.1155/2022/8935213>.

Seyedhosseini, M. J. Esfahani, and M. Ghaffari., 2016. A novel hybrid algorithm based on a harmony search and artificial bee colony for solving a portfolio optimization

problem using a mean-semi variance approach, *Journal of Central South University*, 23(1), pp. 181–188.

Shafaghat, M. and Dezvareh, R., 2021. Support vector machine for classification and regression of coastal sediment transport. *Arabian Journal of Geosciences*, 14(19), pp.2009.

Shah, M., Gandhi, K., Patel, K.A., Kantawala, H., Patel, R. and Kothari, A., 2023, January. Theoretical evaluation of ensemble machine learning techniques. *5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pp. 829-837.

Shahid, S., 2024. Does Portfolio Optimization Still Matter? A Comparison of Traditional and Modern Methods. *Kashf Journal of Multidisciplinary Research*, 1(2), pp.85-93.

Shannon, C.E., 1948. A Mathematical Theory of Communication. *Bell Syst. Tech. J.*, 27, pp.379–423.

Sharifani, K. and Amini, M., 2023. Machine learning and deep learning: A review of methods and applications. *World Information Technology and Engineering Journal*, 10(07), pp.3897-3904.

Sharma, M. and Shekhawat, H.S., 2022. Portfolio optimization and return prediction by integrating modified deep belief network and recurrent neural network. *Knowledge-Based Systems*, 250, pp.109024.

Shenjere, P.A., Ferreira-Schenk, S. and Moodley, F., 2025. Does Investor Sentiment Influence South African ETF Flows During Different Market conditions? *Economies*, 13(1), pp.10.

Shi, S., Li, J., Li, G., Pan, P. and Liu, K., 2021. October. Xpm: An explainable deep reinforcement learning framework for portfolio management. *Proceedings of the 30th ACM international conference on information and knowledge management*, pp. 1661-1670.

Shivaprakash, K.N., Swami, N., Mysorekar, S., Arora, R., Gangadharan, A., Vohra, K., Jadeyegowda, M. and Kiesecker, J.M., 2022. Potential for artificial intelligence (AI) and

machine learning (ML) applications in biodiversity conservation, managing forests, and related services in India. *Sustainability*, 14(12), pp.7154.

Shixiang Z., 2019. Automobile sales forecast based on genetic algorithm optimized support vector machine. *Business Management*, 01, pp.128–131.

Shtini, S., 2023. Asset Allocation and Machine Learning: a performance analysis within distressed market conditions. *Emerging Markets Review*, 6(4), pp.324-345.

Shuai Liu, Chenglin Xiao., 2021. Application and Comparative Study of Optimization Algorithms in Financial Investment Portfolio Problems, *Mobile Information Systems*, vol. 2021, Article ID 3462715, 10 pages, <https://doi.org/10.1155/2021/3462715>

Shubharthi.D., 2016. Forecasting to Classification: Predicting the direction of stock market price using Xtreme Gradient Boosting. In: PESIT South Campus. doi: 10.13140/RG.2.2.15294.48968.

Sieranoja, S., 2020. Clustering with kNN Graph and k-Means (Doctoral dissertation, Itä-Suomen yliopisto).

Siew, L.W., Jaaman, S.H. and Hoe, L.W., 2021. July. Higher moment of portfolio optimization with Polynomial Goal Programming approach. *Journal of Physics: Conference Series* 1988(1), pp. 012001).

Silva A., Neves R. and Horta N., 2015. A hybrid approach to portfolio composition based on fundamental and technical indicators, *Expert Systems Applications*, vol. 42, pp. 2036-2048.

Silva, L.P.D., Alem, D. and Carvalho, F.L.D., 2017. Portfolio optimization using mean absolute deviation (MAD) and conditional value-at-risk (CVaR). *Production*, 27, pp.113-127.

Simar, J., 2023. Enhancing estimation of expected returns in modern portfolio theory through machine learning. *Global Finance Journal*, 25(2), pp.90-107.

Simon, D., 2013. Evolutionary optimization algorithms. *John Wiley & Sons*.

Simonis, N. and Vrins, F., 2022. Stock Selection and Portfolio Optimization: A Machine Learning Approach. *The Review of Financial Studies*, 8(4), pp.919-972.

Sindi, W., 2023, April. Developing a Digital Twin for Offshore Wells using Physics-Rooted Models. Offshore Technology Conference, *The Developing Economies*, 51(1), pp.60-78.

Singla, A. and Jangir, H., 2020, February. A comparative approach to predictive analytics with machine learning for fraud detection of real-time financial data. *International Conference on Emerging Trends in Communication, Control and Computing (ICONC3)*, pp. 1-4.

Sivanandam, S.N., Deepa, S.N., Sivanandam, S.N. and Deepa, S.N., 2008. Genetic algorithms, *Springer Berlin Heidelberg*, pp. 15-37.

Snow, D., 2020. Machine learning in asset management—Part 1: Portfolio construction—Trading strategies. *The Journal of Financial Data Science*, 2(1), pp.10-23.

Soleymani, F. and Paquet, E., 2020. Financial portfolio optimization with online deep reinforcement learning and restricted stacked autoencoder—Deep Breath. *Expert Systems with Applications*, 156, pp113456.

Strielkowski, W., Vlasov, A., Selivanov, K., Muraviev, K. and Shakhnov, V., 2023. Prospects and challenges of the machine learning and data-driven methods for the predictive analysis of power systems: A review. *Energies*, 16(10), pp.4025.

Strydom, N.T., 2021. Stock exchange legitimacy: The case of the Johannesburg Stock Exchange, 1887-1945. University of Johannesburg (South Africa).

Sturgeon, T.J., 2021. Upgrading strategies for the digital economy. *Global strategy journal*, 11(1), pp.34-57.

Sufi, A. and Taylor, A.M., 2022. Financial crises: A survey. *Handbook of international economics*, 6, pp.291-340.

Sugozu, I.H., Yasar, S. and Verberi, C., 2023. The relationship between long-term portfolio investments and growth in the context of asset characteristics and development level. *Borsa Istanbul Review*, 23(1), pp.34-43.

Sulaiman, M., Halim, Z., Lebbah, M., Waqas, M. and Tu, S., 2021. An evolutionary computing-based efficient hybrid task scheduling approach for heterogeneous computing environment. *Journal of Grid Computing*, 19, pp.1-31.

Sun, C., 2021. Machine Learning in Finance: Estimation, Inference and Financial Applications for Correlated Data. *The Review of Financial Studies*, 6(1), pp.191-212.

Surtee, T.G. and Alagidede, I.P., 2022. A novel approach to using Modern Portfolio Theory. *Borsa Istanbul Review. Journal of Futures Markets*, 37(4), pp.411-428.

Swisher, P. and Kasten, G.W., 2005. Post-modern portfolio theory. *Journal of Financial Planning*, 18(9), p.74.

Ta, V.D., Liu, C.M. and Addis, D., 2018, December. Prediction and portfolio optimization in quantitative trading using machine learning techniques. *Proceedings of the 9th International Symposium on Information and Communication Technology*, pp. 98-105.

Taskinsoy, J., 2023. Financial Instability: The Unsolved Mystery of Financial Folly. *Available at SSRN 4395171*.

Teles, G., Rodrigues, J.J., Rabelo, R.A. and Kozlov, S.A., 2021. Comparative study of support vector machines and random forests machine learning algorithms on credit operation. *Software: Practice and Experience*, 51(12), pp.2492-2500.

Thomas Fischer and Christopher Krauss, 2018. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2), pp. 654–669.

Tian, Y. and Zhang, Y., 2022. A comprehensive survey on regularization strategies in machine learning. *Information Fusion*, 80, pp.146-166.

Toochaei, M.R. and Moeini, F., 2023. Evaluating the performance of ensemble classifiers in stock returns prediction using effective features. *Expert Systems with Applications*, 213, pp.119186.

Trichilo, G., 2020. An Explainable Machine Learning Framework for Cross-Sectional Forecast-Based Fund Selection. *École Polytechnique Fédérale de Lausanne (EPFL)*. 94, pp.277-294.

Tsalikis, G., 2020. Exchange traded funds in Europe and the US: performance and tracking errors. *Journal of Financial Planning*, 18(6), pp.30-35.

Tunkiel, A.T., Sui, D. and Wiktorski, T., 2021. Automated Iterative Gap Filling Method for Drilling Logs and Other Data Series. In *International Conference on Offshore Mechanics and Arctic Engineering* Vol. 85208, pp. V010T11A002.

Udo, O.O. and Agbai, E.P., 2023. The Impact of Currency Redenomination on Economic Development. *Journal of Accounting and Financial Management*, 126.

Ustaoğlu, E., 2022. Portfolio optimization with graphical lasso and an application in Borsa Istanbul. *Öneri Dergisi*, 17(58), pp.760-765.

Uygun, F.N., 2024. Machine Learning Applications in Portfolio Optimization (Master's thesis, Middle East Technical University).

Veneeya, V., 2006. Analysis of modern portfolio theory. Retrieved on 12/10/11 from <http://www.articlesbase.com/finance-articles/analysis-of-modern-portfolio-theory-40421.html>.

Vichi, M., Cavicchia, C. and Groenen, P.J., 2022. Hierarchical means clustering. *Journal of Classification*, 39(3), pp.553-577.

Vieira, E.S. and Gomes, J.A., 2009. A comparison of Scopus and Web of Science for a typical university. *Scientometrics*, 81, pp.587-600.

Vithayathil Varghese, N. and Mahmoud, Q.H., 2020. A survey of multi-task deep reinforcement learning. *Electronics*, 9(9), p.1363.

Vo, D.H., Pham, T.N., Pham, T.T.V., Truong, L.M. and Nguyen, T.C., 2019. Risk, return and portfolio optimization for various industries in the ASEAN region. *Borsa Istanbul Review*, 19(2), pp.132-138.

Volta, V., 2021 Portfolio Optimisation during Financial Crisis: Markowitz, LASSO and RIDGE. *Econometrica*, 57(2), pp.357-384.

Vouldis, A.T., Michaelides, P.G. and Tsionas, E.G., 2010. Estimating semi-parametric output distance functions with neural-based reduced form equations using LIML. *Economic Modelling*, 27(3), pp.697-704.

Wamba-Taguimdje, S.L., Fosso Wamba, S., Kala Kamdjoug, J.R. and Tchatchouang Wanko, C.E., 2020. Influence of artificial intelligence (AI) on firm performance: the business value of AI-based transformation projects. *Business Process Management Journal*, 26(7), pp.1893-1924.

Wang, J., Cheng, Q. and Dong, Y., 2023. An XGBoost-based multivariate deep learning framework for stock index futures price forecasting. *Kybernetes*, 52(10), pp.4158-4177.

Wang, J., Tan, L., Wang, Y. and Wang, J.Y., 2024. Strategic thinking: the core of investment decision-making in Fund-of-Fund (FoF). *Journal of Chinese Economic and Business Studies*, 22(4), pp.531-555.

Wang, L., Liu, Q., Dong, S. and Soares, C.G., 2022. Selection of countermeasure portfolio for shipping safety with consideration of investment risk aversion. *Reliability Engineering & System Safety*, 219, pp.108189.

Wang, Z. and Khan, M.M., 2017. Market states and the risk-return tradeoff. *The Quarterly Review of Economics and Finance*, 65, pp.314-327.

Wekwete, T., Kufakunesu, R. and van Zyl, G., 2020. Application of Deep Reinforcement Learning in Asset Liability Management. Available at SSRN 4417872.

Westin, L., 2019. Utilizing short-term noise in non-efficient markets by paired assets: - Introducing the Technical AMH trader.

Weston, R. and Ford, G., 2003. Optimal portfolio allocations for global bank stocks in local currencies and US dollars 1992-2001. *International Business and Economics Research Journal (IBER)*, 2(11), pp. 11-19.

Wilson A. G., Kim B., and Herlands W., 2016. Proceedings of NIPS 2016 Workshop on Interpretable Machine Learning for Complex Systems. [Online]. Available: <https://arxiv.org/abs/1611.09139>

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A., Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C. and Regnell, B., 2012. Empirical Strategies. *Experimentation in Software Engineering*, pp.9-36.

Wongsawatgul, B., 2019. An Investigation of ETF Flows: Asset Allocation Perspectives. *The Journal of Finance*, 48(1), pp.187-211.

Woode, J.K., Owusu Junior, P., Adam, A.M., Assifuah-Nunoo, E. and Adjei, A.F., 2023. Nexus between cryptocurrencies and global uncertainty: A quantile regression approach. *Cogent Economics and Finance*, 11(2), pp.2282809.

Wu, D., Wang, X. and Wu, S., 2022. Construction of stock portfolios based on k-means clustering of continuous trend features. *Knowledge-Based Systems*, 252, pp.109358.

Wu, M.E., Syu, J.H., Lin, J.C.W. and Ho, J.M., 2021. Portfolio management system in equity market neutral using reinforcement learning. *Applied Intelligence*, 51(11), pp.8119-8131.

Xiaolei, S., Mingxi, L., and Zeqian, S. (2020). A novel cryptocurrency price trend forecasting model based on LightGBM. *Finance Research Letters*, 32(8), pp. 31-46. <https://doi.org/10.1016/j.frl.2018.12.032>.

Xing G., Yang S., 2020. First and Second Order Asymptotics of the Spectral Risk Measure for Portfolio Loss Under Multivariate Regular Variation. *Journal of Systems Science and Complexity*, 33(5), pp. 1533-1544.

Xu, Y., Hu, C., Wu, Q., Jian, S., Li, Z., Chen, Y., Zhang, G., Zhang, Z. and Wang, S., 2022. Research on particle swarm optimization in LSTM neural networks for rainfall-runoff simulation. *Journal of Hydrology*, 608, pp.127553.

Yang, L., Zhang, H., Shen, H., Huang, X., Zhou, X., Rong, G. and Shao, D., 2021. Quality assessment in systematic literature reviews: A software engineering perspective. *Information and Software Technology*, 130, pp.106397.

- Yilin Ma, Ruizhu Han, and Weizhong Wang., 2020. Prediction-based portfolio optimization models using deep neural networks. *IEEE Access*, 8, pp. 115393–115405.
- Yiqing L, Wushan C., 2019 Study on face detection of support vector machine based on PCA. *Comput Meas Control*, 27(03), pp. 49–54.
- Yoldas Ayan, B., 2023. A Comparison of Traditional Models and Machine Learning Techniques for Volatility Forecasting (Doctoral dissertation, Dublin Business School).
- Yu P, Lee JS, Kulyatin I, Shi Z, Dasgupta S., 2019 Model-based deep reinforcement learning for dynamic portfolio optimization. *Journal of Financial and Quantitative Analysis*, 29(3), pp.419-444. [arXiv:1901.08740:1-21](https://arxiv.org/abs/1901.08740)
- Yu, C.K., Yip, T.L. and Choy, S.K., 2019. Optimal portfolio choice for ship leasing investments. *Maritime Policy and Management*, 46(7), pp.884-900.
- Yu, Lean, Wang, Shouyang and Lai, Kin Keung., 2008. Portfolio Optimization Using Evolutionary Algorithms. *Information Technology Ecosystems*, pp. 235-245. 10.4018/978-1-59904-627-3.ch014.
- Yuan Y, Yu S, Wang C, Zhou A., 2019. A classification model of surrounding rock stability based on grid search method for support vector machine. *Geol Prospect* 55(02), pp.608–613
- Zaimovic, A., Omanovic, A. and Arnaut-Berilo, A., 2021. How many stocks are sufficient for equity portfolio diversification? A review of the literature. *Journal of Risk and Financial Management*, 14(11), pp.551.
- Zanjirdar, M., 2020. Overview of portfolio optimization models. *Advances in mathematical finance and applications*, 5(4), pp.419-435.
- Zemla, S., 2021. *Financial And Economic Crises in Comparative Perspective* (Doctoral dissertation, University of Pula. Faculty of economics and tourism).

Zenios, S.A. and Kang, P., 1993. Mean-absolute deviation portfolio optimization for mortgage-backed securities. *Annals of Operations Research*, 45(1), pp.433-450.

Zenios, S.A., 1995. Asset/liability management under uncertainty for fixed-income securities. *Ann. Oper. Res.*, 59, pp.77–97.

Zhang, H., B. Jacobsen, F. de Roon, and F. Jiang., 2019. Boosting portfolio choice in the big data era. *Economic Modelling*, 34, pp.59-68.

Zhang, M., Yang, D., Du, J., Sun, H., Li, L., Wang, L. and Wang, K., 2023. A review of SOH prediction of Li-ion batteries based on data-driven algorithms. *Energies*, 16(7), pp.3167.

Zhang, S., He, X., Ding, Y., Shi, Z. and Wu, B., 2024. Supply and demand of platinum group metals and strategies for sustainable management. *Renewable and Sustainable Energy Reviews*, 204, pp.114821.

Zhang, Z., Zohren, S. and Roberts, S., 2020. Deep learning for portfolio optimization. *arXiv preprint arXiv:2005.13665*.

Zhao, Y., Stasinakis, C., Sermpinis, G. and Shi, Y., 2018. Neural network copula portfolio optimization for exchange traded funds. *Quantitative Finance*, 18(5), pp.761-775.

Zhou, H., Dash, G.H. and Kajiji, N., 2021. Artificial Intelligence Function Mapping to Calibrate the Determinants of SMME Performance. Available at SSRN 3310145.

Zhou, J., Yang, J., Sun, H., Liu, Y. and Liu, X., 2021. The influence of entrepreneurial cognition on business model innovation: a hybrid method based on multiple regressions and machine learning. *Frontiers in Psychology*, 12, pp.744237.

Zhou, Q., Zhang, H. and Wang, S., 2021. Artificial intelligence, big data, and blockchain in food safety. *International Journal of Food Engineering*, 18(1), pp.1-14.

Zhou, R.; Cai, R.; Tong, G., 2013. Applications of entropy in finance: A review. *Entropy*, 15, pp. 4909–4931.

Zhou, Y., Zhang, H., Huang, X., Yang, S., Babar, M.A. and Tang, H., 2015, April. Quality assessment of systematic reviews in software engineering: A tertiary study. *Proceedings of the 19th international conference on evaluation and assessment in software engineering* 89(7), pp. 1-14.

Zhu, J.M., Geng, Y.G., Li, W.B., Li, X. and He, Q.Z., 2022. Fuzzy decision-making analysis of quantitative stock selection in VR industry based on random forest model. *Journal of Function Spaces*, 20(7), pp.1-12.

Ziane, M., Sara, C., Fatima, B., Abdelhakim, C. and EL Moutaouakil, K., 2024. Portfolio selection problem: main knowledge and models (A systematic review). *Statistics, Optimization and Information Computing*, 12(3), pp.799-816.

APPENDIX 1: ETFs Information

Symbol	Description	Inception Date	Delisted Date	Type of Benchmark
ETFs still trading at the end of December 2024				
STXFIN	Satrix FINI ETF	15 February 2002		Domestic
GLD	NewGold	11 January 2004		Domestic
SYGEU	Sygnia EuroStoxx 50 ETF*	10 October 2005		Foreign
SYGUK	Sygnia FTSE 100 ETF*	10 October 2005		Foreign
STXRES	Satrix RESI ETF	04 October 2006		Domestic
STXSWX	Satrix SWIX Top 40 ETF	04 October 2006		Domestic
STXDIV	Satrix DIVI Plus ETF	29 August 2007		Domestic
SYGWD	Sygnia MSCI World ETF*	04 January 2008		Foreign
STXSHA	Satrix Shariah T40 ETF	06 Apr 2009		Domestic
FNBT40	FNB TOP 40 ETF	16 October 2008		Domestic
ETFSAP	1Invest SA Property ETF*	14 February 2013		Domestic
NGPLT	NewPlatinum ETF	26 April 2013		Domestic
NGPLD	NewPalladium ETF	04 February 2014		Domestic
ETFGLD	1Invest Gold ETF*	04 July 2014		Domestic
ETFPLT	1Invest Platinum ETF*	04 July 2014		Domestic
CTOP50	CoreShares Top 50 ETF	06 March 2015		Domestic
ETFRHO	1Invest Rhodium ETF*	12 April 2015		Domestic
GLPROP	Coreshares S&P Global Property ETF	11 April 2016		Foreign
AMIB50	AMI Big 50 (ex-SA) ETF	20 April 2017		Foreign
SYGP	Sygnia Itrix Global Property ETF	11 July 2017		Foreign
SYGSW4	Sygnia Itrix SWIX 40 ETF	11 July 2017		Domestic
SYGT40	Sygnia Itrix Top 40 ETF	11 July 2017		Domestic
ETFGGB	1Invest Global Government Bond ETF*	15 March 2018		Foreign
ETFGRE	1Invest Global REIT ETF*	15 March 2018		Foreign
ETF500	1Invest S&P500 INDEX ETF*	15 March 2018		Foreign
STXVEQ	Satrix Value Equity ETF	26 Mar 2018		Domestic

STXQUA	Satrix Quality SA ETF	26 September 2018		Domestic
STXDEQ	Satrix Defensive Eq ETF	25 Feb 2019		Domestic
ETFBND	1invest SA Bond ETF*	15 June 2019		Domestic
CSPROP	CoreShares SAProp	30 October 2019		Domestic
STXCAP	Satrix Capped ALL Share	10 November 2021		Domestic
ETFEMA	1invest MSCI EM Asia ETF	18 August 2022		Foreign
ETFs delisted before the end of December 2024				
NFSWIX	NewFunds SWIX Top 40	27 January 2012	31 December 2019	Domestic

Note: * represents ETFs which had their names changed due to changes in the issuer

APPENDIX 2: R CODES

```
#::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::#
#                                                                    #
#  LOAD DATA                                                         #
#                                                                    #
#::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::#
ProjectFilePath = "H:/My Drive/Projects/zz_projects/Mutemeri_Linah/ETF/"
load(file="H:/My
Drive/Programming_templates/AnalyticsCockpitFireUpCleanNot.rda")
AnalyticsCockpitFireUpCleanNot(c())

{
# From Excel
dataset_name = c('data ETF')
#
v = c("Returns data")
data ETF = read_excel(paste0(ProjectFilePath,dataset_name,".xlsx"),sheet =
c(v))%>%as.data.frame()

# quick name clean up
colnames(data ETF) = data ETF %>% select_all(~ stringr::str_squish(.) %>%
as.character() %>%
                                multigsub(c(" ", ".", '(', ')'),c("_", "_", "'", "'"),..)
%>% gsub("[^[:alpha:]]", "",..)
                                )%>% colnames()

#
data ETF = data ETF %>% rename(
```

```

      '_Invest_SA_Property_ETF_ETFSAP' =
'Invest_SA_Property_ETF_ETFSAP'
      , '_InvestGold_ETF_ETFGLD' = 'InvestGold_ETF_ETFGLD'
      , '_InvestPlatinum_ETF_ETFPLT' =
'InvestPlatinum_ETF_ETFPLT'
      , '_10X_SAProp_Income_CSPROP' =
'10X_SAProp_Income_CSPROP'
      , '_10X_Top_50_CTOP50' = '10X_Top_50_CTOP50'
      , '_InvestRhodium_ETF_ETFRHO' =
'InvestRhodium_ETF_ETFRHO'
      , '_10X_Global_Prop_GLPROP' =
'10X_Global_Prop_GLPROP'
      , '_Invest_S_P500_Index_ETF_ETF500' =
'Invest_S&P500_Index_ETF_ETF500'
      , '_Invest_G7_Govt_Bond_ETF_ETFGGB' =
'Invest_G7_Govt_Bond_ETF_ETFGGB'
      , '_Invest_Global_REIT_ETF_ETFGRE' =
'Invest_Global_REIT_ETF_ETFGRE'
      , '_Invest_SA_Bond_ETF_ETFBND' =
'Invest_SA_Bond_ETF_ETFBND'
      , '_Invest_MSCI_EM_Asia_ETF_ETFEMA' =
'Invest_MSCI_EM_Asia_ETF_ETFEMA'
    )

#
data_explo = data_ETF
seenames(data_explo)

#
unwanted_vars = c()

#
drop_names = names(select_at(data_explo
                             ,vars(starts_with("_x")
                                   ,ends_with("_x")
                                   ,ends_with("_old")
                                   ,contains("_old")
                                   ,matches("_old")
                                   ),));drop_names

data_explo =
data_explo[names(data_explo)%nin%c(drop_names,unwanted_vars)]
seenames(data_explo)

# choose 1
AnalyticsTidyCockpitNames(data_explo)

save(data_explo, file=paste0(ProjectFilePath,"data_explo.rda"))

```

```

}
#::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::#
#
# EXPLORE THE DATA
#
#::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::#
#=====
# Loading libraries
#=====

{
#-----
-

#1. Markowitz Model (Mean-Variance Optimization)
  #if(!require(quadprog)){install.packages("quadprog")} #ok
  library(quadprog)
  #packageurl <- "https://cran.r-project.org/src/contrib/Archive/PortfolioAnalytics/PortfolioAnalytics_1.0.3636.tar.gz"; install.packages(packageurl, repos=NULL, type="source") #3.6/3.3.0

  #if(!require(PortfolioAnalytics)){install.packages("PortfolioAnalytics")}
  #ok
  library(PortfolioAnalytics)
  library(DEoptim) # Can install directly from pane
  library(ROI) # Not working
  library(pso)
  library(GenSA)

#2. Conditional Value at Risk (CVaR)
  #if(!require(RiskPortfolios)){install.packages("RiskPortfolios")} #ok
  library(RiskPortfolios)

  #if(!require(PerformanceAnalytics)){install.packages("PerformanceAnalytics")}
  #ok
  library(PortfolioAnalytics)

#3. Mean Absolute Deviation (MAD)
  #if(!require(lpSolve)){install.packages("lpSolve")} #ok
  library(lpSolve)

  #if(!require(PortfolioAnalytics)){install.packages("PortfolioAnalytics")}
  #ok
  library(PortfolioAnalytics)

```

```

#4.    K4-RANN (K-Nearest Neighbors with Randomly Assigned Neural Networks)

  #if(!require(caret)){install.packages("caret")} #ok

  library(caret)

  #packageurl <- "https://cran.r-project.org/src/contrib/Archive/RcppTOML/RcppTOML_0.1.7.tar.gz";
  install.packages(packageurl, repos=NULL, type="source") # 3.6

  #if(!require(RcppTOML)){install.packages("RcppTOML")} #ok

  library(RcppTOML)

  #packageurl <- "https://cran.r-project.org/src/contrib/Archive/reticulate/reticulate_1.20.tar.gz";
  install.packages(packageurl, repos=NULL, type="source") # 3.6

  #if(!require(reticulate)){install.packages("reticulate")} #ok

  library(reticulate)

  #packageurl <- "https://cran.r-project.org/src/contrib/Archive/kerasR/kerasR_0.6.1.tar.gz";
  install.packages(packageurl, repos=NULL, type="source") # 3.6

  #library(kerasR)

  #if(!require(kerasR)){install.packages("kerasR")} #ok

  #packageurl <- "https://cran.r-project.org/src/contrib/Archive/tensorflow/tensorflow_2.4.0.tar.gz";
  install.packages(packageurl, repos=NULL, type="source") # 3.6

  #if(!require(tensorflow)){install.packages("tensorflow")}

  library(tensorflow)

#5.    Evolutionary (Genetic) Algorithms

  #packageurl <- "https://cran.r-project.org/src/contrib/Archive/GA/GA_3.2.1.tar.gz";
  install.packages(packageurl, repos=NULL, type="source") # 3.6

  #if(!require(GA)){install.packages("GA")} #ok

  library(GA)

  #if(!require(genalg)){install.packages("genalg")} #ok

  library(genalg)

#6.    Extreme Gradient Boosting

  #dotR <- file.path(Sys.getenv("HOME"), ".R")
  #if (!file.exists(dotR))
  #  dir.create(dotR)
  #M <- file.path(dotR, "Makevars.win")
  #if (!file.exists(M))
  #  file.create(M)
  #cat("\nCXX14FLAGS=-O3 -Wno-unused-variable -Wno-unused-function",
  #  "CXX17 = $(BINPREFIX)g++ -m$(WIN) -std=c++1y",
  #  "CXX11FLAGS=-O3 -Wno-unused-variable -Wno-unused-function",
  #  file = M, sep = "\n", append = TRUE)

```

```

# The code above is required before running the below

#packageurl <- "https://cran.r-project.org/src/contrib/Archive/xgboost/xgboost_1.4.1.1.tar.gz";
install.packages(packageurl, repos=NULL, type="source") # 3.6

#if(!require(xgboost)){install.packages("xgboost")} # did not work and we used the above?

library(xgboost)

# all below are dependencies for package 'tidyverse'

#packageurl <- "https://cran.r-project.org/src/contrib/Archive/dbplyr/dbplyr_2.1.1.tar.gz";
install.packages(packageurl, repos=NULL, type="source") # 3.6

#packageurl <- "https://cran.r-project.org/src/contrib/Archive/dtplyr/dtplyr_1.1.0.tar.gz";
install.packages(packageurl, repos=NULL, type="source") # 3.6

#packageurl <- "https://cran.r-project.org/src/contrib/Archive/gargle/gargle_1.1.0.tar.gz";
install.packages(packageurl, repos=NULL, type="source") # 3.6

#packageurl <- "https://cran.r-project.org/src/contrib/Archive/googledrive/googledrive_1.0.1.tar.gz";
install.packages(packageurl, repos=NULL, type="source") # 3.6

#install.packages("ids")

#packageurl <- "https://cran.r-project.org/src/contrib/Archive/googlesheets4/googlesheets4_0.3.0.tar.gz";
install.packages(packageurl, repos=NULL, type="source") # 3.6

#packageurl <- "https://cran.r-project.org/src/contrib/Archive/reprex/reprex_2.0.0.tar.gz";
install.packages(packageurl, repos=NULL, type="source") # 3.6

#packageurl <- "https://cran.r-project.org/src/contrib/Archive/rvest/rvest_1.0.0.tar.gz";
install.packages(packageurl, repos=NULL, type="source") # 3.6

#packageurl <- "https://cran.r-project.org/src/contrib/Archive/tidyverse/tidyverse_1.3.1.tar.gz";
install.packages(packageurl, repos=NULL, type="source") # 3.6

library(tidyverse)

#packageurl <- "https://cran.r-project.org/src/contrib/Archive/TTR/TTR_0.24.2.tar.gz";
install.packages(packageurl, repos=NULL, type="source") # 3.6

#packageurl <- "https://cran.r-project.org/src/contrib/Archive/quantmod/quantmod_0.4.18.tar.gz";
install.packages(packageurl, repos=NULL, type="source") # 3.6

library(quantmod)

}

#=====
# MODEL 1: Markowitz Model (Mean-Variance Optimization)
#=====

{
  ProjectFilePath = "H:/My Drive/Projects/zz_projects/Mutemeri_Linah/ETF/"

```



```

load(file="H:/My
Drive/Programming_templates/AnalyticsCockpitFireUpCleanNot.rda")
AnalyticsCockpitFireUpCleanNot(c())
load(file=paste0(ProjectFilePath,"data_explo.rda"))

names(data_explo)

#-----
seenames(data_explo)
v = names(data_explo[c(2:34)])
seerecode(data_explo[c(v)])
returns = data_explo %>%mutate_at(vars(names(.[c(v)])),list(function(x) {
  ifelse(is.na(x), 0,x)
}));seesummary(returns[c(v)])

row.names(returns) <- returns$date
returns[1] <- NULL
returns = returns %>% as.xts()

names(returns)
portfolio <- portfolio.spec(assets = colnames(returns))
portfolio <- add.constraint(portfolio, type = "full_investment")
portfolio <- add.constraint(portfolio, type = "long_only")
portfolio <- add.objective(portfolio, type = "return", name = "mean")
portfolio <- add.objective(portfolio, type = "risk", name = "var")

# Optimize Portfolio

# DEoptim -----
optimized_portfolio_DEoptim <- optimize.portfolio(
  R = returns
  , portfolio = portfolio
  , optimize_method = c("DEoptim" # Ok
    , "random" # ok
    , "ROI" #?
    , "ROI_old" #?
    , "pso" #ok
    , "GenSA" # Ok
  ) [1]
)

```

```

Opt_weights_DEoptim = extractWeights(optimized_portfolio_DEoptim) %>%
as.data.frame() %>%

  rownames_to_column() %>% rename (
    'ETF' = 'rowname'
  , 'Weights_DEoptim' = '.'
  ) %>% t() %>% as.data.frame()

colnames(Opt_weights_DEoptim) <- lapply(Opt_weights_DEoptim[1, ],
as.character)

Opt_weights_DEoptim = Opt_weights_DEoptim[-1, ] %>% as.matrix()
#names(dataFrame) <- as.character(unlist(dataFrame[row.number,]))
#
modell_DEoptim = optimized_portfolio_DEoptim[["objective_measures"]] %>%
as.data.frame()

modell_DEoptim[['Model']] = 'Modell DEoptim'

# Analyze the performance
opt_port_performance_DEoptim <- Return.portfolio(R = returns, weights =
Opt_weights_DEoptim)

charts.PerformanceSummary(opt_port_performance_DEoptim, main = "Optimized
Portfolio Performance")

# random-----
optimized_portfolio_random <- optimize.portfolio(
  R = returns
  , portfolio = portfolio
  , optimize_method = c("DEoptim" # Ok
    , "random" # ok
    , "ROI" #?
    , "ROI_old" #?
    , "pso" #ok
    , "GenSA" # Ok
  ) [2]
)

Opt_weights_random = extractWeights(optimized_portfolio_random) %>%
as.data.frame() %>%

  rownames_to_column() %>% rename (
    'ETF' = 'rowname'
  , 'Weights_random' = '.'
  )

```

```

#
modell_random = optimized_portfolio_random[["objective_measures"]] %>%
as.data.frame()

modell_random[['Model']] = 'Modell random'
# Analyze the performance
opt_port_performance_random <- Return.portfolio(R = returns, weights =
Opt_weights_random)

charts.PerformanceSummary(opt_port_performance_random, main = "Optimized
Portfolio Performance")
# pso-----
optimized_portfolio_pso <- optimize.portfolio(
      R = returns
      , portfolio = portfolio
      , optimize_method = c("DEoptim" # Ok
        , "random" # ok
        , "ROI" #?
        , "ROI_old" #ok but giving
        , "pso" #ok
        , "GenSA" # Ok
      ) [5]
    )

Opt_weights_pso = extractWeights(optimized_portfolio_pso) %>%
as.data.frame() %>%
  rownames_to_column() %>% rename(
    'ETF' = 'rowname'
    , 'Weights_pso' = '.'
  )

#
modell_pso = optimized_portfolio_pso[["objective_measures"]] %>%
as.data.frame()

modell_pso[['Model']] = 'Modell pso'
# Analyze the performance
opt_port_performance_pso <- Return.portfolio(R = returns, weights =
Opt_weights_pso)

charts.PerformanceSummary(opt_port_performance_pso, main = "Optimized
Portfolio Performance")
# GenSA-----
optimized_portfolio_GenSA <- optimize.portfolio(
      R = returns
      , portfolio = portfolio

```

```

, optimize_method = c("DEoptim" # Ok
, "random" # ok
, "ROI" #?
, "ROI_old" #ok but giving
, "pso" #ok
, "GenSA" # Ok
)[6]

)

Opt_weights_GenSA = extractWeights(optimized_portfolio_GenSA) %>%
as.data.frame() %>%

rownames_to_column() %>% rename(
  'ETF' = 'rowname'
, 'Weights_GenSA' = '.'
)

#
modell_GenSA = optimized_portfolio_GenSA[["objective_measures"]] %>%
as.data.frame()

modell_GenSA[["Model"]] = 'Modell GenSA'
# Analyze the performance
opt_port_performance_GenSA <- Return.portfolio(R = returns, weights =
Opt_weights_GenSA)
charts.PerformanceSummary(opt_port_performance_GenSA, main = "Optimized
Portfolio Performance")
# Merge -----
data_Optimized_weights_mod1 = Reduce(function(x,y) {
  merge(x = x, y = y
, sort = FALSE
, by = "ETF" # if name common to both
, by.x = "row2", by.y = "col1" # if names are just
different but contents the same
, all = TRUE
)}, list(Opt_weight_DEoptim
, Opt_weight_random
, Opt_weight_pso
, Opt_weight_GenSA
)
)
mutate_at(vars(names(.[c('Weights_DEoptim', 'Weights_random', 'Weights_pso', '
Weights_GenSA')])), list(function(x) {
  x %>% as.character() %>% as.numeric() #>%
round(.,4)
})) %>% mutate(

```

```

                                Weight_DEoptim_select = ifelse(Weights_DEoptim ==
0, 'No', 'Yes')
                                ,Weight_DEoptim_cnt
length(which(Weight_DEoptim_select %in%c('Yes'))))

                                ,Weight_random_select = ifelse(Weights_random ==
0, 'No', 'Yes')
                                ,Weight_random_cnt
length(which(Weight_random_select %in%c('Yes'))))

                                ,Weight_pso_select = ifelse(Weights_pso ==
0, 'No', 'Yes')
                                ,Weight_pso_cnt = length(which(Weight_pso_select
%in%c('Yes'))))

                                ,Weight_GenSA_select = ifelse(Weights_GenSA ==
0, 'No', 'Yes')
                                ,Weight_GenSA_cnt
length(which(Weight_GenSA_select %in%c('Yes'))))
                                )

#Selected_ETFs_mod1 = (Optimized_weights %>% filter(Weight_DEoptim_select
== 'Yes'))[["ETF"]]; Selected_ETFs
#
data_Optimized_mean_sd_mod1
rbind(modell_DEoptim,modell_random,modell_pso,modell_GenSA)
#
if(!require(xlsx)){install.packages("xlsx")}
write.xlsx(data_Optimized_weights_mod1
file=paste0(ProjectFilePath,"data_Optimized_weights_mod1.xlsx"),row.names =
FALSE, showNA= FALSE)
write.xlsx(data_Optimized_mean_sd_mod1
file=paste0(ProjectFilePath,"data_Optimized_mean_sd_mod1.xlsx"),row.names =
FALSE, showNA= FALSE)
}
#=====
#    MODEL 2:  Conditional Value at Risk (CVaR)
#=====
{
ProjectFilePath = "H:/My Drive/Projects/zz_projects/Mutemer_Linah/ETF/"
load(file="H:/My
Drive/Programming_templates/AnalyticsCockpitFireUpCleanNot.rda")
AnalyticsCockpitFireUpCleanNot(c())
load(file=paste0(ProjectFilePath,"data_explo.rda"))

```

```

#
v = names(data_explo[c(2:34)])
seerecode(data_explo[c(v)])
returns = data_explo %>%mutate_at(vars(names(.[c(v)])),list(function(x) {
  ifelse(is.na(x), 0,x)
}));seesummary(returns[c(v)])

row.names(returns) <- returns$date
returns[1] <- NULL
returns_data = returns %>% as.xts()
# Calculate the mean for each asset
mean_values <- colMeans(returns_data); mean_values
df_mean_values = mean_values %>% as.data.frame()%>%
  rownames_to_column() %>% rename(
    'ETF' = 'rowname'
    , 'Mean_model2' = '.'
  )
# Calculate the CVaR for each asset
asset_cvar_values <- apply(returns_data, 2, function(x) CVaR(x, p=0.95,
method="historical")); asset_cvar_values
df_asset_cvar_values = asset_cvar_values %>% as.data.frame()%>%
  rownames_to_column() %>% rename(
    'ETF' = 'rowname'
    , 'CVar_model2' = '.'
  )
#
df_mean_sd = merge(df_mean_values,df_asset_cvar_values, by = c('ETF'))
# Set up the portfolio optimization problem
cov_matrix <- cov(returns_data)
df_cov_matrix = cov_matrix %>% as.data.frame()%>%
  rownames_to_column()
# Optimize the portfolio with CVaR
opt_portfolio_weights <- optimalPortfolio(cov_matrix, control = list(type =
c('erc', 'maxdiv', 'maxdec'#, 'mv', 'riskeff', 'minvol', 'invvol'
) [1]
, alpha =
0.95))
asset_names = cov_matrix %>% as.data.frame() %>% colnames()
names(opt_portfolio_weights) = asset_names
print(opt_portfolio_weights)

```

```

# Evaluate the optimized portfolio
# Calculate portfolio mean returns
portfolio_mean_returns <- mean_values %*% opt_portfolio_weights ;
portfolio_mean_returns # 0.006911918
# Calculate portfolio CVar
portfolio_sd_returns <- asset_cvar_values %*% opt_portfolio_weights;
portfolio_sd_returns # -0.9730553
#
df_mean_sd_overall = data.frame(
  mean_model2 = c(portfolio_mean_returns)
  ,sd_model2 = c(portfolio_sd_returns)
)

#
if(!require(devtools)){install.packages("devtools")}
if(!require(r2excel)){devtools::install_github("kassambara/r2excel")}
xlsx.writeMultipleData("Analytics-Consultina Results.xlsx"
  ,df_mean_sd
  ,df_mean_sd_overall
  ,df_cov_matrix
)

xlsx.openFile("Analytics-Consultina Results.xlsx")# view the file
}
#=====
# MODEL 3:
#=====
{
  ProjectFilePath = "H:/My Drive/Projects/zz_projects/Mutemeri_Linah/ETF/"
  load(file="H:/My
Drive/Programming_templates/AnalyticsCockpitFireUpCleanNot.rda")
  AnalyticsCockpitFireUpCleanNot(c())
  load(file=paste0(ProjectFilePath,"data_explo.rda"))
#
v = names(data_explo[c(2:34)])
seerecode(data_explo[c(v)])
returns = data_explo %>%mutate_at(vars(names(.[c(v)])),list(function(x){
  ifelse(is.na(x), 0,x)
}));seesummary(returns[c(v)])

row.names(returns) <- returns$date
returns[1] <- NULL

```

```

returns = returns %>% as.xts()

# Define the assets and the time period
#assets <- v
#start_date <- as.Date("2020-01-01")
#end_date <- as.Date("2021-01-01")
# Get the data
#getSymbols(assets, from = start_date, to = end_date)
# Extract adjusted closing prices
#prices <- do.call(merge, lapply(assets, function(x) Ad(get(x))))
# Calculate returns
#returns <- na.omit(ROC(prices, type = "discrete"))
# Initialize the portfolio
portfolio <- portfolio.spec(assets = colnames(returns));portfolio
# Add constraint for weights to sum to 1
portfolio <- add.constraint(portfolio, type = c(
  'weight_sum', 'leverage', 'weight', 'box', 'group'
  , 'turnover',
  'diversification', 'position_limit'
  , 'return',
  'factor_exposure'
  ,
  'leverage_exposure', "full_investment") [12]
  , min_sum = 1, max_sum = 1
);portfolio

# Add constraint for weights to sum to 1
portfolio <- add.constraint(portfolio, type = c(
  'weight_sum', 'leverage', 'weight', 'box', 'group'
  , 'turnover',
  'diversification', 'position_limit'
  , 'return',
  'factor_exposure'
  , 'leverage_exposure') [1]
  , min_sum = 1, max_sum = 1);portfolio
# Add box constraints for individual asset weights
portfolio <- add.constraint(portfolio, type = c(
  'weight_sum', 'leverage', 'weight', 'box', 'group'
  , 'turnover',
  'diversification', 'position_limit'
  , 'return',
  'factor_exposure'
  , 'leverage_exposure') [4]

```



```

, min = 0, max = 1);portfolio

# Add the MAD objective
#portfolio <- add.objective(portfolio, type = c('return','risk',
'risk_budget'
#
, 'quadratic_utility',
'weight_concentration'))[2]
#
, name = c("MAD","ES","mean","StdDev")[1]
#
)

# Add the MEAN objective
portfolio <- add.objective(portfolio, type = c('return','risk',
'risk_budget'
, 'quadratic_utility',
'weight_concentration'))[1]
, name = c("MAD","ES","mean","StdDev")[3]
);portfolio

# Add the StDev objective
portfolio <- add.objective(portfolio, type = c('return','risk',
'risk_budget'
, 'quadratic_utility',
'weight_concentration'))[1]
, name = c("MAD","ES","mean","StdDev")[4]
);portfolio

# View the portfolio
print(portfolio)

# Optimize the portfolio

# DEoptim
optimized_portfolio_DEoptim <- optimize.portfolio(
R = returns
, portfolio = portfolio
, optimize_method = c("DEoptim" # Ok
, "random" # ok
, "ROI" #?
, "ROI_old" #ok but giving NA
, "pso" #ok
, "GenSA" # Ok
)[1]
, trace = TRUE
)

```

```

# Print the results
print(optimized_portfolio_DEoptim)
# Extract the optimized weights
weights_DEoptim <- extractWeights(optimized_portfolio_DEoptim)
print(weights_DEoptim)
# Calculate the optimized portfolio returns
optimized_returns_DEoptim <- Return.portfolio(returns, weights =
weights_DEoptim)
# Plot the performance
charts.PerformanceSummary(optimized_returns_DEoptim, main = "Optimized
Portfolio Performance: DEoptim")
# Print summary statistics
summary(optimized_returns_DEoptim)
Opt_weights_DEoptim = extractWeights(optimized_portfolio_DEoptim) %>%
as.data.frame() %>%
  rownames_to_column() %>% rename (
    'ETF' = 'rowname'
    , 'Weights_DEoptim' = '.'
  )
#
model3_DEoptim = optimized_portfolio_DEoptim[["objective_measures"]] %>%
as.data.frame()
model3_DEoptim[['Mode3']] = 'Model3 DEoptim'

# random
optimized_portfolio_random <- optimize.portfolio(
  R = returns
  , portfolio = portfolio
  , optimize_method = c("DEoptim" # Ok
    , "random" # ok
    , "ROI" #?
    , "ROI_old" #ok but giving NA
    , "pso" #ok
    , "GenSA" # Ok
  ) [2]
  , trace = TRUE
)

# Print the results
print(optimized_portfolio_random)
# Extract the optimized weights

```

```

weights_random <- extractWeights(optimized_portfolio_random)
print(weights_random)
# Calculate the optimized portfolio returns
optimized_returns_random <- Return.portfolio(returns, weights =
weights_random)
# Plot the performance
charts.PerformanceSummary(optimized_returns_random, main = "Optimized
Portfolio Performance: random")
# Print summary statistics
summary(optimized_returns_random)
Opt_weights_random = extractWeights(optimized_portfolio_random) %>%
as.data.frame() %>%
  rownames_to_column() %>% rename (
    'ETF' = 'rowname'
    , 'Weights_random' = '.'
  )
#
model3_random = optimized_portfolio_random[["objective_measures"]] %>%
as.data.frame()
model3_random[["Mode3"]] = 'Model3 random'

# pso
optimized_portfolio_pso <- optimize.portfolio(
  R = returns
  , portfolio = portfolio
  , optimize_method = c("DEoptim" # Ok
    , "random" # ok
    , "ROI" #?
    , "ROI_old" #ok but giving NA
    , "pso" #ok
    , "GenSA" # Ok
  ) [5]
  , trace = TRUE
)

# Print the results
print(optimized_portfolio_pso)
# Extract the optimized weights
weights_pso <- extractWeights(optimized_portfolio_pso)
print(weights_pso)
# Calculate the optimized portfolio returns

```

```

optimized_returns_pso <- Return.portfolio(returns, weights = weights_pso)
# Plot the performance
charts.PerformanceSummary(optimized_returns_pso, main = "Optimized Portfolio
Performance: pso")
# Print summary statistics
summary(optimized_returns_pso)
Opt_weights_pso      =      extractWeights(optimized_portfolio_pso)      %>%
as.data.frame() %>%
      rownames_to_column() %>% rename(
      'ETF' = 'rowname'
      , 'Weights_pso' = '.'
      )
#
model3_pso      =      optimized_portfolio_pso[["objective_measures"]]      %>%
as.data.frame()
model3_pso[['Mode3']] = 'Model3 pso'

# GenSA
optimized_portfolio_GenSA <- optimize.portfolio(
      R = returns
      , portfolio = portfolio
      , optimize_method = c("DEoptim" # Ok
      , "random" # ok
      , "ROI" #?
      , "ROI_old" #ok but giving NA
      , "pso" #ok
      , "GenSA" # Ok
      ) [6]
      , trace = TRUE
      )

# Print the results
print(optimized_portfolio_GenSA)
# Extract the optimized weights
weights_GenSA <- extractWeights(optimized_portfolio_GenSA)
print(weights_GenSA)
# Calculate the optimized portfolio returns
optimized_returns_GenSA <-      Return.portfolio(returns,      weights      =
weights_GenSA)
# Plot the performance
charts.PerformanceSummary(optimized_returns_GenSA,      main      =      "Optimized
Portfolio Performance: GenSA")

```

```

# Print summary statistics
summary(optimized_returns_GenSA)

Opt_weights_GenSA = extractWeights(optimized_portfolio_GenSA) %>%
as.data.frame() %>%
  rownames_to_column() %>% rename(
    'ETF' = 'rowname'
    , 'Weights_GenSA' = '.'
  )

#
model3_GenSA = optimized_portfolio_GenSA[["objective_measures"]] %>%
as.data.frame()

model3_GenSA[["Mode3"]] = 'Model3 GenSA'

# Merge -----
data_Optimized_weights_mod3 = Reduce(function(x,y){
  merge(x = x, y = y
    , sort = FALSE
    , by = "ETF" # if name common to both
    #, by.x = "row2", by.y = "col1" # if names are just
different but contents the same
    , all = TRUE
  )}, list(Opt_weights_DEoptim
    , Opt_weights_random
    , Opt_weights_pso
    , Opt_weights_GenSA
  )
) %>%
mutate_at(vars(names(.[c('Weights_DEoptim', 'Weights_random', 'Weights_pso', '
Weights_GenSA')])), list(function(x){
  x %>% as.character() %>% as.numeric() #>%
round(.,4)

  ))) %>% mutate(
  Weight_DEoptim_select = ifelse(Weights_DEoptim ==
0, 'No', 'Yes')
    , Weight_DEoptim_cnt =
length(which(Weight_DEoptim_select %in%c('Yes'))))

    , Weight_random_select = ifelse(Weights_random ==
0, 'No', 'Yes')
    , Weight_random_cnt =
length(which(Weight_random_select %in%c('Yes'))))

    , Weight_pso_select = ifelse(Weights_pso ==
0, 'No', 'Yes')

```

```

,Weight_pso_cnt = length(which(Weight_pso_select
%in%c('Yes')))

,Weight_GenSA_select = ifelse(Weights_GenSA ==
0, 'No', 'Yes')

,Weight_GenSA_cnt =
length(which(Weight_GenSA_select %in%c('Yes')))
)

#Selected_ETFs_mod3 = (Optimized_weights %>% filter(Weight_DEoptim_select ==
'Yes'))[["ETF"]]; Selected_ETFs
#
data_Optimized_mean_sd_mod3 = rbind(model3_DEoptim
, model3_random
, model3_pso
, model3_GenSA
)
#
if(!require(xlsx)){install.packages("xlsx")}
write.xlsx(data_Optimized_weights_mod3
file=paste0(ProjectFilePath,"data_Optimized_weights_mod3.xlsx"),row.names =
FALSE, showNA= FALSE)
write.xlsx(data_Optimized_mean_sd_mod3
file=paste0(ProjectFilePath,"data_Optimized_mean_sd_mod3.xlsx"),row.names =
FALSE, showNA= FALSE)
}
#=====
# MODEL 4:
#=====
{
ProjectFilePath = "H:/My Drive/Projects/zz_projects/Mutemeri_Linah/ETF/"
load(file="H:/My
Drive/Programming_templates/AnalyticsCockpitFireUpCleanNot.rda")
AnalyticsCockpitFireUpCleanNot(c())
load(file=paste0(ProjectFilePath,"data_explo.rda"))
#
v = names(data_explo[c(2:34)])
seerecode(data_explo[c(v)])
returns = data_explo %>%mutate_at(vars(names(.[c(v)])),list(function(x) {
ifelse(is.na(x), 0,x)
}));seesummary(returns[c(v)])

```

```

row.names(returns) <- returns$date
returns[1] <- NULL
returns = returns %>% as.xts()

# Load libraries
library(stats)
library(PortfolioAnalytics)

# Example: Perform K-means clustering
set.seed(123) # for reproducibility
k <- 4 # number of clusters
km <- kmeans(returns, centers = k)

# Example: Use RANN to select assets
library(RANN)
rann_index <- nn2(returns, query = km$centers)$nn.idx
selected_assets <- returns[rann_index, ]

# Example: Portfolio optimization using PortfolioAnalytics
library(PortfolioAnalytics)

portfolio <- portfolio.spec(assets = colnames(selected_assets))
portfolio <- add.constraint(portfolio, type = "full_investment")
portfolio <- add.constraint(portfolio, type = "long_only")
portfolio <- add.objective(portfolio, type = "return", name = "mean")
portfolio <- add.objective(portfolio, type = "risk", name = "var")

#portfolio <- optimize.portfolio(selected_assets, portfolio)

# Optimize Portfolio

# DEoptim -----
optimized_portfolio_DEoptim <- optimize.portfolio(
  R = returns
  , portfolio = portfolio
  , optimize_method = c("DEoptim" # Ok
    , "random" # ok
    , "ROI" #?
    , "ROI_old" #?
    , "pso" #ok
  )

```

```

        , "GenSA" # Ok
      ) [1]
    )

Opt_weights_DEoptim = extractWeights(optimized_portfolio_DEoptim) %>%
as.data.frame() %>%
  rownames_to_column() %>% rename (
    'ETF' = 'rowname'
    , 'Weights_DEoptim' = '.'
  )

#
model4_DEoptim = optimized_portfolio_DEoptim[["objective_measures"]] %>%
as.data.frame()
model4_DEoptim[["Model"]] = 'model4 DEoptim'

# Analyze the performance
#opt_port_performance_DEoptim <- Return.portfolio(R = returns, weights =
Opt_weights_DEoptim)
#charts.PerformanceSummary(opt_port_performance_DEoptim, main = "Optimized
Portfolio Performance")

# random-----
optimized_portfolio_random <- optimize.portfolio(
  R = returns
  , portfolio = portfolio
  , optimize_method = c("DEoptim" # Ok
    , "random" # ok
    , "ROI" #?
    , "ROI_old" #?
    , "pso" #ok
    , "GenSA" # Ok
  ) [2]
)

Opt_weights_random = extractWeights(optimized_portfolio_random) %>%
as.data.frame() %>%
  rownames_to_column() %>% rename (
    'ETF' = 'rowname'
    , 'Weights_random' = '.'
  )

#

```



```

model4_random = optimized_portfolio_random[["objective_measures"]] %>%
as.data.frame()

model4_random[['Model']] = 'model4_random'

# Analyze the performance

#opt_port_performance_random <- Return.portfolio(R = returns, weights =
Opt_weights_random)

#charts.PerformanceSummary(opt_port_performance_random, main = "Optimized
Portfolio Performance")

# pso-----
optimized_portfolio_pso <- optimize.portfolio(
    R = returns
    , portfolio = portfolio
    , optimize_method = c("DEoptim" # Ok
        , "random" # ok
        , "ROI" #?
        , "ROI_old" #ok but giving
        , "pso" #ok
        , "GenSA" # Ok
    ) [5]
)

Opt_weights_pso = extractWeights(optimized_portfolio_pso) %>%
as.data.frame() %>%
    rownames_to_column() %>% rename(
        'ETF' = 'rowname'
        , 'Weights_pso' = '.'
    )

#

model4_pso = optimized_portfolio_pso[["objective_measures"]] %>%
as.data.frame()

model4_pso[['Model']] = 'model4_pso'

# Analyze the performance

#opt_port_performance_pso <- Return.portfolio(R = returns, weights =
Opt_weights_pso)

#charts.PerformanceSummary(opt_port_performance_pso, main = "Optimized
Portfolio Performance")

# GenSA-----
optimized_portfolio_GenSA <- optimize.portfolio(
    R = returns
    , portfolio = portfolio
    , optimize_method = c("DEoptim" # Ok

```

```

        , "random" # ok
        , "ROI"     #?
        , "ROI_old" #ok but giving
        , "pso"    #ok
        , "GenSA"  # Ok
      ) [6]
    )

Opt_weights_GenSA = extractWeights(optimized_portfolio_GenSA) %>%
as.data.frame() %>%
  rownames_to_column() %>% rename(
    'ETF' = 'rowname'
    , 'Weights_GenSA' = '.'
  )
#
model4_GenSA = optimized_portfolio_GenSA[["objective_measures"]] %>%
as.data.frame()
model4_GenSA[['Model']] = 'model4 GenSA'
# Analyze the performance
#opt_port_performance_GenSA <- Return.portfolio(R = returns, weights =
Opt_weights_GenSA)
#charts.PerformanceSummary(opt_port_performance_GenSA, main = "Optimized
Portfolio Performance")
# Merge -----
data_Optimized_weights_mod4 = Reduce(function(x,y) {
  merge(x = x, y = y
        , sort = FALSE
        , by = "ETF" # if name common to both
        #, by.x ="row2", by.y ="col1" # if names are just
different but contents the same
        , all = TRUE
      )}, list(Opt_weights_DEoptim
              ,Opt_weights_random
              ,Opt_weights_pso
              ,Opt_weights_GenSA
            )
          )
mutate_at(vars(names(.[c('Weights_DEoptim', 'Weights_random', 'Weights_pso', '
Weights_GenSA')])), list(function(x) {
  x %>% as.character() %>% as.numeric() #>%
round(., 4)
})) %>% mutate(

```

```

Weight_DEoptim_select = ifelse(Weights_DEoptim ==
0, 'No', 'Yes')
,Weight_DEoptim_cnt
length(which(Weight_DEoptim_select %in%c('Yes'))))

,Weight_random_select = ifelse(Weights_random ==
0, 'No', 'Yes')
,Weight_random_cnt
length(which(Weight_random_select %in%c('Yes'))))

,Weight_pso_select = ifelse(Weights_pso ==
0, 'No', 'Yes')
,Weight_pso_cnt = length(which(Weight_pso_select
%in%c('Yes'))))

,Weight_GenSA_select = ifelse(Weights_GenSA ==
0, 'No', 'Yes')
,Weight_GenSA_cnt
length(which(Weight_GenSA_select %in%c('Yes'))))
)

#Selected_ETFs_mod4 = (Optimized_weights %>% filter(Weight_DEoptim_select
== 'Yes'))[["ETF"]]; Selected_ETFs
#
data_Optimized_mean_sd_mod4
rbind(model4_DEoptim,model4_random,model4_pso,model4_GenSA)
#
#
if(!require(devtools)){install.packages("devtools")}
if(!require(r2excel)){devtools::install_github("kassambara/r2excel")}
xlsx.writeMultipleData("Analytics-Consultina Results.xlsx"
, data_Optimized_weights_mod4
, data_Optimized_mean_sd_mod4
)
xlsx.openFile("Analytics-Consultina Results.xlsx")# view the file

}
#=====
# MODEL 5:
#=====
{
ProjectFilePath = "H:/My Drive/Projects/zz_projects/Mutemeri_Linah/ETF/"

```

```

load(file="H:/My
Drive/Programming_templates/AnalyticsCockpitFireUpCleanNot.rda")
AnalyticsCockpitFireUpCleanNot(c())
load(file=paste0(ProjectFilePath,"data_explo.rda"))
#
v = names(data_explo[c(2:34)])
seerecode(data_explo[c(v)])
returns = data_explo %>%mutate_at(vars(names(.[c(v)])),list(function(x) {
    ifelse(is.na(x), 0,x)
}));seesummary(returns[c(v)])

row.names(returns) <- returns$date
returns[1] <- NULL
returns = returns %>% as.xts()

#-----
#   OPTION 1
#-----

#expected_returns = returns
#cov_matrix = cov(returns)
# Number of assets
#n_assets <- ncol(expected_returns)

# Fitness function
fitness_function <- function(weights, returns) {
  portfolio_return <- colSums(returns * weights)
  portfolio_sd <- sqrt(sum((returns %*% weights)^2))
  sharpe_ratio <- mean(portfolio_return) / portfolio_sd
  return(sharpe_ratio)
}

# We aim to maximize return and minimize risk, so we combine these
objectives
#return(portfolio_return / portfolio_risk)
#Use the ga function to optimize the portfolio
ga_optimization1 = ga(type = c("binary", "real-valued", "permutation")[2]
  ,fitness = function(w) fitness_function(w, returns)
  ,lower = rep(0, ncol(returns))
  ,upper = rep(1, ncol(returns))

```

```

#,nBits
#,population = gaControl(type)$population
#,selection = gaControl(type)$selection
#,crossover = gaControl(type)$crossover
#,mutation = gaControl(type)$mutation
,popSize = 50
,pcrossover = 0.8
,pmutation = 0.1
,elitism = base::max(1, round(50*0.05))
,updatePop = FALSE
,postFitness = NULL
,maxiter = 1000
,run = 100
,maxFitness = Inf
, names = NULL
,suggestions = NULL
,optim = FALSE
,optimArgs = list(method = c("Nelder-Mead", "BFGS", "CG", "L-BFGS-B",
"SA", "Brent")) [4]
, poptim = 0.05
, pressel = 0.5
, control = list(fnscale = -1, maxit = 100))
, keepBest = FALSE
#,parallel = FALSE
,monitor = if(interactive()) gaMonitor else FALSE
,seed = 710
)

# Extract optimal weights
optimal_weights1_a <- ga_optimization1@solution; optimal_weights1_a
colnames(optimal_weights1_a) = names(returns); optimal_weights1_a
sum(optimal_weights1_a)
#
optimal_weights1 = optimal_weights1_a %>% as.data.frame() %>% t() %>%
as.data.frame() %>%
  rownames_to_column() %>% rename(
    'ETF' = 'rowname'
    , 'Weights' = 'V1'
  ); optimal_weights1

## Calculate the optimized portfolio return and standard deviation

```

```

optimized_return1_a <- colSums(as.matrix(returns)
as.vector(optimal_weights1_a)); optimized_return1_a
optimized_mean1 = mean(optimized_return1_a) ; optimized_mean1
optimized_return1 = optimized_return1_a %>% as.data.frame() %>%
  rownames_to_column() %>% rename(
    'ETF' = 'rowname'
    , 'Returns' = '.'
  ); optimized_return1

optimized_sd1 <- sqrt(sum((as.matrix(returns)
as.vector(optimal_weights1_a))^2)); optimized_sd1
# Calculate the Sharpe Ratio
optimized_sharpe1 <- mean(optimized_return1_a) / optimized_sd1;
optimized_sharpe1

#
model5_option1_opt_weights = optimal_weights1
model5_option1_opt_mean_sd_sharpe = data.frame(
  mean = c(optimized_mean1)
  , sd = c(optimized_sd1)
  , sharpe_ratio = c(optimized_sharpe1)
)
model5_option1_optimized_return = optimized_return1

#-----
#   OPTION 2
#-----

#
expected_returns = returns
cov_matrix = cov(returns)
# Number of assets
n_assets <- length(expected_returns)
n_assets <- ncol(expected_returns)

# Fitness function
fitness <- function(weights) {
  weights = weights / sum(weights)
  portfolio_return = sum(expected_returns * weights)
  portfolio_risk = sqrt(t(weights) %*% cov_matrix %*% weights)
}

```

```

#Use the ga function to optimize the portfolio
ga_optimization2 = ga(type = c("binary", "real-valued", "permutation")[2]
  ,fitness = fitness
  ,lower = rep(0, n_assets)
  ,upper = rep(1, n_assets)
  #,nBits
  #,population = gaControl(type)$population
  #,selection = gaControl(type)$selection
  #,crossover = gaControl(type)$crossover
  #,mutation = gaControl(type)$mutation
  #,popSize = 50
  ,pcrossover = 0.8
  ,pmutation = 0.1
  ,elitism = base::max(1, round(50*0.05))
  ,updatePop = FALSE
  ,postFitness = NULL
  ,maxiter = 1000
  ,run = 100
  ,maxFitness = Inf
  ,names = NULL
  ,suggestions = NULL
  ,optim = FALSE
  ,optimArgs = list(method = c("Nelder-Mead", "BFGS", "CG", "L-BFGS-B",
    "SANN", "Brent")[4]
    ,poptim = 0.05
    ,pressel = 0.5
    ,control = list(fnscale = -1, maxit = 100))
  ,keepBest = FALSE
  #,parallel = FALSE
  ,monitor = if(interactive()) gaMonitor else FALSE
  ,seed = 710
)

#-----
# Extract optimal weights
optimal_weights2_a <- ga_optimization2@solution; optimal_weights2_a
colnames(optimal_weights2_a) = names(returns); optimal_weights2_a
sum(optimal_weights2_a)

```

```

#
optimal_weights2 = optimal_weights2_a %>% as.data.frame() %>% t() %>%
as.data.frame() %>%

  rownames_to_column() %>% rename(
    'ETF' = 'rowname'
    , 'Weights' = 'V1'
  ); optimal_weights2

## Calculate the optimized portfolio return and standard deviation
optimized_return2_a <- colSums(as.matrix(returns) *
as.vector(optimal_weights2_a)); optimized_return2_a
optimized_mean2 = mean(optimized_return2_a) ; optimized_mean2
optimized_return2 = optimized_return2_a %>% as.data.frame() %>%
  rownames_to_column() %>% rename(
    'ETF' = 'rowname'
    , 'Returns' = '.'
  ); optimized_return2

optimized_sd2 <- sqrt(sum((as.matrix(returns)
as.vector(optimal_weights2_a))^2)); optimized_sd2 %*%
# Calculate the Sharpe Ratio
optimized_sharpe2 <- mean(optimized_return2_a) / optimized_sd2;
optimized_sharpe2

#
model5_option2_opt_weights = optimal_weights2
model5_option2_opt_mean_sd_sharpe = data.frame(
  mean = c(optimized_mean2)
  , sd = c(optimized_sd2)
  , sharpe_ratio = c(optimized_sharpe2)
)
model5_option2_optimized_return = optimized_return2

# -----
if(!require(devtools)){install.packages("devtools")}
if(!require(r2excel)){devtools::install_github("kassambara/r2excel")}
xlsx.writeMultipleData("Analytics-Consultina Results.xlsx"
  , model5_option1_opt_mean_sd_sharpe
  , model5_option2_opt_mean_sd_sharpe
  , model5_option1_opt_weights

```



```

, model5_option2_opt_weights

, model5_option1_optimized_return
, model5_option2_optimized_return

)

xlsx.openFile("Analytics-Consultina Results.xlsx") # view the file
}
#=====
#    MODEL 6:
#=====
{
  ProjectFilePath = "H:/My Drive/Projects/zz_projects/Mutemeri_Linah/ETF/"
  load(file="H:/My
Drive/Programming_templates/AnalyticsCockpitFireUpCleanNot.rda")
  AnalyticsCockpitFireUpCleanNot(c())
  load(file=paste0(ProjectFilePath, "data_explo.rda"))
#
  v = names(data_explo[c(2:34)])
  seerecode(data_explo[c(v)])
  returns = data_explo %>% mutate_at(vars(names(.[c(v)])), list(function(x) {
    ifelse(is.na(x), 0, x)
  })); seesummary(returns[c(v)])

  row.names(returns) <- returns$date
  returns[1] <- NULL
  returns = returns %>% as.xts()

  returns <- na.omit(returns)

  returns_df <- as.data.frame(returns)
  names(returns_df)
  returns_df <- returns_df %>%
    mutate(across(everything(), ~lag(.x, 1), .names = "lag_{col}")) %>%
    drop_na()

# Split into training and testing sets
set.seed(123)
train_index <- sample(seq_len(nrow(returns_df)), size = 0.8 *
nrow(returns_df))
train_data <- returns_df[train_index, ] # 80%
test_data <- returns_df[-train_index, ] # 20%

```

```

# Prepare matrices for XGBoost
train_matrix <- as.matrix(train_data[, -1])
test_matrix <- as.matrix(test_data[, -1])
train_labels <- as.matrix(train_data[, 1])
test_labels <- as.matrix(test_data[, 1])

#4. Train the XGBoost Model
#Train an XGBoost model to predict future returns.
#r
#Copy code
dtrain <- xgb.DMatrix(data = train_matrix, label = train_labels)
dtest <- xgb.DMatrix(data = test_matrix, label = test_labels)

params <- list(
  objective = "reg:squarederror",
  eval_metric = "rmse"
)

xgb_model <- xgb.train(
  params = params,
  data = dtrain,
  nrounds = 100,
  watchlist = list(train = dtrain, eval = dtest),
  early_stopping_rounds = 10,
  verbose = 1
)

#5. Predict Returns and Optimize Portfolio
#Use the trained model to predict returns and then optimize the portfolio.
#r
#Copy code
# Predict returns
predicted_returns <- predict(xgb_model, dtest)

# Calculate expected returns and covariance matrix
expected_returns <- colMeans(returns_df[, -1])
cov_matrix <- cov(returns_df[, -1])

# Define the optimization function

```

```

optimize_portfolio <- function(weights, returns, cov_matrix, risk_free_rate
= 0.01) {
  portfolio_return <- sum(weights * returns)
  portfolio_risk <- sqrt(t(weights) %*% cov_matrix %*% weights)
  sharpe_ratio <- (portfolio_return - risk_free_rate) / portfolio_risk
  return(-sharpe_ratio) # Minimize the negative Sharpe ratio
}

# Optimize the portfolio
library(DEoptim)

result <- DEoptim(optimize_portfolio,
  lower = rep(0, ncol(returns_df) - 1),
  upper = rep(1, ncol(returns_df) - 1),
  returns = expected_returns,
  cov_matrix = cov_matrix,
  DEoptim.control(itermax = 100))

optimal_weights <- result$optim$bestmem
optimal_weights <- optimal_weights / sum(optimal_weights) # Normalize
weights to sum to 1
optimal_weights

optimal_weights_df = optimal_weights %>% as.data.frame() %>%
rownames_to_column(var = 'Param') %>%
  rename(
    'Weights' = '.'
  )

weights_names = data.frame(
  Assets = c(names(returns_df))
)

#6. Analyze the Optimized Portfolio
#Analyze the optimized portfolio by calculating its performance metrics.
#r
#Copy code
# Calculate portfolio return and risk
portfolio_return <- sum(optimal_weights * expected_returns)
portfolio_risk <- sqrt(t(optimal_weights) %*% cov_matrix %*%
optimal_weights)
sharpe_ratio <- (portfolio_return - 0.01) / portfolio_risk

```

```

list(
  Return = portfolio_return,
  Risk = portfolio_risk,
  Sharpe_Ratio = sharpe_ratio
)

model6_option_opt_mean_sd_sharpe = data.frame(
  mean = c(portfolio_return)
  ,sd = c(portfolio_risk)
  ,sharpe_ratio = c(sharpe_ratio)
)

# -----
if(!require(devtools)){install.packages("devtools")}
if(!require(r2excel)){devtools::install_github("kassambara/r2excel")}
xlsx.writeMultipleData("Analytics-Consultina Results.xlsx"
  ,model6_option_opt_mean_sd_sharpe
  ,optimal_weights_df
  ,weights_names
)

xlsx.openFile("Analytics-Consultina Results.xlsx")# view the file
}

#=====
#   MODEL 7:  Option 1
#=====
{
  ProjectFilePath = "H:/My Drive/Projects/zz_projects/Mutemeri_Linah/ETF/"
  load(file="H:/My
Drive/Programming_templates/AnalyticsCockpitFireUpCleanNot.rda")
  AnalyticsCockpitFireUpCleanNot(c())
  load(file=paste0(ProjectFilePath,"data_explo.rda"))
#
  v = names(data_explo[c(2:34)])
  seerecode(data_explo[c(v)])
  returns = data_explo %>%mutate_at(vars(names(.[c(v)])),list(function(x){
    ifelse(is.na(x), 0,x)
  })))%>%      select_if(names(.) %nin%
c('nvestMSCIEMAsiaETFETFEMA')));seesummary(returns[c(v)])

row.names(returns) <- returns$date

```

```

returns[1] <- NULL
returns = returns %>% as.xts()

returns <- na.omit(returns)

#-----
#Step-by-Step Implementation
#Step 1: Data Preparation

# Load necessary libraries

# Define ETFs and get data

# Define periods
normal_period <- returns["2017-01/2020-2"]
distressed_period <- returns["2020-3/2022-07"]

#Step 2: Portfolio Construction
#K4-RANN Portfolio Optimization
#Implementing K4-RANN will depend on your specific methodology. Assuming you
have a function k4_rann_optimize that returns the optimized weights:
names(returns)
# Placeholder function for K4-RANN optimization
k4_rann_optimize <- function(returns) {
  #returns = normal_period
  set.seed(123)
  n <- ncol(returns)
  hidden_layers <- 4 # Example value for K4-RANN

  # Create initial weights
  initial_weights <- runif(n, min = 0, max = 1)
  initial_weights <- initial_weights / sum(initial_weights)

  # Neural network model
  names(returns[, -1])
  expl = paste(names(returns[, -1]), collapse = ' + ')
  dependent = 'NewGoldPalladiumETFNGPLD'
  myformula = as.formula(paste(dependent, "~", expl, collapse="+"))
  nn_model <- neuralnet(myformula, data = as.data.frame(returns), hidden =
hidden_layers, linear.output = TRUE)

```

```

# Extract weights (simplified approach)
optimized_weights <- nn_model$result.matrix[1:n]
optimized_weights <- optimized_weights / sum(optimized_weights) #
Normalize weights

return(optimized_weights)
}

# Get optimized weights for both periods
weights_k4rann_normal <- k4_rann_optimize(normal_period)
weights_k4rann_distressed <- k4_rann_optimize(distressed_period)
sum(weights_k4rann_normal)

names(weights_k4rann_normal) = names(normal_period)
df_weights_k4rann_normal = weights_k4rann_normal %>% as.data.frame() %>%
rownames_to_column() %>%
  rename(
    'Assets' = 'rowname'
    , 'Weights_k4rann_normal' = '.'
  )

names(weights_k4rann_distressed) = names(distressed_period)
df_weights_k4rann_distressed = weights_k4rann_distressed %>% as.data.frame()
%>% rownames_to_column() %>%
  rename(
    'Assets' = 'rowname'
    , 'Weights_k4rann_distressed' = '.'
  )

#Genetic Algorithm Portfolio Optimization

library(GA)

# Objective function to minimize (negative Sharpe ratio)
objective_function <- function(weights, returns) {
  portfolio_return <- sum(colMeans(returns) * weights)
  portfolio_stddev <- sqrt(t(weights) %*% cov(returns) %*% weights)
  sharpe_ratio <- portfolio_return / portfolio_stddev
  return(-sharpe_ratio) # Minimize negative Sharpe ratio

```

```

}

# Run GA for normal period
ga_model_normal <- ga(type = "real-valued",
  fitness      =      function(w)      objective_function(w,
normal_period),
  lower = rep(0, ncol(normal_period)),
  upper = rep(1, ncol(normal_period)),
  maxiter = 500)

# Run GA for distressed period
ga_model_distressed <- ga(type = "real-valued",
  fitness      =      function(w)      objective_function(w,
distressed_period),
  lower = rep(0, ncol(distressed_period)),
  upper = rep(1, ncol(distressed_period)),
  maxiter = 500)

# Get optimized weights
weights_ga_normal <- ga_model_normal@solution
weights_ga_distressed <- ga_model_distressed@solution
sum(weights_ga_normal)

weights_ga_normal_unnamed = as.numeric(weights_ga_normal)
names(weights_ga_normal_unnamed) = names(normal_period)
df_weights_ga_normal = weights_ga_normal_unnamed %>% as.data.frame() %>%
rownames_to_column() %>%
  rename(
    'Assets' = 'rowname'
    , 'Weights_ga_normal' = '.'
  )

weights_ga_distressed_unnamed = as.numeric(weights_ga_distressed)
names(weights_ga_distressed_unnamed) = names(distressed_period)
df_weights_ga_distressed = weights_ga_distressed_unnamed %>% as.data.frame()
%>% rownames_to_column() %>%
  rename(
    'Assets' = 'rowname'
    , 'Weights_ga_distressed' = '.'
  )

#Extreme Gradient Boosting (XGBoost) Portfolio Optimization

```

#This step will also depend on your specific methodology for using XGBoost for portfolio optimization. Assuming you have a function `xgboost_optimize` that returns the optimized weights:

```
# Placeholder function for XGBoost optimization
xgboost_optimize <- function(returns) {
  #returns = normal_period
  set.seed(123)
  n <- ncol(returns)

  # Prepare data for xgboost
  dtrain <- xgb.DMatrix(data = as.matrix(returns[, -1]), label = returns[, 1])
  # XGBoost parameters
  params <- list(objective = "reg:squarederror", max_depth = 2, eta = 0.1,
    nthread = 2)

  # Train the model
  xgb_model <- xgb.train(params, dtrain, nrounds = 100)

  # Extract feature importance (weights)
  importance <- xgb.importance(model = xgb_model)
  optimized_weights <- importance$Gain / sum(importance$Gain) # Normalize
weights

  names(optimized_weights) = importance$Feature
  return(optimized_weights)
  # Return a vector of optimized weights
}

# Get optimized weights for both periods
weights_xgboost_normal <- xgboost_optimize(normal_period)
weights_xgboost_distressed <- xgboost_optimize(distressed_period)
sum(weights_xgboost_normal)

df_weights_xgboost_normal = weights_xgboost_normal %>% as.data.frame() %>%
rownames_to_column() %>%
  rename(
    'Assets' = 'rowname'
    , 'Weights_xgboost_normal' = '.'
  )
```



```

df_weights_xgboost_distressed      =      weights_xgboost_distressed      %>%
as.data.frame() %>% rownames_to_column() %>%

      rename(

        'Assets' = 'rowname'

        , 'Weights_xgboost_distressed' = '.'

      )

#Step 3: Performance Evaluation

# Function to calculate portfolio returns
portfolio_returns <- function(weights, returns) {
  return(returns %*% weights)
}

# Function to evaluate performance
evaluate_performance <- function(weights, returns) {
  port_returns <- portfolio_returns(weights, returns)
  avg_return <- mean(port_returns)
  stddev <- sd(port_returns)
  sharpe_ratio <- avg_return / stddev

  return(list(return = avg_return,   stddev = stddev,   sharpe_ratio =
sharpe_ratio))
}

# Evaluate K4-RANN performance
performance_k4rann_normal  <-  evaluate_performance(weights_k4rann_normal,
normal_period)

performance_k4rann_distressed                                     <-
evaluate_performance(weights_k4rann_distressed, distressed_period)

# Evaluate GA performance
weights_ga_normal_num = unname(weights_ga_normal) %>% as.numeric()
weights_ga_distressed_num = unname(weights_ga_distressed) %>% as.numeric()

performance_ga_normal      <-      evaluate_performance(weights_ga_normal_num,
normal_period)

performance_ga_distressed                                     <-
evaluate_performance(weights_ga_distressed_num, distressed_period)

# Evaluate XGBoost performance
extracted_normal = names(weights_xgboost_normal)
extracted_distressed = names(weights_xgboost_distressed)

```

```

performance_xgboost_normal <- evaluate_performance(weights_xgboost_normal,
normal_period[,extracted_normal])

performance_xgboost_distressed <-
evaluate_performance(weights_xgboost_distressed,
distressed_period[,extracted_distressed])

```

#Step 4: Detailed Comparison

```
# Compile results into a data frame for comparison
```

```

comparison <- data.frame(
  Model = rep(c("K4-RANN", "Genetic Algorithm", "XGBoost"), each = 2),
  Period = rep(c("Normal", "Distressed"), times = 3),
  Return = c(performance_k4rann_normal$return,
performance_k4rann_distressed$return,
performance_ga_normal$return,
performance_ga_distressed$return,
performance_xgboost_normal$return,
performance_xgboost_distressed$return),
  StdDev = c(performance_k4rann_normal$stddev,
performance_k4rann_distressed$stddev,
performance_ga_normal$stddev,
performance_ga_distressed$stddev,
performance_xgboost_normal$stddev,
performance_xgboost_distressed$stddev),
  SharpeRatio = c(performance_k4rann_normal$sharpe_ratio,
performance_k4rann_distressed$sharpe_ratio,
performance_ga_normal$sharpe_ratio,
performance_ga_distressed$sharpe_ratio,
performance_xgboost_normal$sharpe_ratio,
performance_xgboost_distressed$sharpe_ratio)
)

```

```
# Print comparison
```

```

print(comparison)
#
if(!require(devtools)){install.packages("devtools")}
if(!require(r2excel)){devtools::install_github("kassambara/r2excel")}
xlsx.writeMultipleData("Analytics-Consultina Results.xlsx"
, comparison
, df_weights_k4rann_normal
, df_weights_k4rann_distressed
, df_weights_ga_normal

```

```

        ,df_weights_ga_distressed
        ,df_weights_xgboost_normal
        ,df_weights_xgboost_distressed
    )

xlsx.openFile("Analytics-Consultina Results.xlsx")# view the file
}

#=====
#    MODEL 7:  Option 2
#=====

{
    ProjectFilePath = "H:/My Drive/Projects/zz_projects/Mutemeri_Linah/ETF/"
    load(file="H:/My
Drive/Programming_templates/AnalyticsCockpitFireUpCleanNot.rda")
    AnalyticsCockpitFireUpCleanNot(c())
    load(file=paste0(ProjectFilePath,"data_explo.rda"))
#
    v = names(data_explo[c(2:34)])
    seerecode(data_explo[c(v)])
    returns = data_explo %>%mutate_at(vars(names(.[c(v)])),list(function(x) {
        ifelse(is.na(x), 0,x)
        )))%>%
        select_if(names(.) %nin%
c('nvestMSCIEMAsiaETFETFEMA'));seesummary(returns[c(v)])

    row.names(returns) <- returns$date
    returns[1] <- NULL
    returns = returns %>% as.xts()

    returns <- na.omit(returns)

#-----
##Data Collection and Preparation
#1Obtain Historical ETF Data:
#Use financial data APIs like Yahoo Finance, Alpha Vantage, or Quandl to
#collect ETF data covering the period of interest.
#Example code to download data from Yahoo Finance:
#library(quantmod)
#getSymbols("SPY", src = "yahoo", from = "2018-01-01", to = #"2023-12-31")
#etf_data <- na.omit(SPY)
#2Define Distressed and Normal States:

```

```

#Segment the data into "distressed state" (during COVID-19: January 2020 -
#December 2021) and "normal state" (after COVID-19: January 2022 #onwards).
#Example code to segment data:
normal_state <- returns["2017-01/2020-2"]

distressed_state <- returns["2020-3/2022-07"]
#2Preprocess the Data:
#Handle missing values (e.g., using linear interpolation or removing rows
with #NA values).
#Normalize or standardize the data for consistent scale.
#Create any necessary features such as returns, volatility, etc.
#Example code for preprocessing:
library(quantmod)
preprocess_data <- function(data) {
  returns <- dailyReturn(data)
  volatility <- runSD(returns, n = 30)
  data <- na.omit(data.frame(returns, volatility))
  return(data)
}

distressed_data <- preprocess_data(distressed_state)
normal_data <- preprocess_data(normal_state)
#2Implementing the Machine Learning Techniques for portfolio #optimization
#2.1K4-RANN
#Packages: kkn, RSNNS
#Process:
#Implement K-Nearest Neighbors (KNN) and Radial Basis Function Neural
#Network (RBF-NN) separately.
#Combine outputs from KNN and RBF-NN.
#Train KNN and RBF-NN
#Train K-Nearest Neighbors and Radial Basis Function Neural Networks on the
data.
library(kkn)
library(RSNNS)

# Train KNN
train_knn <- function(data) {
  knn_model <- train.kkn(daily.returns ~ volatility, data = data, kmax =
10)
  return(knn_model)
}

```

```

# Train RBF-NN
train_rbf <- function(data) {
  rbf_model <- rbf(as.matrix(data$volatility), data$daily.returns, size =
10, maxit = 100)
  return(rbf_model)
}

knn_model_distressed <- train_knn(distressed_data)
knn_model_normal <- train_knn(normal_data)

rbf_model_distressed <- train_rbf(distressed_data)
rbf_model_normal <- train_rbf(normal_data)
#Combine KNN and RBF-NN Outputs
#Combine the outputs of KNN and RBF-NN to optimize the portfolio.
# Combine KNN and RBF outputs
combine_knn_rbf <- function(knn_model, rbf_model, data) {
  knn_pred <- predict(knn_model, data)
  rbf_pred <- predict(rbf_model, as.matrix(data$volatility))
  combined_pred <- (knn_pred + rbf_pred) / 2
  return(combined_pred)
}

# Fitness function for portfolio optimization
fitness_function <- function(weights, returns) {
  portfolio_return <- sum(weights * returns)
  portfolio_risk <- sqrt(sum((weights^2) * (returns^2)))
  return (portfolio_return / portfolio_risk) # Sharpe ratio as fitness
function
}

optimize_portfolio <- function(data, combined_pred) {
  returns <- combined_pred
  n <- length(returns)
}###

combined_pred_distressed <- combine_knn_rbf(knn_model_distressed,
rbf_model_distressed, distressed_data)
combined_pred_normal <- combine_knn_rbf(knn_model_normal, rbf_model_normal,
normal_data)

knn_rbf_pred_distressed = combined_pred_distressed

```

```

knn_rbf_pred_normal = combined_pred_normal

#2.2Evolutionary (Genetic) Algorithms
#Packages: GA, genalg
#Process:
#Define the fitness function to evaluate portfolio returns.
  #Use the ga function from GA package to optimize the portfolio.
  #Example code:
library(GA)

# Objective function to minimize (negative Sharpe ratio)
objective_function <- function(weights, returns) {
  portfolio_return <- sum(colMeans(returns) * weights)
  portfolio_stddev <- sqrt(t(weights) %*% cov(returns) %*% weights)
  sharpe_ratio <- portfolio_return / portfolio_stddev
  return(-sharpe_ratio) # Minimize negative Sharpe ratio
}

# Run GA for normal period
ga_model_normal <- ga(type = "real-valued",
  fitness = function(w) objective_function(w,
normal_data),
  lower = rep(0, ncol(normal_data)),
  upper = rep(1, ncol(normal_data)),
  maxiter = 500)

# Run GA for distressed data
ga_model_distressed <- ga(type = "real-valued",
  fitness = function(w) objective_function(w,
distressed_data),
  lower = rep(0, ncol(distressed_data)),
  upper = rep(1, ncol(distressed_data)),
  maxiter = 500)

# Get optimized weights
weights_ga_normal <- ga_model_normal@solution
weights_ga_distressed <- ga_model_distressed@solution
sum(weights_ga_normal)

predict_ga <- function(model, data) {
  best_weights <- model@solution[1, ]

```

```

combined_pred <- rowSums(sweep(data[, -1], 2, best_weights, `*`))
return(combined_pred)
}

predict_ga <- function(model, data) {
  #model = ga_model_distressed
  data_a = data[, -1] %>% as.data.frame() %>% setnames(., '.', names(data)[2])
  best_weights <- model@solution[1, ]
  data_a[['x1']] = best_weights[1]
  data_a[['x2']] = best_weights[2]
  #names(data_a)
  ga_pred_normal_a = data_a %>% mutate(
    combined_pred = volatility*(x1 + x2)
  )
  ga_pred_normal = ga_pred_normal_a[['combined_pred']]
  #combined_pred <- rowSums(sweep(data, 2, best_weights, `*`))

  return(combined_pred)
}

ga_pred_normal <- predict_ga(ga_model_normal, normal_data)
ga_pred_distressed <- predict_ga(ga_model_distressed, distressed_data)

weights_ga_normal_unnamed = as.numeric(weights_ga_normal)
names(weights_ga_normal_unnamed) = names(normal_data)
df_weights_ga_normal = weights_ga_normal_unnamed %>% as.data.frame() %>%
rownames_to_column() %>%
  rename(
    'Assets' = 'rowname'
    , 'Weights_ga_normal' = '.'
  )

weights_ga_distressed_unnamed = as.numeric(weights_ga_distressed)
names(weights_ga_distressed_unnamed) = names(distressed_data)
df_weights_ga_distressed = weights_ga_distressed_unnamed %>% as.data.frame()
%>% rownames_to_column() %>%
  rename(
    'Assets' = 'rowname'
    , 'Weights_ga_distressed' = '.'
  )

```

```

#2.3Extreme Gradient Boosting
#Packages: xgboost
#Process:
#Prepare data matrices for xgboost.
#Train the XGBoost model on training data.
#Example code:
#Train and Optimize XGBoost
  #Train the XGBoost model on training data.
  #Example code:
library(xgboost)

train_xgb <- function(data) {
  dtrain <- xgb.DMatrix(data = as.matrix(data[, -1]), label = data[, 1])
  params <- list(objective = "reg:squarederror", eta = 0.1, max_depth = 6)
  xgb_model <- xgb.train(params, dtrain, nrounds = 100)
  return(xgb_model)
}

xgb_model_distressed <- train_xgb(distressed_data)
xgb_model_normal <- train_xgb(normal_data)

predict_xgb <- function(model, test_data) {
  dtest <- xgb.DMatrix(data = as.matrix(test_data[, -1]), label =
test_data[, 1])
  predictions <- predict(model, dtest)
  return(predictions)
}

xgb_pred_distressed <- predict_xgb(xgb_model_distressed, distressed_data)
xgb_pred_normal <- predict_xgb(xgb_model_normal, normal_data)
#3Evaluating the Techniques
#Performance Metrics: Use metrics such as Sharpe ratio, Sortino ratio,
maximum #drawdown, etc.
#Comparison: Compare the performance of each technique during the distressed
and #normal states.
#Example code for performance evaluation:
performance_metrics <- function(actual, predicted) {

```



```

mse <- mean((actual - predicted)^2)
mae <- mean(abs(actual - predicted))
return(list(MSE = mse, MAE = mae))
}

# Compute metrics for each model in distressed and normal states
knn_rbf_metrics_distressed <- performance_metrics(distressed_data[,1],
knn_rbf_pred_distressed)
knn_rbf_metrics_normal <- performance_metrics(normal_data[,1],
knn_rbf_pred_normal)

xgb_metrics_distressed <- performance_metrics(distressed_data[,1],
xgb_pred_distressed)
xgb_metrics_normal <- performance_metrics(normal_data[,1], xgb_pred_normal)

#ga_metrics_distressed <- performance_metrics(distressed_data[,1],
predict(ga_model_distressed, as.matrix(distressed_data[, -1])))
#ga_metrics_normal <- performance_metrics(normal_data[,1],
predict(ga_model_normal, as.matrix(normal_data[, -1])))
weights_ga_normal_num = unname(weights_ga_normal) %>% as.numeric()
weights_ga_distressed_num = unname(weights_ga_distressed) %>% as.numeric()

#performance_ga_normal <- performance_metrics(weights_ga_normal_num,
normal_data)
#performance_ga_distressed <- performance_metrics(weights_ga_distressed_num, distressed_data)

#4Analyzing the Results
#Visualization: Plot the performance of the portfolios over time.
#Statistical Analysis: Conduct hypothesis testing to determine if there are
significant #differences in performance across the techniques and market
conditions.
#Example code for visualization:
library(ggplot2)
#-----
# Normal
#-----

plot_data_normal <- data.frame(
  Time_a = (normal_data %>% rownames_to_column(var =
'Time_a'))[1]
  #Time = index(normal_data)
  , Actual = normal_data[,1]

```

```

      , KNN_RBF = knn_rbf_pred_normal
      , XGB = xgb_pred_normal
      , GA = ga_pred_normal
    ) %>% mutate(
      Time = as.Date(Time_a)
    )

ggplot(plot_data_normal, aes(x = Time)) +
  geom_line(aes(y = Actual, color = "Actual")) +
  geom_line(aes(y = KNN_RBF, color = "KNN_RBF")) +
  geom_line(aes(y = XGB, color = "XGB")) +
  geom_line(aes(y = GA, color = "GA")) +
  labs(title = "Portfolio Performance During Normal State"
    , y = "Returns"
    , color = "Legend")

#-----
# Distressed
#-----

plot_data_distressed <- data.frame(
  Time_a = (distressed_data %>% rownames_to_column(var
= 'Time_a'))[1]

  #Time = index(distressed_data)
  , Actual = distressed_data[,1]
  , KNN_RBF = knn_rbf_pred_distressed
  , XGB = xgb_pred_distressed
  , GA = ga_pred_distressed
) %>% mutate(
  Time = as.Date(Time_a)
)

ggplot(plot_data_distressed
  , aes(x = Time)) +
  geom_line(aes(y = Actual, color = "Actual")) +
  geom_line(aes(y = KNN_RBF, color = "KNN_RBF")) +
  geom_line(aes(y = XGB, color = "XGB")) +
  geom_line(aes(y = GA, color = "GA")) +

```

```

labs(title = "Portfolio Performance During Distressed State"
      , y = "Returns"
      , color = "Legend"
    )

} # END MODEL 7: Option 2


# use
load(file="H:/My
Drive/Programming_templates/AnalyticsConsultinaCockpit.rda")
seenames(returns2) #; seeNames(data_lab)
returns2 = data_explo %>% mutate(
  rowid = 1:dplyr::n()
)

AnalyticsConsultinaCockpit(
  dataframe = returns2
  , SubjectIDVars = c("rowid")
  , ExcludeVars = c()
  , SelectVars = c()
  , Group1ColFacetVar = NULL
  , Group2RowFacetVar = NULL #
  `Group2RowFacetVar`
  , XVar = NULL
  , YVar = NULL
  , GroupColorVar = NULL
  , ConflictsDetachAllpackages = FALSE
)

```

APPENDIX 3A: ETHICAL CLEARANCE LETTER



12 July 2024

Ms Linah Mutemeri (216076569)
School Of Acc: Economics&Fin
Westville

Dear Ms Linah Mutemeri,

Original application number: 00024355

Project title: Machine learning vs traditional portfolio optimisation techniques under different market conditions in exchange traded funds

Exemption from Ethics Review

In response to your application received on 11 July 2024, your school has indicated that the protocol has been granted **EXEMPTION FROM ETHICS REVIEW**.

Any alteration/s to the exempted research protocol, e.g., Title of the Project, Location of the Study, Research Approach and Methods must be reviewed and approved through an amendment/modification prior to its implementation. The original exemption number must be cited.

For any changes that could result in potential risk, an ethics application including the proposed amendments must be submitted to the relevant UKZN Research Ethics Committee. The original exemption number must be cited.

In case you have further queries, please quote the above reference number.

PLEASE NOTE:

Research data should be securely stored in the discipline/department for a period of 5 years.

I take this opportunity of wishing you everything of the best with your study.

Yours sincerely,

Prof Claire Lauren Vermaak
Academic Leader Research
School Of Acc: Economics&Fin

UKZN Research Ethics Office
Westville Campus, Govan Mbeki Building
Postal Address: Private Bag X54001, Durban 4000
Website: <http://research.ukzn.ac.za/Research-Ethics/>

Founding Campuses: Edgewood Howard College Medical School Pietermaritzburg Westville

INSPIRING GREATNESS

APPENDIX 3B: AMENDED ETHICAL CLEARANCE LETTER



13 December 2024

Ms Linah Mutemeri (216076569)
School Of Acc Economics&Fin
Westville

Dear Ms Linah Mutemeri,

Original application number: 00024355

Project title: Machine learning vs traditional portfolio optimisation techniques under different market conditions in exchange traded funds

Amended title: Machine learning vs traditional portfolio optimisation under different market conditions in exchange traded fund portfolios

Exemption from Ethics Review

In response to your **amendment** application received on 12 December 2024, your school has indicated that the amendment has been granted **EXEMPTION FROM ETHICS REVIEW**.

Any alteration/s to the exempted research protocol, e.g., Title of the Project, Location of the Study, Research Approach and Methods must be reviewed and approved through an amendment/modification prior to its implementation. The original exemption number must be cited.

For any changes that could result in potential risk, an ethics application including the proposed amendments must be submitted to the relevant UKZN Research Ethics Committee. The original exemption number must be cited.

In case you have further queries, please quote the above reference number.

PLEASE NOTE:

Research data should be securely stored in the discipline/department for a period of 5 years.

I take this opportunity of wishing you everything of the best with your study.

Yours sincerely,



Prof Claire Lauren Vermaak
Academic Leader Research
School Of Acc Economics&Fin

UKZN Research Ethics Office
Westville Campus, Govan Mbeki Building
Postal Address: Private Bag X54001, Durban 4000
Website: <http://research.ukzn.ac.za/Research-Ethics/>

Founding Campuses: Edgewood Howard College Medical School Pietermaritzburg Westville

INSPIRING GREATNESS